

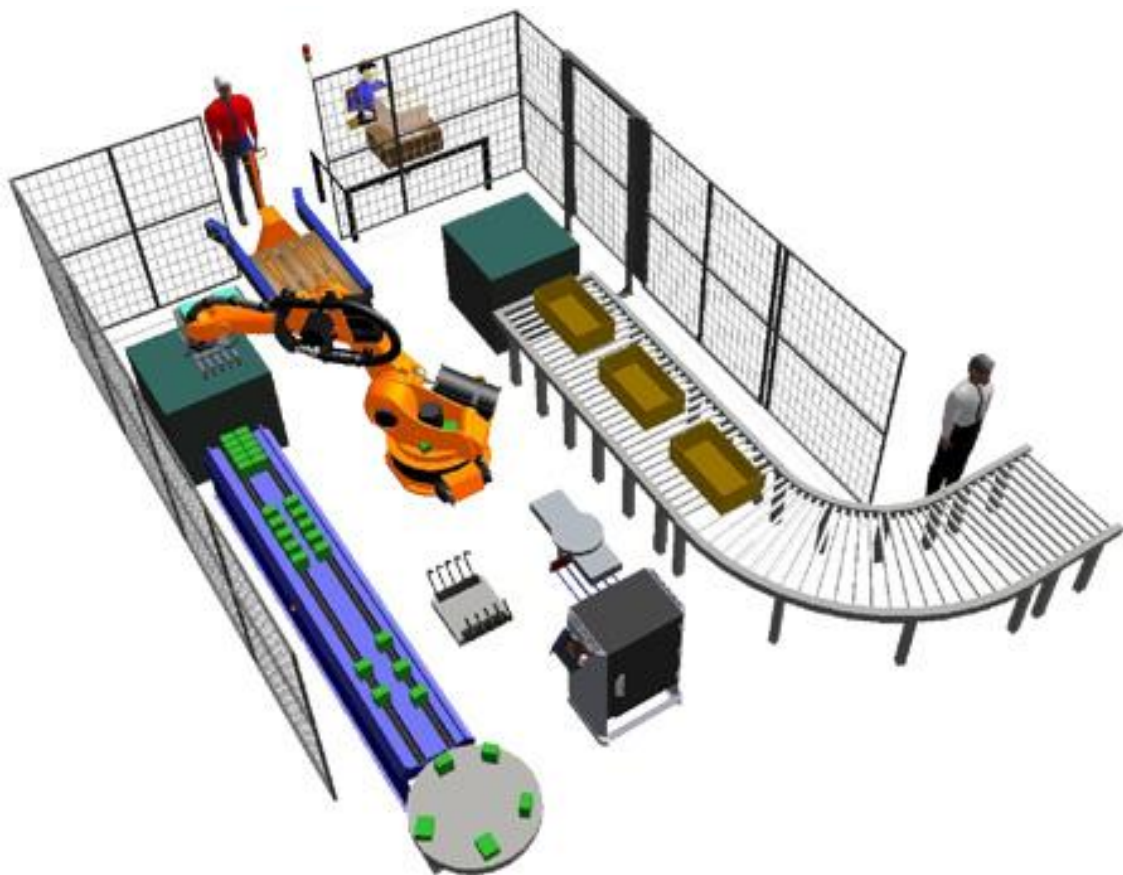
Frédéric Gilet

1997-2015

MES RAPPORTS DE STAGE

Technicien et ingénieur

Automatisme et informatique Industrielle

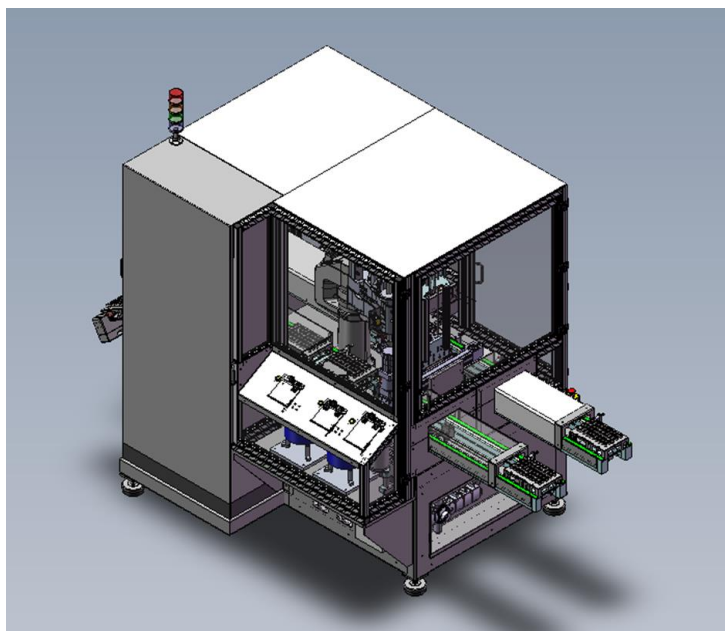


GILET Frédéric

TSAII

Session 2014-2015
Académie de Nantes

MACHINE DE DEPOSE DE RESSORTS SUR DES NOZZLES RAPPORT DE STAGE TECHNICIEN



Fast Intégration

461, rue Saint-Léonard, 49000 ANGERS
TUTEUR ENTREPRISE : M. Xavier Latrouite



GRETA Loire-Atlantique

16, rue Dufour, 44042 NANTES
TUTEUR GRETA : M. Christophe Oligo

SOMMAIRE

SOMMAIRE	2
INTRODUCTION	0
PRÉSENTATION DE L'ENTREPRISE	1
1. Situation géographique	1
2. Organigramme	1
3. Fiche d'identité	2
PRESENTATION DU PROJET	3
1. L'intérêt	3
2. Les étapes.	3
3. Définition des modes de marche	3
4. Cahier des charges des IHM	3
5. Cahier des charges de l'IHM PC	4
ARCHITECTURE DU RESEAU	5
1. Ensemble	5
2. Liaison Applicom	6
a) Console	6
b) Adressage	7
c) Les commandes Visual Basic pour Applicom	8
d) Test de la liaison Applicom entre l'automate et Visual Basic	12
e) Code général permettant de lire et écrire d'après le protocole Applicom	14
ARCHITECTURE DE LA DB	18
IHM	19
1. Fenêtres existantes	19
2. Fenêtres que j'ai créées	23
3. Code que j'ai créé	25
a) Modifications dans les programmes existants pour apporter le protocole Applicom	25
b) Ecriture et lecture de chaînes de caractères	25
TRADUCTION EN ANGLAIS	27
CONCLUSION	30
SOURCES	30
ANNEXES	31



Machine de dépose de ressorts sur des nozzles



1. Annexe 1 : Déclarations pour les DB	31
2. Annexe 2 : Code la fenêtre SuiviProd2	48
3. Annexe 3 : Fichier des langues Affichage.ini	55

INTRODUCTION

Le but de ce stage est de participer à la mise en place de l'automatisme d'une machine de dépose automatique de ressorts sur des nozzles. Le client direct est la société AMS (Anjou Machines Spéciales), le client final est Delphi, un sous-traitant automobile à portée internationale.

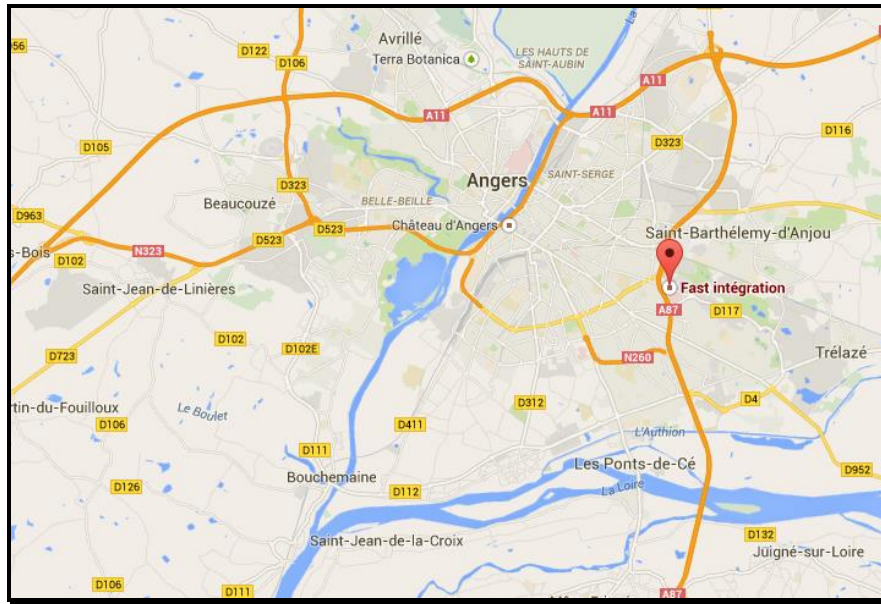
A partir de l'existant, il faut refaire l'IHM et la Partie Commande pour installer l'ensemble de la machine en Roumanie. L'existant marche avec une liaison OPC, il faut le refaire avec une liaison Applicom.

La description du fonctionnement de la machine est la suivante : les cassettes de cinquante nozzles arrivent sur deux tapis roulants. Les ressorts, selon trois références, sont stockés dans des bols vibrants et amenés par des rails vibrants. Le robot STAUBLI, à quatre axes, les prend un par un sur un plateau et les dépose sur les nozzles. Sa tête est équipée d'une caméra pour le contrôle qualité. On détecte ainsi les poses défectueuses. Lorsque la cassette est terminée, elle est évacuée.

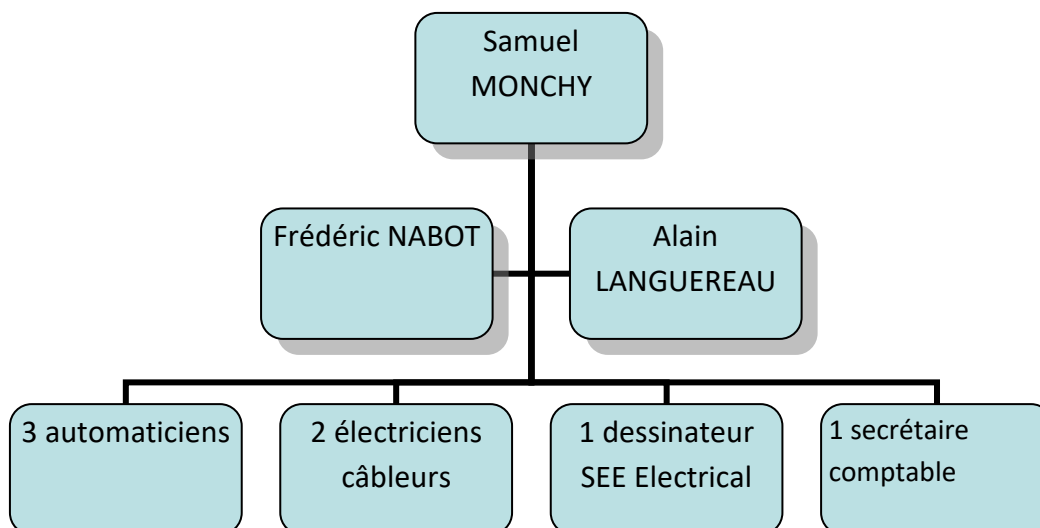
Dans ce projet, je dois m'occuper de la partie supervision avec une IHM en Visual Basic et de la gestion des variables communiquées entre un automate Siemens S7-300 et le PC par une liaison Applicom.

PRÉSENTATION DE L'ENTREPRISE

1. Situation géographique



2. Organigramme



Mon maître de stage est XavierLatrouite, automaticien.

3. Fiche d'identité

La société Fast Intégration est un bureau d'étude intégrateur en électricité, automatisme, robotique, vision et informatique industrielle. Elle mène des projets de l'étude à la mise en service sur site.

Elle est dirigée par ses fondateurs, qui l'ont créée en 2004 et compte aujourd'hui dix salariés.

Elle a notamment réalisé une ligne de fabrication de paraboles, une ligne de montage de cordons de recharge des véhicules électriques, la gestion électrique d'un parking, une ligne de montage autoradios, un palettiseur de cartons de vin, une ligne de convoyage. Elle travaille dans le bâtiment et le froid industriel.

PRESENTATION DU PROJET

1. L'intérêt

L'intérêt du projet est de reprendre et refaire une supervision de la machine avec le logiciel Microsoft Visual Basic.Net et de communiquer des informations avec l'automate par une liaison Applicom.

2. Les étapes.

Les étapes sont les suivantes :

- Reprise des fenêtres existantes et d'une partie du code
- Test de la liaison Applicom (Ethernet Industriel) avec un petit programme en VB
- Programmation de l'interface et de la liaison.

3. Définition des modes de marche

Mode automatique :

- Arrêt cycle automatique : finition de la cassette, vidage plateau, sortie mode automatique
- Mode réglage : commandes manuelles
- Mode initialisation
- Ejection cassette en cours de traitement en mode automatique
- Vidage plateau ressorts : vidage complet des ressorts des huit positions du plateau.

Le cycle automatique est le suivant :

- Convoyage des cassettes en approvisionnement
- Approvisionnement des ressorts dans le bol vibrant (vérifier la bonne référence, verrouillage trappe)
- Approvisionnement des ressorts sur le plateau
- Disposition des ressorts sur les nozzles par la tête de robot
- Evacuation des cassettes travaillées

4. Cahier des charges des IHM

Il y a un pupitre de commande avec un bloc acquittement défaut et un choix de mode (automatique, initialisation, réglage) et le lancement des cycles de contrôle.

Il y a un pupitre de contrôle des cassettes en fin de ligne, avec lecture Datamatrix et requête base de données.

Il y a enfin un pupitre de supervision, en VB. Elle communique avec le PLD par une passerelle Applicom en Industrial Ethernet. Il y a une communication entre la supervision et une base de données Oracle installée sur le PC.

La sécurité est gérée par des boutons d'arrêts d'urgence (physique et écran supervision) et des capteurs de porte ouverte.

5. Cahier des charges de l'IHM PC

Visualiser l'état de l'équipement pour son exploitation :

- Gestion de la production (démarrage, arrêt, type de cycle, etc...)
- Suivi de la production
- Intégration des valeurs d'entrées/sorties d'un processus
- Paramétrage des équipements
- Gestion des données

L'IHM gèrera plusieurs DB en interaction avec l'automate :

- Ecran principal
- Les paramètres machines
- Le suivi de production
- Les informations diagnostic
- Les mouvements manuels
- Les défauts
- La gestion des séries
- La visualisation des séries
- La cartographie

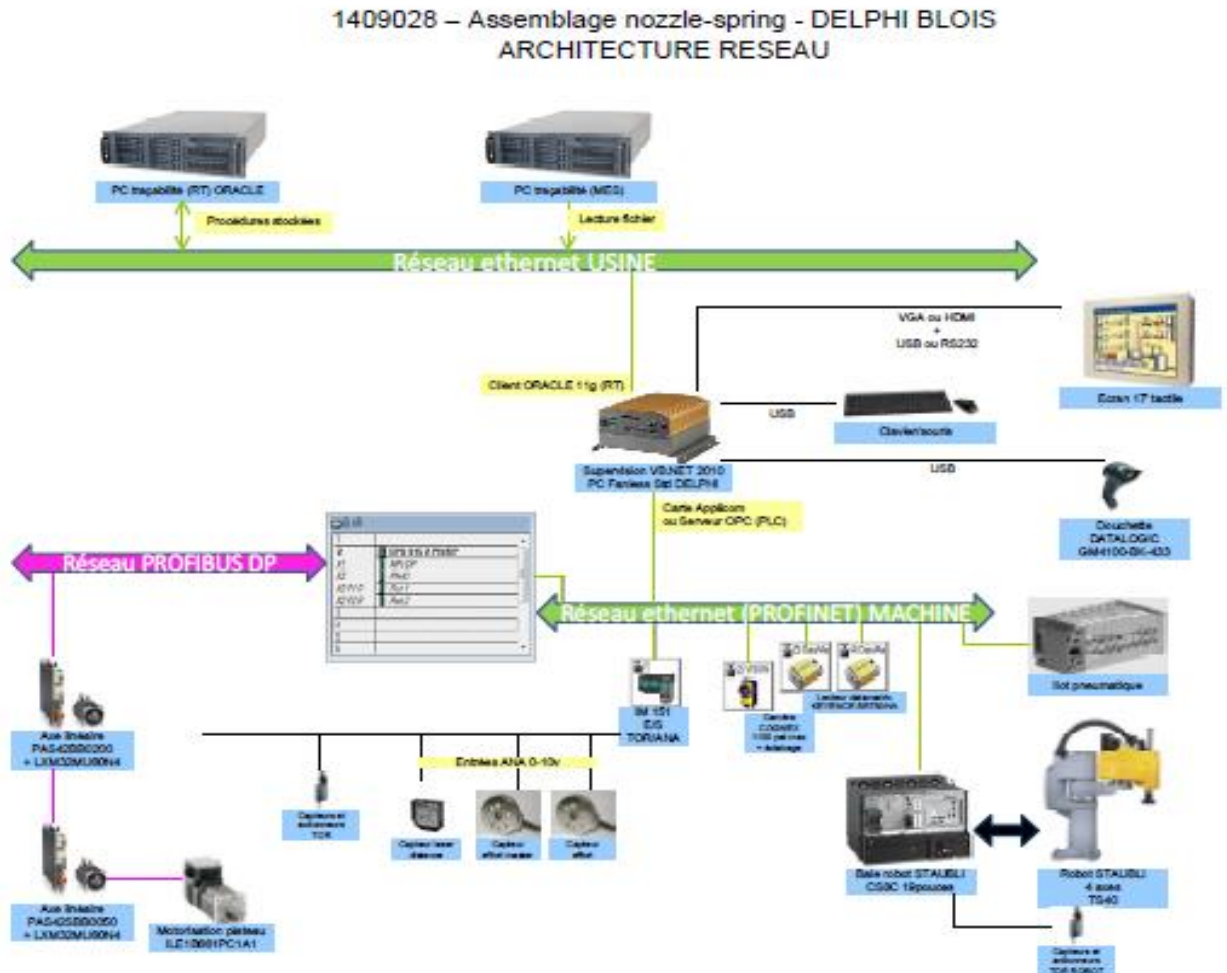
Il y a un niveau opérateur et un niveau régleur

L'IHM contiendra :

- Une page principale
- Une page diagnostics
- Une page cartographie
- Une page états périphériques
- Une page états actionneurs
- Une page paramétrage
- Une page suivi production

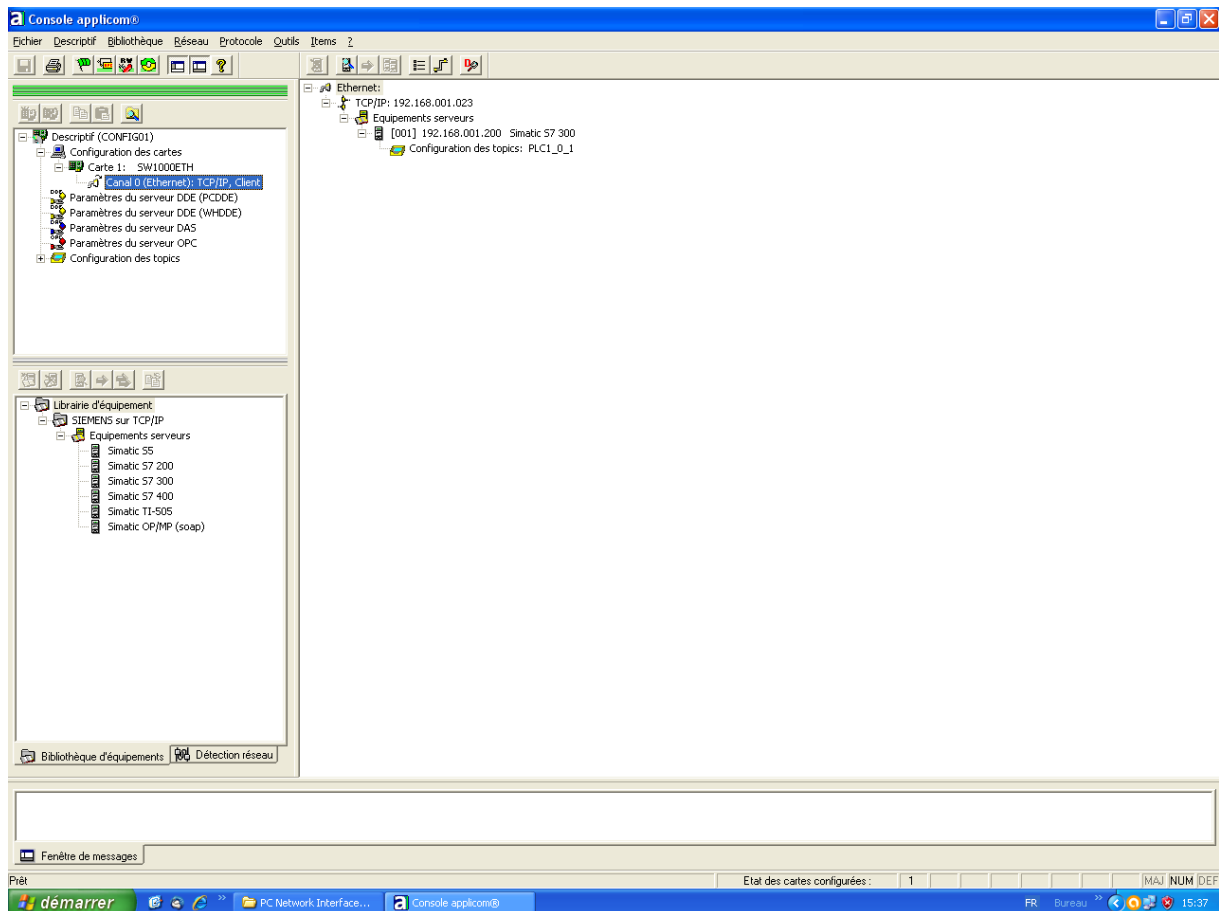
ARCHITECTURE DU RESEAU

1. Ensemble



2. Liaison Applicom

a) Console



C'est de l'Ethernet Industriel.

On configure sur la console de l'application la carte SW1000ETH sur le canal 0 équipement 1. Il y a une adresse IP pour le PC et une adresse IP pour l'automate S7-300.

On lance ensuite la liaison à chaque fois qu'on veut l'utiliser à partir de Visual Basic selon les codes définis dans les pages suivantes.

b) Adressage

- Fonctionnalité

- Client Industrial Ethernet

- Mode d'adressage

Les fonctions gérées par le choix d'un canal Industrial Ethernet et d'un équipement S7 permettent l'accès aux variables définies dans le tableau ci-dessous.

L'adresse de la variable accédée doit être dans certains cas calculée en faisant référence à la colonne « Adressage applicom® » du tableau.

Le descripteur d'item utilisable en PCODE associé à cet adressage est "Siemens Série 7".

Variable S7-300 S7-400	Adressage applicom®	Type d'octets	Fonction applicom® (accès libérés/PLC)
Mx.z DBx.DBXy.z	y*8+z x*524288+y*8+z	Lecture de bits Ecriture de bits	READPACKBIT WRITEPACKBIT
Bx.z	x*8+z	Lecture de bits d'entrée	READPACKBIT
Ay.z	y*8+z	Lecture de bits de sortie Ecriture de bit de sortie	READPACKBIT WRITEPACKBIT
MBy DBx.DBBy	y x*524288+y	Lecture d'octets Ecriture d'octets	READPACKBYTE, READBYTE WRITEPACKBYTE, WRITEBYTE
EBy	x	Lecture d'octets d'entrée	READPACKBYTE, READBYTE
ABy	y	Lecture d'octets de sortie Ecriture d'octets de sortie	READPACKBYTE, READBYTE WRITEPACKBYTE, WRITEBYTE
MWy DBx.DMWy	y x*524288+y	Lecture de mots Ecriture de mots	READWORD WRITEWORD
EWy	x	Lecture de mots d'entrée	READWORD
AWy	y	Lecture de mots de sortie Ecriture de mots de sortie	READWORD WRITEWORD
MDy DBx.DBDy	y x*524288+y	Lecture de mots doubles Ecriture de mots doubles	READDWORD WRITEDWORD
MDyF DBx.DBDyF	y x*524288+y	Lecture de mots flottants Ecriture de mots flottants	READFWORD WRITEFWORD
T n	n	Lecture de temporisateurs Ecriture de temporisateurs	READTIMER WRITETIMER
Z n	n	Lecture de compteurs Ecriture de compteurs	READCOUNTER WRITECOUNTER

x : Numéro du DB

y : Numéro d'octet.

z : Numéro du bit dans l'octet (0 à 7).

n : Numéro du temporisateur ou du compteur (0 à 230)

Par exemple, pour lire le bit DB1.DBX 20.1 on se positionnera à l'adresse 524449 (DB * 524288 + Operande * 8 + Bit). Par contre, la DB doit être configurée en déclarant les variables.

c) Les commandes Visual Basic pour Applicom

Voici les différentes commandes qu'on peut effectuer avec Applicom à partir d'un langage de programmation. Sous Visual Basic, on utilise la DLL ApplicomWra.

Product initialization => INITBUS

Prototypes

Language	Prototype
C	void initbus (short* wStatus)
MS pascal	initbus (var wStatus : integer)
C#	void initbus (out short wStatus)
VB.NET	initbus (ByRef wStatus As short)

Parameters

Parameter	Type
wStatus	Error status

Initialiser la liaison Applicom

Set the maximum number of variable => AuSetApplicationMaxSize

Prototypes:

Language	Prototype
C	BOOL AuSetApplicationMaxSize (short wBufferSize , unsigned long* dwStatus)
MS pascal	function AuSetApplicationMaxSize (wBufferSize : SmallInt, var dwStatus : LongInt) : boolean
C#	bool AuSetApplicationMaxSize (ushort wBufferSize , out int dwStatus)
VB.NET	Sub AuSetApplicationMaxSize (ByVal wBufferSize As UShort, ByRef dwStatus As Integer)

Parameters

Parameter	Type
wbuffersize	Indicate if the application wishes to use the new limits of the applicom [®] library <ul style="list-style-type: none"> • APP_1584BYTES_BASED_LIMITS Utilization of the news limits. • APP_256BYTES_BASED_LIMITS Utilization of the old limits.
dwStatus	Error status.

Configurer la liaison Applicom

Program end => EXITBUS

Prototypes:

Language	Prototype
C	void exitbus (short * <i>wStatus</i>)
MS pascal	procedure exitbus (var <i>wStatus</i> : integer)
C#	void exitbus (out short <i>wStatus</i>)
VB.NET	Sub exitbus (ByRef <i>wStatus</i> As Short)

Parameters

Parameter	Type
<i>wStatus</i>	Not used (always 0)

Description

Fermer la liaison Applicom

Codes de lecture/écriture des données de l'automate vers une application Visual Basic (exemple).

Bit reading => READPACKBIT

Prototypes:

Language	Prototype
C	void readpackbit (short * <i>wChan</i> , short * <i>wNes</i> , short * <i>wNb</i> , long* <i>dwAdr</i> , short * <i>lpwTabl</i> , short * <i>wStatus</i>)
MS pascal	procedure readpackbit (var <i>wChan</i> , <i>wNes</i> , <i>wNb</i> : integer; var: <i>dwAdr</i> integer4; var <i>lpwTabl</i> , <i>wStatus</i> : integer)
C#	void readpackbit (ref ushort <i>wChan</i> , ref ushort <i>wNes</i> , ref short <i>wNb</i> , ref int <i>dwAdr</i> , short[] <i>lpwTabl</i> , out short <i>wStatus</i>)
VB.NET	Sub readpackbit (ByRef <i>wChan</i> As UShort, ByRef <i>wNes</i> As UShort, ByRef <i>wNb</i> As Short, ByRef <i>dwAdr</i> As Integer, ByVal <i>lpwTabl</i> () As Short, ByRef <i>wStatus</i> As Short)

Parameters

Parameter	Type
<i>wChan</i>	Channel number (0-31). Must be 0.
<i>wNes</i>	Device number (0-255)
<i>wNb</i>	Number of bits to be read (1-2048). The maximum number depends on the protocol and target device.
<i>dwAdr</i>	Address of the first bit to be read in the device.
<i>lpwTabl</i>	Table receiving read data.
<i>wStatus</i>	Error status.

Word reading => READWORD

Prototypes:

Language	Prototype
C	void readword(short * wChan, short * wNes, short * wNb, long * dwAdr, short * lpwTabl, short * wStatus)
MS pascal	procedure readword(var wChan, wNes, wNb:integer; var dwAdr:integer4; var lpwTabl, wStatus:integer)
C#	void readword(ref ushort wChan, ref ushort wNes, ref short wNb, ref int dwAdr, short[] lpwTabl, out short wStatus)
VB.NET	Sub readword(ByRef wChan As UShort, ByRef wNes As UShort, ByRef wNb As Short, ByRef dwAdr As Integer, ByVal lpwTabl() As Short, ByRef wStatus As Short)

Parameters

Parameter	Type
wChan	Channel number (0-31). Must be 0.
wNes	Device number (0-255).
wNb	Number of words to be read (1-128 max). The maximum number depends on the protocol and target device.
dwAdr	Address of the first word to be read.
lpwTabl	Table receiving read data.
wStatus	Error status.

- . . .

Bit writing => WRITEPACKBIT

Prototypes:

Language	Prototype
C	void writepackbit(short * wChan, short * wNes, short * wNb, long * dwAdr, short * lpwTabl, short * wStatus)
MS pascal	procedure writepackbit(var wChan, wNes, wNb:integer; var dwAdr:integer4, var lpwTabl, wStatus:integer);
C#	void writepackbit(ref ushort wChan, ref ushort wNes, ref short wNb, ref int dwAdr, short[] lpwTabl, out short wStatus)
VB.NET	Sub writepackbit(ByRef wChan As UShort, ByRef wNes As UShort, ByRef wNb As Short, ByRef dwAdr As Integer, ByVal lpwTabl() As Short, ByRef wStatus As Short)

Parameters

Parameter	Type
wChan	Channel number (0-31). Must be 0.
wNes	Device number (0-255).
wNb	Number of bits to be written (1-2048). The maximum number depends on the protocol and target device.
dwAdr	Address of the first bit to be written.
lpwTabl	Table containing the data.
wStatus	Error status.

Word writing => WRITEWORD

Prototypes:

Language	Prototype
C	void writeword (short * <i>wChan</i> , short * <i>wNes</i> , short * <i>wNb</i> , long * <i>dwAdr</i> , short * <i>lpwTabl</i> , short * <i>wStatus</i>)
MS pascal	procedure writeword (var <i>wChan</i> , <i>wNes</i> , <i>wNb</i> :integer; var <i>dwAdr</i> :integer4; var <i>lpwTabl</i> , <i>wStatus</i> :integer);
C#	void writeword (ref ushort <i>wChan</i> , ref ushort <i>wNes</i> , ref short <i>wNb</i> , ref int <i>dwAdr</i> , short[] <i>lpwTabl</i> , out short <i>wStatus</i>)
VB.NET	Sub writeword (ByRef <i>wChan</i> As UShort, ByRef <i>wNes</i> As UShort, ByRef <i>wNb</i> As Short, ByRef <i>dwAdr</i> As Integer, ByVal <i>lpwTabl</i> () As Short, ByRef <i>wStatus</i> As Short)

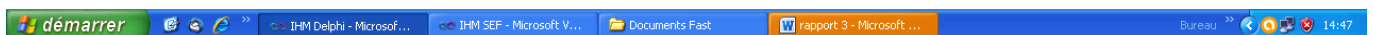
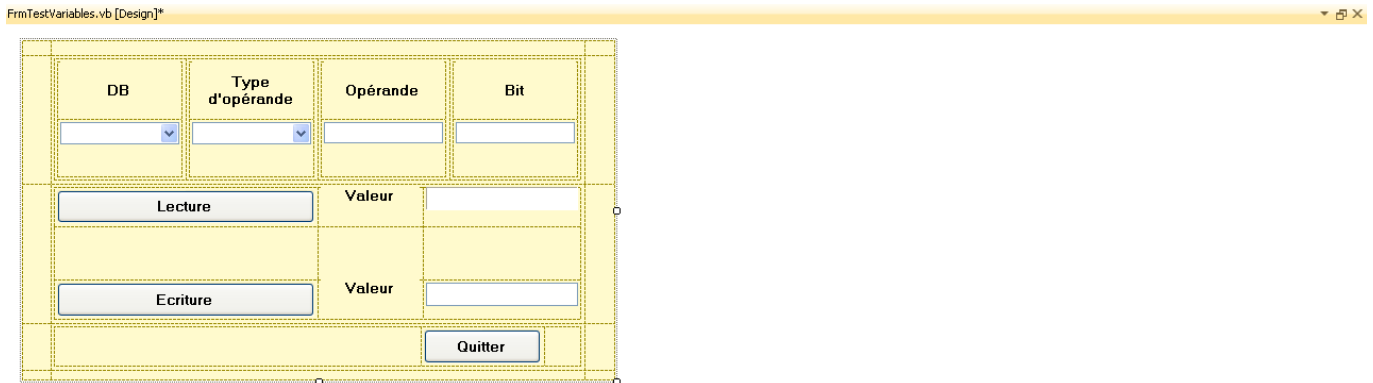
Parameters

Parameter	Type
wChan	Channel number (0-31). Must be 0.
wNes	Device number (0-255).
wNb	Number of words to be written (1-128). The maximum number depends on the protocol and target device.
dwAdr	Address of the first word to be written.
lpwTabl	Table containing the data.
wStatus	Error status.

On intègre ces commandes en Visual Basic pour lire ou écrire les mots, les bits, les doubles mots, les réels ou les bytes stockés dans la DB de l'automate.

d) Test de la liaison Applicom entre l'automate et Visual Basic

J'ai effectué un programme de tests de la liaison en Visual Basic. Il s'agit de lire et d'écrire des bits sur l'automate à partir de l'interface VB du PC.



Voici le code de cette fenêtre :

```
Private Sub ButtonLecture_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonLecture.Click
```

```
    Dim DBCalc As Integer  
    Dim OperandeCalc As Integer  
    Dim BitCalc As Integer  
  
    'Dim lpwTablBit() As Short  
    If (TextBoxOperande.Text <> "" And ComboBoxDB.Text <> "") Then  
        DBCalc = Convert.ToUInt32(ComboBoxDB.Text)  
        OperandeCalc = Convert.ToUInt32(TextBoxOperande.Text)
```

```

End If
If (TextBoxBit.Text <> "") Then
    BitCalc = Convert.ToUInt32(TextBoxBit.Text)
End If

Select Case ComboBoxType.Text
    Case "INT"
        'Mot
        If (TextBoxOperande.Text <> "" And ComboBoxDB.Text <> "") Then
            LectureMot(DBCalc, OperandeCalc, 1, TablMot)
            LabelLecture.Text = TablMot(0).ToString()
        End If
        'Reel
    Case "REAL"
        If (TextBoxOperande.Text <> "" And ComboBoxDB.Text <> "") Then
            LectureReel(DBCalc, OperandeCalc, 1, TablReel)
            LabelLecture.Text = TablReel(0).ToString()
        End If
    Case "DWORD"
        'double mot
        If (TextBoxOperande.Text <> "" And ComboBoxDB.Text <> "") Then
            LectureDMot(DBCalc, OperandeCalc, 1, TablDMot)
            LabelLecture.Text = TablDMot(0).ToString()
        End If

    Case "BOOL"
        'booleen
        If (TextBoxOperande.Text <> "" And TextBoxBit.Text <> "" And
ComboBoxDB.Text <> "") Then
            LectureBit(DBCalc, OperandeCalc, BitCalc, 1, TablBit)
            LabelLecture.Text = TablBit(0).ToString()

        End If
    End Select

```

End Sub

```

Private Sub ButtonEcriture_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ButtonEcriture.Click

```

```

    Dim DBCalc As Integer
    Dim OperandeCalc As Integer
    Dim BitCalc As Integer

    'Dim lpwTablBit() As Short
    If (TextBoxOperande.Text <> "" And ComboBoxDB.Text <> "") Then
        DBCalc = Convert.ToUInt32(ComboBoxDB.Text)
        OperandeCalc = Convert.ToUInt32(TextBoxOperande.Text)
    End If
    If (TextBoxBit.Text <> "") Then
        BitCalc = Convert.ToUInt32(TextBoxBit.Text)
    End If

    Select Case ComboBoxType.Text
        Case "INT"
            'Mot
            If (TextBoxOperande.Text <> "" And ComboBoxDB.Text <> "") Then
                TablMot(0) = Convert.ToInt16(TextBoxEcriture.Text)
                EcritureMot(DBCalc, OperandeCalc, 1, TablMot)
            End If
            'Reel

```

```

    Case "REAL"
        If (TextBoxOperande.Text <> "" And ComboBoxDB.Text <> "") Then
            TablReel(0) = Convert.ToSingle(TextBoxEcriture.Text)
            EcritureReel(DBCalc, OperandeCalc, 1, TablReel)
        End If
    Case "DWORD"
        'double mot
        If (TextBoxOperande.Text <> "" And ComboBoxDB.Text <> "") Then
            TablDMot(0) = Convert.ToInt16(TextBoxEcriture.Text)
            EcritureDMot(DBCalc, OperandeCalc, 1, TablDMot)
        End If

    Case "BOOL"
        'booleen
        If (TextBoxOperande.Text <> "" And TextBoxBit.Text <> "" And
ComboBoxDB.Text <> "") Then
            TablBit(0) = Convert.ToInt16(TextBoxEcriture.Text)
            EcritureBit(DBCalc, OperandeCalc, BitCalc, 1, TablBit)
        End If
    End Select

End Sub

#End Region

```

e) Code général permettant de lire et écrire d'après le protocole Applicom

Il sera fréquemment utilisé pour toute lecture ou écriture vers ou à partir de l'automate dans toute l'application VB Delphi IHM. Il simplifie l'accès Applicom à l'adresse de l'automate car la procédure utilise en entrée un numéro de DB, l'opérande des variables, éventuellement le n° de bit, le nombre de lectures ou écritures qu'on veut réaliser. En lecture, elle retourne par référence la (les) valeurs lues. En écriture, elle fournit la (les) valeurs à écrire.

```

#Region "Fonctions de lecture applicom"

' Procédure de lecture de bits
Public Sub LectureBit(ByVal DB As Integer, ByVal Operande As Integer, ByVal Bit As
Integer, ByVal NbBit As Short, ByRef TableauBit() As Short)
    Dim dwAdr As Integer

    dwAdr = DB * 524288 + Operande * 8 + Bit
    ReDim TableauBit(NbBit)

    AppliCom.readpackbit(wChan, wNes, NbBit, dwAdr, TableauBit, wStatus)

    If wStatus <> 0 Then
        MsgBox("Error : reading data", MsgBoxStyle.Critical)
    End If

End Sub

' Procédure de lecture de mots

```

```

Public Sub LectureMot(ByVal DB As Integer, ByVal Operande As Integer, ByVal NbMot
As Short, ByRef TableauMot() As Short)
    Dim dwAdr As Integer

    dwAdr = DB * 65536 + Operande
    ReDim TableauMot(NbMot)

    AppliCom.readword(wChan, wNes, NbMot, dwAdr, TableauMot, wStatus)

    If wStatus <> 0 Then
        MsgBox("Error : reading data", MsgBoxStyle.Critical)
    End If

End Sub

' Procédure de lecture de doubles mots
Public Sub LectureDMot(ByVal DB As Integer, ByVal Operande As Integer, ByVal
NbDMot As Short, ByRef TableauDMot() As Integer)
    Dim dwAdr As Integer

    dwAdr = DB * 65536 + Operande
    ReDim TableauDMot(NbDMot)

    AppliCom.readdword(wChan, wNes, NbDMot, dwAdr, TableauDMot, wStatus)

    If wStatus <> 0 Then
        MsgBox("Error : reading data", MsgBoxStyle.Critical)
    End If

End Sub

' Procédure de lecture de réels
Public Sub LectureReel(ByVal DB As Integer, ByVal Operande As Integer, ByVal
NbReel As Short, ByRef TableauReel() As Single)
    Dim dwAdr As Integer

    dwAdr = DB * 65536 + Operande
    ReDim TableauReel(NbReel)

    AppliCom.readfword(wChan, wNes, NbReel, dwAdr, TableauReel, wStatus)

    If wStatus <> 0 Then
        MsgBox("Error : reading data", MsgBoxStyle.Critical)
    End If

End Sub

#End Region

#Region "Fonctions d'écriture applicom"

' Procédure d'écriture de bits
Public Sub EcritureBit(ByVal DB As Integer, ByVal Operande As Integer, ByVal Bit
As Integer, ByVal NbBit As Short, ByVal TableauBit() As Short)
    Dim dwAdr As Integer

```

```

dwAdr = DB * 524288 + Operande * 8 + Bit
'ReDim TableauBit(NbBit)

AppliCom.writepackbit(wChan, wNes, NbBit, dwAdr, TableauBit, wStatus)

If wStatus <> 0 Then
    MsgBox("Error : writing data", MsgBoxStyle.Critical)
End If

End Sub

' Procédure d'écriture de mots
Public Sub EcritureMot(ByVal DB As Integer, ByVal Operande As Integer, ByVal NbMot
As Short, ByVal TableauMot() As Short)
    Dim dwAdr As Integer

    dwAdr = DB * 65536 + Operande
'ReDim TableauMot(NbMot)

    AppliCom.writeword(wChan, wNes, NbMot, dwAdr, TableauMot, wStatus)

    If wStatus <> 0 Then
        MsgBox("Error : writing data", MsgBoxStyle.Critical)
    End If

End Sub

' Procédure d'écriture de doubles mots
Public Sub EcritureDMot(ByVal DB As Integer, ByVal Operande As Integer, ByVal
NbDMot As Short, ByVal TableauDMot() As Integer)
    Dim dwAdr As Integer

    dwAdr = DB * 65536 + Operande
'ReDim TableauDMot(NbDMot)

    AppliCom.writedword(wChan, wNes, NbDMot, dwAdr, TableauDMot, wStatus)

    If wStatus <> 0 Then
        MsgBox("Error : writing data", MsgBoxStyle.Critical)
    End If

End Sub

' Procédure d'écriture de réels
Public Sub EcritureReel(ByVal DB As Integer, ByVal Operande As Integer, ByVal
NbReel As Short, ByVal TableauReel() As Single)
    Dim dwAdr As Integer

    dwAdr = DB * 65536 + Operande
'ReDim TableauReel(NbReel)

    AppliCom.writefword(wChan, wNes, NbReel, dwAdr, TableauReel, wStatus)

    If wStatus <> 0 Then
        MsgBox("Error : writing data", MsgBoxStyle.Critical)
    End If

End Sub

```

#End Region

ARCHITECTURE DE LA DB

Le but de mon stage n'est pas d'effectuer le programme automate. Néanmoins, je présente ici les différentes adresses de DB, d'opérande et éventuellement de bits de communication avec le S7 300. A l'origine, la communication était en OPC. J'ai transformé le programme pour utiliser Applicom. On déclare donc les numéros de DB, d'opérande et de bits dans VB avec l'adresse qui correspond à l'automate. Vous trouverez en Annexe 2 ce code.

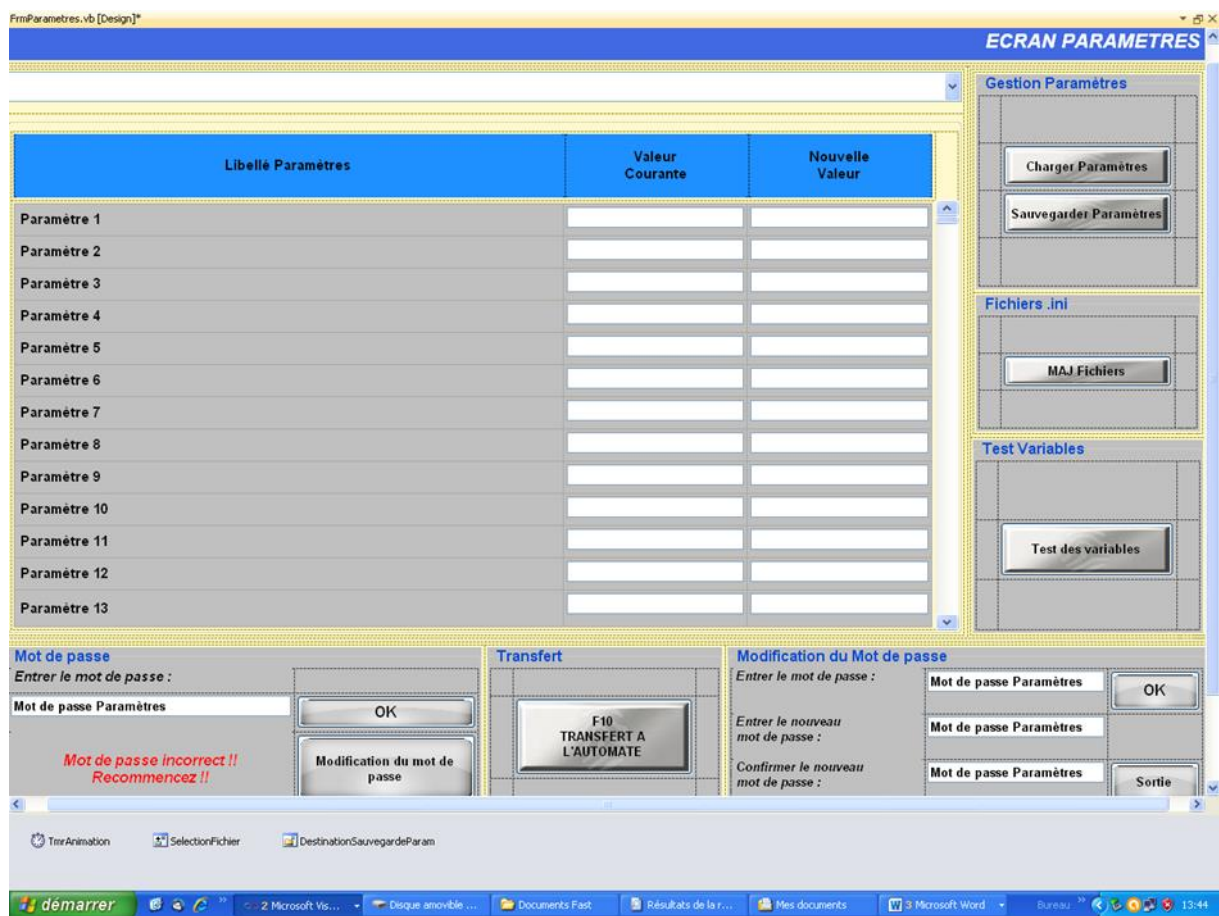
Pour que l'adresse des mots, des bytes, des bits, des doubles-mots ou des réels soit facile à inter-changer, j'ai créé des variables globales qui contiennent le numéro de DB, l'adressage de l'opérande et si nécessaire l'adressage du bit. Ainsi, lorsqu'on lit ou écrit à partir ou vers les DB, on transmet ces informations aux procédures spécifiques que j'ai créées qui calculeront automatiquement l'adresse via Applicom.

Les informations sur l'automate concernent le fonctionnement de la machine, l'état des cassettes, les caractéristiques de la production, des cycles ou des références ainsi que l'état des actionneurs, des entrées-sorties, des grafjets, etc...

IHM

C'est le cœur de mon stage. Vous trouverez les fenêtres que j'ai récupérées sur l'existant. L'essentiel de mon stage a consisté à adapter le code de la liaison OPC vers la liaison Applicom, puis de refaire la fenêtre de suivi de production avec ses compteurs. J'ai donc enlevé et rajouté du code, dont je ne mettrai un exemple que pour la fenêtre SuiviProd. J'ai fait des modifications dans toutes les fenêtres pour y intégrer Applicom à la place d'OPC.

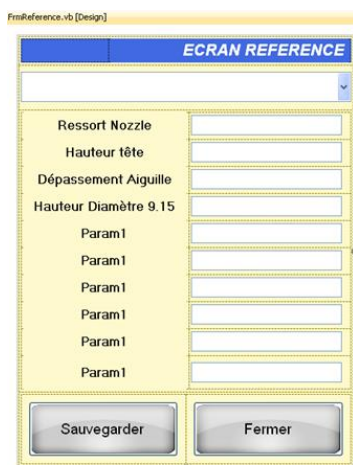
1. Fenêtres existantes



La fenêtre paramètres permet de configurer l'automate avec les paramètres machine.



La fenêtre E/S et Grafcet permettent d'identifier les entrées sorties et de savoir où l'automate est dans les Grafcets qui décrivent son action.



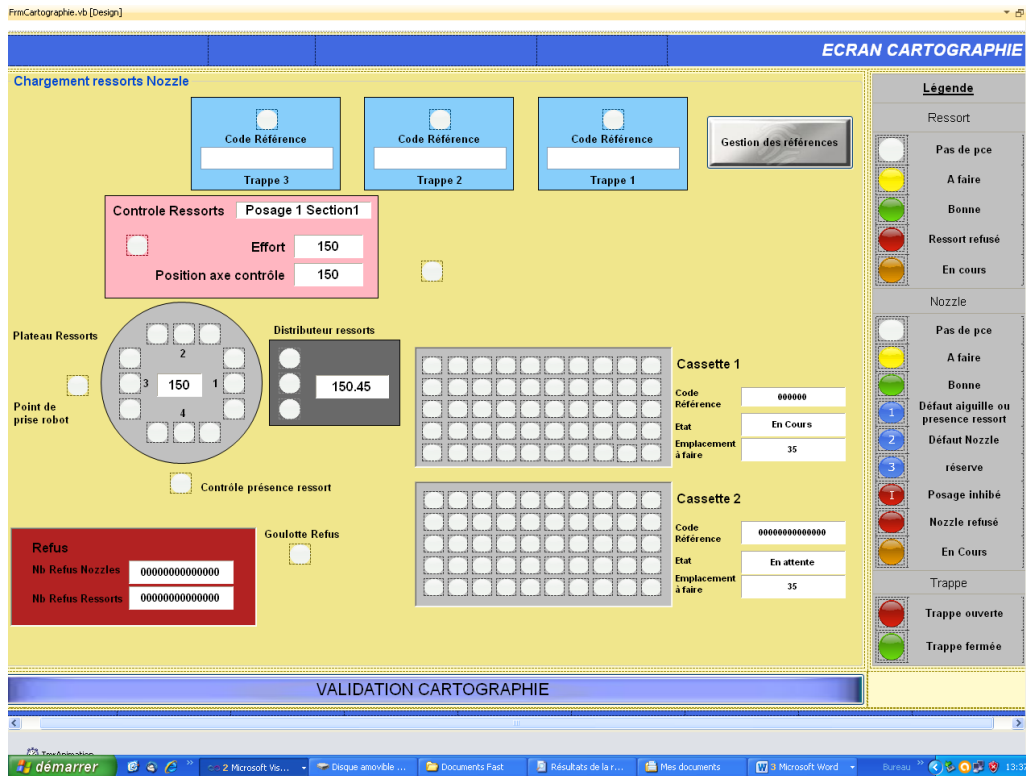
L'écran référence permet de connaître les caractéristiques mécaniques de la référence en cours.



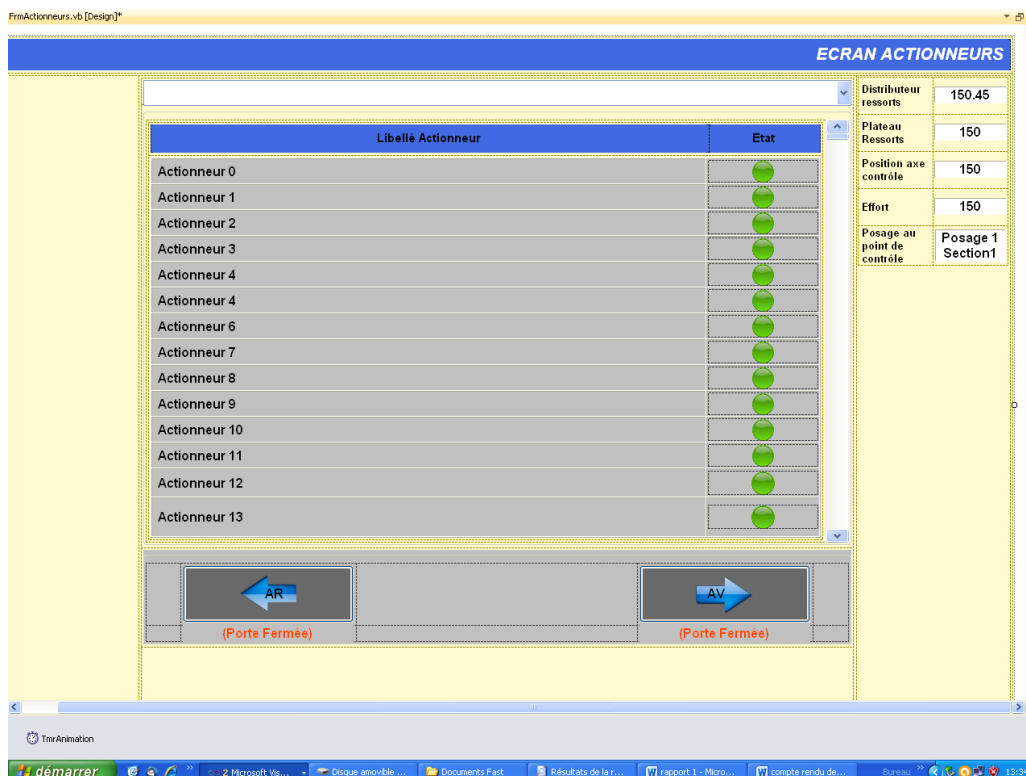
La fenêtre temps de cycle permet de connaître les durées de travail pour chaque poste.



C'est la fenêtre principale. C'est le tableau de bord qui permet de lancer le cycle auto, l'init, les réglages, la vitesse du robot, de provoquer l'arrêt fin de cycle ou l'acquitement, et de suivre en temps réel les défauts.



La fenêtre cartographie permet de visualiser en temps réel l'état du cycle de la machine : l'état des cassettes, du plateau, des trappes, etc.



La fenêtre actionneurs permet de visualiser l'état des actionneurs.

2. Fenêtres que j'ai créées

Tableau des compteurs de production

ECRAN SUIVI DE PRODUCTION

Tableau des compteurs de production

		Compteur Totalisateur :	
		1000	
		Compteur Totalisateur avec RAZ :	
		1000	

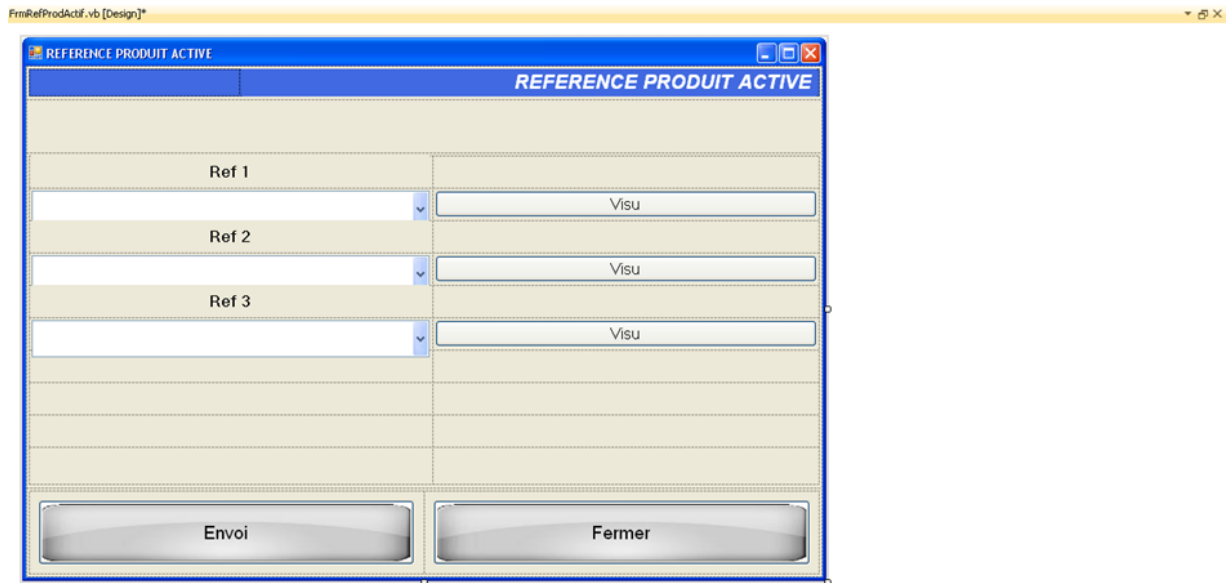
General	OK	Not OK	Process	OK	Not OK
Cassette	<input type="checkbox"/>	<input type="checkbox"/>	Mapping Cassette	<input type="checkbox"/>	<input type="checkbox"/>
Nozzle	<input type="checkbox"/>	<input type="checkbox"/>	Controle Spring	<input type="checkbox"/>	<input type="checkbox"/>
			Spring In Robot Gripper	<input type="checkbox"/>	<input type="checkbox"/>

Divers	OK	Not OK	Controle	OK	Not OK
Spur Overtaking Mapping	<input type="checkbox"/>	<input type="checkbox"/>	Repeat Controle Spring	<input type="checkbox"/>	<input type="checkbox"/>
Spring On Nozzle Mapping	<input type="checkbox"/>	<input type="checkbox"/>	Simple Controle Master Sensor	<input type="checkbox"/>	<input type="checkbox"/>
Cassette Not OK Traca	<input type="checkbox"/>	<input type="checkbox"/>	Repeat Controle Master Sensor	<input type="checkbox"/>	<input type="checkbox"/>
Absence Nozzle Mapping	<input type="checkbox"/>	<input type="checkbox"/>	Controle Master Camera	<input type="checkbox"/>	<input type="checkbox"/>
Presence Nozzle Mapping	<input type="checkbox"/>	<input type="checkbox"/>	Repeat Controle Numeric Axis	<input type="checkbox"/>	<input type="checkbox"/>
			Controle Master Cassette	<input type="checkbox"/>	<input type="checkbox"/>

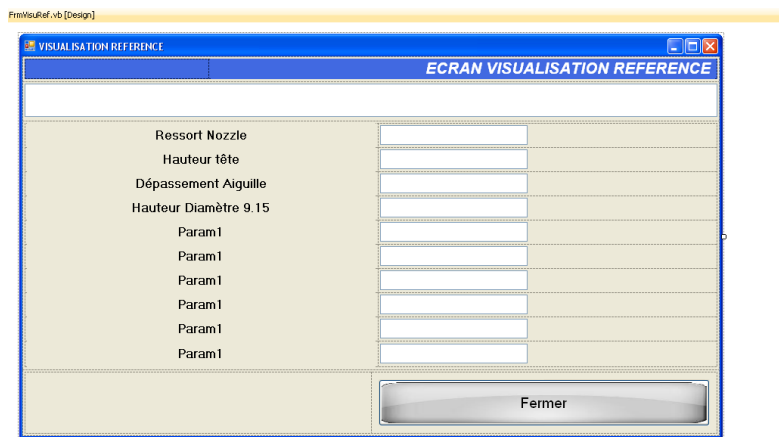
RAZ COMPTEURS FERMER

J'ai conçu cette fenêtre à partir des compteurs fournis par l'automate dans la DB 90. L'automate fournit les valeurs en DINT et on les traitera dans VB en Integer. Ce sont des compteurs généraux (Cassette OK ou pas, Nozzle OK ou pas), de process (Présence Nozzle ou pas, etc...) ou de contrôle (Contrôle master caméra, axes, capteurs ou masters...). Ce sont des comptages de ressorts, de cassette, de nozzles, de mapping, de contrôle qui sont bons ou mauvais. Le code que j'ai créé pour lire ces compteurs et les remettre à 0 est en annexe 3. Vous trouverez le code de cette fenêtre en annexe 1.

a) Référence des produits en cours



La machine traite en permanence 3 références. Grâce à la fenêtre que j'ai conçue, on peut sélectionner chacune des références en cours parmi le catalogue de références contenu dans le fichier .ini. La visu permet de visualiser les caractéristiques de chacune de ces références.



A partir de la référence qu'on a sélectionnée dans la page précédente on publie sur cette page ses caractéristiques mécaniques.

3. Code que j'ai créé

a) Modifications dans les programmes existants pour apporter le protocole Applicom

J'ai fait des lectures et des écritures de bits, par exemple pour le démarrage ou l'arrêt du cycle automatique, l'initialisation, le vidage ou encore les remises à 0 des compteurs. Le code existait en protocole OPC, je l'ai adapté à la communication Applicom en utilisant les fonctions que j'ai créées pour ce faire (Lecture ou Ecriture).

```
'Test si le mode manu est actif
  If (Tab_Applicom_ProcessBit(25) = 1) Then
    Tab_Applicom_ProcessBit(25) = 0
    TablBit(0) = Tab_Applicom_ProcessBit(25)
    EcritureBit(DB_Process_Bit, Operande_Mode_Auto, Bit_Mode_Auto, 1, TablBit)
  End If
```

Ainsi, ce code permet d'écrire le bit comme quoi l'automate doit se mettre en mode manuel en appelant la procédure que j'ai créée pour ce faire. Il en est de même pour la lecture.

Dans les autres fenêtres non détaillées ici figurent les compteurs de bonnes ou mauvaises pièces, l'état des actionneurs, les entrées sorties ou encore la cartographie de la cassette en cours.

L'application échange également des mots. Par exemple :

```
'mise a jour de l'état du controle ressort
  LectureMot(DB_Cartographie, Operande_Etat_Controlle_Ressort, 1, TablMot)
  Applicom_Etat_controlle_Ressort = TablMot(0)
  MiseAJourPosage(Applicom_Etat_controlle_Ressort, PosageControlle1)
```

Cette partie de programme permet de lire le mot état du contrôle ressort et de l'utiliser dans la Fenêtre Cartographie.

b) Ecriture et lecture de chaînes de caractères

J'ai fait les parties de programme en VB suivantes pour écrire et lire des chaînes de caractères de ou vers l'automate. En effet, Applicom ne permet pas d'échanger des formats string. Il faut donc extraire chaque caractère selon la norme ASCII. La longueur de la chaîne est précédée et déterminée par le deuxième caractère échangé.

```
#Region "écriture et lecture ASCII"
```

```
' Fonction d'extraction d'une chaîne à partir d'une DB
```

```

Public Function ByteVersASCII(ByVal DB As Integer, ByVal Operande As Integer) As String
    Dim ChaineDeCaracteres As String
    Dim CarExtrait As Char
    Dim Tabl(1) As Short
    Dim i As Integer
    Dim ByteExtrait As Short
    ChaineDeCaracteres = ""
    Dim NbCar As Integer

    LectureByte(DB, (Operande + 1), 1, Tabl)
    NbCar = Tabl(0)
    For i = 2 To NbCar + 1
        LectureByte(DB, (Operande + i), 1, Tabl)
        ByteExtrait = Tabl(0)
        CarExtrait = Chr(ByteExtrait)
        ChaineDeCaracteres = ChaineDeCaracteres & CarExtrait
        Tabl(0) = 0
    Next

    Return ChaineDeCaracteres
End Function

```

' Fonction d'écriture sur une DB à partir d'une chaîne de caractères

```

Public Sub ASCIIVersByte(ByVal DB As Integer, ByVal Operande As Integer, ByVal Chaine As String)
    Dim CarExtrait As Char
    Dim Tabl(1) As Short
    Dim i As Integer

    Tabl(0) = Len(Chaine)

    EcritureByte(DB, Operande + 1, 1, Tabl)

    For i = 0 To Len(Chaine) - 1

        CarExtrait = Chaine(i)
        Tabl(0) = Asc(CarExtrait)
        EcritureByte(DB, (Operande + i + 2), 1, Tabl)
    Next

End Sub

```

#End Region

TRADUCTION EN ANGLAIS

L'application est destinée au marché Roumain. L'application, via son fichier Affichage.ini, peut contenir jusqu'à 4 traductions. J'ai traduit ce qui ne l'était pas en anglais. Il sera aisé, dans le fichier Affichage.ini, de rajouter le roumain. Voici les principaux termes que j'ai traduits :

FRANCAIS	ANGLAIS
Cartographie	Mapping
Trappe	Trapdoor
Code référence	Reference code
Chargement ressorts nozzles	Spring loaded nozzle
plateau	Tray
Point de prise robot	Point robot outlet
Distributeur	Charging machine
Présence pièce	Presence production
Goulotte	Chute
Refus	Refusal
Etat	State
Emplacement	Location
Référence active	Active reference
Gestion des références	Reference management
En cours	In progress
Contrôle	Control
Produit actif	Active product
Légende	Caption
Cassette	Cassette

vidage	Emptying
Reference en cours	Current reference
Réglage	Setting
Départ cycle	Cycle start
Arrêt fin de cycle	Stop end of cycle
Acquittement	Clearance
Désignation	Designation
Divers	Various variables
Absence	Absence
Présence	Presence
Remise à zéro	Reset
Tableau des compteurs de production	Table of production counters
Posage	Layring
Aiguille	Needle

L'application VB lit le fichier .ini et affiche, selon le code langue, la bonne traduction. Voici un exemple que j'ai réalisé pour la fenêtre de suivi de production du code d'affichage des libellés, boutons ou encore titres :

```
''' <summary>
    ''' Multilangue fonction
    ''' Fonction de chargement des libellés en fonction de la langue en cours
    ''' </summary>
    ''' <remarks></remarks>
Private Sub MAJ_Affichage()
    'attention le fichier .ini est décalé de 1 par rapport au tableau (indice 0
dans le tableau)
    lblTitre.Text =
TableauAffichage.GroupAffichage(6).NomGroupeL(NumLangueEnCours)

    lblLegendeTableau.Text =
TableauAffichage.GroupAffichage(17).Données(0).LibelleL(NumLangueEnCours)
```

```

        lblCompteurTotal.Text =
TableauAffichage.GroupAffichage(17).Données(1).LibelleL(NumLangueEnCours)
        BF10.Text =
TableauAffichage.GroupAffichage(17).Données(5).LibelleL(NumLangueEnCours)
        BPClose.Text =
TableauAffichage.GroupAffichage(17).Données(6).LibelleL(NumLangueEnCours)

        ' Affichage mis à jour par FG
        lblGeneral.Text =
TableauAffichage.GroupAffichage(17).Données(7).LibelleL(NumLangueEnCours)
        lblCassette.Text =
TableauAffichage.GroupAffichage(17).Données(8).LibelleL(NumLangueEnCours)
        lblNozzle.Text =
TableauAffichage.GroupAffichage(17).Données(9).LibelleL(NumLangueEnCours)
        lblDivers.Text =
TableauAffichage.GroupAffichage(17).Données(10).LibelleL(NumLangueEnCours)
        lblSpurOvertakingMapping.Text =
TableauAffichage.GroupAffichage(17).Données(11).LibelleL(NumLangueEnCours)
        lblSpringOnNozzleMapping.Text =
TableauAffichage.GroupAffichage(17).Données(12).LibelleL(NumLangueEnCours)
        lblCassetteNotOKTraca.Text =
TableauAffichage.GroupAffichage(17).Données(13).LibelleL(NumLangueEnCours)
        lblProcess.Text =
TableauAffichage.GroupAffichage(17).Données(14).LibelleL(NumLangueEnCours)
        lblMappingCassette.Text =
TableauAffichage.GroupAffichage(17).Données(15).LibelleL(NumLangueEnCours)
        lblControleSpring.Text =
TableauAffichage.GroupAffichage(17).Données(16).LibelleL(NumLangueEnCours)
        lblSpringInRobotGripper.Text =
TableauAffichage.GroupAffichage(17).Données(17).LibelleL(NumLangueEnCours)
        lblControle.Text =
TableauAffichage.GroupAffichage(17).Données(18).LibelleL(NumLangueEnCours)
        lblRepeatControleSpring.Text =
TableauAffichage.GroupAffichage(17).Données(19).LibelleL(NumLangueEnCours)
        lblSimpleControleMasterSensor.Text =
TableauAffichage.GroupAffichage(17).Données(20).LibelleL(NumLangueEnCours)
        lblRepeatControleMasterSensor.Text =
TableauAffichage.GroupAffichage(17).Données(21).LibelleL(NumLangueEnCours)
        lblControleMasterCamera.Text =
TableauAffichage.GroupAffichage(17).Données(22).LibelleL(NumLangueEnCours)
        lblRepeatControleNumericAxis.Text =
TableauAffichage.GroupAffichage(17).Données(23).LibelleL(NumLangueEnCours)
        lblControleMasterCassette.Text =
TableauAffichage.GroupAffichage(17).Données(24).LibelleL(NumLangueEnCours)
        lblAbsenceNozzleMapping.Text =
TableauAffichage.GroupAffichage(17).Données(25).LibelleL(NumLangueEnCours)
        lblPresenceNozzleMapping.Text =
TableauAffichage.GroupAffichage(17).Données(26).LibelleL(NumLangueEnCours)
        Label13.Text =
TableauAffichage.GroupAffichage(17).Données(27).LibelleL(NumLangueEnCours)

```

End Sub

Vous trouverez la partie du fichier Affichage .ini que j'ai réalisée en Annexe 3.

CONCLUSION

Ce stage a été intéressant car il m'a permis de me familiariser avec l'informatique industrielle et d'approfondir mes connaissances en Visual Basic. Il s'agissait de refaire une supervision IHM en Visual Basic à partir d'une application existante. J'ai pu apprendre ce qui était nécessaire dans une IHM de cette envergure. Il s'agissait ensuite de migrer une liaison avec un automate de l'OPC vers l'Ethernet industriel. Il a donc fallu d'abord tester cette liaison, s'imprégner du code pour le corriger ensuite. J'ai créé des procédures de lecture VB pour utiliser Applicom. Ainsi, j'ai changé tous les appels de la fonction OPC par des appels Applicom. J'ai pu voir le produit mécanique lors d'une visite chez le client AMS (Anjou Machines Spéciales). Ce système de dépose de ressorts sur des nozzles pour un sous-traitant de l'industrie automobile est assez complexe : il comporte un robot 4 axes, des capteurs et des actionneurs ainsi qu'un système de vision par caméra. La société Fast Intégration doit livrer l'ensemble du soft au mois d'avril 2015 en Roumanie.

SOURCES

Document SW Library DLL

Document Industrial Ethernet sur PCI2000ETH (TCP/IP, ISO couche 4) et SW1000ETH (TCP/IP)

ANNEXES

1. Annexe 1 : Déclarations pour les DB

```
Module VarApplicom
```

```
'Nom de la liaison S7  
'KEP  
'Public Const S7 = "DELPHI.Liaison_S1."  
'S7  
'Public Const S7 = "S7:[Liaison_S7_1]"
```

```
Public Adr As String  
'Public Jeton As Integer  
'Public Serveur_Demarre As Boolean
```

```
#Region "Déclaration des numéros de DB"
```

```
'Déclaration des groupes DB d'échanges avec l'automate  
Public DB_Fiche_StatusPosage As Integer = 1 'DB1 (mots des posages)  
Public DB_Process_Bit As Integer = 90 'DB90(bits page principale)  
Public DB_Process_Int As Integer = 90 'DB90(Mots page principale)  
Public DB_Compteurs As Integer = 90 ' DB 90 pour les compteurs  
Public DB_InfoOp As Integer = 90 'DB90 pour info op  
Public DB_Process_TpsCycle As Integer = 90 'DB90 Tps de cycle  
Public DB_Process_Bit_Cartographie As Integer = 90 'DB90(bits page cartographie)  
Public DB_Process As Integer = 90 'DB90 autres variables  
Public DB_Mesure As Integer = 91 'DB91 mesures  
Public DB_Parametrage As Integer = 93 'DB93 page paramétrage  
Public DB_I_O As Integer = 94 'DB94 (page entrées sorties)  
Public DB_Peripherique As Integer = 95 'DB95 (page peripheriques)  
Public DB_Actionneur As Integer = 96 'DB96(Page actionneur)  
Public DB_Cartogrphie_Posage As Integer = 97 'DB97 (mots K7 refus pour la page  
cartographie  
Public DB_Cartographie As Integer = 97 'DB97 autres variables  
Public DB_Suivi_Prod As Integer = 98 'DB98 suivi de production  
Public DB_Default As Integer = 99 'DB99  
Public DB_Fiche As Integer = 11 'DB1 à DB50  
Public DB_Grafcet As Integer = 88 'DB pour la gestion des G77  
Public DB_Btr As Integer = 92 'DB 92 pour la liaison avec  
la btr  
Public DB_Laser As Integer = 81  
'Public DB_Test_Variables As Integer
```

```
#End Region
```

```
#Region "Variables DB1"
```

```
'***** Gestion des mots Posages *****  
'tableau permettant de stocker le Handle des Items Applicom (utilisé pour la  
lecture Directe dans l'API)
```

'la grandeur du tableau dépend du nombre d'items à lire ici 6 items dans le DB1
donc 6-1 car TabHandles_DB1 commence à 0

```
Public NumItemDB1 As Integer = 10
Public TabHandles_DB1(NumItemDB1 - 1) As Long
'tableau d'échanges IHM->Automate
'Public Tab_Applicom_Fiche(NumItemDB1 - 1) As VariantType

'legende pour le tableau Tab_Applicom_Fiche
'0 -> Applicom_Fiche_Status_Pompe => Status de la pompe en cours
'1 -> Applicom_Fiche_Num_Palette => N° de palette
'2 -> Applicom_Fiche_Ref_Pompe => Référence de la pompe en cours
'3 -> Applicom_Fiche_Num_Serie_Pompe => N° de série de la pompe en cours
'4 -> Applicom_Fiche_Type_Pompe => Codé défaut
'5 -> Applicom_Fiche_Valeur_Fuite => Valeur de la fuite
'6 -> Applicom_Fiche_DateDebut_OP => date de début de l'OP
'7 -> Applicom_Fiche_HeureDebut_OP => heure de début de l'OP
'8 -> Applicom_Fiche_Code_Defaut => Code défaut pièce
'9 -> Applicom_Fiche_Code_Station => Code de la station
```

'Variable donnée mesure ou etalonnage DB1

```
Public Applicom_Fiche_Date As Date
```

#End Region

#Region "Variables Fiches posages DB11,12,13,14"

```
Public Applicom_Status_Posage10 As Short
Public Applicom_Status_Posage11 As Short
Public Applicom_Status_Posage12 As Short
Public Applicom_Status_Posage13 As Short
Public Applicom_Status_Posage14 As Short
```

#End Region

#Region "Variables DBXX"

```
'Public Applicom_Capa_Fonction As ApplicomItem
Public AdrCapaApplicom As String
```

#End Region

#Region "Variables process DB90"

```
'***** Gestion des bits *****

'nombre de bits dans le groupe DB_Process_Bit
Public Nombre_Bits_Process As Integer = 72
'tableau d'échanges IHM->Automate
Public Tab_Applicom_ProcessBit(Nombre_Bits_Process) As Short
'Définition de la plage
Public Applicom_Bits_Process(Nombre_Bits_Process) As Short

'***** Gestion des Mots *****

'nombre de mots dans le groupe DB_Process_Bit
```

```

Public Nombre_Mots_Process As Integer = 5
'tableau d'échanges IHM<->Automate
Public Tab_Applicom_ProcessInt(Nombre_Mots_Process) As Short
'Définition de la plage
Public Applicom_Mots_Process(Nombre_Mots_Process) As Short

'***** Gestion Infos Operateur*****
'tableau d'échange IHM<->Automate
Public Tab_Applicom_InfoOp() As Short
'adresse de la variable Applicom (contenue dans le fichier .ini)
Public AdrInfoOperateurApplicom As String
Public DBInfoOperateurApplicom As Integer
Public OperandeInfoOperateurApplicom As Integer
Public NbInfoOperateurApplicom As Short

'définition de l'item Applicom
Public Applicom_Info_Operateur As Short

'***** Gestion des bits cartographie *****
'nombre de bits dans le groupe DB_Process_Bit
Public Nombre_Bits_Process_Cartographie As Integer = 16
'tableau d'échanges IHM<->Automate
Public Tab_Applicom_ProcessBit_Cartographie(Nombre_Bits_Process_Cartographie) As
Short
'Définition de la plage
Public Applicom_Bits_Process_Cartographie(Nombre_Bits_Process_Cartographie) As
Short

'*** Gestion des autres variables du DB90 *****

Public Applicom_Num_Ref_Res As Short
Public Applicom_MDP_Param As String
Public Applicom_Ref_String As String

Public Applicom_Compteur_Totalisateur As Integer
Public Applicom_Compteur_Totalisateur_Raz As Integer
Public Applicom_Decompteur_Etalonnage As Short

Public Applicom_Demande_Lecture_K71 As Short
Public Applicom_Demande_Lecture_K72 As Short

'paramètres de référence pur les K7
Public Applicom_Ref_Ressort_K71 As Short
Public Applicom_Ref_Hauteur_K71 As Short
Public Applicom_Ref_Depassement_Aiguille_K71 As Short
Public Applicom_Ref_Diam_K71 As Short

Public Applicom_Ref_Ressort_K72 As Short
Public Applicom_Ref_Hauteur_K72 As Short
Public Applicom_Ref_Depassement_Aiguille_K72 As Short
Public Applicom_Ref_Diam_K72 As Short

Public Applicom_Ref_Ressort_K7Capa As Short
Public Applicom_Ref_Hauteur_K7Capa As Short
Public Applicom_Ref_Depassement_Aiguille_K7Capa As Short
Public Applicom_Ref_Diam_K7Capa As Short

```

```

Public Applicom_Etat_K71 As Integer
Public Applicom_Etat_K72 As Integer

Public Applicom_Position_Effort_Plateau As Short

'Gestion de la capabilité
Public Applicom_Fonction_Capabilite As Integer
Public Applicom_Ref_Capabilite As Integer

'gestion des pages
'Public Applicom_Num_Page_En_Cours As ApplicomItem

'*** Gestion des bits de mesure (top enregistrement) *****

'nombre de bits dans le groupe DB_Mesure_Bit
Public Nombre_Bits_Mesure_Bit As Integer = 6
'Définition del a plage de bits
Public Tab_Applicom_Mesure_Bit(Nombre_Bits_Mesure_Bit) As Short
Public Applicom_Mesure_Bit(Nombre_Bits_Mesure_Bit) As Short

'***** Gestion des temps de cycle *****

'nombre de mots dans le groupe DB_Process_TpsCycle
Public Nombre_Mots_TpsCycle As Integer = 4
'tableau d'échanges IHM<->Automate

Public Tab_Applicom_Process_TpsCycle(Nombre_Mots_TpsCycle) As Short
'Définition de la plage

Public Applicom_Tps_Cycle(Nombre_Mots_TpsCycle) As Short

'Public Applicom_RAZ_CPT_Ref As ApplicomItem

#End Region

'*****=====*****
'A ranger suivant le DB de destination
Public AdrG7Applicom As String
Public DBG7Applicom As Integer
Public OperandeG7Applicom As Integer
Public Applicom_Grafcet(1) As Short
Public NbG7Applicom As Short

'*****=====*****

' Variables de compteurs de la DB90
Public Applicom_CassetteOK As Integer
Public Applicom_CassetteNOK As Integer
Public Applicom_NozzleOK As Integer
Public Applicom_NozzleNOK As Integer
Public Applicom_CassetteNOKTraca As Integer
Public Applicom_MappingCassetteOK As Integer
Public Applicom_MappingCassetteNOK As Integer
Public Applicom_AbsenceNozzleMapping As Integer
Public Applicom_PresenceNozzleMapping As Integer
Public Applicom_SpurOvertakeMapping As Integer
Public Applicom_SpringOnNozzleMapping As Integer
Public Applicom_CtrlSpringOK As Integer
Public Applicom_CtrlSpringNOK As Integer

```

```

Public Applicom_SpringInRobGripperOK As Integer
Public Applicom_SpringInRobGripperNOK As Integer
Public Applicom_RepeatCtrlSpringOK As Integer
Public Applicom_RepeatCtrlSpringNOK As Integer
Public Applicom_SimplCtrlMastSensorOK As Integer
Public Applicom_SimplCtrlMastSensorNOK As Integer
Public Applicom_RepeatCtrlMastSensorOK As Integer
Public Applicom_RepeatCtrlMastSensorNOK As Integer
Public Applicom_CtrlMasterCamOK As Integer
Public Applicom_CtrlMasterCamNOK As Integer
Public Applicom_RepeatCtrlNumericAxisOK As Integer
Public Applicom_RepeatCtrlNumericAxisNOK As Integer
Public Applicom_CtrlMasterCassetteOK As Integer
Public Applicom_CtrlMasterCassetteNOK As Integer

```

```
#Region "Variables DB91"
```

```

'Variables parametres du DB91

'variables pour la fiche pièce
Public Applicom_Fiche As Short
Public Applicom_Fiche_DataMatrix As String

Public Applicom_Fiche_Reference As String
Public Applicom_Fiche_Op_Data_Faite As Short

```

```
#End Region
```

```
#Region "Variables DB92"
```

```

Public Applicom_CodeRetourBaseEntree As Short
Public Applicom_CodeRetourBaseSortie As Short

Public Applicom_Dem_BTR_Voie1 As Short
Public Applicom_Dem_BTR_Voie2 As Short

Public Applicom_Data_K71 As String
Public Applicom_Data_K72 As String

Public Applicom_Nb_Bonnes_K71 As Short
Public Applicom_Nb_Bonnes_K72 As Short
Public Applicom_Nb_Mauvaises_K71 As Short
Public Applicom_Nb_Mauvaises_K72 As Short

```

```
#End Region
```

```
#Region "Variables DB93"
```

```

'Variables parametres du DB93
Public AdrParametreApplicom As String
Public DBParametreApplicom As Integer
Public OperandeParametreApplicom As Integer
Public NbParametreApplicom As Short
Public Applicom_Parametres() As Short

```

```
#End Region
```

```
#Region "Variables DB94"
```

```

'Variables entrees/sortie du DB94
Public AdrESApplicom As String

```



```

Public DBESApplicom As Integer
Public OperandeESApplicom As Integer
Public NbESApplicom As Short
Public Applicom_Entre_Sortie() As Short
Public Tab_Applicom_IO() As Long

#End Region

#Region "Variables DB95"
'Variables Commande peripherique du DB95
Public AdrPeripheriqueApplicom As String
Public DBPeripheriqueApplicom As Integer
Public OperandePeripheriqueApplicom As Integer
Public NbPeripheriqueApplicom As Short
Public Applicom_Peripherique() As Short

'Variables spécifiques pour le mode périphérique
Public Applicom_NumK7_Perri As Short
Public Applicom_EmplacementK7_Perri As Short
Public Applicom_Traj_Robot_Perri As Short
Public Applicom_RefRessort_Perri As Short
Public Applicom_RefNozzle_Perri As Short

#End Region

#Region "Variables DB96"
'Variables Commande Actionneur du DB96
Public AdrActionneurApplicom As String
Public DBActionneurApplicom As Integer
Public OperandeActionneurApplicom As Integer
Public NbActionneurApplicom As Short
Public Applicom_Actionneur() As Short
Public Tab_Applicom_Action() As Long
#End Region

#Region "Variables DB97"

'***** Gestion de la page cartographie *****
'images des K7 de production
Public Applicom_Image_K71() As Short
Public Applicom_Image_K72() As Short
'etat pour les K7 de production
Public Applicom_Emplacement_K71 As Short
Public Applicom_Emplacement_K72 As Short
'refERENCE en cours pour les K7
Public Applicom_Ref_K7_1 As Short
Public Applicom_Ref_K7_2 As Short
'etat du distributeur ressort
Public Applicom_Etat_Distributeur(3) As Short
'position distributeur ressort
Public Applicom_Position_Distri As Short
'etat du plateau
Public Applicom_Etat_Plateau(24) As Short
'position du plateau
Public Applicom_Position_Plateau As Short
'etat du controle ressort
Public Applicom_Etat_controle_Ressort As Integer
'valeur de l'effort pour le controle
Public Applicom_Valeur_Effort As Short

```

```

Public Applicom_Valeur_Effort_Etalon As Short
'valeur position controle ressort
Public Applicom_Position_Controle As Short
'état de la pince robot
Public Applicom_Pince_Robot As Integer
Public Applicom_Prise_Robot As Integer
'état posage goulotte
Public Applicom_Posage_Refus As Short
'état du controle ressort
Public Applicom_Posage_Controle_Ressort As Short
'état des tiroirs
Public Applicom_Etat_Tiroirs(3) As Integer
'compteur de pièces
Public Applicom_Compteur_Nozzles As Short
Public Applicom_Compteur_Ressorts As Short

#End Region

#Region "Variables DB98"
'Variables Applicom pour le suivi de production
Public Applicom_Suivi_Prod_Ref(40) As Short
Public Applicom_Suivi_Prod_Totaliseurs(5) As Integer

#End Region

#Region "Variables DB99"
'Variables Defaut du DB99

'tableau d'échanges IHM->Automate
Public Tab_Applicom_Defaults() As Long
'Définition de la variable Applicom contenant l'ensemble des Defaults
Public Applicom_Defaut() As Short
'Bool permettant d'indiquer un datachange
Public Flag_Mots_Defaut As Short
Public DBDefautApplicom As Integer
Public OperandeDefautApplicom As Integer
Public NbDefautApplicom As Short

Public AdrDefautApplicom As String

#End Region

#Region "compteurs DB90"
'déclaration pour les opérandes des compteurs de la DB90
Public Operande_CassetteOK As Integer = 0
Public Operande_CassetteNOK As Integer = 4
Public Operande_NozzleOK As Integer = 8
Public Operande_NozzleNOK As Integer = 12
Public Operande_CassetteNOKTraca As Integer = 40
Public Operande_MappingCassetteOK As Integer = 80
Public Operande_MappingCassetteNOK As Integer = 84
Public Operande_AbsenceNozzleMapping As Integer = 88
Public Operande_PresenceNozzleMapping As Integer = 92
Public Operande_SpurOvertakeMapping As Integer = 96
Public Operande_SpringOnNozzleMapping As Integer = 100
Public Operande_CtrlSpringOK As Integer = 104
Public Operande_CtrlSpringNOK As Integer = 108
Public Operande_SpringInRobGripperOK As Integer = 112
Public Operande_SpringInRobGripperNOK As Integer = 116
Public Operande_RepeatCtrlSpringOK As Integer = 160

```

```

Public Operande_RepeatCtrlSpringNOK As Integer = 164
Public Operande_SimplCtrlMastSensorOK As Integer = 168
Public Operande_SimplCtrlMastSensorNOK As Integer = 172
Public Operande_RepeatCtrlMastSensorOK As Integer = 176
Public Operande_RepeatCtrlMastSensorNOK As Integer = 180
Public Operande_CtrlMasterCamOK As Integer = 184
Public Operande_CtrlMasterCamNOK As Integer = 188
Public Operande_RepeatCtrlNumericAxisOK As Integer = 192
Public Operande_RepeatCtrlNumericAxisNOK As Integer = 196
Public Operande_CtrlMasterCassetteOK As Integer = 200
Public Operande_CtrlMasterCassetteNOK As Integer = 204

```

#End Region

#Region "Opérandes et bits Applicom"

```

'*****
'
'                               INITIALISATION DES ITEMS
'
'*****

```

```

'fiche du posage 11
Public Operande_Status_Posage11 As Integer = 0
'fiche du posage 12
Public Operande_Status_Posage12 As Integer = 2
'fiche du posage 13
Public Operande_Status_Posage13 As Integer = 4
'fiche du posage 14
Public Operande_Status_Posage14 As Integer = 6
'fiche du posage 10
Public Operande_Status_Posage10 As Integer = 8

```

'Items à ajouter : Page principale boutons,compteurs,mode de marche...

```

'*****
'DB 90 Bits
'Gestion des bits du DB90

```

```

'Legende plage bits process (correspondance avec tab_opc_bitprocess)
'0 DB90,0.0 -> OPC_TEST_LAMPES
Public Operande_Test_Lampe As Integer = 0
Public Bit_Test_Lampe As Integer = 0

'1 DB90,0.1 -> OPC_Eclairage_Cellule
'2 DB90,0.2 ->
'3 DB90,0.3 ->
'4 DB90,0.4 -> Vidage type 1
Public Operande_Vidage_Type_1 As Integer = 0
Public Bit_Vidage_Type_1 As Integer = 4
'5 DB90,0.5 -> Vidage type 2
Public Operande_Vidage_Type_2 As Integer = 0
Public Bit_Vidage_Type_2 As Integer = 5
'6 DB90,0.6 -> Vidage type 3
'7 DB90,0.7 -> Machine_Vide
Public Operande_Machine_Vide As Integer = 0

```

```

Public Bit_Machine_Vide As Integer = 7
'8 DB90,1.0 ->
'9 DB90,1.1 ->
'10 DB90,1.2 -> RAZ compteurs page svui prod
Public Operande_RAZ_Cpt_Suivi_Prod As Integer = 1
Public Bit_RAZ_Cpt_Suivi_Prod As Integer = 2
'11 DB90,1.3 -> RAZ Compteurs page principale
Public Operande_RAZ_Cpt_Page_Principale As Integer = 1
Public Bit_RAZ_Cpt_Page_Principale As Integer = 3
'12 DB90,1.4 -> RAfraichissement E/S
Public Operande_Raffraichissement_ES As Integer = 1
Public Bit_Raffraichissement_ES As Integer = 4
'13 DB90,1.5 -> RAZ Cellule (bouton cartographie)
Public Operande_RAZ_Cellule As Integer = 1
Public Bit_RAZ_Cellule As Integer = 5
'14 DB90,1.6 ->
'15 DB90,1.7 ->
'16 DB90,2.0 -> Demande étalonnage manuel
Public Operande_Dem_Etalonnage_Manuel As Integer = 2
Public Bit_Dem_Etalonnage_Manuel As Integer = 0
'17 DB90,2.1 -> Demande étalonnage automatique
'18 DB90,2.2 ->
'19 DB90,2.3 ->
'20 DB90,2.4 -> Mode pas à pas
'21 DB90,2.5 -> P+
'22 DB90,2.6 ->
'23 DB90,2.7 ->
'24 DB90,3.0 -> Demande Init
Public Operande_Demande_Init As Integer = 3
Public Bit_Demande_Init As Integer = 0
'25 DB90,3.1 -> Mode auto (1 = auto 0 = réglages)
Public Operande_Mode_Auto As Integer = 3
Public Bit_Mode_Auto As Integer = 1
'26 DB90,3.2 -> Demande départ cycle
Public Operande_DCY As Integer = 3
Public Bit_DCY As Integer = 2
'27 DB90,3.3 -> demande arrêt cycle
Public Operande_ACY As Integer = 3
Public Bit_ACY As Integer = 3
'28 DB90,3.4 -> Acquiescement défauts
Public Operande_Acq As Integer = 3
Public Bit_Acq As Integer = 4
'29 DB90,3.5 ->
'30 DB90,3.6 ->
'31 DB90,3.7 ->
'32 DB90,4.0 ->
'33 DB90,4.1 ->
'34 DB90,4.2 ->
'35 DB90,4.3 ->
'36 DB90,4.4 ->
'37 DB90,4.5 ->
'38 DB90,4.6 ->
'39 DB90,4.7 ->
'40 DB90,5.0 -> Resultat correlation OK
'41 DB90,5.1 -> Résultat correlation NOK
'42 DB90,5.2 ->
'43 DB90,5.3 ->
'44 DB90,5.4 ->
'45 DB90,5.5 ->
'46 DB90,5.6 ->
'47 DB90,5.7 ->
'48 DB90,6.0 ->

```

```

'49 DB90,6.1 ->
'50 DB90,6.2 ->
'51 DB90,6.3 ->
'52 DB90,6.4 ->
'53 DB90,6.5 ->
'54 DB90,6.6 ->
'55 DB90,6.7 ->
'56 DB90,7.0 -> Demande enregistrement données
'57 DB90,7.1 -> Demande enregistrement SPC
'58 DB90,7.2 ->
'59 DB90,7.3 -> Demande affichage carto
Public Operande_Affichage_Carto As Integer = 7
Public Bit_Affichage_Carto As Integer = 3
'60 DB90,7.4 -> RAZ compteur Ref
'61 DB90,7.5 -> Demande enregistrement etalon
'62 DB90,7.6 ->
'63 DB90,7.7 ->
'64 DB90,8.0 -> Mouvement Actionneur AV
Public Operande_Mouvement_Actionneur_AV As Integer = 8
Public Bit_Mouvement_Actionneur_AV As Integer = 0
'65 DB90,8.1 -> Mouvement Actionneur AR
Public Operande_Mouvement_Actionneur_AR As Integer = 8
Public Bit_Mouvement_Actionneur_AR As Integer = 1
'66 DB90,8.2 -> Peripherique DCY
'67 DB90,8.3 -> Peripherique PaP
'68 DB90,8.4 -> R&R DCY
'69 DB90,8.5 -> R&R Arret cycle
'70 DB90,8.6 -> Bit enregistrement donnee Cycle Capa
Public Operande_Enr_Cycle_Capa As Integer = 8
Public Bit_Enr_Cycle_Capa As Integer = 6
'71 DB90,8.7 ->

```

```

'*****
'Db90 process Words
'Gestions des mots du DB90
'*****

```

```

'Legende plage mots process
'0 DB90,W12 -> Compteur pièces bonnes
Public Operande_Cpt_Pieces_Bonnes As Integer = 12
'1 DB90,W14 -> Compteur pièces mauvaises
Public Operande_Cpt_Pieces_Mauvaises As Integer = 14
'2 DB90,W16 -> Mode de marche de la cellule
Public Operande_Mode_Marche As Integer = 16
'3 DB90,W18 -> Vitesse du robot
Public Operande_Vitesse_Robot As Integer = 18
'4 DB90,W20 ->

```

```

'*****
'DB90 Operator informations
'Gestions des Infos Operateur DB90
'*****

```

```

'*****
'DB90 Mapping informations
'Gestions des Infos Cartographie DB90
'*****

```

```

'Legende plage bits process cartographie (correspondance avec
Tab_OPC_ProcessBit_Cartographie)
'0 DB90,9.0 -> Mode dégradé 1
Public Operande_Mode_Degrade_1 As Integer = 9
Public Bit_Mode_Degrade_1 As Integer = 0
'1 DB90,9.1 -> Mode dégradé 2
'2 DB90,9.2 -> Mode dégradé 3
'3 DB90,9.3 -> Mode dégradé 4
'4 DB90,9.4 -> Mode dégradé 5
'5 DB90,9.5 -> Mode dégradé 6
'6 DB90,9.6 -> Mode dégradé 7
'7 DB90,9.7 -> Mode dégradé 8
Public Operande_Mode_Degrade_8 As Integer = 9
Public Bit_Mode_Degrade_8 As Integer = 7
'8 DB90,10.0 -> Etat Activation Posage 1
'9 DB90,10.1 -> Etat Activation Posage 2
'10 DB90,10.2 -> Etat Activation Posage 3
'11 DB90,10.3 -> Etat Activation Posage 4
'12 DB90,10.4 ->

*****
'Cycle time OPC items
'Gestions des temps de cycle DB90
*****

'Legende plage mots process
'0 DB90,W22 -> Temps de cycle machine
Public Operande_Tps_CY_Machine As Integer = 22

'1 DB90,W24 -> Temps de cycle Poste 1
Public Operande_Tps_CY_P1 As Integer = 24

'2 DB90,W26 -> Temps de cycle Poste 2
Public Operande_Tps_CY_P2 As Integer = 26

'3 DB90,W28 -> Temps de cycle Poste 3
Public Operande_Tps_CY_P3 As Integer = 28

'SPC counter
'Compteur SPC
Public Operande_Decompteur_Etalonnage As Integer = 30

'Type posage à la position effort
'Compteur SPC
Public Operande_Position_Effort_Plateau As Integer = 56

'General counter (DWord)
'compteur totalisateur sur un double mot
Public Operande_Cpt_Tot As Integer = 42

'General counter (DWord)
'compteur totalisateur sur un double mot pouvant etre raz
Public Operande_Cpt_Tot_RAZ As Integer = 38

'Password for plc downloading settings
'gestion du mot de passe pour le transfert vers API
'   Adr = S7 & "DB90,STRING126.12"
'   Jeton = 18
'   OPC_MDP_Param = DB_Process.AddItem(Adr, Jeton)

```

```

'Actual reference
'r f rence en cours
'      Adr = S7 & "DB90,STRING76.12"
'      Jeton = 18
'      OPC_Ref_String = DB_Process.AddItem(Adr, Jeton)

'Reference number
'Num ro de la r f rence en cours
Public Operande_Num_Ref_Res As Integer = 96

'Bit demande de lecture datamatrix K71
Public Operande_Demande_Lecture_K71 As Integer = 98
Public Bit_Demande_Lecture_K71 As Integer = 0

'Bit demande de lecture datamatrix K72
Public Operande_Demande_Lecture_K72 As Integer = 100
Public Bit_Demande_Lecture_K72 As Integer = 0

'code ressort K71
Public Operande_Ref_Ressort_K71 As Integer = 102

'code hauteur K71
Public Operande_Ref_Hauteur_K71 As Integer = 104

'code d passement aiguille
Public Operande_Ref_Depassement_Aiguille_K71 As Integer = 106
'code diametre 9.15
Public Operande_Ref_Diam_K71 As Integer = 108
'code ressort K72
Public Operande_Ref_Ressort_K72 As Integer = 114

'code hauteur K72
Public Operande_Ref_Hauteur_K72 As Integer = 116
'code d passement aiguille
Public Operande_Ref_Depassement_Aiguille_K72 As Integer = 118
'code diametre 9.15
Public Operande_Ref_Diam_K72 As Integer = 120

' *****
' DB97
' *****
'Variables partag e entre process et Cartographie (DB97)
'etat en cours de la K71
'      Adr = S7 & "DB97,DBB0"
'      Jeton = 18
Public Operande_Etat_K71 As Integer = 0
'etat en cours de la K72
'      Adr = S7 & "DB97,DBB60"
'      Jeton = 18
'      OPC_Etat_K72 = DB_Process.AddItem(Adr, Jeton)
Public Operande_Etat_K72 As Integer = 60

'Ref K7 1
Public Operande_Ref_K7_1 As Integer = 56
'Ref K7 2
Public Operande_Ref_K7_2 As Integer = 116

' *****
' DB90
' *****
'Ref K7 2

```

```

'      Adr = S7 & "DB90,DBB6"
'      Jeton = 505
'      OPC_Num_Page_En_Cours = DB_Process.AddItem(Adr, Jeton)

' Type fonction capabilité
'      Adr = S7 & "DB90,DBB58"
'      Jeton = 505
'      OPC_Fonction_Capabilite = DB_Process.AddItem(Adr, Jeton)
Public Operande_Fonction_Capabilite As Integer = 58
' Type de pièce utilisé pour la capabilité
'      Adr = S7 & "DB90,DBB59"
'      Jeton = 505
'      OPC_Ref_Capabilite = DB_Process.AddItem(Adr, Jeton)
Public Operande_Ref_Capabilite As Integer = 60

' code ressort K7 CAPA
Public Operande_Ref_Ressort_K7Capa As Integer = 154

' code hauteur K7 CAPA
Public Operande_Ref_Hauteur_K7Capa As Integer = 156

' code dépassement aiguille
Public Operande_Ref_Depassement_Aiguille_K7Capa As Integer = 158
' code diametre 9.15
Public Operande_Ref_Diam_K7Capa As Integer = 160

' Fiche des données à sauvegarder
' KEP
'
' *****
' DB91
' *****

'      Adr = S7 & "DB91,W0[6]"
' S7
' Adr = S7 & "DB91,INT0,6"
'      Jeton = 1
'      OPC_Fiche = DB_Mesure.AddItem(Adr, Jeton)

' Datamatrix to save
' Fiche -> DataMatrix
'      Adr = S7 & "DB91,STRING46.33"
'      Jeton = 8
'      OPC_Fiche_DataMatrix = DB_Mesure.AddItem(Adr, Jeton)

' Datamatrix result
' Fiche -> Résultat de l'opération détection de data
Public Operande_Fiche_Op_Data_FAite As Integer = 30
Public Bit_Fiche_Op_Data_Faite As Integer = 5
' Reference
' Fiche Reference 10 caracteres
' Adr = S7 & "DB91,STRING32.12"
' Jeton = 7
' OPC_Fiche_Reference = DB_Mesure.AddItem(Adr, Jeton)

' Adr = AdrParametreOPC
' Jeton = 1
' OPC_Parametres = DB_Parametrage.AddItem(Adr, Jeton)

```



```

' IO Items creation
' Création des items IO

'     Adr = AdrESOPC
'     Jeton = 1
'     OPC_Entre_Sortie = DB_I_O.AddItem(Adr, Jeton)

' Actuators OPC items
' Création des Items des actionneurs

'     Adr = AdrActionneurOPC
'     Jeton = 1
'     OPC_Actionneur = DB_Actionneur.AddItem(Adr, Jeton)

'Peripheral cycle OPC items
'Création des items périphériques

'     Adr = AdrPeripheriqueOPC
'     Jeton = 600
'     OPC_Peripherique = DB_Peripherique.AddItem(Adr, Jeton)

'*****
'DB95
'*****
,
Public Operande_NumK7_Peri As Integer = 4

Public Operande_EmplacementK7_Peri As Integer = 6

Public Operande_Traj_Robot_Peri As Integer = 8

Public Operande_RefRessort_Peri As Integer = 10

Public Operande_RefNozzle_Peri As Integer = 12

' Mapping OPC items
' création des items de cartographie

'*****
'Gestions des Donnees Cartographie DB97
'*****

'GESTION DES K7 DE PRODUCTION

'     Adr = S7 & "DB97,DBB0[51]"
'     Jeton = 504
'     OPC_Image_K71 = DB_Cartographie.AddItem(Adr, Jeton)
'Public Operande_Image_K71 As Integer = 154
'image des emplacement dans la K7 1
'Emplacement 1 : en bas à gauche
'emplacement 10 en bas à droite
'emplacement 50 en haut à droite
Public Operande_Image_K71 As Integer = 0

'     Adr = S7 & "DB97,DBB60[51]"
'     Jeton = 505
'     OPC_Image_K72 = DB_Cartographie.AddItem(Adr, Jeton)
'image des emplacement dans la K7 2

```

```

'Emplacement 1 : en bas à gauche
'emplacement 10 en bas à droite
'emplacement 50 en haut à droite
Public Operande_Image_K72 As Integer = 120

Public Operande_Emplacement_K71 As Integer = 200
'etat de la K71 en cours nb emplacement

Public Operande_Emplacement_K72 As Integer = 202
'etat de la K72 en cours nb emplacement

'GESTION DU DISTRIBUTEUR RESSORTS
'Adr = S7 & "DB97,DBB121[3]"
'Jeton = 505
'OPC_Etat_Distributeur = DB_Cartographie.AddItem(Adr, Jeton)
'position du distributeur
Public Operande_Etat_Distributeur As Integer = 204

Public Operande_Position_Distri As Integer = 176
'GESTION DU PLATEAU
'Adr = S7 & "DB97,DBB126[24]"
'Jeton = 505
'OPC_Etat_Plateau = DB_Cartographie.AddItem(Adr, Jeton)
Public Operande_Etat_Plateau As Integer = 206
'position du plateau
Public Operande_Position_Plateau As Integer = 180
'GESTION DU CONTROLE RESSORT
Public Operande_Etat_Controle_Ressort As Integer = 150

Public Operande_Valeur_Effort As Integer = 156

Public Operande_Valeur_Effort_Etalon As Integer = 152

'position du controle

Public Operande_Position_Controle As Integer = 184
'GESTION DE LA PINCE ROBOT
'      Adr = S7 & "DB97,DBB160"
'      Jeton = 505
'      OPC_Pince_Robot = DB_Cartographie.AddItem(Adr, Jeton)
Public Operande_Pince_Robot As Integer = 160

'GESTION DE LA PRISE ROBOT
'      Adr = S7 & "DB97,DBB161"
'      Jeton = 505
'      OPC_Prise_Robot = DB_Cartographie.AddItem(Adr, Jeton)
Public Operande_Prise_Robot
'GESTION DU POSAGE REFUS
Public Operande_Posage_Refus As Integer = 172

'GESTION DU CONTROLE RESSORT
Public Operande_Posage_Controle_Ressort As Integer = 174
'GESTION DES TIROIRS
'      Adr = S7 & "DB97,DBB165[3]"
'      Jeton = 505
'      OPC_Etat_Tiroirs = DB_Cartographie.AddItem(Adr, Jeton)
Public Operande_Etat_Tiroirs As Integer = 165
'GESTION DES COMPTEURS
'compteur de ressorts mauvais

```

```

Public Operande_Compteur_Ressorts As Integer = 168

Public Operande_Compteur_Nozzles As Integer = 170

'*****
' DB98
'*****

' Following production OPC items
' Création des items de suivi de production

'      For i = 0 To TableauReference.NbReference - 1
'          'L'application lit les compteurs de la référence 1
'          ' 16 : Valeur du mot de départ
'          ' 42 : Nb de décalage
'      8 : Nb de compteurs liés à la référence i
'KEP
'          Adr = S7 & "DB98,DBD" & CStr(32 + i * 32) & "[8]"
'S7
'Adr = S7 & "DB98,int" & CStr(16 + i * 42) & ",13"
'      Jeton = 502 + i
'          OPC_Suivi_Prod_Ref(i) = DB_Suivi_Prod.AddItem(Adr, Jeton)
'      Next i
Public Operande_Suivi_Prod_Ref As Integer = 0

'compteurs totalisateurs

'KEP
'      Adr = S7 & "DB98,DBD0[5]"
'      Jeton = 502
'      OPC_Suivi_Prod_Totalisateurs = DB_Suivi_Prod.AddItem(Adr, Jeton)
Public Operande_Suivi_Prod_Totalisateurs As Integer = 0

' Default OPC items
' Création des items de défaut

'Items à ajouter : mot d'échanges pour les défauts
'      Try
'          Adr = AdrDefautOPC
'          Jeton = 1
'          OPC_Defaut = DB_Defaut.AddItem(Adr, Jeton)

' Grafquets OPC Items
' Création des items pour les grafquets

'      Adr = AdrG70PC
'      Jeton = 1
'      OPC_Grafcet = DB_Grafcet.AddItem(Adr, Jeton)

' Real time database OPC Items
' Création des item pour le lien avec la BTR
'
'*****
'DB92
'*****

'mot de retour de base pour l'entrée Recherche UC lot et ctrl Gamme
'mot de retour de base pour la sortie valide OP

```

```

Public Operande_CodeRetourBaseEntree As Integer = 0

Public Operande_CodeRetourBaseSortie As Integer = 2

'bit demande enregistrement BTR validOP pour la voie 1
Public Operande_Dem_BTR_Voie1 As Integer = 4
Public Bit_Dem_BTR_Voie1 As Integer = 1
'bit demande enregistrement BTR validOP pour la voie 2
Public Operande_Dem_BTR_Voie2 As Integer = 4
Public Bit_Dem_BTR_Voie2 As Integer = 2

'nombre de pièces bonnes pour la voie 1
Public Operande_Nb_Bonnes_K71 As Integer = 8

'nombre de pièces bonnes pour la voie 2
Public Operande_Nb_Bonnes_K72 As Integer = 24
'nombre de pièces mauvaises pour la voie 1
Public Operande_Nb_Mauvaises_K71 As Integer = 12
'nombre de pièces mauvaises pour la voie 2
Public Operande_Nb_Mauvaises_K72 As Integer = 28

'string contenant le datamatrix de la K71
' Adr = S7 & "DB92,STRING40.33"
' Jeton = 134
'OPC_Data_K71 = DB_BTR.AddItem(Adr, Jeton)
Public Operande_Data_K71 As Integer = 40
'string contenant le datamatrix de la K71
'Adr = S7 & "DB92,STRING76.33"
'Jeton = 134
'OPC_Data_K72 = DB_BTR.AddItem(Adr, Jeton)
Public Operande_Data_K72 As Integer = 76

#End Region

End Module

```

2. Annexe 2 : Code la fenêtre SuiviProd2

```
Public Class FrmSuiviProd2

    Private Sub FrmSuiviProd2_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        MAJ_Affichage()
        Affiche_CPT()
    End Sub

#Region "Fonctions"
    Public Sub Affiche_CPT()

        '
        'refresh general counter
        'mise à jour du compteur totalisateur
        LectureDMot(DB_Process_TpsCycle, Operande_Cpt_Tot, 1, TablDMot)
        Applicom_Compteur_Totalisateur = TablDMot(0)
        TextBoxCompteurTotal.Text = Applicom_Compteur_Totalisateur

        LectureDMot(DB_Process_TpsCycle, Operande_Cpt_Tot_RAZ, 1, TablDMot)
        Applicom_Compteur_Totalisateur_Raz = TablDMot(0)
        TextBoxCompteurTotalRaz.Text = Applicom_Compteur_Totalisateur_Raz

        ' Mise à jour de chaque compteur

        'Compteur Cassette OK
        LectureDMot(DB_Compteurs, Operande_CassetteOK, 1, TablDMot)
        Applicom_CassetteOK = TablDMot(0)
        TxtCassetteOK.Text = Applicom_CassetteOK

        'Compteur Cassette NOK
        LectureDMot(DB_Compteurs, Operande_CassetteNOK, 1, TablDMot)
        Applicom_CassetteNOK = TablDMot(0)
        TxtCassetteNotOK.Text = Applicom_CassetteNOK

        'Compteur Nozzle OK
        LectureDMot(DB_Compteurs, Operande_NozzleOk, 1, TablDMot)
        Applicom_NozzleOK = TablDMot(0)
        TxtNozzleOK.Text = Applicom_NozzleOK

        'Compteur Nozzle NOK
        LectureDMot(DB_Compteurs, Operande_NozzleNok, 1, TablDMot)
        Applicom_NozzleNOK = TablDMot(0)
        TxtNozzleNotOK.Text = Applicom_NozzleNOK

        'Compteur Cassette NOK Traca
        LectureDMot(DB_Compteurs, Operande_CassetteNOKTraca, 1, TablDMot)
        Applicom_CassetteNOKTraca = TablDMot(0)
        TxtCassetteNotOKTraca.Text = Applicom_CassetteNOKTraca

        'Compteur Mapping Cassette OK
        LectureDMot(DB_Compteurs, Operande_MappingCassetteOK, 1, TablDMot)
        Applicom_MappingCassetteOK = TablDMot(0)
        TxtMappingCassetteOk.Text = Applicom_MappingCassetteOK

        'Compteur Mapping Cassette NOK
        LectureDMot(DB_Compteurs, Operande_MappingCassetteNOK, 1, TablDMot)
    End Sub
End Class
```

```

Applicom_MappingCassetteNOK = TablDMot(0)
TxtMappingCassetteNotOK.Text = Applicom_MappingCassetteNOK

'Compteur Absence Nozzle Mapping
LectureDMot(DB_Compteurs, Operande_AbsenceNozzleMapping, 1, TablDMot)
Applicom_AbsenceNozzleMapping = TablDMot(0)
TxtAbsenceNozzleMapping.Text = Applicom_AbsenceNozzleMapping

'Compteur Présence Nozzle Mapping
LectureDMot(DB_Compteurs, Operande_PresenceNozzleMapping, 1, TablDMot)
Applicom_PresenceNozzleMapping = TablDMot(0)
TxtPresenceNozzleMapping.Text = Applicom_PresenceNozzleMapping

'Compteur SpurOvertake Mapping
LectureDMot(DB_Compteurs, Operande_SpurOvertakeMapping, 1, TablDMot)
Applicom_SpurOvertakeMapping = TablDMot(0)
TxtSpurOvertakingMapping.Text = Applicom_SpurOvertakeMapping

'Compteur Spring On nozzle mapping
LectureDMot(DB_Compteurs, Operande_SpringOnNozzleMapping, 1, TablDMot)
Applicom_SpringOnNozzleMapping = TablDMot(0)
TxtSpringOnNozzleMapping.Text = Applicom_SpringOnNozzleMapping

'Compteur Ctrl Spring OK
LectureDMot(DB_Compteurs, Operande_CtrlSpringOK, 1, TablDMot)
Applicom_CtrlSpringOK = TablDMot(0)
TxtControleSpringOK.Text = Applicom_CtrlSpringOK

'Compteur Ctrl Spring NOK
LectureDMot(DB_Compteurs, Operande_CtrlSpringNOK, 1, TablDMot)
Applicom_CtrlSpringNOK = TablDMot(0)
TxtControleSpringNotOK.Text = Applicom_CtrlSpringNOK

'Compteur Spring in robot gripper ok
LectureDMot(DB_Compteurs, Operande_SpringInRobGripperOK, 1, TablDMot)
Applicom_SpringInRobGripperOK = TablDMot(0)
TxtSpringInRobotGripperOK.Text = Applicom_SpringInRobGripperOK

'Compteur Spring in robot gripper Nok
LectureDMot(DB_Compteurs, Operande_SpringInRobGripperNOK, 1, TablDMot)
Applicom_SpringInRobGripperNOK = TablDMot(0)
TxtSpringInRobotGripperNotOK.Text = Applicom_SpringInRobGripperNOK

'Compteur Repeat ctrl spring ok
LectureDMot(DB_Compteurs, Operande_RepeatCtrlSpringOK, 1, TablDMot)
Applicom_RepeatCtrlSpringOK = TablDMot(0)
TxtRepeatControleSpringOK.Text = Applicom_RepeatCtrlSpringOK

'Compteur Repeat ctrl spring Nok
LectureDMot(DB_Compteurs, Operande_RepeatCtrlSpringNOK, 1, TablDMot)
Applicom_RepeatCtrlSpringNOK = TablDMot(0)
TxtRepeatControleSpringNotOK.Text = Applicom_RepeatCtrlSpringNOK

'Compteur Simple Ctrl Master Sensor OK
LectureDMot(DB_Compteurs, Operande_SimplCtrlMastSensorOK, 1, TablDMot)
Applicom_SimplCtrlMastSensorOK = TablDMot(0)
TxtSimpleControleMasterSensorOK.Text = Applicom_SimplCtrlMastSensorOK

'Compteur Simple Ctrl Master Sensor NOK
LectureDMot(DB_Compteurs, Operande_SimplCtrlMastSensorNOK, 1, TablDMot)

```

```

Applicom_SimplCtrlMastSensorNOK = TablDMot(0)
TxtSimpleControleMasterSensorNotOK.Text = Applicom_SimplCtrlMastSensorNOK

'Compteur Repeat Ctrl Mast Sensor OK
LectureDMot(DB_Compteurs, Operande_RepeatCtrlMastSensorOK, 1, TablDMot)
Applicom_RepeatCtrlMastSensorOK = TablDMot(0)
TxtRepeatControleMasterSensorOK.Text = Applicom_RepeatCtrlMastSensorOK

'Compteur Repeat Ctrl Mast Sensor NOK
LectureDMot(DB_Compteurs, Operande_RepeatCtrlMastSensorNOK, 1, TablDMot)
Applicom_RepeatCtrlMastSensorNOK = TablDMot(0)
TxtRepeatControleMasterSensorNotOK.Text = Applicom_RepeatCtrlMastSensorNOK

'Compteur Ctrl Master Cam OK
LectureDMot(DB_Compteurs, Operande_CtrlMasterCamOK, 1, TablDMot)
Applicom_CtrlMasterCamOK = TablDMot(0)
TxtControleMasterCameraOK.Text = Applicom_CtrlMasterCamOK

'Compteur Ctrl Master Cam NOK
LectureDMot(DB_Compteurs, Operande_CtrlMasterCamNOK, 1, TablDMot)
Applicom_CtrlMasterCamNOK = TablDMot(0)
TxtControleMasterCameraNotOK.Text = Applicom_CtrlMasterCamNOK

'Compteur Repeat Ctrl numeric axis OK
LectureDMot(DB_Compteurs, Operande_RepeatCtrlNumericAxisOK, 1, TablDMot)
Applicom_RepeatCtrlNumericAxisOK = TablDMot(0)
TxtRepeatControleNumericAxisOK.Text = Applicom_RepeatCtrlNumericAxisOK

'Compteur Repeat Ctrl numeric axis NOK
LectureDMot(DB_Compteurs, Operande_RepeatCtrlNumericAxisNOK, 1, TablDMot)
Applicom_RepeatCtrlNumericAxisNOK = TablDMot(0)
TxtRepeatControleNumericAxisNotOK.Text = Applicom_RepeatCtrlNumericAxisNOK

'Compteur Master cassette OK
LectureDMot(DB_Compteurs, Operande_CtrlMasterCassetteOK, 1, TablDMot)
Applicom_CtrlMasterCassetteOK = TablDMot(0)
TxtControleMasterCassetteOK.Text = Applicom_CtrlMasterCassetteOK

'Compteur Master cassette NOK
LectureDMot(DB_Compteurs, Operande_CtrlMasterCassetteNOK, 1, TablDMot)
Applicom_CtrlMasterCassetteNOK = TablDMot(0)
TxtControleMasterCassetteNotOK.Text = Applicom_CtrlMasterCassetteNOK

```

End Sub

Public Sub RemiseAZeroCPT()

```

' Mise à 0 de chaque compteur

' Valeur 0 qi sera attribuée à chaque compteur
TablDMot(0) = 0

' Mise à 0 du compteur totalisateur avec RAZ
EcritureDMot(DB_Process_TpsCycle, Operande_Cpt_Tot_RAZ, 1, TablDMot)
TextBoxCompteurTotalRaz.Text = 0

'Compteur Cassette OK
EcritureDMot(DB_Compteurs, Operande_CassetteOK, 1, TablDMot)
TxtCassetteOK.Text = 0

'Compteur Cassette NOK
EcritureDMot(DB_Compteurs, Operande_CassetteNOK, 1, TablDMot)

```

```

TxtCassetteNotOK.Text = 0

'Compteur Nozzle OK
EcritureDMot(DB_Compteurs, Operande_NozzleOK, 1, TablDMot)
TxtNozzleOK.Text = 0

'Compteur Nozzle NOK
EcritureDMot(DB_Compteurs, Operande_NozzleNOK, 1, TablDMot)
TxtNozzleNotOK.Text = 0

'Compteur Cassette NOK Traca
EcritureDMot(DB_Compteurs, Operande_CassetteNOKTraca, 1, TablDMot)
TxtCassetteNotOKTraca.Text = 0

'Compteur Mapping Cassette OK
EcritureDMot(DB_Compteurs, Operande_MappingCassetteOK, 1, TablDMot)
TxtMappingCassetteOk.Text = 0

'Compteur Mapping Cassette NOK
EcritureDMot(DB_Compteurs, Operande_MappingCassetteNOK, 1, TablDMot)
Applicom_MappingCassetteNOK = TablDMot(0)
TxtMappingCassetteNotOK.Text = 0

'Compteur Absence Nozzle Mapping
EcritureDMot(DB_Compteurs, Operande_AbsenceNozzleMapping, 1, TablDMot)
TxtAbsenceNozzleMapping.Text = 0

'Compteur Présence Nozzle Mapping
EcritureDMot(DB_Compteurs, Operande_PresenceNozzleMapping, 1, TablDMot)
TxtPresenceNozzleMapping.Text = 0

'Compteur SpurOvertake Mapping
EcritureDMot(DB_Compteurs, Operande_SpurOvertakeMapping, 1, TablDMot)
TxtSpurOvertakingMapping.Text = 0

'Compteur Spring On nozzle mapping
EcritureDMot(DB_Compteurs, Operande_SpringOnNozzleMapping, 1, TablDMot)
TxtSpringOnNozzleMapping.Text = 0

'Compteur Ctrl Spring OK
EcritureDMot(DB_Compteurs, Operande_CtrlSpringOK, 1, TablDMot)
TxtControleSpringOK.Text = 0

'Compteur Ctrl Spring NOK
EcritureDMot(DB_Compteurs, Operande_CtrlSpringNOK, 1, TablDMot)
TxtControleSpringNotOK.Text = 0

'Compteur Spring in robot gripper ok
EcritureDMot(DB_Compteurs, Operande_SpringInRobGripperOK, 1, TablDMot)
TxtSpringInRobotGripperOK.Text = 0

'Compteur Spring in robot gripper Nok
EcritureDMot(DB_Compteurs, Operande_SpringInRobGripperNOK, 1, TablDMot)
TxtSpringInRobotGripperNotOK.Text = 0

'Compteur Repeat ctrl spring ok
EcritureDMot(DB_Compteurs, Operande_RepeatCtrlSpringOK, 1, TablDMot)
TxtRepeatControleSpringOK.Text = 0

'Compteur Repeat ctrl spring Nok

```



```

EcritureDMot(DB_Compteurs, Operande_RepeatCtrlSpringNOK, 1, TablDMot)
TxtRepeatControleSpringNotOK.Text = 0

'Compteur Simple Ctrl Master Sensor OK
EcritureDMot(DB_Compteurs, Operande_SimplCtrlMastSensorOK, 1, TablDMot)
TxtSimpleControleMasterSensorOK.Text = 0

'Compteur Simple Ctrl Master Sensor NOK
EcritureDMot(DB_Compteurs, Operande_SimplCtrlMastSensorNOK, 1, TablDMot)
TxtSimpleControleMasterSensorNotOK.Text = 0

'Compteur Repeat Ctrl Mast Sensor OK
EcritureDMot(DB_Compteurs, Operande_RepeatCtrlMastSensorOK, 1, TablDMot)
TxtRepeatControleMasterSensorOK.Text = 0

'Compteur Repeat Ctrl Mast Sensor NOK
EcritureDMot(DB_Compteurs, Operande_RepeatCtrlMastSensorNOK, 1, TablDMot)
TxtRepeatControleMasterSensorNotOK.Text = 0

'Compteur Ctrl Master Cam OK
EcritureDMot(DB_Compteurs, Operande_CtrlMasterCamOK, 1, TablDMot)
TxtControleMasterCameraOK.Text = 0

'Compteur Ctrl Master Cam NOK
EcritureDMot(DB_Compteurs, Operande_CtrlMasterCamNOK, 1, TablDMot)
TxtControleMasterCameraNotOK.Text = 0

'Compteur Repeat Ctrl numeric axis OK
EcritureDMot(DB_Compteurs, Operande_RepeatCtrlNumericAxisOK, 1, TablDMot)
TxtRepeatControleNumericAxisOK.Text = 0

'Compteur Repeat Ctrl numeric axis NOK
EcritureDMot(DB_Compteurs, Operande_RepeatCtrlNumericAxisNOK, 1, TablDMot)
TxtRepeatControleNumericAxisNotOK.Text = 0

'Compteur Master cassette OK
EcritureDMot(DB_Compteurs, Operande_CtrlMasterCassetteOK, 1, TablDMot)
TxtControleMasterCassetteOK.Text = 0

'Compteur Master cassette NOK
EcritureDMot(DB_Compteurs, Operande_CtrlMasterCassetteNOK, 1, TablDMot)
TxtControleMasterCassetteNotOK.Text = 0
End Sub

''' <summary>
''' Multilangue function
''' Fonction de chargement des libellés en fonction de la langue en cours
''' </summary>
''' <remarks></remarks>
Private Sub MAJ_Affichage()
    'attention le fichier .ini est décalé de 1 par rapport au tableau (indice 0
dans le tableau)
    lblTitre.Text =
TableauAffichage.GroupAffichage(6).NomGroupeL(NumLangueEnCours)

    lblLegendeTableau.Text =
TableauAffichage.GroupAffichage(17).Données(0).LibelleL(NumLangueEnCours)
    lblCompteurTotal.Text =
TableauAffichage.GroupAffichage(17).Données(1).LibelleL(NumLangueEnCours)
    BF10.Text =
TableauAffichage.GroupAffichage(17).Données(5).LibelleL(NumLangueEnCours)

```

```

        BPClose.Text =
TableauAffichage.GroupAffichage(17).Données(6).LibelleL(NumLangueEnCours)

        ' Affichage mis à jour par FG
        LblGeneral.Text =
TableauAffichage.GroupAffichage(17).Données(7).LibelleL(NumLangueEnCours)
        LblCassette.Text =
TableauAffichage.GroupAffichage(17).Données(8).LibelleL(NumLangueEnCours)
        LblNozzle.Text =
TableauAffichage.GroupAffichage(17).Données(9).LibelleL(NumLangueEnCours)
        LblDivers.Text =
TableauAffichage.GroupAffichage(17).Données(10).LibelleL(NumLangueEnCours)
        LblSpurOvertakingMapping.Text =
TableauAffichage.GroupAffichage(17).Données(11).LibelleL(NumLangueEnCours)
        LblSpringOnNozzleMapping.Text =
TableauAffichage.GroupAffichage(17).Données(12).LibelleL(NumLangueEnCours)
        LblCassetteNotOKTraca.Text =
TableauAffichage.GroupAffichage(17).Données(13).LibelleL(NumLangueEnCours)
        LblProcess.Text =
TableauAffichage.GroupAffichage(17).Données(14).LibelleL(NumLangueEnCours)
        LblMappingCassette.Text =
TableauAffichage.GroupAffichage(17).Données(15).LibelleL(NumLangueEnCours)
        LblControleSpring.Text =
TableauAffichage.GroupAffichage(17).Données(16).LibelleL(NumLangueEnCours)
        LblSpringInRobotGripper.Text =
TableauAffichage.GroupAffichage(17).Données(17).LibelleL(NumLangueEnCours)
        LblControle.Text =
TableauAffichage.GroupAffichage(17).Données(18).LibelleL(NumLangueEnCours)
        LblRepeatControleSpring.Text =
TableauAffichage.GroupAffichage(17).Données(19).LibelleL(NumLangueEnCours)
        LblSimpleControleMasterSensor.Text =
TableauAffichage.GroupAffichage(17).Données(20).LibelleL(NumLangueEnCours)
        LblRepeatControleMasterSensor.Text =
TableauAffichage.GroupAffichage(17).Données(21).LibelleL(NumLangueEnCours)
        LblControleMasterCamera.Text =
TableauAffichage.GroupAffichage(17).Données(22).LibelleL(NumLangueEnCours)
        LblRepeatControleNumericAxis.Text =
TableauAffichage.GroupAffichage(17).Données(23).LibelleL(NumLangueEnCours)
        LblControleMasterCassette.Text =
TableauAffichage.GroupAffichage(17).Données(24).LibelleL(NumLangueEnCours)
        LblAbsenceNozzleMapping.Text =
TableauAffichage.GroupAffichage(17).Données(25).LibelleL(NumLangueEnCours)
        LblPresenceNozzleMapping.Text =
TableauAffichage.GroupAffichage(17).Données(26).LibelleL(NumLangueEnCours)
        Label3.Text =
TableauAffichage.GroupAffichage(17).Données(27).LibelleL(NumLangueEnCours)

```

End Sub

#End Region

```

Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TxtAbsenceNozzleMapping.TextChanged

```

End Sub

```

Private Sub BF10_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles BF10.Click
    RemiseAZeroCPT()

```

End Sub

```
Private Sub BPClose_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BPClose.Click
    Me.Close()
End Sub
```

3. Annexe 3 : Fichier des langues Affichage.ini

FICHER AFFICHAGE MULTI LANGUE
MULTI LANGUAGE DISPLAY FILE

**** LEGENDE ****

[SectionXX]

L1NomSection = Nom de la page dans la langue 1

L2NomSection = Name of the page in language 2

L3NomSection = Dil 3'te sayfa adi

L4NomSection =

NbCleSection = Nombre de libellé dans cette section / Number of label in this section

L1NomCle1Section1 = Libellé du bouton ou parametre dans la langue 1

L2NomCle1Section1 = Label of buton or parameter in language 2

L3NomCle1Section1 = Buton yada parametrenin Dil 3 te etiketi

L4NomCle1Section1 =

[Entete]

'Nombre de page paramétrées

NbSection = 18

***** SOUS SECTYON 1 *****

[Section1]

L1NomSection = PAGE PRINCIPALE

L2NomSection = HOME PAGE

L3NomSection = ANA SAYFA

L4NomSection =

NbCleSection = 36

L1NomCle1 = Général

L2NomCle1 = Home Page

L3NomCle1 = ANA SAYFA

L4NomCle1 =

L1NomCle2 = Langue

L2NomCle2 = Language

L3NomCle2 = DIL SECIMI

L4NomCle2 =

L1NomCle3 = Référence :

L2NomCle3 = Reference :

L3NomCle3 = REFERANS

L4NomCle3 =
L1NomCle4 = Temps de cycle
L2NomCle4 = Cycle time
L3NomCle4 = CEVRIM ZAMANI
L4NomCle4 =
L1NomCle5 = Ýnstantané
L2NomCle5 = Last product
L3NomCle5 = SON URUN
L4NomCle5 =
L1NomCle6 = Suivi de production
L2NomCle6 = Following products
L3NomCle6 = TAKIP EDEN URUNLER
L4NomCle6 =
L1NomCle7 = Pièces
L2NomCle7 = Parts
L3NomCle7 = PARCALAR
L4NomCle7 =
L1NomCle8 = Bonnes
L2NomCle8 = Good
L3NomCle8 = IYI
L4NomCle8 =
L1NomCle9 = Mauvaises
L2NomCle9 = Bad
L3NomCle9 = KOTU
L4NomCle9 =
L1NomCle10 = RAZ
L2NomCle10 = Reset
L3NomCle10 = YENÝDEN BASLAT
L4NomCle10 =
L1NomCle11 = N°
L2NomCle11 = N°
L3NomCle11 = N°
L4NomCle11 =
L1NomCle12 = Désignation
L2NomCle12 = Designation
L3NomCle12 = ATAMA
L4NomCle12 =
L1NomCle13 = Station
L2NomCle13 = Station
L3NomCle13 = ÝSTASYON
L4NomCle13 =
L1NomCle14 = Niveau
L2NomCle14 = Level
L3NomCle14 = ASAMA
L4NomCle14 =

L1NomCle15 = Etalonnage
L2NomCle15 = Calibration
L3NomCle15 = KALÝBRASYON
L4NomCle15 =
L1NomCle16 = Demande d'Etalonnage Manuel
L2NomCle16 = Manual calibration request
L3NomCle16 = MANUEL KALÝBRASYON ÝSTEGÝ
L4NomCle16 =
L1NomCle17 = Nb Pièces avant Alerte étalonnage
L2NomCle17 = Nb of parts before calibration alert
L3NomCle17 = KALIBRASYON ALARMI ONCESI ADET
L4NomCle17 =
L1NomCle18 = Nb Pièces avant étalonnage
L2NomCle18 = Nb of parts before calibration
L3NomCle18 = KALIBRASYON ONCESÝ ADET
L4NomCle18 =
L1NomCle19 = Vidage
L2NomCle19 = Emptying
L3NomCle19 = BOSALTMA
L4NomCle19 =
L1NomCle20 = Cellule
L2NomCle20 = Cell
L3NomCle20 = HUCRE
L4NomCle20 =
L1NomCle21 = Eclairage Cellule
L2NomCle21 = Lighting cell
L3NomCle21 = AYDINLATMA HUCRESI
L4NomCle21 =
L1NomCle22 = Eclairage Chargement
L2NomCle22 = Lighting Load station
L3NomCle22 = AYDINLATMA YUKLEME ISTASYONU
L4NomCle22 =
L1NomCle23 = Base temps réel
L2NomCle23 = Real Time Data base
L3NomCle23 = GERCEK ZAMANLI VERI TABANI
L4NomCle23 =
L1NomCle24 = Gestion de lots
L2NomCle24 = Batch
L3NomCle24 = PARTÝ
L4NomCle24 =
L1NomCle25 = Lot en cours
L2NomCle25 = Current Batch
L3NomCle25 = MEVCUT PARTÝ
L4NomCle25 =
L1NomCle26 = Valider lot

L2NomCle26 = Valid Batch
L3NomCle26 = PARTÝÝÝ ONAYLA
L4NomCle26 =
L1NomCle27 = Etat BTR
L2NomCle27 = RTDB state
L3NomCle27 = GZVT durumu
L4NomCle27 =
L1NomCle28 = Code CtrlGamme
L2NomCle28 = CtrlGamme result
L3NomCle28 = CtrlGamme SonuClari
L4NomCle28 =
L1NomCle29 = Code ValidOP
L2NomCle29 = ValidOP result
L3NomCle29 = Onaylama operasyonu sonuClari
L4NomCle29 =
L1NomCle30 = Vidage du plateau
L2NomCle30 = Plate emptying
L3NomCle30 =
L4NomCle30 =
L1NomCle31 = Vitesse Robot
L2NomCle31 = Robot Speed
L3NomCle31 =
L4NomCle31 =
L1NomCle32 = Init
L2NomCle32 = Initialization
L3NomCle32 =
L4NomCle32 =
L1NomCle33 = Auto
L2NomCle33 = Automatic Cycle
L3NomCle33 =
L4NomCle33 =
L1NomCle34 = Réglages
L2NomCle34 = Settings
L3NomCle34 =
L4NomCle34 =
L1NomCle35 = Arret Fin de cycle
L2NomCle35 = End of Cycl
L3NomCle35 =
L4NomCle35 =
L1NomCle36 = Acquittement
L2NomCle36 = Clearance
L3NomCle36 =
L4NomCle36 =

***** SOUS SECTYON 2 *****

[Section2]

L1NomSection = ECRAN CARTOGRAPHIE

L2NomSection = MAPPING SCREEN

L3NomSection = HARÝTA

L4NomSection =

NbCleSection = 66

L1NomCle1 = Légende

L2NomCle1 = Legend

L3NomCle1 = YETKÝLER

L4NomCle1 =

L1NomCle2 = Pas de pce

L2NomCle2 = No part

L3NomCle2 = PARCA YOK

L4NomCle2 =

L1NomCle3 = A faire

L2NomCle3 = Do work

L3NomCle3 = ÝSLEM YAP

L4NomCle3 =

L1NomCle4 = Bonne

L2NomCle4 = Good

L3NomCle4 = ÝÝÝ

L4NomCle4 =

L1NomCle5 = Pièce faite

L2NomCle5 = Parts made

L3NomCle5 = ÝSLENECEK PARCA

L4NomCle5 =

L1NomCle6 = Pièce refusée

L2NomCle6 = Parts rejected

L3NomCle6 = RET PARCA

L4NomCle6 =

L1NomCle7 = Poste Activé

L2NomCle7 = Active station

L3NomCle7 = AKTÝF ÝSTASYON

L4NomCle7 =

L1NomCle8 = Poste Désactivé

L2NomCle8 = Ýnactive station

L3NomCle8 = PASÝF ÝSTASYON

L4NomCle8 =

L1NomCle9 = Etat Posages

L2NomCle9 = Fittings state

L3NomCle9 = POSAJ DURUMU

L4NomCle9 =

L1NomCle10 = Posage 1
L2NomCle10 = Fitting 1
L3NomCle10 = POSAJ 1
L4NomCle10 =
L1NomCle11 = Posage 2
L2NomCle11 = Fitting 2
L3NomCle11 = POSAJ 2
L4NomCle11 =
L1NomCle12 = Posage 3
L2NomCle12 = Fitting 3
L3NomCle12 = POSAJ 3
L4NomCle12 =
L1NomCle13 = Posage 4
L2NomCle13 = Fitting 4
L3NomCle13 = POSAJ 4
L4NomCle13 =
L1NomCle14 = Modes dégradés
L2NomCle14 = Degraded modes :
L3NomCle14 = BOZULMUS MODLAR
L4NomCle14 =
L1NomCle15 = Mode Dégradé 1
L2NomCle15 = Degraded mode 1
L3NomCle15 = BOZULMUS MOD 1
L4NomCle15 =
L1NomCle16 = Mode Dégradé 2
L2NomCle16 = Degraded mode 2
L3NomCle16 = BOZULMUS MOD 2
L4NomCle16 =
L1NomCle17 = Mode Dégradé 3
L2NomCle17 = Degraded mode 3
L3NomCle17 = BOZULMUS MOD 3
L4NomCle17 =
L1NomCle18 = Mode Dégradé 4
L2NomCle18 = Degraded mode 4
L3NomCle18 = BOZULMUS MOD 4
L4NomCle18 =
L1NomCle19 = Références
L2NomCle19 = References
L3NomCle19 = REFERANSLAR
L4NomCle19 =
L1NomCle20 = Gravage Datamatrix sur CPÝ
L2NomCle20 = Engraving Data Matrix on NHB
L3NomCle20 = NHB DATAMATRÝX YAZÝMÝ
L4NomCle20 =
L1NomCle21 = Laser

L2NomCle21 = Laser
L3NomCle21 = LAZER
L4NomCle21 =
L1NomCle22 = Lecture Data
L2NomCle22 = Reading Data Matrix
L3NomCle22 = DATA MATRÝX OKUMA
L4NomCle22 =
L1NomCle23 = Chargement / Déchargement
L2NomCle23 = Loading / Unloading
L3NomCle23 = YUKLEME / BOSALTMA
L4NomCle23 =
L1NomCle24 = Caméra
L2NomCle24 = Camera
L3NomCle24 = KAMERA
L4NomCle24 =
L1NomCle25 = RAZ CELLULE
L2NomCle25 = RESET CELL
L3NomCle25 = HUCREYÝ YENÝDEN BASLAT
L4NomCle25 =
L1NomCle26 = VALÝDE CARTOGRAPHÝE
L2NomCle26 = VALÝD MAPPÝNG
L3NomCle26 = HARÝTAYÝ ONAYLA
L4NomCle26 =
L1NomCle27 = Pièce incertaine
L2NomCle27 = Unknown status
L3NomCle27 = BELÝRSÝZ PARCA
L4NomCle27 =
L1NomCle28 = Refus Datamatrix
L2NomCle28 = Datamatrix default
L3NomCle28 = DATAMATRÝX VARSAYÝLAN
L4NomCle28 =
L1NomCle29 = Refus vision
L2NomCle29 = NHB type camera default
L3NomCle29 = nhb KAMERA TÝPÝ: VARSAYÝLAN
L4NomCle29 =
L1NomCle30 = A retourner
L2NomCle30 = Bad Position
L3NomCle30 = KOTU POZÝSYON
L4NomCle30 =
L1NomCle31 = Etalon
L2NomCle31 = Master
L3NomCle31 = Mastar
L4NomCle31 =
L1NomCle32 = BTR
L2NomCle32 = RTDB

L3NomCle32 = GZVT
L4NomCle32 =
L1NomCle33 = Rework
L2NomCle33 = Rework
L3NomCle33 = Rework
L4NomCle33 =
L1NomCle34 = Chargement ressorts nozzle
L2NomCle34 = Spring loaded nozzle
L3NomCle34 =
L4NomCle34 =
L1NomCle35 = Code reference
L2NomCle35 = Reference Code
L3NomCle35 =
L4NomCle35 =
L1NomCle36 = Trappe 1
L2NomCle36 = Trapdoor 1
L3NomCle36 =
L4NomCle36 =
L1NomCle37 = Trappe 2
L2NomCle37 = Trapdoor 2
L3NomCle37 =
L4NomCle37 =
L1NomCle38 = Trappe 3
L2NomCle38 = Trapdoor 3
L3NomCle38 =
L4NomCle38 =
L1NomCle39 = Controle ressort
L2NomCle39 = Spring Control
L3NomCle39 =
L4NomCle39 =
L1NomCle40 = Effort
L2NomCle40 = Effort
L3NomCle40 =
L4NomCle40 =
L1NomCle41 = Position axe controle
L2NomCle41 = Axis controle position
L3NomCle41 =
L4NomCle41 =
L1NomCle42 = Gestion des références
L2NomCle42 = Reference management
L3NomCle42 =
L4NomCle42 =
L1NomCle43 = Ref Produit active
L2NomCle43 = Active reference
L3NomCle43 =

L4NomCle43 =
L1NomCle44 = Plateau Ressorts
L2NomCle44 = Springs tray
L3NomCle44 =
L4NomCle44 =
L1NomCle45 = Distributeur ressorts
L2NomCle45 = Springs distributor
L3NomCle45 =
L4NomCle45 =
L1NomCle46 = Point de prise robot
L2NomCle46 = Point robot outlet
L3NomCle46 =
L4NomCle46 =
L1NomCle47 = Controle presence ressort
L2NomCle47 = Presence spring control
L3NomCle47 =
L4NomCle47 =
L1NomCle48 = Goulotte refus
L2NomCle48 = Chute refusal
L3NomCle48 =
L4NomCle48 =
L1NomCle49 = Refus
L2NomCle49 = Refusal
L3NomCle49 =
L4NomCle49 =
L1NomCle50 = Nb refus nozzles
L2NomCle50 = Nb nozzles refused
L3NomCle50 =
L4NomCle50 =
L1NomCle51 = Nb refus Ressorts
L2NomCle51 = Nb spring refused
L3NomCle51 =
L4NomCle51 =
L1NomCle52 = Cassette 1
L2NomCle52 = Cassette 1
L3NomCle52 =
L4NomCle52 =
L1NomCle53 = Cassette 2
L2NomCle53 = Cassette 2
L3NomCle53 =
L4NomCle53 =
L1NomCle54 = Code Reference
L2NomCle54 = Reference Code
L3NomCle54 =
L4NomCle54 =

L1NomCle55 = Etat
L2NomCle55 = State
L3NomCle55 =
L4NomCle55 =
L1NomCle56 = Emplacement a faire
L2NomCle56 = To do location
L3NomCle56 =
L4NomCle56 =
L1NomCle57 = En cours
L2NomCle57 = In process
L3NomCle57 =
L4NomCle57 =
L1NomCle58 = Ressorts
L2NomCle58 = Springs
L3NomCle58 =
L4NomCle58 =
L1NomCle59 = Defaut aiguille ou absence ressort
L2NomCle59 = needle default or spring missing
L3NomCle59 =
L4NomCle59 =
L1NomCle60 = Defaut nozzle
L2NomCle60 = Nozzle default
L3NomCle60 =
L4NomCle60 =
L1NomCle61 = Reserve
L2NomCle61 = Stockpile
L3NomCle61 =
L4NomCle61 =
L1NomCle62 = Posage inhibe
L2NomCle62 = laying inhibited
L3NomCle62 =
L4NomCle62 =
L1NomCle63 = Nozzle refusé
L2NomCle63 = Nozzle refused
L3NomCle63 =
L4NomCle63 =
L1NomCle64 = Trappe
L2NomCle64 = Trapdoor
L3NomCle64 =
L4NomCle64 =
L1NomCle65 = Trappe ouverte
L2NomCle65 = Opened trapdoor
L3NomCle65 =
L4NomCle65 =
L1NomCle66 = Trappe fermée

L2NomCle66 = Close trapdoor
L3NomCle66 =
L4NomCle66 =

***** SOUS SECTION 18 *****

[Section18]

L1NomSection = ECRAN SUIVI DE PRODUCTION 2
L2NomSection = FOLLOWING PARTS 2
L3NomSection = URETÝM TAKÝP EKRANÝ 2
L4NomSection =
NbCleSection = 28

L1NomCle1 = Tableau de suivi de production
L2NomCle1 = Following parts table
L3NomCle1 = URETÝM ÝZLEME TABOSU
L4NomCle1 =

L1NomCle2 = Compteur Totalisateur:
L2NomCle2 = Total Counter :
L3NomCle2 = TOPLAM SAYÝ :
L4NomCle2 =

L1NomCle3 = Libellé Référence
L2NomCle3 = Reference
L3NomCle3 = REFERANS
L4NomCle3 =

L1NomCle4 = Pièce Bonnes
L2NomCle4 = Good parts
L3NomCle4 = ÝÝÝ PARCA
L4NomCle4 =

L1NomCle5 = Pièces Mauvaises
L2NomCle5 = Bad parts
L3NomCle5 = KOTU PARCA
L4NomCle5 =

L1NomCle6 = F10 - RAZ COMPTEURS
L2NomCle6 = F10 - RESET COUNTERS
L3NomCle6 = F10 - SAYACLARÝ SÝFYRLA
L4NomCle6 =

L1NomCle7 = Fermer
L2NomCle7 = Closed
L3NomCle7 = KAPALÝ
L4NomCle7 =

L1NomCle8 = General
L2NomCle8 = General
L3NomCle8 =
L4NomCle8 =

L1NomCle9 = Cassette

L2NomCle9 = Cassette
L3NomCle9 =
L4NomCle9 =
L1NomCle10 = Nozzle
L2NomCle10 = Nozzle
L3NomCle10 =
L4NomCle10 =
L1NomCle11 = Divers
L2NomCle11 = Various
L3NomCle11 =
L4NomCle11 =
L1NomCle12 = Cartographie dépassement aiguille
L2NomCle12 = Spur overtaking mapping
L3NomCle12 =
L4NomCle12 =
L1NomCle13 = Cartographie ressort sur nozzle
L2NomCle13 = Spring on nozzle mapping
L3NomCle13 =
L4NomCle13 =
L1NomCle14 = Tracabilité cassette not OK
L2NomCle14 = Cassette not OK traca
L3NomCle14 =
L4NomCle14 =
L1NomCle15 = Process
L2NomCle15 = Process
L3NomCle15 =
L4NomCle15 =
L1NomCle16 = Cartographie Cassette
L2NomCle16 = Mapping Cassette
L3NomCle16 =
L4NomCle16 =
L1NomCle17 = Controle ressort
L2NomCle17 = Controle spring
L3NomCle17 =
L4NomCle17 =
L1NomCle18 = Ressort dans pince robot
L2NomCle18 = spring in robot Gripper
L3NomCle18 =
L4NomCle18 =
L1NomCle19 = Controle
L2NomCle19 = Control
L3NomCle19 =
L4NomCle19 =
L1NomCle20 = Controle répété des ressorts
L2NomCle20 = Repeat controle spring

L3NomCle20 =
L4NomCle20 =
L1NomCle21 = Capteur simple controle Master
L2NomCle21 = Simple controle master sensor
L3NomCle21 =
L4NomCle21 =
L1NomCle22 = Controle capteur master répété
L2NomCle22 = Repeat control master sensor
L3NomCle22 =
L4NomCle22 =
L1NomCle23 = Controle master camera
L2NomCle23 = Control master camera
L3NomCle23 =
L4NomCle23 =
L1NomCle24 = Controle répété de l'axe numérique
L2NomCle24 = Repeat controle numeric axis
L3NomCle24 =
L4NomCle24 =
L1NomCle25 = Controle master cassette
L2NomCle25 = Control master cassette
L3NomCle25 =
L4NomCle25 =
L1NomCle26 = Absence cartographie nozzle
L2NomCle26 = Absence Nozzle mapping
L3NomCle26 =
L4NomCle26 =
L1NomCle27 = Presence cartographie nozzle
L2NomCle27 = Presence nozzle mapping
L3NomCle27 =
L4NomCle27 =
L1NomCle28 = Total compteur avec RAZ
L2NomCle28 = Total counter with reset
L3NomCle28 =
L4NomCle28 =

**TESTS D'UNE UNITE
DE FERMENTATION INDUSTRIELLE
RAPPORT DE STAGE TECHNICIEN**



Pierre Guérin SAS

179 Grand Rue, 79210 MAUZE-LE-MIGNON
TUTEUR ENTREPRISE : M. Christophe Leblanc



GRETA Loire-Atlantique

16 rue Dufour, 44042 NANTES
TUTEUR GRETA : M. Thierry Mironneau

SOMMAIRE

SOMMAIRE	69
INTRODUCTION	1
PRÉSENTATION DE L'ENTREPRISE	2
1. Situation géographique	2
2. Organigramme du service Installation	2
3. Fiche d'identité	3
PRESENTATION DU PROJET	4
1. L'intérêt	4
2. Les étapes	4
DEFINITION DES MODES DE MARCHE	5
TEST DES STRATEGIES	7
1. Stérilisation à vide	8
2. Chargement média	9
3. Stérilisation média	10
4. Attente culture	11
5. Management de l'inoculation	12
6. Culture	13
7. Récolte	14
8. Décontamination	15
9. Managements acide, alcali, nutriment ou échantillon	16
10. Management agitateur pot de garniture	16
11. Remplissage double-enveloppe	17
IHM	18
. BOUCLES DE REGULATION	18
1. Actionneurs	19
2. Pompes et vannes	19
INTERLOCKS	20
Liste des contrôles et tests réalisés en atelier	21
CONCLUSION	22

INTRODUCTION

Le but de ce stage est de se familiariser avec une installation de process de fermentation. Le client est Albit Scientific, en Russie. Sur le site de Pierre Guérin, à Niort, nous allons réaliser les différents tests des automates, des commandes, des actionneurs, des capteurs, des joints puis tester les différentes stratégies à l'eau osmosée (stérilisation, remplissage, culture, récolte, etc...).

Les compétences requises sont la précision, les connaissances en électricité, en connectique, en hydraulique, en pneumatique, en automatisme et en informatique.

Tous les tests sont référencés dans un document que le client pourra consulter. Ils garantissent avant la mise en service la qualité et l'exactitude du montage réalisé par une autre équipe.

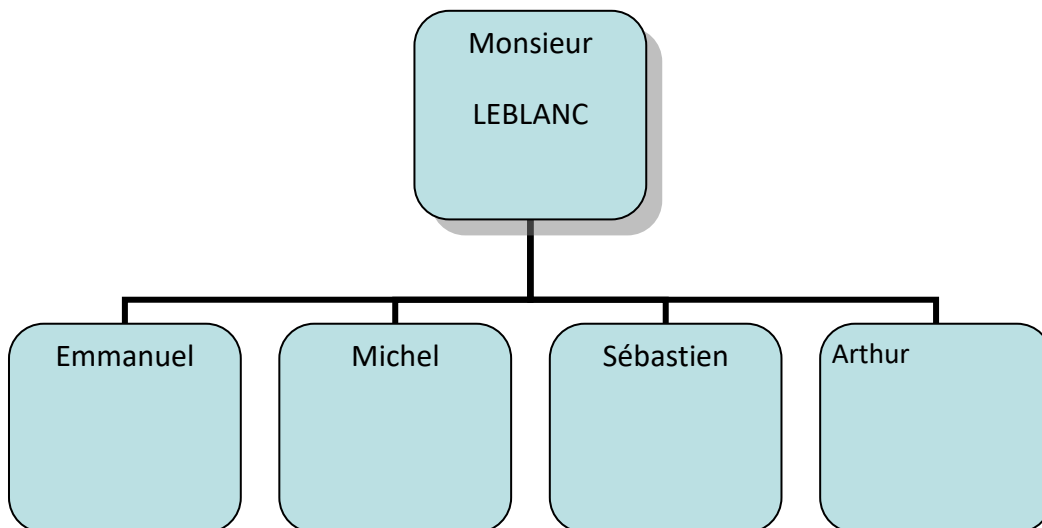
PRÉSENTATION DE L'ENTREPRISE

4. Situation géographique



Notre lieu de stage est l'usine Pierre Guérin de Niort. Le siège social est à Mauzé (79)

5. Organigramme du service Installation



6. Fiche d'identité

La société Pierre Guérin a été fondée en 1949 par Monsieur Pierre Guérin à Mauzé-sur-le-Mignon. L'usine de Niort, où je réalise mon stage, a été inaugurée en 1962. La société développe des valeurs de qualité, d'expertise, d'innovation et d'excellence dans son domaine. Elle a fortement investi sur les marchés internationaux.

L'entreprise propose des solutions globales pour équiper les entreprises en matériel dans les industries biotechnologiques, pharmaceutiques, cosmétiques et agroalimentaires. Elle propose des cuves en inox, des fermenteurs industriels, des équipements de laiterie, des cuves à vin, à fromage, des bioréacteurs, des stations de nettoyage.

La société maîtrise l'ensemble de la fabrication de ces systèmes, de l'étude à l'installation en passant par la fabrication.

La société possède trois sites de production : Niort (France), Burgos (Espagne) et St Cloud (USA). Elle fabrique plus de 2000 cuves par an.

Le cœur de métier est l'étude de la mécanique, de l'électricité et des automatismes, la fabrication des réseaux de tuyauterie, le câblage, la fabrication des cuves, le montage et les tests.

Pour chaque process les équipes réalisent les équipements d'agitation et de mélange, l'automation, les réseaux d'acheminement, les systèmes de nettoyage et de stérilisation.

Pierre Guérin est un acteur incontournable dans la pharmacie, les produits de soin, la laiterie-fromagerie, les produits alimentaires, la nutrition infantile et clinique, les vins et spiritueux, les biotechnologies.

De grands noms de l'industrie confient à Pierre Guérin la fabrication de systèmes : Andros, Coca Cola, Harry's, Lactalis, Unilever, Albit, Baxter, Institut Français du Pétrole, Sanofi Aventis...

PRESENTATION DU PROJET

7. L'intérêt

L'intérêt du projet est

- de se confronter à un système de process complexe
- de le comprendre de l'automate aux actionneurs grâce aux plans et à l'observation de l'installation
- d'apprendre à manipuler les actionneurs
- d'observer les différents capteurs
- d'utiliser un écran de contrôle.

8. Les étapes

Les étapes sont les suivantes :

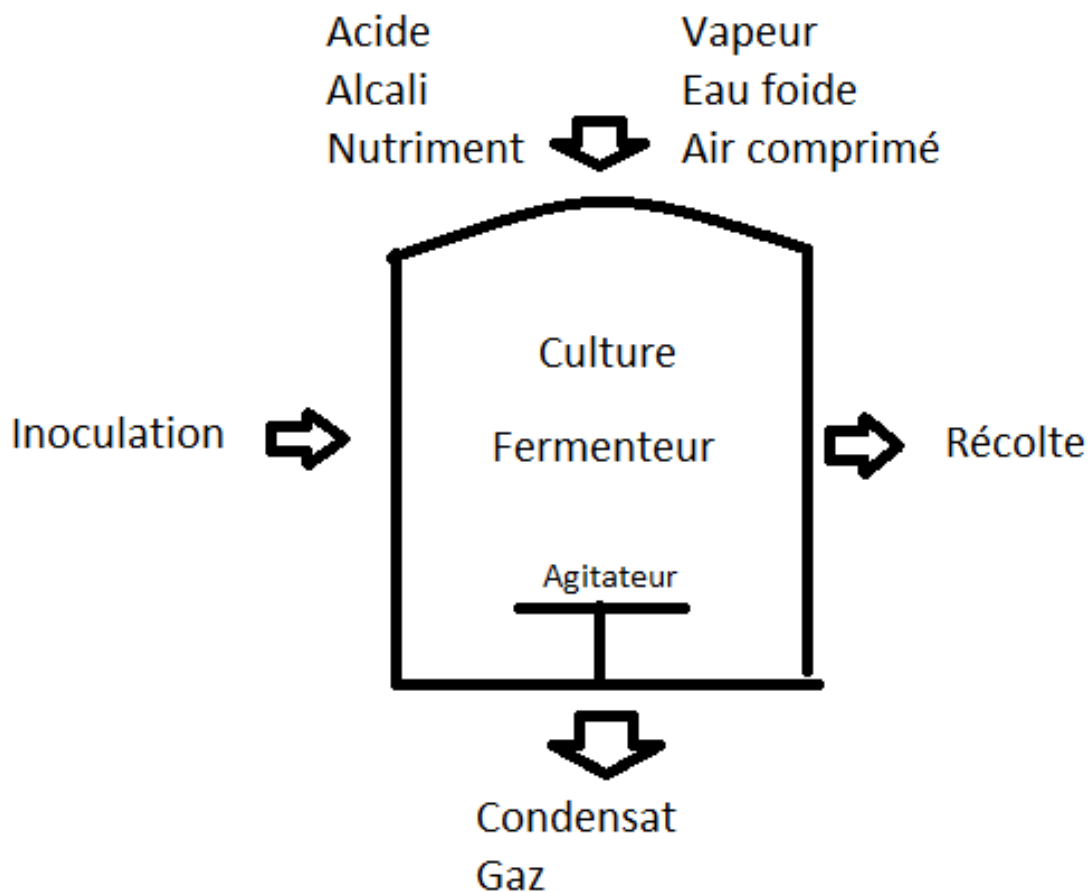
- Branchement des circuits d'eau, d'air, de vapeur et d'eau osmosée.
- Tests de l'ouverture et de la fermeture des vannes automatiques commandées par la console (test visuel).
- Contrôle du bon fonctionnement des agitateurs, réglage des pompes péristaltiques (étalonnage), réglage de la mesure du niveau de liquide dans les cuves (pression différentielle), tests des sondes pH et O2.
- Remplissage manuel en eau des pots de garniture et tests de fuites.
- Tests de tous les colliers de jonction des canalisations d'air avec un vaporisateur d'eau savonneuse (mille bulles).
- Test de remplissage de la cuve, contrôle visuel des fuites d'eau aux soudures et aux colliers.
- Mise en pression de chaque cuve. Contrôle du maintien de la pression.
- Lors de l'ouverture de chaque vanne, contrôle du maintien de la pression.
- Mise en eau du circuit de condensat (avant les tests en vapeur) pour tester l'étanchéité.
- Test de la double-enveloppe et de son circuit (avec la pompe).
- Test du circuit de garniture (avec la pompe).
- Réglage de la vitesse de rotation des agitateurs cuve remplie (avec un tachymètre).
- Mise en pression de la cuve remplie, les agitateurs fonctionnant.
- Mise en vapeur.
- Tests des stratégies.
- Corrections à apporter aux stratégies.

DEFINITION DES MODES DE MARCHE

L'installation va servir à produire de l'engrais pour lequel le système est conçu. L'ensemble comprend 2 cuves de 1500 litres et 1 cuve de 750 litres.

Un champignon microscopique est inoculé dans la cuve. On contrôle sa fermentation par la pression, la température, le pH, la dissolution en oxygène et par la vitesse de rotation. En sortie on obtient la récolte, c'est-à-dire l'engrais.

Voici le schéma simplifié d'un fermenteur :



En entrée on dispose de la vapeur pour stériliser et pour chauffer le circuit de la double-enveloppe. Il y a aussi l'inoculation, le substrat (appelé le milieu contenant le nutriment), l'air pour le process (introduit dans la cuve) et les instruments, l'acide et l'alcali pour contrôler le pH, l'eau froide (pour refroidir la double-enveloppe et pour le seal pot).

En sortie, on obtient la culture (récolte). Il sort également du condensat, du gaz (avec filtrage) et le résidu du nettoyage.

Le circulateur de la double-enveloppe de la cuve permet de contrôler la température (échangeur de vapeur pour chauffer l'eau du circuit de la double enveloppe ou eau froide pour le refroidir).

Le pot de garniture fournie de l'eau stérile à partir de vapeur d'eau à la garniture de la cuve, isolant celle-ci de l'air extérieur.

On chauffe ou on refroidit le contenu de la cuve par la double-enveloppe ce qui permet de contrôler la température. On introduit de l'air dans le process en contrôlant la pression. On contrôle la vitesse de rotation du mélangeur. L'agitateur permet de mélanger le contenu de la cuve. Une pompe sur le circuit de garniture permet la circulation d'eau stérile pour la garniture. Il y a une autre pompe sur le circuit d'eau de la double-enveloppe. Les pompes péristaltiques sont utilisées pour amener l'alcali et l'acide.

L'ensemble est piloté à partir d'un écran de contrôle pour chaque cuve. Les vannes sont manuelles ou automatiques, l'ensemble est équipé de rotamètres, de débitmètres (air), de manomètres (pression), de sondes O₂ et pH, de détendeurs vapeur, de pompes péristaltiques, de filtres, de pots de détente, de pots de garniture (seal pot). En fonctionnement normal, l'opérateur démarre une phase (stratégie) et l'automate gère l'ensemble des éléments décrits précédemment pour obtenir la récolte voulue, soit en pilotant les vannes et actionneurs automatiquement, soit en demandant par l'interface graphique à l'utilisateur d'ouvrir ou de fermer les vannes et actionneurs. 'Il y a les états suivants : démarrage en cours, marche, défaut, pause, repos et démarrage non autorisé.

Le réseau est en Profibus. L'automate est un Siemens S7.

Les contrôleurs de l'automate permettent de maîtriser la fabrication :

- Vitesse d'agitation
- Pression de la cuve
- pH alcali
- pH acide
- Taux d'oxygène
- Flux d'air interne
- Vitesse de la pompe à nutriments
- Température de la culture à l'intérieur de la cuve
- Température de la double-enveloppe
- Stérilisation

TEST DES STRATEGIES

Les stratégies sont les opérations différentes réalisées de manière semi-automatique pour la stérilisation, le chargement, la culture, etc. L'automate gère les vannes automatiques et l'interface utilisateur en demandant des actions manuelles (ouverture ou fermeture de vannes, etc...)

Les stratégies sont les suivantes :

- Chargement média
- Stérilisation média
- Attente culture
- Management de l'inoculation
- Culture
- Récolte
- Décontamination
- Management acide
- Management alcali
- Management nutrition
- Management échantillon
- Pot de garniture
- Remplissage double-enveloppe

Le management pot de garniture est lancé systématiquement et en parallèle de toute opération.

Le remplissage double-enveloppe se fait avant la culture.

Les phases managements acide, alcali, nutrition doivent être lancées avant la phase de culture.

Le but des autres opérations est d'alimenter l'installation en média, de lancer la culture et de faire la récolte.

La stérilisation média se fait également avant toutes ces opérations.

9. Stérilisation à vide

- Drainage
 - Augmentation de la température
 - Plateau température
 - Augmentation de la pression
 - Stérilisation plateau
 - Séchage filtres
 - Refroidissement et drainage
 - Refroidissement 2
 - Remplissage
 - Attente chargement
- Paramètre stérilisation à vide
- Timer du drainage de la double enveloppe
 - Timer à 100°C
 - Température de stérilisation
 - Timer de stérilisation
 - Timer de séchage des filtres
 - Choix de mode de stérilisation (temps)
 - Point de température refroidissement 1
 - Point de température refroidissement 2
 - Temps de fonctionnement de l'agitateur
 - Repos de l'agitateur
 - Flux d'air voulu

10.Chargement média

C'est le chargement du média dans le fermenteur par une ligne dédiée. Le transfert se fait avec pression d'air. Il y a la possibilité de faire la stérilisation avant.

- Setup ligne
 - Stérilisation ligne
 - Refroidissement ligne
 - Attente transfert
 - Chargement média
 - Attente décontamination ligne
 - Décontamination ligne
 - Décompression ligne
- Paramètres chargement média
- Mode fin de transfert (opérateur ou niveau)
 - Niveau à la fin du transfert
 - Flux d'air avant chargement média
 - Flux d'air pendant chargement média
 - Flux d'air après chargement média
 - Niveau pour lancer le refroidissement ou le réchauffement
 - Niveau pour lancer l'agitateur
 - Vitesse de l'agitateur
 - Pression
 - Température

11. Stérilisation média

Cette phase est permise par l'injection de vapeur d'eau dans la cuve et la double-enveloppe avec augmentation de la pression. S'ensuit un refroidissement à température de culture par circulation d'eau froide dans la double-enveloppe avec un minimum de pression d'air filtré.

- Drainage
 - Augmentation de la température
 - Plateau température
 - Stérilisation plateau
 - Séchage filtre
 - Refroidissement et drainage
 - Refroidissement 2
 - Remplissage double enveloppe
- Paramètres stérilisation média
- Timer du drainage de la double enveloppe
 - Timer à 100°C
 - Température stérilisation
 - Timer stérilisation
 - Timer séchage filtres
 - Point de température refroidissement 1
 - Point de température refroidissement 2
 - Choix de mode stérilisation
 - Vitesse de l'agitateur
 - Flux d'air

12.Attente culture

Cette phase calibre la sonde O2. Elle ajuste la sonde PH. Elle contrôle la stabilisation des boucles avant l'inoculation.

- Attente approbation calibration et lignes additifs prêtes
 - Ajustement pH média
 - Attente culture
- Paramètres attente culture
- Vitesse de l'agitateur en calibration
 - Flux d'air pendant la calibration (rien, surface, sparger)
 - Ajustement pH média (rien, acide, acide + alcali, alcali)
 - Contrôle pH durant l'attente (rien, acide, acide + alcali, alcali)
 - pH durant l'ajustement
 - pH durant l'attente
 - Vitesse de l'agitateur après calibration
 - Choix du flux d'air après calibration (rien, surface, sparger)
 - Flux d'air après calibration
 - Pression
 - Température
 - pH bas pour ajustement
 - pH haut pour ajustement

13. Management de l'inoculation

Cette phase stérilise et refroidit la ligne d'inoculation. Elle démarre le transfert d'inoculation. Elle démarre également la phase de culture et annule la phase d'attente culture. Elle permet enfin de décontaminer la ligne d'inoculation.

- Attente approbation calibration et ligne additifs prête
- Ajustement pH média
- Attente culture

- Paramètres de l'inoculation
 - Fin de transfert (opérateur ou niveau)
 - Niveau de fin de transfert
 - Timer de stérilisation de la ligne
 - Timer du refroidissement

14.Culture

Cette phase paramètre les conditions de culture dans le fermenteur afin d'assurer l'élevage de micro-organismes. Elle est stoppée par l'opérateur ou automatiquement.

- Culture
- Stop culture

- Paramètres culture
 - Vitesse de l'agitateur voulue durant la culture
 - Température voulue pendant la culture
 - Pression voulue pendant la culture
 - Mode contrôle pH (rien, acide, acide + alcali, alcali)
 - Flux d'air voulu pendant la culture (rien, surface, sparger)
 - Vitesse de rotation de la pompe à nutriments voulue pendant la culture
 - Contrôle O2 voulu pendant la culture (rien ou cascade)
 - Taux d'O2 voulu pendant la culture
 - Flux d'air voulu pendant la culture
 - Flux d'air voulu pendant l'attente
 - Temps de culture (0 : stop par utilisateur)
 - Vitesse de l'agitateur pendant l'attente
 - Température voulue pendant l'attente
 - Pression voulue pendant l'attente
 - Flux d'air voulu pendant l'attente (rien, surface, sparger)
 - Temps d'attente avant départ pompe à nutriments
 - Temps de fonctionnement de la pompe à nutriments

15.Récolte

Cette phase stérilise et refroidit la ligne avant le transfert. Elle transfert la production de la culture. Elle décontamine la ligne après transfert (air stérile).

- Setup ligne
 - Stérilisation ligne
 - Refroidissement ligne
 - Attente transfert
 - Transfert
 - Attente décontamination ligne
 - Décontamination ligne
 - Décompression ligne
- Paramètres de la récolte
- Fin de transfert (opérateur ou niveau)
 - Niveau de fin de transfert
 - Timer stérilisation ligne
 - Timer refroidissement ligne
 - Niveau de démarrage refroidissement ou chauffage
 - Niveau pour démarrer l'agitateur
 - Vitesse d'agitation
 - Pression
 - Température cuve

16. Décontamination

- Drainage
 - Augmentation température
 - Plateau température
 - Augmentation pression
 - Décontamination plateau
 - Séchage filtres
 - Refroidissement 1 et drainage
 - Refroidissement 2
- Paramètres décontamination
- Timer drainage double enveloppe
 - Timer 100°C
 - Température de décontamination
 - Timer décontamination
 - Timer séchage filtres
 - Température à la fin du refroidissement 1
 - Température à la fin du refroidissement 2
 - Mode de décontamination
 - Flux d'air
 - Temps d'agitation
 - Temps de repos d'agitation
 - Timer avant démarrage circulateur

17. Managements acide, alcali, nutriment ou échantillon

Cette phase connecte la ligne, puis stérilise et refroidit la ligne. Elle autorise l'injection et la règle. Enfin, elle désinfecte et refroidit la ligne après utilisation.

- Setup ligne
 - Stérilisation ligne
 - Refroidissement ligne
 - Attente transfert
 - Transfert
 - Attente décontamination ligne
 - Décontamination ligne
 - Décompression ligne
- Paramètres management
- Timer stérilisation ligne
 - Timer refroidissement ligne

18. Management agitateur pot de garniture

C'est une phase manuelle qui connecte la ligne de la garniture de l'agitateur. Il y a stérilisation et refroidissement de la ligne. Puis il y a une autorisation d'injecter la vapeur et de régler la ligne. Enfin il y a désinfection et refroidissement.

- Drainage pot de garniture
 - Drainage mécanique
 - Stérilisation pot de garniture
 - Pressurisation pot de garniture
 - Maintien en position pot de garniture
- Paramètres agitateur pot de garniture
- Timer drainage
 - Timer stérilisation
 - Timer production condensats
 - Timer pression OK

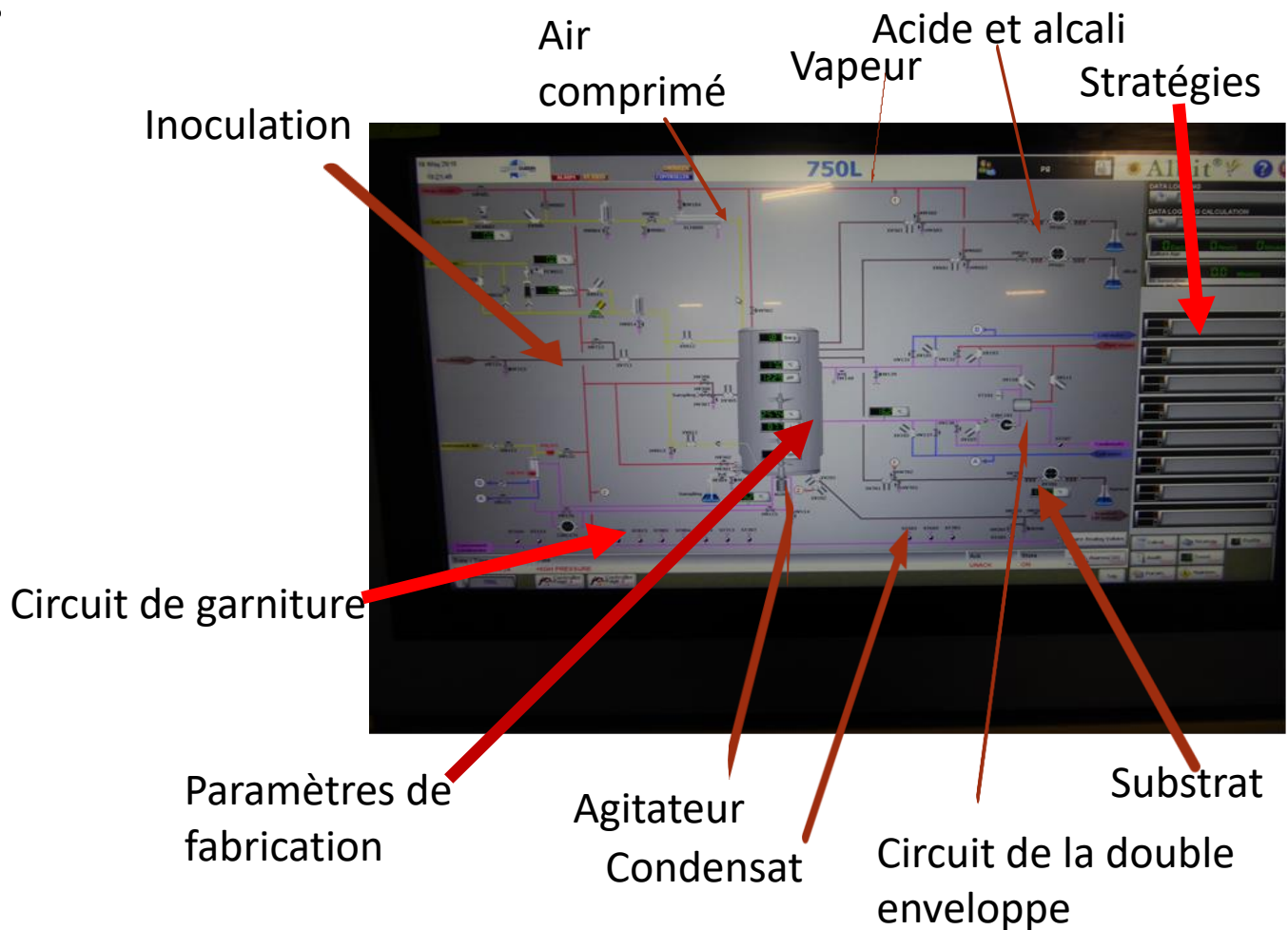
19. Remplissage double-enveloppe

Il se fait après stérilisation à vide et stérilisation média.

- Remplissage double-enveloppe 1
 - Remplissage double-enveloppe 2
 - Remplissage double enveloppe 3
- Paramètres remplissage double enveloppe
- Temps étape 1
 - Temps étape 2
 - Temps étape 3

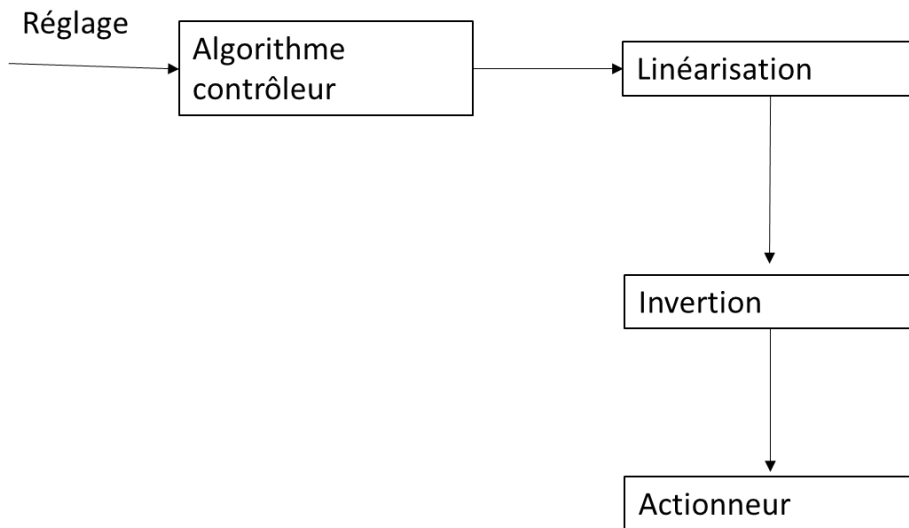
IHM

Voici la photo de l'écran IHM. La cuve, les pompes, les vannes, les entrées, les sorties sont représentées par des symboles. Lorsqu'on clique sur l'actionneur sélectionné, on peut à distance, s'il est automatique, modifier son état : ouvert ou fermé. On peut donc visualiser le plan de fonctionnement grâce à ce graphique. D'autres écrans indiquent les warnings, des pourcentages de degré de fonctionnement, etc... Dans son mode opérationnel, l'interface graphique permet de lancer les stratégies.

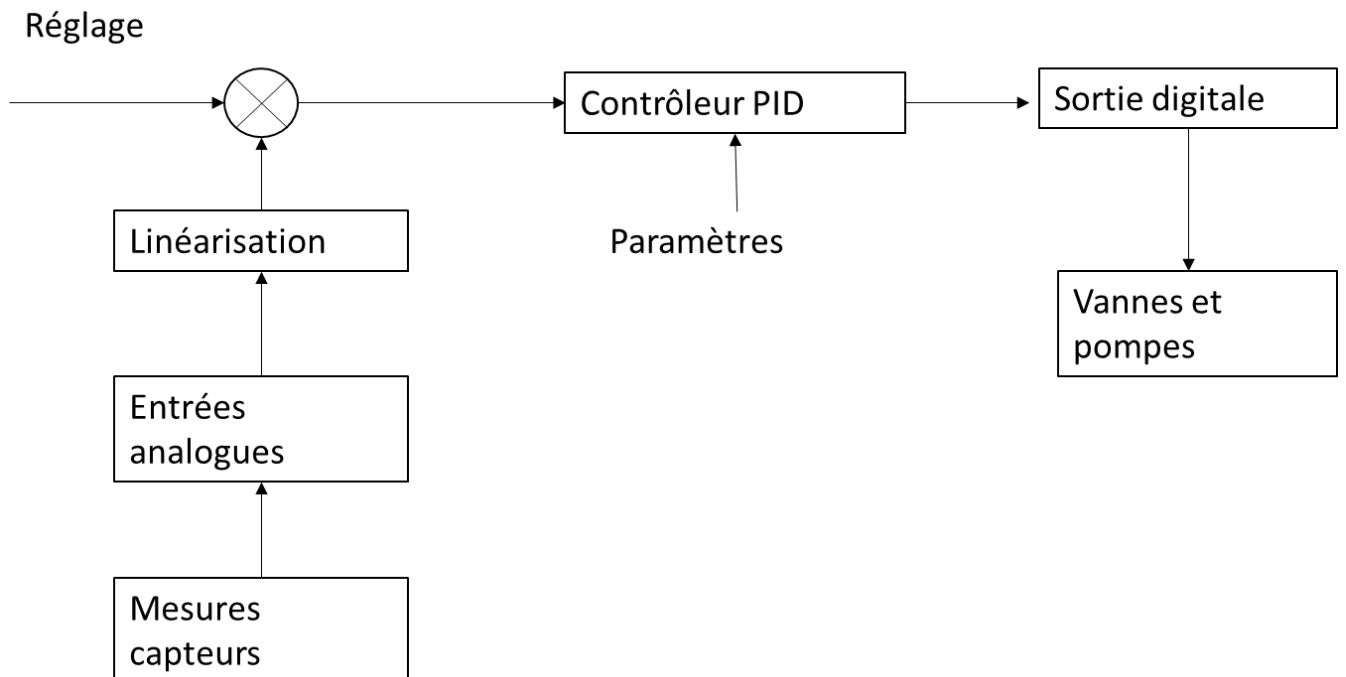


BOUCLES DE REGULATION

20. Actionneurs



21. Pompes et vannes



INTERLOCKS

Quand alarme haute pression cuve :

- Vannes de la vapeur enveloppe non ouvrables.
- Vannes entrée air non ouvrable.
- Vanne de la vapeur de la cuve non ouvrable.
- Ouverture de l'échappement gaz jusqu'à bonne pression.
- Les pompes péristaltiques ne peuvent être démarrées.

Quand alarme niveau haut cuve :

- Vannes de production non ouvrables.
- Pompes ne peuvent être démarrées.

L'agitateur ne peut être démarré que si le circuit de garniture est prêt.

Quand l'alarme niveau bas cuve est déclenchée l'agitateur tourne à la vitesse minimum.

Quand le circulateur de la double enveloppe est à l'arrêt la vanne de vapeur double enveloppe ne peut être ouverte.

Si la vanne de l'échangeur n'est pas ouverte le circulateur de double enveloppe ne peut pas démarrer.

Si la vanne acide n'est pas ouverte la pompe ne peut pas démarrer.

Si la vanne alcali n'est pas ouverte la pompe ne peut pas démarrer.

Si la vanne nutriment n'est pas ouverte, la pompe ne peut pas démarrer.

LISTE DES CONTROLES ET TESTS REALISES EN ATELIER

Revue documentaire des dossiers de construction

- Revue documentaire du dossier de construction de la cuve incluant les contrôles de conformité aux spécifications de conception
- Revue documentaire du dossier de construction de l'environnement incluant les contrôles de conformité aux spécifications de conception
- Revue documentaire du dossier de construction de la console de commande incluant les contrôles de conformité aux spécifications de conception

Tests électriques

- Tests électriques et pneumatiques préliminaires
- Tests électriques et pneumatiques des vannes
- Tests électriques du système d'agitation
- Tests électriques des pompes
- Tests des circuits électriques d'alarmes et de sécurité

Tests d'étanchéité

- Tests d'étanchéité de la cuve
- Tests d'étanchéité des lignes fluides procédés et auxiliaires

Tests d'instrumentation

- Tests des chaînes d'instrumentation (incluant les contrôles de la calibration)

Tests fonctionnels et d'automatisme

- Tests électriques des entrées/sorties
- Tests des réseaux de communication
- Tests dynamiques des logiciels
- Tests fonctionnels

Tests de performances

CONCLUSION

J'ai donc pu pendant ce stage me familiariser avec une installation complexe. La découverte du fonctionnement m'a permis l'aisance de compréhension et de manipulation d'un tel système.

Il s'agissait d'effectuer les tests en usine de l'unité de fermentation. Quelques défauts ont été constatés et corrigés, comme des fuites ou des problèmes de connexion automate.

L'installation pourra donc être livrée au client et la seconde phase de tests commencera dans ses locaux.

Pendant ce stage j'ai appris à lire les plans de l'installation, à repérer les différents capteurs et actionneurs, à mettre en œuvre un processus de contrôle, à tester le matériel, à actionner les différentes parties de l'installation, à observer la conception, à suivre le fonctionnement, etc... Ce fut donc très instructif et concluant du point de vue étude et application dans le domaine des process industriels en général et de la fermentation en particulier. Le contact avec le monde ouvrier a été enrichissant.

SOURCES

Documents Pierre Guérin pour le projet Albit Scientific

Site internet Pierre Guérin

Frédéric GILET

E.N.S.A.M. Angers

*RAPPORT DE STAGE
TECHNICIEN*



1^{er} juillet – 14 août 1997

INTRODUCTION

Etudiant à l'ENSAM d'Angers, j'ai effectué un stage de 7 semaines à la S.E.M.T. Pielstick sur le site de Saint-Nazaire de juillet à mi-août 1997.

Ce stage technicien a pour objectif la réalisation d'un cahier des charges de progiciel de gestion de magasin. Il me permet également de découvrir l'entreprise : son activité, son organisation, les manières de travailler, l'ambiance au travail.

Ce rapport illustre ce que j'ai appris, avec une présentation générale de la société et du site de Saint-Nazaire, une présentation du service logistique et du magasin, division où j'ai fait mon stage. Il rappelle enfin succinctement le travail que j'ai réalisé.

PRESENTATION GENERALE DE S.E.M.T. PIELSTICK

Présentation générale

La S.E.M.T. Pielstick est une Société Anonyme au capital de 51 millions de francs, détenue à parité par les groupes allemands MAN B&W et MTU. Elle réalise l'étude, le développement, la construction, le montage, la commercialisation, le service après-vente de moteurs diesels de grande puissance quatre temps rapides (PA) et semi rapides (PC).

Elle a réalisé en 1996 un chiffre d'affaire de 1263 millions de francs (répartition du CA en annexe).

Elle compte environ 1000 employés répartis dans trois sites : St Nazaire (environ 690 employés), St Denis (environ 250 employés), Jouet sur Aubois (environ 70 employés).

Historique de l'entreprise

En 1946, cinq entreprises fondent, sous l'impulsion de la Marine Nationale Française, la S.E.M.T. (Société d'Etude de Machines Thermiques), pour participer à l'effort de reconstruction de l'après-guerre. Ce sont :

- La Société des Chantiers et Ateliers de Penhoët à St Nazaire
- La Société Générale de Construction Mécanique à La Courneuve
- La Société des Chantiers et Ateliers Augustin Normand au Havre
- La Société des Chantiers et Ateliers de Bretagne à Nantes
- La Société des Aciéries du Nord.

Chacune possède 20% du capital.

Cette société a pour objectif de développer des moteurs diesels rapides et semi rapides adaptés aux navires, aux centrales électriques et aux chemins de fer. N'ayant pas de moyens de production, elle accorde des licences aux usines mères.

Son premier directeur technique est **Gustave PIELSTICK (1890-1961)**. Il fut jusqu'en 1945 directeur technique de MAN-AUGSBURG. Il conçut et développa plusieurs moteurs à haute puissance nominale destinés aux applications marines, particulièrement pour la marine allemande.

Les premiers moteurs PA et PC prototypes sont mis au banc d'essai à la Courneuve en **1951**, à Nantes en **1955**, à St Nazaire en **1956**. Le concept d'une puissance massive supérieure aux techniques existantes (Diesel lent et turbine à vapeur) est très innovant.

La S.E.M.T. commence à conquérir le marché international à partir de **1953**, année marquée par les premières ventes à l'étranger de moteurs construits en France (centrale de Bamako en PC et Chemin de Fer Congo Océan en PA), et par les premiers contrats de licences hors de France.

En **1973**, Alsthom fusionne avec les Chantiers de l'Atlantique. Les établissements mécaniques, fabricant des moteurs diesels, des turbines à vapeur et des chaudières marines, ont alors un effectif de 3400 personnes.

En **1987**, Alsthom cède aux deux groupes allemands, Man B&W et MTU (à parité), la majorité de son activité diesel. La raison sociale de la société anonyme ainsi filialisée est S.E.M.T. Pielstick (voir annexe 1).

Les gammes de moteurs de S.E.M.T. Pielstick

Les moteurs diesel à vitesse rapide PA (1000 à 1500 tr/mn)

Ces moteurs sont destinés aux marchés de la traction ferroviaire, de la propulsion de navires (militaires essentiellement) et aux groupes électrogènes. Ils constituent 23% du chiffre d'affaire de S.E.M.T. La puissance va de 141 kW par cylindre à 295 kW par cylindre. Un moteur PA peut avoir jusqu'à 20 cylindres, soit une puissance de 5.9 MW maximum. (Voir annexe 2)

Les moteurs diesel à vitesse semi rapide (375 à 520 tr/mn)

Ces moteurs sont destinés aux marchés de la propulsion de navires et aux centrales électriques. Ils constituent 72% du chiffre d'affaire. La puissance va de 550 kW par cylindre à 1215 kW par cylindre. Un moteur PC peut avoir jusqu'à 18 cylindres, soit une puissance de 21 MW maximum (voir annexe 3)

Les applications des moteurs diesel

Groupe électrogène et centrales terrestres

C'est aujourd'hui le principal secteur d'activité, avec 40% du chiffre d'affaire en 1992 (voir annexe). S.E.M.T. fournit des centrales électriques fonctionnant avec moteur Diesel et des groupes électrogènes de secours (notamment pour centrales nucléaires).

Le service Engineering de Centrales est chargé d'étudier et de concevoir tout l'environnement du moteur diesel (alternateur, sécurités, etc...) afin de livrer des centrales « clé en main ».

Moteurs pour la propulsion marine

Il s'agit essentiellement de moteurs pour la marine militaire. S.E.M.T. a bénéficié depuis sa création de commandes de la Marine Nationale. Elle a développé et construits des moteurs diesels adaptés à la propulsion et à la production d'énergie sur tous types de bâtiments militaires : frégates, sous-marins diesel, sous-marins nucléaires (groupes électrogènes), pétroliers ravitailleurs, etc...

Les moteurs S.E.M.T. équipent également des navires marchands (ferries, paquebots de croisière). Cette activité est restreinte (aucune commande en 1996, sauf pour les licenciés).

Moteurs pour la traction ferroviaire

La S.E.M.T. a équipé entre 1966 et 1976 tous les moteurs diesel de la SNCF. Avec une trentaine de moteurs par mois, c'était sa principale activité. Cette activité est aujourd'hui très réduite du fait de l'électrification du réseau SNCF.

Les différents sites (voir annexe 4)

Le site de Saint Nazaire

Les différents services

- Bureaux d'étude, d'industrialisation et de personnalisation des moteurs selon leurs applications.
- Le service contrôle-commande
- Les services commerciaux
- Les services calculs, mesures et essais
- Le service assurance qualité
- Le service achats
- Le service logistique, duquel dépend le magasin de stockage des composants de moteurs

Les différents bâtiments

- Ateliers de fabrication (essentiellement usinage de composants et fabrication de la tuyauterie)
- Les ateliers de montage (et d'essais) des moteurs
- Le magasin de stockage de composants de moteurs

Le site de Saint Denis

C'est le siège social du groupe. Il est constitué :

- De la direction générale
- Du département licence, qui négocie et gère les contrats des licenciés
- Du centre d'étude de machine thermique
- De l'établissement centrales électriques diesel qui étudie les centrales équipées

Le site de Jouet-sur-l'Aubois

Ce site est spécialisé dans la fabrication de systèmes d'injection.

L'activité actuelle

43% du chiffre d'affaire 1992 est réalisé grâce à l'activité rechange et services (voir annexe 5)

La S.E.M.T. apporte des services d'assistance performants comportant :

- L'assistance technique, assurée par le SAV et le réseau mondial de réparateurs agréés
- L'expédition au client de toute pièce de rechange nécessaire
- La formation des opérateurs qui assureront l'entretien et la conduite des installations (école Diesel située à Saint-Nazaire)
- La réhabilitation des moteurs

Afin d'améliorer ce service, un logiciel destiné au suivi des moteurs S.E.M.T. Pielstick a été récemment conçu. C'est le système INSPECT (Intelligent System for Pielstick Engine Control). Mesurant les paramètres de fonctionnement, il signale par satellite toute anomalie et permet d'intervenir à temps.

40% du chiffre d'affaire 1992 est réalisé par l'activité centrales électriques (annexe 5).

9% du CA est réalisé par l'activité licenciés. La S.E.M.T. gère des contrats de licence pour 26 licenciés dans le monde qui construisent des moteurs Pielstick et paient des royalties (voir annexe 6).

En 1996, pour un CA hors taxes de 1263 millions de francs (en retrait de 6% par rapport à 1995), le bénéfice après impôts a été de 50.3 millions de francs.

La part d'exportations est importante (65% du CA) et devrait continuer de croître.

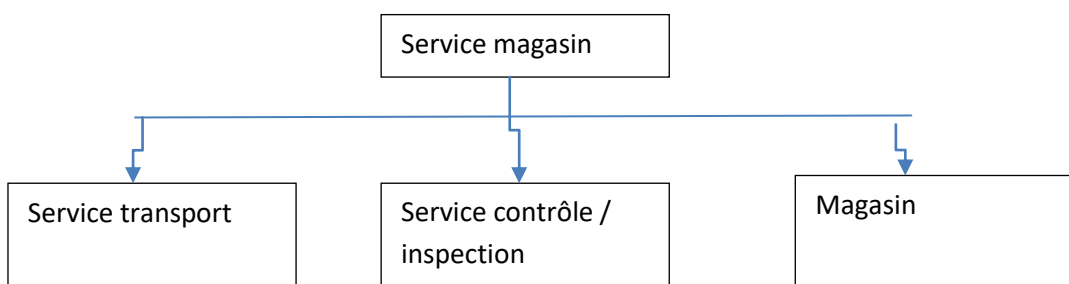
Les résultats restent favorables dans un contexte où le marché militaire est atone et où les intentions d'investissement en centrales électriques sont longues à se concrétiser.

PRESENTATION DU SERVICE DU MAGASIN

Les grandes activités du service

J'ai été accueilli durant mon stage au département logistique, et plus particulièrement au magasin.

Organigramme du service magasin



Le service transport

Il a pour fonctions :

- De trouver des solutions de transport et de les optimiser en respectant les délais pour certaines expéditions, principalement à l'étranger.
- De régler les litiges sur les transports.

Il travaille en collaboration avec plusieurs services et plusieurs personnes :

- Pielstick Service, service des pièces de rechange, pour expédier des pièces de rechange dans le monde entier.
- Le service des Devis, pour réaliser des devis transport pour des projets d'expédition et d'installation de nouveaux moteurs.
- Les ingénieurs projets, pour réaliser des transports ou des devis pour des projets en cours
- Le magasin.

Le service contrôle / inspection

Il a plusieurs missions :

- Inspection : aller chez les fournisseurs contrôler la démarche qualité et leur aptitude à fournir des articles de qualité à S.E.M.T.
- Le contrôle : contrôler les pièces à leur réception en magasin
- Métrologie : étalonnage du matériel de contrôle de la société

Le magasin

Il a pour mission de :

- Réceptionner et entrer en stock les articles commandés à un fournisseur, fabriqués par l'atelier ou retournés par le client
- Sortir du stock, préparer et expédier les articles commandés par un client, ou sortir des articles demandés par l'atelier
- Assurer la gestion de stock

Il travaille en étroite collaboration avec plusieurs services :

- Le service achats, qui lui communique les commandes fournisseurs attendues et auquel il communique les commandes réceptionnées
- L'atelier, qui lui fournit des articles à entrer en stock et auquel il fournit les articles dont il a besoin
- Le service comptabilité, auquel il communique les articles réceptionnés, les quantités en stocks (pour chaque magasin) et les articles expédiés. La comptabilité peut ainsi payer les factures des fournisseurs, envoyer les factures aux clients et connaître la valeur du stock.
- Pielstick Service, qui lui demande de sortir de stock et de préparer des articles pour les expédier aux clients.
- Le service transport.

Les réunions de service

Le service organise chaque semaine une réunion à laquelle participent les différents responsables dans le service, pour définir l'activité logistique à très court terme et discuter des problèmes.

Des réunions informelles sont fréquemment organisées afin de discuter ou de résoudre des problèmes concrets, ou afin d'informer.

Un responsable du service logistique participe aux réunions mensuelles du PDP (Plan Directeur de Production, et aux réunions semestrielles du PIC (Programme Industriel et Commercial), organisé par la direction.

Le PDP a pour objectif de mettre en œuvre un système de gestion des prévisions de production à court terme permettant d'apprécier mensuellement les charges, les délais et les capacités des différents secteurs de l'entreprise. Il permet de simuler la planification générale (plan d'approvisionnement, plan d'activité des services techniques, plan de fabrication), à partir des prévisions et des rapports d'avancement des projets en cours.

Le responsable logistique y participe car il est directement concerné par le plan d'approvisionnement.

Le PIC a pour objectif d'orienter l'allocation des ressources clés de l'entreprise : main-d'œuvre, capacité machine, approvisionnements longs (stocks stratégiques), en fonction d'une analyse des projets en cours et des prévisions commerciales par activité (neuvage, rechange, licenciés, partenaires).

Les réunions permettent de discuter et d'en informer chaque secteur concerné.

Le responsable logistique y participe car il est directement concerné par les approvisionnements longs

Fonctionnement du magasin

Durant mon stage, j'ai surtout été amené à étudier le fonctionnement du magasin.

La réception

Objectifs : assurer la réception et l'entrée en stock d'articles commandés à un fournisseur, fabriqués par les ateliers S.E.M.T. Pielstick ou retourné par le client.

Voir organigramme de fonctionnement en annexe 8

La sortie d'articles et la préparation de la commande

Objectif : sortir de stock les articles demandés par un client, l'atelier ou d'autres services (consommables). Assurer la préparation et l'expédition des commandes destinées aux clients.

Voir organigramme de fonctionnement en annexe 7

La gestion des emplacements

La force du magasin est son palettier automatisé qui emmène ou ramène, selon la commande informatique, les articles stockés sur des racks en hauteur. Le reste du stock est sur étagères et tiroirs accessibles en chariots élévateurs ou à la main pour de petits objets.

La procédure d'inventaire est performante informatiquement. A chaque fois qu'on prend un article sur la palette, l'ensemble de la palette est recomptée par l'opérateur. Si les quantités sont cohérentes, lors de l'inventaire, elles ne seront pas recomptées.

Les déménagements physiques d'un lot d'articles d'un emplacement à un autre est suivi d'une transaction informatique de déménagement administrative.

Observations sur le fonctionnement du magasin

Répétitivité des tâches

Les tâches effectuées au magasin sont relativement répétitives, à quelques exceptions près (litiges, articles particuliers). En voici quelques exemples :

- A la réception physique des colis, débarquer, compter succinctement les articles et les localiser (diversité des colis, déplacements à effectuer rendent le travail plus varié)
- A la réception administrative, édition du journal des entrées, d'une fiche de suivi réception (diversité : litiges et cas particuliers)
- A la mise en stock, comptage et placement par chariot élévateur ou manuellement dans le casier par l'opérateur. Le palettier amène automatiquement le casier à son emplacement : c'est la tâche la plus répétitive.

Distribution des tâches

Chaque entité du magasin a un nombre constant d'employés (22 au total).

Les tâches à accomplir sont rarement assignées par un supérieur. Elles dépendent des commandes fournisseurs en réception, des demandes du service rechange ou de l'atelier en sortie de stock.

La plupart du temps, le partage des tâches se fait naturellement en fonction du domaine de compétence et de l'initiative de l'opérateur.

Il existe pour la plupart des tâches que le magasin doit réaliser des procédures précises que l'opérateur doit respecter. Leur apprentissage se fait par information verbale. Ce n'est pas un inconvénient, au vu des nombreux stagiaires, intérimaires et employés d'été au magasin.

La stratégie de l'entreprise et du magasin

Amélioration de la productivité

C'est une première caractéristique de la stratégie de l'entreprise, afin d'avoir des prix de vente et des délais conformes aux normes du marché des moteurs diesel.

Elle est notamment discutée lors de réunions de groupes, et fait l'objet d'informations sous forme d'affichages.

Le service logistique doit participer à cet effort :

- Diminuer les délais (passer de 10 à 7 jours pour la mise en stock)
- Réduire les coûts (objectif du magasin : -35%)

Il s'agit d'analyser quels secteurs représentent une part importante du chiffre d'affaire, pour y affecter prioritairement les investissements. Il s'agit dans un deuxième temps de recueillir les propositions faites pour apporter des améliorations au système.

Dans cet effort de restructuration, des modifications ont déjà été effectuées depuis 2 ans au magasin :

- Construction d'un local contrôle réception plus fonctionnel, mieux placé
- Construction d'un nouveau palettier appelé silo pour les articles de faible dimension (gain de temps de transport des articles, gestion facilitée par l'informatique et l'utilisation de codes barre)
- Restructuration de certaines surfaces de stockage
- Construction d'un quai réception poids lourds
- Achat d'un chariot élévateur latéral plus performant que l'ancien
- Achat d'un pont de levage de 12 tonnes et dégagement d'une aire de stockage au magasin pour des articles lourds (bâti, arbre manivelle, etc...)

D'autres modifications sont en prévision :

- Achat d'un nouveau progiciel de gestion d'entrepôt permettant entre autres l'utilisation de codes barre
- Optimisation des flux en réorganisant certaines surfaces de travail et de stockage

Augmentation du service

C'est une orientation stratégique essentielle pour la S.E.M.T. : le marché du service après-vente et des pièces de rechange est plus dynamique que le marché des moteurs neufs, et sa part dans le CA devrait augmenter. L'entreprise améliore donc ses services, notamment avec le système INSPECT.

Le magasin est concerné par cette tendance, puisqu'il stock et fournit les pièces de rechange. Il doit donc faire face et s'adapter chaque année à une augmentation du nombre d'articles traités, en respectant les délais, grâce aux efforts de productivité et de rationalisation.

MON STAGE : LA REALISATION DU CAHIER DES CHARGES DU PROGICIEL DE GESTION D'ENTREPOT

Le logiciel actuellement en place est relativement fonctionnel. Cependant, un changement prochain du logiciel en réseau de l'entreprise rend nécessaire l'achat d'un logiciel métier pour le magasin.

Ma mission a été d'observer dans un premier temps le fonctionnement du magasin, et de rédiger une analyse fonctionnelle du logiciel existant. J'ai pu observer dans le détail tous les domaines du magasin, et voir d'éventuelles améliorations à apporter à sa gestion :

- Utilisation généralisée de codes barre
- Communication optimisée entre les logiciels des palettiers et le progiciel
- Modification des supports papiers pour les rendre plus fonctionnels et supprimer les feuilles inutiles
- La gestion administrative dans le magasin doit être réalisée dans un lieu proche de la gestion physique à laquelle elle se rapporte.

Dans un deuxième temps, j'ai rédigé l'expression du besoin. J'ai défini les caractéristiques générales, les capacités de traitement du progiciel, listé les fonctionnalités nouvelles intéressantes. J'ai fait l'analyse fonctionnelle par la méthode SADT

Ce travail servira définir la liste de besoins fonctionnels lors de l'étude préalable au logiciel, vérifiera l'adéquation entre les besoins fonctionnels existants et prévisionnels et les fonctionnalités standards du progiciel existant. Cela permettra le maquettage des développements spécifique à la S.E.M.T.

CONCLUSION

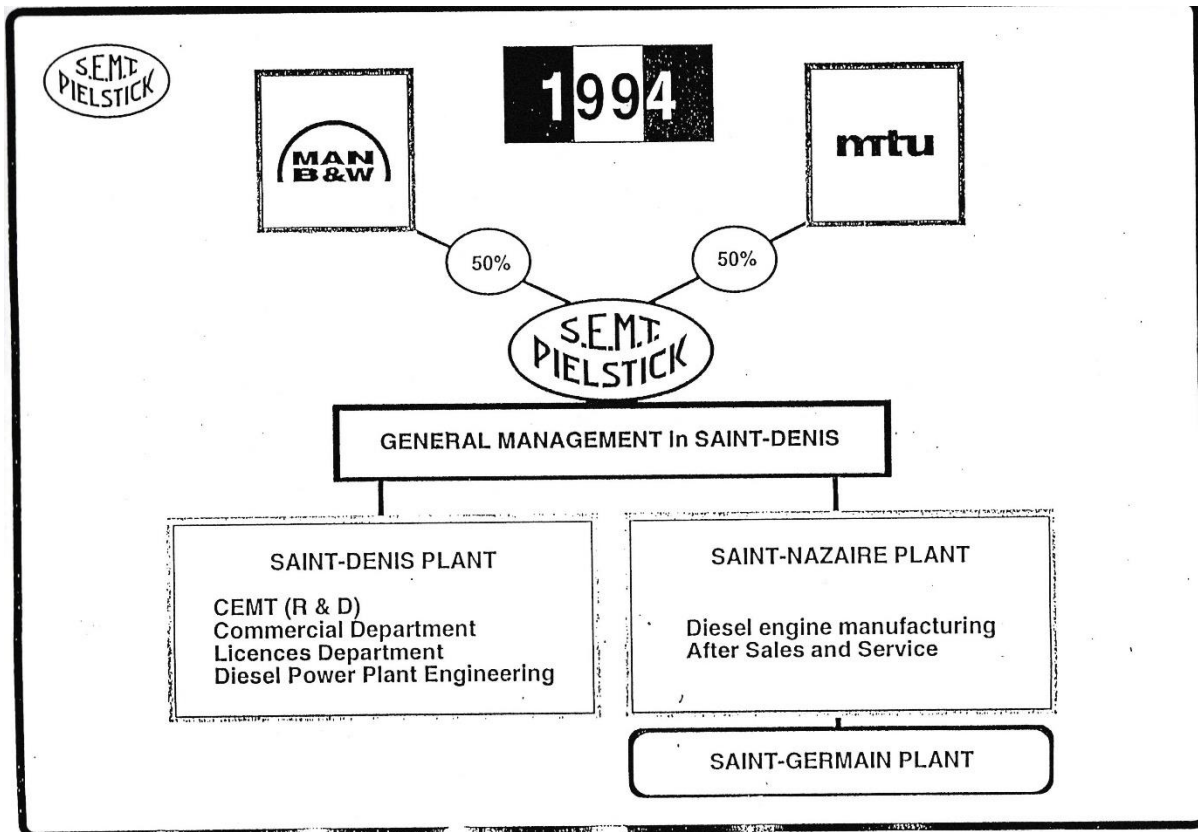
Ce stage a été très enrichissant.

D'abord le travail que j'ai effectué m'a permis de mieux appréhender la conduite de projets, les difficultés qui en découlent, ne rien oublier des paramètres importants.

De plus, il m'a permis d'approfondir ma connaissance du monde du travail.

ANNEXES

Annexe 1 : Répartition du capital de S.E.M.T. Pielstick



Annexe 2 : Coupe transversale d'un moteur diesel à vitesse rapide

PA6 B

COUPE TRANSVERSALE
CROSS SECTION

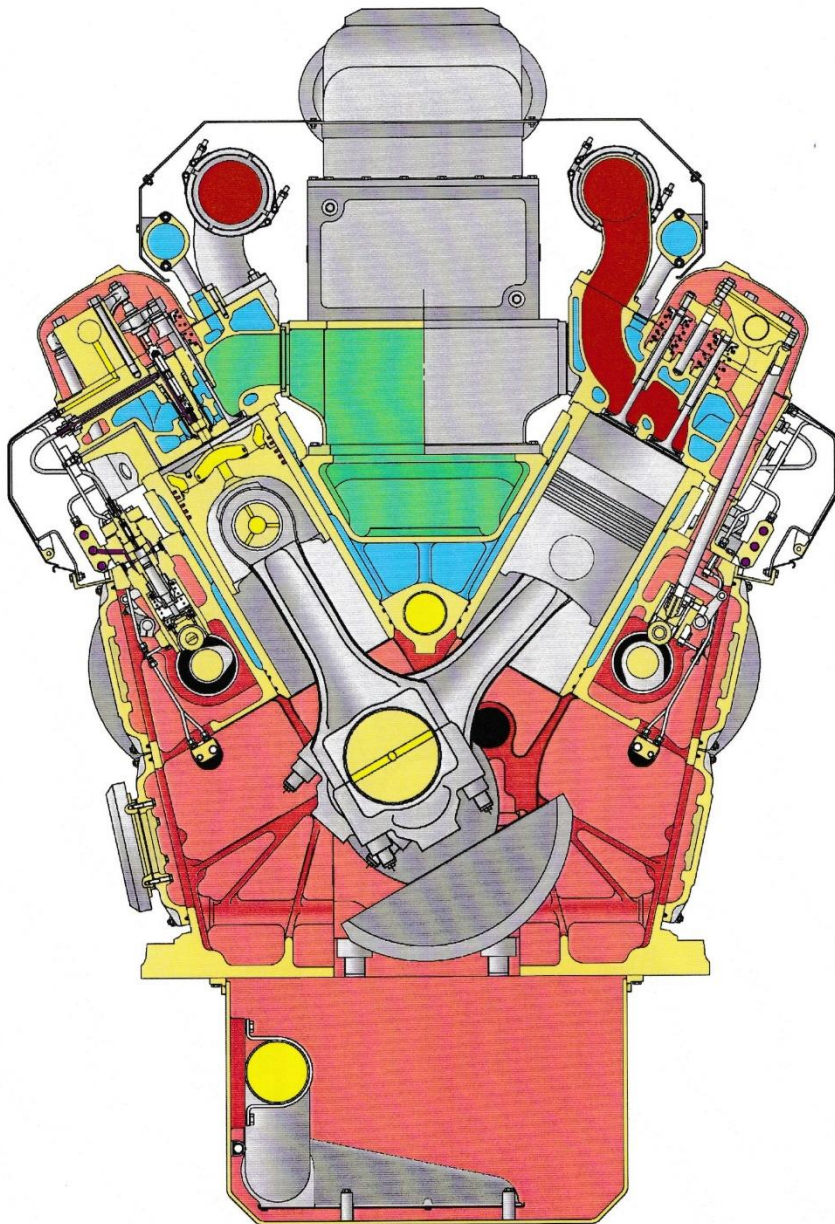


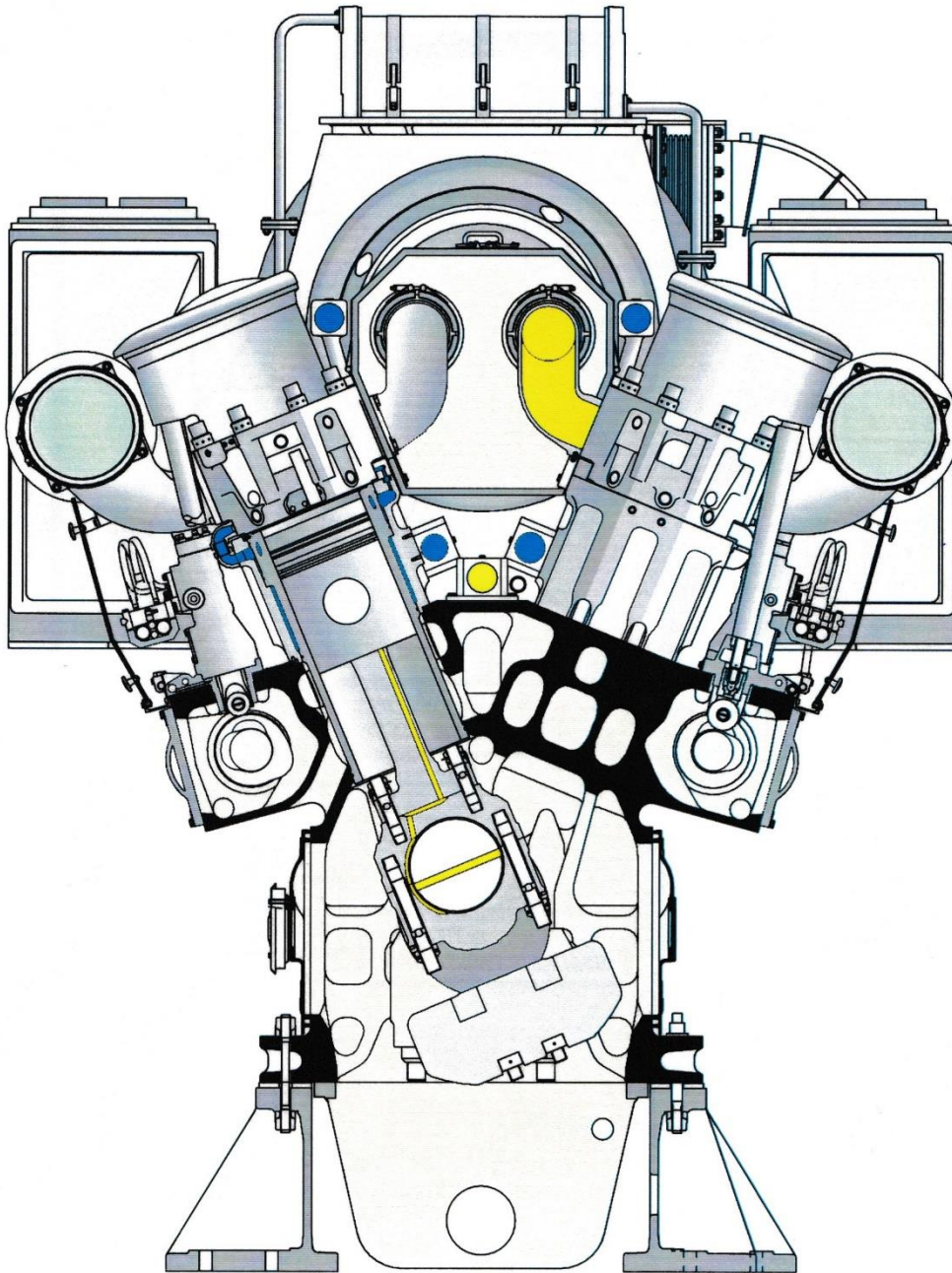
FIGURE 1



Coupe transversale d'un moteur diesel à vitesse semi rapide

PC 2-6 B

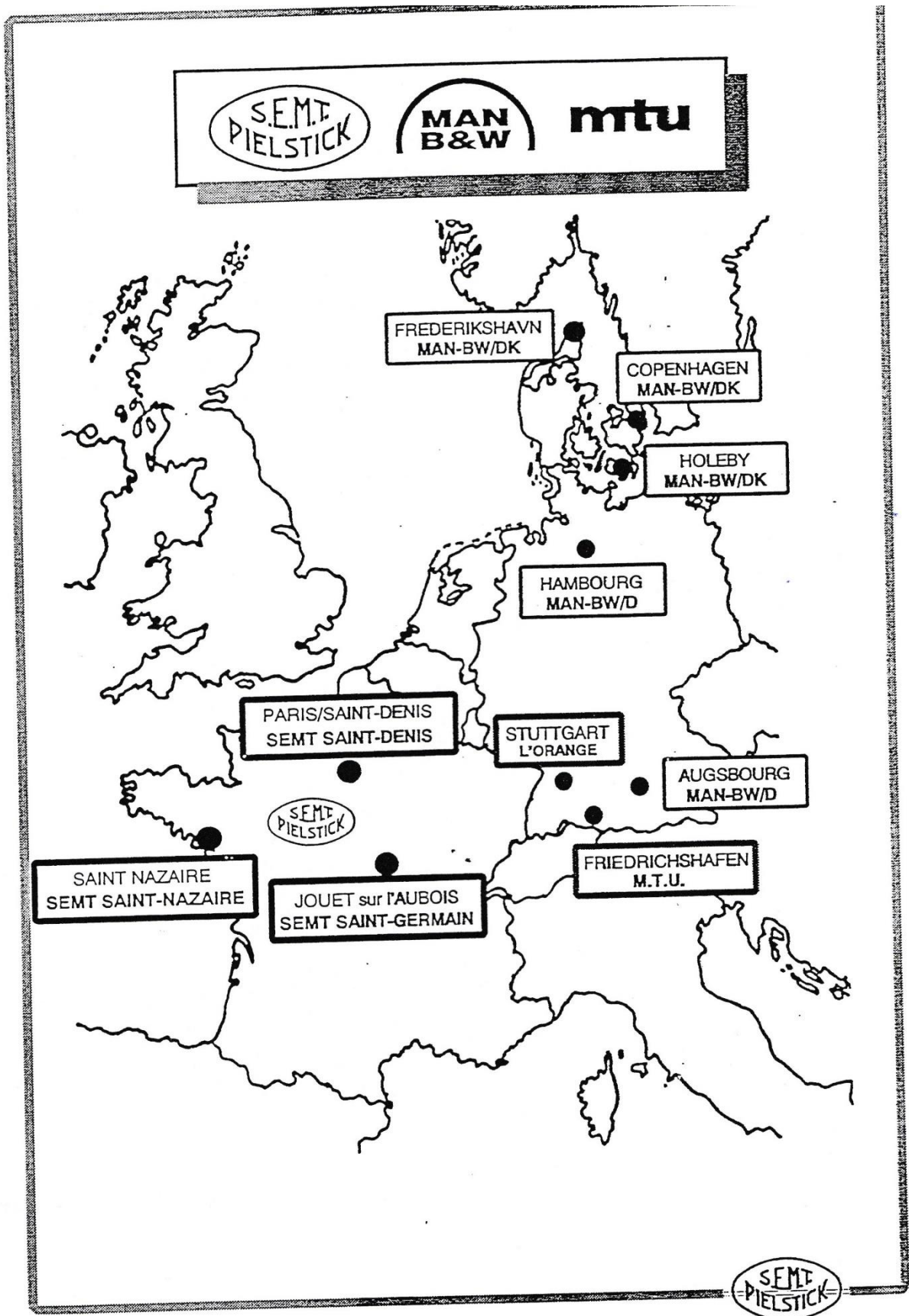
600 tr/min



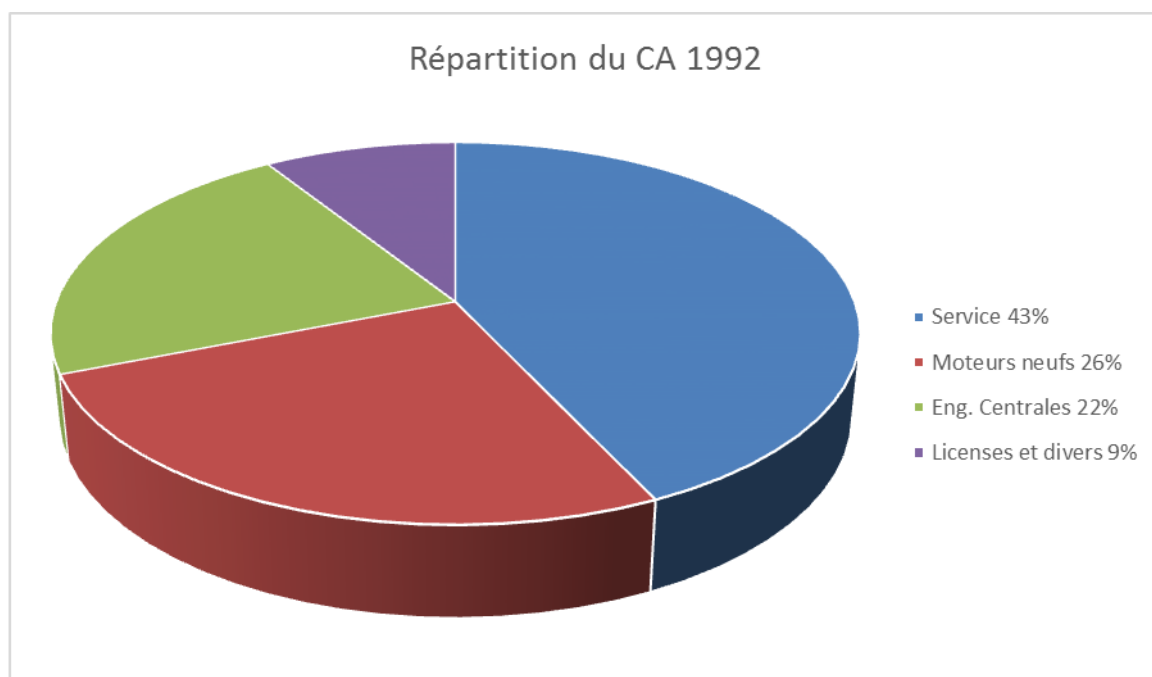
COUPE TRANSVERSALE
CROSS SECTION



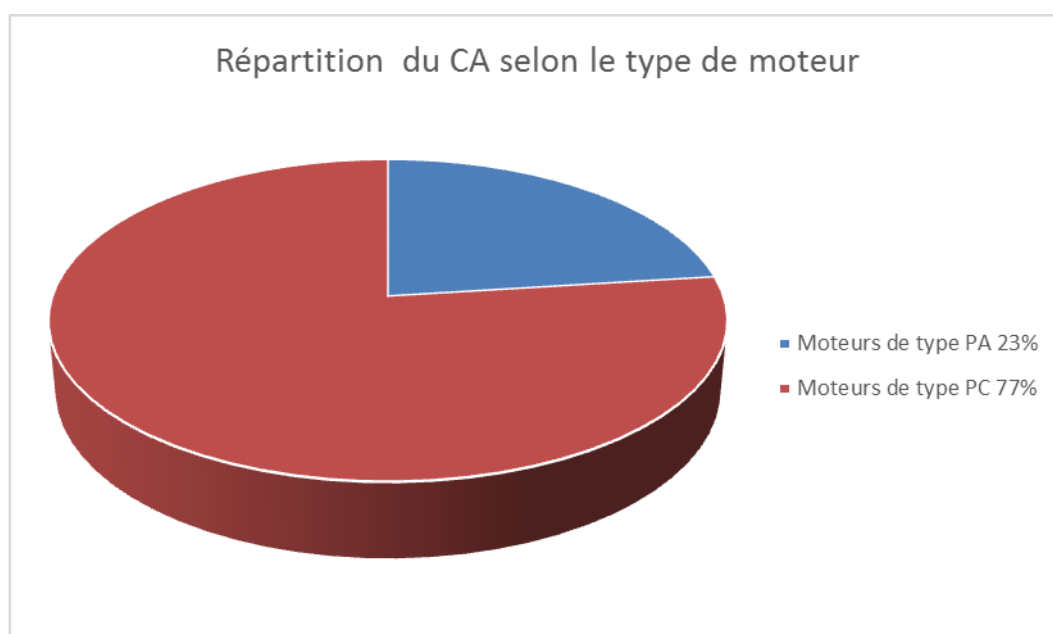
Annexe 4 : Les différents sites SEMT Pielstick et MAN-MTU en Europe



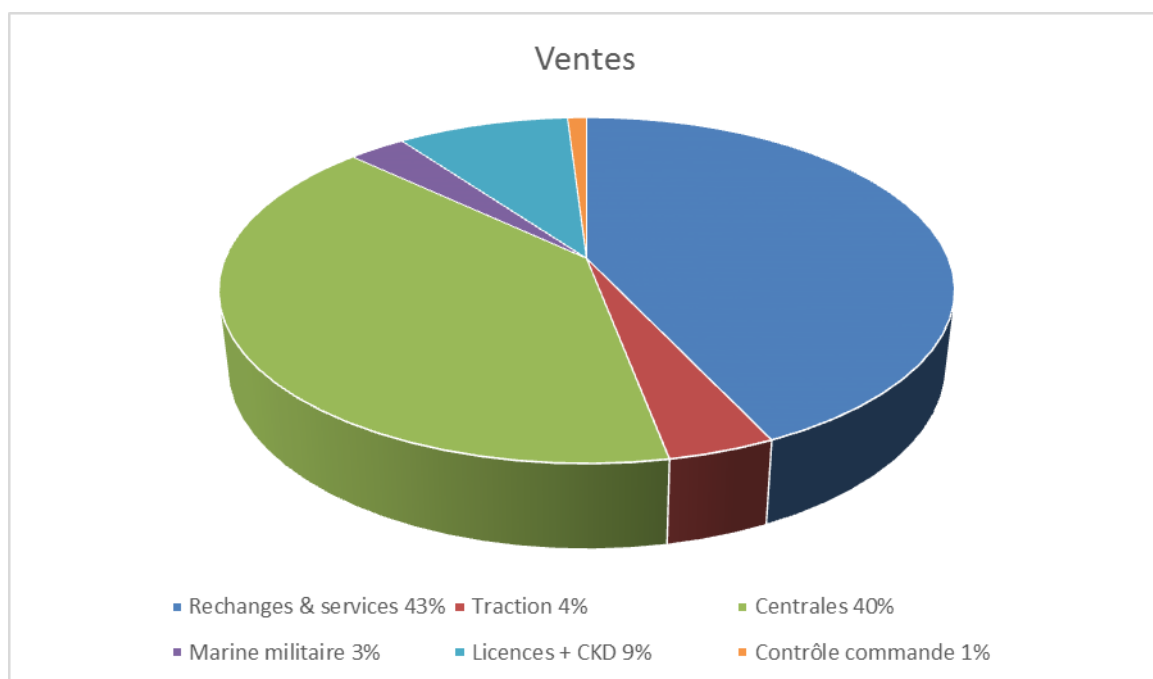
Annexe 5A : Répartition du CA selon les services



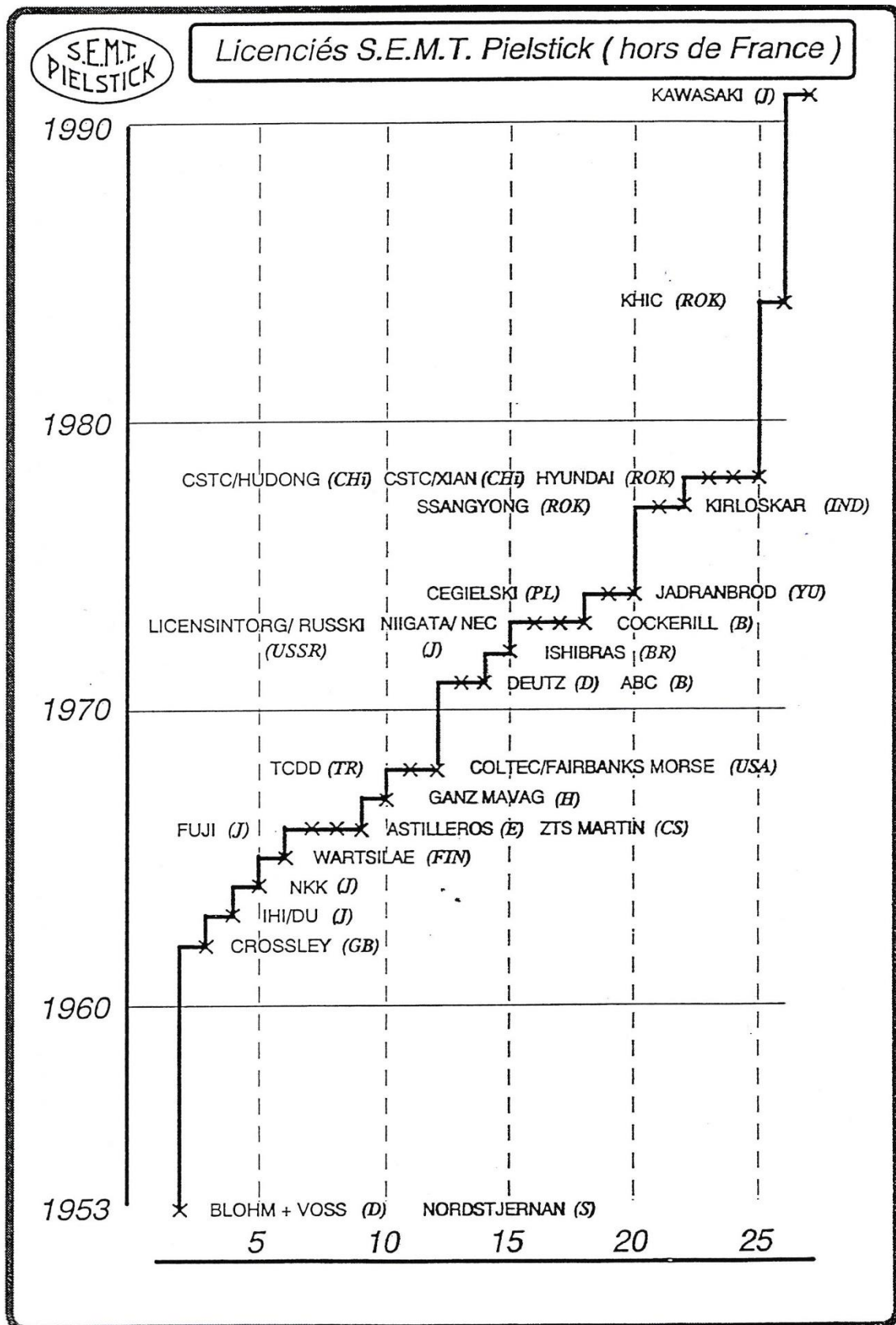
Annexe 5B : Répartition du CA selon le type de moteurs



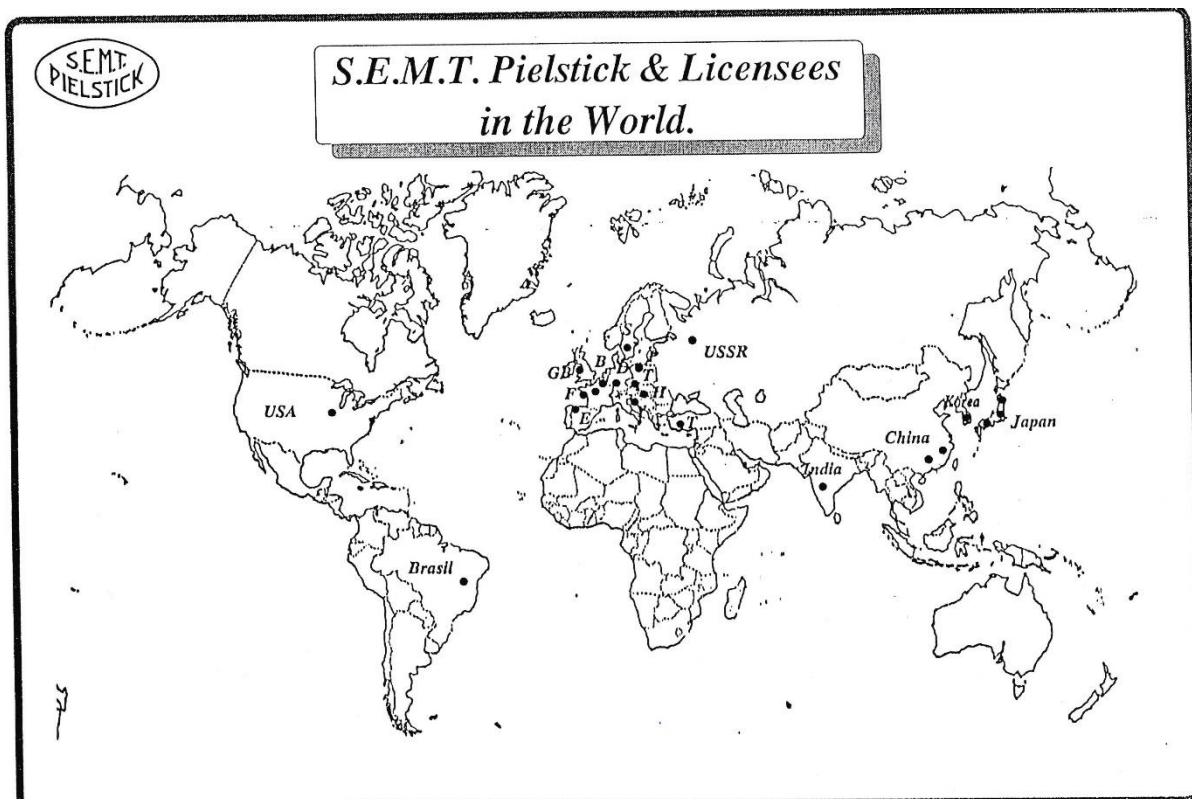
Annexe 5C : répartition du CA en fonction du type de clientèle



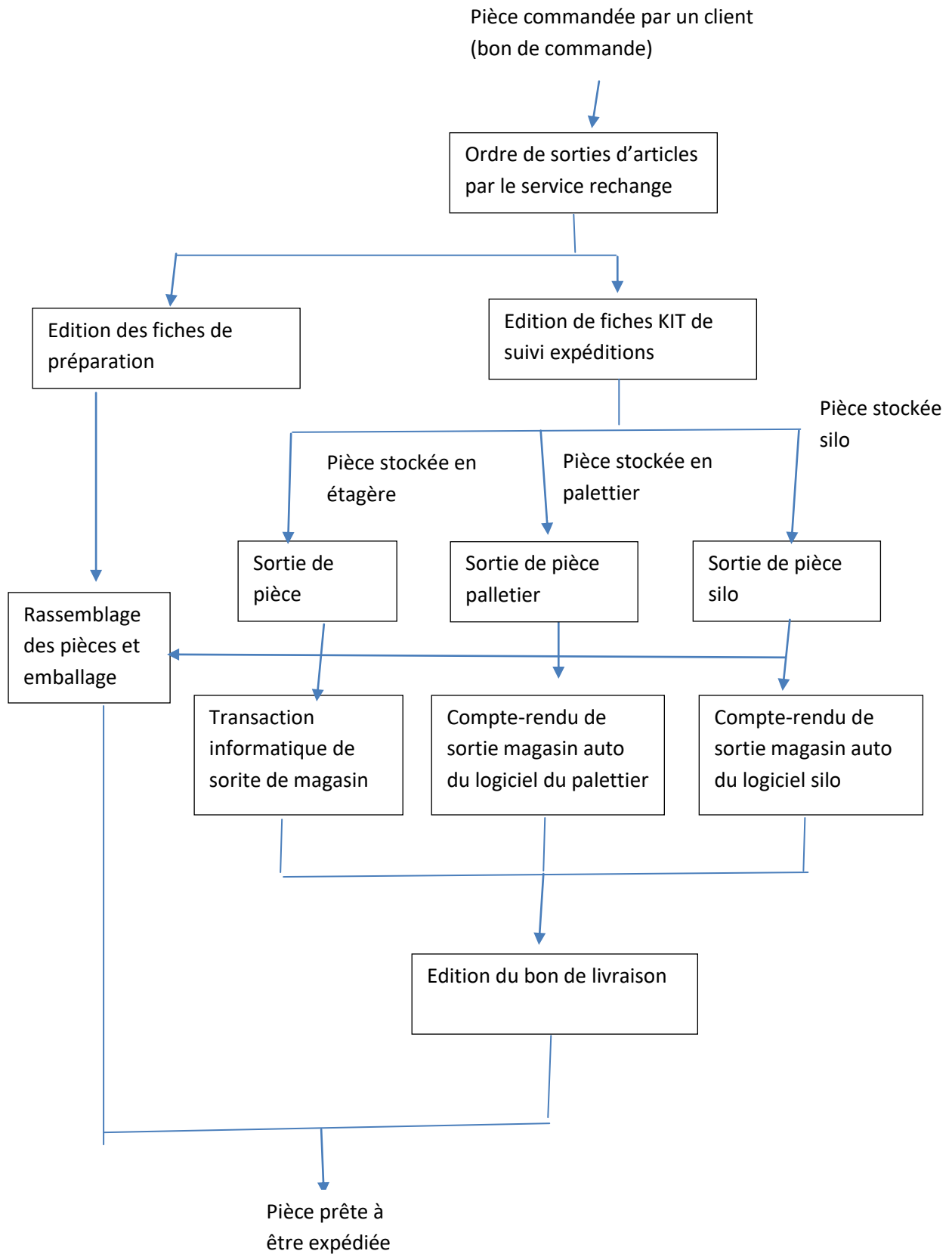
Annexe 6A : les licenciés



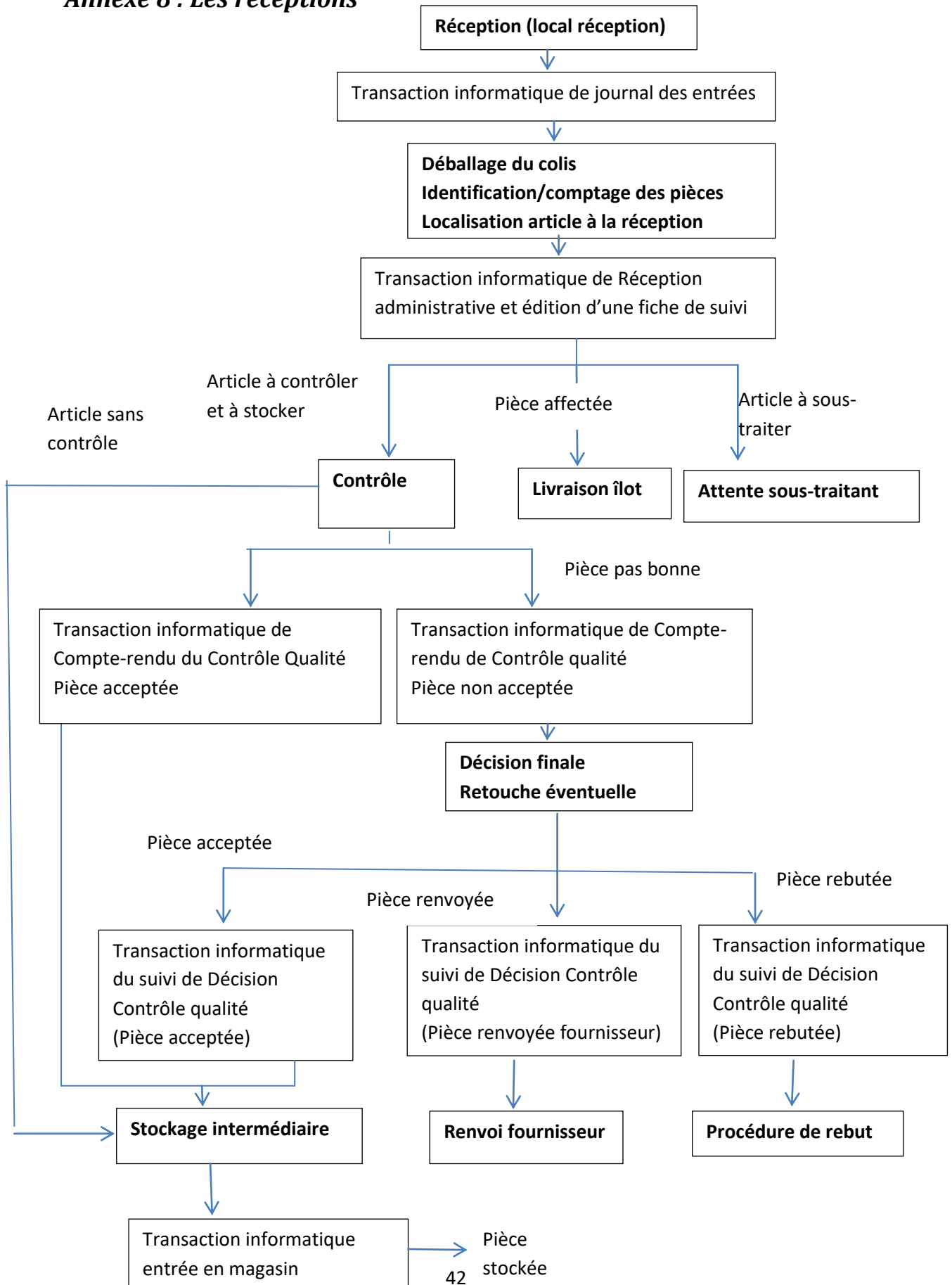
Annexe 6B : répartition des licenciés



Annexe 7 : les expéditions

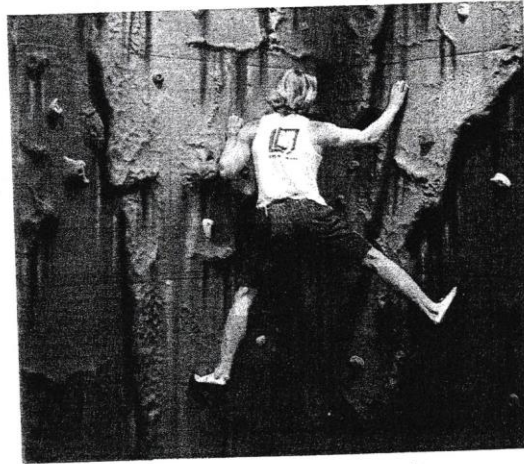


Annexe 8 : Les réceptions



CLIMBING WALL SYSTEM PROJECT

Frederic Gilet



Supervisor : Derek Seward

Company : DR climbing Wall

25th october 1998

CONTENTS

SUMMARY	6
INTRODUCTION.....	7
CHAPTER 1 : INTRODUCTION TO THE PROBLEM	
1) ROTATE PANELS.....	9
2) SHOW ROUTE.....	11
3) ROTATE HOLDS.....	12
4) AIMS OF THE PROJECT	12
CHAPTER 2 : REQUIREMENT ANALYSIS	
1) STATEMENT OF REQUIREMENT	14
A) STATEMENT OF REQUIREMENT FOR THE CONFIGURABLE WALL CONCEPT	14
a) <i>Operational requirement</i>	14
b) <i>Functional requirement</i>	14
c) <i>Non-functional requirement</i>	14
B) BUBBLE DIAGRAM FOR THE CONFIGURABLE WALL CONCEPT	15
C) STRUCTURE CHART FOR THE CONFIGURABLE WALL CONCEPT.....	16
D) DATA FLOW DIAGRAMS FOR THE CONFIGURABLE WALL CONCEPT	16
a) <i>Context diagram : figure 7a</i>	17
b) <i>Diagram 0 : figure 7b</i>	18
c) <i>Diagram 3 : Use climbing wall : figure 7c</i>	19
d) <i>Diagram 3.1 : Change wall characteristics : figure 7d</i>	20
e) <i>Diagram 3.2 : Select route : figure 7e</i>	21
f) <i>Diagram 3.3 : select wall configuration : figure 7f</i>	22
g) <i>Diagram 3.4 : Change route. Figure 7g</i>	23
h) <i>Diagram 3.5 : Change configuration : figure 7h</i>	24
E) PROCESS SPECIFICATION AND PROCESS DESCRIPTION OF THE DATA FLOW DIAGRAM.....	25
a) <i>Diagram 3.1 : Change wall characteristics</i>	25
b) <i>Diagram 3.2 : Select route</i>	25
c) <i>Diagram 3.3 : Select wall configuration</i>	25
d) <i>Diagram 3.4 : Change route</i>	25
e) <i>Diagram 3.5 : Change configuration</i>	25
f) <i>Diagram 3.1.1 : Build wall</i>	25
i) <i>Diagram 3.1.4 : Build socket</i>	27
j) <i>Diagram 3.1.5 : put a hold</i>	27
j) <i>Diagram 3.2.1 : Initialise data</i>	27
k) <i>Diagram 3.2.2 : Build wall</i>	28

k)Diagram 3.2.3 :

Select a hold	29
l) Diagram 3.2.4 : Select a color.....	29
m) Diagram 3.2.5 : save route.....	29
n) Diagram 3.3.1 : Initialise data	30
k) Diagram 3.3.2 : Build wall.....	30
o) Diagram 3.3.3 : Select a side.....	31
p) Diagram 3.3.4 : Rotate a side.....	31
q) Diagram 3.3.5 : Rotate hold.....	31
r) Diagram 3.3.6 : save configuration.....	31
s) Diagram 3.4.1 : Build code	32
t) Diagram 3.4.2 : transmit code.....	32
u) Diagram 3.5.1 : Build code	32
v) Diagram 3.5.2 : transmit code.....	33
F) DATA DICTIONNARY	34
a) Context Diagram.....	34
b) Diagram 0.....	34
c) Diagram 3.....	34
d) Diagram 3.1 : change wall characteristics.....	35
d) diagram 3.2 : select route	35
e) Diagram 3.3 : select wall configuration.....	36
f) diagram 3.4 : change route.....	37
g) diagram 3.5 : change configuration	37
G) STATEMENT OF REQUIREMENT FOR THE SHOW ROUTE CONCEPT	38
a) Operational requirement.....	38
b) Functional requirement.....	38
c) Non-functional requirement	38
2) USER INTERFACE	38

CHAPTER 3 : TECHNICAL OPTIONS

1) FUNCTION-MEANS TABLE.....	40
2) TECHNICAL OPTIONS SELECTION.....	40
A) INDICATE ROUTE.....	40
B) LOW LEVEL CONTROL UNIT AND BUS	40
a) I2C bus, serial/I2C interface, LED driver with I2C interface	40
b) I2C bus, I2C interface, PLD+latches	41
c) Serial bus, PLD, latches	42
d) Conclusion.....	43
C) BUS.....	43
a) The I2C bus and how to use it.....	43
b) Conclusion.....	45
D) LOW LEVEL CONTROL SOFTWARE	45
E) HIGH LEVEL CONTROL UNIT	45
F) HIGH LEVEL CONTROL SOFTWARE	45
G) POWER SUPPLY	45

3) TECHNICAL PROPOSITION AND SCHEMATIC DESCRIPTION 45

CHAPTER 4 : IMPLEMENTATION

1) THE CONTROLLER BOARD 49

A) THE MICROCONTROLLER	49
B) THE BOARD	49
a) <i>Functional overview of the system</i>	49
b) <i>Memory organisation</i>	50
c) <i>Serial communication</i>	51
d) <i>I2C bus</i>	52
e) <i>60-way expansion connector</i>	52
C) DESCRIPTION OF FUNCTION	52
D) BOARD LAYOUT	54
E) TEST MODULE	55
a) <i>Testing recommendations</i>	55
b) <i>Testing the board</i>	55

2) THE LED BOARD 56

A) FUNCTIONAL OVERVIEW OF THE BOARD	56
B) EXPLANATION OF THE FUNCTIONNING	57
a) <i>The expansion connector</i>	57
b) <i>Programmable logic device</i>	58
c) <i>Latch</i>	58
d) <i>UDN</i>	58
e) <i>Power supply</i>	59
B) DESCRIPTION OF THE FUNCTIONING	59
C) BOARD LAYOUT	59

3) THE RIG 60

CHAPTER 5 : SOFTWARE IMPLEMENTATION

1) SOFTWARE FOR THE GANGLION BOARDS..... 62

A) SCHEMATIC DIAGRAM FOR THE PROCEDURE BOARD GATEWAY	62
B) GET DATA FROM THE SERIAL PORT	64
C) GET DATA FROM THE I2C BUS	64
D) GET DATA PROCEDURE	65
E) PUT DATA ON THE I2C BUS	66
F) PUT DATA	67
G) SCHEMATIC DIAGRAM FOR THE BOARD MOVETO.	68
H) PUT DATA TO THE EXPANSION CONNECTOR	69

2) USER INTERFACE SOFTWARE 70

A) PROCEDURE BUILD WALL CHARACTERISTICS	72
B) CREATE PANELS ON THE WALL	73
C) CREATE A LED SOCKET	74
D) ROTATE PANEL	75

E) SAVE WALL CHARACTERISTICS	76
F) BUILD A CONFIGURATION.....	77
G) DRAW PANELS	78
H) CHANGE PANEL SIDE.....	79
I) SAVE CONFIGURATION	80
J) BUILD A ROUTE.....	81
K) SAVE ROUTE	82

CHAPTER 6 : RECOMMANDATIONS AND CONCLUSION

1) USER INSTRUCTIONS	84
A) PLUG IN THE ELECTRONICS	84
B) CONNECT THE HYPERTERMINAL	84
C) USE THE VISUAL BASIC PROGRAM.....	85
2) COST.....	85
A) COST FOR THE LED BOARD.....	85
B) PRICE FOR THE GANGLION BOARD.....	86
C) PRICE FOR THE SYSTEM.....	86
3) LEDS MOUNT.....	87
A) FIRST SOLUTION.....	87
B) SECOND SOLUTION.....	87
C) THIRD SOLUTION.....	88
D) FOURTH SOLUTION.....	88
E) 4) RECOMMANDATIONS FOR FUTURE WORK.....	89
A) ELECTRONICS	89
B) SOFTWARE.....	91
C) GENERAL RECOMMANDATIONS.....	91
CONCLUSION.....	92
THANKS.....	92
BIBLIOGRAPHY	93
FIGURES.....	93

SUMMARY

The company wants to develop a prototype climbing wall where the user can rotate panels, show a route with LEDs and in the future rotate a hold. The company developed, by itself, the rotating panels. The panels have three sides. The prototype is a six panel wall where the user turns panels, by clicking an icon on the computer.

The goal of the climbing wall system project was to propose a mechatronic solution for the configurable wall concept : show a route and rotate a hold. This goal has been partly reached : a prototype has been built. It drives LEDs for one panel from a computer.

The electronic solution was partly re-used from the M.Eng WARP project, using a ganglion board with a 80C552 microcontroller which calculates the algorithm to control the LEDs board. It uses a serial bus and should use an I2C bus. The LEDs board, which controls the LEDs, was designed by myself.

The software in Visual Basic to control the system from a computer is partly finished. It controls the LEDs route for one panel thanks to a user-friendly user interface.

The work which remains is the following :

- build the five boards and make it work in order to drive the LEDs of six panels with the boards and an I2C bus.
- Continue to develop the software to make it work with the six panels.
- Integrate the program into the main program developed by the company.
- Continue to develop some algorithms of the program.
- Finally, when it is done, the main target will be to make the rotate hold concept work.

INTRODUCTION

DR international Climbing Wall is an important builder of traditional climbing walls. This company designs and builds walls and holds for climbers. It is a company which is at the forefront of the development. So it wants to integrate electronics and computer sciences in its traditionally mechanic systems, to improve the quality of the service offered.

It studied a concept where the user would just have to click on a button on a user-friendly interface of a computer to change the wall configuration. The configurable climbing wall concept, where the user could rotate panels on the wall, show a route with LEDs and rotate holds, from a computer, was born.

The company developed by itself the rotate panels part : a prototype is now built where the user clicks on a button on the user interface of the computer to rotate the panel selected.

The next step was to show the route with LEDs. It is the starting point for my the Climbing Wall System Project. My task was to propose a solution which meets the requirements, to drives the LEDs of the wall from a computer, and to build a prototype. The work was huge :

- Reflect on the concept and the requirements
- Reflect on the electronic parts to use (a microcontroller, a LED driver, latches, etc ...)
- Reflect on the software to use
- Reflect on the further developments, etc
- Then build a prototype.

The plan of this report follows the progression of the project. After an introduction to the problem in chapter 1, where I describe the configurable concept, I do a requirements analysis in chapter 2. Then the technical options, with the function means analysis, are done in chapter 3. The electronics implementation is described in chapter 4 and the software implementation described in chapter 5. Recommendations and user instructions are given in chapter 6 and recommendations for future work given in chapter 7.

CHAPTER 1

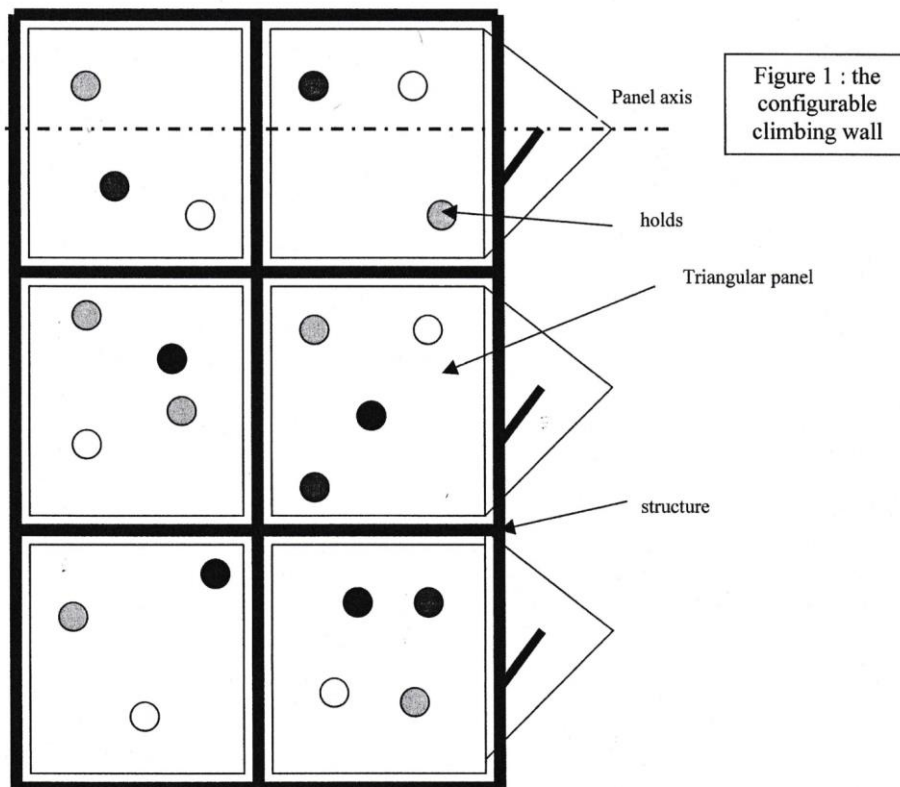
Introduction to the problem

The new concept created by DR international climbing wall may be revolutionary in the world of the climbing walls : the configurable climbing wall. This concept is a mechatronic solution to drive an automatic climbing wall from a computer. This concept can be divided into 3 major concepts : rotate panels, show route, rotate holds.

1) Rotate panels

The first concept, rotate panels, is currently developed by the company. A former student of the Mechatronic course in Lancaster University, J.Ward, is responsible for the design and the building of a prototype. His work is financed by a grant from the government. This concept was carried out in 1998 with a prototype made in the company workshop.

This wall is composed of triangular panels of 1 meter by 1 meter. They are set on a frame in metal. They rotate around their horizontal axis.



The concept follows these rules :

- each panel can be rotated around its axis, showing one of the 3 faces. It increases dramatically the number of available configurations : 46656 possibilities for a 6 panels wall.
- Each panel is rotated by a motor, so that the new configuration is driven automatically.

Figure 2 :
one panel

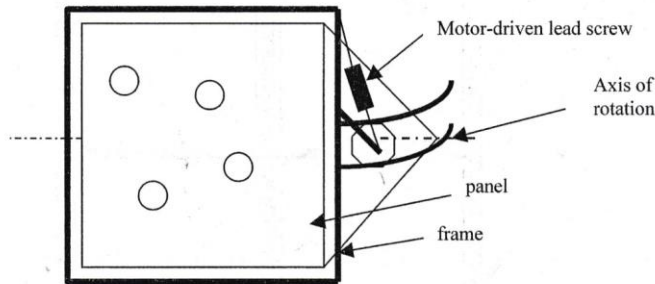
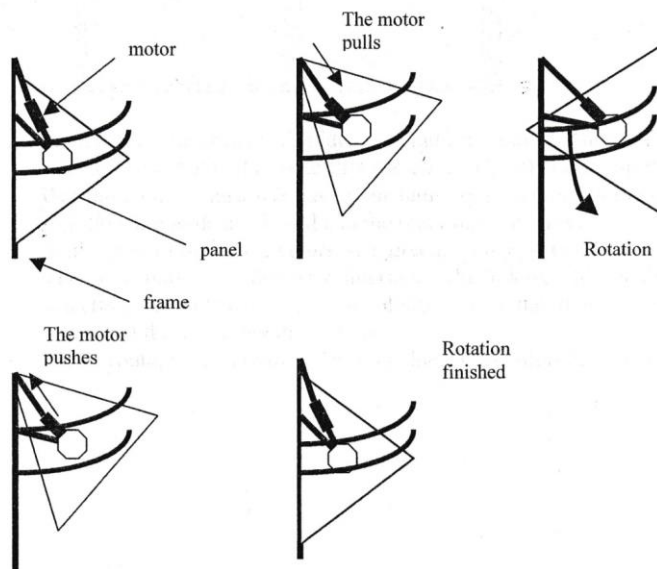


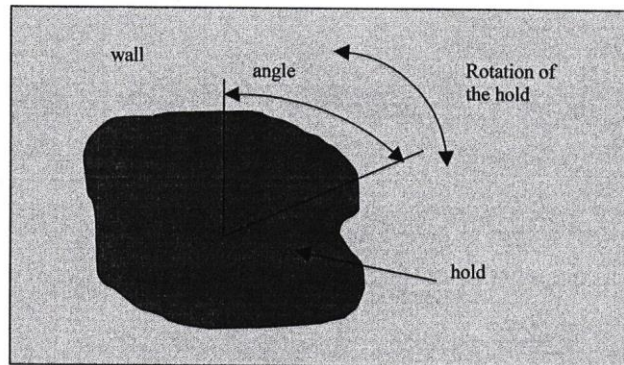
Figure3 : Way of
rotating panel



3) Rotate holds

- For each hold, the user chooses the hold angle on the user interface.
- The holds are then rotated by actuators to reach the correct angle on the wall.

Figure 5 : rotate hold



4) Aims of the project

The aims of the project was ambitious : create, with the company, a mechatronic solution of the configurable wall concept. By lack of time, it has been reduced to the development of a mechatronic solution for the concept show the climbing route. The Statement of requirement is sensibly different from the requirement for the whole system

CHAPTER 2

Requirement analysis

1) Statement of requirement

A) Statement of requirement for the configurable wall concept

The statement of requirement is done globally for the configurable wall. It is the transcription of the configurable wall concept. This statement of requirement includes the functional requirements, which specify what the system has to do, and the non-functional requirements, which are constraints.

a) Operational requirement

Change the climbing wall configuration and route

b) Functional requirement

Create and modify the panels and the wall characteristics : the wall characteristics are the position of each panel, position of the sockets, position of the holds.

Create a new configuration : the wall configuration is the panel side shown for each panel, the hold angle for each hold

Modify a configuration

Save and load a configuration : save data of a specified configuration

Indicate a route : with a particular configuration, show the climber the route he must follow by switching the lights of the holds he must use.

Save and load a route

c) Non-functional requirement

Safety :

Electrical : avoid electrocution by using low voltage

Mechanical : avoid mechanical breakdowns, particularly for actuators rotating the hold

General : avoid modification of the configuration during climbing

Cost :

maintain the configurable wall system in a price range acceptable to the customer. The extra-price must be compensated by the improvement of the system compared to a standard wall.

Reliability

Speed :

The acceptable time to change the configuration is 1 minute

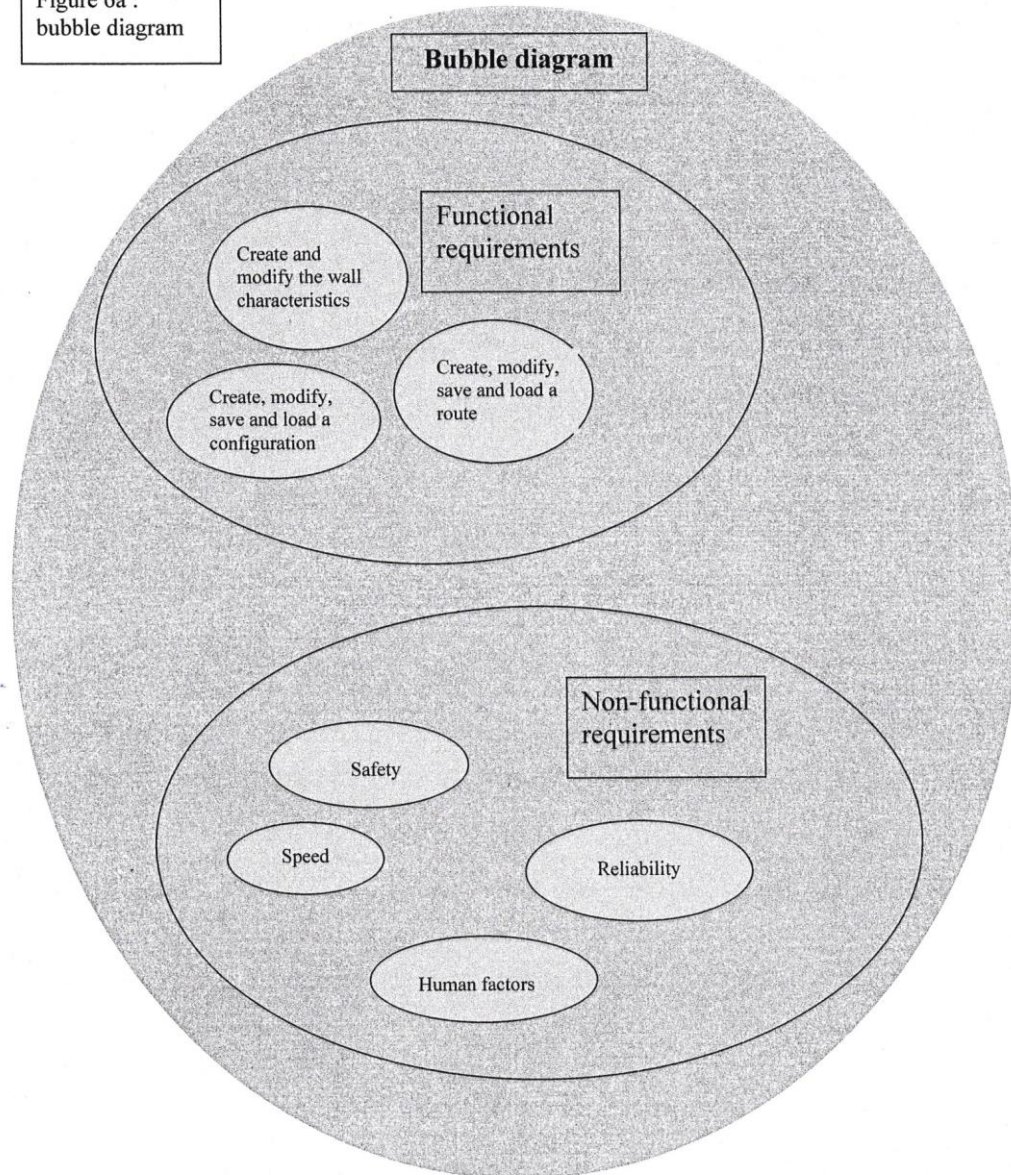
Human factor :

Easy to use

User interface : computer

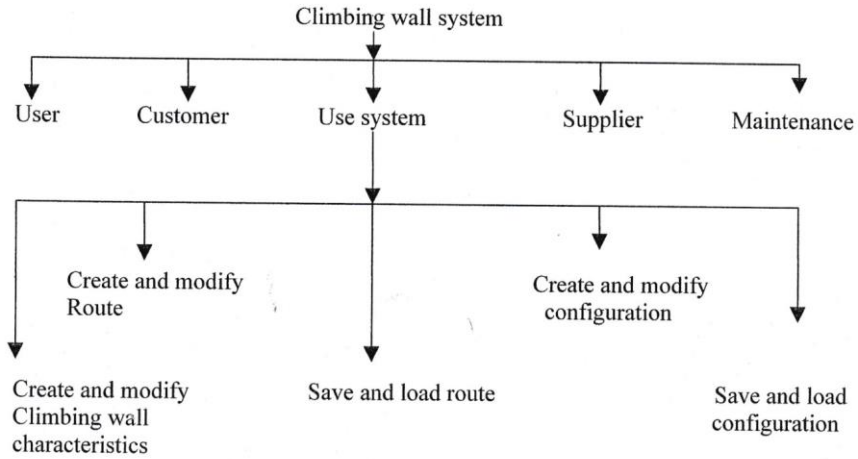
B) Bubble diagram for the configurable wall concept

Figure 6a :
bubble diagram



C) Structure chart for the configurable wall concept

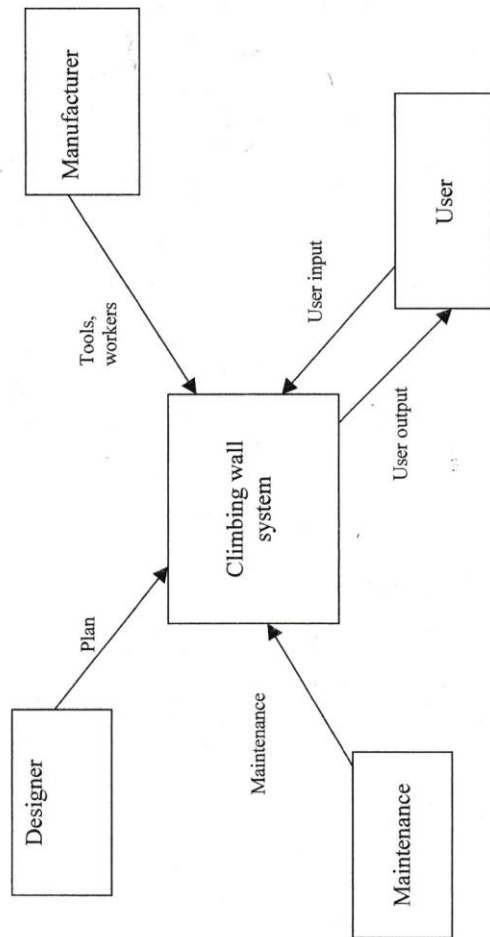
Figure 6b :
structure chart



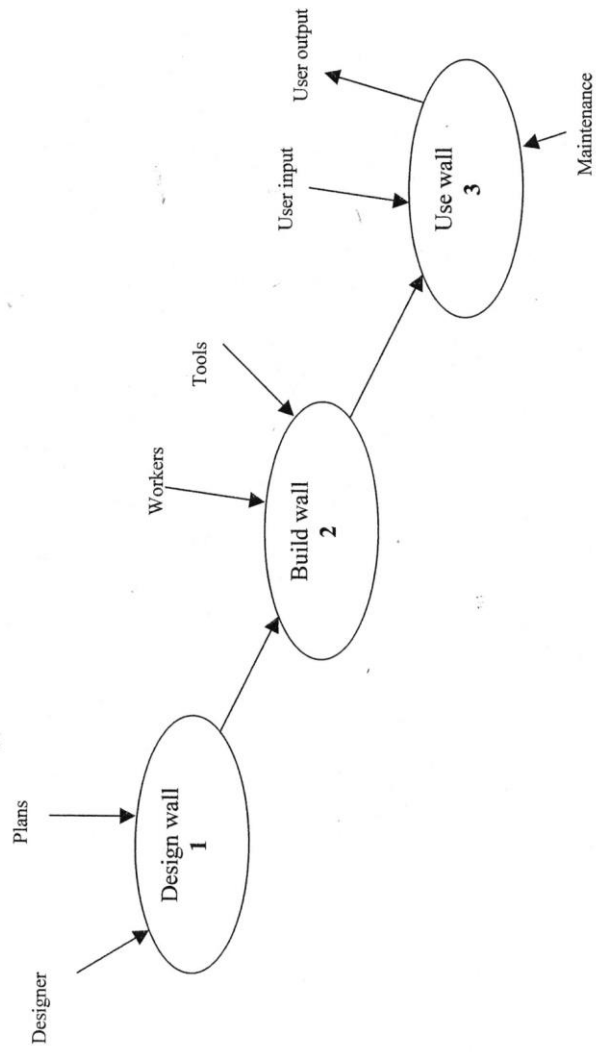
D) Data flow diagrams for the configurable wall concept

The data flow diagrams of the De Marco analysis is shown one next page.

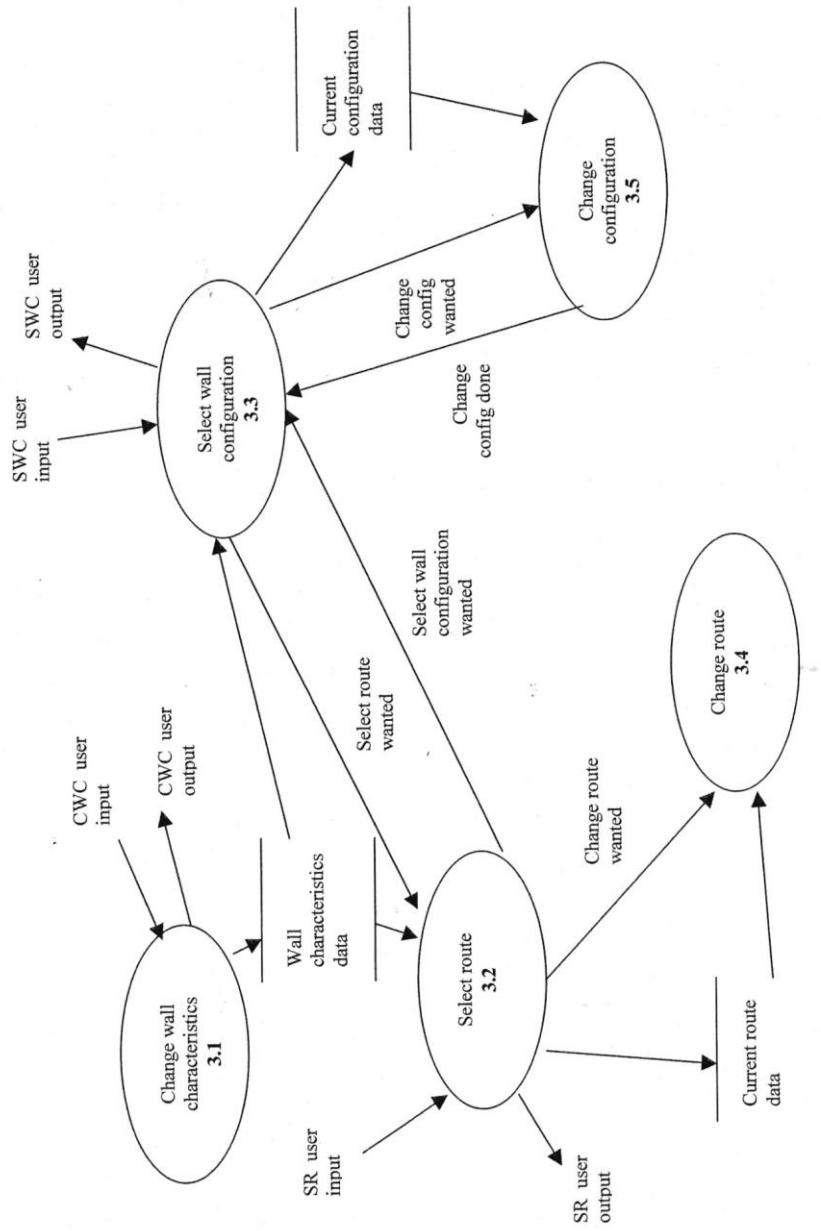
a) Context diagram : figure 7a



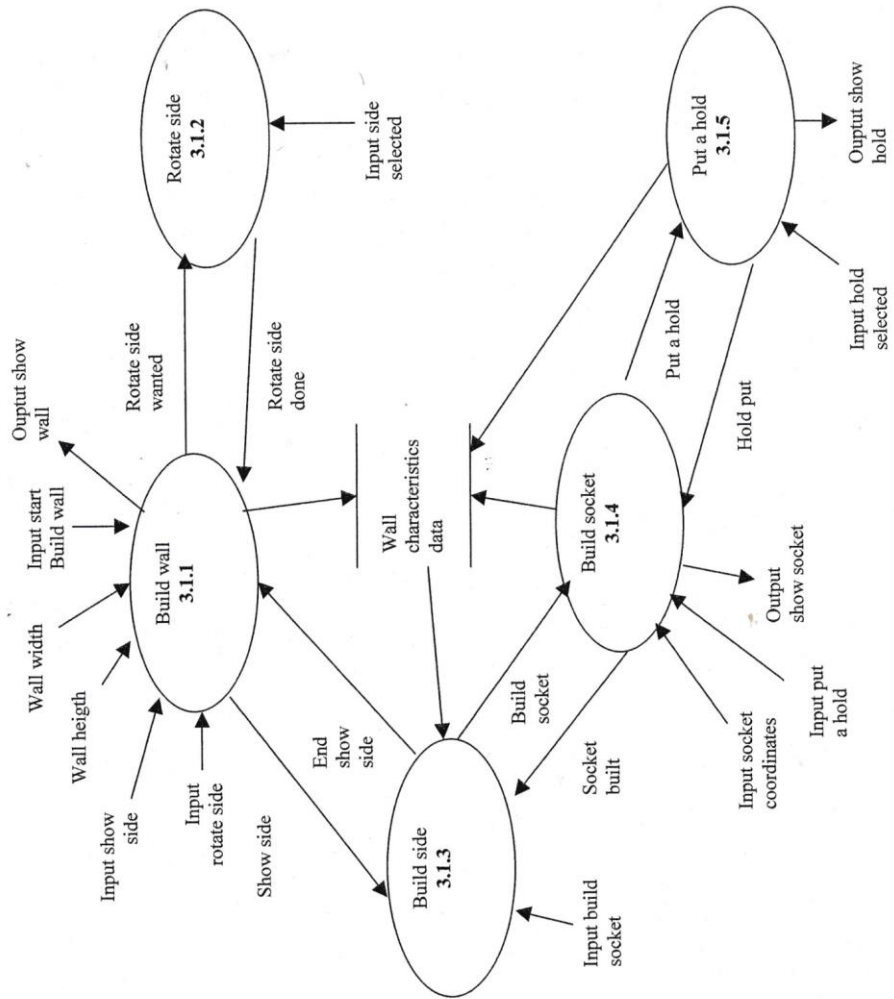
b) Diagram 0 : figure 7b



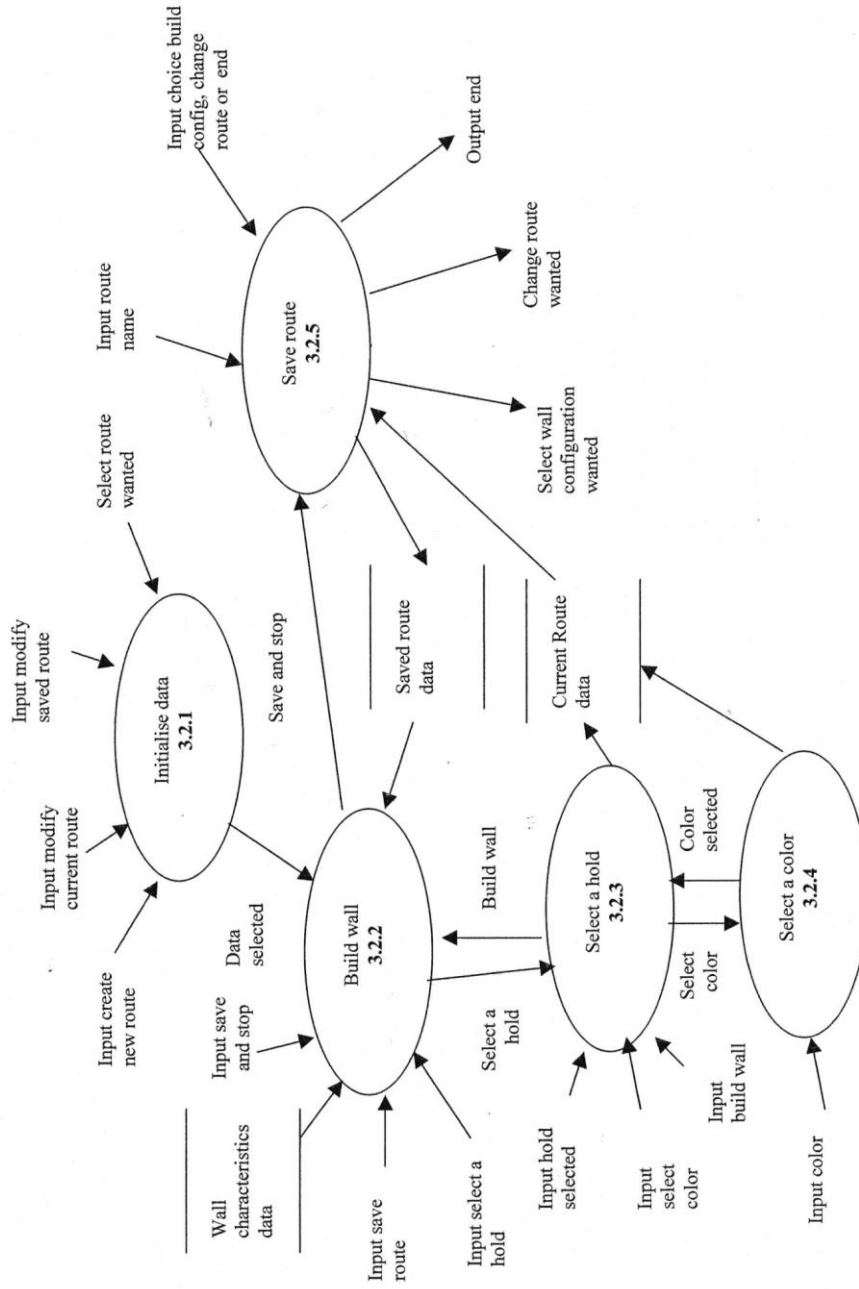
c) Diagram 3 : Use climbing wall : figure 7c



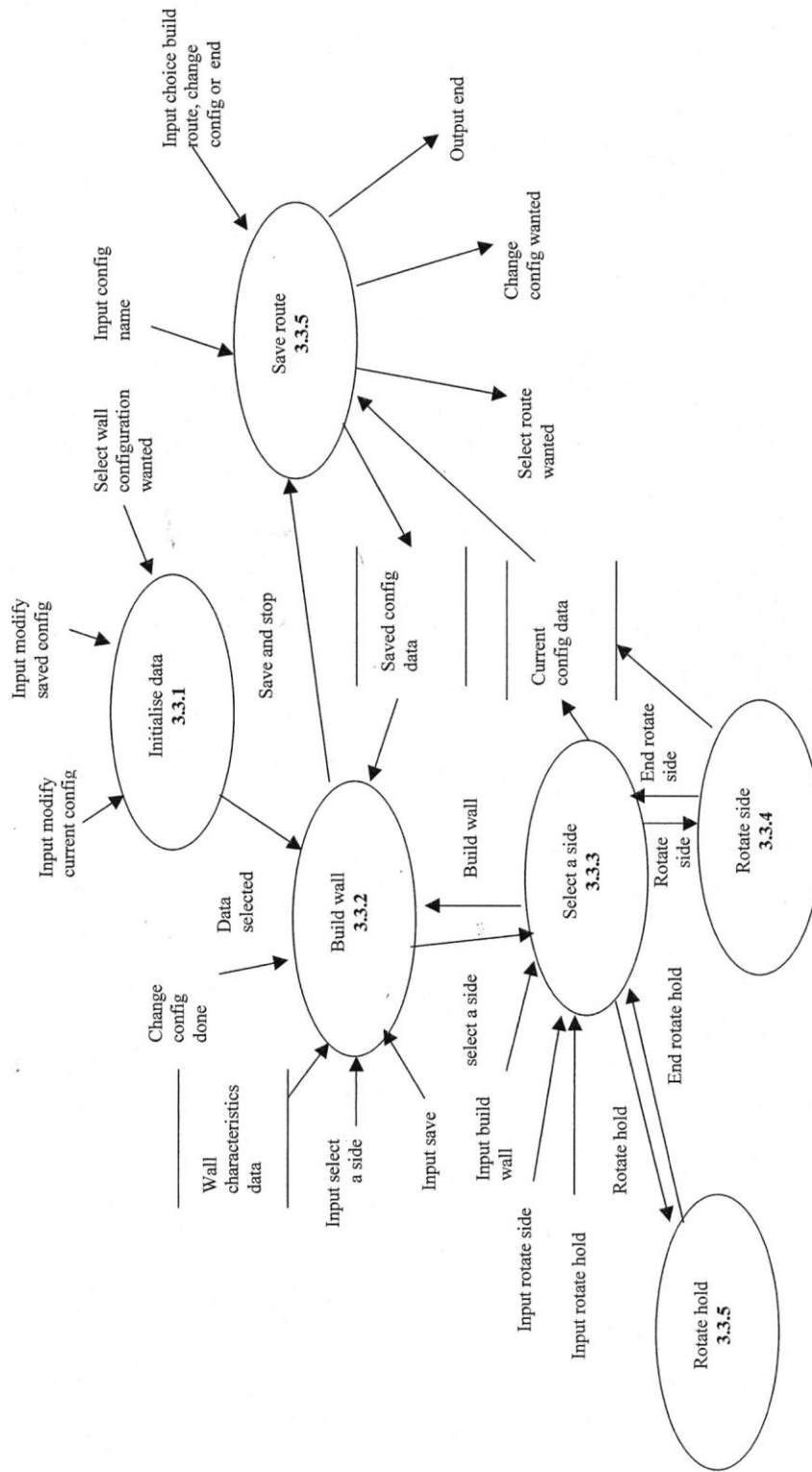
d) Diagram 3.1 : Change wall characteristics : figure 7d



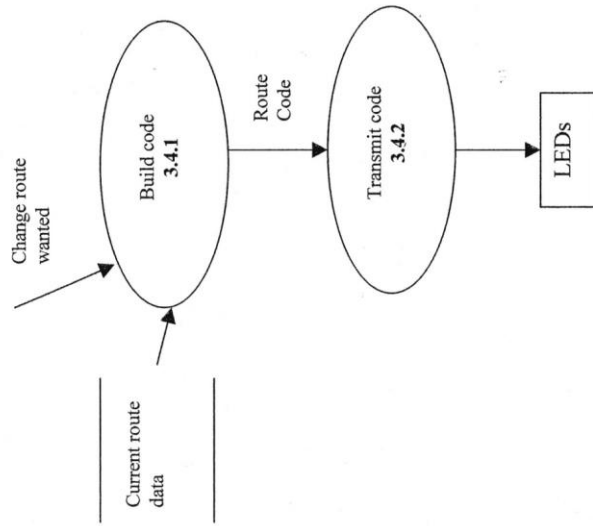
e) Diagram 3.2 : Select route : figure 7e



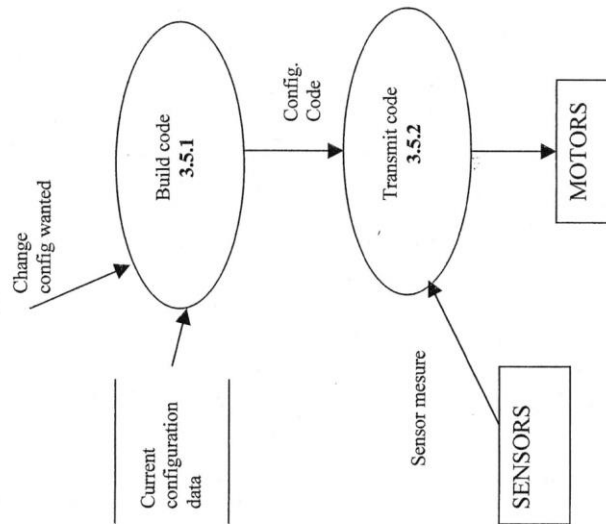
f) Diagram 3.3 : select wall configuration : figure 7f



g) Diagram 3.4 : Change route. Figure 7g



h) Diagram 3.5 : Change configuration : figure 7h



E) Process specification and process description of the data flow diagram

a) Diagram 3.1 : Change wall characteristics

This process allows the user to change the wall characteristics. It means that he can change the socket position for each panel and side, change the number of panels and change the side image on the computer.

b) Diagram 3.2 : Select route

This process allows the user to select a route on a particular configuration of the wall. It means that the user selects the lights of the holds he wants to use.

c) Diagram 3.3 : Select wall configuration

This process allows the user to select a configuration. A configuration is a particular arrangement of panels sides and hold angles.

d) Diagram 3.4 : Change route

This process does the actuation to change a route

e) Diagram 3.5 : Change configuration

This process does the actuation to change the configuration

f) Diagram 3.1.1 : Build wall

This process draws a wall on the screen of the computer

IF *input start build wall* message or *rotate side done* message or *end show side* message received

```
(
  Get wall width from the user
  Get wall heigth from the user
  FOR number of panels selected
    (
      Calculate panel coordinates
      Draw the panel
```

```
FOR each socket
(
  Calculate socket coordinates
  Draw sockets
  IF hold present in wall characteristics data
    Draw hold
  ENDIF
)
ENDFOR
)
ENDIF

IF input show side message received
Output show side message
ENDIF

IF input rotate side message received
Output rotate side wanted message
ENDIF

Modify Wall characteristics data
```

g) Diagram 3.1.2 : rotate side

This process allows the user to rotate a side on the computer screen

```
IF rotate side wanted message received
(
  Get input Side selected and wall characteristics data for the side selected
  Erase side
  Draw new side and new sockets
  Send rotate side done message
)
ENDIF
```

h) Diagram 3.1.3 : Build side

This process allows the user to draw a side on the entire screen

```
IF show side message or socket built message received
(
  Erase wall draw
  Draw side in function of side selected
  Draw sockets
  Draw holds
)
Send message build socket
ENDIF
```

i) Diagram 3.1.4 : Build socket

This process allows the user to put a socket on the side of a panel

```
IF build socket message received
  (
    FOR each socket
      (
        Get input socket coordinates
        Draw socket
        Save socket position in wall characteristics data
      )
    ENDFOR

    Send message put a hold
  )
ENDIF
```

j) Diagram 3.1.5 : put a hold

This process allows the user to put a hold on the side of a panel

```
IF Put a hold message received
  (
    FOR each hold selected
      (
        Get input hold selected
        Draw hold
        Save hold position in wall characteristics data
      )
    ENDFOR

    Send message hold put
  )
ENDIF
```

j) Diagram 3.2.1 : Initialize data

This process initializes the data used to create a route

```
IF input is input create new route
  Send Data selected message=New data
ENDIF

IF input is input modify current route or select route wanted
  Send Data selected message=Current data
ENDIF
```

```
IF input is input modify saved route  
    Send data selected message=saved data  
ENDIF
```

k) Diagram 3.2.2 : Build wall

This process draws a wall on the screen of the computer

```
SELECT data selected  
(  
    CASE is = new data  
        Create new route data  
    CASE is = current data  
        Download current route data  
    CASE is = saved data  
        Get file path  
        Download saved route data  
)  
END SELECT  
FOR number of panels  
(  
    Calculate panel coordinates  
    Draw the panel  
    FOR each socket  
    (  
        Calculate socket coordinates  
        Draw sockets  
        IF hold present in wall characteristics data  
            Draw hold  
            Draw LED  
        ENDIF  
    )  
    ENDFOR  
)  
ENDFOR  
  
IF input save and stop message received  
    Send save and stop message  
ELSE  
    Send select a hold message  
ENDIF
```


k) Diagram 3.2.3 : Select a hold

This process allows the user to select a particular hold in order to select its color

```
IF select a hold message received
(
  Get hold selected input
  Select this hold
  Send select color message
)
ENDIF
```

l) Diagram 3.2.4 : Select a color

This process allows the user to select the color of the LED of a hold

```
IF Select color message received
(
  Get color
  Change hold color
  Modify current route data
  Send color selected message
)
ENDIF
```

m) Diagram 3.2.5 : save route

This process allows the user to save a route built

```
IF Save and stop message received
(
  Get input route name
  Download current route data
  Save it as Saved route data
  SELECT input choice build config, change route or end
    CASE is = build config
      Send select wall configuration wanted message
    CASE is = change route
      Send change route wanted message
    CASE is = end
      Send output end message
  END SELECT
)
```

n) Diagram 3.3.1 : Initialize data

This process allows to initialize data used to build a configuration

```
IF input is input modify current config or select wall configuration wanted  
  Send Data selected message=Current data  
ENDIF
```

```
IF input is input modify saved config  
  Send data selected message=saved data  
ENDIF
```

k) Diagram 3.3.2 : Build wall

This process draw a wall on the screen of the computer

```
SELECT data selected  
  (  
    CASE is = current data  
      Download current config data  
    CASE is = saved data  
      Get file path  
      Download saved config data  
  )  
END SELECT  
FOR number of panels  
  (  
    Calculate panel coordinates  
    Draw the panel  
    FOR each socket  
      (  
        Calculate socket coordinates  
        Draw sockets  
        IF hold present in wall characteristics data  
          Draw hold  
          Draw LED  
        ENDIF  
      )  
    ENDFOR  
  )  
ENDFOR  
  
IF input save and stop message received  
  Send save and stop message  
ELSE  
  Send select a side message  
ENDIF
```

o) Diagram 3.3.3 : Select a side

This process allows the user to select a side in order to rotate it or to rotate a hold.

```
IF select a side message received
(
  Get side selected input
  Select this side
  Send end select side message
)
ENDIF
```

p) Diagram 3.3.4 : Rotate a side

This process allows the user to rotate the side wanted on the screen

```
IF End select side message received and input rotate side message received
(
  Get side selected
  Erase side
  Draw new side
  Send End rotate side message
)
ENDIF
```

q) Diagram 3.3.5 : Rotate hold

This process allows the user to rotate a hold of a side on the screen

```
IF End select side message received and input rotate hold message received
(
  Get hold selected
  Get hold angle
  Rotate hold on the screen
  Modify current config data
)
ENDIF
```

r) Diagram 3.3.6 : save configuration

This process allows the user to save a particular configuration

```
IF Save and stop message received
(
  Get input config name
  Download current config data
  Save it as Saved config data
)
ENDIF
```

```
SELECT input choice build route, change config or end
  CASE is = build route
    Send select route wanted message
  CASE is = change config
    Send change config wanted message
  CASE is = end
    Send output end message
END SELECT
```

s) Diagram 3.4.1 : Build code

This process allows to build the LEDs codes which will be transmitted to the LEDs

```
IF change route wanted message received
  (
    Get current route data
    Build Route code for each panel
    Send route code
  )
ENDIF
```

t) Diagram 3.4.2 : transmit code

This process allows to transmit the LEDs code to the LEDs and to decode it.

```
IF route code message received
  (
    Transmit route code to the LEDs
    Decode route code
  )
ENDIF
```

u) Diagram 3.5.1 : Build code

This process allows to build the code sent to the motor to rotate panels and rotate holds

```
IF change config wanted message received
  (
    Get current configuration data
    Get sensor measure
    Build config code for each panel
    Send config code
  )
ENDIF
```

v) Diagram 3.5.2 : transmit code

This process allows the user to transmit the code to the actuators rotating the panels and rotating the holds

```
IF config code message received
(
  Transmit config code to the LEDs
  Decode config code
  IF sensor measure message received
    Send sensor measure message
  ENDIF
)
ENDIF
```

F) Data dictionary

a) Context Diagram

Plans : plans results of the design of the wall
Tools : tools used to build the wall
Workers : workers who build the wall
Maintenance : maintenance work on the wall
User input = (CWC user input, SWC user input, SR user input)
User output = (CWC user output, SWC user output, SR user output)

b) Diagram 0

Plans : plans result of the design of the wall
Designer : designer of the configurable wall
Tools : tools used to build the wall
Workers : workers who build the wall
Maintenance : maintenance work on the wall
User input = (CWC user input, SWC user input, SR user input)
User output = (CWC user output, SWC user output, SR user output)

c) Diagram 3

CWC user input = (input start build wall, wall width, wall height, input show side, input rotate side, input build socket, input put a hold, input socket coordinates, input hold selected)
CWC user output = (output show wall, output show socket, output show hold)
SR user input = (input create new route, input modify current route, input modify saved route, input route name, input choice build config change route or end, input save route, input select a hold, input select a color, input build wall, input color)
SR user output = (output end)
SWC user input = (input modify current config, input modify saved config, input config name, input choice build route, change config or end, input select a side, input save, input rotate hold, input rotate side, input build wall)
SWC user output = (output end)
Select route wanted : message from the select wall configuration module to launch the select route module
Select wall configuration wanted : message from the select route module to launch the select wall configuration module
Change route wanted : message from the select route module to launch the change route module
Change config wanted : message from the select wall configuration module to launch the change configuration module
Change config done : message from the change configuration module to tell the select wall configuration module that the configuration is changed

Wall characteristics data = {panel number, panel coordinates, {panel side, {socket number, socket coordinates, presence of hold}}}
Current route data = {panel number, {panel side, {socket number, LED color}}}
Current configuration data = {panel number, panel side wanted, {hold angle}}

d) Diagram 3.1 : change wall characteristics

Wall width : user input giving the wall number of panels in width
Wall height : user input giving the wall number of panels in height
Input start build wall : user input to launch the module
Input show side : user input to launch the build side module
Input rotate side : user input to launch the rotate side module
Output show wall : wall shown on the screen
Rotate side wanted : message to launch the rotate side module from the build wall module
Rotate side done : message from the rotate side module to tell the build wall module that the side is turned
Input side selected : user input to give to side he wants to be shown
Show side : message from the build wall module to launch the build side module
End show side : message from the build side module to launch the build wall module
Input build socket : user input to launch the build socket module
Build socket : message to launch the build socket module from the build side module
Socket built : message to launch the build side module from the build socket module
Input socket coordinates : user input to give the coordinates of the socket selected
Input put a hold : user input to launch to put a hold module
Output show socket : output on the screen to show the socket
Put a hold : message from the build socket module to launch the put a hold module
Hold put : message from the put a hold module to launch the build socket module
Input hold selected : user input to select the hold shown
Output show hold : output on the screen to show a hold

d) diagram 3.2 : select route

Input create new route : user input to launch the select route module and create a new route
Input modify current route : user input to launch the select route module and modify a current route
Input modify saved route : user input to launch the select route module and modify a saved route
Select route wanted : message from the select wall configuration to launch the select route module
Data selected = data transmitted to the build wall module
Input save route : user input to launch the save route module
Input select a hold : user input to launch the select a hold module
Input save and stop : user input to launch the save route module
Save and stop : message from the build wall module to launch the save route module
Input route name : user input name of the route saved
Input choice build config, change route or end : user input to select the option
Select wall configuration wanted : message from the select route module to launch the select wall configuration module

Change route wanted : message to launch the change route module
Output end : output to end the program
Select a hold : message from the build wall module to launch the select a hold module
Build wall : message from the select a hold module to launch the build wall module
Input select a color : user input to launch the select a color module
Input build wall : user input to launch the build wall module
Input hold selected : hold selected by the user
Select color : message from the select a hold module to launch the select a color module
Color selected : message from the select a color module to launch the select a hold module
Input color : user input to give the color of the hold

Saved route data = {panel number, {panel side{LED number, LED color}}}
Current route data = {panel number,{ panel side{LED number, LED color}}}

e) Diagram 3.3 : select wall configuration

Input modify current config : user input to launch the select wall configuration module
modifying the current config
Input modify saved config : user input to launch the select wall configuration module
modifying a saved config
Select wall configuration wanted : message from the select route module to launch the select
wall configuration module
Data selected : data transmitted to the build wall module
Change config done : message from the change config module to tell that the configuration
has been changed
Input select a side : user input to launch the select a side module
Input save : user input to launch the save route module
Save and stop : message to launch the save route module
Input config name : user input to give the name of the config saved
Input choice build route, change config or end : user input to do the choice
Select route wanted : message to launch the select route module
Change config wanted : message to launch the change config module
Output end : end of the program
Select a side : message to launch the select a side module
Build wall : message to launch the build wall module
Input rotate hold : user input to launch the rotate hold module
Input rotate side : user input to launch the rotate side module
Rotate hold : message to launch the rotate hold module
End rotate hold : message to launch the select a side module
Rotate side : message to launch the rotate side module
End rotate side : message to launch the select a side module

Saved config data = {panel number, {panel side,{socket number, hold angle}}}
Current config data = {panel number,{ panel side,{socket number, hold angle}}}

f) diagram 3.4 : change route

Change route wanted : message from the select route module to launch the change route module

Route code : code transmitted to the LEDs

g) diagram 3.5 : change configuration

Change config wanted : message from the select wall configuration module to launch the change configuration module

Config code : code transmitted to the motor

Sensor mesure : mesure from the sensor

G) Statement of requirement for the show route concept

a) Operational requirement

Show the route to the climber

b) Functional requirement

Create a new route on the user interface:

Modify a route on the user interface

Save and load a route : save data of a specified route

Indicate the route on the wall

c) Non-functional requirement

Safety : Electrical : avoid electrocution by using low voltage

Cost : maintain the show route system in a price range acceptable by a customer

Reliability

Speed : The acceptable time to transfert data is 15 ms

Human factor :

Easy to use

User interface : computer

2) User interface

A particular attention must be dedicated to the user interface. Some conditions are required :

- The user interface must be user-friendly
- It must show a representation of the current climbing wall
- It can show the representation of one particular side, with its holds, sockets and LEDs
- By clicking on panels, the user should be able to select a side and rotate it on the screen, and so, change the wall configuration.
- Selecting a LED, the user can change the LED color, and so, create or modify a route
- Selecting a hold, the user can rotate it and select its angle
- When the user likes its configuration or its route, he can operate the real transformation of the wall
- the user can save a configuration (panels) or a route (LEDs)
- The user can load a configuration or a route

CHAPTER 3

Technical options

1) Function-means table

The next step is to analyse technical options thanks to the function-means table. It gives the different technical options available to satisfy the customer requirements. It allows the development of a design solution for the configurable wall concept.

Figure 8 : Function-means table for the configurable wall concept

Function	means				
Power supply	5V	12V	24V	unicolor	Tricolor
Indicate route	LED 5V	LED 12V	LED 24V	LED 220V	LIGHT 220V
rotate panel	steering motor	stepper motor	ac motor	dc motor	
rotate hold	steering motor	stepper motor	ac motor	dc motor	
Bus	serial bus	I2C bus	parallel bus		
low level Control unit	PC	microprocessor	LED driver	latches + PLD	serial latches
low level Control software	assembler	C/C++			
high level control unit/user interface	PC	microprocessor			
high level control software	C	Visual basic			

2) Technical options selection

The power supply chosen by Jim Ward to rotate panels is 24V. It gives enough power for the actuation and it is safe, avoiding electrocution.

A) Indicate route

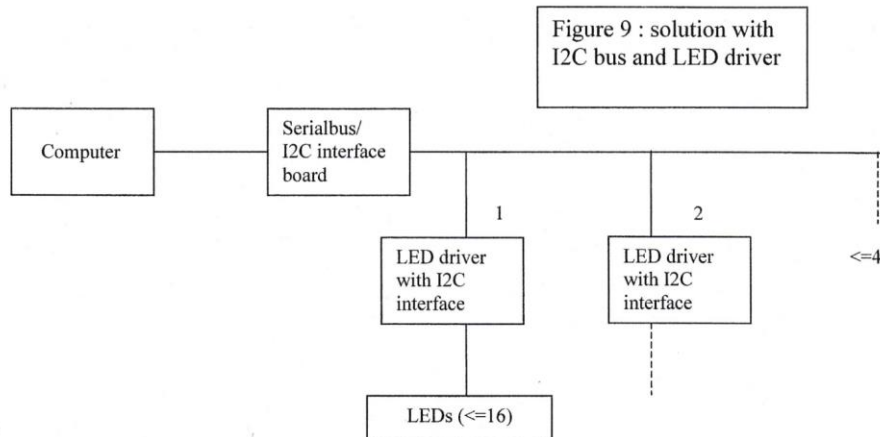
The customer requires that several routes can be showed by using different colors. The best way to indicate route is a LED. It is required that the system must be electric-safe and must show routes with differents colors.

To indicate route, LED tri-color using a 5V supply are selected.

B) Low level control unit and bus

a) I2C bus, serial/I2C interface, LED driver with I2C interface

With this option, the user command is transmitted to a LED driver, first by a serial bus, and then, after signal conditionning, by the serial/I2C interface. This LED driver can control up to 16 LEDs tri-color. The LED driver receives the data from the I2C bus and then powers the right LEDs.



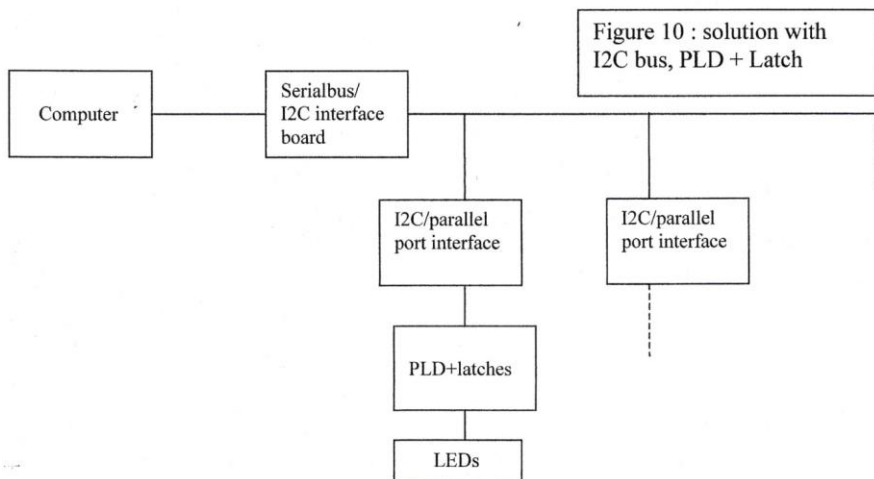
Advantage :

The I2C protocol is well known, reliable, quite easy to use
 The I2C is a serial bus. It avoids all the disadvantages of a parallel bus : loss of information due to a long distance transmission or to magnetic perturbations
 It is quite a cheap solution

Disadvantage :

There are only four LEDs drivers addressable per system and 16 LEDs maximum per driver. A solution is to use one serial/I2C interface for four LED drivers. But in this case, the cost is increased dramatically
 This item is not easy to get from Philips semiconductors

b) I2C bus, I2C interface, PLD+latches



The user command is transmitted to the I2C/parallel port interface , first by a serial bus, and then, after signal conditioning, by the I2C bus. This command is a 24 bits word giving the address of the I2C/parallel port interface, the address for the PLD and the data (LEDs code).

Advantage :

The I2C protocol is well known, reliable, quite easy to use.

The I2C bus is a serial bus : it avoids all the disadvantages of a parallel bus : loss of information due to a long distance transmission or to magnetic perturbations.

The serial/I2C and I2C/parallel port interfaces, with their software, have been built for the M.Eng project WARP. These components can be re-used without any modification.

Disadvantage :

The main disadvantage is the price. Each board cost a bit more than 40 pounds, and the latches+PLD board costs 20 pounds.

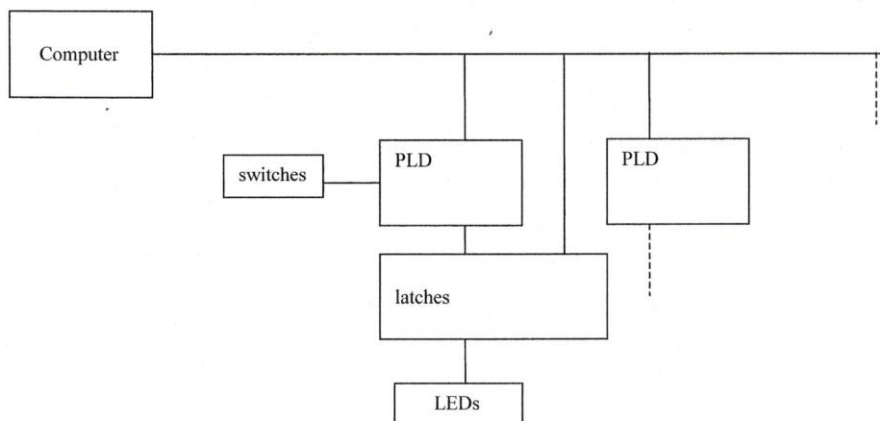
The other disadvantage is that the electronics are built in the university. It means that the company will have to build it or get it built, where it would have preferred to buy a ready-to-use board.

It is quite a complicate solution to control just LEDs. But the advantage is that the same electronics and software may be used by the company for further development.

c) Serial bus, PLD, latches

The previous solution can be modified to avoid the use of expensive interfaces. The PLD would be used not as a combinatorial logic device, but as a sequential logic device.

Figure 10b :
solution with serial
bus, PLD, Latches



The bus is a 2 wires serial bus, one wire for the clock, the other one for the data. The user command would be coded as a 16 bits word. The 8 first bits would be the PLD address. The address of the PLD is given by the switches. When a PLD is addressed, it enables the latch. The next 8 bits are then decoded as data by the latch and transmitted to the LEDs. The cycle continues then.

Advantage

It is a cheap solution

It is a simple solution, where the electronics are not complicate and don't need any software except the programming of the PLD

Disadvantage

The serial bus is not reliable. The use of sequential logic is the main reason. If, for one reason or an other, a PLD loses the sequence, then it won't work correctly. A solution is to use a special code to specify the beginning of a word.

d) Conclusion

The first solution, with I2C LED drivers, is not selected, because there are only four LEDs addressable for one bus.

The third solution, with PLDs and latches, is not selected because it is not reliable.

The second solution is selected, even if it is more complicate and expensive than the third solution. The main reasons why it is selected are that it allows further development on the same boards, and that these boards already exist and work. This solution is robust.

C) Bus

The technical solution selected implies that both a serial bus and an I2C bus are used. The reason why a serial bus is employed is that there is no other cheap solution to link a PC with a cheap board having an I2C interface.

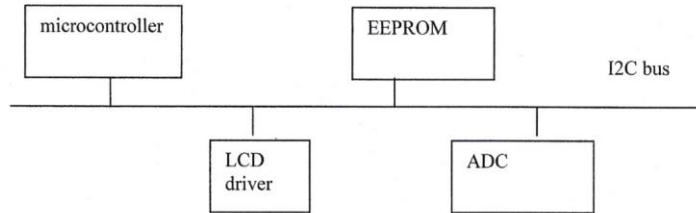
a) The I2C bus and how to use it

This is a resume of the Philips semiconductors application note "The I2C bus and how to use it" shown in appendix 6.

The I2C bus is a serial bus with three cables (the serial data line SCL, the serial clock line SCA and the ground GRD) which can link differents systems : microcontroller, RAM, EEPROM, LCD driver, LED driver, ADC, etc ...

The SDA and the SCL carry information between the devices connected to the bus. Each device is recognized by a unique address and can operate either as a transmitter or as a receiver. The transmitter is the device which sends the data to the bus and the receiver is the device which receives the data from the bus. Each device can be a master or a slave. The master is the device which initiates the transfer, generates clocks and terminates a transfer.

Figure 11a : I2C bus



Advantages of the I2C bus :

- it is a simple 2-wires serial bus which minimizes the interconnection. So ICs have fewer pins and there are not so many PCB tracks.
- It is a completely integrated I2C bus protocol which eliminates the need for address decoders and other glue logic
- Multi-master capability

A system with an I2C bus is very flexible. The system can be upgraded to keep designs up-to-date by adding new equipments or changing some equipments.

Resume of the I2C bus protocol

Start and stop conditions :

The start condition is indicated by a High to Low transition on the SDA line while SCL is High.

The stop condition is indicated by a Low to High transition on the SDA line while SCL is High.

Data validity :

The High or Low state of the data line can only change when the clock signal on the SCL line is Low.

The byte format :

Every byte put on the SDA line must be 8-bits long. The number of bytes is unrestricted. Each byte has to be followed by an acknowledge bit. Data is transferred with the most significant bit first. If a receiver cannot receive data, it can force the SCL Low to force the transmitter into a wait state.

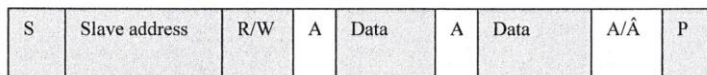


Figure 11b : Byte format for the I2C protocol

From master to slave

From slave to master

A = Acknowledge (SDA low)
 Ā = Not acknowledge (SDA high)
 S = start condition
 P = Stop condition

b) Conclusion

The I2C bus has been chosen for all the advantages said. But as the card to connect it to the PC has not been bought, a serial bus must be used to connect the PC with the I2C board.

D) Low level control software

The technical solution selected implies that some software for the low level control is re-used from the WARP project. For the moment, this software is written in assembler. Sean Collins is working to make it work in C.

E) High level control unit

The high level control unit selected is a PC.

F) High level control software

The choice has to be made between the C language and Visual basic language.

The reasons why the language chosen is Visual Basic are :

- the work already done by the company has been written in Visual Basic.
- Visual basic meets the user interface requirements :
 - It can show a representation of the current climbing wall and of a side
 - Click-buttons, drag and drop mouse, menus; et..., are available to make the user interface user-friendly
 - It can operate the real configuration of the wall
 - It is user friendly
- It is an object-oriented language. Each panel and each LED is an object. Then Visual Basic is well adapted to the work required.

G) Power supply

The power supply required by the board and the LED is a 5V power supply.

3) Technical proposition and schematic description

The technical proposition is a summary of the technical options selected

The user interface is a PC where the language used is Visual basic. The user types is command on a user-friendly user interface.

The user command is transmitted via a serial bus to a serial/I2C interface which transform the serial protocol data into I2C protocol data.

The command is then transmitted via the I2C bus to the I2C/parallel port interface addressed.

This interface transforms the I2C protocol data into a EWR signal, the address and the data.

The address and the EWR signal are transmitted to the PLD of the LEDs board via the parallel port. The EWR signal enables the PLD. When it is enabled and in function of the address, the PLD enables one latch or the other.

The data is transmitted to the latches of the LEDs board via the parallel port. When it is enabled, the latch enables its output in function of the input data.

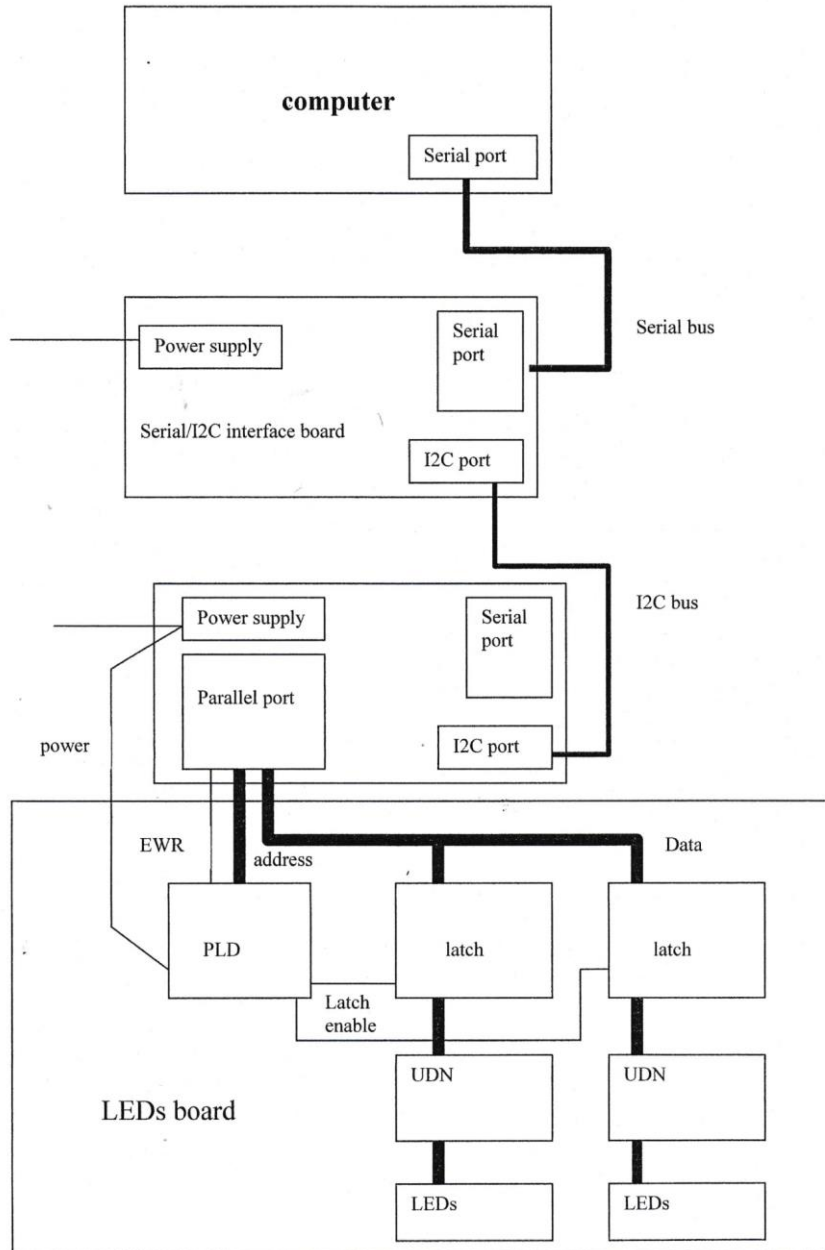
The latch output is connected to an UDN, which amplify the signal.

The UDN output is connected to resistors and tri-color 5V LEDs.

To resume, After the transmission via a serial bus and an I2C bus, the command is decrypted, amplified on the LEDs board and transmitted to the LEDs.

The schematic of the system is shown on the next page

Figure 12 : schematic
of the system



CHAPTER 4

Implementation

1) The controller board

The technical proposition is done. The next step is to implement the system.

As said before, the system re-uses the embedded controller board made by the Walking autonomous robot team. All the information on this board is an extract from the 80C552 embedded computer board user manual shown in appendix 10 and the WARP report.

A) The microcontroller

The board is based on the 80C552 16 MHz microcontroller from Philips. The 80C552 is a microcontroller from the widespread 8051 family, with the same instructions.

Characteristics of the microcontroller :

- 256*8 read/write memory
- five 8 bits I/O port
- three 16 bits timers
- 8*10-bit ADC
- a serial I2C interface and a serial UART interface
- a watchdog timer

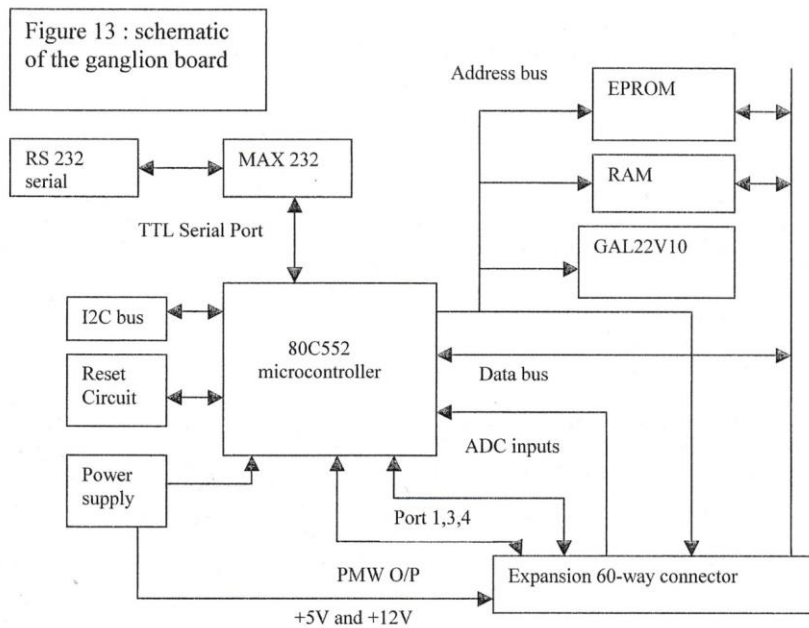
Cf appendix 9 for the microcontroller data sheet

B) The board

a) Functional overview of the system

The board features the following components

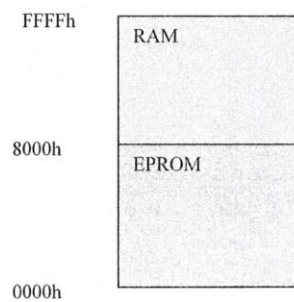
- 80C552 microcontroller
- 32K Eprom
- 32K RAM
- Serial RS232 to TTL conversion
- GAL22V10 5memory decoding^o
- Power regulation
- Reset circuitry
- Expansion connector
- I2C connector



b) Memory organisation

The 80C552 can address up to 64K bytes of program memory and up to 64K bytes of external RAM. The memory map is organised as following : the 64 K bytes of address space are divided into two 32K bytes segments. The EPROM is located on the lower segment (0000h to 07FFh) and the RAM is located on the upper segment (8000h to FFFFh).

Figure 14 :
Memory
organisation



The memory configuration is similar to that described in the Philips application note AN440. The program code can be downloaded via the serial port on the SRAM and then executed from the SRAM, by using a bootstrap downloader program. This program is installed onto the EPROM. When the development is finished, the program code can be programmed directly on the EPROM. The advantage of using the bootstrap program is that, during development, the EPROM doesn't need to be reprogrammed after each change.

The memory decoding is done by a PLD.

c) Serial communication

The microcontroller serial port is connected to the computer by the RS232, a 9-way D connector. The serial RS232 to TTL logic conversion is handled by the MAX232 circuit.

The 9-way D connector pin-out are connected as follow

Pin	Name	function	note
1	DCD	Data Carrier Detect	
2	TX	Transmit	Connected
3	RX	Receive	Connected
4	DTR	Data Terminal Ready	Connected to DSR
5	GND	Ground connection	Connected
6	DSR	Data set ready	Connected to DTR
7	RTS	Request to send	Connected to CTS
8	CTS	Clear to send	Connected to RTS
9	RI	Ring indicator	

Board port

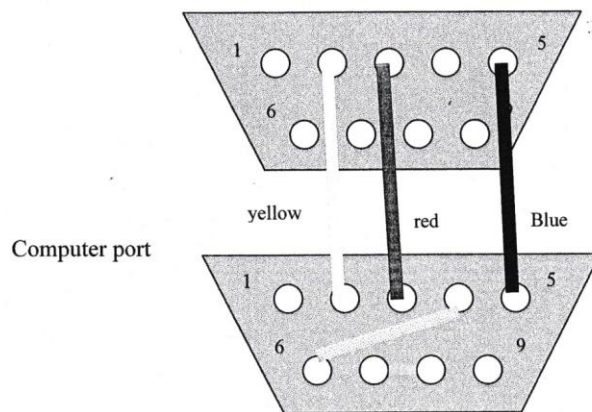


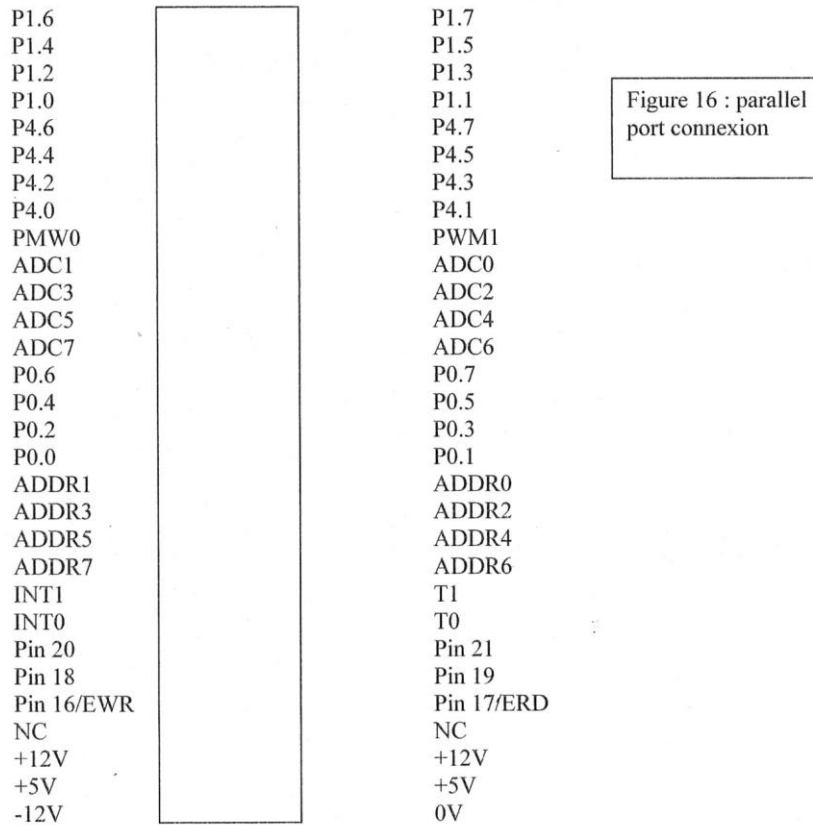
Figure 15 : serial port connexion

d) I2C bus

The I2C bus is a serial bus which has two lines, the data(SDA) and the clock (SCL). The board has a 3 pins connector which is connected to the microcontroller pins.

e) 60-way expansion connector

The expansion connector is the link between the ganglion board and the LED board.



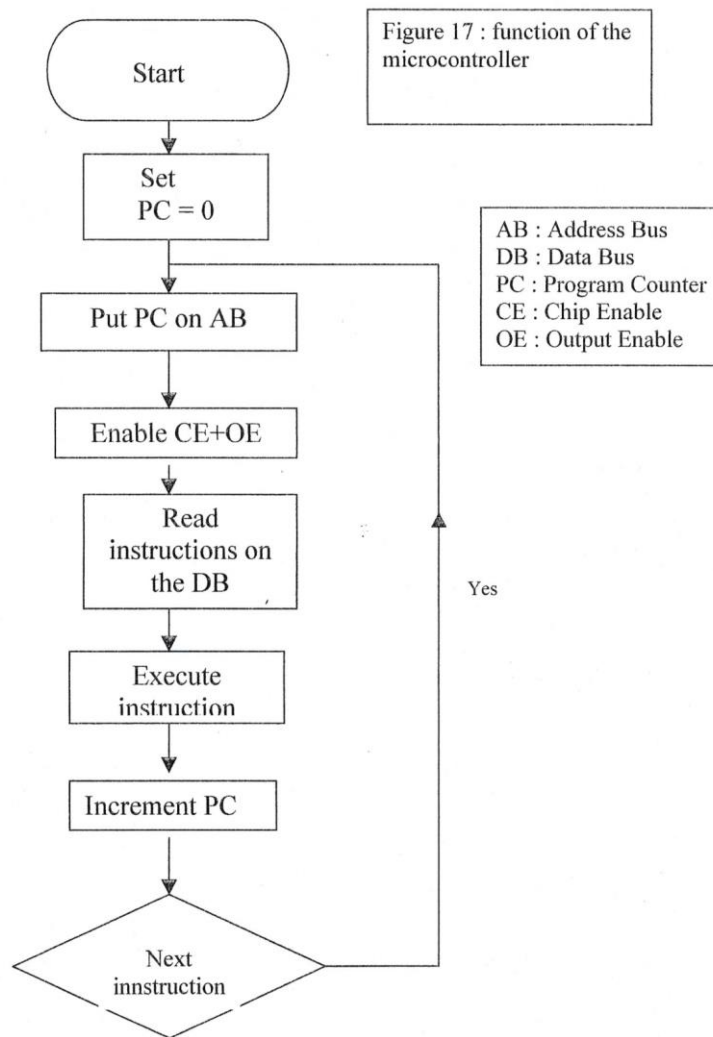
C) Description of function

The board is first reseted by the push button. The internal reset algorithm writes 0s to all the SFRs (Special function register) except the port latches, the Stack pointer and the serial buffer. This operation is done to initialize the counter and avoid the CPU executing instructions from an indeterminate location.

The data is received from the serial port or from the I2C bus. The program is written so that it knows whether the data comes from the serial port or from the I2C bus.

The microcontroller then works basically as shown below. When it is reseted, the CPU sets the counter=0. Then it increments the counter, and gets from the RAM or the EPROM (it depends on the previous instruction) the code pointed by the counter. Then it executes the code. Then it starts again the loop.

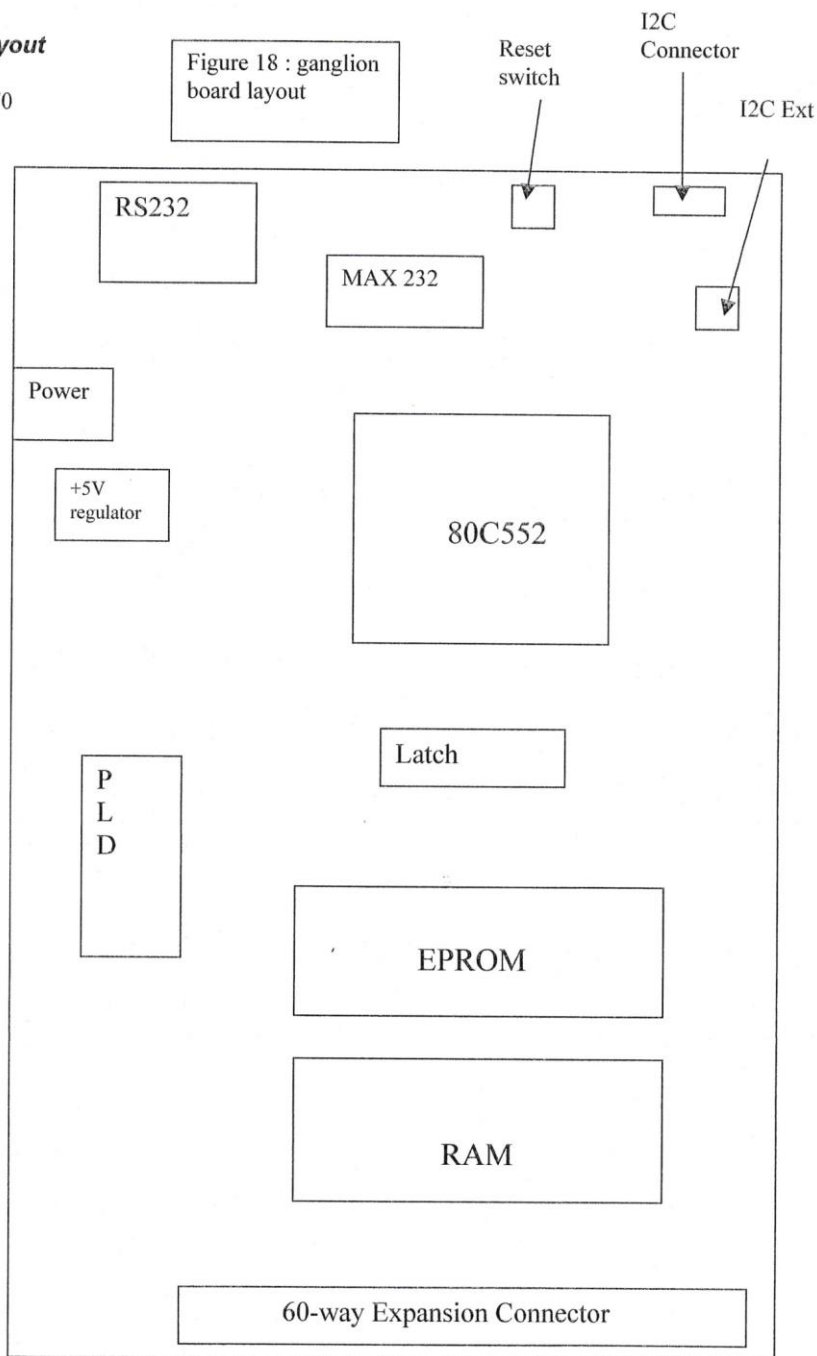
Some instructions command to outport or inport codes on the expansion 60-way connector, on the serial port or on the I2C bus. Codes are read on the serial port when an instruction has to be downloaded from the computer. Codes are read on the I2C port when an instruction has to be downloaded from the selected address board. Codes are written on the parallel port when an instruction has to be downloaded on the LEDs board.



D) board layout

Size : 130*170

Figure 18 : ganglion
board layout



E) Test module

a) Testing recommendations

Several stages are required to test the modules. These tests are recommended by the board developers in the user manual. I followed the procedure below :

First stage :

Check that the tracks are not disrupted visually and with a voltmeter if necessary.

Second stage :

Power the board without any ICs inserted. The Power LED should light, signifying a 5V signal from the regulator. If the power LED doesn't light, check the power supply circuitry. Otherwise, do the third stage.

Third stage :

Insert all ICs, including the microcontroller, the EPROM, the PLD, the Latch, the RAM. The PLD GAL22V10 must be programmed with WARP.JED. The EPROM is programmed with bootan2.hex, the bootstrap program. Download the test code strobe.hex on the RAM. This program simply toggles the pins on port 1, producing a square wave. Use an hyperterminal on the computer to download this codes. If the strobe program doesn't work, check all data, address, output and chip enable lines with an oscilloscope.

Fourth stage

Download the wanted program. In the case of the project, the program is moveto.hex. If the codes can be downloaded and executed, then the board is fully working.

b) Testing the board

Two board have been built for the project.

The first one fully succeeded the three first stages of the testing. One week was necessary to make it succeed the fourth stage. It was not because the board was not working, but because the LED prototype board was not working. All the tracks and all the components have been checked successfully.

The second board was not working. It did not check the third stage testing requirements. All the tracks and components have been checked. The crystal, giving the clock speed, was out of order. There was no signal on the oscilloscope. It was replaced. The RAM was not working, and was the second reason why it was not working. It was replaced. Now it works successfully.

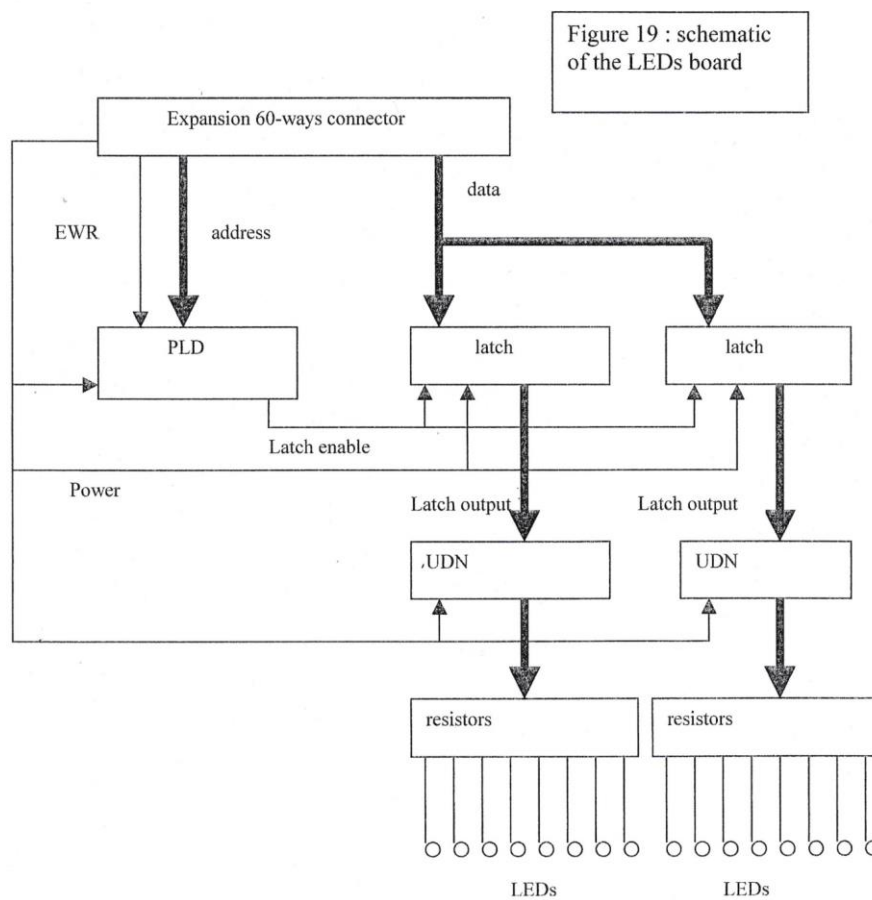
2) The LED board

I designed this board. The first step was the conceptual design

A) Functional overview of the board

The board features the following components :

- A 60-way expansion connector
- A PLD GAL22V10
- Two latches
- Two UDN
- 16 resistors



B) Explanation of the functioning

a) The expansion connector

The expansion connector is mapped exactly the same way as the expansion connector of the ganglion board. The pins connected are the power supplies, the EWR pin, the address pins and the data pins (Port 0).

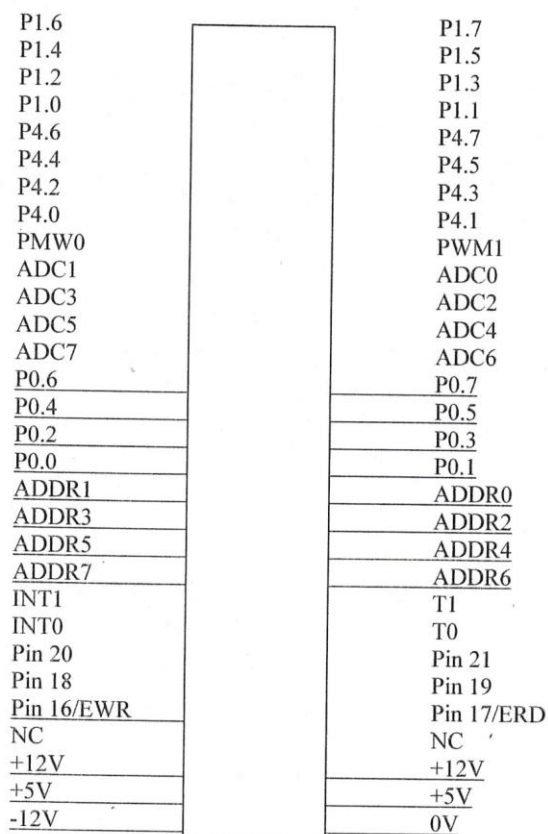


Figure 20 : parallel port connexion 2

The pins connected are the ones for which the specification is underlined.

The power supply pins provide power to all the ICs of the board : PLD, latch, UDN.
The EWR pin is the PLD enable.

The address pins are connected to the PLD. They give the 8 bits address for the data.

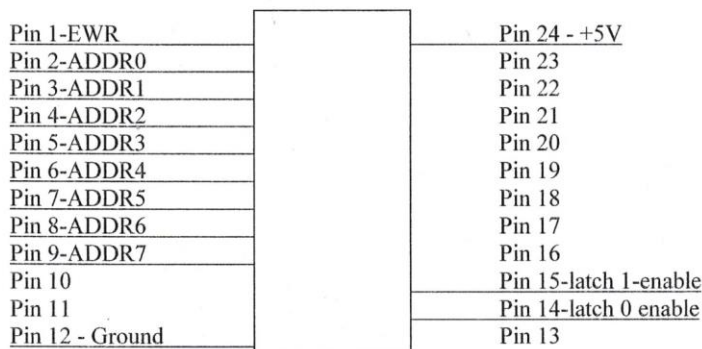
The data pins are connected to the latches. The data is a 8 bits code. If the LED color must be lighted, the bit is high. Otherwise it is low.

b) Programmable logic device

The PLD is a popular GAL 22V10. When the EWR pin is logic low, on pin 1, the PLD is enabled. It gets the address from the connector on the pins 2 to 9. The address is a 8 bits code. If all the bit of the address are logic low, then it enables the pin 14 which is connected to the latch enable of the latch 0. If all the bits of the address are logic low except ADDR0 connected to pin 2 which is logic high, then it enables the pin 15 which is connected to the latch enable pin of the latch 1. Otherwise, it doesn't enable any latch.

The PLD is programmable. It means that a program has to be written on a software called Palasm. The codes of the software are in annexe 5 : Fredled.pds. Then the device is programmed on an emulator with the software Fredled.jed.

Figure 21 : PLD connexion



The pins connected are those underlined.

The GAL 22V10 datasheet is shown in appendix 7 .

c) Latch

When the latch enable is enabled, which means logic high, the latch input logic data is put high on the output and kept. The latch inputs and outputs are 8-bits words.

The latch datasheet is shown in appendix 8.

d) UDN

The UDN aims to power amplify the signal received from the latch, in order to switch the LEDs. The UDN accepts.

e) Power supply

The power supply is the 5V power supply of the ganglion board. The board does not have any independant power supply. The current is transmitted through the 60-way connector.

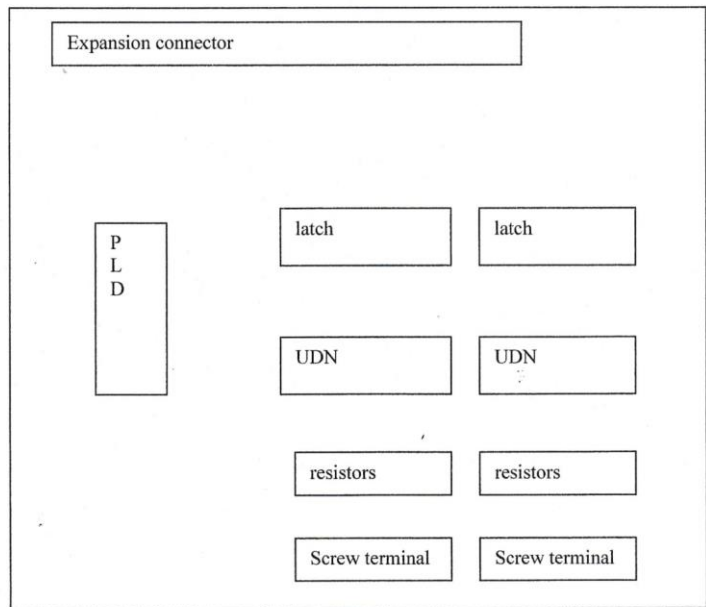
B) Description of the function

When the EWR signal is logic low on the pin 1 of the PLD, the address of the latch is red on the pins 2 to 9 of the PLD. The PLD enables the latch addressed. This latch puts the data input on the output and keeps it. This signal is then amplified by the UDN and sent to the LEDs via resistors.

C) Board layout

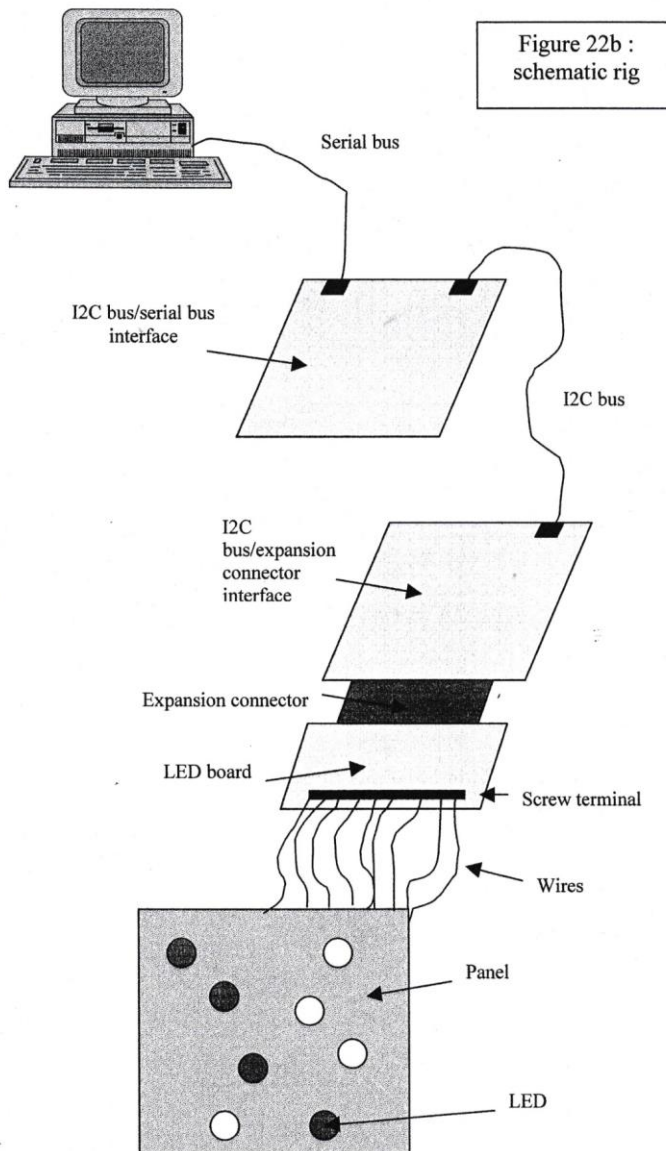
Cf appendix 11 for the board layout

Figure 22a : LEDs
board Layout



3) The rig

The association of the boards with the computer and the LEDs on the panel gives the rig. The schematic of this rig is shown below



CHAPTER 5

Software Implementation

1) Software for the ganglion boards

The software for the ganglion is re-used. It was written by Sean Collins from the M.Eng WARP team.

The first board has the software gateway.hex on it. This software is in appendix 2. The program gateway gets the data from the serial input and transmits the data to the ganglion board addressed by the I2C bus.

A) Schematic diagram for the procedure board gateway

The program gets the data from the serial port which is a code like 04P1F3.

The code looks like this : 04P1F3 in hexadecimal.

The 04 command is the address of the ganglion controlling one particular set of LEDs.

The P command is the name of the command to switch the LEDs.

The 1 is the address command for the PLD

The F and the 3 are two four bits datas. They code the color of the LEDs. The link between the color of the LEDs and the hexadecimal code is shown below

Hexadecimal	decimal	color LED1	Color LED2
0	0	n	n
1	1	R	n
2	10	G	n
3	11	O	n
4	100	n	R
5	101	R	R
6	110	G	R
7	111	O	R
8	1000	n	G
9	1001	R	G
A	1010	G	G
B	1011	O	G
C	1100	n	O
D	1101	R	O
E	1010	G	O
F	1111	O	O

n : No color

R : Red color

G : Green color

O : Orange color

The procedures get data from the serial port and put the data to the I2C bus with the I2C protocol are detailed on next page.

As explained by the schematic, the program simply gets data from the serial port of the board linked to the computer and puts it on the I2C bus of the board.

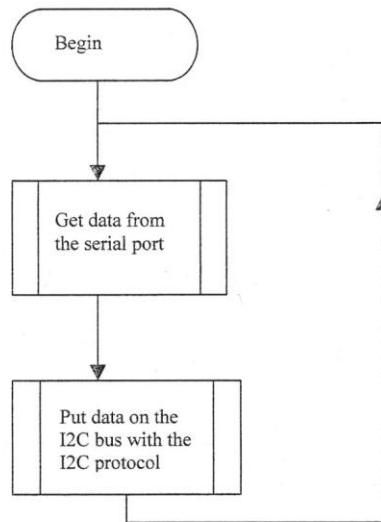


Figure 23 : procedure board gateway

B) Get data from the serial port

This is a sub-procedure of the procedure board gateway. It gets data from the serial port of the board. It first "installs" the serial bus protocol of the board and then it gets data from this serial bus with the get data procedure detailed on next page.

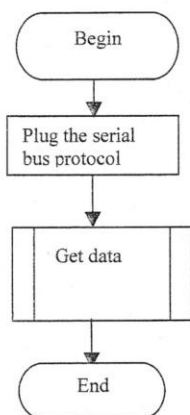


Figure 24 : sub-procedure get data from serial port

C) Get data from the I2C bus

This procedure is similar to the procedure get data from the serial bus. In this case, the bus is the I2C bus. This procedure is a sub-procedure of the procedure board moveto. It first "installs" the I2C bus protocol of the board and then it gets data from this I2C bus with the get data procedure detailed on next page.

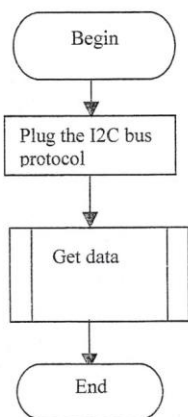


Figure 25 : sub-procedure get data from the I2C bus

D) Get data procedure

This procedure is a sequential sub-procedure of the procedures get data from the serial bus and get data from the I2C bus. It first gets a start signal which begins all transmission. Then it gets the ganglion address, which is the 04 of the example code 04P1F4. It then get the PLD address, which is the P1. It then gets the first and second LED data, which are respectively the F and the 4. It then goes back to the main procedure

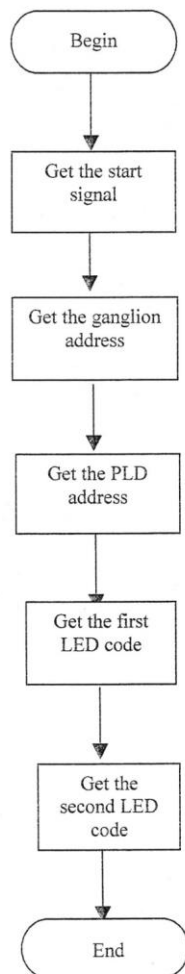


Figure 26 : sub procedure get data

E) Put data on the I2C bus

This is a sub-procedure of the gateway procedure. When the data has been received by the gateway board thanks to the get data from the serial bus procedure, it has to be put on the I2C bus thanks to this procedure. This procedure installs the I2C bus protocol to export data. Then it launch the sub-procedure put data and come back to the main procedure gateway.

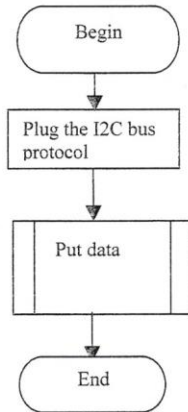


Figure 27 : sub procedure put data on the I2C bus

F) Put data

This procedure is a sub-procedure of the procedure put data on the I2C bus. It first puts the start signal on the bus. Then it puts the ganglion address, the PLD address, the first and second LED codes.

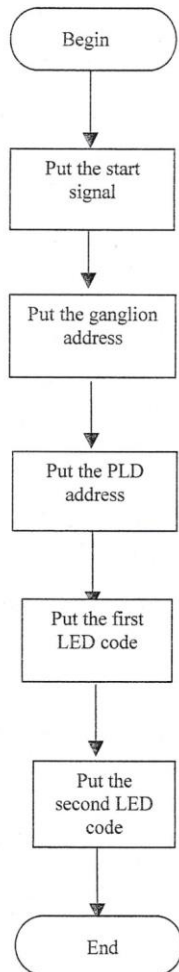
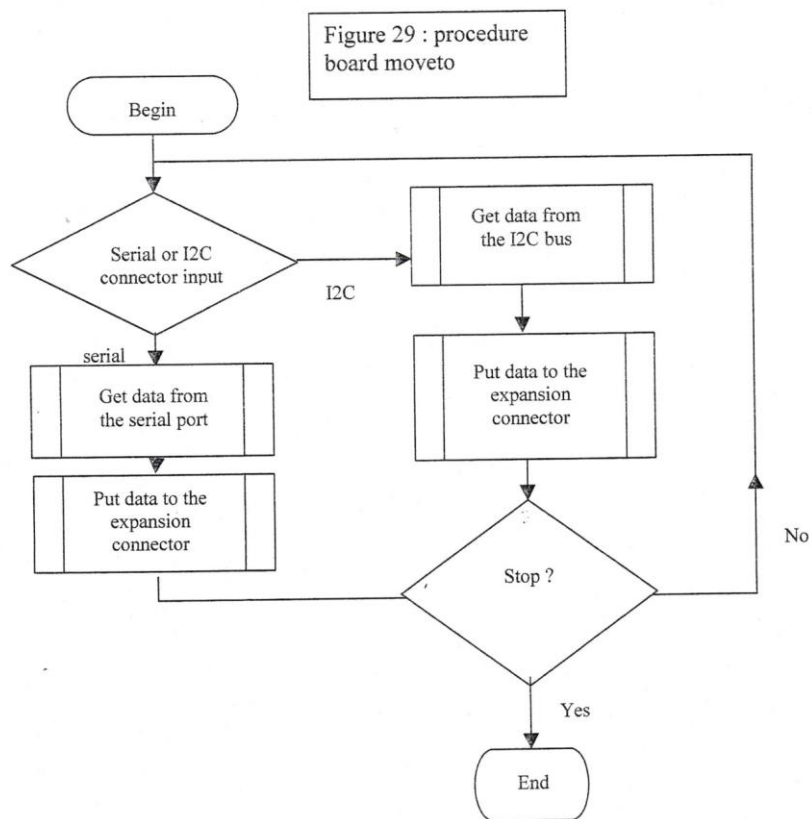


Figure 28 : sub procedure put data

The second program is moveto.hex
What this program does is receive a code from the serial or the I2C bus, and output on the EXR, address and data pins of the expansion connector the right signal. The program is written in assembler.

G) Schematic diagram for the board moveto.

The program recognize automatically if the input is serial or I2C. If it is serial, it launches the get data from serial port procedure and the put data to the expansion connector. If it is I2C bus, it launches the get data from I2C bus procedure and put data to the expansion connector.



H) Put data to the expansion connector

This is a sub-procedure of the procedure board moveto. When the data is received by the board, it is then put on the expansion connector. The first step is to put the EWR signal on the expansion connector (ALE on the diagram below). Then when the timing allows it, it puts the address A0 to A7 and then the data out on the port 0.

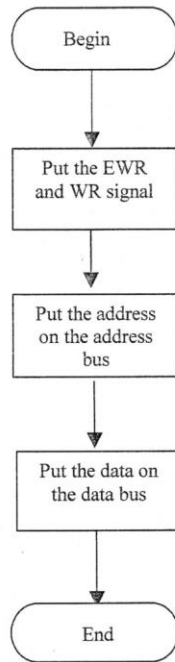


Figure 30 : sub procedure put data on the expansion connector

2) User interface software

This software is written in Visual Basic. There are several parts in this program. Basically, there are three parts :

- One part allows the user to build a new description of the wall on the computer and save it. This is the build wall characteristics part.
- An other part allows the user to modify the configuration of the wall and save it. This is the build configuration part
- An other part allows the user to build and modify a route and save it

The user interface is shown on next page

On the left are the six panels with the LEDs representing the wall.
On the right are the command buttons.

The user clicks on a color and then clicks on a LED, then the LED takes the desired color .

When the user clicks on the "change side" option button and then clicks on a panel, the side of the panel changes.

When the user clicks on the "modify route" button, the program modify the LED color on the real wall.

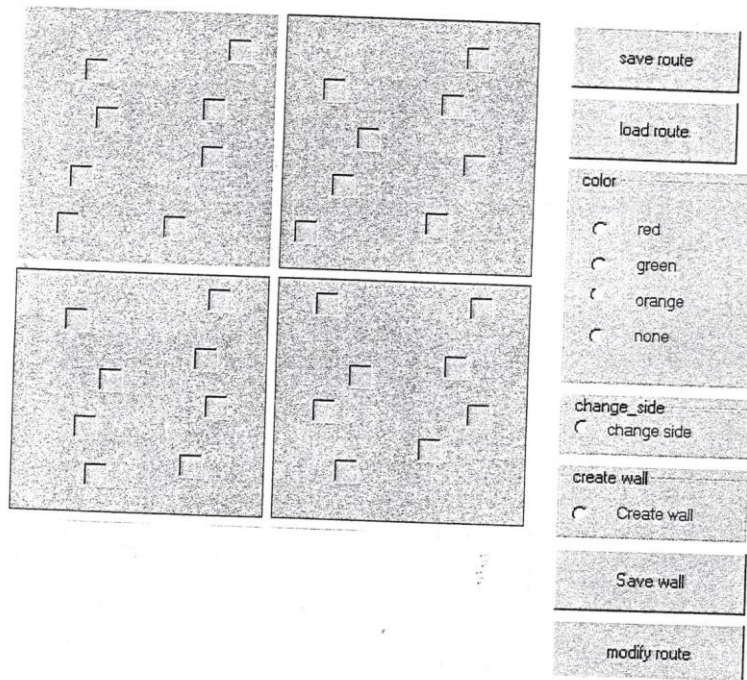
When the user clicks on the "save route" button, the route selected is saved under a name.

When the user clicks on the "save wall" button, the configuration is saved under a name.

When the user clicks on the "load route" button, a route with a path name should be loaded.

When the user clicks on the "show current route" button, the program should show the current route which is on the wall.

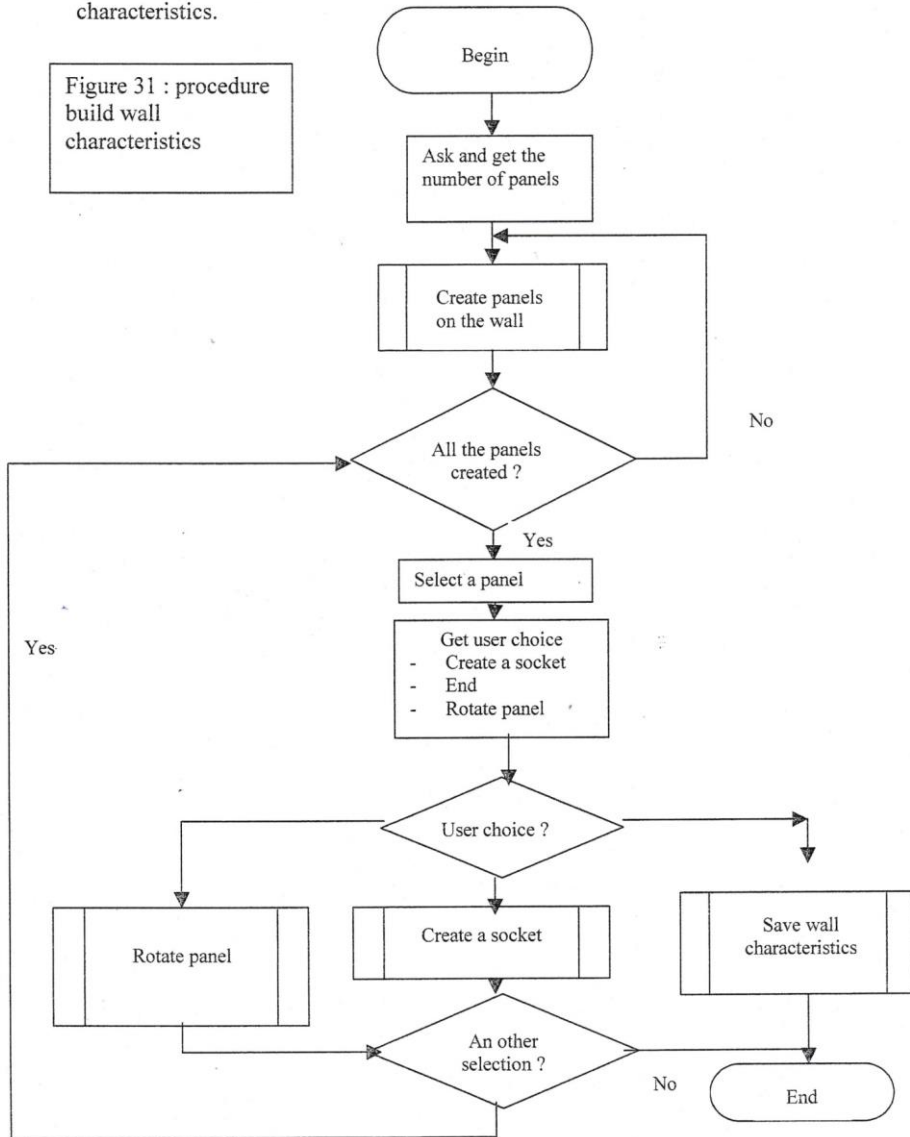
When the user clicks on the "create wall" option button, he should be able to modify the characteristics of the wall. It means that he can drag and drop the sockets to give it a new place for each side.



On the next pages are shown the schematics for the user interface procedures.

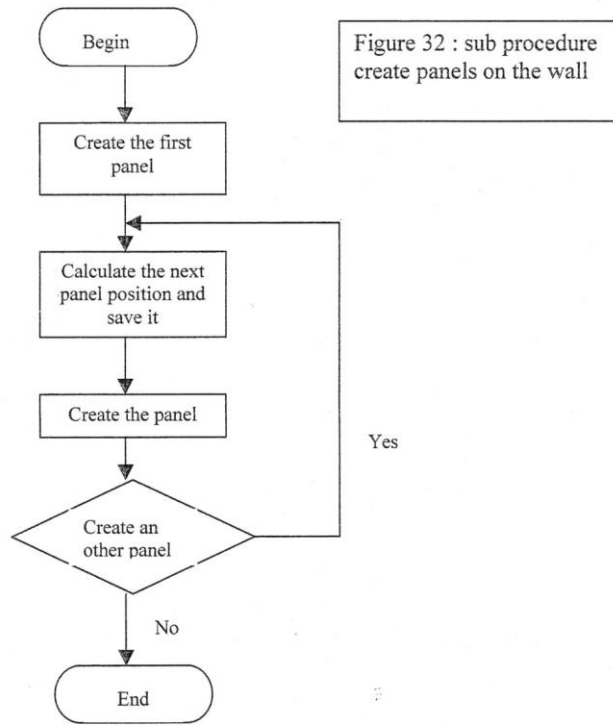
A) Procedure build wall characteristics

This procedure builds a new description of the wall, its panels and its sockets. It first asks the number of panels on the wall, and its width and its height. Then it draws the panels with the create panels on the wall procedure. Then the user can rotate panels, create sockets or quit and save the procedure with the sub-procedures rotate panels, create sockets and save characteristics.



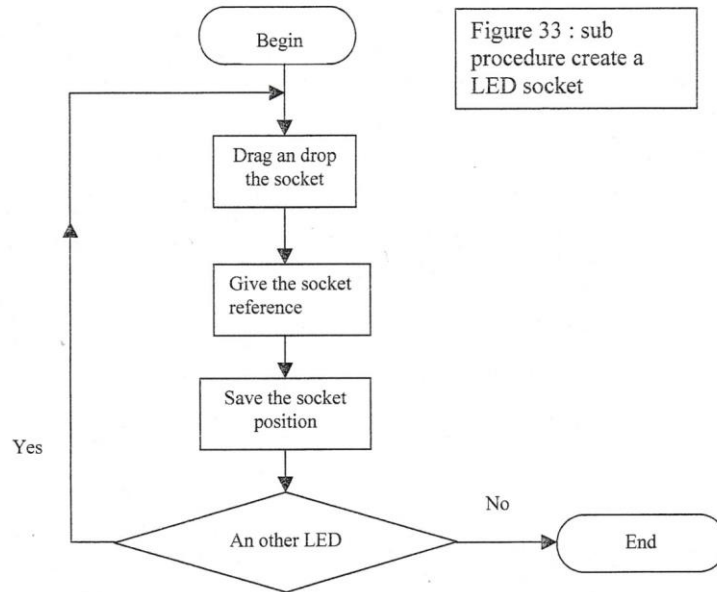
B) Create panels on the wall

This is a sub routine of the build wall characteristics procedure. It creates the first panel and then, by iteration, calculate the next panel coordinates and creates the next panel. It does it until all the panels are created.



C) Create a LED socket

The user drags and drops the LED socket to the position wanted. Then the computer gives the coordinates of the socket and saves the socket position. It is then ready for the next socket.



D) Rotate panel

This is a sub-procedure of the build wall characteristics procedure. When a panel is selected and the user wants to rotate the side, he clicks on the panel. It erases the side, gets the new side specifications and draws the new side. It then draws each socket.

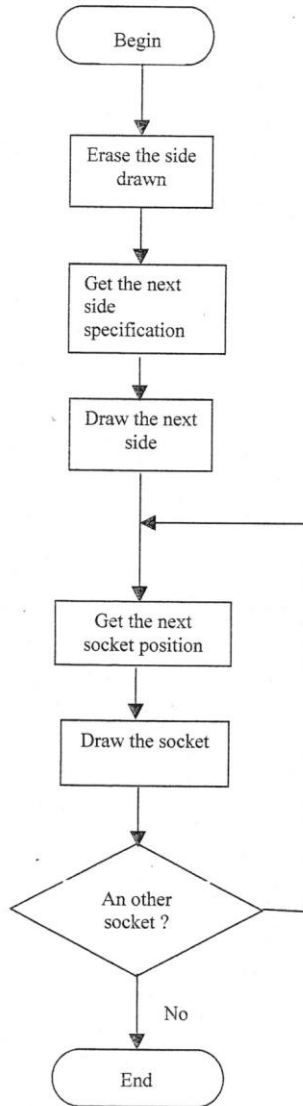
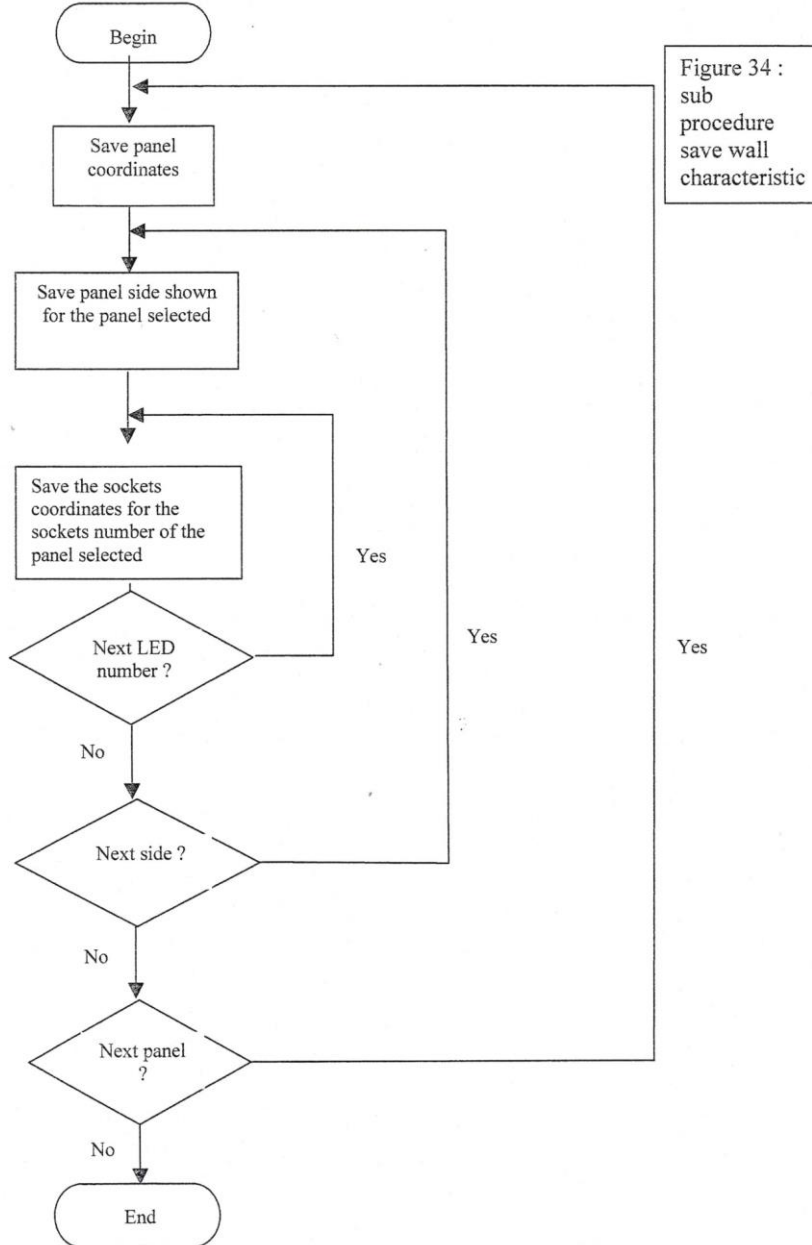


Figure 34 : sub procedure rotate panel

E) Save wall characteristics

This procedure, a sub procedure of the build wall characteristics procedure, saves the wall characteristics. It first saves the panels coordinates, then saves the side shown. For each socket of the panel, it saves the coordinates. It does it until all the panels characteristics have been saved



F) Build a configuration

This procedure allows the user to change the wall configuration. It means that the user can change the panel side.

This procedure draws the panels. If the user wants to change the panel side, he clicks on the panel and rotates it. If there is no other rotation, the computer saves the configuration.

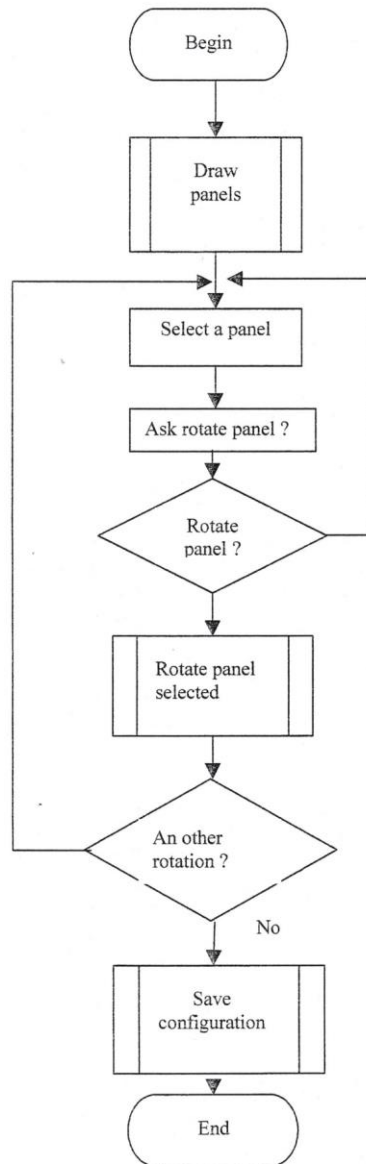


Figure 34 : procedure
build wall
configuration

G) Draw panels

This is a sub-procedure of the build configuration procedure. It creates the drawing of the panels with the sockets. It first creates the first panel, then calculates the next panel position and draws it. Then it draws the LEDs sockets for each panel.

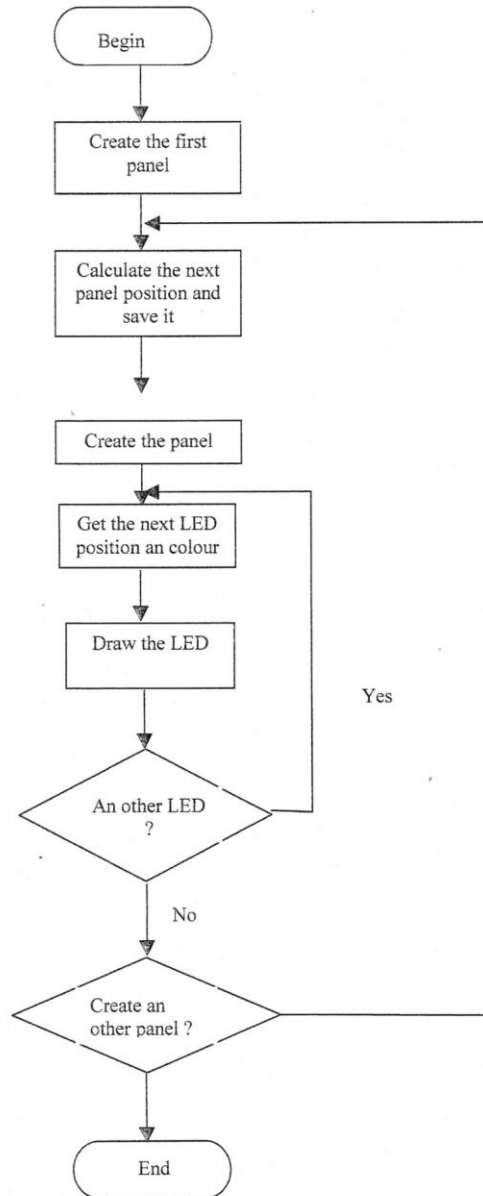


Figure 35 : sub procedure draw panels

H) Change panel side

This sub-procedure first erases the side drawn, then draws a new side with the specifications and the LEDs.

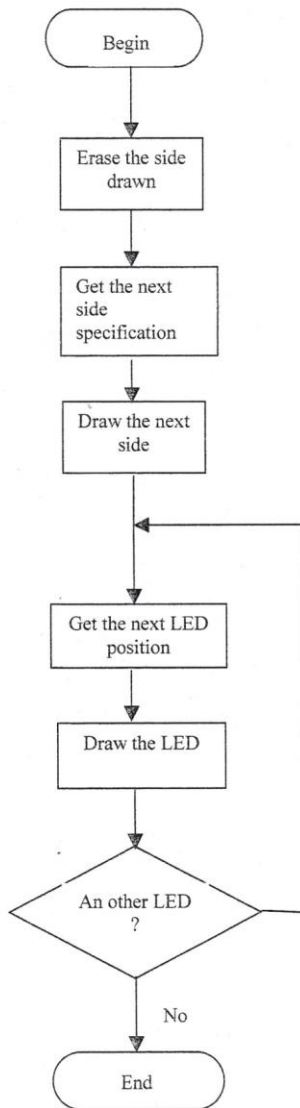
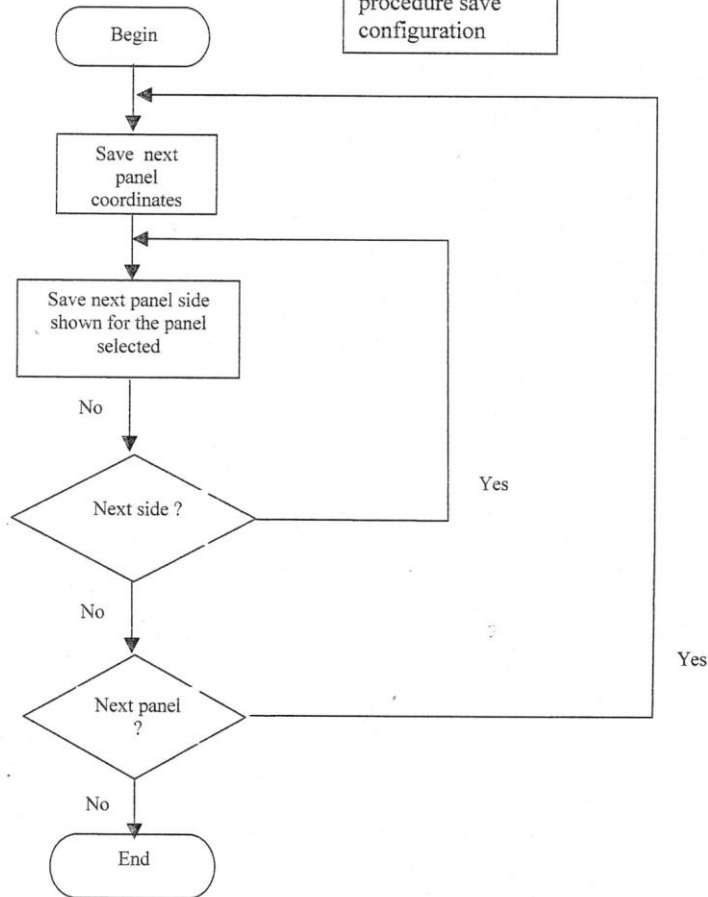


Figure 36 : sub procedure change panel side

1) Save configuration

This sub-procedure saves a configuration made by the user. It saves first the panel coordinates, then the side shown. It does it until every panel configuration is saved.

Figure 37 : sub procedure save configuration



J) Build a route

This procedure allows the user to build the route to show to the climber with LEDs. It first draws panels, then when a panel is selected and the configuration chosen, the user selects the LED color and the LEDs to switch on. Then it saves the route with the save route procedure.

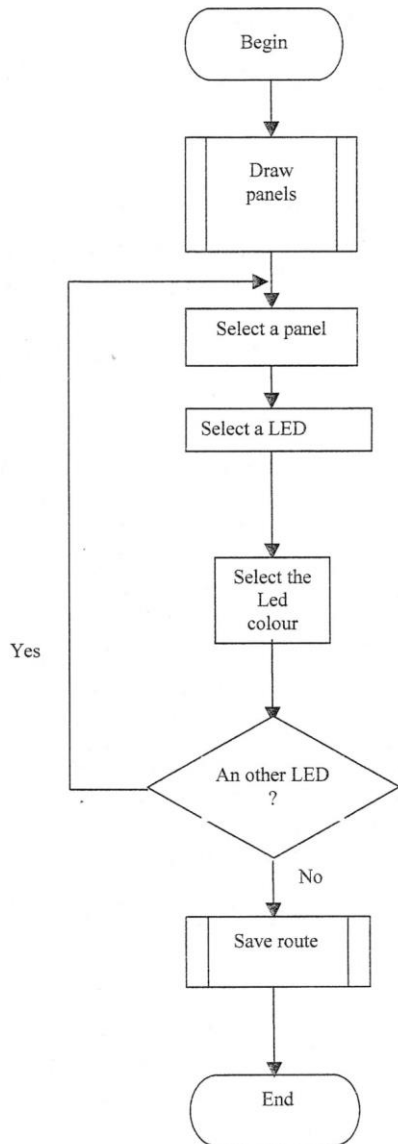
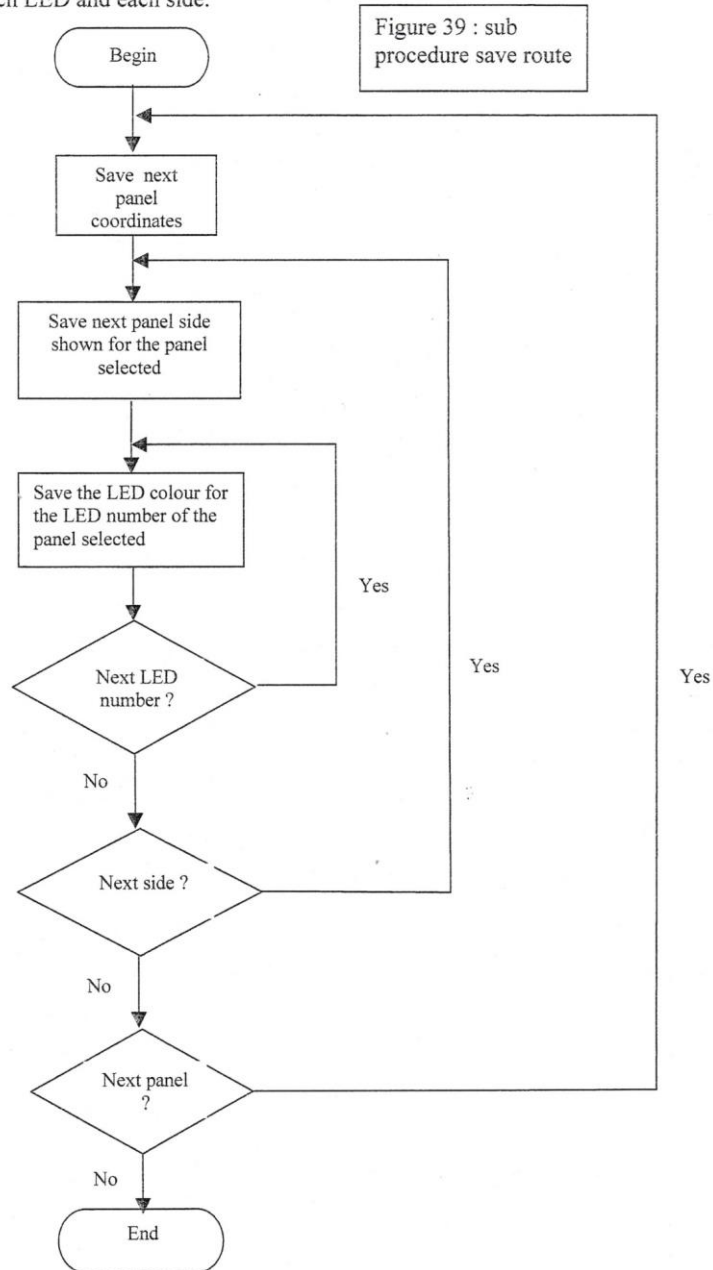


Figure 38 : procedure build a route

K) Save route

This sub-procedure saves the route. It first saves the panel coordinates, the side, the LED color for each LED and each side.



CHAPTER 6

Recommendations and conclusion

1) User instructions

A) Plug in the electronics

- 1) Plug in all the Led wires following the numbers and the colors on the screw terminal of the LED driver board.
- 2) Plug in the 9 wires D connection with the computer on the serial port 1.
- 3) Put the power supply to 6 volts and plug it.

B) Connect the hyperterminal

- 1) Open the hyperterminal.
- 2) Check the hyperterminal settings.
 - 2-1) Create a new hyperterminal.
 - Run hypertrm.exe file
 - Give the new name and the new icon and press OK
 - Give a telephone number and press OK
 - Press cancel for the new screen. The new hyper terminal is now created.
 - 2-2) Set the hyperterminal (to be done only one time).
 - In File, click properties
 - In properties, click the settings folder
 - Function is terminal key
 - Backroll buffer line is 500
 - Emulation is ANSI
 - Terminal setup
 - Block is no
 - Underline is yes
 - Blink is yes
 - Click OK
 - ASCII setup
 - Send line ends with line feeds is yes
 - Echo typed character locally is yes
 - Line delay is 0 ms
 - Character delay is 0 ms
 - Wrap line that exceed terminal width is yes
 - Click OK
 - In properties, click the phone number folder
 - Connect using Direct to COM1.
 - 2-3) Connect the board to the hyperterminal.
 - Click the menu call and then connect or click the telephone icon to connect the board to the hyperterminal

Switch the reset button of the ganglion board. A \$ should appear on the screen
Open the directory moveto.txt, select all and copy
In the hyperterminal, select edit and past to host. One the screen, you see the data transmitted and the following code

(B2BE)

:

type /8000 and then return

You should see the following code

@fu

The hyperterminal is now fully working. Type codes like P1FF, P0FF to test it.
All the lights should become orange.

You can control the LEDs from the hyperterminal by typing a code like
P+0 or 1+hexadecimal code+hexadecimal code+return

Examples : P0D2, P138, P13F, P0AC.

Be sure that the characters are capital letters.

C) Use the visual basic program

Open the Visual Basic 5.0 program

Open the project fred

Run the program

To build a route on the screen, click the option button of the color wanted, then click the LED icons you want to switch

When the route is built, change the panel configuration by clicking the change route button

2) Cost

A) Cost for the LED board

Parts list

Component	Function	Quantity	RS order code	price
74FCT573	Octal latch	2	300-754	1.6
GAL22V10B25LP	PLD	1	405-968	5
UDN	amplifier	2		6
Resistor	220 ohms	10		
LED tricolor		10		

The price without the sockets is 12.6 pounds

The price for the LED board is about 20 pounds, with sockets and LEDs.

B) Price for the ganglion board

Parts list

Parts list					
Component	Function	Quantity	RS order code	Price (pounds)	
80C552	microcontroller	1	310-701	14.08	
MAX232	RS232	1	225-8510	2	
74FCT573	Octal latch	1	300-754	0.8	
27C256	32K * 8 EPROM	1	297-636	4.77	
CXK5864	32K * 8 SRAM	1	311-007	3.29	
GAL22V10B25LP	PLD	1	405-968	5	
SPST switch	reset switch	1	183-701	0.44	
9 way D-connector	serial port	1	457-478	1.21	
IN4001 Diode	polarity protector	1	261-148	0.91	
LED	power indication	1			
2.1 mm socket	power in	1	486-628	0.31	
PCB mount power socket	power in	1	486-662		
Crystal 11.0592 MHz	XTAL	1	226-1768	1.08	
7805 LM 340TS	5V regulator	1		?	
socket 16 way, 0.3 inch	Max 232 socket	1	813-137		
socket 24 way, 0.3 inch	Octal latch socket	1	813-165		
socket, 20 way, 0.3 inch	PLD socket	1	813-159		
socket 28 way, 0.6 inch	RAM and EPROM socket	1	813-187		
BICC VERO 905 7218	expansion connector	1	1992-7726	5.26	
class3 socket	expansion connector	2	192-7524		
470R resistor		1			
330R resistor		2			
1K resistor		1			
10K resistor		2			
22pF capacitor		2			
100 nF capacitor		9			
22uF capacitor		1			
100uF capacitor		1			

The price for the ganglion board is 39.15 pounds without the sockets. With the sockets it is 45 pounds.

C) Price for the system

The price for the control of one panel would be 45 pounds.
For a prototype of six panels, it would be $45 + 45*6 + 20* 6=435$ pounds.

3) LEDs mount

Mounting the LEDs is a particular problem. Several solutions have been proposed. Advantages and disadvantages have been written next to each solution

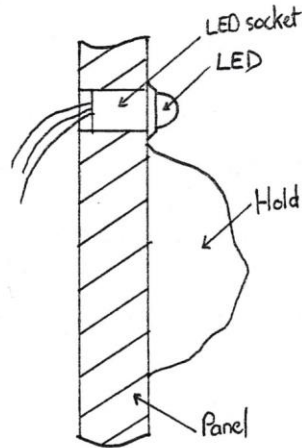
A) First solution

Advantage :

- simple
- LED easily fixed
- No problem when modifying the hold

Disadvantage :

- LED not on the hold
- Not easily seen by the user



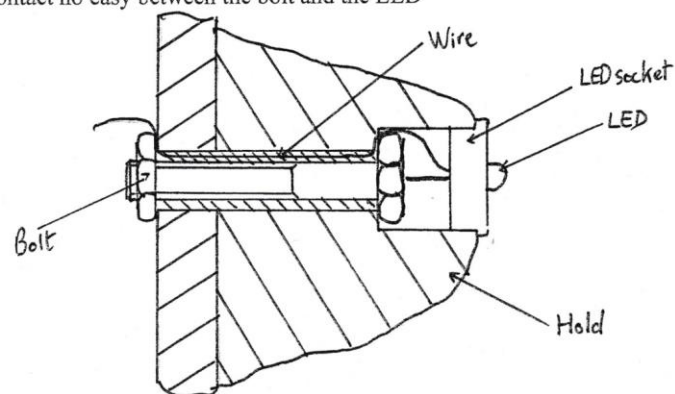
B) Second solution

Advantage :

- LED on the hold

Disadvantage :

- Problems to fix the LED
- Problems when re-orienting the LED
- Wire easily destroyed
- Contact no easy between the bolt and the LED



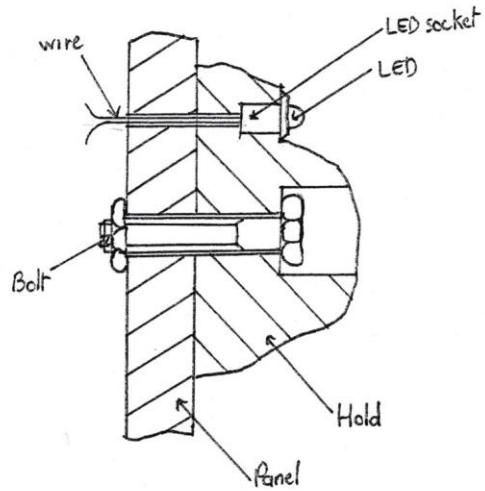
C) Third solution

Advantage :

- LED on the hold

Disadvantage :

- Problems to fix the LED
- Cannot re-orienter the LED



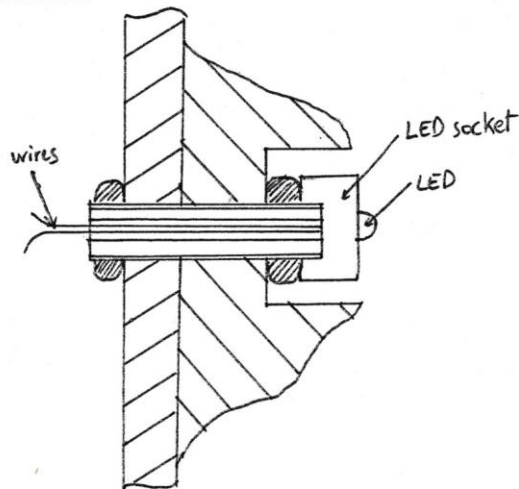
D) Fourth solution

Advantage :

- no problem when re-orienteering the hold
- no problem to fix the LED

Disadvantage :

- cost



E) 4) Recommendations for future work

A) Electronics

The work which remains for the electronics is the following : build the five boards and make it work in order to drive the LEDs of six panels with the boards and an I2C bus.

The I2C bus is extremely difficult to make work and is unreliable. An other solution could be proposed to drive the LEDs. Use only one ganglion board with a serial bus and six LEDs boards.

Advantage :

- avoid the using of the I2C bus
- decrease the price because five ganglion boards would not be used

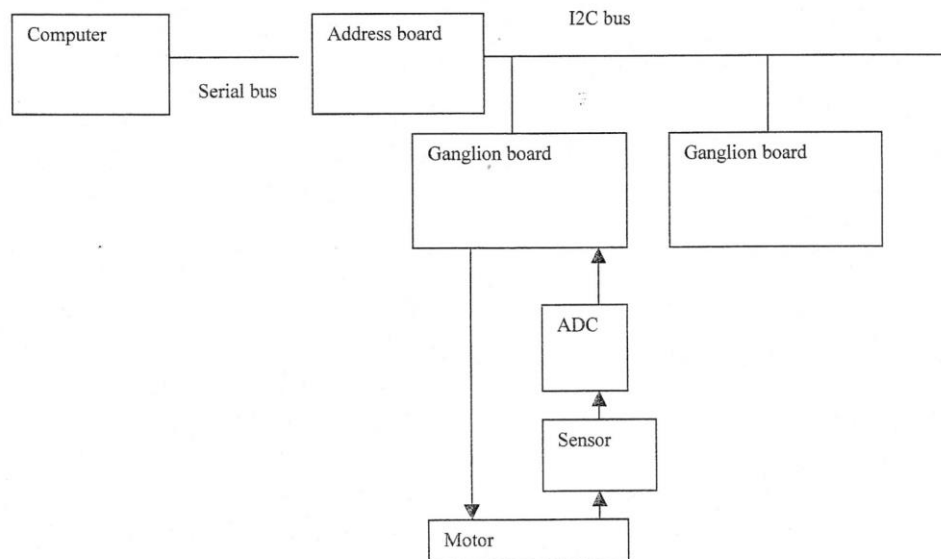
Disadvantages :

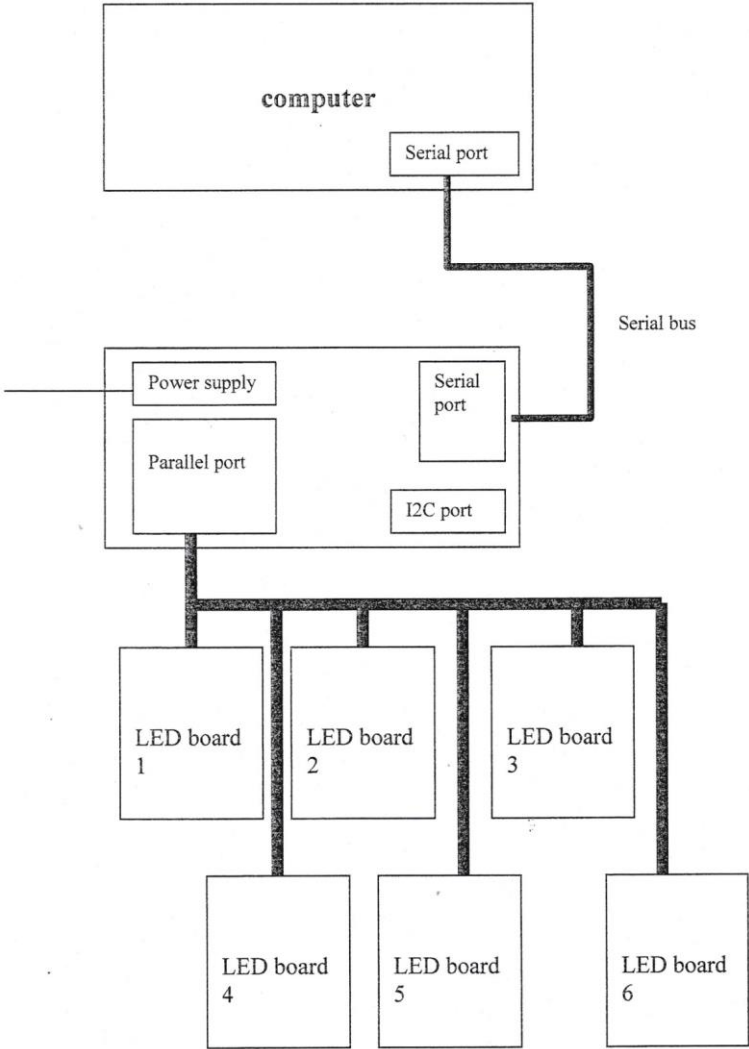
- does not allow further development, in particular develop the rotate hold process.

The schematic of the solution proposed is shown on next page.

The electronics should be finished after this work for the show route concept

The rotate hold concept will have then to be done. If the I2C bus is working, I propose a solution similar to the one used with the WARP project. The hold motor would be a stepper motor





B) Software

The main work will be to continue to develop the software to make it work with the six panels and integrate the program into the main program developed by the company. The functions which remain to be done are the following :

- Build wall characteristics
- Change configuration
- Save route and configuration

Then the software will have to be developed to do the rotate hold concept work

C) General recommendations

From this experience, I can give some recommendations to avoid future errors :

- At each step of the work, ask always yourself why you choose one solution rather than an other. It will avoid to start to do something unadapted.
- Have the work made by a technician as often as possible. Keep for yourself only the design and the report
- Give regular progress report to the supervisor and the company. It will avoid to misunderstand something.

CONCLUSION

During this project, I learnt a lot about electronics, software engineering, project management. With the electronics, I learnt how to design a PCB, program RAM and EPROM, test a module, make it work, etc... With the software, I learnt how to do conceptual design and software implementation.

The product show route concept is a mechatronic solution which involves many components of the mechatronic course : microcontrollers, control, software engineering, interfacing, etc ... It is a good complement to learn about mechatronic.

The original target was to finish the concept show route. It has not been finished, as shown in the recommendation for future work paragraph. A prototype allowing to control the route for one panel has been built. What remains to do is complete the show route concept for the prototype of six panels and then build the rotate panel prototype.

The software has not been finished. The part select route and show it on the wall has been finished. What remains to do is the part build wall characteristics, build a configuration, save route and configuration.

THANKS

Many thanks to :

Mr Derek Seward
Mr Bob Chaplain
Mr Jim Ward and the DR International Climbing Wall Company
Mr Jean-Pierre Trottignon
The technicians
Mr Sean Collins
Mr Pavli
Mr Philippe Joubert
Mr Moreau, Mr Prissard and his team
My parents and my sister Florence
Thoses who helped me and are not cited

For their cooperation and their understanding.

BIBLIOGRAPHY

W.A.R.P report by Sean Collins, Andrew Smith, Joseph Piperakis, Keith Garvey and Stephen Watts

80C552/83C552 Embedded Computer Board User manual & Technical Specifications by Sean Collins and Andrew Smith

The I2C bus and how to use it, an application note from Philips semiconductors

The 80C552 single-chip 8-bit microcontroller data sheet, an application note from Philips semiconductors

The MM74HCT573 Octal latch data sheet, an application note from National semiconductors

The GAL22V10 data sheet, an application note from Lattice

The Max220 data sheet, an application note from Maxim

The NMC87C257 EPROM data sheet, an application note from National Semiconductors

FIGURES

Figure 1	The configurable climbing wall
Figure 2	One panel
Figure 3	Way of rotating panels
Figure 4	Show route with LED
Figure 5	Rotate hold
Figure 6a	Bubble diagram
Figure 6b	Structure chart
Figure 7a	Context diagram
Figure 7b	Diagram 0
Figure 7c	Diagram 3 : use climbing wall
Figure 7d	Diagram 3.1 : change wall characteristics
Figure 7e	Diagram 3.2 : select route
Figure 7f	Diagram 3.3 : select wall configuration
Figure 7g	Diagram 3.4 : change route
Figure 7h	Diagram 3.5 : change configuration
Figure 8	Function means table for the configurable wall concept
Figure 9	Solution with I2C bus and LED driver
Figure 10a	Solution with I2C bus, PLD+latches
Figure 10b	Solution with serial bus, PLD, latches
Figure 11a	I2C bus
Figure 11b	Byte format for the I2C protocol

Figure 12	Schematic of the system
Figure 13	Schematic of the ganglion board
Figure 14	Memory organisation
Figure 15	Serial bus connexion
Figure 16	Parallel port connexion
Figure 17	Functionment of the microcontroller
Figure 18	Ganglion board layout
Figure 19	Schematic of the LED board
Figure 20	Parallel port connexion 2
Figure 21	PLD connexion
Figure 22a	LEDs board layout
Figure 22b	Rig schematic
Figure 23	Procedure board gateway
Figure 24	Sub procedure get data from the serial port
Figure 25	Sub procedure get data from the I2C port
Figure 26	Sub procedure get data
Figure 27	Sub procedure put data on the I2C bus
Figure 28	Sub procedure put data
Figure 29	Procedure board moveto
Figure 30	Sub procedure put data on the expansion connector
Figure 31	Procedure build wall characteristics
Figure 32	Sub procedure create panels on the wall
Figure 33	Sub procedure create a LED socket
Figure 34	Procedure build wall configuration
Figure 35	Sub procedure draw panels
Figure 36	Sub procedure change panel side
Figure 37	Sub procedure save configuration
Figure 38	Procedure build a route
Figure 39	Sub procedure save route

APPENDIX

INDEX

Appendix 1 : program moveto.hex

Appendix 2 : program gateway.hex

Appendix 3 : visual basic program

Appendix 4 : WARP.pds program

Appendix 5 : fredled.pds program

Appendix 6 : I2C bus data sheet

Appendix 7 : PLD Lattice GAL22V10 data sheet

Appendix 9 : microcontroller 80C552 data sheet

Appendix 10 : 80C552/83C552 embedded computer board user manual and technical options

Appendix 11 : board layout

Appendix 1 : program moveto.hex

```

;*****
;*
;* passthru.asm - accepts commands from serial port / I2c bus
;* and sets valve openings / returns ADC readings
;*
;* Sean Collins 15/5/98
;*
;*****
$MOD552
CSEG

CODE_START      EQU      8000h
ORG      CODE_START
      jmp      start
$INCLUDE(W_ISR.TAB.ASM)
#include(W_IIC_OS.asm)
$INCLUDE(w_serial.ASM)

BSEG
VC_GoodChar:    DBIT      1
serial_echo:    DBIT      1
foot_state:     DBIT      1
i2c_chan:       DBIT      1

DSEG
s_comm_ptr:     DS          01h
s_comm_buf:     ; serial command buffer
                DS          16h
s_comm_end:     ; end of buffer

CSEG

IIC_My_Addr:
                DB          04h

Valid_Char:     CJNE      A,#'0',AH1      ; Test for ASCII numbers.
AH1:            JC        VC_Bad          ; Is character is less than a '0'?
                CJNE      A,#'Z'+1,AH2    ; Test value range.
AH2:            JC        VC_Good        ; Is character is between '0' and
'9'?
VC_Bad:
                clr       VC_GoodChar
                ret
VC_Good:
                setb      VC_GoodChar
                ret

io_A_hex:       ; converts the lower nibble in A to hex
                anl      a, #0fh
                cjne     a, #0ah, ioa_1
ioa_1:
                jc       ioa_0_to_9
                add      a, #7
ioa_0_to_9:
                add      a, #'0'
                ret

io_Hex_to_A:
                mov      a, @r0          ;get first character of hex byte
                clr      r

```

```

        subb a, #'0'           ;convert ascii to number (ish)
        cjne a, #0ah, ioah_1
ioah_1:
        jc    ioah_0_to_9
        clr  c
        subb a, #07h           ;convert A-F to number
ioah_0_to_9:
        swap a
        mov  b, a
        inc  r0
        mov  a, @r0           ;get second character of hex byte
        clr  c
        subb a, #'0'           ;convert ascii to number (ish)
        cjne a, #0ah, ioah_2
ioah_2:
        jc    ioah_2_0_to_9
        clr  c
        subb a, #07h           ;convert A-F to number
ioah_2_0_to_9:
        orl  a, b
        ret

io_A_to_Hex:
        mov  b, a
        swap a
        anl  a, #0fh
        cjne a, #0ah, ioah_1
ioah_1:
        jc    ioah_0_to_9
        add  a, #7
ioah_0_to_9:
        add  a, #'0'
        movx @dptr, a
        inc  dptr
        mov  a, b
        anl  a, #0fh
        cjne a, #0ah, ioah_2
ioah_2:
        jc    ioah_0_to_9_2
        add  a, #7
ioah_0_to_9_2:
        add  a, #'0'
        movx @dptr, a
        mov  a, b
        ret

;*****
;*
;* Start here. Initialise Serial and I2c comms
;* reset valve openings
;*
;*****
start:
        acall Ser_init
        acall IIC_INIT
        clr  a
        mov  dptr, #ram_updown
        movx @dptr, a
        inc  dptr
        movx @dptr, a
        inc  dptr

```

```

        movx   @dptr, a
        call  s_comm_init

mainloop:
s_in:    ;serial input
;       mov  a, #'.'
;       call putChar
;       acall getch
;       jnb  SER_getch, s_no_serial
;       mov  b, a
;       mov  a, #'@'
;       call putChar
;       mov  b, a
;       clr  i2c_chan      ; serial channel active
;       jnb  serial_echo, s_no_echo
;       acall putChar
s_no_echo:
;       jmp  input_done
s_in_ok:
        call  s_comm_char      ;put char in command buffer
        cjne  a, #0ah, s_no_LF_yet      ;if last char LF, do command
        call  do_command
s_no_LF_yet:
        jmp  input_done
s_no_serial:
        jnb  IIC_Rx_waiting, no_iic
        setb i2c_chan      ; i2c chan active
        call IIC_command
        call do_command
        clr  IIC_Rx_Waiting

no_iic:
input_done:
;*****
;*
;* State loop here
;*
;*****
stm_0:
        mov  dptr, #gang_states      ; external devices
        movx a, @dptr      ; get the first channel state
        cjne a, #STATE_IDLE, stm_0_1
        sjmp stm_1      ; check the next channel
stm_0_1:
        ; do the state for this chan
        cjne a, #STATE_MOVE, stm_1 ; no other state than move implemented
        inc  dptr      ; address of moveto value
        movx a, @dptr      ; moveto value in r1
        mov  r1, a
        clr  a
        call ADC_get      ; get the ADC value on this channel
        mov  dptr, #0ff00h
        clr  c
        mov  r2, a      ; save a in r2
        subb a, r1
        jnz  stm_0_2
        movx @dptr, a      ; turn valve off
        sjmp stm_1
stm_0_2:
        jc  stm_0_3      ; jump if moveto > adc
        rl  a      ; gain is 2
        rl  a      ; gain is 2
        setb acc.0

```



```

        movx @dptr, a
        jmp  stm_1
stm_0_3:
        mov  a, r1      ; a is desired value
        subb a, r2     ; subtract ADC value
        rl   a         ; gain is 2
        rl   a         ; gain is 2
        movx @dptr, a

stm_1:
        mov  dptr, #gang_state_1 ; external devices
        movx a, @dptr           ; get the 1 channel state
        cjne a, #STATE_IDLE, stm_1_1
        sjmp  stm_2           ; check the next channel
stm_1_1:
        ; do the state for this chan
        cjne a, #STATE_MOVE, stm_2 ; no other state than move implemented
        inc  dptr             ; address of moveto value
        movx a, @dptr        ; moveto value in r1
        mov  r1, a
        inc  dptr             ; last control value address
        movx a, @dptr        ; get last control in a
        mov  r2, a           ; last control in r2
        mov  a, #1
        call ADC_get         ; get the ADC value on this channel
        call calc_Control    ; calculate the control output
        movx @dptr, a       ; save it for next time
        mov  dptr, #0ff01h
        movx @dptr, a
        jnz  stm_1_2
        mov  dptr, #gang_state_1
        mov  a, #STATE_IDLE
        movx @dptr, a
        sjmp  stm_2

stm_1_2:
;---debug below
;   xch  a, r2
;   mov  dptr, #em_debug_1
;   call io_A_to_Hex
;   mov  dptr, #em_debug
;   call ser_putMess
;---debug above

stm_2:
        mov  dptr, #gang_state_2
        movx a, @dptr           ; get the 2 channel state
        cjne a, #STATE_IDLE, stm_2_1
        sjmp  stm_end         ; check the next channel
stm_2_1:
        ; do the state for this chan
        cjne a, #STATE_MOVE, stm_end ; no other state than move implemented
        inc  dptr             ; address of moveto value
        movx a, @dptr        ; moveto value in r1
        mov  r1, a
        inc  dptr             ; last control value address
        movx a, @dptr        ; get last control in a
        mov  r2, a           ; last control in r2
        mov  a, #2
        call ADC_get         ; get the ADC value on this channel
        call calc_Control    ; calculate the control output
        movx @dptr, a       ; save it for next time
        mov  dptr, #0ff02h

```

```

        movx @dptr, a
        jnz stm_2_2
        mov dptr, #gang_state_2
        mov a, #STATE_IDLE
        movx @dptr, a
        sjmp stm_end
stm_2_2:
;   mov dptr, #em_debug_1
;   call io_A_to_Hex
;   mov dptr, #em_debug
;   call ser_putMess

stm_end:
        jb  foot_State, foot_1      ; last state 1?
        jnb P4.0, foot_done        ; last state 0, finish if switch at 0
        mov dptr, #em_foot_up; 0 to 1 transition
        call putMess
        setb foot_State
        jmp  foot_done
foot_1:
        jb  P4.0, foot_done        ; last state 1, finish if switch 1
        mov dptr, #em_foot_down    ; 1 to 0 transition
        call putMess
        clr  foot_State
foot_done:
        jmp  mainloop

calc_Control:
; have actual (ADC reading) in a, desired in r1, and last control value in
r2
; returns new control output in a
        clr  c
        mov  r3, a      ; save a in r3
        subb a, r1      ; a = actual - desired
        jz   Calc_Con_End ; if different, jump and do some more calcs
Calc_Con_1:
        jc   Calc_Con_3 ; jump if desired > actual (need to reverse
actuator)
; got difference, add it onto I if same direction, zero
; I otherwise! Currently reversing actuator, so value is odd
        xch  a, r2      ; last control in a, difference in r2
        jb  acc.0, Calc_Con_1a ; if it's odd, that's good
        clr  a          ; otherwise zero a
Calc_Con_1a:
        clr  c
        addc a, r2      ; add this difference to lastcontrol (Integral
action)
        jnc Calc_Con_1b ; if I overflows,
        mov  a, #0feh   ; set it to full on
Calc_Con_1b:
; a now holds I action, r2 holds difference
        xch  a, r2      ; difference back in a, r2 has I action
        clr  c          ; don't include carry in:
        mov  b, #2H     ; P gain calculation
        mul  ab         ; sets overflow if result > 255
        jnb OV, Calc_Con_2 ; < 255 so use it
        mov  a, #0feh   ; else set to 254 (even value = forward actuator)
Calc_Con_2:
        setb acc.0      ; reverse actuator
        jmp  Calc_Con_4
Calc_Con_3:
; last instruction was subb a, r1 (a = ADC reading
- desired)

```

```

        movx @dptr, a
        jnz stm_2_2
        mov  dptr, #gang_state_2
        mov  a, #STATE_IDLE
        movx @dptr, a
        sjmp stm_end
stm_2_2:
;      mov  dptr, #em_debug_1
;      call io_A_to_Hex
;      mov  dptr, #em_debug
;      call ser_putMess

stm_end:
        jb   foot_State, foot_1      ; last state 1?
        jnb  P4.0, foot_done         ; last state 0, finish if switch at 0
        mov  dptr, #em_foot_up ; 0 to 1 transition
        call putMess
        setb foot_State
        jmp  foot_done

foot_1:
        jb   P4.0, foot_done         ; last state 1, finish if switch 1
        mov  dptr, #em_foot_down ; 1 to 0 transition
        call putMess
        clr  foot_State

foot_done:
        jmp  mainloop

calc_Control:
; have actual (ADC reading) in a, desired in r1, and last control value in
r2
; returns new control output in a
        clr  c
        mov  r3, a      ; save a in r3
        subb a, r1      ; a = actual - desired
        jz   Calc_Con_End ; if different, jump and do some more calcs
Calc_Con_1:
        jc   Calc_Con_3 ; jump if desired > actual (need to reverse
actuator)
; got difference, add it onto I if same direction , zero
; I otherwise! Currently reversing actuator, so value is odd
        xch  a, r2      ; last control in a, difference in r2
        jb  acc.0, Calc_Con_1a ; if it's odd, that's good
        clr  a          ; otherwise zero a
Calc_Con_1a:
        clr  c
        addc a, r2      ; add this difference to lastcontrol (Integral
action)
        jnc  Calc_Con_1b ; if I overflows,
        mov  a, #0feh   ; set it to full on
Calc_Con_1b:
; a now holds I action, r2 holds difference
        xch  a, r2      ; difference back in a, r2 has I action
        clr  c          ; don't include carry in:
        mov  b, #2H     ; P gain calculation
        mul  ab         ; sets overflow if result > 255
        jnb  OV, Calc_Con_2 ; < 255 so use it
        mov  a, #0feh   ; else set to 254 (even value = forward actuator)
Calc_Con_2:
        setb acc.0      ; reverse actuator
        jmp  Calc_Con_4
Calc_Con_3:
; last instruction was subb a, r1 (a = ADC reading
- desired)

```

```

        mov a, r1      ; a is desired value
        clr c
        subb a, r3     ; subtract ADC value
; got difference, add it onto I if same direction , zero
; I otherwise! Currently reversing actuator, so value is odd
        xch a, r2     ; last control in a, difference in r2
        jnb acc.0, Calc_Con_3a ; if it's even, that's good
        clr a         ; otherwise zero a
Calc_Con_3a:
        clr c
        addc a, r2     ; add this difference to lastcontrol (Integral
action)
        jnc Calc_Con_3b
        mov a, #0feh
Calc_Con_3b:
        ; a now holds I action,r2 holds difference
        xch a, r2     ; difference back in a, r2 has I action
        mov b, #2H
        mul ab
        jnb OV, Calc_Con_3c
        mov a, #0feh
calc_Con_3c:
        clr acc.0     ; reverse direction
calc_Con_4:
        ; a now holds P action, r2 has I action
        xch a, r2     ; make I action even, so direction is set
        clr acc.0
        xch a, r2     ; by P action
        clr c
        addc a, r2     ; P + I
        jnc calc_Con_End ; jmp if P + I < 255
        orl a, #0feh   ; set upper 7 bits to 1s
calc_Con_End:
        mov r2, a
        ret

IIC_command:
        call s_comm_init;
        mov r1, #Slave_in
IIC_Command_0:
        mov a, @r1
        cjne a, #0ah, IIC_command_1
        sjmp IIC_Command_2
IIC_command_1:
        call s_comm_char
        inc r1
        sjmp IIC_command_0
IIC_command_2:
        ret

; mov dptr, #em_debug_4x
; mov a, Slave_in
; movx @dptr, a
; mov a, Slave_in + 1
; inc dptr
; movx @dptr, a
; mov a, Slave_in + 2
; inc dptr
; movx @dptr, a
; mov a, Slave_in + 3
; inc dptr
; movx @dptr, a
; mov dptr, #em_debug_4x

```

```

;   call SER_putMess
;   ret
s_comm_init:
    mov     s_comm_ptr, #s_comm_buf
    mov     s_comm_buf, #0dh
    ret

s_comm_char:
;   call putChar
    mov     r0, s_comm_ptr
    mov     @r0, a
    inc     r0
    cjne   r0, #s_comm_end, s_comm_char_end
    mov     dptr, #em_s_comm_overflow
    call    putMess
    mov     r0, #s_comm_buf
s_comm_char_end:
    mov     s_comm_ptr, r0
    ret

do_command:
    mov     a, s_comm_buf
    cjne   a, #'G', s_not_G      ;if it's not Get, try another
    mov     a, s_comm_buf + 1    ; get ADC channel to read
    mov     dptr, #em_ADC_cv_X0  ; put ADC channel number
    movx   @dptr, a              ; into message
    clr     c
    subb   a, #'0'               ; convert ASCII to number
    call   ADC_get               ; get value of ADC channel
    mov     dptr, #em_ADC_cv_XX1 ;
    call   io_A_to_Hex           ; put it in message
    mov     dptr, #em_ADC_chanval ; and print it
    call   PutMess
    jmp    do_command_end

s_not_G:
    cjne   a, #'P', s_not_P
    mov     a, s_comm_buf + 1    ; get DAC channel to set
    clr     c
    subb   a, #'0'               ; ASCII to number
    mov     dptr, #0ff00h        ; external devices
    add    a, dpl                ; no chance of carry!
    mov     dpl, a
    mov     r0, #s_comm_buf + 2
    call   IO_Hex_to_A
    movx   @dptr, a              ; write byte to DAC
    jmp    do_command_end

s_not_P:
    cjne   a, #'M', s_not_M
    mov     a, s_comm_buf + 1    ; get channel to move
    clr     c
    subb   a, #'0'               ; ASCII to number
    mov     b, #gang_state_size
    mul    ab
    mov     dptr, #gang_states   ; external devices
    add    a, dpl                ; no chance of carry!
    mov     dpl, a
    mov     a, #STATE_MOVE
    movx   @dptr, a
    mov     r0, #s_comm_buf + 2
    call   IO_Hex_to_A
    inc    dptr                  ; moveto byte is next after state

```

```

        movx @dptr, a          ; write byte to
        inc  dptr             ; last control is next again
        clr  a                ; must be zero
        movx @dptr, a
        jmp  do_command_end

s_not_M:
        cjne a, #'Q', s_not_Q ; is it Q?
        call init_all
        mov  dptr, #em_quit
        call putMess
        jmp  do_command_end

s_not_Q:
        cjne a, #'F', s_not_F ; is it F?
        mov  dptr, #em_foot_X ; ready to change msg
        jb   foot_state, s_F_1 ; is switch 1?
        mov  a, #'0'          ; foot switch is 0
        jmp  s_F_f            ; finished

s_F_1:
        mov  a, #'1'          ; foot switch is 1

s_F_f:
        movx @dptr, a
        mov  dptr, #em_foot_msg ; print it
        call putMess
        jmp  do_command_end

s_not_F:
        cjne a, #'W', s_not_W
        mov  dptr, #em_who_XX
        mov  a, SIADR
        call io_A_to_Hex
        mov  dptr, #em_who
        call putMess
        jmp  do_command_end

s_not_W:
        cjne a, #'*', s_not_star
        jmp  0000h                ;reset!

s_not_star:
        cjne a, #'{', s_its_shite
        cpl  serial_echo
        jmp  do_command_end

s_its_shite:
        mov  dptr, #em_what
        call putMess

do_command_end:
        call s_comm_init
        ret

ADC_get:
ADCS EQU 08h          ; start bit
ADCI EQU 010h        ; finished bit
        anl  a, #07h
        mov  ADCON, a
        add  a, #ADCS
        mov  ADCON, a

ADC_wait:
        cpl  p1.0
        mov  a, ADCON
        jb   acc.4, ADC_done
        jmp  ADC_wait

ADC_done:
        mov  a, ADCH
        ret

```

```

putMess:
    jb    i2c_chan, putMess_i2c
    call  SER_putMess
    ret
putMess_i2c:
    clr   i2c_chan
i2c_mess:
    mov   r0, #IIC_data_buffer
    mov   a, S1ADR
    call  IO_A_Hex
    mov   @r0, a
    inc   r0
i2c_mess_0:
    movx  a, @dptr
    jz    i2c_mess_1
    mov   @r0, a
    inc   r0
    inc   dptr
    sjmp  i2c_mess_0
i2c_mess_1:
    mov   aux_adrs, #02h    ;gateway address
    call  WI_Send
    ret

;*****
;*
;* WI_Send. Set aux_adrs to the slave address, fill IIC_data_buffer
;* (4 bytes). Call WI_Send
;*
;*****
WI_Send:
    mov   dptr, #WIS_file
    call  do_iic
    jbc   IIC_failure, WI_Send_error
    ret
WI_Send_Error:
    mov   dptr, #em_WI_SEND
    call  SER_putMess
    ret

em_Wi_send:
    DB    'IIC send error', 0ah, 0dh, 00h

WIS_file:
    DB    use_aux_adrs_      ; slave address in 'aux_adrs'
    DB    ioBuffer_
    DB    08h
    DB    iic_end_

init_all:
    mov   a, #0             ; 0th chan
    call  init_chan
    mov   a, #1             ; 1st chan
    call  init_chan
    mov   a, #2             ; 2nd chan
    call  init_chan
    ret

init_chan:

```

```

mov b, a ; save a in b
mov dptr, #0ff00h ; start of external devices
add a, dpl ; add offset in a
mov dpl, a ; move back to pointer
mov a, 00h
movx @dptr, a ; write a 0 on the pointer
mov a, b ; restore a
mov b, #gang_state_size ; state block size
mul ab ; 'a'th block offset
mov dptr, #gang_states ; external devices
add a, dpl ; add pointer to offset
mov dpl, a ; restore in pointer
mov a, #STATE_IDLE
movx @dptr, a ; update state info
ret

em_who:
DB 'w='
em_who_XX:
DB 'XX', 0dh, 0ah, 00h
em_quit:
DB 'Quit', 0dh, 0ah, 00h
em_foot_up:
DB 'fu', 0dh, 0ah, 00h
em_foot_down:
DB 'fd', 0dh, 0ah, 00h
em_foot_msg:
DB 'f='
em_foot_X:
DB 'X', 0dh, 0ah, 00h
em_debug:
DB 0dh, 0ah, 'Debug('
em_debug_1:
DB 'XX)', 0dh, 0ah, 00h
em_debug_4x:
DB 'XXXX!', 0dh, 0ah, 00h
em_prompt:
DB '> ', 00h
em_ADC_chanval:
DB 'g'
em_ADC_cv_X0:
DB 'X='
em_ADC_cv_XX1:
DB 'XX', 0dh, 0ah, 00h

em_s_comm_overflow:
DB 'CBO'
DB 0dh, 0ah, 00h
em_what:
DB '?', 0DH, 0AH, 00h

STATE_IDLE EQU 00h
STATE_MOVE EQU 01h

CSEG AT 0fe00h ; 256 bytes before external devices
gang_states:
gang_state_0:
DB 00h
gang_moveto_0:
DB 00h
gang_control_0:

```



```
        DB    00h
gang_state_size EQU $ - gang_states
gang_state_1:
        DB    00h
gang_moveto_1:
        DB    00h
gang_control_1:
        DB    00h
gang_state_2:
        DB    00h
gang_moveto_2:
        DB    00h
gang_control_2:
        DB    00h
```

```
RAM_UPDOWN      EQU    0ff00h
RAM_FOREBACK    EQU    0ff01h
RAM_LEFTRIGHT   EQU    0ff02h
```

```
END
```

2) Appendix 2 : program gateway .hex

```

$MOD552
CODE_START      EQU      8000h
ORG      CODE_START
      jmp      start
$INCLUDE(W_ISR TAB.ASM)
$include(W_IIC_OS.asm)
$INCLUDE(w_serial.ASM)

DSEG
s_comm_ptr:
      DS      01h
s_comm_buf:      ; serial command buffer
      DS      16h
s_comm_end:      ; end of buffer

CSEG

IIC_My_Addr:
      DB      02h

A_Hex_Dec:
      clr     c
      subb   a, #'0'           ;convert ascii to number (ish)
      cjne  a, #0ah, ahd_1
ahd_1:
      jc     ahd_0_to_9
      clr     c
      subb   a, #07h           ;convert A-F to number
ahd_0_to_9:
      ret

;*****
;*
;* WI_Send. Set aux_adrs to the slave address, fill IIC_data_buffer
;* (4 bytes). Call WI_Send
;*
;*****
WI_Send:
      mov    dptr, #WIS_file
      call  do_iic
;      jbc   IIC_failure, WI_Send_error
      ret
WI_Send_Error:
      mov    dptr, #em_WI_SEND
      call  SER_putMess
      ret

em_Wi_send:
      DB    'IIC send error', 0ah, 0dh, 00h

WIS_file:
      DB    use_aux_adrs_           ; slave address in 'aux_adrs'
      DB    ioBuffer_
      DB    08h
      DB    iic_end_

;*****
;*
;* Start here. Initialise Serial and I2c comms
;* reset valve openings
;*

```

```

;*****
start:
    acall  Ser_init
    acall  IIC_INIT
    call   s_comm_init

mainloop:
s_in:    ;serial input
    acall  getch
    jnb    SER_getch, s_no_serial
;    jnb    serial_echo, s_no_echo
;    acall  putChar
;s_no_echo:
;    jmp    input_done
s_in_ok:
    call   s_comm_char    ;put char in command buffer
    cjne   a, #0ah, s_no_LF_yet    ;if last char LF, do command
    call   ser_to_IIC
s_no_LF_yet:
    jmp    input_done
s_no_serial:
    jnb    IIC_Rx_waiting, no_iic
    clr    IIC_Rx_waiting
    call   IIC_to_ser
no_iic:
input_done:
;*****
;*
;* State loop here
;*
;*****
    jmp    mainloop

s_comm_init:
    mov    s_comm_ptr, #s_comm_buf
    mov    s_comm_buf, #0dh
    ret

s_comm_char:
    mov    r0, s_comm_ptr
    mov    @r0, a
    inc    r0
    cjne   r0, #s_comm_end, s_comm_char_end
    mov    dptr, #em_s_comm_overflow
    call   SER_putMess
    mov    r0, #s_comm_buf
s_comm_char_end:
    mov    s_comm_ptr, r0
    ret

ser_to_IIC:
    xch    a, r0
    push  acc
    xch    a, r1
    push  acc
    xch    a, r0
    mov    a, s_comm_buf    ; SHOULD BE SLAVE ADDRESS
    call   a_Hex_Dec        ; hex - dec conversion fix it!
    mov    aux_adrs, a      ; set up slave address
    mov    r0, #s_comm_buf + 1 ; start of data
    mov    r1, #IIC_data_buffer ; start of send buffer

```

```

ser_to_IIC_0:
    mov     a, @r0                ; get a character
    cjne   a, #0ah, ser_to_IIC_1 ; until LF
    sjmp   ser_to_IIC_2
ser_to_IIC_1:
    ; call  putchar
    mov    @r1, a                ; put in send buffer
    inc   r0
    inc   r1
    sjmp  ser_to_IIC_0
ser_to_IIC_2:
    mov    @r1, a
    call  WI_Send
    call  s_comm_init
    pop   acc
    xch   a, r1
    pop   acc
    xch   a, r0
    ret

IIC_to_ser:
    xch   a, r0
    push  acc
    xch   a, r0
    mov   r0, #Slave_in
IIC_to_ser_0:
    mov   a, @r0
    cjne a, #0ah, IIC_to_ser_1
    sjmp IIC_to_ser_2
IIC_to_ser_1:
    call  putchar
    inc   r0
    sjmp IIC_to_ser_0
IIC_to_ser_2:
    call  putchar
    pop   acc
    xch   a, r0
    ret

em_s_comm_overflow:
    DB    'Serial command buffer overflow'
    DB    0dh, 0ah, 00h

END

```

3) Appendix 3 : Visual basic program

change_side - 1

```
Private Sub select_color(pic As PictureBox)

If Option1.Value = True Then
pic.Picture = LoadPicture("c:\fred\fred1.bmp")
panel(1).side(1).Color = 1
End If

If Option2.Value = True Then
pic.Picture = LoadPicture("c:\fred\fred2.bmp")
panel(1).side(1).Color = 10
End If

If Option3.Value = True Then
pic.Picture = LoadPicture("c:\fred\fred3.bmp")
panel(1).side(1).Color = 11
End If

If Option4.Value = True Then
pic.Picture = LoadPicture("c:\fred\fred4.bmp")
panel(1).side(1).Color = 0
End If

End Sub

Private Sub Command2_Click()

End Sub

Private Sub Command1_Click()

End Sub

Private Sub Command3_Click()
Open "c:\fred\currentwall.txt" For Input As #2
Input #1, panel(1).side(1).ledcolor(1), panel(1).side(1).ledcolor(2), panel(1).side(1).ledcolor(3), panel(1).side(1).ledcolor(4)
Close #1

End Sub

Private Sub Command5_Click()
Dim b As Integer
Dim s As Integer
Dim DATA1 As String
Dim DATA2 As String
Call calcul_ledcode
Open "c:\fred\currentwall.txt" For Output As #1

Write #1, panel(1).side(1).ledcolor(1), panel(1).side(1).ledcolor(2), panel(1).side(1).ledcolor(3), panel(1).side(1).ledcolor(4)
Close #1
DATA1 = "P" & "0" & panel(1).side(1).hexcode(2) & panel(1).side(1).hexcode(1)
DATA2 = "P" & "1" & panel(1).side(1).hexcode(4) & panel(1).side(1).hexcode(3)
MsgBox "P" & "0" & panel(1).side(1).hexcode(2) & panel(1).side(1).hexcode(1)
MsgBox "P" & "1" & panel(1).side(1).hexcode(4) & panel(1).side(1).hexcode(3)
MSCComm1.Output = DATA1 & Chr$(13) & Chr$(10)
Timer1.Interval = 1
MSCComm1.Output = DATA2 & Chr$(13) & Chr$(10)
End Sub

Private Sub Form_Load()

End Sub

Private Sub Picture1_Click(Index As Integer)
Call select_color(Picture1(Index))
panel(1).side(1).ledcolor(1) = panel(1).side(1).Color
```

change_side - 2

```
End Sub

Private Sub Picture2_Click(Index As Integer)
Call select_color(Picture2(Index))
panel(1).side(1).ledcolor(2) = panel(1).side(1).Color
End Sub

Private Sub Picture3_Click(Index As Integer)
Call select_color(Picture3(Index))
panel(1).side(1).ledcolor(3) = panel(1).side(1).Color
End Sub

Private Sub Picture4_Click(Index As Integer)
Call select_color(Picture4(Index))
panel(1).side(1).ledcolor(4) = panel(1).side(1).Color
End Sub

Private Sub Picture5_Click(Index As Integer)
Call select_color(Picture5(Index))
panel(1).side(1).ledcolor(5) = panel(1).side(1).Color
End Sub

Private Sub Picture6_Click(Index As Integer)
Call select_color(Picture6(Index))
panel(1).side(1).ledcolor(6) = panel(1).side(1).Color
End Sub

Public Sub Picture7_Click(Index As Integer)
Call select_color(Picture7(Index))
panel(1).side(1).ledcolor(7) = panel(1).side(1).Color
End Sub

Public Sub Picture8_Click(Index As Integer)
Call select_color(Picture8(Index))
panel(1).side(1).ledcolor(8) = panel(1).side(1).Color
End Sub

Private Sub calcul_ledcode()
Dim i As Integer

panel(1).side(1).deccode(1) = (panel(1).side(1).ledcolor(2) * 100 + panel(1).side(1).led
color(1))
panel(1).side(1).deccode(2) = (panel(1).side(1).ledcolor(4) * 100 + panel(1).side(1).led
color(3))
panel(1).side(1).deccode(3) = (panel(1).side(1).ledcolor(6) * 100 + panel(1).side(1).led
color(5))
panel(1).side(1).deccode(4) = (panel(1).side(1).ledcolor(8) * 100 + panel(1).side(1).led
color(7))

For i = 1 To 4
Select Case panel(1).side(1).deccode(i)
Case Is = 0
panel(1).side(1).hexcode(i) = "0"
Case Is = 1
panel(1).side(1).hexcode(i) = "1"
Case Is = 10
panel(1).side(1).hexcode(i) = "2"
Case Is = 11
panel(1).side(1).hexcode(i) = "3"
Case Is = 100
panel(1).side(1).hexcode(i) = "4"
Case Is = 101
panel(1).side(1).hexcode(i) = "5"
Case Is = 110
panel(1).side(1).hexcode(i) = "6"
Case Is = 111
panel(1).side(1).hexcode(i) = "7"
Case Is = 1000
panel(1).side(1).hexcode(i) = "8"
```


change_side - 4

```
Case Is = 10
Picture4(Index).Picture = LoadPicture("c:fred.fred2.bmp")
Case Is = 11
Picture4(Index).Picture = LoadPicture("c:fred.fred3.bmp")
Case Is = 0
Picture4(Index).Picture = LoadPicture("c:fred.fred4.bmp")
End Select

Picture5(Index).Left = panel(Index).side(Current_side).LedX(5)
Picture5(Index).Top = panel(Index).side(Current_side).LedY(5)
Select Case panel(Index).side(Current_side).ledcolor(5)
Case Is = 1
Picture5(Index).Picture = LoadPicture("c:fred.fred1.bmp")
Case Is = 10
Picture5(Index).Picture = LoadPicture("c:fred.fred2.bmp")
Case Is = 11
Picture5(Index).Picture = LoadPicture("c:fred.fred3.bmp")
Case Is = 0
Picture5(Index).Picture = LoadPicture("c:fred.fred4.bmp")
End Select

Picture6(Index).Left = panel(Index).side(Current_side).LedX(6)
Picture6(Index).Top = panel(Index).side(Current_side).LedY(6)
Select Case panel(Index).side(Current_side).ledcolor(6)
Case Is = 1
Picture6(Index).Picture = LoadPicture("c:fred.fred1.bmp")
Case Is = 10
Picture6(Index).Picture = LoadPicture("c:fred.fred2.bmp")
Case Is = 11
Picture6(Index).Picture = LoadPicture("c:fred.fred3.bmp")
Case Is = 0
Picture6(Index).Picture = LoadPicture("c:fred.fred4.bmp")
End Select

Picture7(Index).Left = panel(Index).side(Current_side).LedX(7)
Picture7(Index).Top = panel(Index).side(Current_side).LedY(7)
Select Case panel(Index).side(Current_side).ledcolor(7)
Case Is = 1
Picture7(Index).Picture = LoadPicture("c:fred.fred1.bmp")
Case Is = 10
Picture7(Index).Picture = LoadPicture("c:fred.fred2.bmp")
Case Is = 11
Picture7(Index).Picture = LoadPicture("c:fred.fred3.bmp")
Case Is = 0
Picture7(Index).Picture = LoadPicture("c:fred.fred4.bmp")
End Select

Picture8(Index).Left = panel(Index).side(Current_side).LedX(8)
Picture8(Index).Top = panel(Index).side(Current_side).LedY(8)
Select Case panel(Index).side(Current_side).ledcolor(8)
Case Is = 1
Picture8(Index).Picture = LoadPicture("c:fred.fred1.bmp")
Case Is = 10
Picture8(Index).Picture = LoadPicture("c:fred.fred2.bmp")
Case Is = 11
Picture8(Index).Picture = LoadPicture("c:fred.fred3.bmp")
Case Is = 0
Picture8(Index).Picture = LoadPicture("c:fred.fred4.bmp")
End Select

End If

End Sub
```

```
Private Sub side1_DragDrop(Index As Integer, Source As Control, X As Single, y As Single)
ngle)
```

```
If Option6.Value = True Then
```

change_side - 5

```
Picture1(Index).DragMode = 0
Picture1(Index).Left = X
Picture1(Index).Top = y

Picture2(Index).DragMode = 0
Picture2(Index).Left = X
Picture2(Index).Top = y

Picture3(Index).DragMode = 0
Picture3(Index).Left = X
Picture3(Index).Top = y

Picture4(Index).DragMode = 0
Picture4(Index).Left = X
Picture4(Index).Top = y

Picture5(Index).DragMode = 0
Picture5(Index).Left = X
Picture5(Index).Top = y

Picture6(Index).DragMode = 0
Picture6(Index).Left = X
Picture6(Index).Top = y

Picture7(Index).DragMode = 0
Picture7(Index).Left = X
Picture7(Index).Top = y

Picture8(Index).DragMode = 0
Picture8(Index).Left = X
Picture8(Index).Top = y
End Sub
```

4) Appendix 4 : Warp.pds program

```

;PALASM Design Description

;----- Declaration Segment -----
TITLE    WARP controller logic
PATTERN  Sean
REVISION 0
AUTHOR   Sean Collins
COMPANY  University of Lancaster
DATE     04/01/98

CHIP     _warp PAL22V10

;----- PIN Declarations -----
PIN 1      PSEN          COMBINATORIAL ; INPUT
PIN 2      RD            COMBINATORIAL ; INPUT
PIN 3      WR            COMBINATORIAL ; INPUT
PIN 4      A11           COMBINATORIAL ; INPUT
PIN 5      A12           COMBINATORIAL ; INPUT
PIN 6      A14           COMBINATORIAL ; INPUT
PIN 7      A13           COMBINATORIAL ; INPUT
PIN 8      A15           COMBINATORIAL ; INPUT
PIN 9      A10           COMBINATORIAL ; INPUT
PIN 10     A9            COMBINATORIAL ; INPUT
PIN 11     A8            COMBINATORIAL ; INPUT
PIN 18     EXT           COMBINATORIAL ;
OUTPUT
PIN 14     SRAM_CE       COMBINATORIAL ;
OUTPUT
PIN 15     SRAM_RD       COMBINATORIAL ;
OUTPUT
PIN 16     EWR           COMBINATORIAL ;
OUTPUT
PIN 17     ERD           COMBINATORIAL ;
OUTPUT

;----- Boolean Equation Segment -----
EQUATIONS

IF (A8 * A9 * A10 * A11 * A12 * A13 * A14 * A15) THEN
BEGIN
EXT = 0
END
ELSE
BEGIN
EXT = 1
END

SRAM_RD = RD * PSEN
SRAM_CE = /(A15 * EXT) ;SRAM enabled when addr > 7FFF and EXT HI
EWR = EXT + WR
ERD = EXT + RD

;----- Simulation Segment -----
SIMULATION

TRACE_ON A8 A15 RD WR EXT SRAM_RD SRAM_CE ERD EWR
SETF A8 A9 A10 A11 A12 A13 A14 A15 /psen /rd wr
CHECK /EXT SRAM_CE
setf psen
check /SRAM_RD SRAM_CE /ERD ; SRAM NOT enabled RD doesn't matter
SETF /A8

```

CHECK EXT /SRAM_RD /SRAM_CE ERD ; SRAM enabled, RD used
SETF A8 /WR RD
CHECK SRAM_CE /EWR /EXT ERD
TRACE_OFF

-

5) Appendix 5: program Fredled.pds

```

;PALASM Design Description

;----- Declaration Segment -----
TITLE    Fred's 1st LED controller
PATTERN  1.0
REVISION 0
AUTHOR   Frederic Gilet & Sean Collins
COMPANY  Lancaster University
DATE     06/18/98

CHIP    _FredLED PAL22V10

;----- PIN Declarations -----
PIN 1      EWR      COMBINATORIAL ; INPUT
PIN 2      A0       COMBINATORIAL ; INPUT
PIN 3      a1       COMBINATORIAL ; INPUT
PIN 4      A2       COMBINATORIAL ; INPUT
PIN 5      A3       COMBINATORIAL ; INPUT
PIN 6      A4       COMBINATORIAL ; INPUT
PIN 7      A5       COMBINATORIAL ; INPUT
PIN 8      A6       COMBINATORIAL ; INPUT
PIN 9      A7       COMBINATORIAL ; INPUT
PIN 13     LATCH_0  COMBINATORIAL ;
OUTPUT
PIN 14     LATCH_1  COMBINATORIAL ;
OUTPUT

;----- Boolean Equation Segment -----
EQUATIONS

if (/EWR * /A7 * /A6 * /A5 * /A4 * /A3 * /A2 * /A1 * /A0) then
  begin
    LATCH_0 = 1
  end
else
  begin
    LATCH_0 = 0
  end

if (/EWR * /A7 * /A6 * /A5 * /A4 * /A3 * /A2 * /A1 * A0) then
  begin
    LATCH_1 = 1
  end
else
  begin
    LATCH_1 = 0
  end

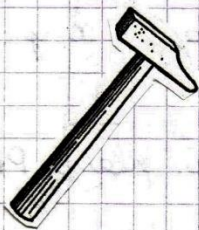
;----- Simulation Segment -----
SIMULATION

;-----

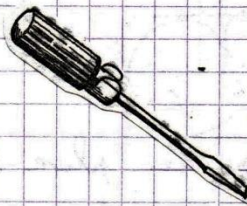
```



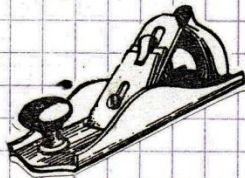
CAHIER



DE



TECHNOLOGIE



INTRODUCTION

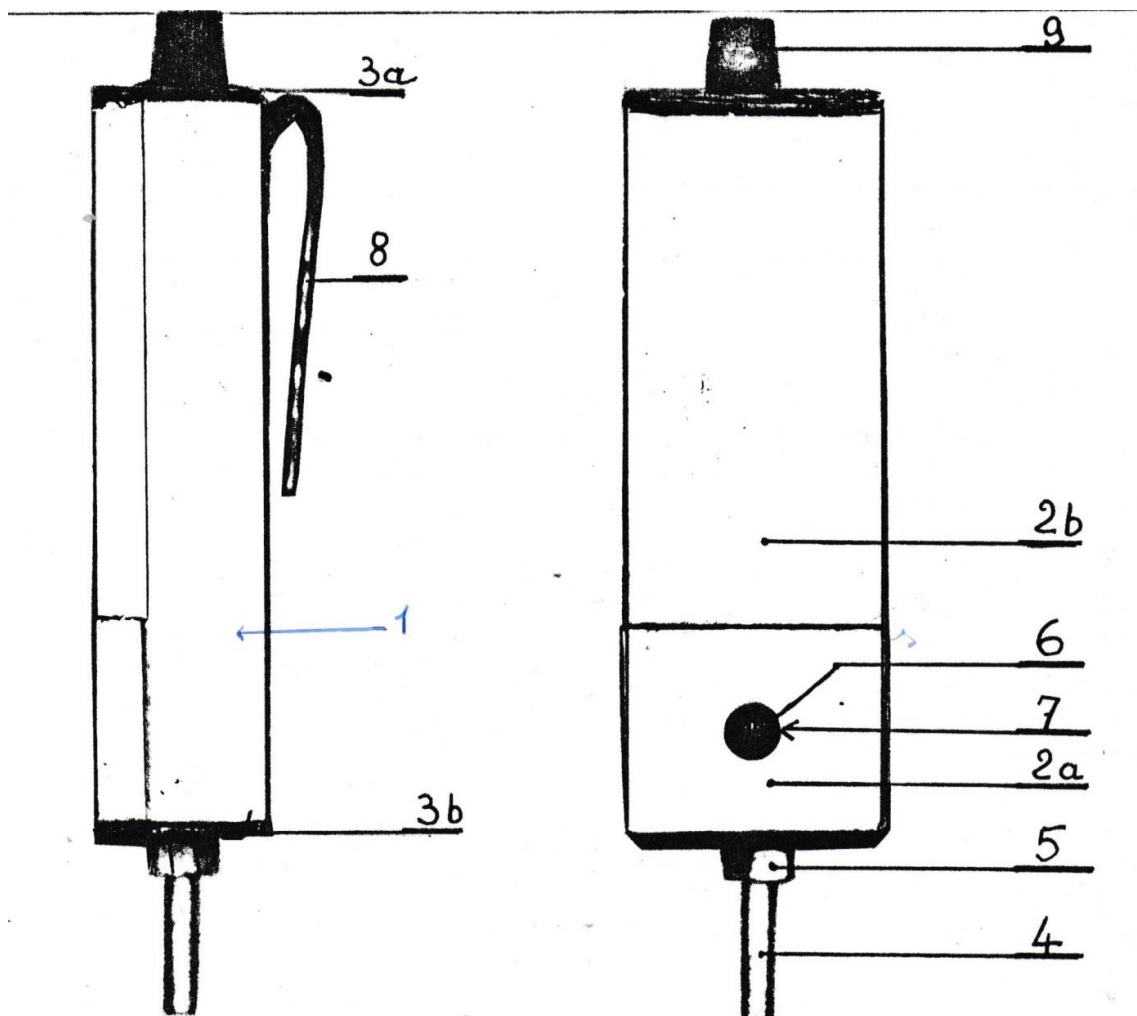
Ce document est inspiré de mon cahier de technologie de 6^{ème}.

La réalisation d'un objet est considérée comme un travail en mini-entreprise : plans, outils, commande des composants, réalisation d'après un plan dessiné, facture, étude de marché, etc...

Elle exige découpe, gravure et perçage d'un circuit imprimé, sélection et soudure des composants, montage dans un boîtier à fabriquer.

LE PRODUIT A REALISER

C'est un testeur binaire de présence de courant. On branche l'appareil au deux extrémités à mesurer. La LED s'allume si il y a du courant.



LES OUTILS

ELECTRONIQUE



Pince coupante
Pour couper les fils et les pattes des composants

Pince à bec plat
Pour tordre les pattes des composants

Pince à dénuder
Pour retirer l'isolant des fils



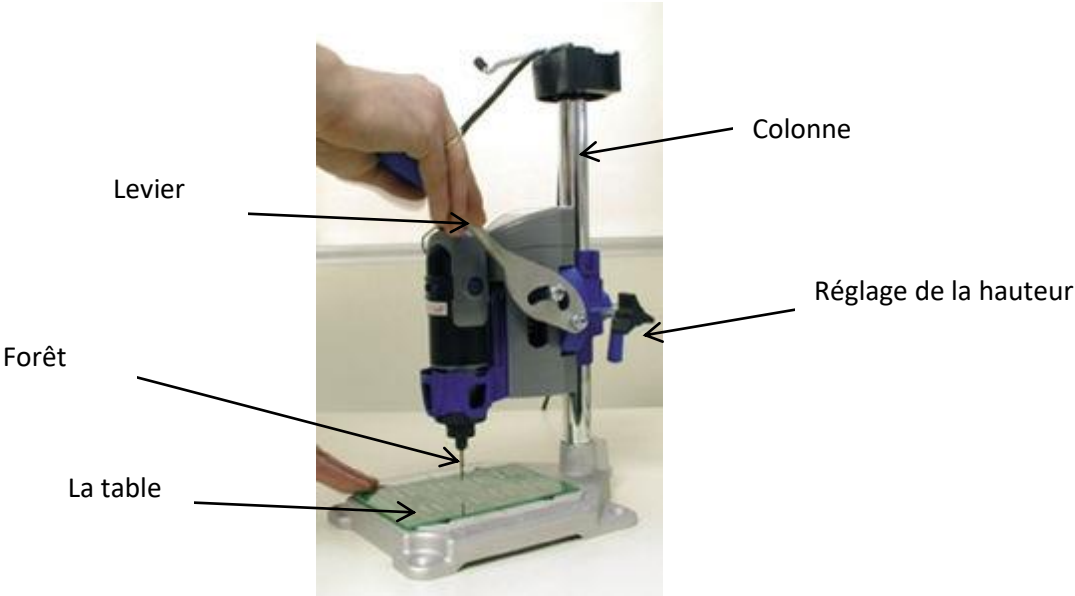
Le fer à souder



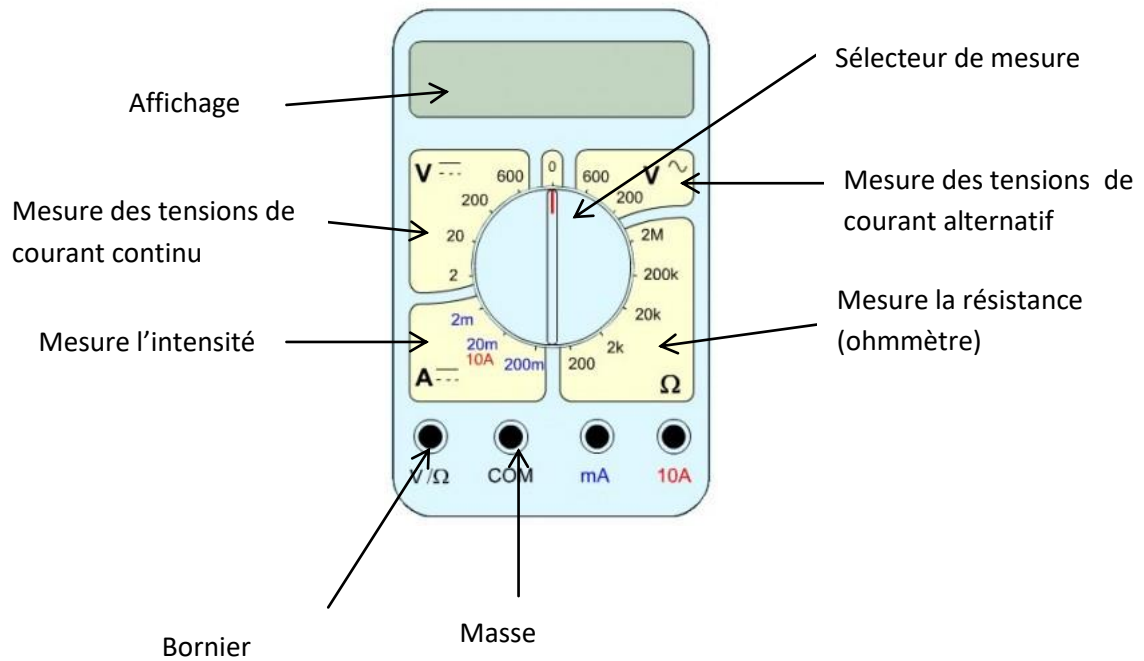
La pompe à dessouder
Pour aspirer la soudure liquide



MINI-PERCEUSE



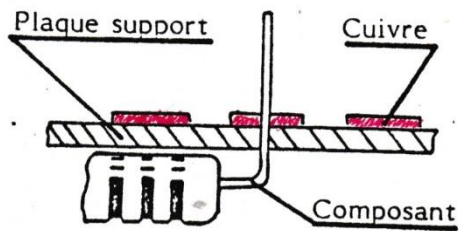
LE MULTIMETRE



Utilisation :

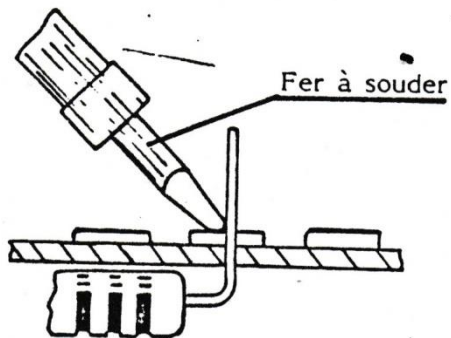
- Brancher le fil rouge sur la borne adéquate (tension, intensité, résistance, tension alternative)
- Brancher le fil noir sur la masse
- Positionner le sélecteur sur la mesure adéquate (tension, intensité, résistance, tension alternative) en choisissant le calibre

LA FORMATION AU SOUDAGE



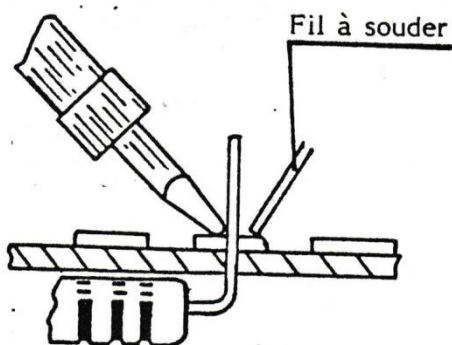
1^{ère} Opération :

Implanter les composants (côté non cuivré)



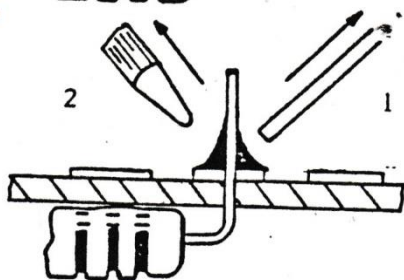
2^{ème} Opération :

On chauffe avec le fer la patte du composant et la pastille du circuit



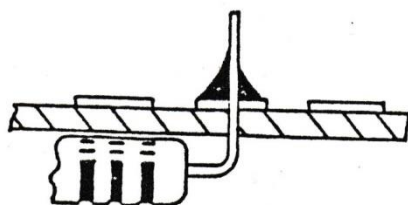
3^{ème} Opération :

A l'opposé de la panne de fer, on place le fil à souder en contact avec la patte et la pastille



4^{ème} Opération :

On retire en même temps le fer et le fil à souder



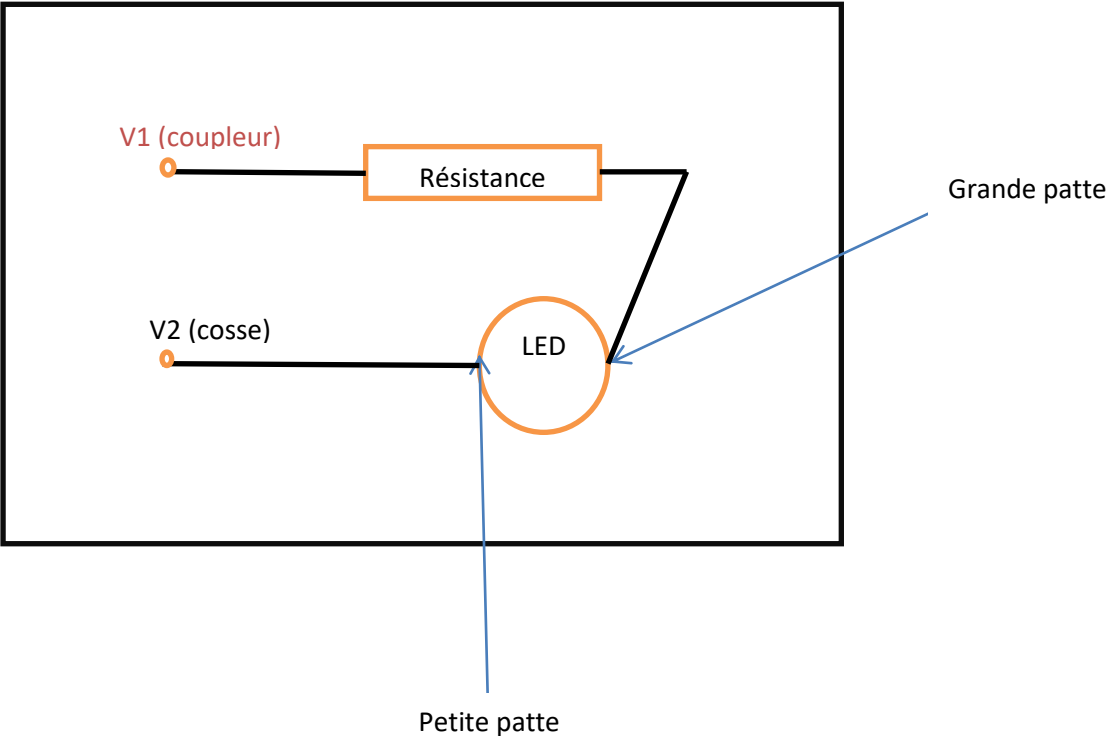
5^{ème} Opération :

On vérifie la soudure :

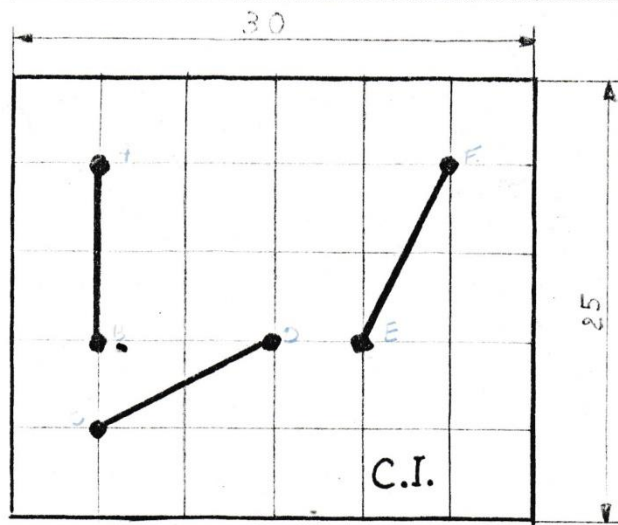
- Bonne : on coupe la patte
- Mauvaise : on réessaye de chauffer la soudure

LES PLANS ELECTRONIQUES

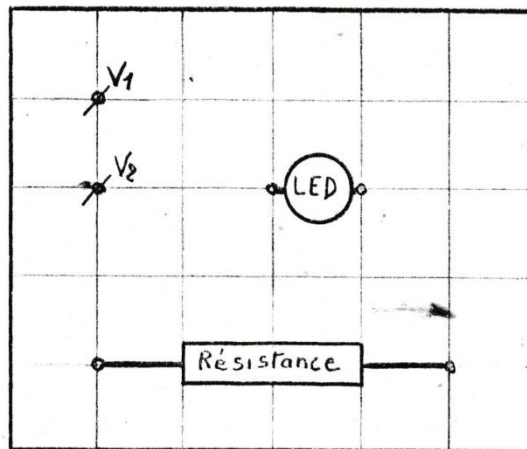
LE CIRCUIT DU TESTEUR



CIRCUIT IMPRIME



COTE CIRCUIT



COTE COMPOSANTS

Ech 3:1 Collège Privé de THOUARCE

Nom :

CIRCUIT DU TESTEUR

Date :

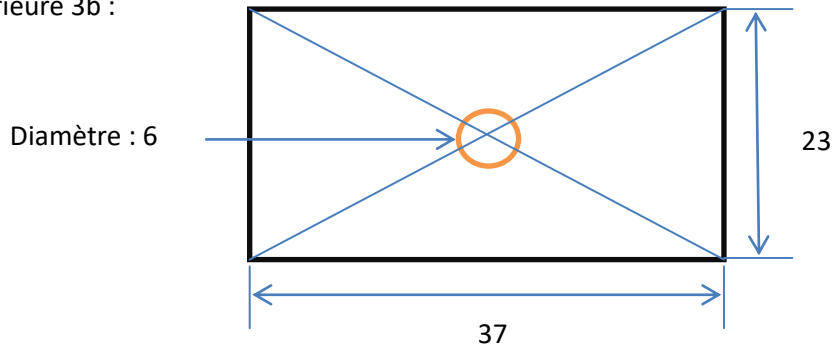
N° 2

LES PLANS MECANIQUES : LE CORPS DU BOITIER

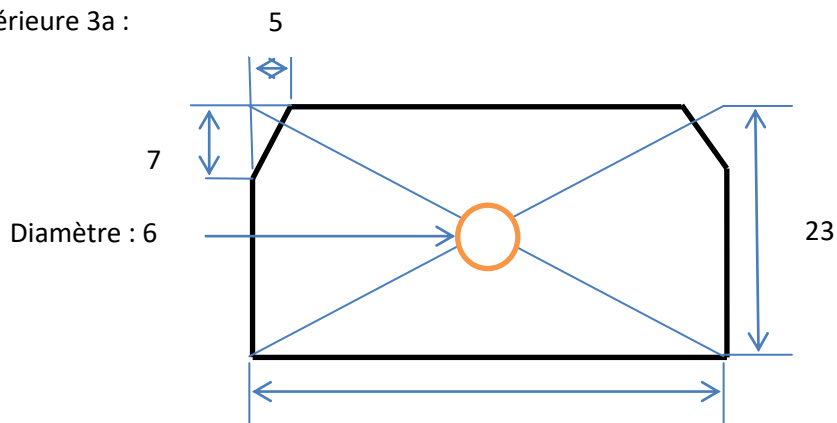
On utilise un corps de PVC.

FABRICATION DES EXTREMITES

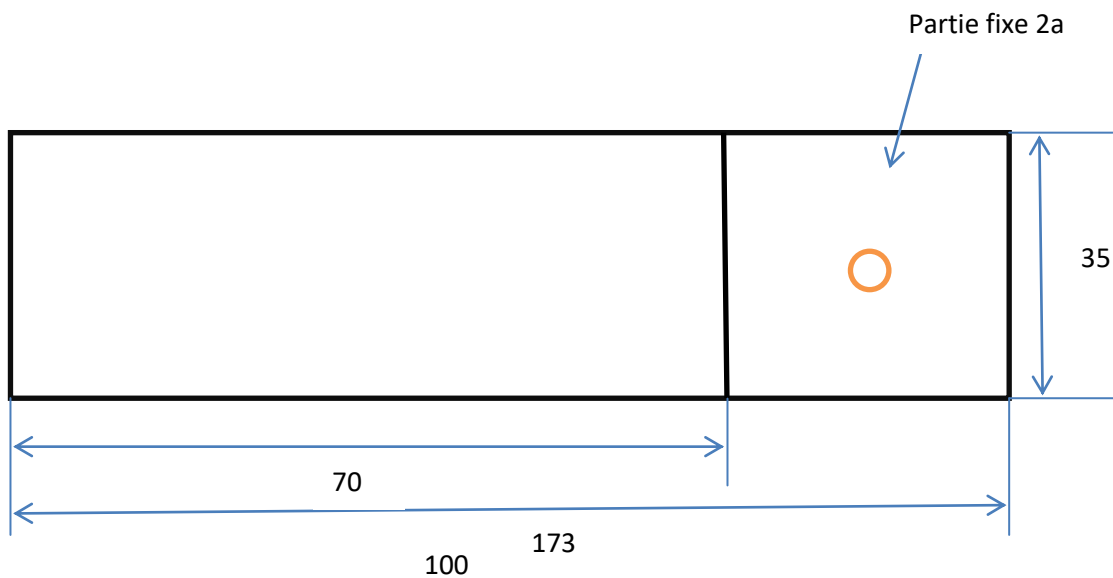
Inférieure 3b :



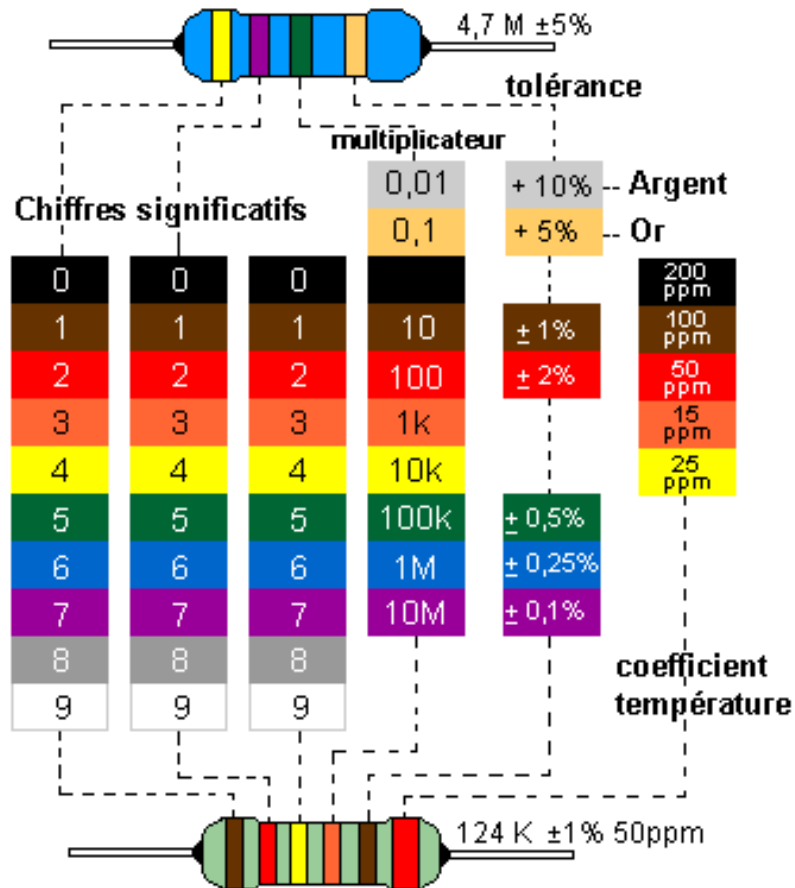
Supérieure 3a :



Corps du boîtier 1 :

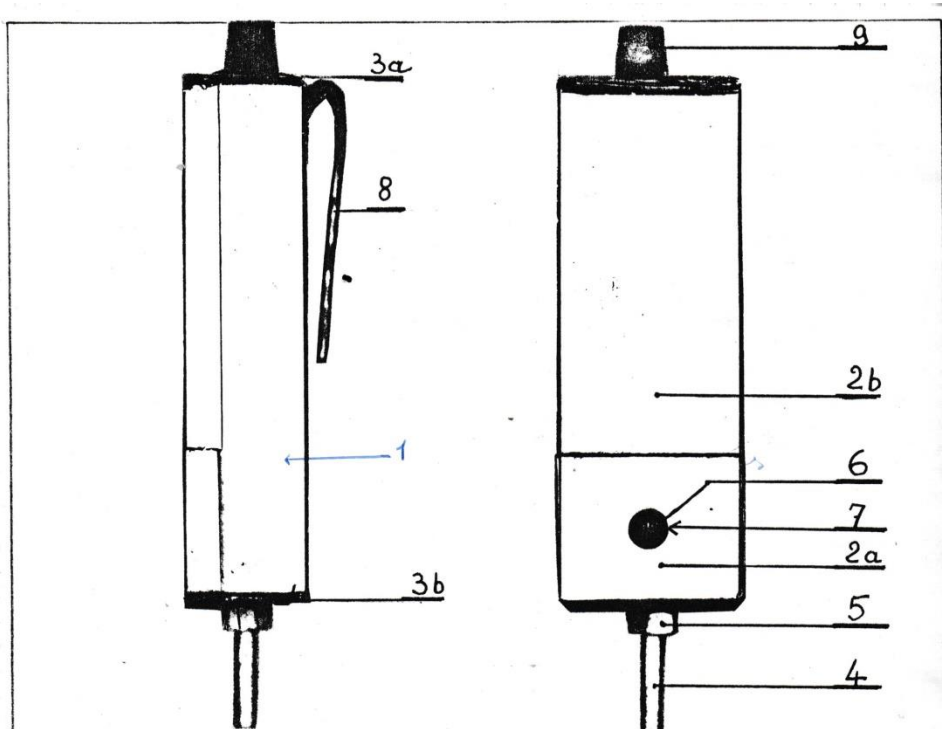


CONSULTATION DE DOCUMENTATION TECHNIQUE : LA RESISTANCE



ORDRE DE FABRICATION : LE MONTAGE MECANIQUE

- Couper les pièces 3a et 3b aux bonnes dimensions
- Monter la LED sur le couvercle fixe 2a
- Monter la vis sur l'extrémité inférieure 3b sans oublier la cosse du fil noir
- Coller le couvercle fixe 2a sur la pièce 1
- Coller la pièce 3b sur la pièce 1
- Monter la borne sur l'extrémité supérieure 3a sans oublier la cosse du fil rouge
- Coller la pièce 3a sur la pièce 1
- Insérer le circuit imprimé par l'ouverture du couvercle



9	1	Borne		
8	1	Agrafe	PVC.	Plaque ep.2
7	1	Support de LED		
6	1	LED rouge		
5	1	Ecrou	Acier	M 6
4	1	Pointe de touche	Acier	Vis 6 X 30
3	2	Extrémité ^a Supérieure _b Inférieure	PVC	Plaque ep.2
2	1	Couvercle ^a fixe _b mobile	PVC	Gaine 35x23
1	1	Corps de boîtier	PVC	Gaine 35x23
Rep	Nbr	Désignation	Matériau	Observations
Ech:1:1	TESTEUR DE POCHE			Nom
□ ⊙				N° 1

LA COMMANDE DES COMPOSANTS

EXEMPLE DE BON DE
COMMANDE

ATLANTIQUE COMPOSANTS

S.A.R.L. au capital de 50.000 Francs
189, Avenue Pasteur
49100 ANGERS
Telephone: 41.43.42.30

Domiciliations Bancaires:
C.I.D. ANGERS. 30047 299 055901L 78
C.C.P. NANTES. 20041 1011 989168032 35
SIRET:320 090 269 00012 CODE APE:5804

COLLEGE PRIVE THOUARCE

BON DE LIVRAISON

NO 02317 LE 110387

rue du 11 Novembre
49380 THOUARCE

REF COMMANDE :MR PASSIGNAT

REFERENCE	DESIGNATION	QTE CDEE	QTE LIVREE	RESTE A LIVR	PU.HT	MONTANT HT
S M5B	Cosses a souder MFOM N°5 6,2mm	300.00	300.00	0.00	0.08	22.80
DPC5	RTC757A Clip plastique led 5mm	100.00	100.00	0.00	0.38	37.90
ACFR423	EPOXY presensibilise 1 face 200x300	4.00	4.00	0.00	62.15	248.60
R 0.25	Resistance 0.25W 5% serie E24	170.00	170.00	0.00	0.15	25.84
C H25470	Condensateur chimique 470MF25V FITCO 032	85.00	35.00	50.00	4.46	156.07
C P40022	Condensateur plastique C368 22NF400V	85.00	85.00	0.00	0.76	64.52
TRBC547	BC 547B Transistor NPN 45V 0.1A T092	170.00	170.00	0.00	0.76	129.37
D 20.43.9	Diode zener 0.4W 3V9	85.00	85.00	0.00	0.76	64.52
P H101M	Pot. ajust. PIHER horizontal 1 00hms	155.00	155.00	0.00	1.16	179.96
I 5636	1 inver.3A/250V 2 pos.fix.P:7mm APR:5636	170.00	70.00	100.00	4.90	342.79
U B712	Buzzer B-12V 22mA Frequence 300 a 500 hz	85.00	5.00	80.00	7.83	39.13
I S90TR	Poussoir ECONOMIQUE 1 T. 1A/250V rouge	85.00	85.00	0.00	2.21	188.28

COLLEGE PRIVE THOUARCE

BON DE LIVRAISON

NO 02455 LE 200387

rue du 11 Novembre
49380 THOUARCE

REF COMMANDE :DU 19.3.87

REFERENCE	DESIGNATION	QTE CDEE	QTE LIVREE	RESTE A LIVR	PU.HT	MONTANT HT
K C816	Support double lyre 16 broches	15.00	15.00	0.00	1.21	18.21
T TR4006	Triac standart 6A 400V T0220	15.00	15.00	0.00	4.55	68.30
D 20.46.8	Diode zener 0.4W 6V8	2.00	2.00	0.00	0.76	1.52
S CPP	Connection a pression pour pile R622 9v	100.00	100.00	0.00	0.71	71.30
G SM4R	Fiche BANANE 4 mm male a souder rouge	100.00	100.00	0.00	1.79	178.70
G SM4N	Fiche BANANE 4 mm male a souder noire	100.00	100.00	0.00	1.79	178.70
A EC910	Transfert ALFAC 306 pastilles 2.40 mm	10.00	10.00	0.00	6.94	69.44
ACFR423	EPOXY presensibilise 1 face 200x300	2.00	2.00	0.00	62.15	124.30
G SM4N	Fiche BANANE 4 mm male a souder noire	100.00	100.00	0.00	1.80	179.90
DPRS	CRV 24 Led rouge 5mm	50.00	0.00	50.00	0.54	0.00

G MFY71	300.0	X	Picot sabre MFOM pour C.1.percage1.3mm	3	0.06	18.30
G SD4N	64.0	X	Douille BANANE isolee 4 mm noire	3	1.74	92.25

LA FACTURATION

LA FACTURE

		Collège privé rue du 11 novembre 49xxx Pelouaille			
FACTURE N°1		Client :	Frédéric Gilet		
LE 10 AVRIL 1987			avenue de Gaulle 49xxx Pelouaille		
Référence article	quantité	Désignation	Prix unitaire HT	Montant net HT	
SMSB	2	Cosse à souder 3,2 mm	0,08	0,16	
OPC5	1	clip plastique LED 5 mm	0,38	0,38	
ACFR 423		EPOXY présensibilisé 200*300	62,15	0,78	
R0,25	1	résistance	0,15	0,15	
5CPP	1	connection à pression pour pile	0,71	0,71	
G.SM4N	2	fiche banane	1,79	3,58	
OPR5	1	LED rouge 5 mm	0,54	0,54	
SMFY71	2	picot sabre	0,06	0,12	
GSD4N	1	douille banane	1,44	1,44	
			TOTAL	7,86	
Montant brut	TVA	Montant TVA	Montant TTC	Règlement	NET A PAYER
7,86	18,60%	1,46196	9,32196	Par chèque	9,32

LE MODE DE REGLEMENT

En espèce (argent liquide)

Par chèque bancaire ou postal

Par carte bancaire

Par mandat

Par virement (passage d'argent d'un compte à un autre)

Par prélèvement automatique

Par traites (règlements retardés)

UNE ENTREPRISE AU COLLEGE

Il faut :

- | | |
|--------------------------|---|
| - Un nom | TEC (Testeurs du collège) |
| - Un produit à fabriquer | Atelier mécanique et électrique du collège |
| - Du matériel | Matériel du collège |
| - De l'argent (capitaux) | Argent du collège |
| - Des fournisseurs | Atlantique Composants, quincaillerie Doriéssine |
| - Des clients | Electrome |
| - Des points de vente | Elève |

Pour vendre le produit, il faut connaître avec une précision suffisante :

- | | |
|---|--------------------|
| - Les personnes qui pourraient acheter le produit : | LA CLIENTELE |
| - Les entreprises qui fabriquent et vendent un produit proche : | LES CONCURRENTS |
| - Ce que va coûter la fabrication du produit | LE COUT DE REVIENT |
| - Ce que l'on peut gagner grâce à la vente du produit | LE BENEFICE |

C'est l'étude de marché.

EXEMPLE D'ETUDE DE MARCHE SUR LA CONSOMMATION DE PILES :

Questions posées

- 1) Quels types de piles utilisez-vous ?
 - R03
 - R6
 - R14
 - R20
 - 6F22 ou 6N75P
 - 4.5V
 - Autre type
- 2) Quelle est chez vous la marque la plus utilisée ?
- 3) Quels objets font-elles fonctionner ?
 - Lampe de poche
 - Réveil
 - Jouet
 - Testeur
 - Poste radio
 - Calculatrice
 - Appareil photo
 - autre

4) Quand vous achetez des piles, choisissez-vous ?

- | | | |
|----------------------------|-----|-----|
| - la moins chère | oui | non |
| - préférez-vous une marque | oui | non |

Après dépouillement du panel de clients interrogés, qui habitent dans la vallée, il s'avère que :

- 1) tous les types de pile sont utilisés dans la plupart des cas
- 2) la marque la plus utilisée est Mazda
- 3) elles font fonctionner en majorité appareil photo et caméras, lampes de poche, calculatrices, testeurs, réveils-montre-horloge

4) Le choix entre une marque et la moins chère est équilibré

Conclusion

Les familles du coin consomment un grand nombre de piles de types différents car elles possèdent de nombreux appareils électriques non raccordés au secteur EDF. La marque préférée est Mazda. La consommation semble pouvoir augmenter. D'autres marques semblent pouvoir s'y implanter.

LE MARKETING

Le but est de concevoir et de réaliser une affiche destinée à faire connaître :

- La société, TEC
- Le produit, le testeur

Il faut inventer et utiliser quelques :

- Slogans
- Dessins
- Couleurs



L'INGENIEUR ET SES FONCTIONS

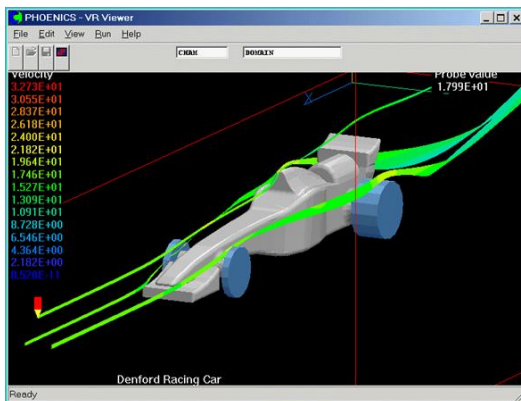
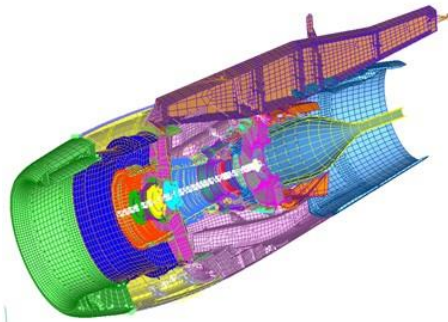
L'ingénieur est un travailleur qui a fait des études supérieures en technologie.

L'ingénieur de bureau d'étude

- Mécanique (conception assistée par ordinateur)
- Electronique (conception des cartes)
- Automatisation (programme des machines)
- Structure (calculs)



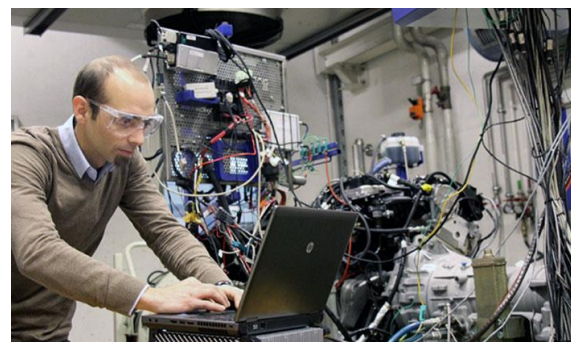
- Aérodynamisme



- Thermodynamique (moteurs thermiques)
- Electrotechnique (moteurs électriques)

L'ingénieur d'essais

Il utilise son ordinateur ou son laboratoire effectuer des essais.



pour

L'ingénieur méthodes

Il organise la production techniquement (bons de fabrication) en collaboration avec les achats et les commerciaux.

L'ingénieur d'atelier

Il dirige :

- Une forge
- Une fonderie

- Un atelier d'usinage (fraiseuses, tours numériques)
- Un atelier de déformation plastique (presse à chaud ou à froid)
- Un atelier de soudure robotisé ou non
- Une ligne de montage (automobile, industrie, aéronautique, etc...)



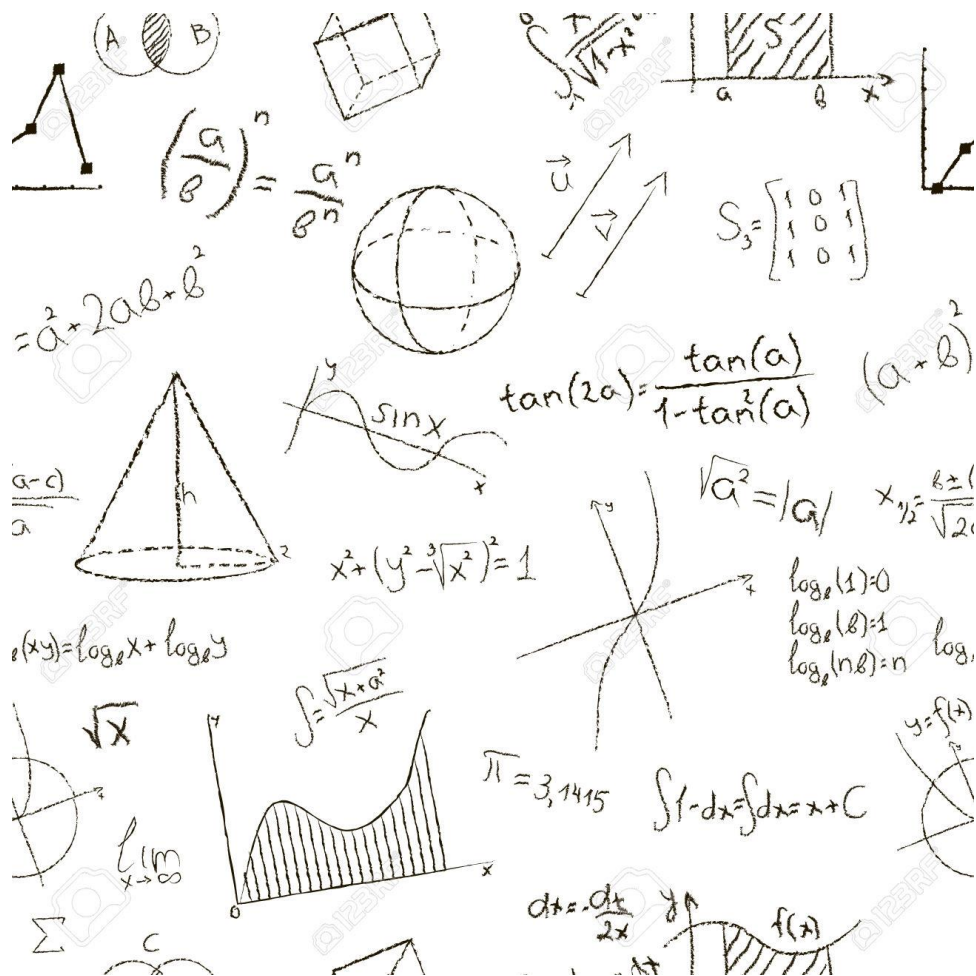
Il est responsable des problèmes (urgences), des hommes (recrutement et RH), de la bonne marche de l'atelier, des implantations et investissements (nouvelle organisation, nouvelle machine).

Il est en perpétuelle évolution : formation aux nouvelles technologies, réunions de travail, visites, etc...

L'ingénieur en informatique

- Il conçoit et entretient les programmes qui font fonctionner les logiciels :
 - Systèmes
 - Bases de données
 - Programmation

Mathématiques de l'ingénieur



Séries

Séries numériques.

La série est w si la suite des sommes partielles est w .

Somme de la série : $S = \lim_{n \rightarrow \infty} S_n = \lim_{n \rightarrow \infty} \sum_{p=0}^n u_p = \sum_0^{\infty} u_n$

La série géométrique de terme général $u_n = u_0 q^n$ est w si $|q| < 1$

alors $\Rightarrow S = \sum_{n=0}^{\infty} \frac{u_0}{1-q}$ et $R_n = \frac{u_0 q^{n+1}}{1-q}$

Propriété Condition nécessaire mais NON SUFFISANTE :

$$\lim_{n \rightarrow \infty} u_n = 0$$

Lorsque $u_n \not\rightarrow 0$, on dit que la série diverge grossièrement.

Théorèmes : une CNS pour que la série à termes positifs w est que la suite des Σ partielles soit majorée.

Comparaison : $0 \leq u_n < v_n$

a) si $\sum v_n w \rightarrow \sum u_n w$

b) si $\sum u_n dw \rightarrow \sum v_n dw$

Equivalence : $(\sum u_n)$ et $(\sum v_n)$ tq $u_n \sim v_n$
les séries ont même nature

Comparaison avec l'intégrale impropre - séries de Riemann
 $f: [a, +\infty[\rightarrow \mathbb{R}$, fonction continue, positive, décroissante

alors
$$\int_k^{k+1} f(x) dx \leq f(k) \leq \int_{k-1}^k f(x) dx$$

d'où
$$\int_1^{n+1} f(x) dx \leq S_{n+1} \leq \int_1^n f(x) dx + f(1)$$

Les séries de Riemann cr si $\alpha > 1$

Critère de d'Alembert : comparaison à 1 série géométrique

1) $\frac{u_{n+1}}{u_n} < k < 1$ alors la série $(\sum u_n)$ cr.
 etc.

2) $\frac{u_{n+1}}{u_n} \geq 1$, alors la série $(\sum u_n)$ dr.

3) $\frac{u_{n+1}}{u_n} < 1$, on ne peut conclure

$\lim_{n \rightarrow \infty} \frac{u_{n+1}}{u_n} = l$

1) si $0 \leq l < 1$, la série cr

$l > 1$, la série dr

$l = 1$ ou $l = 1^+$, on ne peut conclure

$l = 1^-$, série dr

Lim suite série

pour étudier la suite (u_n) , il est équivalent d'étudier
 la série $v_n = u_{n+1} - u_n$

Calcul : ex: $u_n = \frac{1}{n(n+1)} = \frac{1}{n} - \frac{1}{n+1} \dots$

$$u_n = \frac{n^2+1}{n(n+1)} = \frac{n(n-1)+n+1}{n(n+1)} \dots$$

Opéral sur les séries

$\sum u_n$ et $\sum v_n$ cr

1) $\alpha \sum u_n$ cr et $\sum (u_n + v_n)$ cr.

2) si $\sum u_n$ cr et $\sum v_n$ cr, $\Rightarrow \sum (u_n + v_n)$ cr.

3) si $\sum u_n$ cr et $\sum v_n$ cr, on ne peut rien conclure.

Séries absolues cr (lorsque $\sum |u_n|$ cr)

1) toute série absolue cr est cr.

Séries alternées $u_n = (-1)^n |u_n|$.

Critère spécial des séries alternées $\begin{cases} |u_{n+1}| < |u_n| \\ \lim |u_n| = 0 \end{cases}$

alors la série alternée est cr.

Pour 1 série vérifiant le CSSA:

1) $R_n \leq |u_{n+1}|$

2) la somme est du signe de u_0 .

Séries de fonctions

I) Cr simple, uniforme, normale.

la série cr simple sur A sci, $\forall x \in A$, la série num. $\sum u_n(x)$ cr.
 $\forall x \in A, \forall \varepsilon > 0, \exists n > N_{\varepsilon, x}, \Rightarrow |S(x) - S_n(x)| < \varepsilon$

cr uniforme

$\forall \varepsilon > 0, \exists N_{\varepsilon} \text{ tq } n > N_{\varepsilon} \Rightarrow |S(x) - S_n(x)| < \varepsilon, \forall x \in A$

Critères de cr uniforme.

1) $\sum u_n$ cr uniformément si

$$\sup |S(x) - S_n(x)| = \sup |R_n(x)| = M_n(x)$$

$$\lim_{n \rightarrow \infty} M_n(x) \rightarrow 0$$

2) $\sum u_n$ cr unif si \exists une suite numérique ε_n tq

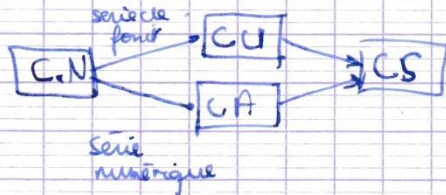
$$\lim_{n \rightarrow \infty} \varepsilon_n = 0, |S(x) - S_n(x)| < \varepsilon_n, \forall x \in A$$

3) $\sum u_n$ ne cr pas uniformément si \exists une suite de pts x_n tq
 $|S(x) - S_n(x)| = R_n(x_n)$ ne td pas vers 0 lorsque $n \rightarrow \infty$

Convergence normale

On dit que la série $\sum_{n=0}^{\infty} u_n(x)$ cr normale sur A si il
 ex 1 série numérique $\sum \varepsilon_n$ cr tq

$$|u_n| < \varepsilon_n \quad \forall x \in A, \text{ et } \forall n \in \mathbb{N}. (n \geq N_0)$$



Remarque : la cr normale est v à la cr de la
série num $\sum \|u_n\|$ où $\|u_n\| = \sup_{x \in I} |u_n(x)|$

Théorème de continuité.

$u_n(x)$ est une suite numérique continue
 $\sum u_n(x)$ est une série de fonct cr uniformém^r sur I
⇒ alors $S(x) = \sum_0^{\infty} u_n(x)$ est une fonct continue sur I

Contre-exemple : u_n est 1 suite d'applicat^r continue
Sur $\sum u_n$ n'est pas continue sur I
la série de fonct n'est pas unif cr sur I

Théorème d'intégration : soit $u_n(x)$ une suite d'applicat^r
continues sur $[a, b]$ $\sum u_n$ cr uniformément sur $[a, b]$

$$\text{alors } \int_a^b \sum_{n=0}^{\infty} u_n(x) dx = \sum_{n=0}^{\infty} \int_a^b u_n(x) dx$$

Théorème de dérivation.

soit une applicat^r numérique u_n qui est C' sur I
int I tq

a) $\exists x_n \in I$ tq $\sum u_n(x_n)$ cr.

b) la série $\sum u_n'$ cr uniformém^r sur I

alors $\rightarrow \sum u_n$ cr simplem^r sur I et uniformém^r
sur tt intervalle borné de I

$\rightarrow S(x)$ est une fonct C' et $S'(x) = \sum_{n=0}^{\infty} u_n'(x)$

Remarque : cr unif sur $[a, +\infty[$

si $x > 0$, on peut par ex choisir $a = \frac{x}{2}$

\Rightarrow cr unif sur $]0, +\infty[$.

Séries entières

Rayon de w .

1) Lemme d'Abel

Soit $a_n z_0$ la suite $a_n z_0^n$ suite bornée, pour $|z| < |z_0|$
la série entière est abs cv $\sum a_n z^n$ est abs cv. Elle
normalement cv sur tt disque fermé.

$$D_k = \{z \in \mathbb{C} / |z| \leq k|z_0| = 0, (0, k|z_0|)\}, k \in [0, 1] \text{ fixe}$$

Rayon de w . Théorème.

$\sum a_n z^n$ série entière, \exists 1 seul nombre $R \in (0, +\infty)$

1) $|z| < R$: la série est abs cv

2) $|z| < R$, la série est normalement cv, $0 < r < R$

3) $|z| > R$, la série diverge grossièrement.

• Soit a Rayon de w de la série entière $\sum a_n x^n$.

$$R = \frac{1}{L} \text{ avec } \lim_{n \rightarrow \infty} \frac{a_{n+1}}{a_n} = L \in \mathbb{R}^+$$

Opérations algébriques sur les séries entières.

2 séries entières $\sum a_n x^n$ de rayons R_1 et R_2

1) si $R_1 \neq R_2$, R de $\sum (a_n + b_n) x^n$ est $\inf(R_1, R_2)$

2) si $R_1 = R_2$, R est $\geq R_1$

3) R_1, R_2 , R de $(\sum a_n x^n) (\sum b_n x^n)$ est $\inf(R_1, R_2)$

Théorème 1) la \sum d'1 série entière de var réelles est

1) fonction continue en tt pt de l'intervalle.

2) $\sum a_n x^n$ série entière. Alors on peut dériver terme à terme

3) la \sum d'1 série entière est \mathcal{C}^∞ .

Développements classiques en série entière

$$e^x = \sum_{n=0}^{+\infty} \frac{x^n}{n!} \quad R = +\infty$$

$$\ln(1+x) = \sum_{n=1}^{+\infty} \frac{(-1)^{n-1} x^n}{n} \quad R = 1$$

$$\frac{1}{1-x} = \sum_{n=0}^{+\infty} x^n \quad R = 1$$

$$\cos x = \sum_{p=0}^{+\infty} \frac{(-1)^p x^{2p}}{(2p)!} \quad R = +\infty$$

$$\sin x = \sum_{p=0}^{+\infty} \frac{(-1)^p x^{2p+1}}{(2p+1)!} \quad R = +\infty$$

$$(1+x)^m = \sum_{n=0}^{+\infty} \binom{m}{n} x^n \quad R = 1$$

$$\cosh x = \sum_{p=0}^{+\infty} \frac{x^{2p}}{(2p)!} \quad R = +\infty$$

$$\sinh x = \sum_{p=0}^{+\infty} \frac{x^{2p+1}}{(2p+1)!} \quad R = +\infty$$

III) Fonct complexes classiques.

$$z \rightarrow \sum_{n=0}^{+\infty} \frac{a_n z^n}{n!} \quad \text{c'est la fonction complexe}$$

$$\operatorname{ch} iy = \cos y$$

$$\operatorname{sh} iy = i \sin y$$

$$\text{cosinus complexe} : \cos z = \sum_{p=0}^{+\infty} \frac{(-1)^p z^{2p}}{(2p)!}$$

sinus complexe

cos hyperbolique

}

Séries de Fourier.

I) Séries trigonométriques.

on appelle série trigo & série de fourier $\sum a_n \cos nx$

$$u_n(x) = a_n \cos nx + b_n \sin nx$$

en complexe : $c_n = \frac{a_n - ib_n}{2}$

$$\sum u_n(x) = \sum_{-\infty}^{+\infty} c_n e^{inx}$$

Convergence:

Propriété : si $\sum a_n$ est \mathbb{R} et si $\sum b_n$ est \mathbb{R} , alors la série trigo. est normalement \mathbb{R} sur \mathbb{R} .

Propriété : si la fonct. trigo u ~~est~~ simplement, alors sa somme est 1 fonct. 2π -périodique

Generalisation coefficients de Fourier

f 2π -périodique, continue par morceaux, à valeurs réelles ou complexes

$$a_0 = \frac{1}{2\pi} \int_a^{a+2\pi} f(x) dx$$

$$a_n = \frac{1}{\pi} \int_a^{a+2\pi} f(x) \cos nx dx$$

$$b_n = \frac{1}{\pi} \int_a^{a+2\pi} f(x) \sin nx dx$$

$$c_n = \frac{1}{2\pi} \int_a^{a+2\pi} f(x) e^{-inx} dx$$

Théorème de Dirichlet

Soit f fonction ^{périodique} C^1 par morceaux sur \mathbb{R} , alors la série de Fourier associée à f converge simplement sur \mathbb{R} et a pour somme

- 1) $S_f(x) = f(x)$ lorsque f est continue en x
- 2) $S_f(x) = \frac{f(x+0) + f(x-0)}{2}$ lorsque f est discontinue en x

Théorème de Dirichlet vers^o uniforme

Soit f fonction périodique C^1 par morceaux et continue. Alors la série de Fourier associée à f converge ~~uniformément~~ normalement sur \mathbb{R} et $S_f = f$

Fonction de période T .

$$a_0 = \frac{1}{T} \int_0^T f(x) dx \quad a_n = \frac{2}{T} \int_0^T f(x) \cos(n\omega x) dx$$

$$b_n = \frac{2}{T} \int_0^T f(x) \sin(n\omega x) dx \quad \omega = \frac{2\pi}{T}$$

$$c_n = \frac{1}{T} \int_0^T e^{-in\omega x} f(x) dx$$

alors $u_n(x) = a_n \cos(n\omega x) + b_n \sin(n\omega x)$.

$$\text{et } S_f(x) = \sum_{n=-\infty}^{+\infty} c_n e^{-in\omega x}$$

Formule de Parseval

$$\frac{1}{T} \int_a^{a+T} |f(x)|^2 dx = \sum_{n=-\infty}^{+\infty} |c_n|^2$$

$$\text{soit } |a_n|^2 + \frac{1}{2} \sum (a_n^2 + b_n^2) = \frac{1}{T} \int_a^{a+T} |f(x)|^2 dx$$

Algebre Réduct d'endomorphismes

- $\left. \begin{array}{l} \vec{x} \text{ vecteur propre} \\ \lambda \text{ valeur propre} \end{array} \right\} \text{éléments propres.}$
ils sont tels que $u(x) \in \mathcal{L}_n(K)$ sat
 $u(x) = \lambda \vec{x}$.

Définit°: 0 est val prop de u si u n'est pas injective.

Définit°: à tt vecteur propre \vec{x} est associé une seule valeur propre λ .

à tte valeur propre λ est associée 1 espace propre
 $E_\lambda = \text{Ker}(u - \lambda \text{id})$
 $= \{0\} \cup \{\text{vecteurs propres associés à } \lambda\}$

Endomorphisme nilpotent:

$$\exists \lambda \in \mathbb{N}^* \setminus \{0,1\} \quad u^\lambda(x) = 0 \text{ et } u^{\lambda-1}(x) \neq 0$$
$$u^\lambda(x) = \lambda^\lambda x \Rightarrow \lambda^\lambda = 0$$
$$u^{\lambda-1}(x) = \lambda^{\lambda-1} x \Rightarrow \lambda \cdot \text{Spec } u = \{0\}$$

Diagonaliser - trigonaliser

Vecteur propre de $A: X$ tq. $AX = \lambda X$

λ valeur propre s'il existe X vect non nul tq $AX = \lambda X$

E_λ espace vectoriel propre associé à λ

$$E_\lambda = \text{Ker}(A - \lambda \text{In}).$$

Polynôme caractéristique $\chi_A(x) = \det(A - xI_n)$

$$\sum \lambda_i = \text{Tr } A$$

$$\prod \lambda_i = \det A$$

Deux matrices semblables ont un polynôme caractéristique

Valeurs propre : $\det(A - \lambda I_n) = 0$

espace propre E_λ : $\text{Ker}(A - \lambda I_n) \Rightarrow (A - \lambda I_n)(x) = 0$

$$\dim E_\lambda = \dim E - \text{rang}(E - \lambda I)$$

A est diagonalisable s'il existe 1 base B' de laquelle la matrice de A est diagonalisable.

A est diagonalisable (c'est 1 matrice carrée) : $A = PDP^{-1}$

Une CNS est que pour que A soit diagonalisable est qu'il existe une base de E formée des vecteurs propres.

Théorèmes de diagonalisation :

1^{er} théorème : A est diagonalisable, (A est matrice carrée d'ordre n) si elle a n valeurs propres distinctes.

on dit que le polynôme est scindé.

2^{ème} théorème : so $\text{Spec } u = \{\lambda_1, \dots, \lambda_p\}$, $\begin{cases} p \leq n \\ \lambda_i \neq \lambda_j \end{cases}$

u est diagonalisable si

$$1) E = E_{\lambda_1} \oplus \dots \oplus E_{\lambda_p} = \bigoplus_{i=1}^p E_{\lambda_i}$$

$$2) \dim E = \sum_{i=1}^p \dim E_{\lambda_i}$$

$$3) \dim E_{\lambda_i} = \nu_i \text{ (ordre de multiplicité de } \lambda_i)$$

Espaces préhilbertiens

Définition du produit scalaire

Soit un espace vectoriel réel
 $f: E \times E \rightarrow \mathbb{R}$ telle que
 $f(\vec{x}, \vec{y})$

a) f est bilinéaire
 $\left\{ \begin{array}{l} f_y(\vec{x}) \\ f_x(\vec{y}) \end{array} \right\}$ sont linéaires

b) symétrique
 $f(\vec{x}, \vec{y}) = f(\vec{y}, \vec{x}), \forall (\vec{x}, \vec{y}) \in E^2$

c) défini positif
 $\forall \vec{x} \in E, f(\vec{x}, \vec{x}) \geq 0$
et $f(\vec{x}, \vec{x}) = 0 \Rightarrow \vec{x} = 0$

Soit un espace vect. complexe
 $f: E \times E \rightarrow \mathbb{R}$ telle que
 $f(\vec{x}, \vec{y})$

a) f est linéaire à droite et
semi-linéaire à gauche

$$\begin{aligned} f(\vec{x}, \lambda \vec{y}) &= \lambda f(\vec{x}, \vec{y}) \\ f(\lambda \vec{x}, \vec{y}) &= \overline{\lambda} f(\vec{x}, \vec{y}) \end{aligned}$$

b) f est symétrique hermitien
 $f(\vec{x}, \vec{y}) = \overline{f(\vec{y}, \vec{x})}$

c) défini positif idem

Un espace préhilbertien est un espace vectoriel réel ou complexe muni d'un produit scalaire

forme polaire de q : $f(\vec{x}, \vec{y}) = \frac{1}{2}(q(\vec{x} + \vec{y}) - q(\vec{x}) - q(\vec{y}))$

Inégalité de Cauchy-Schwarz

$$|\langle \vec{x}, \vec{y} \rangle| \leq \|\vec{x}\| \|\vec{y}\|$$

Soit un espace préhilbertien, une norme est
 $\|\vec{x}\|^2 = \langle \vec{x}, \vec{x} \rangle$

Orthogonalité

2 vecteurs \vec{x} et \vec{y} sont orthogonaux si $\langle \vec{x}, \vec{y} \rangle = 0$

A 1 partie non vide de E . On appelle orthogonale de A la partie $A^\perp = \{ y \in E, \forall x \in A, \langle x, y \rangle = 0 \}$
 A^\perp est 1 SEV de E .

Def 4 : la famille (\vec{e}_i) est orthogonale si les vecteurs sont orthogonaux 2 à 2, c-à-d $\langle \vec{e}_i, \vec{e}_j \rangle = 0, i \neq j$
la famille (\vec{e}_i) est orthonormale si les vecteurs sont orthogonaux 2 à 2 unitaires $\langle \vec{e}_i, \vec{e}_j \rangle = \begin{cases} 0 & \text{si } i \neq j \\ 1 & \text{si } i = j \end{cases}$ / δ_{ij} symbole de Kronecker

Prop : Toute famille orthogonale de vect non null est 1 fam de base

pour 2 vecteurs orthogonaux $\|\vec{x} + \vec{y}\| = \|\vec{x}\| + \|\vec{y}\|$
de dim

Tout espace préhilbertien \checkmark fini admet 1 base orthog

Procédé de Gram-Schmidt : pour fabriquer une base famille orthogonale

$$\vec{f}_1 = \vec{e}_1 \quad \vec{f}_2 = \frac{\vec{f}_1 + \vec{e}_2}{\|\vec{f}_1 + \vec{e}_2\|} \quad \langle \vec{f}_1, \vec{f}_2 \rangle = 0 \Rightarrow \|\vec{f}_1\|^2 + \langle \vec{f}_1, \vec{e}_2 \rangle = 0$$

Théorème de la projection orthogonale

$(E, \langle \cdot, \cdot \rangle)$ espace préhilbertien, F_n une SEV de E de dim n :

a) F_n admet 1 supp ortho

b) $\forall \vec{x} \in E, \exists ! \vec{y}_n = p(\vec{x}) \in F_n$ tq $\|\vec{x} - p(\vec{x})\| = \inf \|\vec{x} - \vec{y}\|$
 $p(\vec{x})$ est la proj ortho de \vec{x} sur F_n et $\|\vec{x} - p(\vec{x})\| = d(\vec{x}, F_n)$

c) $d(\vec{x}, F_n) = \|\vec{x} - p(\vec{x})\|$ est la distance de \vec{x} à F_n

Espaces euclidiens

Une matrice carrée d'ordre n est orthogonale si

a) ${}^t A \cdot A = I$

d) A est la mat de pass d 1 base

b) $A^t A = I$

à 1 base ortho

c) $A^{-1} = {}^t A$

e) les vect ~~ortho~~ colonnes sont 2 à 2 orthog et norme 1

théorème fondamental

Soit A endo sym d'un espace euclidien E , de matrice A , symétrique réelle, soit B base ortho de E , alors

1) les valeurs propres de A (ou A) sont réelles

2) les espaces propres sont 2 à 2 orthog

3) A est diag d'une base orthonormale de vect propres

Isométrie de \mathbb{R}^3

Pour montrer que A est la matrice d'une rotation sectorielle on vérifie néc

$$A \in SO(\mathbb{R}^3) \implies A \text{ orthog}$$

$$\implies \det A = 1$$

b) on cherche les vecteurs invariants $AX = X$
d'où \vec{K} qui dirige l'axe de rotation.

c) pour trouver l'angle de rotation

$$\text{tr} A = 2 \cos \theta + 1 \quad \text{d'où } \cos \theta$$

$$\text{mais } (\vec{x}, \vec{u}(\theta), \vec{K}) = \sin \theta (x^2 + y^2) \text{ du signe de } \theta$$

si $\det A = -1$

$$\vec{K} \text{ tel que } AX = -X$$

$$\text{tr} A = -1 + 2 \cos \theta$$

GEOMETRIE

ENVELOPPES - DEVELOPPEES - DEVELOPPANTES

enveloppe $\begin{cases} f(x, y, t) = 0 \\ \frac{\partial f}{\partial t}(x, y, t) = 0 \end{cases}$

developpée 1) enveloppe des normales, $\int \vec{p} \cdot d\vec{\sigma} = 0$
 2) lieu geometrique des centres $\frac{d^2x}{dt^2}$ de courbure

developpante: $X = x + (K-s) \frac{dx}{ds}$ $Y = y + (K-s) \frac{dy}{ds}$

Repère de Frenet - courbure - Torsion

$r(t)$ \perp pt tri-régulier. le repère de Frenet est

$$\vec{T} = \frac{d\vec{\sigma}}{ds} \cdot \frac{dt}{ds}$$

$$\vec{N} = \frac{1}{\|\frac{d\vec{T}}{ds}\|} \frac{d\vec{T}}{ds}$$

$$\vec{B} = \vec{T} \wedge \vec{N}$$

$$\frac{d\vec{T}}{ds} = \frac{\vec{N}}{R}$$

$$\frac{d\vec{B}}{ds} = -\frac{\vec{N}}{\rho}$$

$$\frac{d\vec{N}}{ds} = -\frac{\vec{T}}{R} + \frac{\vec{B}}{\rho}$$

SURFACES

Points réguliers - bi-réguliers.

- pt régulier: $\vec{F}'(t_0) = \frac{d\vec{\sigma}}{dt} \neq 0$ ou $\frac{\partial \vec{\sigma}}{\partial u} \wedge \frac{\partial \vec{\sigma}}{\partial v} \neq 0$

- pt bi-régulier: $\vec{F}'(t_0) \wedge \vec{F}''(t_0) \neq 0$

- plan osculateur: défini par $\vec{\sigma}, \frac{d\vec{\sigma}}{dt}, \frac{d^2\vec{\sigma}}{dt^2}$
 si M n'est pas bi-régulier, le pla osculateur
 est dirigé par $\frac{d^2\vec{\sigma}}{dt^2}$ et $\frac{d^3\vec{\sigma}}{dt^3}$

plan osculateur défini par

$$\begin{vmatrix} x-x_0 & x'(t_0) & x''(t_0) \\ y-y_0 & y'(t_0) & y''(t_0) \\ z-z_0 & z'(t_0) & z''(t_0) \end{vmatrix}$$

Normale à \mathcal{M} : droite passant par \mathcal{M} \perp à la courbe
plan normale: plan contenant \mathcal{M} et \perp à la tge en \mathcal{M}

Normale principale: intersection du plan normal et du plan osculateur

lignes coordonnées: on fixe $u = u_0$ ou $v = v_0$

vecteur orientant la normale à la surface

$$\begin{cases} \frac{\partial \mathbf{OM}}{\partial u} \wedge \frac{\partial \mathbf{OM}}{\partial v} \\ \text{grad } f \end{cases}$$

équat cartésienne du plan tgt en $M_0 \in (S)$ (M_0 régulier)
 $M_0 \text{ grad } f = 0 \Rightarrow (x-x_0) \frac{\partial f}{\partial x} + \dots + (z-z_0) \frac{\partial f}{\partial z}$

CYLINDRE

$$\vec{PM} = \lambda \vec{w}$$

on appelle rayon cylindrique de direction \vec{x}
 et de génératrice \vec{w} la surface paramétrée
 définie par: $\vec{PM} = \lambda \vec{w} \quad \vec{w}(a, b, c)$

$$\begin{aligned} x &= a + \lambda a \\ &\vdots \\ &\vdots \end{aligned}$$

équat cartésienne de la forme $f(x, y) = 0$

Cylindre circonscrit

Définition géométrique: 1 surface (E) est circonscrite à (S) si les plans tgts à (E) et (S) sont confondus en tt pts de la courbe de contact

méthode: $\begin{cases} \vec{w} \text{ grad } f = 0 \\ f(x, y, z) = 0 \end{cases} \Rightarrow \text{courbe de contact ou contour apparent}$

puis $\vec{OA} = \lambda \vec{w}$.

on détermine λ en fonction de x, y, z

$$f(x, y, z) = 0 \Rightarrow f(x, y, z) = 0$$

CONE.

cône de sommet S et de directrice (γ) : la surface engendrée par les droites (génératrices) passant par S et s'appuyant sur (γ)

$$\vec{SH} = \lambda \vec{SA} \Rightarrow X = a(1-\lambda) + \lambda x$$

$$(t, \lambda) \rightarrow (1-\lambda)\vec{OA} + \lambda \vec{u}(H)$$

$$f(x, y, z) = 0 \quad \text{ou} \quad f(p, q, r) = 0$$

ce sont les équats d'un cône de sommet O .

$$f(p, q, r) = 0$$

c'est l'équat d'un cône de sommet S

Cône circonscrit

$$\vec{w} \cdot \begin{cases} \text{grad } f \cdot \vec{SA} = 0 \\ f(x, y, z) = 0 \end{cases} \Rightarrow \text{courbe de contact}$$

puis $\vec{SH} = \lambda \vec{SA}$: on détermine λ , et x , puis x, y, z

$f(x^2+y^2, z) = 0$
est l'équation d'une surface de révolution
d'axe (O, \vec{z})

$f(S, \rho)$ est l'équation d'une surface de révolution
d'axe passant par le centre de la sphère
et dirigée par la normale au plan

Quadriques

Sphères : $x^2 + y^2 + z^2 = r^2$
ellipsoïdes : $\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$

$$\begin{cases} x = a \cos u \sin v \\ y = b \sin u \sin v \\ z = c \cos v \end{cases}$$

Hyperboloïde à 1 nappes

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = 1$$

$$\begin{cases} x = a \cosh u \cos v \\ y = b \sinh u \cos v \\ z = c \sinh u \end{cases}$$

Hyperboloïde à 2 nappes : $\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = -1$

$$\begin{cases} x = a \cosh u \sin v \\ y = b \sinh u \sin v \\ z = \pm c \cosh u \end{cases}$$

paraboloides elliptique: $\frac{x^2}{a^2} + \frac{y^2}{b^2} = z$

$$x = a u \cos v$$

$$y = b u \sin v$$

$$z = u^2$$

paraboloides hyperbolique: $\frac{x^2}{a^2} - \frac{y^2}{b^2} = z$

$$\left\{ \begin{array}{l} x = \frac{a}{2}(u+v) \\ y = \frac{b}{2}(u-v) \\ z = uv \end{array} \right.$$

Cône: $\frac{x^2}{a^2} + \frac{y^2}{b^2} = \frac{z^2}{c^2}$

$$\left\{ \begin{array}{l} x = a u \cos v \\ y = b u \sin v \\ z = c u \end{array} \right.$$

cylindre parabolique: $y = x^2$

quadrique dégenérée en plan: $\frac{x^2}{a^2} - \frac{y^2}{b^2} = 0$

cylindre: $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$ (cylindre elliptique)

$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$ (cylindre hyperbolique)

Intersect de surfaces: ex: fenêtre de Bernoulli

projet sur 1 plan $P // \vec{a}$ à 1 direct \vec{w}

2 équats cartésiennes,

on écrit $\left\{ \begin{array}{l} \text{équats du cône} \\ 2 \text{ équats } f \text{ et } g \end{array} \right.$

en fonction de λ .

$$\left. \begin{array}{l} x_A = x_0 + \lambda a \\ \vdots \\ \vdots \end{array} \right\} \text{ or } \left. \begin{array}{l} f(x_0, y_0, z_0) = f(x_0 + \lambda a, \dots) \\ g(x_0, y_0, z_0) = g(x_0 + \lambda a, \dots) \end{array} \right\}$$

l'ort $(\Gamma) = (\mathcal{E}) \cap (\mathcal{E}')$ (\mathcal{E}) déf par $f(x, y, z) = 0$

(\mathcal{E}') $g(x, y, z) = 0$

la projet de (Γ) sur le plan xOy est obtenue en éliminant z entre les 2 équats

$$\text{soit } (\Gamma) = (\mathcal{E}') \cap (\mathcal{E}) \text{ est avec } \left\{ \begin{array}{l} (\mathcal{E}) = f(x, y, z) \\ (\mathcal{E}') = g(x, y, z) \end{array} \right.$$

alors (Γ) est aussi tracée sur $f(x, y, z) + \lambda g(x, y, z) = 0$

Contours apparent selon le direct $\vec{w} = (a, b, c)$

équa cartés $\left\{ \begin{array}{l} \vec{w} \cdot \text{grad } f = 0 \\ f(x, y, z) = 0 \end{array} \right.$

équa param: $\left\{ \begin{array}{l} \vec{w} \left(\frac{\partial x}{\partial u}, \frac{\partial y}{\partial u}, \frac{\partial z}{\partial u} \right) \\ f \end{array} \right. \Rightarrow u = \varphi(w)$

Contours apparent pour 1 observateur
situé en \mathcal{D} .

$$\begin{cases} \vec{DM} \cdot \vec{\text{grad}} f = 0 \\ f(x, y, z) = 0. \end{cases}$$

Lignes de niveau, ligne de pente:

ligne de niveau: sectⁿ de (E) par des plans \parallel à P

lignes de pente: lignes tracées sur (E) et qui sont
des trajectoires orthogonales aux lignes de niveau

Surfaces réglées - surfaces développables

une surface réglée est 1 surface géométrique rep par
 $\vec{F}(u, v) = \vec{r}(u) + v \vec{H}(u)$

surface développable: soit 1 surface réglée, elle
est développable si $(\vec{r}'(u), \vec{H}(u), \vec{H}'(u))$ forment 1
famille liée.

ex: $x = x(u) + v x'(u)$

$$y = y(u) + v y'(u)$$

$$z = z(u) + v z'(u)$$

INTEGRALES CURVILIGNES

INTEGRALES DE SURFACE

si $V(x,y)$ dérive d'un potentiel scalaire, $\vec{V} = \text{grad} U$
 $\int \vec{V}(x,y) \cdot d\vec{r} = U(B) - U(A)$

Formule de Green-Riemann:

A compair élémentaire, de frontière (Γ) orientée \oplus

si P et Q 2 fois C^1

$$\int P dx + Q dy = \iint \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) dx dy$$

Propriété: soit $w = P dx + Q dy$ forme diff C^1 sur D
n'est exacte si $\int w$ sur un chemin fermé γ de classe C^1 (par morceaux) est nulle

éléments d'aires d'un arc suppose homogène

$$x = \frac{f(t)}{g(t)} \quad \text{avec} \quad \frac{ds}{dt} = \sqrt{x'(t)^2 + y'(t)^2}$$

$$dA(\varepsilon) = \iint d\vec{\sigma} = \iint \frac{\partial \vec{\sigma}}{\partial u} \wedge \frac{\partial \vec{\sigma}}{\partial v} du dv$$

$$\text{ou} \quad d\vec{\sigma} = \sqrt{1 + f_u'^2 + f_v'^2} du dv$$

2^{ème} théorème de Pappus

$$A = 2\pi \int r ds = 2\pi r_c L$$

Flux, angle solide

~~Φ~~

Pour 1 morceau de surface orientée, \vec{V} champs de vecteurs défini et continu sur 1 ouvert contenant (ε)

$$\Phi_{\varepsilon}(\vec{v}) = \iint \vec{v} \cdot \vec{n} \, d\sigma = \iint \left(\frac{\partial x}{\partial u} \frac{\partial y}{\partial v} \right) du dv$$

$$\vec{n} = \frac{\partial \vec{M}}{\|\partial \vec{M}\|} \text{ ds le cas d'une sphère}$$

angle solide : flux ds lequel on voit (ε) d'un pt A :
flux du champ de vecteurs $\vec{v} \rightarrow \vec{v}(M) = \frac{\vec{r}}{\|\vec{r}\|^3}$

Formule de Stokes

Soit ε un morceau de surface (ε) délimité par une courbe fermée, ce qui ouvre la surface. Soit \vec{v} un champ de vecteurs de classe C^1 sur tout ouvert contenant (ε)

$$\text{alors } \int_{\partial \varepsilon} \vec{v}(M) \cdot d\vec{\sigma} = \iint_{\varepsilon} \text{rot } \vec{v}(M) \cdot \vec{n} \, d\sigma$$

1) ds le plan, on utilise la formule de Green-Riemann
si le champ dérive d'un potentiel scalaire, $\int \vec{v}(M) \cdot d\vec{\sigma} = 0$

3) si le champ \vec{w} dérive d'un pot vecteur, $\vec{w} = \text{rot } \vec{V}(M)$
 $\int_{\partial \varepsilon} \vec{w}(M) \cdot d\vec{\sigma} = \iint \vec{v} \cdot d\vec{\sigma}$

Théorème d'Ostrogradsky

Soit ε un compact élémentaire de \mathbb{R}^3 délimité par une surface orientée et fermée, et soit $\vec{v}(M)$ un potentiel vecteur de classe C^1 sur ε ouvert contenant ε

$$\iint \vec{v} \cdot \vec{n} \, d\sigma = \iiint \text{div } \vec{v} \, dx dy dz$$

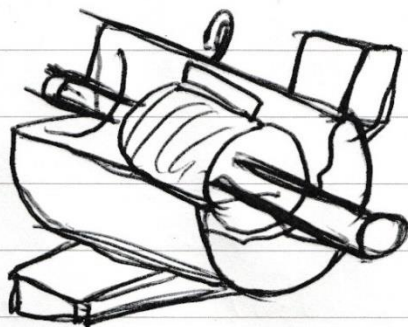
1) si le champ dérive d'un pot vecteur $\text{div } \vec{v} = 0$

Fredéric Gilet

22/10/2017

ELECTROTECHNIQUE

Etude des moteurs electriques



Termes : η : rendement

P_u : puissance utile (W)

P_e : puissance absorbée (W)

C : couple (N.m)

I) Introduction

Les moteurs sont de trois types principaux :

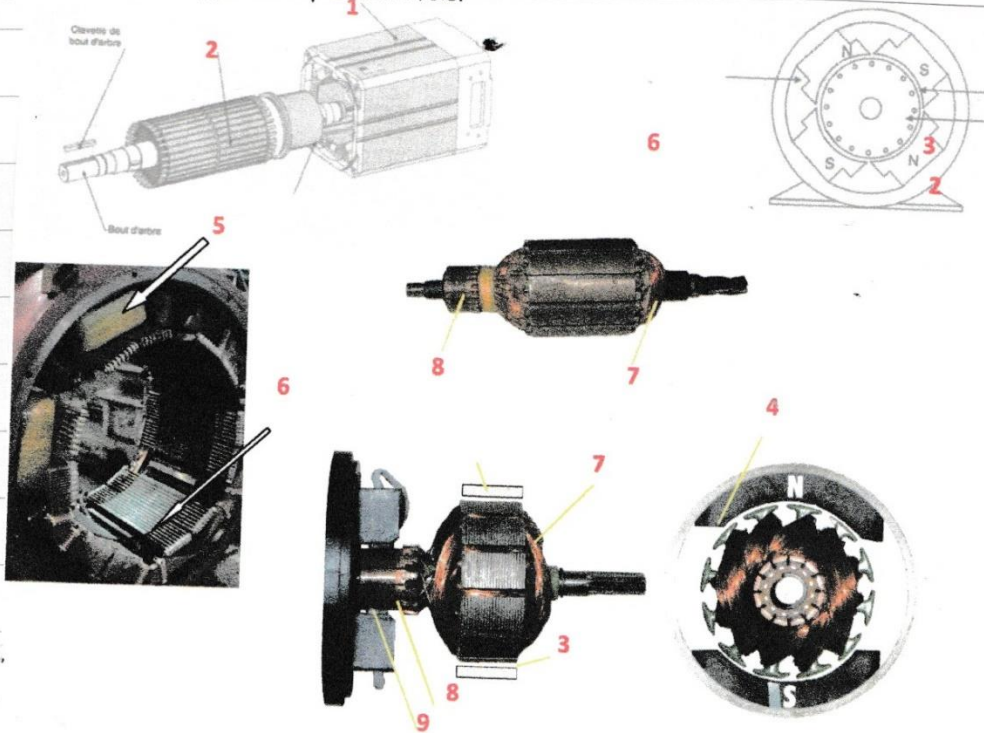
- Moteur à courant continu
- Moteur synchrones triphasés
- Moteur asynchrone

Nous étudierons uniquement le moteur à courant continu et le moteur asynchrone.

	Le type de courant qui alimente l'induit	Le type de courant qui alimente l'inducteur	Où se trouve l'induit et l'inducteur	L'inducteur peut-il ou est-il à aimants permanents
Asynchrone triphasé		Alternatif	Induit = R / Induc = S	NON
Courant continu	Continu	Continu	Induit = R / Induc = S	OUI
Asynchrone monophasé		Alternatif	Induit = R / Induc = S	NON
Pas à Pas		Continu	Induit = R / Induc = S	NON
Synchrone	Alternatif	Continu	Induit = S / Induc = R	OUI
Universel	Alternatif	Alternatif	Induit = R / Induc = S	NON

II) Le moteur à courant continu

Le numéro peut apparaître plusieurs fois.



- 1) Partie fixe, le stator ou inducteur
- 2) Partie mobile - le rotor ou induit
- 3) Entree: separation
- 4) aimant ou ferite ou Si bobines inductrices
- 5) conducteurs logés dans des encoches (induit)
- 6) collecteur
- 7) balais

Permettent de relier le circuit électrique de l'induit à celui extérieur et le machine les balais frottent sur les lames des collecteurs → courant électrique → induit tourne

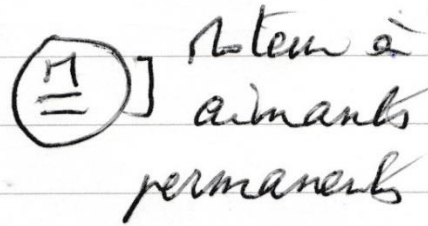
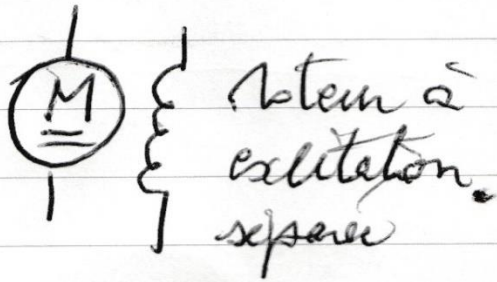
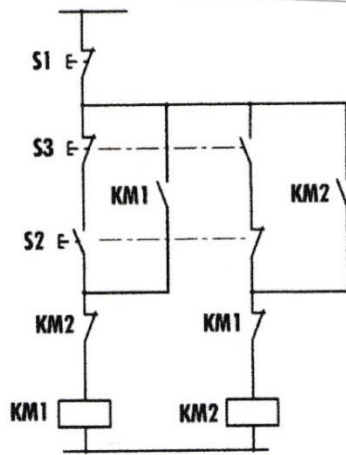
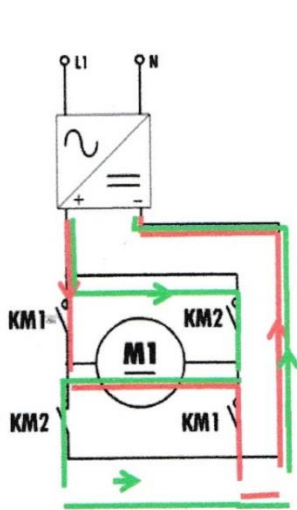


Schéma de commande et de puissance
 Commander une machine à courant continu
 vitesse fixe avec des contacteurs (KM)



Quel est le rôle de S1, S2 et S3 ?

S1 : Bouton poussoir arrêt

S2 : Bouton poussoir pour commander KM1 (sens 1)

S3 : Bouton poussoir pour commander KM3 (sens 2)

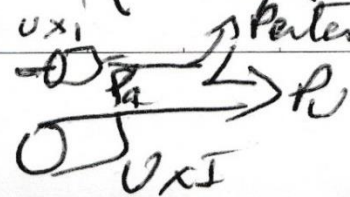
Bilan de puissance d'un moteur à courant continu

$P = U \times I$ aux bornes d'alimentation du rotor $C \cdot \Omega = \frac{2\pi N}{60}$

$P = U \times I$ aux bornes du stator $P_u = C \times \Omega$

$\eta = \frac{P_u}{P_a} = \frac{C \times \Omega}{(U \times I) + (U \times I)}$ $P_a = (U \times I) + (U \times I)$

rendement



Identification et comptement d'une machine à courant continu

tension d'excitation.

Couple nominal

Puissance mécanique développée

LR 57008		LEROY SOMER		2 102 451 / A			
IEC 34.1.1990		MADE IN FRANCE					
MOTEUR A COURANT CONTINU DIRECT CURRENT MOTOR							
TYPE: LSK 1604 S 02 N° 700000/10 9/1992 M 249 kg							
Classe / Ins class H		IM 1001		IP 23 IC 06			
M _{max} / Rated torque 301 N.m		Altit. 1000 m		Temp. 40 °C			
	kW	min ⁻¹	V	A	V	A	
Nom./Rat.	36,3	1150	440	95,5	360	3	
	3,63	115	44	95,5	360	3	
	36,3	1720	440	95,5	240		
T		Système d'armature: I		Induit / Arm.		Exct. / Field	
○ Service / Duty S1		DE 6312 2RS C3		NDE 6312 2RS C3		○	

Vitesse de rotation nominale

Tension nominale aux bornes de l'induit

Courant nominal dans l'induit

Puissance électrique absorbée

$$P_a = (U \times I) + (U \times i) = (440 \times 95,5) + (360 \times 3)$$

Rendement = 43100 W

$$\eta = \frac{P_m}{P_a} = \frac{36,3}{43,1} = 0,842$$

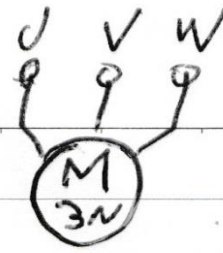
Couple P_m 43,1

$$P_m = C \times \omega \text{ donc } C = \frac{P_m}{\omega} = \frac{36300}{120,43} = 301,4 \text{ N.m}$$

couple où $\omega = \frac{2\pi \times 1150}{60}$

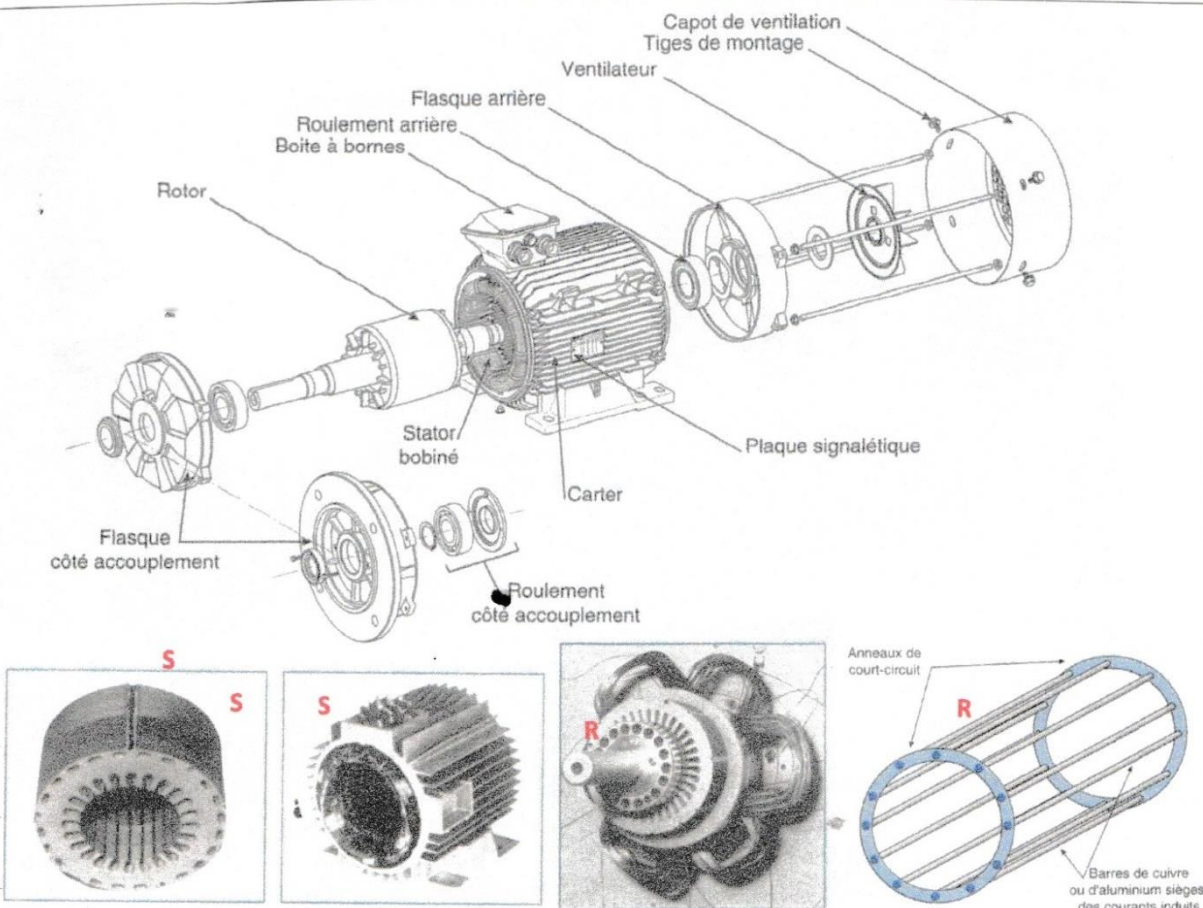
La puissance est directement proportionnelle à la vitesse

Moteur asynchrone



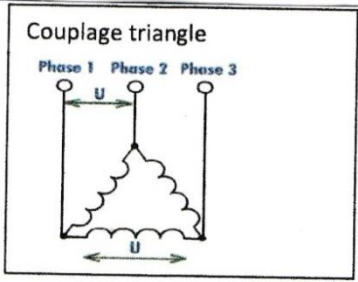
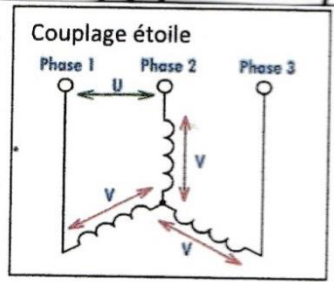
Le stator est composé de 3 bobines qui créent trois champs magnétiques variables et tournants qui entraînent le rotor en rotation par le mécanisme d'une force électro-magnétique.

L'inducteur se trouve au rotor R
L'inducteur se trouve au stator S



Le rotor est une cage composée de 2 anneaux et de barres en aluminium

* Détermination de couplage



La Tension est de 400 V



couplage en étoile

LEROY-SOMER MOT. 3 ~ JS 100 L
N° 152/50 21 kg

Code : T

IP 55	i cl. F	40°C	S1	%	c/h
Δ 380	50	2840	3	0,89	6,4
Δ 400	50	2860	3	0,83	6,3
Δ 415	50	2870	3	0,79	6,7

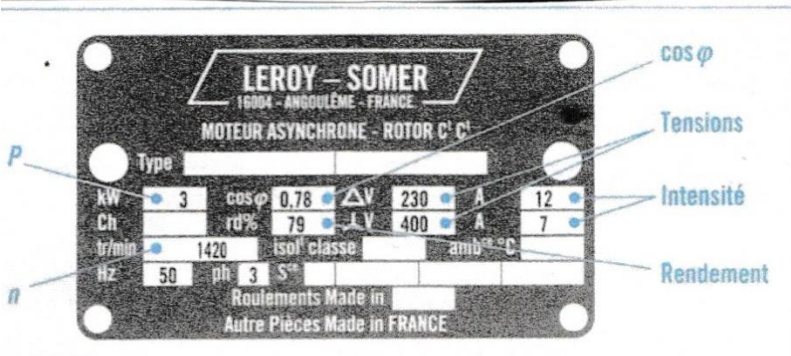
DF 6206 C3
NDE 6205 Z2C3

TEC INT. 102

couplage en triangle

Bien tourner dans le sens contraire, il faut inverser deux phases de réseau d'alimentation.

Calcul d'un moteur triphasé



Attention la puissance donnée sur une plaque signalétique est toujours la puissance mécanique utile

Calculer la puissance absorbée

$$P_a = U \times I \times \sqrt{3} \times \cos \varphi = 400 \times 7 \times \sqrt{3} \times 0,78 = 3,7 \text{ kW}$$

ou

$$P_a = \frac{P_u}{\eta} = \frac{3,7}{0,79} = 3,7 \text{ kW}$$

Couple nominal de la machine

$$P_u = C \times \omega \Rightarrow C = \frac{P_u}{\omega} = \frac{3000}{148,7} = 20,17 \text{ Nm}$$

$$\omega = \frac{2\pi \times n}{60} = \frac{2\pi \times 1420}{60} = 148,7$$

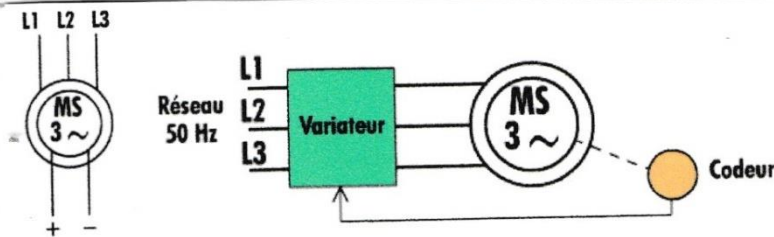
La tension entre phase est de 400V donc il devra être couplé en étoile.

Le moteur synchrone

L'inducteur alimenté en courant continu est au rotor, le stator est alimenté par un courant alternatif. Si ce moteur est entraîné par une turbine, c'est un alternateur.

Le moteur ne peut pas démarrer seul.

Lorsque l'inducteur est à aimant permanent, que le stator est alimenté par un variateur commandé par un capteur placé sur l'arbre (codeur) on parle d'un moteur brushless (sans balais).



1) Introduction

Le but de ce TP est d'étudier et de réaliser des traitements de gestion des modes de marche sur API.

Pour ce faire nous sommes partis du GEMMA (Guide d'Etude des Modes de Marches et d'Arrêts) d'une poinçonneuse de porte d'armoires. Il décrit les différentes étapes et transitions du fonctionnement de cette machine, qu'elle soit en P.C. hors énergie, en procédures d'arrêt de la partie opérative (remise en route ou arrêt), en procédures en défaillance de la PO ou en procédures de fonctionnement (mise en ou hors service ou essais et vérifications).

Nous avons réalisé à partir de ce schéma avec les étapes et les transitions les différents GRAFCETS : GS (Graphique de sécurité), GRAS (Graphique de Reprise après Arrêt de Sécurité), le GC (Graphique de Conduite), le GFN (Graphique de Fonctionnement Normal).

A partir de ces graphiques le but de ce TP est d'entrer dans l'automate Siemens les différents GRAFCETS (en ladder, en SCL ou directement en grafcet) et de programmer une IHM Nous réalisons ainsi la partie commande.

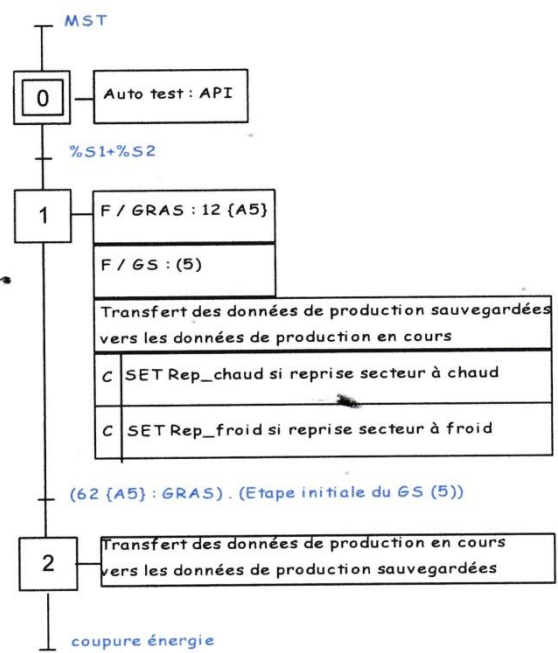
Ce compte-rendu définit le paramétrage de l'automate et de l'IHM, contient les blocs de programmes et de données ainsi que les variables API et IHM et explique le fonctionnement global du système et ses particularités.

IV - 5 - 1 - Grafset de reprise secteur (GRS)

VERSION du 13 mai 2003

GRAFSET de reprise secteur (GRS)

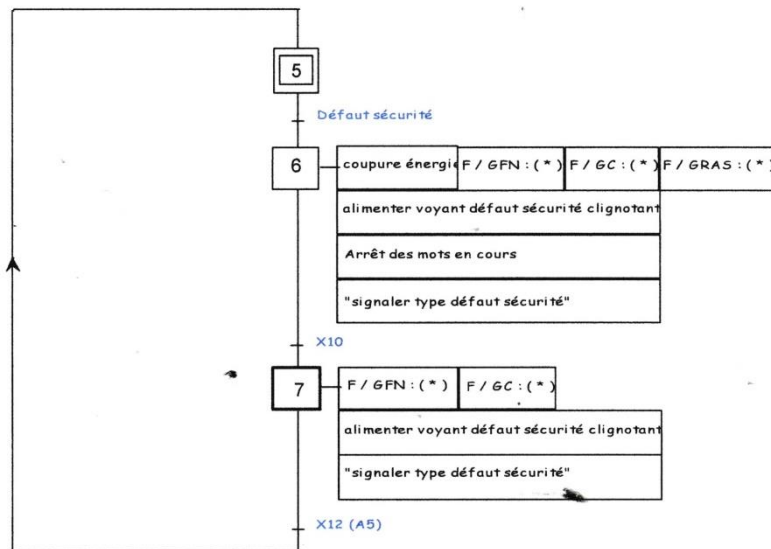
Point de vue PO , Spécifications fonctionnelles



IV - 5 - 2 - Grafcet de sécurité (GS)

VERSION du 13 mai 2003

GRAF CET de sécurité (GS) point de vue PO , spécifications fonctionnelles



Défaut sécurité = (ARU 1 + ARU 2) + (Carter 1 + Carter 2) + défaut réseau ethernet (X74)
 + défaut réseau unitelway (contrôle autorisation Magelis)
 + défaut hydraulique + Alarme + défaut armoire + défaut index



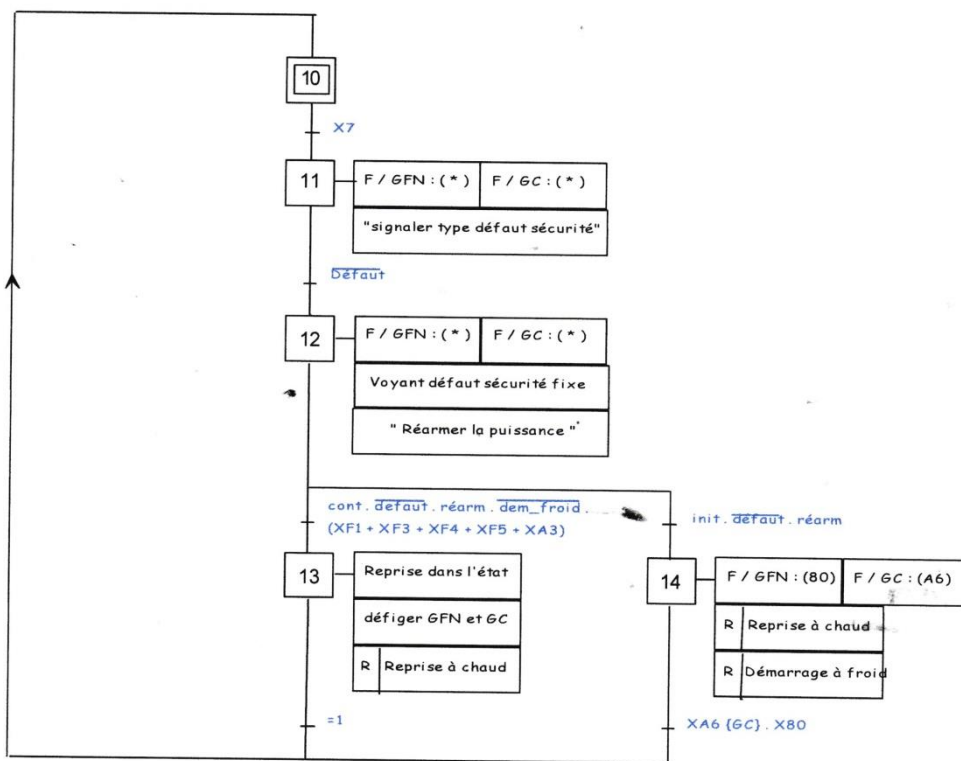
POINÇONNEUSE DE PORTES



IV - 5 - 3 - Grafset de Reprise après Arrêt Sécurité (GRAS)

VERSION du 13 mai 2003

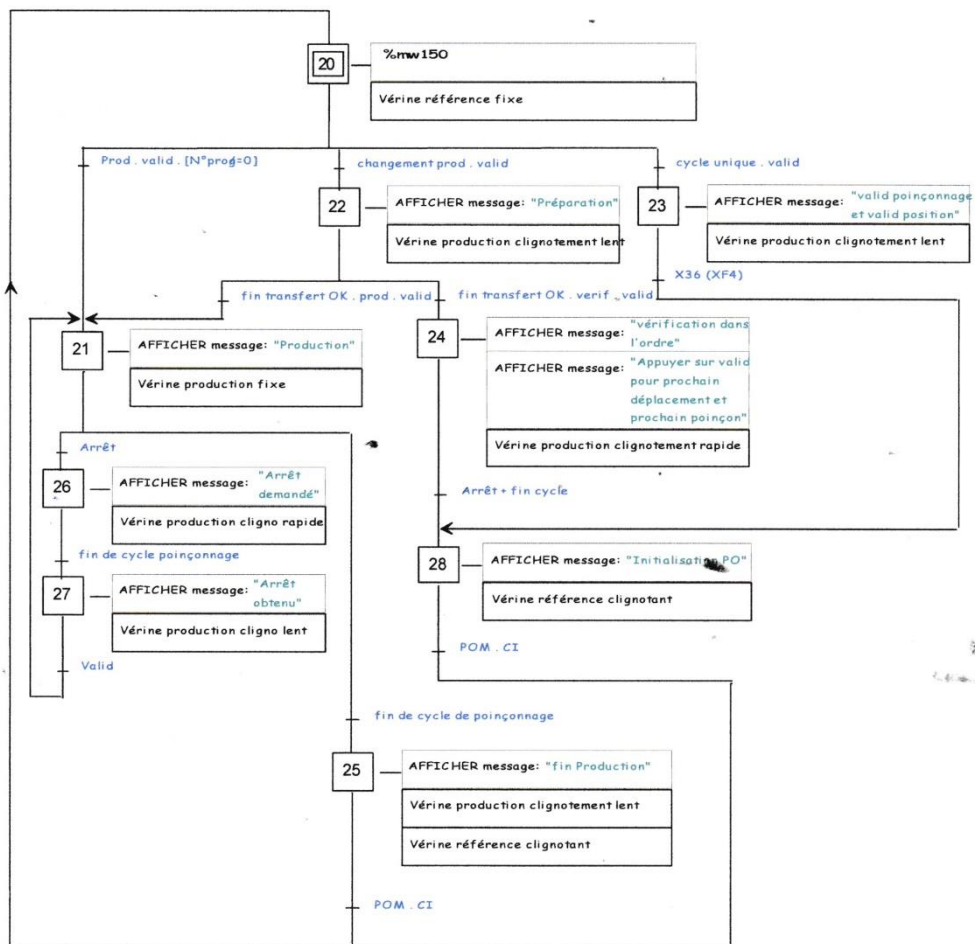
GRAFSET de reprise après arrêt secteur (GRAS) point de vue PO , spécifications fonctionnelles



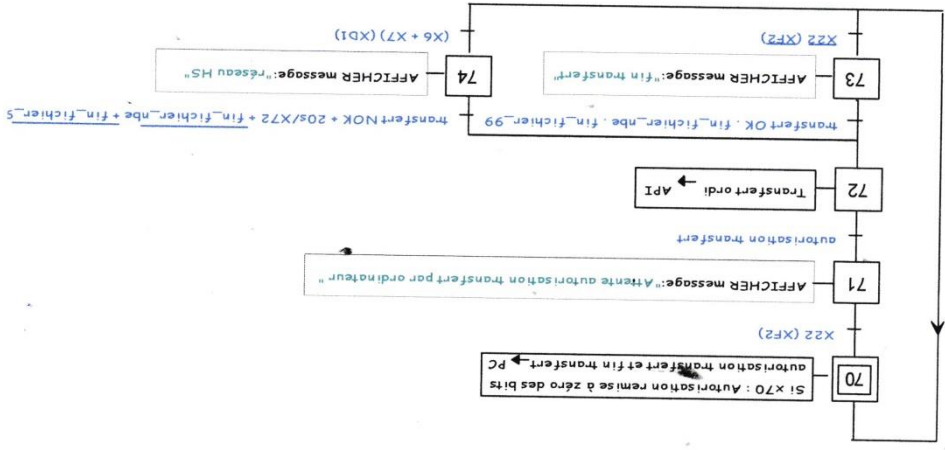
IV - 5 - 4 - Grafset de Conduite (GC)

VERSION du 13 mai 2003

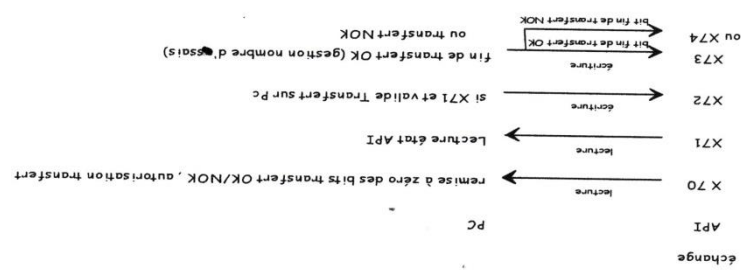
GRAFSET de conduite (GC) Point de vue PC , spécification fonctionnelles



GRAFSET d'autorisation de transfert
programme par réseau ethernet (F2)
transfert prog : Gestion plan de poinçonnage



fin_fichier_nbre OK si Debut_adresse[200+3x(Nbre_poinc)]/100 = Nbre_poinc
 fin_fichier_99 OK si Debut_adresse[200+3x(Nbre_poinc)]/100 = 99

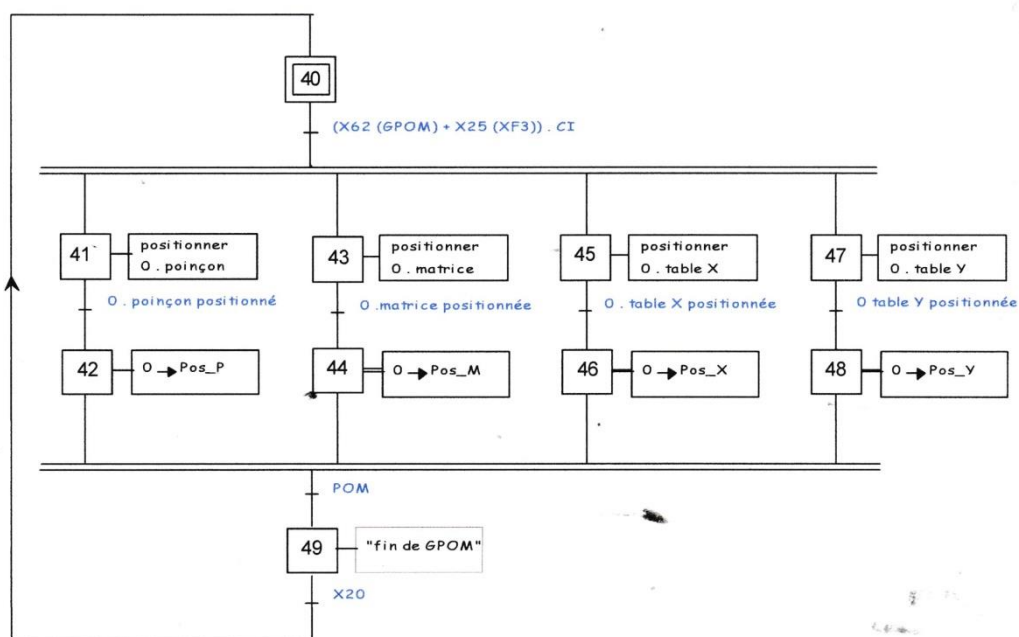


remarque
 le transfert du PC vers l'API, met les mots restants à 0 (de %mw200+? à %mw502)

IV – 5 – 6 – Grafcet de Prise Origine Machine (GPOM)

VERSION du 13 mai 2003

GRAF CET de prise d'origine machine (GPOM)
 point de vue PO , spécifications fonctionnelles



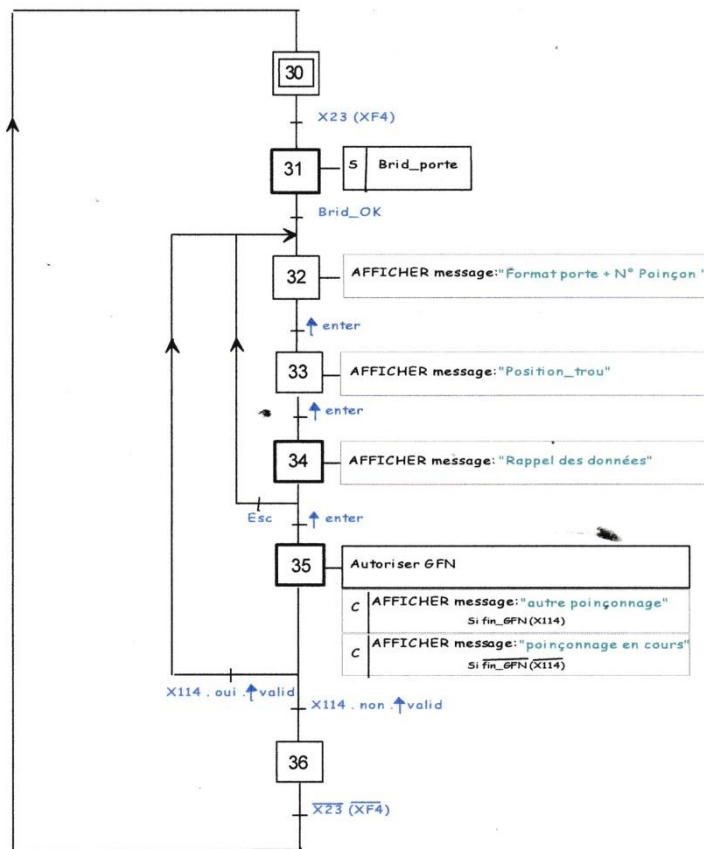
CI = déveteisseur rentré . poinçon rentré . matrice rentré . poin_ini . matr_ini

POM = [Pos_P=0] . [Pos_M=0] . [Pos_X=0] . [Pos_Y=0]

IV - 5 - 7 - Graficet de Marche Manuelle (GF4)

VERSION du 13 mai 2003

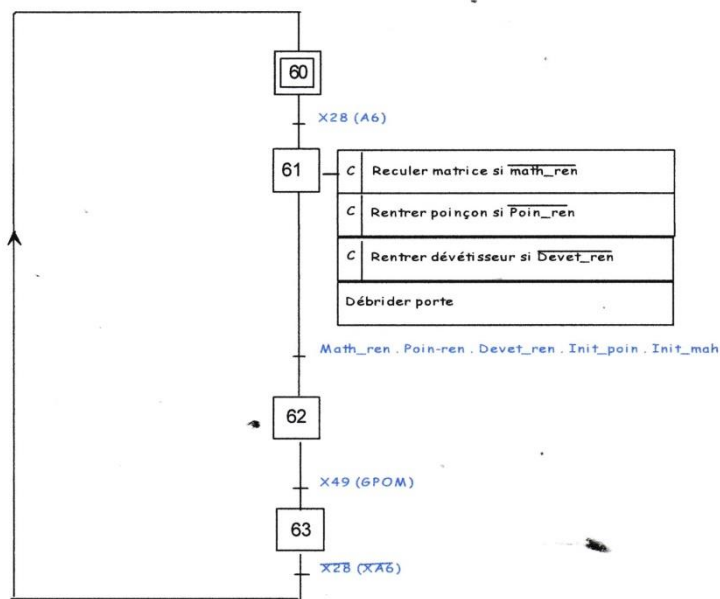
GRAFICET de marche manuelle (F4) Point de vue PO , spécification fonctionnelles



IV - 5 - 8 - Grafcet d'Initialisation (GA6)

VERSION du 13 mai 2003

GRAF CET d'initialisation (A6) Point de vue PO , Spécifications fonctionnelles

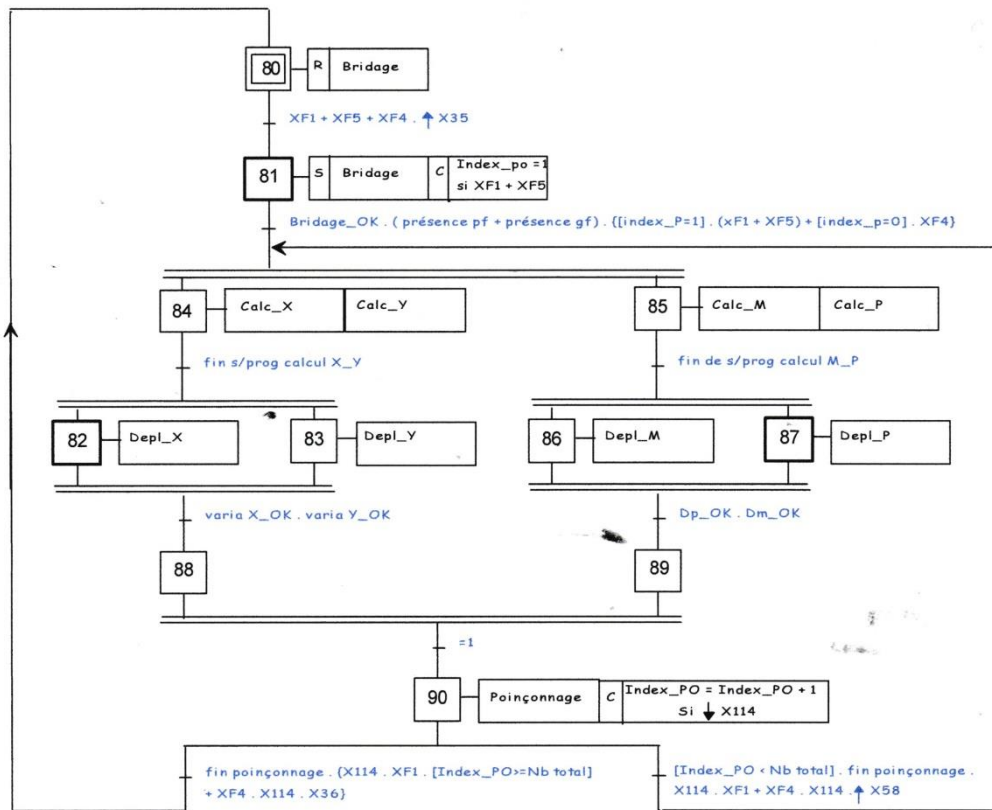


CI = dévétisseur rentré . poinçon rentré . matrice rentré . init_poin . init_mah

IV – 5 –9 – Grafset de Fonctionnement Normale (GFN)

VERSION du 13 mai 2003

GRAFÇET de fonctionnement normal (GFN)
 point de vue PO , spécifications fonctionnelles





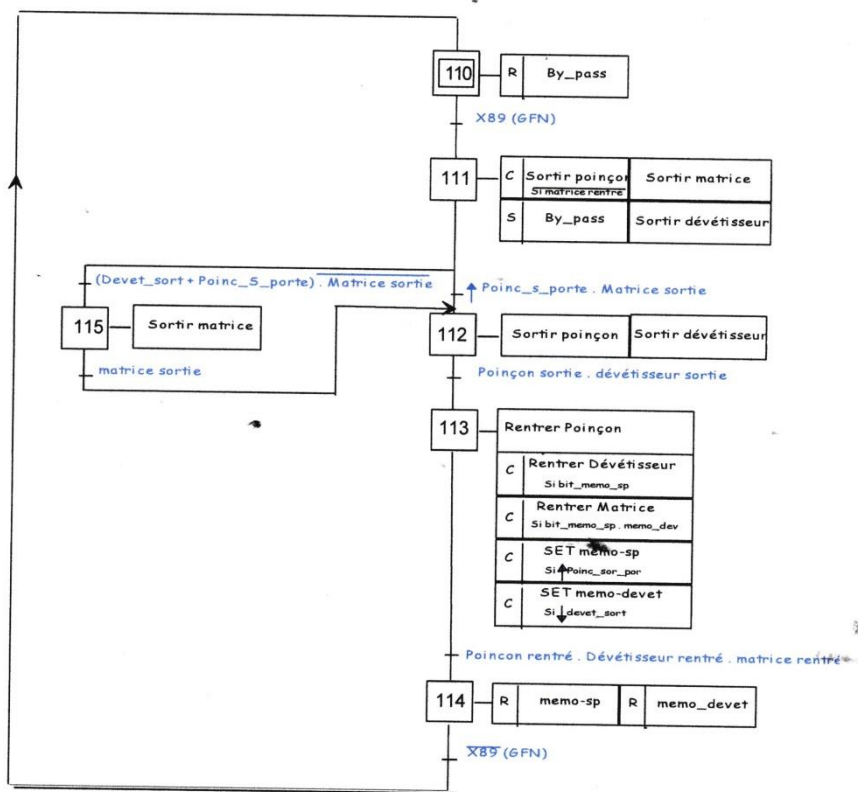
POINÇONNEUSE DE PORTES



IV - 5 - 10 - Grafcet de la tâche 1 (GT1)

VERSION du 13 mai 2003

GRAF CET de Tâche 1 (GT1) Point de vue PO , spécification fonctionnelles Tâche poinçonnage



IV - 5 - 11 - Grafcet de la tâche 2 (GT2)

Le bridage se fait actuellement manuellement, donc pas de grafcet, en attente de modification si l'automatisation est demandée par le client.



GRETANANTES INDUSTRIE greta.industrie@ac-nantes.fr
16, Rue Dufour - BP 94225 - 44042 Nantes cedex 1 - Tél. 02 40 14 56 56
www.greta-paysdelaloire.fr - N° SIRET : 194 400 297 00025 - Code APE : 8539A





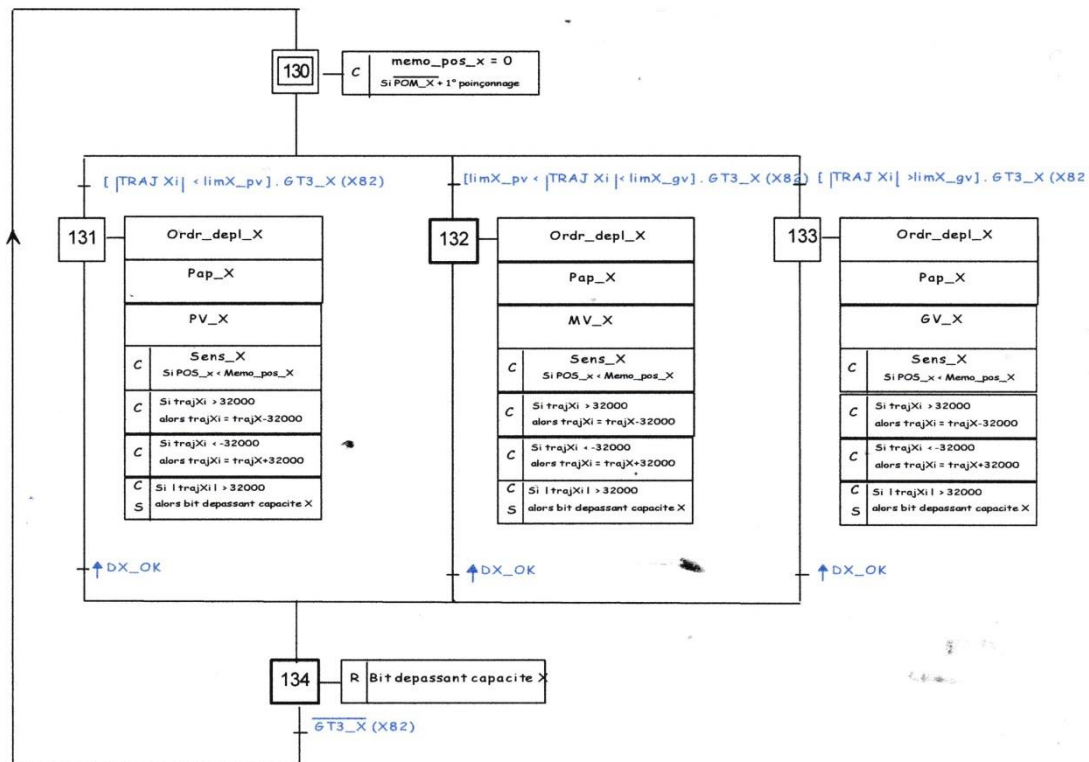
POINÇONNEUSE DE PORTES



IV - 5 - 12 - Graficet de la tâche 3 pour l'axe X (GT3X)

VERSION du 13 mai 2003

GRAFICET de tâche 3 (GT3X) point de vue PO , spécifications fonctionnelles Positionnement X



$limX_pv =$
 $limX_gv =$

$POS_X = POS_X$ à atteindre en 1/10 mm
 $Memo_pos_X = POS_X$ antérieur
 $Traj_Xi = Nbre pas moteur X$



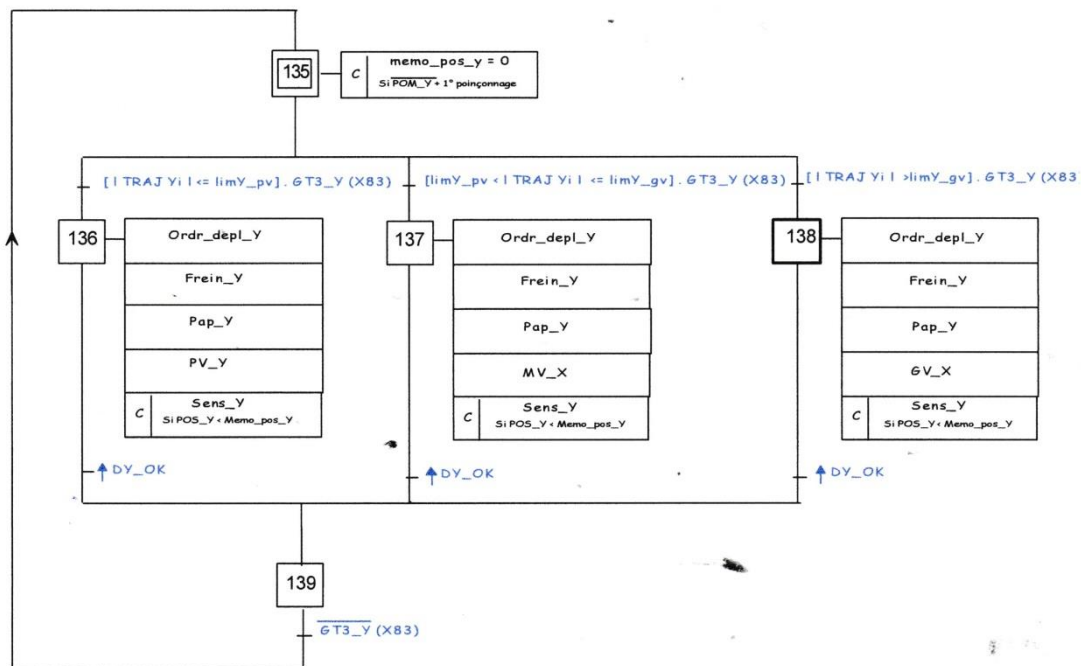
GRETANANTES INDUSTRIE greta.industrie@ac-nantes.fr
16, Rue Dufour - BP 94225 - 44042 Nantes cedex 1 - Tél. 02 40 14 56 56
www.greta-paysdelaloire.fr - N° SIRET: 194 400 297 00025 - Code APE: 8559A



IV - 5-13 - Grafset de la tâche 3 pour l'axe Y (GT3Y)

VERSION du 13 mai 2003

GRAFSET de tâche 3 (GT3Y) point de vue PQ, spécifications fonctionnelles Positionnement Y



limY_pv =
limY_gv =

POS_Y = POS_Y à atteindre en 1/10 mm
Memo_pos_Y = POS_Y antérieur
Traj_Yi = Nbre pas moteur Y

IV – 5 –14 – Grafçet de la tâche 4 (GT4)

A – Déplacement matrice

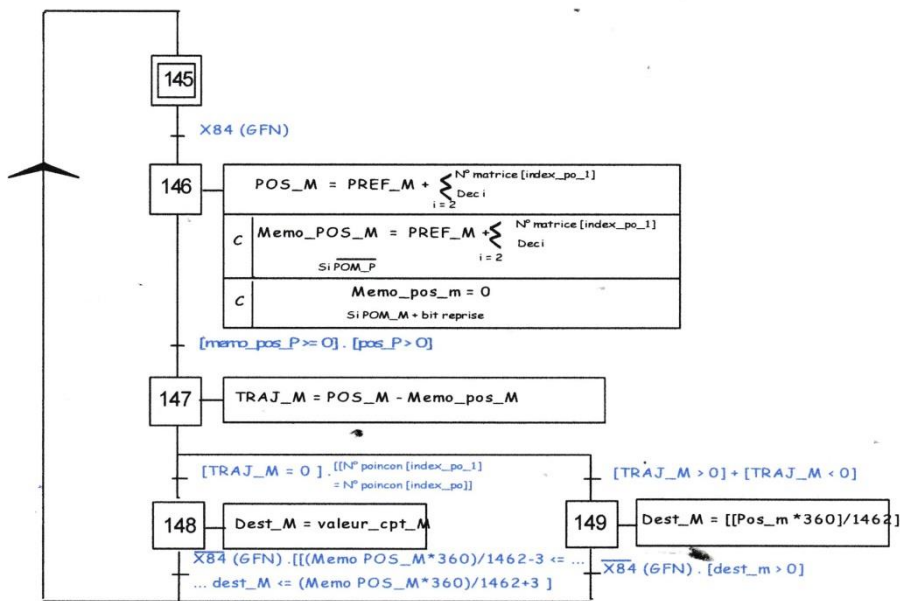
VERSION du 13 mai 2003

GRAFÇET de tâche 4 (GT4)

point de vue PO , spécifications fonctionnelles

Calcul déplacement Matrice,Poinçon

Matrice



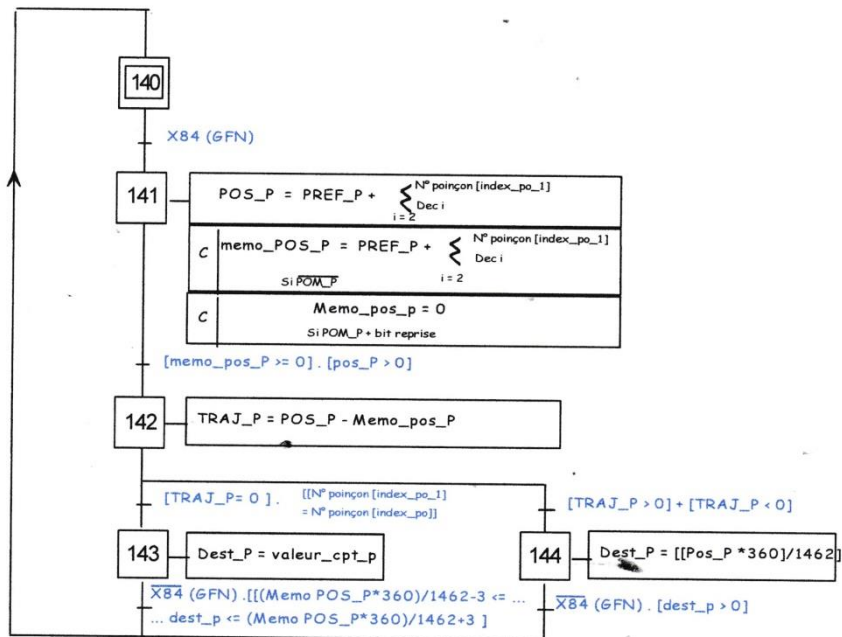
POS_M = POS_M à atteindre en 1/10 mm
 Memo_pos_M = POS_M antérieur
 Traj_M = Nbre_point_M

B – Déplacement poinçon

VERSION du 13 mai 2003

GRAFCEET de tâche 4 (GT4)
 point de vue PO , spécifications fonctionnelles
 Calcul déplacement Matrice,Poinçon

Poinçon

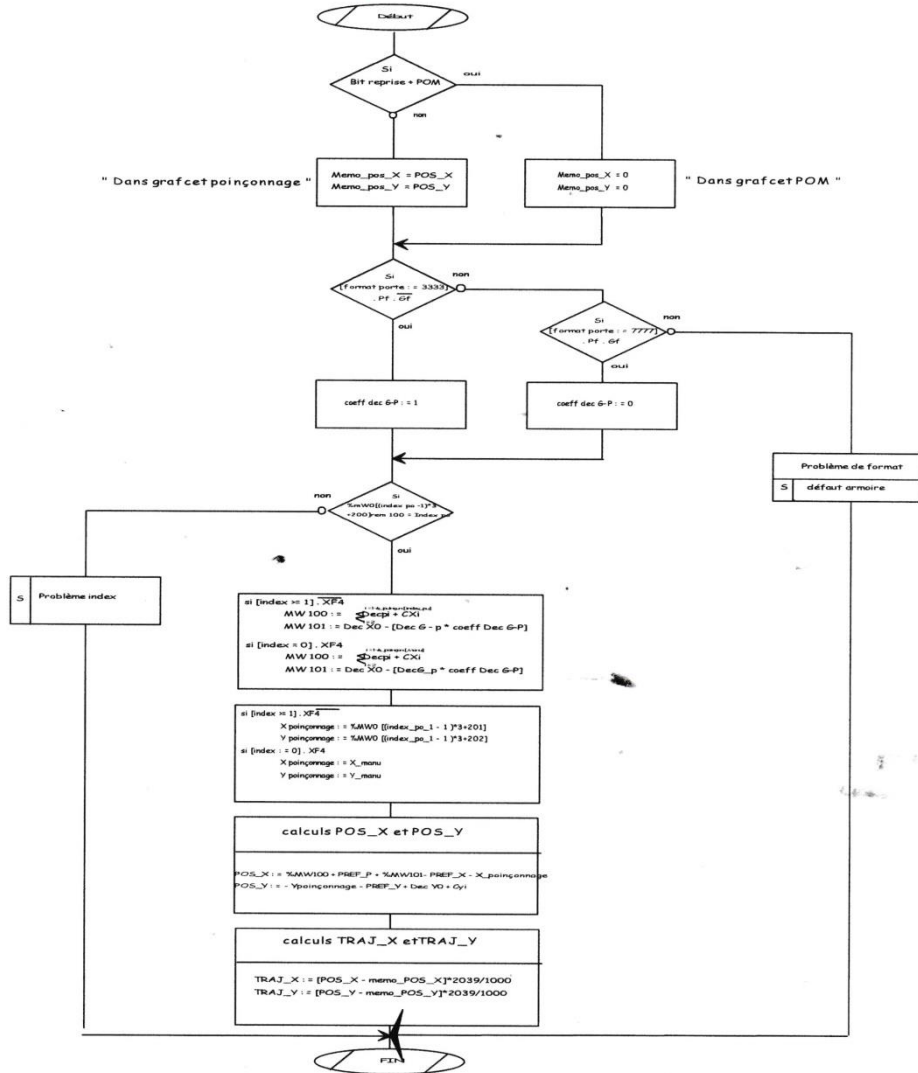


POS_P = POS_P à atteindre en 1/10 mm
 Memo_pos_P = POS_P antérieur
 Traj_P = Nbre_point_P

IV - 5 - 15 - Grafcet de la tâche 5 (GT5)

VERSION du 13 mai 2003

ORGANIGRAMME de tâche 5 (GT5)
 point de vue PO : spécifications fonctionnelles
 Calcul déplacement axes X et Y



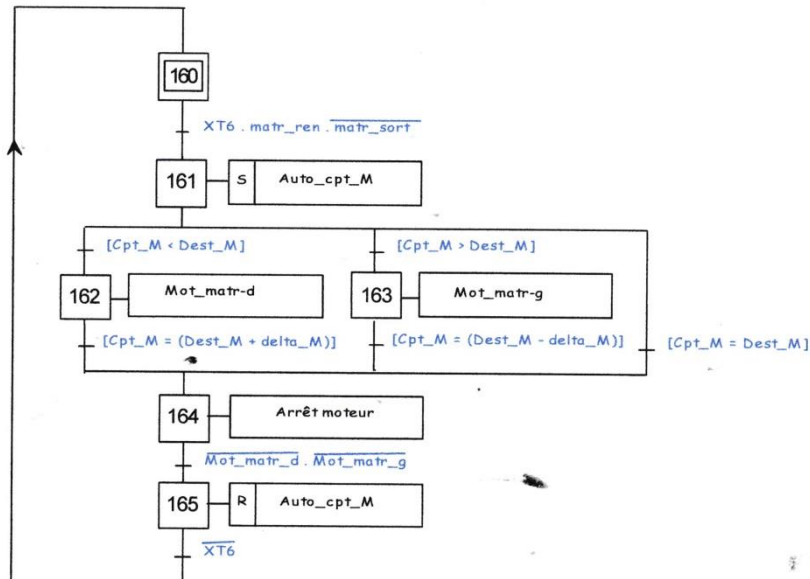
IV - 5 - 16 - Grafcet de la tâche 6 (GT6)

VERSION du 13 mai 2003

GRAF CET de tâche 6 (GT6)

point de vue PO , spécifications fonctionnelles

Positionnement matrice



Cpt_M : capteur codeur Matrice (comptage ↑ si Mot_Mat_g)
(comptage ↓ si Mot_mat_d)

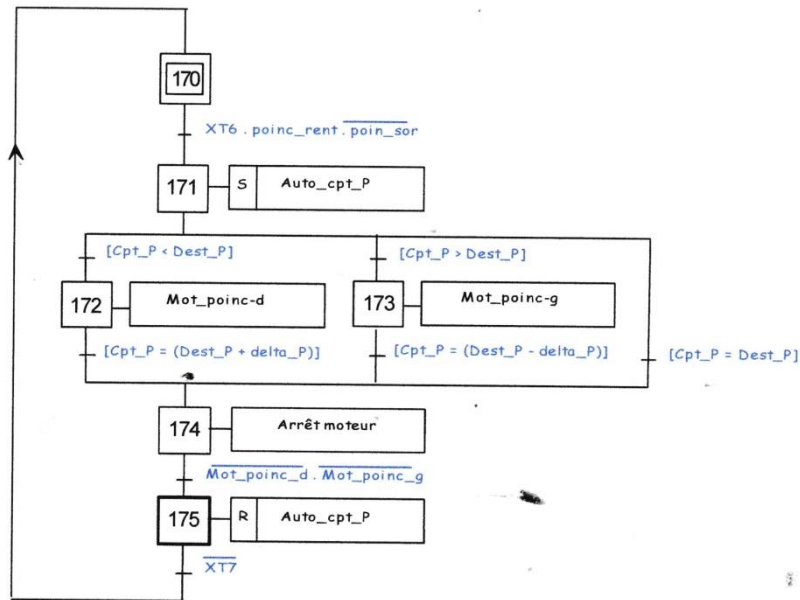
Dest_M : destination M en point

delta_M : distance d'arrêt matrice en point

IV - 5 - 17 - Grafcet de la tâche 7 (GT7)

VERSION du 13 mai 2003

GRAF CET de tâche 7 (GT7)
 point de vue PO , spécifications fonctionnelles
 Positionnement poinçon



Cpt_P : capteur codeur Poinçon (comptage ↑ si Mot_Poin_g)
 (comptage ↓ si Mot_Poin_d)

Dest_P : destination P en point

delta_P : distance d'arrêt poinçon en point

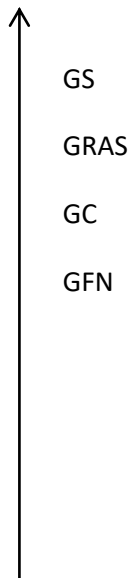
II) Cahier des charges

a) *Type*

Le matériel est un API SIEMENS SIMATIC S7-1500.
Le logiciel utilisé est Portal TIA V12 Siemens
L'API a pour adresse IP 192.168.21.228
L'IHM a pour adresse IP 192.168.21.237

b) *Réalisation à effectuer*

Il s'agit d'entrer les différents Grafquets suivants dans l'automate avec le langage approprié



Pour le GS et le GRAS on choisit le langage SCL
Pour le GC et le GFN on choisit d'entrer directement le grafcet
Pour la gestion des verrines et autres on utilisera le ladder

III) Initialisation du matériel

IV) GS et GRAS

Les défauts de sécurité peuvent être externes ou internes.

Externes :

- Arrêt d'urgence
- Thermique (moteur)
- Carter, capteurs de porte
- Barrière immatérielle
- Pressostat (pneumatique, hydraulique)

Internes (Analyse programmée de situations dangereuses)

- Surveillance des chaînes fonctionnelles
- Etat des capteurs
- Contrôle de couples

Dans le GEMMA, on décrit les procédures spécifiques de coupures et de démarrage, avec possibilité de récupération des données sauvegardées.

Le Grafcet de sécurité GS décrit les spécifications fonctionnels du point de vue PO.

Il décrit les cases D1 et D2.

Le Grafcet de reprise après arrêt secteur GRAS décrit les spécifications fonctionnelles de reprise après arrêt secteur du point de vue PO.

Il décrit dans ce cas particulier les cases D2, A5, A6 et A7.

A noter que le circuit de sécurité de la machine se déclenche indépendamment de l'automate et que lorsqu'une procédure en défaillance de la PO est détectée depuis tous les états on active dans l'automate le GS pour gérer ce défaut de sécurité puis le GRAS pour gérer le redémarrage secteur.

Le but est d'écrire le GS et le GRAS en SCL

Nous utilisons pour cela la méthode asynchrone.

Nous décrivons les conditions d'évolutions $CE(i)_{i \rightarrow (i+1)}$ et les activités des étapes $Et(i)$ comme suit :

IF...	CASE... OF...	FOR... TO DO...	WHILE... DO...	(*...*)
-------	------------------	--------------------	-------------------	---------

```

1 // Condition evolution GS
2 "Def_interne":="AlwaysFALSE";
3 "Def_secu":="Def_interne" OR "Aru";
4 "CE5_6":="Et5" AND "Def_secu";
5 "CE6_7":="Et6" AND "Prise_compte" AND "Et10";
6 "CE7_5":="Et7" AND "Et13";
7 // Bit etape du GS
8 "Et5":=("CE7_5" OR "Et5" AND NOT("CE5_6")) OR "FirstScan";
9 "Et6":=("CE5_6" OR "Et6" AND NOT("CE6_7")) AND NOT("FirstScan");
10 "Et7":=("CE6_7" OR "Et7" AND NOT("CE7_5")) AND NOT("FirstScan");
11 // Condition evolution GRAS
12 "CE10_11":="Et10" AND "Et7";
13 "CE11_12":="Et11" AND "Acquit" AND NOT("Def_secu");
14 "CE12-13":="Et12" AND "Rearm";
15 "CE13_16":="Et13" AND "Init_PO" AND NOT("Def_secu");
16 "CE13_17":="Et13" AND "Cont" AND NOT("Def_secu");
17 "CE16_10":="Et16" AND "BD_G7_GC".A6.X;
18 "CE17_10":="Et17" AND "Valid";
19 //Bit etape GRAS
20 "Et10":=((("CE16_10" OR "CE17_10" OR "Et10" AND NOT("CE10_11")) OR "FirstScan") OR "Et6");
21 "Et11":=((("CE10_11" OR "Et11" AND NOT("CE11_12")) AND NOT("FirstScan")) AND NOT("Et6"));
22 "Et12":=((("CE11_12" OR "Et12" AND NOT("CE12-13")) AND NOT("FirstScan"))AND NOT("Et6"));
23 "Et13":=((("CE12-13" OR "Et13" AND NOT("CE13_16") AND NOT("CE13_17")) AND NOT("FirstScan"))AND NOT("Et6"));
24 "Et16":=((("CE13_16" OR "Et16" AND NOT("CE16_10")) AND NOT("FirstScan"))AND NOT("Et6"));
25 "Et17":=((("CE13_17" OR "Et17" AND NOT("CE17_10")) AND NOT("FirstScan"))AND NOT("Et6"));
26
27

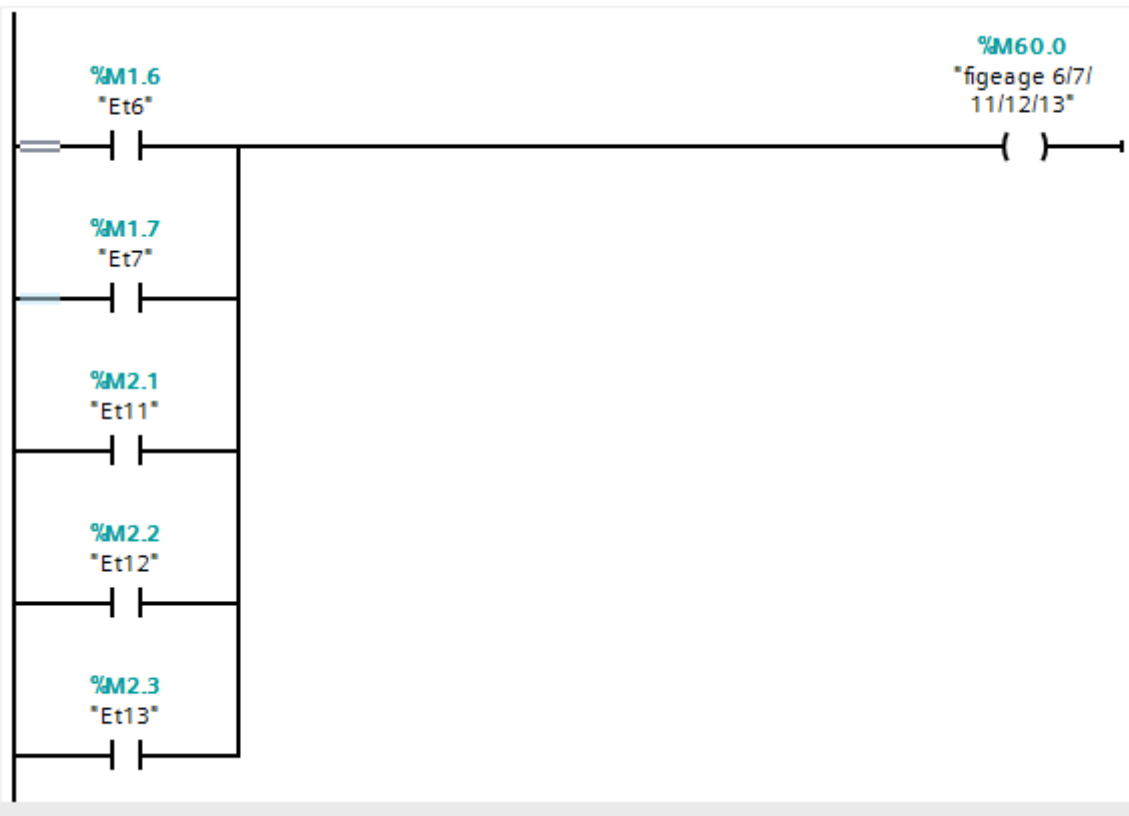
```

Par exemple, l'étape 5 « Et5 » n'est valide que si la transition d'avant CE7_5 est vraie ou que l'on est dans cette étape et qu'on ne passe pas à la condition suivante CE5_6 ou que l'on est à l'init_PC.

Par exemple, la condition d'évolution CE5_6 n'est valide que si on est à l'étape 5 et que l'on a la condition Def_secu

Forçage

Lorsqu'on est étape 6 du Graphique de sécurité (défaut de sécurité), on force le GRAS à l'étape initiale (10) Ainsi, la sécurité est toujours valide même pendant la reprise après défaut sécurité.



Lorsqu'on est dans les étapes 6 ou 7 ou 11 ou 12 ou 13 définies plus haut, on valide un bit de figeage. Dans le GC et le GFN, à chaque transition, on vérifie ce bit. S'il est à 0, le GFN ou le GC ont un fonctionnement normal. S'il est valide (défaut sécurité ou reprise après arrêt sécurité) le GFN ou le GC sont figés à l'étape où ils sont rendus puisqu'il n'y a plus les conditions de transition.

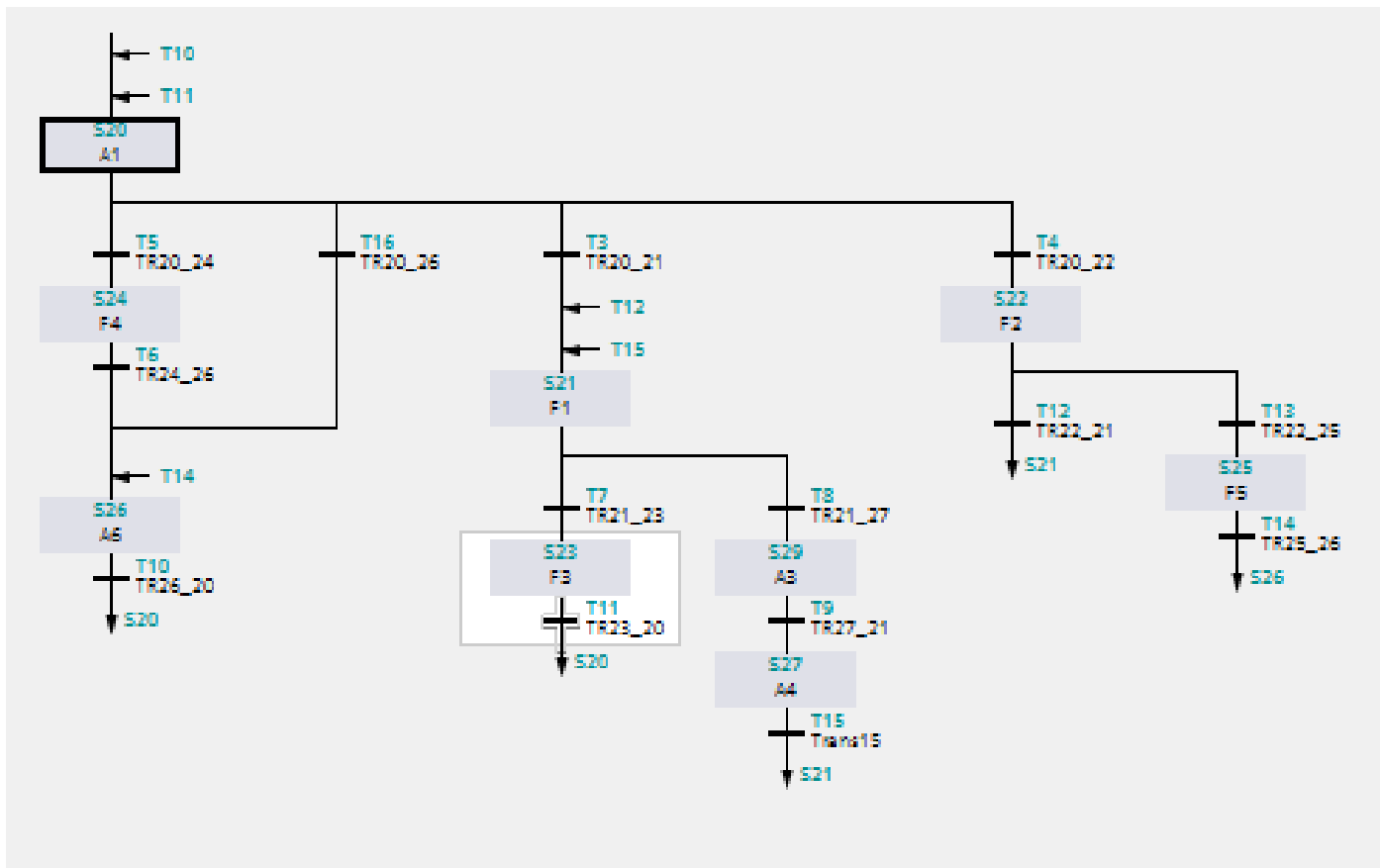
VJGC

Le Graphique de Conduite est la description de la vie de la machine hors gestion de la mise hors/sous tension et procédures de sécurité.

Il comprend les cases suivantes :

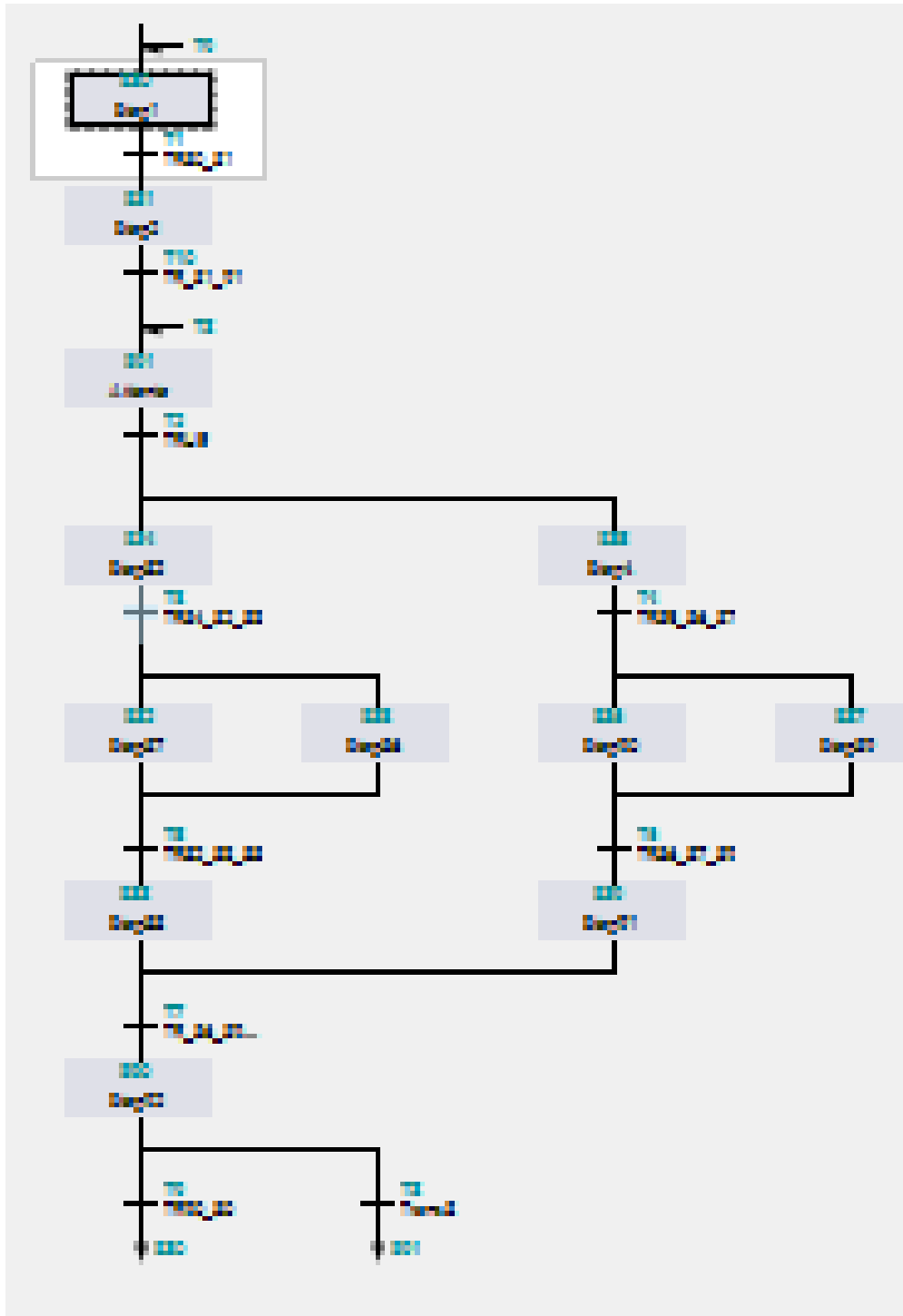
- A6 : Mise P.O. dans l'état
- A1 : Arrêt dans état initial
- A3 : Arrêt demandé dans un état déterminé
- A4 : Arrêt obtenu
- F1 : Fonctionnement normal
- F2 : Marche de préparation
- F3 : Marche de clôture
- F4 : Marche de vérification dans le désordre
- F5 : Marche de vérification dans l'ordre

Toutes ces procédures interagissent entre elles par des événements. Voici la description du Graphique de Conduite GC qui décrit ces étapes et ces transitions comme prévu dans le GEMMA.



VI) GFN

Le GFN est le Graphique de Fonctionnement Normal. Il décrit encore plus précisément la procédure F1 : Production normale.



VII) Réalisation de l'IHM

II) Variables,

Le tableau des variables est le suivant :

Variables API								
	Nom	Table des variables s..	Type de données	Adresse	Réma...	Visibl...	Acces...	Cc
1	Et5	Table de variables s..	Bool	%M1.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	Et6	Table de variables s..	Bool	%M1.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	Et7	Table de variables s..	Bool	%M1.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	Et10	Table de variables s..	Bool	%M2.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	Et11	Table de variables s..	Bool	%M2.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	Et12	Table de variables s..	Bool	%M2.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	Et13	Table de variables s..	Bool	%M2.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	Et16	Table de variables s..	Bool	%M2.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	Et17	Table de variables s..	Bool	%M2.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	CE5_6	Table de variables s..	Bool	%M3.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
11	CE6_7	Table de variables s..	Bool	%M3.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
12	CE7_5	Table de variables s..	Bool	%M3.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
13	CE10_11	Table de variables s..	Bool	%M4.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
14	CE11_12	Table de variables s..	Bool	%M4.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
15	CE12-13	Table de variables s..	Bool	%M4.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
16	CE13_16	Table de variables s..	Bool	%M4.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
17	CE13_17	Table de variables s..	Bool	%M4.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
18	CE16_10	Table de variables s..	Bool	%M4.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
19	CE17_10	Table de variables s..	Bool	%M4.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
20	Clock_Byte	Table de variables s..	Byte	%MB10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
21	Clock_10Hz	Table de variables s..	Bool	%M10.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
22	Clock_5Hz	Table de variables s..	Bool	%M10.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
23	Clock_2.5Hz	Table de variables s..	Bool	%M10.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
24	Clock_2Hz	Table de variables s..	Bool	%M10.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
25	Clock_1.25Hz	Table de variables s..	Bool	%M10.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
26	Clock_1Hz	Table de variables s..	Bool	%M10.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
27	Clock_0.625Hz	Table de variables s..	Bool	%M10.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
28	Clock_0.5Hz	Table de variables s..	Bool	%M10.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
29	System_Byte	Table de variables s..	Byte	%MB11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
30	FirstScan	Table de variables s..	Bool	%M11.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
31	DiagStatusUpdate	Table de variables s..	Bool	%M11.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
32	AlwaysTRUE	Table de variables s..	Bool	%M11.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
33	AlwaysFALSE	Table de variables s..	Bool	%M11.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
34	Aru	Table de variables s..	Bool	%I0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
35	W_PO	Table de variables s..	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
36	Prise_compte	Table de variables s..	Bool	%M100.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
37	Acquit	Table de variables s..	Bool	%M100.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
38	Rearm	Table de variables s..	Bool	%I0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
39	Cont	Table de variables s..	Bool	%M100.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
40	Valid	Table de variables s..	Bool	%M100.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

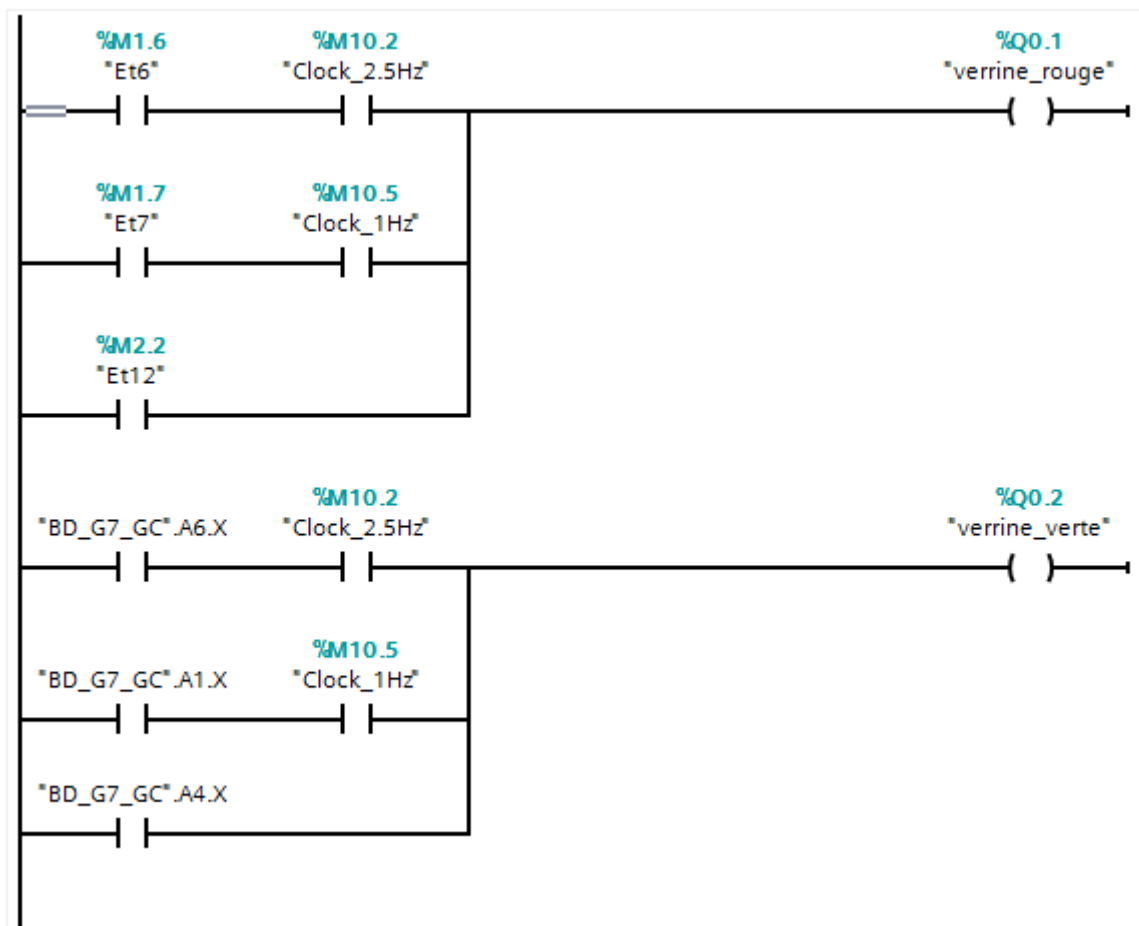
Variables API								
	Nom	Table des variables	Type de données	Adresse	Réma...	Visibl...	Acces...	Commentaire
39	Cont	Table de variables s...	Bool	%M100.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
40	Valid	Table de variables s...	Bool	%M100.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
41	Init_PO	Table de variables s...	Bool	%M100.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
42	Def_interne	Table de variables s...	Bool	%M0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
43	Def_secu	Table de variables s...	Bool	%M0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
44	x26_a_modifier	Table de variables s...	Bool	%M50.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
45	Tag_1	Table de variables s...	Bool	%Q0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
46	POM	Table de variables s...	Bool	%M20.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
47	CI	Table de variabl... ▼	Bool	%M20.1 ▼	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
48	cycle_unique_valid	Table de variables s...	Bool	%M100.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
49	prod_valid	Table de variables s...	Bool	%M101.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
50	chgt_prod_valid	Table de variables s...	Bool	%M101.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
51	arret	Table de variables s...	Bool	%M101.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
52	fin_cycle	Table de variables s...	Bool	%M101.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
53	fin_transfert	Table de variables s...	Bool	%M101.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
54	prod_verif	Table de variables s...	Bool	%M101.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
55	var-fet16	Table de variables s...	Bool	%M40.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
56	verrine_rouge	Table de variables s...	Bool	%Q0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
57	verrine_verte	Table de variables s...	Bool	%Q0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
58	verrine_bleue	Table de variables s...	Bool	%Q0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
59	figeage 6/7/11/12/13	Table de variables s...	Bool	%M60.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
60	index	Table de variables s...	Int	%MW200	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
61	nb_poinconnages	Table de variables s...	Int	%MW202	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
62	bridage_ok	Table de variables s...	Bool	%M150.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
63	fin_sprogX_Y	Table de variables s...	Bool	%M150.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
64	fin_sprogM_P	Table de variables s...	Bool	%M150.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
65	variaX_ok	Table de variables s...	Bool	%M150.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
66	variaY_ok	Table de variables s...	Bool	%M150.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
67	dp_ok	Table de variables s...	Bool	%M150.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
68	dm_ok	Table de variables s...	Bool	%M150.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
69	fin_poinconnage	Table de variables s...	Bool	%M150.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
70	poubelle	Table de variables s...	Bool	%M40.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
71	var_f_valid	Table de variables s...	Bool	%M40.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
72	page_affiche	Table de variables s...	Word	%MW154	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
73	<Ajouter>				<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

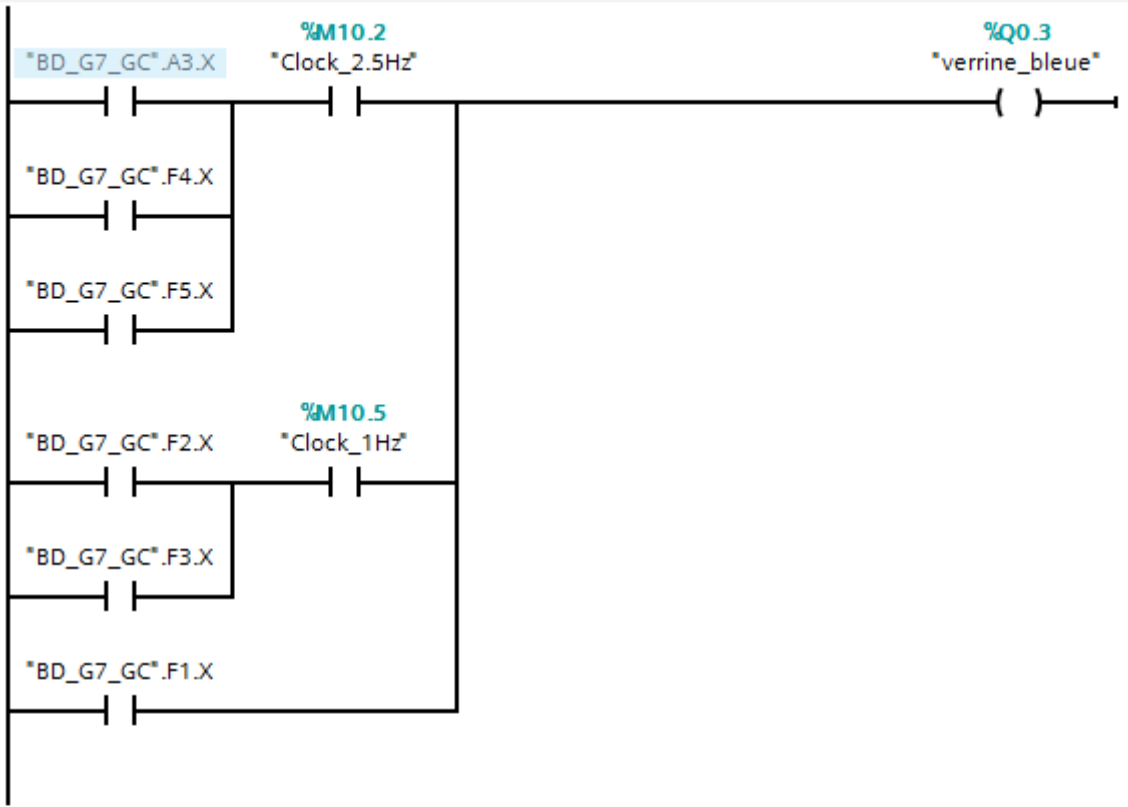
Verrines

Il y a trois couleurs de verrines correspondant chacune a un type d'activité :

- Rouge : Arrêt sécurité
- Vert ; Remise en route ou arrêt
- Bleue : Production

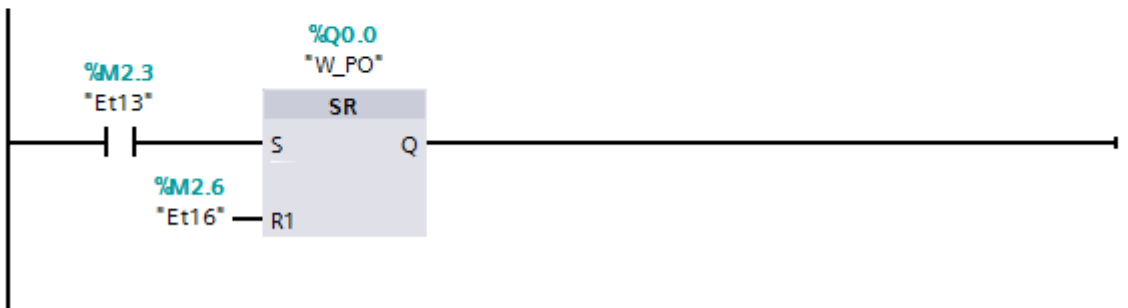
Selon l'étape du GEMMA, elles n'ont pas la même fréquence de clignotement. On utilisera donc des variables d'horloge de la CPU pour ajuster cette fréquence.





Réseau 3 :

Commentaire



VII) Réalisation de l'IHM

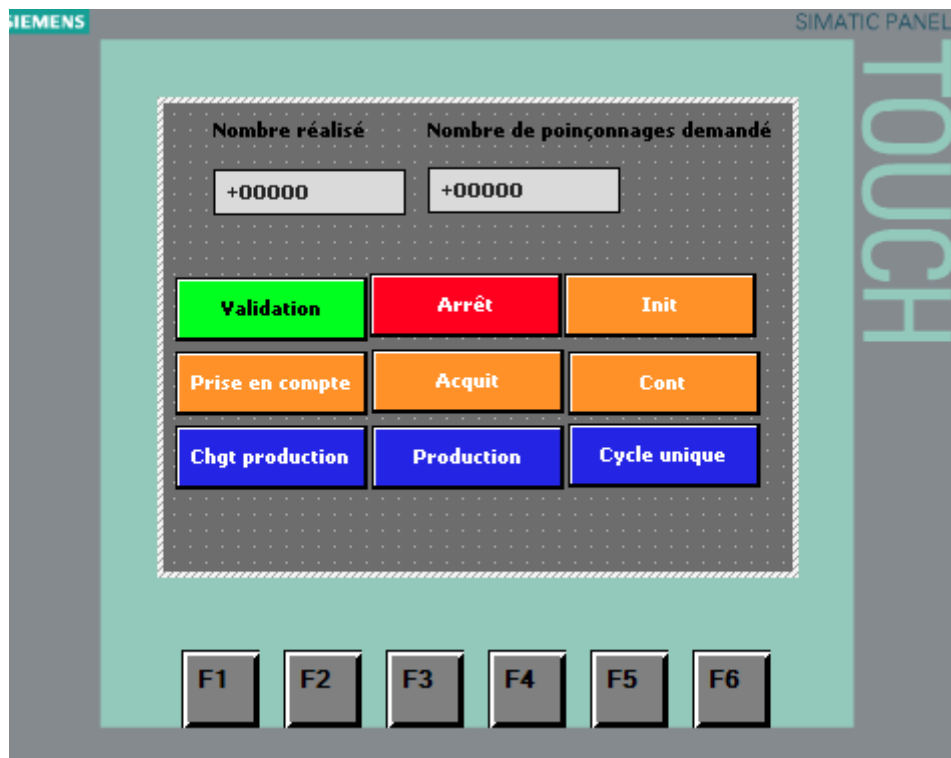
L'IHM est le maître et l'API est l'esclave.

L'interface graphique permet d'appuyer sur différents boutons pour déclencher les événements du GEMMA qui nécessitent la commande de l'opérateur.

Pour cela on crée d'abord les variables IHM, qui vont pointer vers les variables API

Variables IHM								
Nom	Table des variables	Type	Connexion	Nom API	Variable API	Adresse	Mode d'accès	Cyc
acquit_ihm	Table de variables standard	Bool	Liaison_IHM...	PLC_8	Acquit	%M10...	<accès absol...	...
arret_ihm	Table de variables standard	Bool	Liaison_IHM_2	PLC_8	arret	%M101.2	<accès absolu>	1 s
chgt_production_ihm	Table de variables standard	Bool	Liaison_IHM_2	PLC_8	chgt_prod_valid	%M101.1	<accès absolu>	1 s
cont_ihm	Table de variables standard	Bool	Liaison_IHM_2	PLC_8	Cont	%M100.2	<accès absolu>	1 s
cycle_unique_ihm	Table de variables standard	Bool	Liaison_IHM_2	PLC_8	cycle_unique_valid	%M100.7	<accès absolu>	1 s
index_ihm	Table de variables standard	Int	Liaison_IHM_2	PLC_8	index	%MW200	<accès absolu>	1 s
init_ihm	Table de variables standard	Bool	Liaison_IHM_2	PLC_8	Init_PO	%M100.4	<accès absolu>	1 s
nb_poinconnage_ihm	Table de variables standard	Int	Liaison_IHM_2	PLC_8	nb_poinconnages	%MW202	<accès absolu>	1 s
Numéro_vue_variable	Table de variables standard	UInt	<Variable intern...		<indéfini>			1 s
prise_en_compte_ihm	Table de variables standard	Bool	Liaison_IHM_2	PLC_8	Prise_compte	%M100.0	<accès absolu>	1 s
prod_verif_ihm	Table de variables standard	Bool	Liaison_IHM_2	PLC_8	prod_verif	%M101.5	<accès absolu>	1 s
production_ihm	Table de variables standard	Bool	Liaison_IHM_2	PLC_8	prod_valid	%M101.0	<accès absolu>	1 s
valid_ihm	Table de variables standard	Bool	Liaison_IHM_2	PLC_8	Valid	%M100.3	<accès absolu>	1 s
<ajouter>								

Dans un premier cas, il s'agit de mettre sur une même fenêtre tous les boutons et compteurs. Il y a 9 boutons comme décrits sur la fenêtre suivante :



Le problème avec cette IHM est qu'on propose toutes les options à l'utilisateur sur une seule fenêtre alors que des tâches ne sont pas disponibles car l'API n'est pas dans l'étape du GC qui valide cet ordre précis.

La règle est de faire évoluer l'interface de l'IHM à chaque fois que l'utilisateur déclenche un événement. Autrement dit, à chaque fois que l'utilisateur appuie sur un bouton ou modifie un état, son interface graphique doit changer en fonction de l'ordre donné de façon visuelle et fonctionnelle.

Nous allons donc concevoir une fenêtre spécifique à chaque procédure (A1, F1...). C'est l'automate qui demandera (en fonction de l'action dans laquelle il est) par un ordre précis quelle fenêtre afficher (celle définie par le programmeur comme correspondant à l'état du grafcet où l'API est).

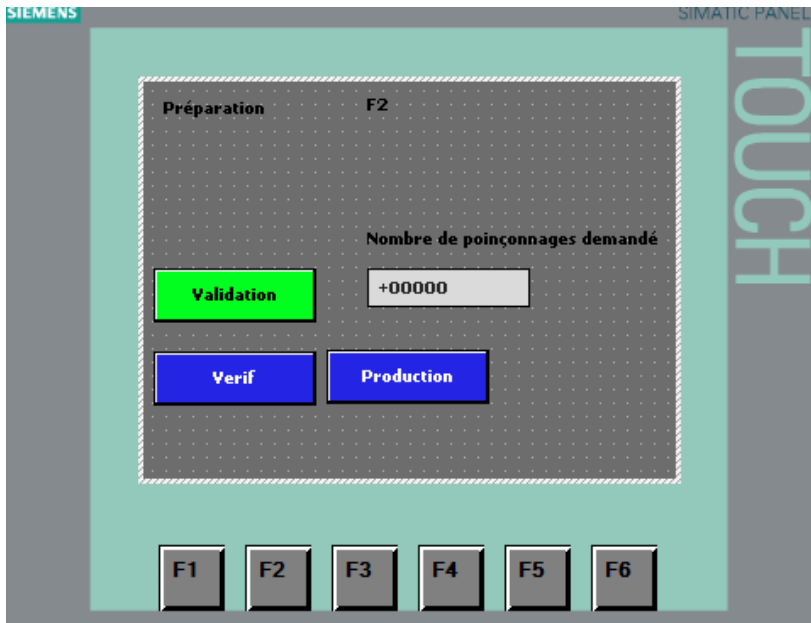
La fenêtre racine est la vue F1

Voici le tableau avec les numéros de chaque page du GEMMA en décimal et en hexadécimal.

TABLEAU DES NUMEROS DE PAGE

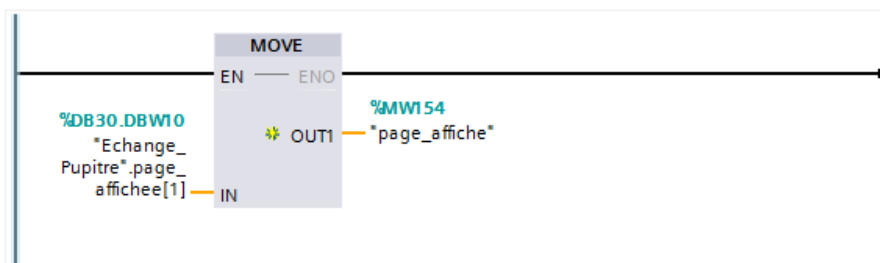
Page	No en décimal	No en hexadécimal
A1	110	006E
A3	130	82
A4	140	8C
A5		96
A6	160	A0
A7		AA
D1		D2
D2		DC
F1	310	136
F2	320	140
F3	330	14A
F4	340	154
F5	350	15E

Cela donne par exemple l'une des fenêtres suivantes, qui est associée à la vue F2



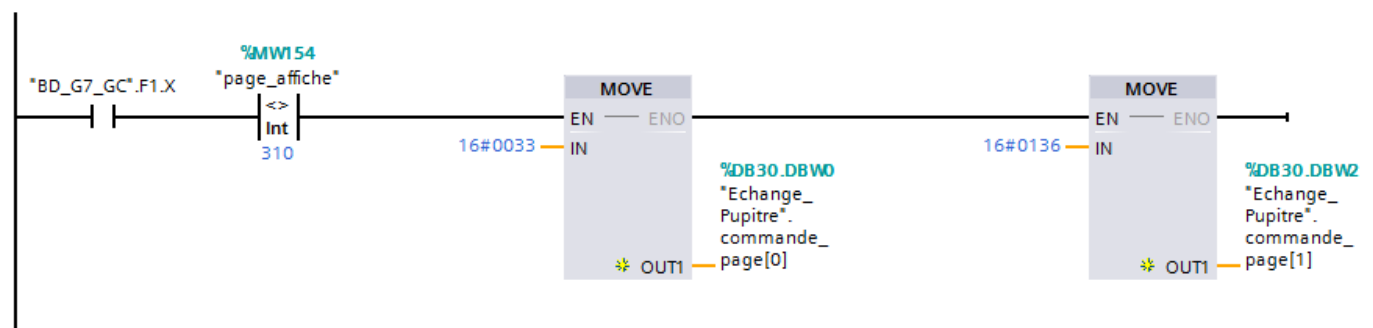
On crée ensuite un nouveau bloc de données qui paramètre les fonctionnalités attendues.

On crée ensuite pour chaque appel de fenêtre par l'API un réseau i tel que celui décrit dessous



Réseau 2 : Page F1

Commentaire



Exemple :

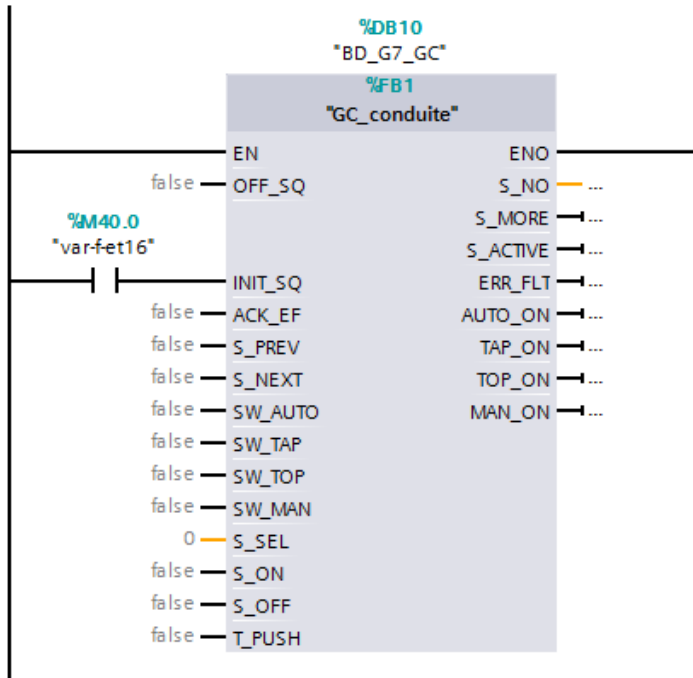
Lorsqu'on arrive dans la fonction F1, si la page affichée est différente de la page demandée, on applique la requête 33 (hexadécimal) dans le premier mot de la DB et on demande la page 510 (136 en hexadécimal) dans le second mot de la DB (selon la documentation constructeur).

Requête 33 : demande d'affichage sur l'IHM par l'API

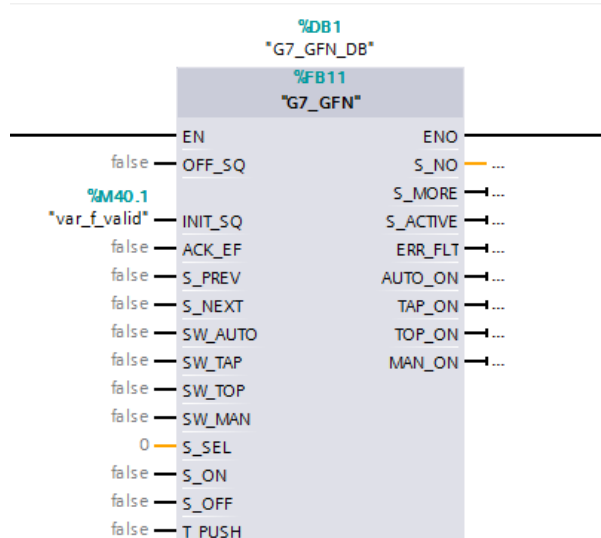
VIII) Tests et validation

.On a le bloc main qui lance les blocs fonctions suivant :

- GS_GRAS
- GC_conduite avec le paramétrage suivant



- Verrines
- Figeage
- G7_GFN avec le paramétrage suivant :



- Gestion_pages

On a deux blocs de données BD_G7_GC pour le graphique de conduite et G7_GFN_DB pour le graphique de Fonctionnement normal. On les utilisera pour appeler les étapes (par exemple « BD_G7_GC ».F1.X

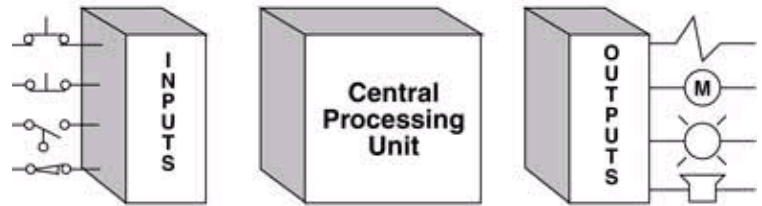
Conclusion

Nous avons pu grâce à l'application du GEMMA de la poinçonneuse de porte d'armoires faire les Grafsets puis les programmer en langage API. Nous avons pu nous familiariser avec les fonctionnalités d'un PLC, de son IHM grâce à leur logiciel de programmation TIA Portal V12.

Le but d'étudier et de réaliser des traitements de gestion de modes de marches est atteint puisque nous avons réalisé et programmé les différents cycles : en arrêt, en remise en route, en mise hors ou en service, en production, en essais et vérifications ou encore en défaillance. L'objectif de tester les différentes étapes en activant les transitions adéquates (forçage ou à partir de l'IHM) a également été couronné de succès puisque nous avons vérifié la réponse de l'automate aux différentes commandes (marche, arrêt, etc...).

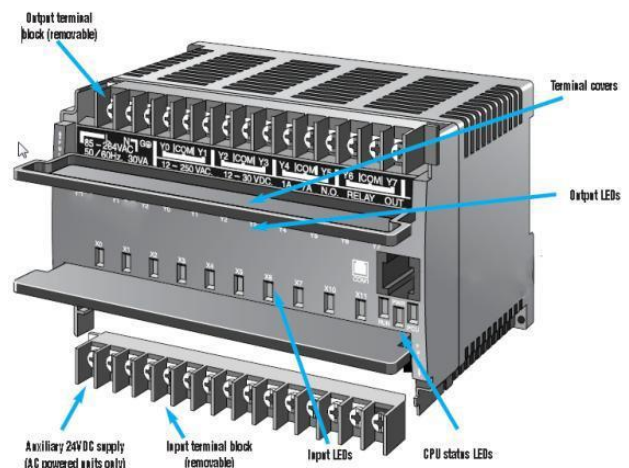
Ce TP nous a donc permis de comprendre le fonctionnement global d'un système et les composants qui gravitent autour du fonctionnement normal. Cette application a pour but de prévoir tous les dysfonctionnements et fonctionnements et de gérer cet ensemble de manière complète, construite ; concise et structurée.

PLC'S IN ENGLISH



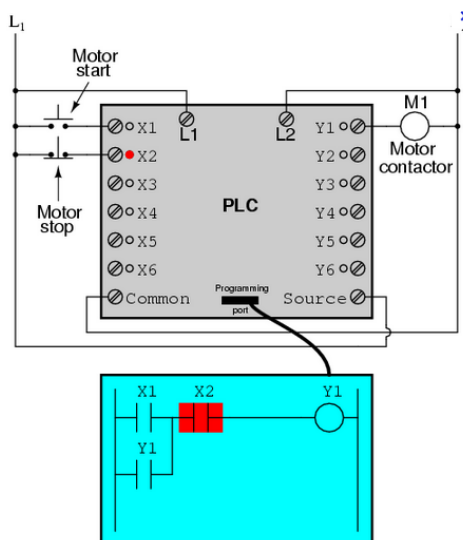
What's inside

The sensors, run/stop commands are connected to the input of PLC's on the built-in program ports of the central processing unit. The program uploaded in the PLC's calculates the consequences on the outputs (lights, motor, screens, HMI's, etc...).



Wiring

A PLC's is wired on inputs to buttons or sensors and on the output on the command of actuators and preactuators. The network can be used to connect the PLC's to others digital devices, like the master/slave protocol.



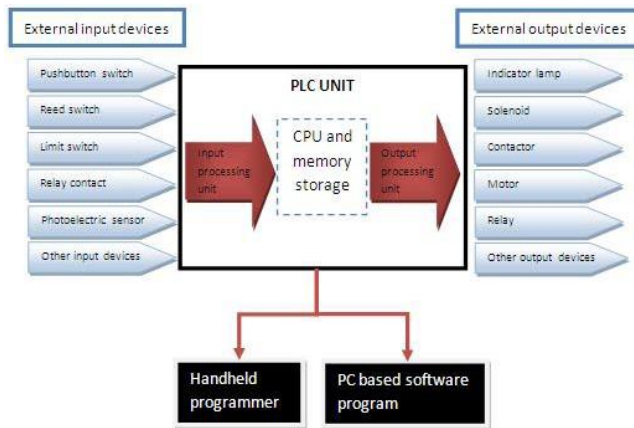
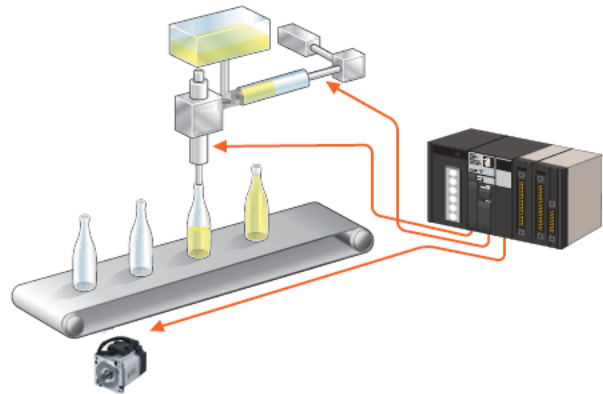
Use general description

The buttons are wired to the PLC in the inputs. The uploaded program in ladder, or in state transition control diagram commands the outputs. The motor contactor is wired to the output and the mechanical input (push the

button) has a mechanical output (start or stop the motor) via a digital command device.

Use of PLC's in process

The PLC is of common use in the industry. The programming technician writes the program uploaded in the PLC connected to sensors and actuators. He draws the events on the output (here an actuator cylinder or a floodgate to full the bottles or the motor of the conveyor belt) when received the inputs. Each element is connected to the network by wires.



Connections

A lot of devices can be connected to the PLC's and their signal can be treated by the program:

Inputs:

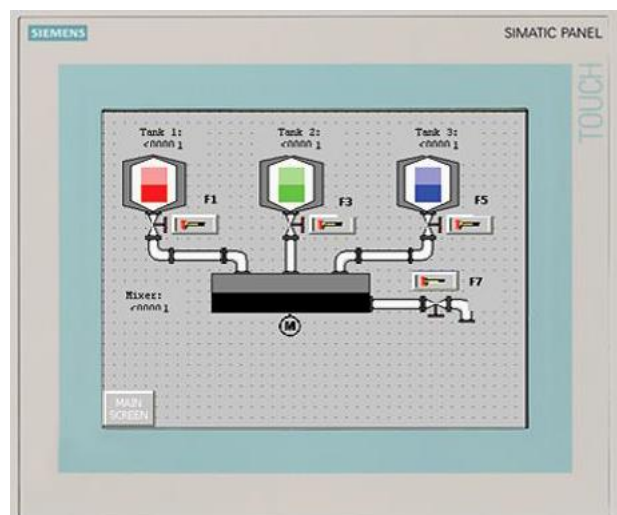
- Pushbutton switch (security, start and stop)
- Limit switch (security)
- Relay contact
- sensors

Outputs:

- Lamps, solenoids, contactors (to order a movement), motors, relays, etc.

Using HMI's

An HMI's is a digital interface which facilitate the use of PLC's. The tactile screen sends orders to the PLC's and receives state of sensors and actuators. It's directed by the PLC's. The user can use it to command and follow the system.





Société

WMI :

ETUDE DE FAISABILITE

Par Frédéric Gilet

2013

MON ENTREPRISE : WMI

L'utilisation de tablettes ou de terminaux dédiés à l'industrie permettrait de suivre le cycle de fabrication de produits, notamment dans l'aéronautique ou l'automobile. Le chef chargerait la commande par câble USB ou Wifi dans la tablette à partir de la commande de l'ordinateur relié au réseau informatique. Chaque ouvrier sur son poste lirait la tâche qu'il lui serait dédiée par l'informatique avec les précisions nécessaires. Il effectuerait son travail et le validerait sur la tablette ou brancherait la tablette, préalablement chargée avec les paramètres de production nécessaires, à sa machine pour effectuer sa tâche. Celle-ci effectuée, il transmettrait le produit avec sa tablette au poste suivant. A la fin, le chef réceptionnerait le produit fini et déchargerait les informations de la tablette vers le SI de l'entreprise, pour valider la livraison. Cela permettra un suivi irréprochable de la qualité et des lots. Elle permettrait un gain de productivité et de qualité substantiels où l'ouvrier serait dédié à sa tâche. Ce système est adapté à la production en ligne.

Description du produit

Produits et services principaux

Développement d'un progiciel de gestion d'un atelier dans une usine, pour le suivi des commandes à chaque poste et la traçabilité du produit.

Les besoins auxquels le produit répond

Usine voulant améliorer sa productivité, sa traçabilité, sa flexibilité, sa qualité.

Différence avec la concurrence

Ça n'existe pas ou peu en tant que tel dans les ateliers

Plus

La tablette qui remplace le papier pour valider les opérations.

L'écran de contrôle tactile.

Les clients

Son identité

Les industriels, plus tard les entreprises de service.

Localisation

France entière.

Clients potentiels

Les entreprises de l'industrie ayant des ateliers.

Marché

PME-PMI, grandes entreprises (certains ateliers ciblés)

Fournisseurs

Fabricants de tablettes, d'ordinateurs, de câbles, d'imprimantes, papier, logiciels de programmation (Visual Studio de Microsoft), bases de données.

Le potentiel

Prospection dans la France entière là où les usines ne sont pas équipées en progiciels.

Utilité

Traçabilité, productivité accrue (facilité de gestion des commandes), transmission facilitée du métier concerné.

Usage

1 chef par atelier qui paramètre les bases de données lors de l'initiation du projet et au jour le jour est responsable de la bonne utilisation du produit + un technicien qui utilise l'ordinateur quotidiennement pour gérer le produit + les ouvriers qui manipulent la tablette à chaque poste.

Performance potentielle

De l'idée au projet

Produit et services principaux

Un logiciel qui permet de rentrer les commandes dans l'ordinateur à écran tactile, puis de la traiter directement sur les chaînes de l'usine. La commande est décomposée en gammes, que l'on imprime ou que l'on place sur une tablette numérique pour être suivie à chaque étape de la fabrication (en mode série). Chaque moment du processus de la construction d'un produit est validé en temps réel, puis lorsque l'article est fini, cet état est fourni à l'ordinateur qui donne le bon de fabrication avec les spécificités du modèle produit, le bon de livraison et le bon de transport. En parallèle, c'est une gamme qui est répétée et on peut connecter des automates.

Besoins

Produits complémentaires

Sa différence avec la concurrence

Idée nouvelle ou peu répandue.

Clients potentiels

Leur identification

Les investisseurs, les acheteurs, les directeurs d'usines

La cible

PME-PMI dans un premier temps, puis grandes entreprises (ateliers)

Où

France entière

Phase de test

Dans les bureaux de WMI puis sur les 1ères entreprises.

Le marché

A défricher et à occuper

Motivation

Objectifs

Savoir faire

Infologic + connaissance atelier + connaissance d'un superviseur d'atelier.

Contacts

Les acheteurs, les responsables d'usines, les chefs d'atelier, les chefs d'entreprises.

L'élaboration du projet

Etude commerciale

Hypothèse de CA

Produit, prix, distribution, communication

Tendances du marché

Produit vendu

Clients

Type

Capacités d'investissement

Besoins

Où sont-ils ?

Leur comportement

Attente du client

- Gain de temps
- Gain d'argent
- Meilleure traçabilité
- Rapidité de réaction
- Meilleure production
- Suivi des modèles fabriqués

Comment vendre

- Par un commercial
- Par un réseau
- Par internet

Les concurrents

Qui ?

SSII

Combien ?

Beaucoup

Où ?

Toute la France

Leur proposition ?

- Un ERP
- Du sur-mesure

Leur prix ?

Moment propice à l'achat ?

Lors des réunions d'investissement.

Type de demande ?

Ponctuel

Achats ?

Programmés

Où se placer ?

Prix ?

Les caractéristiques de la concurrence ?

Peu développée sur ce cœur de métier.

Action commerciale ?

- Action de prospection
- Site internet
- Relance

ETUDE DE PRIX DE LA CONCURRENCE

forfait 3 550 € hors taxes, comprenant :

- la licence du logiciel,
- le manuel d'utilisation, illustré d'images tirées d'une base de données (fournie) de présentation : 230 pages en couleur,
- la mise à disposition de notre bibliothèque de 170 documents Crystal Reports[®], pour un déploiement rapide du logiciel et sa mise en exploitation effective sous trois à quatre semaines,
- trois demi-journées passées dans votre entreprise à reprendre vos données actuelles, ou personnaliser à votre entreprise la bibliothèque d'états, ou bien vous former au paramétrage et à l'utilisation,
- nos frais de déplacements,
- le service d'assistance et des mises à jour pendant trois mois ;

attention !

cette intervention unique peut dans des cas rares se révéler insuffisante pour atteindre l'opérationnel ; en cas de nécessité, nous nous réservons la possibilité de présenter un devis complémentaire de prestations, notamment pour la réalisation d'états de sortie spécifiques et des journées supplémentaires d'accompagnement ;

900 € hors taxes l'annuité, à compter du quatrième mois suivant cette première intervention, contrat forfaitaire couvrant pour 12 mois l'abonnement aux mises à jour et l'assistance téléphonique à l'exploitation.

Accès au [code à barres](#), pour la saisie des temps passés par les compagnons, sur les OF :

Variante 1 - Douchette de lecture

350 € hors taxes la licence de la **police de code à barres (code 3/9)**, pour chaque poste de travail sortant des documents où figurent les codes barres,

- ➔ 1 000 € hors taxes le logiciel de lecture et la fourniture d'une douchette, par poste de travail à équiper.

■ Variante 2 - Terminal portable de saisie ("stylo")

- ➔ 350 € hors taxes la licence de la **police de code à barres (code 3/9)**, pour chaque poste de travail qui sortira des documents où figurent les codes barres.

- ➔ 1 200 € hors taxes la licence du logiciel de récupération des temps mémorisés dans le stylo, par poste de travail à rendre capable de récupérer ces temps.

- ➔ 1 250 € hors taxes le terminal portable à lecture

Laser, distance de lecture jusqu'à 70 cm, nombre de lectures 36 scans/s, taille mémoire 512 KO, autonomie 18 heures, poids 184 g, afficheur 4 lignes de 16 caractères, clavier alphanumérique 25 touches, connexion RS 232 via le chargeur/transmetteur livré avec socle complet et batteries NiCd (batteries NiMh 600mA disponibles sur demande).

- ➔ 700 € hors taxes la journée, hors frais, pour l'installation, le paramétrage et la formation.

+ Formation sur site ou prestations complémentaires

- ➔ 860 € hors taxes la journée sur site, outre les frais de déplacement à refacturer sur justificatifs.



Dossier de création d'entreprise

10 rue du Pré-Pigeon, 49100 ANGERS

☎ 06 82 03 16 50

✉ : fredgilet49@orange.fr

I PRESENTATION GENERALE DU PROJET	87
II HISTORIQUE ET MOTIVATIONS :	87
III PRESENTATION DU OU DES CREATEURS	87
IV PRESENTATION DE L'ACTIVITE	89
V L'ENVIRONNEMENT GENERAL DE LA PROFESSION (MACRO-ENVIRONNEMENT)	90
VI LE MARCHÉ VISE (MICRO-ENVIRONNEMENT)	90
6.1/ La clientèle :	90
6.2/ L'implantation ou la zone d'intervention :	90
6.3/ Les prescripteurs :	91
6.4/ - La concurrence :	91
6.5/ - Les collègues:	91
VII LES MOYENS MIS EN OEUVRE	92
7.1/ Le positionnement marketing	92
7.2/ Les actions commerciales	92
7.2.1/ Les moyens de communication envisagés	92
7.2.2/ Le budget communication	93
7.2.3/ Planification des actions commerciales de la 1ère année	93
7.3/ Les moyens de production	93
7.3.1/ Le local ou l'atelier	93
7.3.2/ Le matériel	93
7.3.3/ Le personnel	93
7.3.4/ La forme juridique	93
7.3.5/ Les options fiscales	93
7.3.6/ Le statut social	93
VIII LE FINANCEMENT	93
8.1/ Les investissements de départ	93
8.2/ Le BFR	93
8.3/ Les ressources de financement de départ	93
8.4/ Le plan de financement de départ	94
IX CHIFFRE D'AFFAIRES / RENTABILITE	94
9.1/ Le chiffres d'affaires prévisionnel	94
9.2/ Le compte de résultats	94
9.3/ L'autofinancement et le disponible net	94
X CONCLUSION ET PERSPECTIVES DE DEVELOPPEMENT	94

I PRESENTATION GENERALE DU PROJET

Le progiciel de gestion de production que je propose est composé de différents modules pour gérer un atelier. Il va de la prise de commandes jusqu'à la livraison. C'est un service aux entreprises industrielles novateur puisqu'il propose de gérer le flux de marchandises grâce à l'outil informatique à toutes les étapes de la fabrication. Son point fort est l'utilisation des dernières technologies numériques dans les ateliers. Destiné à la France, la création de mon entreprise aurait lieu en 2014

II HISTORIQUE ET MOTIVATIONS :

Ayant une formation adaptée et par la suite une expérience significative dans les progiciels de gestion et dans l'informatique, je suis porté par une idée novatrice que j'ai à cœur de mettre en œuvre. Il existe déjà des logiciels de gestion d'atelier mais je pense apporter une touche innovatrice et personnelle dans la manière de gérer un atelier. Etant sans emploi et voyant un marché potentiel, je saisis l'opportunité d'avoir du temps pour aborder le sujet. Mon but est d'en vivre et de développer mon activité dans un secteur porteur.

III PRESENTATION DU OU DES CREATEURS

Le créateur et ses atouts :

Frédéric GILET (42 ans)

10, rue du Pré-Pigeon

49100 ANGERS

☎ 06 82 03 16 50

✉ fredgilet49@orange.fr

Informatique industrielle

Automatisme



✚ FORMATION

2015 Obtention du titre **T.S.A.I.I.** (*Technicien Supérieur en Automatismes et en Informatique Industrielle*).
GRETA Loire-Atlantique.

1999 **Ingénieur diplômé de l'E.N.S.A.M. (Paris Tech Arts et Métiers).**

1998 **Master of Sciences délivré par l'Université de Lancaster (R.U.).**

Etude de la **mécatronique** (*interface mécanique, électronique, programmation*).

1993 Obtention du **Baccalauréat C** (mention AB).

Titulaire du **B.A.F.A.** (*Brevet d'Aptitude aux Fonctions d'Animateur*).

LANGUES ETRANGERES

Anglais courant (une année d'études au Royaume-Uni).

Espagnol : notions

✚ CONNAISSANCES

INFORMATIQUE

Système d'exploitation : Unix, Windows

Bases de données : Unidata, Ingres, SQL

Langages : Visual Basic, C, C++, Python

Bureautique : Pack Office, Movie Maker

Réseaux : TCP/IP, Modbus

Gestion de projet : Mindview 4

AUTOMATISME

Grafset, Ladder, Siemens TIA Portal V12, Schneider Unity Pro M, Winrelais, PCVue

✚ EXPÉRIENCE PROFESSIONNELLE

Mai-juin 2015

Pierre Guérin SAS (79) – stage technicien.

Mise en service d'une unité de fermentation industrielle.

Février 2015

Fast Intégration (49) – stage technicien.

Développement d'une supervision IHM.

2013-2014

Bénévole à l'atelier journal de L'**ESAT du Bord de Loire.**

03/2008 à 01/ 2011

ESAT du Bord de Loire (49).

Travail en atelier, saisie informatique.

01/2001 à 09/ 2004

SNCF (région parisienne) – ingénieur d'étude et de développement.

Participation à l'étude et au développement d'un logiciel d'affectation des trains à quai.

08/1999 à 12/ 2000

Infologic (44) – ingénieur réalisateur en informatique.

Participation aux développements des modules production et expéditions d'un logiciel de gestion dans l'industrie agroalimentaire, connexion d'automates, installations.

1998 (6 mois)

DR Climbing Wall (Royaume-Uni) - stage ingénieur.

Réalisation du pilotage central de LEDS multicolores pour un mur d'escalade.

1993, 1994, 1995

Animateur en centre de vacances pour enfants handicapés avec l'APF.

✚ CENTRES D'INTÉRÊT

Écriture, dessin, peinture, photographie, musique (clarinette), informatique, jeux de stratégie.

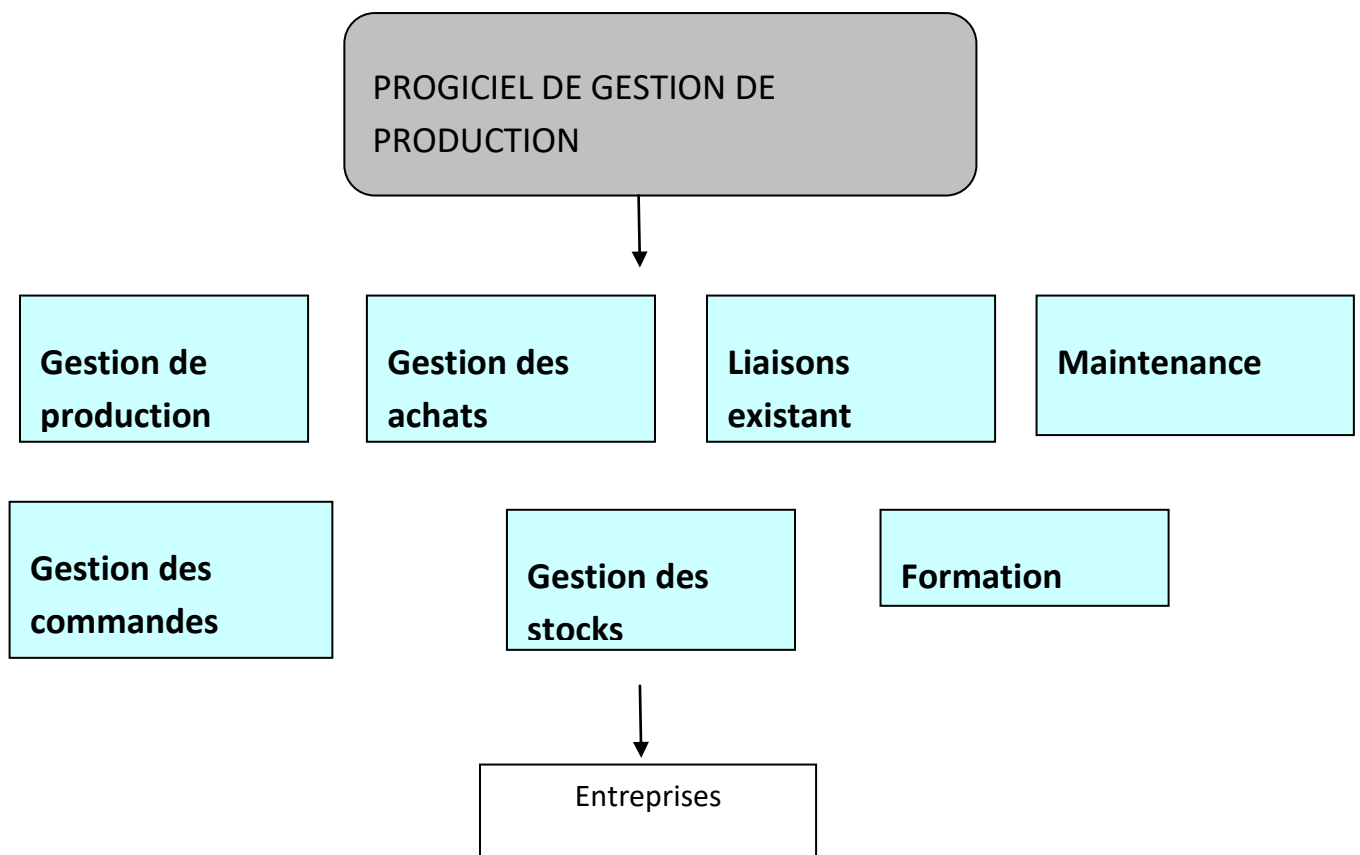
Les associés éventuels ou son entourage :

Je ne compte pas avoir d'associés dans un premier temps.

L'entourage familial est composé de mes parents et de ma sœur et de sa famille.

IV PRESENTATION DE L'ACTIVITE

La présentation visuelle de l'activité par l'arbre à produit :



Le détail des prestations :

Les prestations vendues sont la vente de logiciels, leur installation, la formation du personnel et la maintenance. Le logiciel est vendu en fonction du nombre de modules et du nombre de postes. La formation est vendue en nombre de journée. La maintenance est vendue au forfait annuel. Le logiciel est installé par les soins de mon équipe sur place. La formation est réalisée par les soins de mon équipe. La maintenance se fera par téléphone. Le matériel est préconisé (ordinateurs HP et tablettes Samsung)

V L'ENVIRONNEMENT GENERAL DE LA PROFESSION (MACRO-ENVIRONNEMENT)

Environnement juridique et réglementaire :

Les contraintes réglementaires sont le respect de la loi informatique et libertés. Cela exige par ailleurs de garantir le secret professionnel et la sécurité des données stockées par la base de donnée de mon logiciel.

Les principaux chiffres de la profession :

40000 entreprises d'informatique en France pour un C.A. de 41 milliards d'euros et 485000 emplois en France. Croissance du marché de 1.5% en 2010. L'industrie représente 35% du C.A. des entreprises. Les services représentent 47% du C.A. pour 40% des emplois. Les logiciels représentent 25% du C.A.(soit 12 milliards d'euros) pour 7% des emplois. Evolution stable du nombre de créations d'entreprises à 450 en 2010. Défaillances à 1000 entreprises (stable) mais cessations en hausse. C.A moyen de 150000 euros à 3 millions d'euros pour les petites structures.

(sources APCE)

VI LE MARCHE VISE (MICRO-ENVIRONNEMENT)

6.1/ La clientèle :

La clientèle est composée de PME-PMI dans un premier temps désireuses de faire évoluer leur outil de production en termes de productivité, d'amélioration du processus de production et de distribution, de transmission du savoir-faire et de connaissance de l'avancement de la fabrication.

L'acheteur est l'entreprise, le décideur d'achat est au niveau du système d'information souvent le PDG, conseillé par le département des achats et aiguillé par le responsable de la production.

Cet achat est pratiquement définitif sauf changements majeurs dans l'organisation du client. Il se fait au départ (installation du logiciel) et en continu (améliorations de versions, formation, SAV). Le prix doit être justifié par la qualité du service.

6.2/ L'implantation ou la zone d'intervention :

Les entreprises clientes ont un siège social situé en France et des sites de production implantés un peu partout dans les zones industrielles. Des déplacements sur site sont à prévoir pour l'installation et la formation.

6.3/ Les prescripteurs :

Les prescripteurs sont des entreprises déjà clientes qui vont être les ambassadeurs de mon logiciel. Une visite d'usine à un client potentiel garantit la position de partenaire privilégié.

6.4/ - La concurrence :

- *Dans un premier temps, le créateur doit apporter au lecteur une vision globale mais synthétique de la concurrence et de son importance sur le marché envisagé.*
- *Dans un second temps, il s'agit de lister les principaux concurrents directs (même activité que le porteur de projet) et indirects (répondant au même besoin de base). Un tableau synthétique peut être utilisé pour rappeler les positionnements adoptés par ces concurrents et des critères importants pour l'activité.*
- *Quelles conclusions tirez vous de cette analyse pour votre projet ?*

	Concurrent 1	Concurrent 2	Concurrent 2	Concurrent 2
CA HT				
Effectif				
Clientèle visée				
Prestations proposées				
Tarifs				
Communication				
Notoriété				
Forme Juridique				
Points forts				
Points faibles				

6.5/ - Les collègues:

- *Dans un premier temps, le créateur doit identifier 2 ou 3 collègues/confrères susceptibles de lui donner des conseils.
(même activité, même cible clientèle, zone géographique différente)*
- *Dans un second temps, le créateur doit construire un guide d'entretien qu'il pourra utiliser lors de la rencontre avec ses collègues.*
- *Quelles conclusions tirez vous de cette analyse pour votre projet ?*

VII LES MOYENS MIS EN OEUVRE

7.1/ Le positionnement marketing

Cible	Produits/Services	Prix	Circuit de distribution (emplacement)	Communication
PME-PMI	Installation	1000 ^E	Déplacements chez le client	Démarche téléphonique et publicité sur internet
	Formation	200E par jour	Chez les clients	Négociation avec le client
	SAV	Forfait annuel	Par le réseau	Négociation avec le client

7.2/ Les actions commerciales

7.2.1/ Les moyens de communication envisagés

Prospecter : contacts téléphoniques, relances e-mails, salons, site internet.

Cultiver l'image de marque : salons, articles de journal, site internet.

Fidéliser : remises ou modules à prix réduits.

7.2.2/ Le budget communication

7.2.3/ Planification des actions commerciales de la 1ère année

7.3/ Les moyens de production

7.3.1/Le local ou l'atelier

- ses caractéristiques (surface ouverture, réserve, vitrine...)

Chez moi.

7.3.2/Le matériel

Téléphone, ordinateur tactile, tablette numérique, imprimante, véhicule.

7.3.3/Le personnel

Seul dans un premier temps.

7.3.4/La forme juridique

Justifier votre choix :

- EI / EURL / SARL ...

- la répartition des parts et le nom du gérant si société

7.3.5/Les options fiscales

- TVA, impôts sur les bénéfices, impôt sur le revenu...

7.3.6/Le statut social

- le statut social du chef d'entreprise ou des associés

VIII LE FINANCEMENT

8.1/ Les investissements de départ

8.2/ Le BFR

8.3/ Les ressources de financement de départ

- apports personnels

- aides

- emprunt(s)

8.4/ Le plan de financement de départ

IX CHIFFRE D'AFFAIRES / RENTABILITE

9.1/ Le chiffres d'affaires prévisionnel

Justifiez-le et décomposez-le en unités d'œuvre

9.2/ Le compte de résultats

Justifiez votre calcul

9.3/ L'autofinancement et le disponible net

X CONCLUSION ET PERSPECTIVES DE DEVELOPPEMENT



NOTE BOOK

*Most advanced quality
Gives best writing features*

Frédéric Gilet

WMI

Etude technique

APICA

50

No. _____

Date. _____

W.M.I.

ETUDE DU LOGICIEL

22 janvier 2018

MENU

Fichier	Saisie des Données	Gestion	Paramètres
<ul style="list-style-type: none"> Quitter Sauvegarder Druid MMI Admin Utilisateur 	<ul style="list-style-type: none"> Entreprise Site Alélier Poste Article Utilisateur Article → Description modèle → Type Article 	<ul style="list-style-type: none"> Stokes Approvisionnement → saisie des ventes → saisie des dépenses → type → Production 	<ul style="list-style-type: none"> Paramétrage Base de données sans Tablette sans ordi Bureau
	<ul style="list-style-type: none"> Gamme → Description étape → Description gamme 	<ul style="list-style-type: none"> Commande → saisie de commande → correction EDI → fichier 	<ul style="list-style-type: none"> Commande Expéditions Statistiques
	<ul style="list-style-type: none"> Modèle type → description Modèle type → Approvisionnement Modèle type 	<ul style="list-style-type: none"> Article produit → saisie d'un production → saisie d'un gamme → édition → exportable - stable 	

SAISIE CLIENT

No client :

Nom client :

Adresse

Ville

Code postal

Pays

No article	Nom article	Type marque	Description marque	Prix

Type caractéristique	No Caractéristique

SAISIE ARTICLE

No article

Non article

1. .

Dimensions

→ hauteur
→ largeur
→ Poids
D Poids variable

Description

Code barre

No fournisseur	No fournisseur	Prix

Type de modèle	Non modèle	site en poste	Prix
----------------	------------	---------------------	------

No client	Non client	Prix.

No produit	No article fin	No produit pour le faire	No game pour de faire

SAISIE COMMANDES

No commande

Type :

No client

1 : whisky

No client

2 : Petite

Adresse

3 : Rds venable

4 : Rds jre

5 : unite

N° bon article	No Article	No modele	qte modele	prix	No etel com

No etel commande

0 : Non commence

1 : en cours

2 : termine

SAISIE GAMME A FAIRE

No Gamme

No Gamme

De [date] à [date]

De [heure] à [heure]

Gamme	Tâches	Temps Total	Heures	
			deb	Fin

SAISIE DESCRIPTIF GAMME

No / Gamme Nom	No Tâche (code chrono)	No tâche	Description Tâche	Type Tâche

No gamme	Produits début	No produit fin	No produit

SAISIE ~~RSE~~ ATELIER

No atelier

~~No poste~~ No eleve
Site

Type atelier

No gamme

No ordre chronologique	No Poste	No Tâche	No Tâche

SAISIE ENTREPRISE

No entreprise

No entreprise

No SIRET

Adresse siège social

Adresses → Adresse
→ Facturation

Description

Sites

SAISIE SITES

No enhyuse

No site	No site	Adresse	Type site	Type fabrication

SAISIE TYPE FABRICATION

No fabrication	No alche	No alche	Description fab	No note

SAISIE POSTE

No poste	No poste	Non alche	Description poste

No poste	No poste	No fab	No fab

SAISIE UTILISATEUR

Nb operateur

Nb operateur

Badge

Type → operateur
→ chef de file
→ administrateur
→ WRTI

SAISIE TÂCHE

No tâche	No tâche	No/No premier fin tâche	No/No premier debut tâche

SAISIE TABLETTE

No tablette	No lebleh	No sub	No entreprise
No bureau	No atelier	Adress IP	

SAISIE ORDINATEUR BUREAU

No ordi	No ord	No sub	No entreprise
No bureau	No atelier	Adress IP	
Non	Non		

SALIS MODEL

No model	No model	No site	No out	No Role	No game	Type job

Type job
① product fin
② product online dan
③ product base.

ASSOCIATION MODEL - CARRE

No model	No game	Reg game

ASSOCIATION PRODUCT - CARRE

VISUALISATION DIVERSES

CARNET DE COMMANDES

De (Date) à (Date)

No / No commandes No terminées	Modelé	Temps

VISUALISATION DES GAMES

De (Date) à (Date)

De (Heure) à (Heure)

No / No game	Tâche	Temps total

RECAPITULATIF D'UNE JOURNÉE

No / No model	quants	Temps
total		

GÉNÉRALITÉS

Chaîne : série
 parallèle
 JST

Nb. model = \neq ,
 Nb produit, Nb gen, Nb model \neq

Cyberseu : fait suite les commandes et pour
 l'heure départ et l'heure estimer gamme
 → possible Urgence (haut de pile)
 → visu des commandes dans cybase
 (état d'avancement)

→ Prendre plus et regner Teblette

→ On assure à un modèle 1 gamme.

→ succession des tâches

Nb produit	Nb produit précédé
Nb produit	Nb produit succédé

→ Editions PDF → page
 → base Teblette

No. _____

Date. _____

VISUALISATION CARTRE

No veh	No game	Taibes	temps game	heure debut	Type d'agence
		1 <input type="checkbox"/>			
		2 <input type="checkbox"/>			
		3 <input type="checkbox"/>			

Type d'agence : 0 : à faire sur de suite
1 : temps perdu
2 : majoré d'erreur.

Heure perdue de dispar

Temps : temps révisé
temps estimé



WMI Company

Work Management and Industry

USERS' MANUAL

February 2018

INTRODUCTION

The WMI Application is a software from the WMI Company. It's an ERP which manage production, customers and suppliers. There will be a supervisor for production and shipping.

The software contains an administrator solution (only in English) for the parameters (users, files, machines) and the database administration. The users are parametred with there pin-code wich is necessary to use the ERP.

When WMI is launched, the menu window appears. The user click on the menu File and user and then appears the connexion window: the user enters his username and his password parametred by the administrator and then can accede to the windows allowed by the administrator and necessary for the work of the user.

The windows will be explained in the rest of the document. Some indications will be told in the part "Company" for a good understanding and will not be repeated if not necessary.

Before starting, know that the software is available in French and in English, the langage is selected for each user by the administrator.

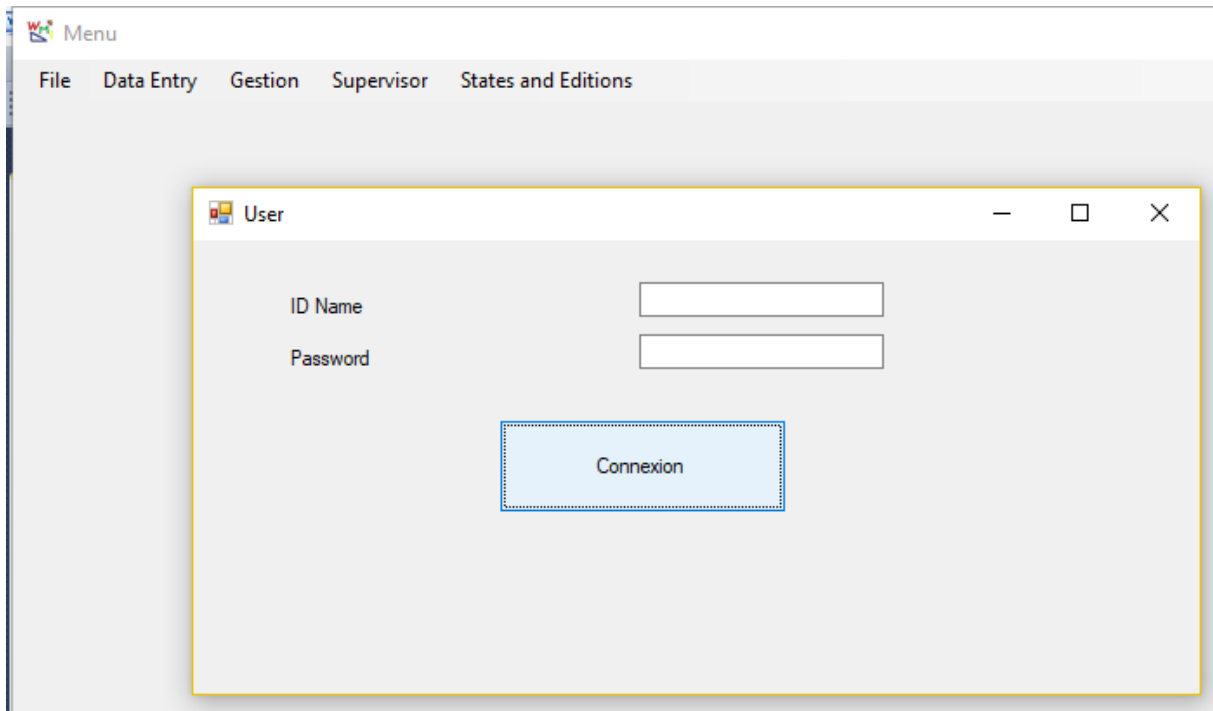
The database is writed in French (because this is the origine of the company WMI) but a traduction is available in the database descripttion for administrators.

Usually, the names have 20 characters, the address 30, the shortnames 10, the country field 20, the description 60, the contact 30, the legal notice 80, the ZIP code is numeric.

MENU

- File
 - o User
 - o Quit
- Data Entry
 - o Company
 - o Sites
 - o Workshop
 - o Workplace
 - o Workstation Type
 - o Item
 - o Supplier
 - o Customer
 - o Model
 - o Range of Manufacturing
 - Task Entry
 - Description Range Entry
 - o Availability
- Gestion
 - o Stocks
 - o Supplies
 - Supply Order Entry
 - Entering entries
 - o Order
 - Customer Order Entry
 - Entering a delivery
 - o Production Management
 - Entering a production
 - o EDI Connexions
- Supervisor
 - o Supply Supervisor
 - o Production Supervisor
 - o Shipping Supervisor
- States and editions
 - o Listings
 - o Stocks
 - o Supplies
 - o Production Management
 - Range of Manufacturing
 - Good of Manufacturing
 - o Orders
 - o Shipping Bill
 - Shipping Bill Edition

- Statistics
 - CEO Statistics
 - Trading Statistics
 - Order Book
 - Star Item
 - Manufacturing Statistics
 - Working Workload
 - Day Overview



DESCRIPTION OF THE WINDOWS ENTRIES

Window “Company”

In this section we will explain the bases of the software.

There are as many companies as you want defined by a number unique.

When you register a company for the first time, you are not forced to enter a number, the software will define itself a default number. If you want a precise number, you enter it.

If the company number exists, all the informations will appear on the screen.

If the company number doesn't exist, you can fulfill the fields which seems relatively easy.

You accede to the following field with the mouse of the “tab” key. In a table, you accede to the following row with the “enter” key. You validate the buttons with the mouse or the “Enter” key.

You can cut with the “Ctrl+X” key, you can copy with the “Ctrl+C” key, you can paste with the “Ctrl+V” key like any Microsoft Application.

You should define a shortname (10 characters maximum) which is unique and will be used in the rest of the software. When you want to search the company by this shortname, you can use the field “Quick Search”. Each time you enter a character, the software watch if it exists in the database.

The company may have several sites. In the table, you can define the numbers of this sites. When the line is finished, if the site is found out, his name will appears. Each time you consult the company, her sites numbers and sites' name will be informed. You can add a site by entering his number.

The zip code is numeric. You are not forced to give informations for all the fields (which are characters and numeric).

The software manage the pictures for the company, the site, the workplace, the workstation, the items. The administrator will create the good directories. For each picture, the name must be : 'Shortname.jpg'. If the picture doesn't exists, the software will indicate the default picture.

When the capture is finished, you register by the “Save Button”. You can delete the current recording by the “delete button” (a message will appears to ensure you agree)(and it will not delete the rest of the database, be sure). You can quit without saving with the “Quit” Button (the system will ensure that you don't want to save).

When you have deleted or recorded, the fields are empty and you can start a new entry.

When you quit the “Company Window”, you come back to the menu.

You can change the user or quit the application.

You can use this window for entering, modifying information or simply consult the informations (in this case you don't save).

If you are not registreted by the administrator as allowed to the entries menu, you can't accede to all his windows, in particular the "company" window.

Now let's see the window :

The screenshot shows a software window titled "Company Entry". It contains several input fields for company data:

- Company Number (with a blue border)
- Quick Search
- Company Name
- Short Name
- SIRET Number
- Address 1
- E-Mail
- Adresse 2
- Phone Number
- ZIP Code
- Fax Number
- City
- Country
- Web Site
- Description
- Mention Légale

At the bottom left, there is a table with two columns: "Site Number" and "Site Name". The first row contains an asterisk (*) in the "Site Number" column. Below the table is a large grey rectangular area.

At the bottom right, there are three buttons: "Save", "Delete", and "Quit".

Window "Sites"

Site Entry

Short Name

Quick Search

Site Number

Site Name

Address 1

Address 2

ZIP Code

City

Country

Description

E-Mail

Phone Number

Fax Number

Web Site

Contacts

Facturation

Address 1

Address 2

ZIP Code

City

Country

E-Mail

Phone Number

Fax Number

Web Site

Contacts

Livraison

Address 1

Address 2

ZIP Code

City

Country

E-Mail

Phone Number

Fax Number

Web Site

Contacts

Availability Number

Save Delete Quit

	Workshop Number	Workshop Name
*		

Window "WorkShop"

Worshop Entry

Workshop Number	<input type="text"/>	Quick Search	<input type="text"/>
Workshop Name	<input type="text"/>	Short Name	<input type="text"/>
Site Number	<input type="text"/>		
Site Name	<input type="text"/>		
Description	<input type="text"/>		
Contacts	<input type="text"/>		
Availability Number	<input type="text"/>		
Availability name	<input type="text"/>		

Workshop Type

Parallel
 Line
 Production Area

	Workplace Number	Workplace Name
*		

Window “Workplace”

Workplace Entry

Workplace Number

Workplace Name

Workshop Name

Site Name

Description

Quick Search

Short Name

Engine 1

Engine 2

IP Address

	Task Number	Task Name	Duration
*			

Workstation type

Receipts Availability Number

Manufacturing Availability name

Shipping Station

Management

Supervisor

Window “Type Workstation”

Saisie Type Poste

Workstation type

Type Poste Name

Quick Search

Short Name

	Workplace Number	Workplace Name	
»»			

Save Delete Quit

Window "Item"

Article Entry
— □ ×

Article Number

Article Name

Description

Packaging type

Bar Code

VAT

Type d'article

Lower Rank Item Number

Lower Rank Item Name

Lower Rank Quantity

Quantity Type

Dimensions

Lenght

Width

Height

Weight

Fixed Price

Weight type

Variable Weight

Fixed weight

Quick Search

Short Name

Article Type

Purchased Product

Semi Finished Product

Finished Product

	Supplier Number	Supplier Name	Model Number	Model Name	Price	Price Type	Delivery Time
*							

	Model Number	Model Name	Price	Price Type
*				

	Customer Number	Customer Name	Model Number	Model Name	Price	Price Type	Delivery Time
*							

Supplier Entry

Supplier Entry

Supplier Number	<input type="text"/>	Facturation	Quick Search	<input type="text"/>
Supplier Name	<input type="text"/>		Short Name	<input type="text"/>
Address 1	<input type="text"/>	Address 1		<input type="text"/>
Address 2	<input type="text"/>	Address 2		<input type="text"/>
ZIP Code	<input type="text"/>	ZIP Code		<input type="text"/>
City	<input type="text"/>	City		<input type="text"/>
Country	<input type="text"/>	Country		<input type="text"/>
E-Mail	<input type="text"/>	E-Mail		<input type="text"/>
Phone Number	<input type="text"/>	Phone Number		<input type="text"/>
Fax Number	<input type="text"/>	Web Site		<input type="text"/>
Web Site	<input type="text"/>	Site Web		<input type="text"/>
Contacts	<input type="text"/>	Contacts		<input type="text"/>
Description	<input type="text"/>			
Delivery Deadline	<input type="text"/>			

	Article Number	Article Name	Model Number	Model Name	Price	Price Type
*						

Window "Customer Entry"

Customer Entry

Quick Search

Short Name

Customer Number

Customer Name

Address 1

Address 2

ZIP Code

City

Country

E-Mail

Phone Number

Fax Number

Web Site

Delivery

Address 1

Address 2

ZIP Code

City

Country

E-Mail

Phone Number

Web Site

Site Web

Contacts

Contacts

Billing

Address 1

Address 2

ZIP Code

City

Country

E-Mail

Phone Number

Fax Number

Web Site

Contacts


Description

Delivery Time

Save Delete Quit

	Article Number	Article Name	Model Number	Model Name	Price	Price Type
*						

Window "Model"

 Model Entry

Model Number

Model Name

Description

Article Number

Article Name


Quick Search

Short Name

	Range Rank	Range Number	Range Name
*			

	Technical Characteristic Nb	Technical Characteristic Name	Value	Value Type
*				

Window "Task"

 Task Entry

Task Number

Task Name

Description

Site Name

Workshop Name

Range Number

Range Name

Finished Article Number

Finished Article Name

Finished Model Type

Finished Model Name

Description

Quick Search

Short Name

	Item Number to do it	Item Name to do it	Model type to do it	Model Name to do it	Quantity	Quantity Type
*						

< >

	Workplace Number	Workplace Name	Workstation type	Estimated Time of Execution	Type temps
*					

Save Delete Quit

Window “Range of Manufacturing”

Description RAnge Entry

Range Number
Range Name
Description
Mother Range Number

Quick Search
Short Name

Save

Delete

Quit

	Task Rank	Task Number	Task Name
*			

Window "Availability"

The screenshot shows a window titled "Availability" with standard window controls (minimize, maximize, close) in the top right corner. The window contains the following elements:

- Availability Number:** A text input field.
- Availability Name:** A text input field.
- Reference Recherche Rapide:** A text input field.
- Recherche Rapide:** A text input field.
- Days:** A vertical list of days with checkboxes:
 - Monday
 - Tuesday
 - Wednesday
 - Thursday
 - Friday
 - Saturday
 - Sunday
- Time and Breaks:** Four text input fields labeled "Beginning Hour", "Ending Hour", "Breaks", and "Public Holiday".
- Entire Day:** A text input field.
- Unavailable:** A checkbox labeled "Unavailable".
- Buttons:** Three buttons labeled "Save", "Delete", and "Quit" are located at the bottom right.

Windows Management

Window "Stocks"

Stock Management

Article Number

Article Name

	Model Number	Model Name	Price	Price Type	Quantity	Quantity Type	Difference
*						▼	

Total Quantity ▼

Save Delete Quit

Window "Supplier Order Entry"

Supplier Order Entry

Order Number

Supplier Number

Supplier Name

Delivery Day

Delivery Hour

Delivery WorkPlace

Delivery finished


	Article Number	Article Name	Model Number	Model Name	Quantity	Quantity Type	Price	Price Type
*						▼		

Save

Delete

Quit

Entering Entries

 Entering Entries

Order Number

Supplier Number

Supplier Name

Delivery Day

Delivery Hour

Delivery WorkPlace

	Article Number	Article Name	Model Number	Model Name	Quantity	Quantity Type	Price	Price Type
*						▼		

Save Delete Quit

Window "Customer Order Entry"

Customer Order Entry

Order Number

Customer Number

Customer Name

Delivery Day

Delivery Hour

Delivery WorkPlace

Order State

Finished

Started

Finie

Delivery order printed

Billing Printed

	Article Number	Article Name	Model Number	Model Name	Quantity	Quantity Type	Price	Price Type
*						▼		

Window “Entering a delivery”

Entering a delivery

Order Number

Customer Number

Customer Name

Delivery Day

Delivery Hour

Delivery WorkPlace

	Article Number	Article Name	Model Number	Model Name	Quantity	Quantity Type	Price	Price Type
*						▼		

Save Delete Quit

Supervisor

Accueil Superviseur

Supervisor[Type Supervisor]

Workstation User

Bar Code Manufacturing Number Order Number End Task Time

Article Name Model Name Order Number Customer Name Task Name Task End Time

Item Modèle Order Number Customer Name Task Task End Time

Task Description

Number	Task Name	Article Number	Article Name	Model Number	Model Name	Workplace Number	Workplace Name	Quantity	Quantity Type	Debut Day	Debut Time	Duration	Duration Type

Article Number	Article Name	Model Number	Model Name	Quantity	Quantity Type	Manufacturing Number	Customer Number	Customer Name	Emergency	Manufacturing State	Debut Day	Debut 1

Technical Characteristic Name	Wanted Value	Produced Value	Value Type

Made Quantity / Total Quantity

/ Type Value

Task Finished Manufacturing Finished

Printing Manufacturing Quit

Urgent Order
Necessary "ITEM" Missing
Item for order No Order canceled
Etc..

DataBase Administration Document

DATABASE DESCRIPTION

To help Database Administrators, here are the Tables of WMI Database

The Database Table is designed with french names.

To help international workers you have in red the traduction

Table	Champs	Traduction	Lenght
Entreprise	NoEntreprise NomEntreprise RechercheNomEntreprise NoSIRET Adr1 Adr2 CP Pays Description Email NoTel NoFax SiteWeb MentionLegale Contact	Company CompanyNb CompanyName QuickSearch SIRETNumber Address 1 Address2 ZipCode Country Description Email TelNumber FaxNumber WebSite LegalNotice Contact	PRIMARY KEY VARCHAR(20) VARCHAR(10) VARCHAR(20) VARCHAR(30) VARCHAR(30) INT VARCHAR(20) VARCHAR(60) VARCHAR(20) VARCHAR(20) VARCHAR(20) VARCHAR(20) VARCHAR(40) VARCHAR(80) VARCHAR(30)
Adresse	NoAdresse NomAdresse Adr1 Adr2 CP Pays Description Email NoTel NoFax SiteWeb Contact	Address NbCompany AddressName Address 1 Address 2 ZipCode Country Description Email TelNumber FaxNumber WebSite Contact	PRIMARY KEY VARCHAR(20) VARCHAR(30) VARCHAR(30) INT VARCHAR(20) VARCHAR(60) VARCHAR(20) VARCHAR(20) VARCHAR(20) VARCHAR(20) VARCHAR(40) VARCHAR(30)
LiaisonEntrepriseSite	NoLiaisonEntrepriseSite	LinkCompanySiteNb	PRIMARY KEY

	NoSite NoEntreprise	SiteNb CompanyNb	INT INT
Site	NoSite NomSite RechercheNomSite NoAdresseSite NoAdresseLivraison NoAdresseFacturation Contact NoDisponibilite Description	SiteNb SiteName QuickSearch SiteAddressNb DeliveryAddressNb BillingAddressNb Contact AvailabilityNb Description	PRIMARY KEY VARCHAR(20) VARCHAR(10) INT INT INT VARCHAR(30) INT VARCHAR(60)
LiaisonSiteAtelier	NoLiaisonSiteAtelier NoSite NoAtelier	LinkSiteWorkshop SiteNb WorkshopNb	PRIMARY KEY INT INT
Atelier	NoAtelier NomAtelier RechercheNomAtelier Description NoPoste NoDisponibilite Contact TypeAtelier	Workshop WorkshopNb WorkshopName QuickSearch Description WorkSationNb Availability Contact WorkshopType	PRIMARY KEY VARCHAR(20) VARCHAR(10) VARCHAR(60) INT INT VARCHAR(30) INT
LiaisonAtelierPoste	NoLiaisonAtelierPoste NoAtelier NoPoste NoTypePoste	LinkWorkshopWorkstation LinkWorkshopWorkstationNb WorkshopNb WorkStationNb TypeWorkstationNb	PRIMARY KEY INT INT INT
Poste	NoPoste NomPoste RechercheNomPoste Description NoDisponibilite NoType NoMachineEntree NoMachineSortie AdresseIP Superviseur	Workstation WorkstationNb WorkstationName Quicksearch Description AvailabilityNb TypeNb MachineInNb MachineOutNb IPAddress Supervisor	PRIMARY KEY VARCHAR(20) VARCHAR(10) VARCHAR(60) INT INT INT INT VARCHAR(20) INT

TypePoste	TypeWorkstation	
Id	Id	PRIMARY KEY
NoTypePoste	TypeWorkstationNb	INT
NomTypePoste	TypeWorkstationName	VARCHAR(20)
RechercheNomtypePoste	QuickSearch	VARCHAR(10)
Description	Description	VARCHAR(60)
NoTache	taskNb	INT
NoDisponibilite	AvailabilityNb	INT
NoPoste	WorkstationNb	INT
Client	Customer	
NoClient	CustomerNb	PRIMARY KEY
NomClient	CustomerName	VARCHAR(20)
RechercheNomClient	QuickSearch	VARCHAR(10)
NoAdresseClient	CustomerAddressNb	INT
NoAdresseLivraison	DeliveryAddressNb	INT
NoAdresseFacturation	BillingAddressNb	INT
Description	Description	VARCHAR(60)
LiaisonArticleClient	LinkItemCustomer	
NoLiaisonArticleClient	NoLinkItemCustomer	PRIMARY KEY
NoArticle	ItemNb	INT
NoClient	CustomerNb	INT
TypeModele	ModeleTypeNb	INT
PrixClient	CustomerPrice	DOUBLE
TypePrix	PriceType	INT
DelaisDeLivraison	DeliveringDeadline	TIME
Fournisseur	Supplier	
NoFournisseur	SupplierNb	PRIMARY KEY
NomFournisseur	SupplierName	VARCHAR(20)
RechercheNomFournisseur	QuickSearch	VARCHAR(10)
NoAdresseFournisseur	SupplierAddressNb	INT
NoAdresseFacturation	BillingAddressNb	INT
NoDetail	DetailNb	INT
Description	Description	VARCHAR(60)
LiaisonArticleFournisseur	LinkItemSupplier	
NoLiaisonArticleFournisseur	LinkItemSupplierNb	PRIMARY KEY
NoArticle	ItemNb	INT
NoFournisseur	SupplierNb	INT
TypeModele	ModelType	INT
PrixFournisseur	SupplierPrice	DOUBLE
TypePrix	PriceType	INT
DelaiDeLivraison	DeliveringDeadline	TIME
LiaisonArticleModele	LinkItemModel	

NoLiaisonArticleModele NoArticle TypeModele Prix TypePrix DelaiDeLivraison	LinkItemModelNb ItemNb ModelType Price PriceType DeliveringDeadline	PRIMARY KEY INT INT DOUBLE INT TIME
DetailFournisseur NoDetail NoCommande NoArticle TypeModele Quantite QuantitéLivrée TypeQuantite Prix PrixLivré TypePrix	SupplierDetails Details Ordernb ItemNumber ModelType Quantity deliverdQuantity QuantityType Price DeliveredPrice PriceType	PRIMARY KEY INT INT INT DOUBLE DOUBLE INT DOUBLE DOUBLE INT
Detail Client NoDetail NoCommande NoArticle TypeModele Quantite QuantitéLivrée TypeQuantite Prix PrixLivré TypePrix	Customer detail Details Ordernb ItemNumber ModelType Quantity deliverdQuantity QuantityType Price DeliveredPrice PriceType	PRIMARY KEY INT INT INT DOUBLE DOUBLE INT DOUBLE DOUBLE INT
CommandeFournisseur NoCommandeFournisseur NoFournisseur DateLivraison HeureLivraison DateCommande NoPosteLivraison EtatCommande	SupplierOrder SupplierOrderNb SupplierOrderName DeliveringDeadlineDate DeliveringDeadlineHour SupplyDeadlineDate DeliveringWorkstationNb SupplyState	PRIMARY KEY INT DATE TIME DATE INT INT
LienCommandeFournisseurDetail NoLienCommandeFournisseurDetail NoCommandeFournisseur NoDetail	linkSupplierOrderDetail linkSupplierOrderDetail SupplierOrderNb DetailNb	PRIMARY KEY INT INT INT
Entrees NoCommandeFournisseur	Entries SupplierOrderNb	PRIMARY KEY

NoFournisseur DateLivraison HeureLivraison DateCommande NoPosteLivraison NoDetail EtatCommande	SupplierOrderName DeliveringDeadlineDate DeliveringDeadlineHour SupplyDeadlineDate DeliveringWorkstationNb DetailNb SupplyState	INT DATE TIME DATE INT INT INT
LienEntreesDetail NoLienEntreesDetail NoCommandeFournisseur NoDetail	linkentriesDetail linkentriesDetailNb SupplierOrderNb DetailNb	PRIMARY KEY INT INT INT
CommandeClient NoCommandeClient NoClient DateLivraison HeureLivraison DateCommande NoPosteExpedition EtatCommande	CustomerOrder CustomerOrderNb CustomerOrderName DeliveringDeadlineDate DeliveringDeadlineHour SupplyDeadlineDate ShipmentsWorkstationNb SupplyState	PRIMARY KEY INT DATE TIME DATE INT INT
LienCommandeClientDetail NoLienCommandeClientDetail NoCommandeClient NoDetail	linkCustomerOrderDetail linkShippingDetailsNb CustomerorderNb DetailNb	PRIMARY KEY INT INT INT
Livraison NoCommandeClient NoClient DateLivraison HeureLivraison DateCommande NoPosteExpedition EtatCommande BonLivraisonEdite FactureEditee	Delivery CustomerOrderNb CustomerOrderName DeliveringDeadlineDate DeliveringDeadlineHour SupplyDeadlineDate ShipmentsWorkstationNb SupplyState DeliveryOrderPrinted BillPrinted	PRIMARY KEY INT DATE TIME DATE INT INT INT INT INT
LienLivraisonDetail NoLienLivraisonDetail NoCommande NoDetail	linkShippingDetails linkShippingDetailsNb ShippingNb DetailNb	PRIMARY KEY INT INT INT
Article NoArticle NomArticle RechercheNomArticle	Item ItemNb ItemName QuickSearch	PRIMARY KEY VARCHAR(20) VARCHAR(10)

	Description Longueur Largeur Hauteur Poids PrixFixe PoidsVariable CodeBarre CategorieArticle TypeConditionnement NoArticleCondInf QuantiteCondInf TypeQuantiteCondInf TVA	Description Length Width Height Weight FixedPrice VariableWeight BarCode ItemCategory TypeOfPackaging ItemNbPackagingLower QuantityPackagingLower QuantityTypePackagingLower VAT	VARCHAR(60) DOUBLE DOUBLE DOUBLE DOUBLE INT INT INT VARCHAR(20) INT INT INT INT DOUBLE
Modele	TypeModele NomModele RechercheNomModele Description NoGamme RangGamme	Model ModeType ModelName Quicksearch Description ProductLineNb ProductLineRank	PRIMARY KEY VARCHAR(20) VARCHAR(10) VARCHAR(60) INT INT
Disponibilite	NoDisponibilite NomDisponibilite RechercheNomDisponibilite JourHeureSemaine HeureDebut HeureFin PauseDejeunerDebut PauseDejeunerFin Pauses JourFerie JourEntier Indisponibilite	Availability AvailabilityNb AvailabilityName QuickSearch WeekDay StartingHour FinishingHour BreakStarting BreakFinishing Breaks PublicDay WholeDay Unavailable	PRIMARY KEY VARCHAR(20) VARCHAR(10) VARCHAR(60) TIME TIME TIME TIME VARCHAR(60) VARCHAR(100) DATE INT
Stocks	Id NoArticle TypeModele Prix TypePrix Quantite TypeQuantite QuantiteTotale DateMiseAJour	Stocks Id ItemNb ModelType Price PriceType Quantity QuantityType TotalQuantity Update Date	PRIMARY KEY INT INT DOUBLE INT DOUBLE INT DOUBLE DATE

Variation	PriceChanges	DOUBLE
Tache		
Id	Id	PRIMARY KEY
NoTache	TaskNb	INT
NomTache	TaskName	VARCHAR(20)
RechercheNomTache	QuickSearch	VARCHAR(10)
Description	Description	VARCHAR(60)
TempsExec	RunTime	TIME
NoGamme	ProductLineNb	INT
NoArticleFini	FinishedItemNb	INT
NoTypePoste	TypePosteNb	INT
NoArticlePourLeFaire	ItemToDoltNb	INT
TypeModeleFini	FinishedModelType	INT
typeModelePourLeFaire	ModelTypeToDolt	INT
Gamme	ProductLine	
NoGamme	ProductLineNb	PRIMARY KEY
NomGamme	ProductLineName	VARCHAR(20)
RechercheNomGamme	QuickSearch	VARCHAR(10)
Description	Description	VARCHAR(60)
RangTache	TaskRank	INT
NoTache	TaskNb	INT
NoGammeMere	ProductLineMother	INT
LiaisonGammeTache	LinkProductLineTask	
NoLiaisonGammeTache	LinkProductLineTask	PRIMARY KEY
NoGamme	ProductLineNb	INT
NoTache	TaskNb	INT
RangTache	TaskRank	INT
LiaisonModeleGamme	LinkProductLineModel	
NoLiaisonModeleGamme	LinkProductLineModelNb	PRIMARY KEY
NoArticle	ItemNb	INT
TypeModele	ModelType	INT
NoGamme	ProductLineNb	INT
LiaisonTachePoste	LinkTaskWorkstation	PRIMARY KEY
NoLiaisonTachePoste	NoLinkTaskWorstation	INT
NoPoste	WorkstationNb	INT
NoTypePoste	TypeWorkstationNb	INT
TempsExecution	Duration	TIME
TypeTemps	TimeType	INT
Fabrication		
Id	Id	PRIMARY KEY
NoFabrication	ProductionNb	INT

NomFabrication	ProductionName	VARCHAR(20)
NoGamme	ProductLineNb	INT
NoPoste	Workstation?n	INT
Description	Description	VARCHAR(60)
DateDebut	StartingDate	DATE
HeureDebut	StartingTime	TIME
NoCommande	OrderingNumber	INT
EtatFabrication	ProductionState	INT
Urgence	Urgency	INT
NoTache	TaskNb	INT
NoArticle	ItemNb	INT
TypeModele	ModelType	INT
Quantite	Quantity	DOUBLE
TypeQuantite	QuantityType	INT
DateFinPrevue	FinishingDateExpected	DATE
HeureFinPrevue	FinishingTimeExpected	TIME
NoTypePoste	TypeWorkstationNb	INT
NoPoste	WorkstationNb	INT
CaracteristiquesTechniques	TechnicalCharacteristics	
NoCaracteristiqueTechnique	TechnicalCharacteristicsNb	PRIMARY KEY
NomCaracteristiqueTechnique	TechnicalCharacteristicsName	VARCHAR(20)
RechercheNomCaracteristiqueTechnique	QuickSearch	VARCHAR(10)
ValeurVoulue	WantedValue	DOUBLE
TypeValeur	TypeValue	INT
LienCommandeCarTech	LinkOrderTechCar	
NoLienCommandeCarTech	LinkOrderTechCarNb	PRIMARY KEY
NoArticle	ItemNb	INT
TypeModele	ModelType	INT
NoCommande	OrderingNb	INT
Valeur	Value	DOUBLE
NoCaracteristiqueTechnique	TechnicalCharacteristicsNb	INT
LiaisonModeleCarTech	LinkItemTechCar	PRIMARY KEY
NoArticle	ItemNb	INT
NoCaracteristiqueTechnique	TechnicalCharacteristicsNb	INT
ValeurVoulue	WantedValue	DOUBLE
NoLiaisonModeleCarTeh	LinkItemTechCarNb	INT
Machine	Machine	
NoMachine	MachineNb	PRIMARY KEY
RechercheNomMachine	QuickSearch	VARCHAR(10)
NomMarchine	MachineName	VARCHAR(20)
NoTypeMachine	TypeMachineNb	INT
AdresseIP	IPAddress	VARCHAR(20)

Traduction French-English

FICHIER AFFICHAGE DES LANGUES

[Entete]

NbSection = 3

[Section1]

L1NomSection = page principale

L2NomSection = main page

L3NomSection =

L4NomSection =

NbCleSection = 193

L1NomCle1 = Fichier

L2NomCle1 = File

L3NomCle1 =

L4NomCle1 =

L1NomCle2 = Utilisateur

L2NomCle2 = User

L3NomCle2 =

L4NomCle2 =

L1NomCle3 = Droits

L2NomCle3 = Rights

L3NomCle3 =

L4NomCle3 =

L1NomCle4 = Langage

L2NomCle4 = Language

L3NomCle4 =

L4NomCle4 =

L1NomCle5 = Administrateur

L2NomCle5 = Administrator

L3NomCle5 =

L4NomCle5 =

L1NomCle6 = Utilisateur

L2NomCle6 = User

L3NomCle6 =

L4NomCle6 =

L1NomCle7 = Saisie Données

L2NomCle7 = Data Entry

L3NomCle7 =

L4NomCle7 =

L1NomCle8 = Entreprise

L2NomCle8 = Company

L3NomCle8 =

L4NomCle8 =

L1NomCle9 = Atelier

L2NomCle9 = Workshop

L3NomCle9 =

L4NomCle9 =

L1NomCle10 = Poste

L2NomCle10 = Workplace

L3NomCle10 =

L4NomCle10 =

L1NomCle11 = Article

L2NomCle11 = Item

L3NomCle11 =

L4NomCle11 =

L1NomCle12 = Modèle

L2NomCle12 = Model

L3NomCle12 =

L4NomCle12 =

L1NomCle13 = Site

L2NomCle13 = Site

L3NomCle13 =

L4NomCle13 =

L1NomCle14 = Gamme
L2NomCle14 = Range of Manufacturing
L3NomCle14 =
L4NomCle14 =

L1NomCle15 = Type de Modèle
L2NomCle15 = Model Type
L3NomCle15 =
L4NomCle15 =

L1NomCle16 = Descriptif Modèle
L2NomCle16 = Model Description
L3NomCle16 =
L4NomCle16 =

L1NomCle17 = Stocks
L2NomCle17 = Stocks
L3NomCle17 =
L4NomCle17 =

L1NomCle18 = Approvisionnements
L2NomCle18 = Supplies
L3NomCle18 =
L4NomCle18 =

L1NomCle19 = Gestion de Production
L2NomCle19 = Production Management
L3NomCle19 =
L4NomCle19 =

L1NomCle20 = Commandes
L2NomCle20 = Order
L3NomCle20 =
L4NomCle20 =

L1NomCle21 = Etats et Editions
L2NomCle21 = States and Editions
L3NomCle21 =
L4NomCle21 =

L1NomCle22 = Stocks
L2NomCle22 = Stocks
L3NomCle22 =
L4NomCle22 =

L1NomCle23 = Approvisionnements
L2NomCle23 = Supplies
L3NomCle23 =
L4NomCle23 =

L1NomCle24 = Gestion de Production
L2NomCle24 = Production Management
L3NomCle24 =
L4NomCle24 =

L1NomCle25 = Gamme
L2NomCle25 = Range of Manufacturing
L3NomCle25 =
L4NomCle25 =

L1NomCle26 = Bon de Fabrication
L2NomCle26 = Good of Manufacturing
L3NomCle26 =
L4NomCle26 =

L1NomCle27 = Commandes
L2NomCle27 = Orders
L3NomCle27 =
L4NomCle27 =

L1NomCle28 = Expéditions
L2NomCle28 = Shipping Bill
L3NomCle28 =
L4NomCle28 =

L1NomCle29 = Statistiques
L2NomCle29 = Statistics
L3NomCle29 =
L4NomCle29 =

L1NomCle30 = Paramètres
L2NomCle30 = Parameters
L3NomCle30 =
L4NomCle30 =

L1NomCle31 = Paramètres bases de données
L2NomCle31 = Database Parameters
L3NomCle31 =
L4NomCle31 =

L1NomCle32 = Saisie Tablette
L2NomCle32 = Tablets
L3NomCle32 =
L4NomCle32 =

L1NomCle41 = Description
L2NomCle41 = Description
L3NomCle41 =
L4NomCle41 =

L1NomCle33 = Saisie Ordinateur
L2NomCle33 = Computers
L3NomCle33 =
L4NomCle33 =

L1NomCle42 = Dimensions
L2NomCle42 = Dimensions
L3NomCle42 =
L4NomCle42 =

L1NomCle34 = Numéro d'Article'
L2NomCle34 = Article Number
L3NomCle34 =
L4NomCle34 =

L1NomCle43 = Dimensions
L2NomCle43 = Dimensions
L3NomCle43 =
L4NomCle43 =

L1NomCle35 = Nom Article
L2NomCle35 = Article Name
L3NomCle35 =
L4NomCle35 =

L1NomCle44 = Longueur
L2NomCle44 = Lenght
L3NomCle44 =
L4NomCle44 =

L1NomCle36 = Numéro Modèle
L2NomCle36 = Model Number
L3NomCle36 =
L4NomCle36 =

L1NomCle45 = Largeur
L2NomCle45 = Width
L3NomCle45 =
L4NomCle45 =

L1NomCle37 = Nom Modèle
L2NomCle37 = Model Name
L3NomCle37 =
L4NomCle37 =

L1NomCle46 = Hauteur
L2NomCle46 = Height
L3NomCle46 =
L4NomCle46 =

L1NomCle38 = Prix (HT)
L2NomCle38 = Price
L3NomCle38 =
L4NomCle38 =

L1NomCle47 = Poids
L2NomCle47 = Weight
L3NomCle47 =
L4NomCle47 =

L1NomCle39 = Quantité
L2NomCle39 = Quantity
L3NomCle39 =
L4NomCle39 =

L1NomCle48 = Poids Variable
L2NomCle48 = Variable Weight
L3NomCle48 =
L4NomCle48 =

L1NomCle40 = Quantité Totale
L2NomCle40 = Total Quantity
L3NomCle40 =
L4NomCle40 =

L1NomCle49 = Type d'Article
L2NomCle49 = Article Type
L3NomCle49 =
L4NomCle49 =

L1NomCle50 = Produit acheté
L2NomCle50 = Purchased Product
L3NomCle50 =
L4NomCle50 =

L1NomCle51 = Produit Semi-Fini
L2NomCle51 = Semi Finished Product
L3NomCle51 =
L4NomCle51 =

L1NomCle52 = Produit Fini
L2NomCle52 = Finished Product
L3NomCle52 =
L4NomCle52 =

L1NomCle53 = Numéro Fournisseur
L2NomCle53 = Supplier Number
L3NomCle53 =
L4NomCle53 =

L1NomCle54 = Nom Fournisseur
L2NomCle54 = Supplier Name
L3NomCle54 =
L4NomCle54 =

L1NomCle55 = Nom tâche
L2NomCle55 = Task Name
L3NomCle55 =
L4NomCle55 =

L1NomCle56 = Numéro Atelier
L2NomCle56 = Workshop Number
L3NomCle56 =
L4NomCle56 =

L1NomCle57 = Nom Atelier
L2NomCle57 = Workshop Name
L3NomCle57 =
L4NomCle57 =

L1NomCle58 = Numéro Site
L2NomCle58 = Site Number
L3NomCle58 =

L4NomCle58 =

L1NomCle59 = Nom Site
L2NomCle59 = Site Name
L3NomCle59 =
L4NomCle59 =

L1NomCle60 = Numéro Poste
L2NomCle60 = Workplace Number
L3NomCle60 =
L4NomCle60 =

L1NomCle61 = Nom Poste
L2NomCle61 = Workplace Name
L3NomCle61 =
L4NomCle61 =

L1NomCle62 = Ville
L2NomCle62 = City
L3NomCle62 =
L4NomCle62 =

L1NomCle63 = Pays
L2NomCle63 = Country
L3NomCle63 =
L4NomCle63 =

L1NomCle64 = Temps de livraison
L2NomCle64 = Delivery Time
L3NomCle64 =
L4NomCle64 =

L1NomCle65 = Adresse 1
L2NomCle65 = Address 1
L3NomCle65 =
L4NomCle65 =

L1NomCle66 = Adresse 2
L2NomCle66 = Address 2
L3NomCle66 =
L4NomCle66 =

L1NomCle67 = Code Postale
L2NomCle67 = ZIP Code
L3NomCle67 =

L4NomCle67 =

L1NomCle68 = Livraison

L2NomCle68 = Delivery

L3NomCle68 =

L4NomCle68 =

L1NomCle69 = Facturation

L2NomCle69 = Billing

L3NomCle69 =

L4NomCle69 =

L1NomCle70 = Type Prix

L2NomCle70 = Price Type

L3NomCle70 =

L4NomCle70 =

L1NomCle71 = Numéro de Commande

L2NomCle71 = Order Number

L3NomCle71 =

L4NomCle71 =

L1NomCle72 = Etat de la Commande

L2NomCle72 = Order State

L3NomCle72 =

L4NomCle72 =

L1NomCle73 = Forcer démarrage de la commande

L2NomCle73 = Force Order Start

L3NomCle73 =

L4NomCle73 =

L1NomCle74 = Jour de Livraison

L2NomCle74 = Delivery Day

L3NomCle74 =

L4NomCle74 =

L1NomCle75 = heure de Livraison

L2NomCle75 = Delivery Hour

L3NomCle75 =

L4NomCle75 =

L1NomCle76 = Poste de livraison

L2NomCle76 = Delivery WorkPlace

L3NomCle76 =

L4NomCle76 =

L1NomCle77 = Etat Commande

L2NomCle77 = Order State

L3NomCle77 =

L4NomCle77 =

L1NomCle78 = Non commencée

L2NomCle78 = Not Started

L3NomCle78 =

L4NomCle78 =

L1NomCle79 = Commencée

L2NomCle79 = Started

L3NomCle79 =

L4NomCle79 =

L1NomCle80 = Finie

L2NomCle80 = Finished

L3NomCle80 =

L4NomCle80 =

L1NomCle81 = Numéro Gamme

L2NomCle81 = Range Number

L3NomCle81 =

L4NomCle81 =

L1NomCle82 = Nom Gamme

L2NomCle82 = Range Name

L3NomCle82 =

L4NomCle82 =

L1NomCle83 = Rang Tâche

L2NomCle83 = Task Rank

L3NomCle83 =

L4NomCle83 =

L4NomCle90 =

L1NomCle84 = Numéro Tâche

L2NomCle84 = Task Number

L3NomCle84 =

L4NomCle84 =

L1NomCle91 = marge d'erreur

L2NomCle91 = Margin of Error

L3NomCle91 =

L4NomCle91 =

L1NomCle85 = Heure début

L2NomCle85 = Debut Time

L3NomCle85 =

L4NomCle85 =

L1NomCle92 = Durée

L2NomCle92 = Duration

L3NomCle92 =

L4NomCle92 =

L1NomCle86 = Date début

L2NomCle86 = Debut Day

L3NomCle86 =

L4NomCle86 =

L1NomCle93 = Type Durée

L2NomCle93 = Duration Type

L3NomCle93 =

L4NomCle93 =

L1NomCle87 = Etat Fabrication

L2NomCle87 = Manufacturing State

L3NomCle87 =

L4NomCle87 =

L1NomCle94 = Date Fin Prévue

L2NomCle94 = Expected End Date

L3NomCle94 =

L4NomCle94 =

L1NomCle88 = Urgence

L2NomCle88 = Emergency

L3NomCle88 =

L4NomCle88 =

L1NomCle95 = Heure fin Prévue

L2NomCle95 = Expected End Hour

L3NomCle95 =

L4NomCle95 =

L1NomCle89 = Maintenant

L2NomCle89 = Now

L3NomCle89 =

L4NomCle89 =

L1NomCle96 = Nom Entreprise

L2NomCle96 = Company Name

L3NomCle96 =

L4NomCle96 =

L1NomCle90 = Temps tendu

L2NomCle90 = Tense Time

L3NomCle90 =

L1NomCle97 = Numéro Siret
L2NomCle97 = SIRET Number
L3NomCle97 =
L4NomCle97 =

L1NomCle98 = Temps estimé d'exécution'
L2NomCle98 = Estimated Time of Execution
L3NomCle98 =
L4NomCle98 =

L1NomCle99 = Numéro article fini
L2NomCle99 = Finished Article Number
L3NomCle99 =
L4NomCle99=

L1NomCle100 = Nom Article Fini
L2NomCle100 = Finished Article Name
L3NomCle100 =
L4NomCle100 =

L1NomCle101 = Numéro d'article pour le faire'
L2NomCle101 = Article Number to do it
L3NomCle101 =
L4NomCle101 =

L1NomCle102 = Numéro Modèle pour le faire
L2NomCle102 = Model Number to do it
L3NomCle102 =
L4NomCle102 =

L1NomCle103 = Nom article pour le faire
L2NomCle103 = Article Number to do it
L3NomCle103 =
L4NomCle103 =

L1NomCle104 = Nom Modèle pour le faire
L2NomCle104 = Model Number to do it
L3NomCle104 =
L4NomCle104 =

L1NomCle105 = Numéro Client
L2NomCle105 = Customer Number

L3NomCle105 =
L4NomCle105 =

L1NomCle106 = Nom Client
L2NomCle106 = Customer Name
L3NomCle106 =
L4NomCle106 =

L1NomCle107 = Description
L2NomCle107 = Description
L3NomCle107 =
L4NomCle107 =

L1NomCle108 = Fournisseur
L2NomCle108 = Supplier
L3NomCle108 =
L4NomCle108 =

L1NomCle109 = Client
L2NomCle109 = Customer
L3NomCle109 =
L4NomCle109 =

L1NomCle110 = Superviseur
Approvisionnement
L2NomCle110 = Supply Supervisor
L3NomCle110 =
L4NomCle110 =

L1NomCle111 = Connexion EDI
L2NomCle111 = EDI Connection
L3NomCle111 =
L4NomCle111 =

L1NomCle112 = Superviseur de production
L2NomCle112 = Product Supervisor
L3NomCle112 =
L4NomCle112 =

L1NomCle113 = Superviseur expéditions
L2NomCle113 = Shipping Supervisor
L3NomCle113 =
L4NomCle113 =

L1NomCle114 = Numéro Entreprise

L2NomCle114 = Company Number
L3NomCle114 =
L4NomCle114 =

L1NomCle115 = Enregistrer
L2NomCle115 = Save
L3NomCle115 =
L4NomCle115 =

L1NomCle116 = Effacer
L2NomCle116 = Delete
L3NomCle116 =
L4NomCle116 =

L1NomCle117 = Quitter
L2NomCle117 = Quit
L3NomCle117 =
L4NomCle117 =

L1NomCle118 = Type Poste
L2NomCle118 = Workingstation type
L3NomCle118 =
L4NomCle118 =

L1NomCle119 = Superviseur
L2NomCle119 = Supervisor
L3NomCle119 =
L4NomCle119 =

L1NomCle120 = Connexion EDI
L2NomCle120 = EDI Connexion
L3NomCle120 =
L4NomCle120 =

L1NomCle121 = Edition bon de transport
L2NomCle121 = Shipping bill Edition
L3NomCle121 =
L4NomCle121 =

L1NomCle122 = Statistiques de Direction
L2NomCle122 = CEO Statistics
L3NomCle122 =
L4NomCle122 =

L1NomCle123 = Statistiques commerciales

L2NomCle123 = Trading Statistics
L3NomCle123 =
L4NomCle123 =

L1NomCle124 = Statistiques de production
L2NomCle124 = Manufacturing Statistics
L3NomCle124 =
L4NomCle124 =

L1NomCle125 = Charge Postes
L2NomCle125 = Workingstation Workload
L3NomCle125 =
L4NomCle125 =

L1NomCle126 = Récapitulatif d'une journée
L2NomCle126 = Day overview
L3NomCle126 =
L4NomCle126 =

L1NomCle127 = Carnet de commande
L2NomCle127 = Order Book
L3NomCle127 =
L4NomCle127 =

L1NomCle128 = Produit vedette
L2NomCle128 = Star Item
L3NomCle128 =
L4NomCle128 =

L1NomCle129 = Email
L2NomCle129 = E-Mail
L3NomCle129 =
L4NomCle129 =

L1NomCle130 = Numéro de téléphone
L2NomCle130 = Phone Number
L3NomCle130 =
L4NomCle130 =

L1NomCle131 = Fax
L2NomCle131 = Fax Number
L3NomCle131 =
L4NomCle131 =

L1NomCle132 = Site Web

L2NomCle132 = Web Site
L3NomCle132 =
L4NomCle132 =

L2NomCle141 = Production Area
L3NomCle141 =
L4NomCle141 =

L1NomCle133 = Mention Légale
L2NomCle133 = Legal Notice
L3NomCle133 =
L4NomCle133 =

L1NomCle142 = Nom Tache
L2NomCle142 = Task Name
L3NomCle142 =
L4NomCle142 =

L1NomCle134 = Recherche Rapide
L2NomCle134 = Quick Search
L3NomCle134 =
L4NomCle134 =

L1NomCle143 = Machine
L2NomCle143 = Engine
L3NomCle143 =
L4NomCle143 =

L1NomCle135 = Nom Abrégé
L2NomCle135 = Short Name
L3NomCle135 =
L4NomCle135 =

L1NomCle144 = Adresse IP
L2NomCle144 = IP Address
L3NomCle144 =
L4NomCle144 =

L1NomCle136 = Type Poste
L2NomCle136 = Workstation Type
L3NomCle136 =
L4NomCle136 =

L1NomCle145 = Type de poste
L2NomCle145 = Workstation type
L3NomCle145 =
L4NomCle145 =

L1NomCle137 = Contact
L2NomCle137 = Contacts
L3NomCle137 =
L4NomCle137 =

L1NomCle146 = Réception
L2NomCle146 = Receipts
L3NomCle146 =
L4NomCle146 =

L1NomCle138 = TypeAtelier
L2NomCle138 = WorkshopType
L3NomCle138 =
L4NomCle138 =

L1NomCle147 = Production
L2NomCle147 = Manufacturing
L3NomCle147 =
L4NomCle147 =

L1NomCle139 = En parallèle
L2NomCle139 = Parallel
L3NomCle139 =
L4NomCle139 =

L1NomCle148 = Expéditions
L2NomCle148 = Shipping Station
L3NomCle148 =
L4NomCle148 =

L1NomCle140 = En ligne
L2NomCle140 = Line
L3NomCle140 =
L4NomCle140 =

L1NomCle149 = Gestion
L2NomCle149 = Management
L3NomCle149 =
L4NomCle149 =

L1NomCle141 = En îlot

L1NomCle150 = Listings

L2NomCle150 = listings
L3NomCle150 =
L4NomCle150 =

L1NomCle151 = No Article pour le faire
L2NomCle151 = Item Number to do it
L3NomCle151 =
L4NomCle151 =

L1NomCle152 = Lom Article pour le faire
L2NomCle152 = Item Name to do it
L3NomCle152 =
L4NomCle152 =

L1NomCle153 = type Modèle pour le faire
L2NomCle153 = Model type to do it
L3NomCle153 =
L4NomCle153 =

L1NomCle154 = Nom Modele pour le faire
L2NomCle154 = Model Name to do it
L3NomCle154 =
L4NomCle154 =

L1NomCle155 = Type Quantité
L2NomCle155 = Quantity Type
L3NomCle155 =
L4NomCle155 =

L1NomCle156 = Type Modele fini
L2NomCle156 = Finished Model Type
L3NomCle156 =
L4NomCle156 =

L1NomCle157 = Nom Modele Fini
L2NomCle157 = Finished Model Name
L3NomCle157 =
L4NomCle157 =

L1NomCle158 = No Disponibilité
L2NomCle158 = Availability Number
L3NomCle158 =
L4NomCle158 =

L1NomCle159 = Nom Disponibilité

L2NomCle159 = Availability name
L3NomCle159 =
L4NomCle159 =

L1NomCle160 = Numéro Gammme Mère
L2NomCle160 = Mother Range Number
L3NomCle160 =
L4NomCle160 =

L1NomCle161 =Rang Gamme
L2NomCle161 = Range Rank
L3NomCle161 =
L4NomCle161 =

L1NomCle162 = Nom Type Poste
L2NomCle162 = Type Poste Name
L3NomCle162 =
L4NomCle162 =

L1NomCle163 = Valeur
L2NomCle163 = Value
L3NomCle163 =
L4NomCle163 =

L1NomCle164 = Type Valeur
L2NomCle164 = Value Type
L3NomCle164 =
L4NomCle164 =

L1NomCle165 = No Caractéristique technique
L2NomCle165 = Technical Characteristic Nb
L3NomCle165 =
L4NomCle165 =

L1NomCle166 = Nom Caractéristique
Technique
L2NomCle166 = Technical Characteristic Name
L3NomCle166 =
L4NomCle166 =

L1NomCle167 = Type Poids
L2NomCle167 = Weight type
L3NomCle167 =
L4NomCle167 =

L1NomCle168 = Poids fixe
L2NomCle168 = Fixed weight
L3NomCle168 =
L4NomCle168 =

L1NomCle169 = TVA
L2NomCle169 = VAT
L3NomCle169 =
L4NomCle169 =

L1NomCle170 = Code Barre
L2NomCle170 = Bar Code
L3NomCle170 =
L4NomCle170 =

L1NomCle171 = Type de conditionnement
L2NomCle171 = Packaging type
L3NomCle171 =
L4NomCle171 =

L1NomCle172 = Prix Fixe
L2NomCle172 = Fixed Price
L3NomCle172 =
L4NomCle172 =

L1NomCle173 = No Article Conditionnement
inférieur
L2NomCle173 = Lower Rank Item Number
L3NomCle173 =
L4NomCle173 =

L1NomCle174 = Nom Article
Conditionnement inférieur
L2NomCle174 = Lower Rank Item Name
L3NomCle174 =
L4NomCle174 =

L1NomCle175 = Quantité Conditionnement
inférieur
L2NomCle175 = Lower Rank Quantity
L3NomCle175 =
L4NomCle175 =

L1NomCle176 = Quantite Faite
L2NomCle176 = Made Quantity

L3NomCle176 =
L4NomCle176 =

L1NomCle177 = Quantite Totale
L2NomCle177 = Total Quantity
L3NomCle177 =
L4NomCle177 =

L1NomCle178 = Tache Terminée
L2NomCle178 = Task Finished
L3NomCle178 =
L4NomCle178 =

L1NomCle179 = Fabrication Terminée
L2NomCle179 = Manufacturing Finished
L3NomCle179 =
L4NomCle179 =

L1NomCle180 = Edition Fabrication
L2NomCle180 = Printing Manufacturing
L3NomCle180 =
L4NomCle180 =

L1NomCle181 = Valeur Voulu
L2NomCle181 = Wanted Value
L3NomCle181 =
L4NomCle181 =

L1NomCle182 = Valeur Produite
L2NomCle182 = Produced Value
L3NomCle182 =
L4NomCle182 =

L1NomCle183 = Manquant
L2NomCle183 = Missing
L3NomCle183 =
L4NomCle183 =

L1NomCle184 = No de Fabrication
L2NomCle184 = Manufacturing Number
L3NomCle184 =

L4NomCle184 =

L1NomCle185 = Délais de livraison

L2NomCle185 = Delivery Deadline

L3NomCle185 =

L4NomCle185 =

L1NomCle186 = Type Prix

L2NomCle186 = Price Type

L3NomCle186 =

L4NomCle186 =

L1NomCle187 = Livraison Effectuée

L2NomCle187 = Delivery finished

L3NomCle187 =

L4NomCle187 =

L1NomCle188 = Bon de Livraison édité

L2NomCle188 = Delivery order printed

L3NomCle188 =

L4NomCle188 =

L1NomCle189 = Facture édité

L2NomCle189 = Billing Printed

L3NomCle189 =

L4NomCle189 =

L1NomCle190 = Variation

L2NomCle190 = Difference

L3NomCle190 =

L4NomCle190 =

L1NomCle191 = Bond de fabrication édité

L2NomCle191 = Manufacturing Order Printed

L3NomCle191 =

L4NomCle191 =

L1NomCle192 = No De Fabrication

L2NomCle192 = Manufacturing Number

L3NomCle192 =

L4NomCle192 =

L1NomCle193 = Nom de Fabrication

L2NomCle193 = Manufacturing Name

L3NomCle193 =

L4NomCle193 =

[Section2]

L1NomSection = Nom des fenêtres

L2NomSection = Windows Name

L3NomSection =

L4NomSection =

NbCleSection = 17

L1NomCle1 = Gestion de stocks

L2NomCle1 = Stock Management

L3NomCle1 =

L4NomCle1 =

L1NomCle2 = Saisie Article

L2NomCle2 = Article Entry

L3NomCle2 =

L4NomCle2 =

L1NomCle3 = Saisie Atelier

L2NomCle3 = Workshop Entry

L3NomCle3 =

L4NomCle3 =

L1NomCle4 = Saisie Client

L2NomCle4 = Customer Entry

L3NomCle4 =

L4NomCle4 =

L1NomCle5 = Saisie Commande Client

L2NomCle5 = Customer Order Entry

L3NomCle5 =

L4NomCle5 =

L1NomCle6 = Saisie Commande Fournisseur

L2NomCle6 = Supplier Order Entry

L3NomCle6 =

L4NomCle6 =

L1NomCle7 = Saisie Descriptif Gamme

L2NomCle7 = Description RAnge Entry
L3NomCle7 =
L4NomCle7 =

L1NomCle8 = Saisie des entrées
L2NomCle8 = Entering Entries
L3NomCle8 =
L4NomCle8 =

L1NomCle9 = Saisie d'une Fabrication
L2NomCle9 = Entering a production
L3NomCle9 =
L4NomCle9 =

L1NomCle10 = Saisie d'une livraison
L2NomCle10 = Entering a delivery
L3NomCle10 =
L4NomCle10 =

L1NomCle11 = Saisie Entreprise
L2NomCle11 = Company Entry
L3NomCle11 =
L4NomCle11 =

L1NomCle12 = Saisie Fournisseur
L2NomCle12 = Supplier Entry
L3NomCle12 =
L4NomCle12 =

L1NomCle13 = Saisie Modèle
L2NomCle13 = Model Entry
L3NomCle13 =
L4NomCle13 =

L1NomCle14 = Saisie Poste
L2NomCle14 = Workplace Entry
L3NomCle14 =

L4NomCle14 =

L1NomCle15 = Saisie Site
L2NomCle15 = Site Entry
L3NomCle15 =
L4NomCle15 =

L1NomCle16 = Saisie Tache
L2NomCle16 = Task Entry
L3NomCle16 =
L4NomCle16 =

L1NomCle17 = Menu
L2NomCle17 = Menu
L3NomCle17 =
L4NomCle17 =

[Section3]

L1NomSection = MsgBox
L2NomSection = MsgBox
L3NomSection =
L4NomSection =
NbCleSection = 2

L1NomCle1 = Voulez-vous enregistrer ?
L2NomCle1 = Do you want to save ?
L3NomCle1 =
L4NomCle1 =

L1NomCle2 = Etes vous sur de vouloir
supprimer ?
L2NomCle2 = Are you sure you want to delete
?

L3NomCle2 =
L4NomCle2 =



CODES EN VISUAL BASIC

DataBase Programmation

```
Imports System
Imports System.Data
Imports System.Data.OleDb
```

```
Module FormulesRequetesSQL
```

```
    Public cnx As OleDbConnection
    Public cmd As OleDbCommand
    Public dta As OleDbDataAdapter
    Public sql As String
    Public dtt As DataTable
    Public dtr As DataRow
    Public rownum As Integer
    Public cnxstr As String
    Public cmdb As OleDbCommandBuilder
    Public dts As New DataSet
    Public NomTable As String
```

```
    Sub ConnexionDB()
        cnxstr = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source= " &
TableauParametres.Parametres(0).Données(1).Adr & " ;Persist Security Info=False"
        cnx = New OleDbConnection
        cnx.ConnectionString = cnxstr
        cnx.Open()
    End Sub
```

```
    Sub Requete()
        cmd = New OleDbCommand(sql)

        dta = New OleDbDataAdapter(cmd)

        cmd.Connection() = cnx
        dts.Clear()
        dta.Fill(dts, NomTable)

        dtt = dts.Tables(NomTable)
```

```
    End Sub
```

```
    Sub CreationTableDetailFournisseur()
        sql = "CREATE TABLE DetailFournisseur (NoDetail INT PRIMARY KEY NOT NULL,
NoCommande INT, NoArticle INT, TypeModele INT, Quantite DOUBLE, TypeQuantite INT, Prix
DOUBLE, TypePrix INT, QuantiteLivree DOUBLE, PrixLivree DOUBLE)"
        NomTable = "DetailFournisseur"
        Requete()
    End Sub
```

```
    Sub DropTableDetailFournisseur()
        sql = "DROP TABLE DetailFournisseur"
        NomTable = "DetailFournisseur"
        Requete()
    End Sub
```

```
    Sub CreationTableDetailClient()
```

```

    sql = "CREATE TABLE DetailClient (NoDetail INT PRIMARY KEY NOT NULL,
NoCommande INT, NoArticle INT, TypeModele INT, Quantite DOUBLE, TypeQuantite INT, Prix
DOUBLE, TypePrix INT, QuantiteLivree DOUBLE, PrixLivre DOUBLE)"
    NomTable = "DetailClient"
    Requete()
End Sub

Sub DropTableDetailClient()
    sql = "DROP TABLE DetailClient"
    NomTable = "DetailClient"
    Requete()
End Sub

Sub CreationTableAdresse()
    sql = "CREATE TABLE Adresse (NoAdresse INT NOT NULL IDENTITY(1,1) PRIMARY KEY,
NomAdresse VARCHAR(20), Adr1 VARCHAR(30), Adr2 VARCHAR(30), CP INT, Ville VARCHAR(20),
Pays VARCHAR(20), Description VARCHAR(60),Email VARCHAR(20), NoTel VARCHAR(20), NoFax
VARCHAR(20), SiteWeb VARCHAR(40), Contact VARCHAR(30))"
    NomTable = "Adresse"
    Requete()
End Sub

Sub DropTableAdresse()
    sql = "DROP TABLE Adresse"
    NomTable = "Adresse"
    Requete()

End Sub

Sub CreationTableEntreprise()
    sql = "CREATE TABLE Entreprise (NoEntreprise INT PRIMARY KEY NOT NULL,
NomEntreprise VARCHAR(20), RechercheNomEntreprise VARCHAR(10), NoSIRET VARCHAR(20),
Adr1 VARCHAR(30), Adr2 VARCHAR(30), CP INT, Ville VARCHAR(20), Pays VARCHAR(20),
Description VARCHAR(60), Email VARCHAR(20), NoTel VARCHAR(20), NoFax VARCHAR(20),
SiteWeb VARCHAR(40), MentionLegale VARCHAR(80), Contact VARCHAR(30))"
    NomTable = "Entreprise"
    Requete()
End Sub

Sub DropTableEntreprise()
    sql = "DROP TABLE Entreprise"
    NomTable = "Entreprise"
    Requete()
End Sub

Sub CreationLiaisonEntrepriseSite()
    sql = "CREATE TABLE LiaisonEntrepriseSite (NoLiaisonEntrepriseSite INT NOT
NULL IDENTITY(1, 1) PRIMARY KEY, NoSite INT, NoEntreprise INT)"
    NomTable = "LiaisonEntrepriseSite"
    Requete()
End Sub

Sub DropTableLiaisonEntrepriseSite()
    sql = "DROP TABLE LiaisonEntrepriseSite"
    NomTable = "LiaisonEntrepriseSite"
    Requete()
End Sub

Sub CreationTableSite()

    sql = "CREATE TABLE Site (NoSite INT PRIMARY KEY NOT NULL, NomSite
VARCHAR(20), RechercheNomSite VARCHAR(10), Description VARCHAR(60), NoAdresseSite INT,

```

```
NoAdresseLivraison INT, NoAdresseFacturation INT, Contact VARCHAR(30), NoDisponibilite INT)"
```

```
    NomTable = "Site"
```

```
    Requete()
```

```
End Sub
```

```
Sub DropTableSite()
```

```
    sql = "DROP TABLE Site"
```

```
    NomTable = "Site"
```

```
    Requete()
```

```
End Sub
```

```
Sub CreationTableLiaisonSiteAtelier()
```

```
    sql = "CREATE TABLE LiaisonSiteAtelier (NoLiaisonSiteAtelier INT NOT NULL  
IDENTITY(1, 1) PRIMARY KEY, NoSite INT, NoAtelier INT)"
```

```
    NomTable = "LiaisonEntrepriseSite"
```

```
    Requete()
```

```
End Sub
```

```
Sub DropTableLiaisonSiteAtelier()
```

```
    sql = "DROP TABLE LiaisonSiteAtelier"
```

```
    NomTable = "LiaisonSiteAtelier"
```

```
    Requete()
```

```
End Sub
```

```
Sub CreationTableLiaisonAtelierPoste()
```

```
    sql = "CREATE TABLE LiaisonAtelierPoste (NoLiaisonAtelierPoste INT NOT NULL  
IDENTITY(1, 1) PRIMARY KEY, NoAtelier INT, NoPoste INT)"
```

```
    NomTable = "LiaisonAtelierPoste"
```

```
    Requete()
```

```
End Sub
```

```
Sub DropTableLiaisonAtelierPoste()
```

```
    sql = "DROP TABLE LiaisonAtelierPoste"
```

```
    NomTable = "LiaisonAtelierPoste"
```

```
    Requete()
```

```
End Sub
```

```
Sub CreationTableClient()
```

```
    sql = "CREATE TABLE Client (NoClient INT PRIMARY KEY NOT NULL, NomClient  
VARCHAR(20), RechercheNomClient VARCHAR(10), NoAdresseClient INT, Description  
VARCHAR(60), NoAdresseLivraison INT, NoAdresseFacturation INT)"
```

```
    NomTable = "Client"
```

```
    Requete()
```

```
End Sub
```

```
Sub DropTableClient()
```

```
    sql = "DROP TABLE Client"
```

```
    NomTable = "Client"
```

```
    Requete()
```

```
End Sub
```

```
Sub CreationTableLiaisonArticleClient()
```

```
    sql = "CREATE TABLE LiaisonArticleClient (NoLiaisonArticleClient INT NOT NULL  
IDENTITY(1, 1) PRIMARY KEY, NoArticle INT, NoClient INT, TypeModele INT, PrixClient  
DOUBLE, TypePrix INT, DelaisDeLivraison TIME)"
```

```
    NomTable = "LiaisonArticleClient"
```

```
    Requete()
```

```
End Sub
```

```
Sub DropTableLiaisonArticleClient()
```

```

    sql = "DROP TABLE LiaisonArticleClient"
    NomTable = "LiaisonArticleClient"
    Requete()

End Sub

Sub CreationTableLiaisonArticleFournisseur()
    sql = "CREATE TABLE LiaisonArticleFournisseur (NoLiaisonArticleFournisseur INT
NOT NULL IDENTITY(1, 1) PRIMARY KEY, NoArticle INT, NoFournisseur INT, TypeModele INT,
PrixFournisseur DOUBLE, TypePrix INT, DelaisDeLivraison TIME)"
    NomTable = "LiaisonArticleFournisseur"
    Requete()
End Sub

Sub DropTableLiaisonArticleFournisseur()
    sql = "DROP TABLE LiaisonArticleFournisseur"
    NomTable = "LiaisonArticleFournisseur"
    Requete()

End Sub

Sub CreationTableLiaisonArticleModele()
    sql = "CREATE TABLE LiaisonArticleModele (NoLiaisonArticleModele INT NOT NULL
IDENTITY(1, 1) PRIMARY KEY, NoArticle INT, TypeModele INT, Prix DOUBLE, TypePrix INT,
DelaisDeLivraison TIME)"
    NomTable = "LiaisonArticleModele"
    Requete()
End Sub

Sub DropTableLiaisonArticleModele()
    sql = "DROP TABLE LiaisonArticleModele"
    NomTable = "LiaisonArticleModele"
    Requete()

End Sub

Sub CreationTableFournisseur()
    sql = "CREATE TABLE Fournisseur (NoFournisseur INT PRIMARY KEY NOT NULL,
NomFournisseur VARCHAR(20), RechercheNomFournisseur VARCHAR(10), NoAdresseFournisseur
INT, Description VARCHAR(60), NoAdresseFacturation INT)"
    NomTable = "Fournisseur"
    Requete()
End Sub

Sub DropTableFournisseur()
    sql = "DROP TABLE Fournisseur"
    NomTable = "Fournisseur"
    Requete()
End Sub

Sub CreationTableAtelier()
    sql = "CREATE TABLE Atelier (NoAtelier INT PRIMARY KEY NOT NULL, NomAtelier
VARCHAR(20), RechercheNomAtelier VARCHAR(10), Description VARCHAR(60), NoDisponibilite
INT, TypeAtelier INT, Contact VARCHAR(30))"
    NomTable = "Atelier"
    Requete()
End Sub

Sub DropTableAtelier()
    sql = "DROP TABLE Atelier"
    NomTable = "Atelier"

```

```

    Requete()
End Sub
Sub CreationTablePoste()
    sql = "CREATE TABLE Poste (NoPoste INT PRIMARY KEY NOT NULL, NomPoste
VARCHAR(20), RechercheNomPoste VARCHAR(10), Description VARCHAR(60), NoDisponibilite
INT, NoTypePoste INT, NoType INT, NoMachine1 INT, NoMachine2 INT, AdresseIP
VARCHAR(20), Superviseur INT)"
    NomTable = "Poste"
    Requete()
End Sub

Sub DropTablePoste()
    sql = "DROP TABLE Poste"
    NomTable = "Poste"
    Requete()
End Sub
Sub CreationTableTypePoste()
    sql = "CREATE TABLE TypePoste (id INT PRIMARY KEY NOT NULL, NomTypePoste
VARCHAR(20), NoTypePoste INT, RechercheNomTypePoste VARCHAR(10),Description
VARCHAR(60), NoDisponibilite INT)"
    NomTable = "TypePoste"
    Requete()
End Sub

Sub DropTableTypePoste()
    sql = "DROP TABLE TypePoste"
    NomTable = "TypePoste"
    Requete()
End Sub
Sub CreationTableCommandeFournisseur()
    sql = "CREATE TABLE CommandeFournisseur (NoCommandeFournisseur INT PRIMARY KEY
NOT NULL, NoFournisseur INT, DateLivraison DATE, HeureLivraison TIME, DateCommande
DATE, NoPosteLivraison INT, EtatCommande INT)"
    NomTable = "CommandeFournisseur"
    Requete()
End Sub

Sub DropTableCommandeFournisseur()
    sql = "DROP TABLE CommandeFournisseur"
    NomTable = "CommandeFournisseur"
    Requete()
End Sub

Sub CreationTableLiaisonCommandeFournisseurDetail()
    sql = "CREATE TABLE LiaisonCommandeFournisseurDetail
(NoLiaisonCommandeFournisseurDetail INT NOT NULL IDENTITY(1, 1) PRIMARY KEY,
NoCommandeFournisseur INT, NoDetail INT)"
    NomTable = "LiaisonCommandeFournisseurDetail"
    Requete()
End Sub

Sub DropTableLiaisonCommandeFournisseurDetail()
    sql = "DROP TABLE LiaisonCommandeFournisseurDetail"
    NomTable = "LiaisonCommandeFournisseurDetail"
    Requete()

End Sub

Sub CreationTableEntrees()
    sql = "CREATE TABLE Entrees (NoCommandeFournisseur INT PRIMARY KEY NOT NULL,
NoFournisseur INT, DateLivraison DATE, HeureLivraison TIME, DateCommande DATE,
NoPosteLivraison INT, EtatCommande INT)"

```

```

        NomTable = "Entrees"
        Requete()
    End Sub

Sub DropTableEntrees()
    sql = "DROP TABLE Entrees"
    NomTable = "Entrees"
    Requete()
End Sub

Sub CreationTableLiaisonEntreeDetail()
    sql = "CREATE TABLE LiaisonEntreesDetail (NoLiaisonEntreesDetail INT NOT NULL
IDENTITY(1, 1) PRIMARY KEY, NoCommandeFournisseur INT, NoDetail INT)"
    NomTable = "LiaisonEntreesDetail"
    Requete()
End Sub

Sub DropTableLiaisonEntreeDetail()
    sql = "DROP TABLE LiaisonEntreesDetail"
    NomTable = "LiaisonEntreesDetail"
    Requete()

End Sub

Sub CreationTableCommandeClient()
    sql = "CREATE TABLE CommandeClient (NoCommandeClient INT PRIMARY KEY NOT NULL,
NoClient INT, DateCommande DATE, DateLivraison DATE, HeureLivraison TIME,
NoPosteExpedition INT, EtatCommande INT)"
    NomTable = "CommandeClient"
    Requete()
End Sub

Sub DropTableCommandeClient()
    sql = "DROP TABLE CommandeClient"
    NomTable = "CommandeClient"
    Requete()
End Sub

Sub CreationTableLiaisonCommandeClientDetail()
    sql = "CREATE TABLE LiaisonCommandeClientDetail (NoLiaisonCommandeClientDetail
INT NOT NULL IDENTITY(1, 1) PRIMARY KEY, NoCommandeClient INT, NoDetail INT)"
    NomTable = "LiaisonCommandeClientsDetail"
    Requete()
End Sub

Sub DropTableLiaisonCommandeClientDetail()
    sql = "DROP TABLE LiaisonCommandeClientDetail"
    NomTable = "LiaisonCommandeClientDetail"
    Requete()

End Sub

Sub CreationTableLivraison()
    sql = "CREATE TABLE Livraison (NoCommandeClient INT PRIMARY KEY NOT NULL,
NoClient INT, DateCommande DATE, DateLivraison DATE, HeureLivraison TIME,
NoPosteExpedition INT, EtatCommande INT, BonLivraisonEdite INT, FactureEditee INT)"
    NomTable = "Livraison"
    Requete()
End Sub

Sub DropTableLivraison()
    sql = "DROP TABLE Livraison"
    NomTable = "Livraison"

```

```

    Requete()
End Sub

Sub CreationTableLiaisonLivraisonDetail()
    sql = "CREATE TABLE LiaisonLivraisonDetail (NoLiaisonLivraisonDetail INT NOT
NULL IDENTITY(1, 1) PRIMARY KEY, NoCommandeClient INT, NoDetail INT)"
    NomTable = "LiaisonLivraisonDetail"
    Requete()
End Sub

Sub DropTableLiaisonLivraisonDetail()
    sql = "DROP TABLE LiaisonLivraisonDetail"
    NomTable = "LiaisonLivraisonDetail"
    Requete()

End Sub

Sub CreationTableModele()
    sql = "CREATE TABLE Modele (TypeModele INT PRIMARY KEY NOT NULL, NomModele
VARCHAR(20), RechercheNomModele VARCHAR(10), Description VARCHAR(60), NoGamme INT,
RangGamme INT)"
    NomTable = "Modele"
    Requete()
End Sub

Sub DropTableModele()
    sql = "DROP TABLE Modele"
    NomTable = "Modele"
    Requete()
End Sub

Sub CreationTableDisponibilite()
    sql = "CREATE TABLE Disponibilite (NoDisponibilite INT PRIMARY KEY NOT NULL,
NomDisponibilite INT, RechercheNomDisponibilite VARCHAR(10), JourSemaine VARCHAR(15),
HeureDeb TIME, HeureFin TIME, JourFerie DATE, JourEntier DATE, Indisponibilite INT)"
    NomTable = "Disponibilite"
    Requete()
End Sub

Sub DropTableDisponibilite()
    sql = "DROP TABLE Disponibilite"
    NomTable = "Disponibilite"
    Requete()
End Sub

Sub CreationTableArticle()
    sql = "CREATE TABLE Article (NoArticle INT PRIMARY KEY NOT NULL, NomArticle
VARCHAR(20), RechercheNomArticle VARCHAR(10), Description VARCHAR(60), Longueur DOUBLE,
Largeur DOUBLE, Hauteur DOUBLE, Poids DOUBLE, PrixFixe INT, PoidsVariable INT,
CodeBarre INT, CategorieArticle VARCHAR(20), TypeConditionnement INT, NoArticleCondInf
INT, QuantiteCondInf INT, TypeQuantiteCondInf INT, TVA DOUBLE)"
    NomTable = "Article"
    Requete()
End Sub

Sub DropTableArticle()
    sql = "DROP TABLE Article"
    NomTable = "Article"
    Requete()
End Sub

```

```

Sub CreationTableStocks()
    sql = "CREATE TABLE Stocks (Id INT PRIMARY KEY NOT NULL, NoArticle INT,
TypeModele INT, Prix DOUBLE, TypePrix INT, Quantite DOUBLE, TypeQuantite INT,
QuantiteTotale DOUBLE, Variation DOUBLE, DateMiseAJour DATE)"
    NomTable = "Stocks"
    Requete()
End Sub

Sub DropTableStocks()
    sql = "DROP TABLE Stocks"
    NomTable = "Stocks"
    Requete()
End Sub

Sub CreationTableTache()
    sql = "CREATE TABLE Tache (Id INT PRIMARY KEY NOT NULL, NoTache INT, NomTache
VARCHAR(20), RechercheNomTache VARCHAR(10),Description VARCHAR(60), TempsExec TIME,
NoGamme INT, NoArticleFini INT, NoTypePoste INT, NoArticlePourLeFaire INT,
TypeModeleFini INT, TypeModelePourLeFaire INT)"
    NomTable = "Tache"
    Requete()
End Sub

Sub DropTableTache()
    sql = "DROP TABLE Tache"
    NomTable = "Tache"
    Requete()
End Sub

Sub CreationTableGamme()
    sql = "CREATE TABLE Gamme (NoGamme INT PRIMARY KEY NOT NULL, RechercheNomGamme
VARCHAR(10), NomGamme VARCHAR(20), Description VARCHAR(60), RangTache INT, NoTache
INT, NoGammeMere INT)"
    NomTable = "Gamme"
    Requete()
End Sub

Sub DropTableGamme()
    sql = "DROP TABLE Gamme"
    NomTable = "Gamme"
    Requete()
End Sub

Sub CreationTableFabrication()
    sql = "CREATE TABLE Fabrication (Id INT PRIMARY KEY NOT NULL, NoFabrication
INT, NomFabrication VARCHAR(20), NoPoste INT, Description VARCHAR(60), DateDebut DATE,
HeureDebut TIME, NoCommande INT, EtatFabrication INT, Urgence INT, NoTache INT,
NoArticle INT, TypeModele INT, Quantite DOUBLE, TypeQuantite INT, DateFinPrevue DATE,
HeureFinPrevue TIME, NoTypePoste INT)"
    NomTable = "Fabrication"
    Requete()
End Sub

Sub DropTableFabrication()
    sql = "DROP TABLE Fabrication"
    NomTable = "Fabrication"
    Requete()
End Sub

Sub CreationTableLiaisonModeleGamme()
    sql = "CREATE TABLE LiaisonModeleGamme (NoLiaisonModeleGamme INT NOT NULL
IDENTITY(1, 1) PRIMARY KEY, NoArticle INT,TypeModele INT, NoGamme INT)"
    NomTable = "LiaisonModeleGamme"
    Requete()
End Sub

Sub DropTableLiaisonModeleGamme()

```



```

    sql = "DROP TABLE LiaisonModeleGamme"
    NomTable = "LiaisonModeleGamme"
    Requete()

End Sub

Sub CreationTableLiaisonGammeTache()
    sql = "CREATE TABLE LiaisonGammeTache (NoLiaisonGammeTache INT NOT NULL
IDENTITY(1, 1) PRIMARY KEY, NoGamme INT,RangTache INT, NoTache INT)"
    NomTable = "LiaisonGammeTache"
    Requete()
End Sub

Sub DropTableLiaisonGammeTache()
    sql = "DROP TABLE LiaisonGammeTache"
    NomTable = "LiaisonGammeTache"
    Requete()

End Sub

Sub CreationTableLiaisonTachePoste()
    sql = "CREATE TABLE LiaisonTachePoste (NoLiaisonTachePoste INT NOT NULL
IDENTITY(1, 1) PRIMARY KEY, NoPoste INT, NoTypePoste INT, TempsExecution TIME,
TypeTemps INT)"
    NomTable = "LiaisonTachePoste"
    Requete()
End Sub

Sub DropTableLiaisonTachePoste()
    sql = "DROP TABLE LiaisonTachePoste"
    NomTable = "LiaisonTachePoste"
    Requete()

End Sub

Sub CreationTableCaracteristiquesTechniques()
    sql = "CREATE TABLE CaracteristiquesTechniques (NoCaracteristiqueTechnique INT
PRIMARY KEY NOT NULL, NomCaracteristiqueTechnique VARCHAR(20),
RechercheCaracteristiqueTechnique VARCHAR(10),ValeurVoulue DOUBLE, TypeValeurVoulue
INT)"
    NomTable = "CaracteristiquesTechniques"
    Requete()
End Sub

Sub DropTableCaracteristiquesTechniques()
    sql = "DROP TABLE CaracteristiquesTechniques"
    NomTable = "CaracteristiquesTechniques"
    Requete()

End Sub

Sub CreationTableMachine()
    sql = "CREATE TABLE Machine (NoMachine INT PRIMARY KEY NOT NULL, NomMachine
VARCHAR(20), RechercheNomMachine VARCHAR(10),NoTypeMachine INT, AdresseIP
VARCHAR(20))"
    NomTable = "Machine"
    Requete()
End Sub

Sub DropTableMachine()
    sql = "DROP TABLE Machine"
    NomTable = "Machine"

```

```
Requete()
```

```
End Sub
```

```
Sub CreationTableLiaisonCommandeCarTech()
```

```
sql = "CREATE TABLE LiaisonCommandeCarTech (NoLiaisonCommandeCarTech INT NOT  
NULL IDENTITY(1, 1) PRIMARY KEY, NoArticle INT, TypeModele INT, NoCommande INT, Valeur  
DOUBLE, NoCaracteristiqueTechnique INT)"
```

```
NomTable = "Liaison CommandeCarTech"
```

```
Requete()
```

```
End Sub
```

```
Sub DropTableLiaisonCommandeCarTech()
```

```
sql = "DROP TABLE LiaisonCommandeCarTech"
```

```
NomTable = "LiaisonCommandeCarTech"
```

```
Requete()
```

```
End Sub
```

```
Sub CreationTableLiaisonModeleCarTech()
```

```
sql = "CREATE TABLE LiaisonModeleCarTech (NoLiaisonModleCarTech INT NOT NULL  
IDENTITY(1, 1) PRIMARY KEY, NoArticle INT, TypeModele INT, NoCommande INT,  
ValeurVoulue DOUBLE, NoCaracteristiqueTechnique INT)"
```

```
NomTable = "Liaison ModeleCarTech"
```

```
Requete()
```

```
End Sub
```

```
Sub DropTableLiaisonModeleCarTech()
```

```
sql = "DROP TABLE LiaisonModeleCarTech"
```

```
NomTable = "LiaisonModeleCarTech"
```

```
Requete()
```

```
End Sub
```

```
End Module
```

FonctionGlobales

```
Imports System.Runtime.InteropServices.Marshal
Imports System.Net.Sockets
Imports System.Net
```

```
Module OuvertureDB
```

```
End Module
```

```
Module FonctionsGlobales
```

```
Private Declare Function GetPrivateProfileString Lib "kernel32.dll" Alias
"GetPrivateProfileStringA" (ByVal lpApplicationName As String, ByVal lpKeyName As
String, ByVal lpDefault As String, ByVal lpReturnedString As String, ByVal nSize As
Short, ByVal lpFileName As String) As Integer
```

```
Private Declare Function WritePrivateProfileString Lib "kernel32.dll" Alias
"WritePrivateProfileStringA" (ByVal lpApplicationName As String, ByVal lpKeyName As
String, ByVal lpString As String, ByVal lpFileName As String) As Long
```

```
'Lire un fichier INI
```

```
Public Function LireINI(ByVal StCheminInffichierINI As String, ByVal StSection As
String, ByVal StCle As String)
Dim stRetour As String
Dim valRetour As String
```

```
'Declare Function GetUserName Lib "advapi32.dll" Alias "GetUserNameA" (ByVal
lpBuffer As String, ByRef nSize As Integer) As Integer
```

```
' Declare Sub SetData Lib "..\LIB\UnmgdLib.dll" (ByVal x As Short,
<System.Runtime.InteropServices.MarshalAsAttribute(System.Runtime.InteropServices.Unma
nagedType.AsAny)> ByVal o As Object)
```

```
' declare Function GetPrivateProfileString Lib "kernel32" Alias
"GetPrivateProfileStringA" (ByVal lpApplicationName As String, ByVal lpKeyName As Any,
ByVal lpDefault As String, ByVal lpReturnedString As String, ByVal nSize As Short,
ByVal lpFileName As String) As Integer
```

```
' Declare Function WritePrivateProfileString Lib "kernel32" Alias
"WritePrivateProfileStringA" (ByVal lpApplicationName As String, ByVal lpKeyName As
String, ByVal lpString As String, ByVal lpFileName As String) As Long
```

```
Try
```

```
stRetour = StrDup(255, Chr(0))
GetPrivateProfileString(StSection, StCle, "", stRetour, Len(stRetour),
StCheminInffichierINI)
```

```
valRetour = Left(stRetour, stRetour.IndexOf(Chr(0)))
```

```
Return valRetour
```

```
Catch ex As Exception
    MsgBox("Error lire INI" & ex.Message, MsgBoxStyle.Critical)
```

```
End Try
```

```
End Function
```

```
'Ecrire un fichier INI
'Public Function EcrireINI(ByVal StCheminInfFichierINI As String, ByVal StSection
As String, ByVal StCle As String, ByVal StValeur As String) As String
'    EcrireINI = WritePrivateProfileString(StSection, StCle, StValeur,
StCheminInfFichierINI)
'' End Function
```

```
End Module
```

SubINI

Module SubINI

```
'Fonction de lecture du fichier INI

' Lecture du fichier INI

' Sub LectureFichierReference()
'Dim i As Integer
' Dim j As Integer
' Dim k As Integer
'
' Try
'lecture du nombre de sections
' StrNbSection = LireINI(DIR_PARAM & FICHER_REF, "Entete", "NbSection")
' NbSection = CInt(StrNbSection)
' TableauReference.NbGroup = NbSection

'Redimensionnement du tableau
' TableauReference.NbParametre = CStr(LireINI(DIR_PARAM & FICH_REF,
"Entete", "NbParametre"))

'lecture du nombre de clés
' TableauReference.NbReference = CInt(LireIni(DIR_PARAM & FICH_REF,
"Section1", "NbCle"))

' ReDim TableauReference.Reference(0 To TableauReference.NbReference -
1)
' For i = 0 To TableauReference.NbReference - 1
' ReDim TableauReference(i).Param(0 To TableauReference.NbParametre -
1)
' Next i

' End Sub

Sub LectureFichierLangue()
Try

' DIR_PARAM = "C:\Users\d\Documents\WMI\WMI VB\WMI 1\WMI 1\FichiersINI\"
DIR_PARAM = TableauParametres.Parametres(0).Données(0).Adr & "\"
FICH_LANGUE = "FichierLangue.ini"

' Lecture du numéro de langue
NumLangueEnCours = CInt(LireINI(DIR_PARAM & FICH_LANGUE, "Entete",
"NumLang"))
If (NumLangueEnCours < 1) Or (NumLangueEnCours > 4) Then
NumLangueDem = 1
Else
NumLangueDem = NumLangueEnCours
End If

Catch ex As Exception
MsgBox("Erreur lecture fichier" & DIR_PARAM & FICH_LANGUE)
```

```

End Try

End Sub

Sub LectureFichierAffichage()
    Dim i As Integer
    Dim j As Integer
    Dim k As Integer

    Try
        DIR_PARAM = TableauParametres.Parametres(0).Données(0).Adr & "\"

        FICH_AFFICHAGE = "FichierAffichage.ini"
        TableauAffichage.NbGroup = CInt(LireINI(DIR_PARAM & FICH_AFFICHAGE,
"Entete", "NbSection"))
        'Redimensionnement du fichier affichage
        ReDim TableauAffichage.GroupAffichage(TableauAffichage.NbGroup - 1)
        For i = 0 To TableauAffichage.NbGroup - 1
            ReDim TableauAffichage.GroupAffichage(i).NomGroup(NbLangue - 1)
            For k = 0 To NbLangue - 1
                TableauAffichage.GroupAffichage(i).NomGroup(k) = LireINI(DIR_PARAM
& FICH_AFFICHAGE, "Section" & CStr(i + 1), "L" & CStr(k + 1) & "NomSection")

                Next k

                TableauAffichage.GroupAffichage(i).NbCle = CInt(LireINI(DIR_PARAM &
FICH_AFFICHAGE, "Section" & CStr(i + 1), "NbCleSection"))
                ReDim
TableauAffichage.GroupAffichage(i).Donnees(TableauAffichage.GroupAffichage(i).NbCle -
1)

                For j = 0 To TableauAffichage.GroupAffichage(i).NbCle - 1
                    ReDim
TableauAffichage.GroupAffichage(i).Donnees(j).Libelle(NbLangue - 1)
                    For k = 0 To NbLangue - 1
                        TableauAffichage.GroupAffichage(i).Donnees(j).Libelle(k) =
LireINI(DIR_PARAM & FICH_AFFICHAGE, "Section" & CStr(i + 1), "L" & CStr(k + 1) &
"NomCle" & CStr(j + 1))
                    Next k
                Next j

            Next i

        Catch ex As Exception
            MsgBox("erreur de lecture du fichier " & FICH_AFFICHAGE)
        End Try

    End Sub

''' <summary>
''' Read INI file : Settings
''' Fonction de lecture du fichier Parametres.ini
''' </summary>
''' <remarks>Creation d'un tableau permettant de stocker les
informations</remarks>
Sub LectureFichierParametres()

    Dim i As Integer

```

```

Dim j As Integer

Try
DIR_PARAM = "C:\Users\GILET\Documents\WMI 1\WMI 1\FichiersINI\"
FICH_PARAM = "Parametres.ini"

'TableauParametres.NbParametres = 0

'Lecture du nombre de section
StrNbSection = LireINI(DIR_PARAM & FICH_PARAM, "Entete", "NbSection")
NbSection = CInt(StrNbSection)

'Redimensionnement du tableau en fonction du Nb de section
ReDim TableauParametres.Parametres(NbSection - 1)

'Redimensionnement du tableau en fonction du Nb de section
For i = 0 To NbSection - 1
    ReDim TableauParametres.Parametres(i).Données(CInt(LireINI(DIR_PARAM &
FICH_PARAM, "Section" & CStr(i + 1), "NbCle"))) - 1)

Next i

'Affectation du nombre de groupe
TableauParametres.NbGroup = NbSection
TableauParametres.NbParametres = 0

'Lecture des valeurs du fichier parametre
For i = 0 To TableauParametres.NbGroup - 1
    'On redimensionne le tableau en fonction du nombre de langue
    ReDim TableauParametres.Parametres(i).NomGroupParametreL(NbLangue - 1)
    For k = 0 To NbLangue - 1
        'On récupère les libellés des sections dans les différentes
langues
            TableauParametres.Parametres(i).NomGroupParametreL(k) =
LireINI(DIR_PARAM & FICH_PARAM, "Section" & CStr(i + 1), "L" & k + 1 & "NomSection")
            Next k
            'On récupère le type de la section : sécurisée par mot de passe ou non
            TableauParametres.Parametres(i).Securise = LireINI(DIR_PARAM &
FICH_PARAM, "Section" & CStr(i + 1), "TypeSection")
            'On récupère le nombre de parametres de la section
            TableauParametres.Parametres(i).NbParametreGroup = LireINI(DIR_PARAM &
FICH_PARAM, "Section" & CStr(i + 1), "NbCle")
            'On incrémente le nombre de parametres total de la cellule
            'TableauParametres.NbParametres = TableauParametres.NbParametres +
TableauParametres.Parametres(i).NbParametreGroup

            'On balaie tous les parametres de la section
            For j = 0 To TableauParametres.Parametres(i).NbParametreGroup - 1
                'On récupère l'adresse du parametre
                TableauParametres.Parametres(i).Données(j).Adr = LireINI(DIR_PARAM
& FICH_PARAM, "Section" & CStr(i + 1), "AdrCle" & CStr(j + 1))
                'On redimensionne le tableau en fonction du nombre de langue
                ReDim TableauParametres.Parametres(i).Données(j).LibelleL(NbLangue
- 1)

                For k = 0 To NbLangue - 1
                    'On récupère le libellé du parametre dans les différentes
langues
                        TableauParametres.Parametres(i).Données(j).LibelleL(k) =
LireINI(DIR_PARAM & FICH_PARAM, "Section" & CStr(i + 1), "L" & k + 1 & "NomCle" &
CStr(j + 1))
                    Next k
                
```

```

        ' If (TableauParametres.NbParametres <
TableauParametres.Parametres(i).Données(j).Adr) Then
        'TableauParametres.NbParametres =
TableauParametres.Parametres(i).Données(j).Adr
        ' End If

    Next j
Next i

' TableauParametres.NbParametres = TableauParametres.NbParametres + 1
' Lecture Adresse du DB des parametres
' AdresseDB = LireINI(DIR_PARAM & FIC_PARAM, "Entete", "AdresseDB")
' DebutDB = LireINI(DIR_PARAM & FIC_PARAM, "Entete", "DebutDB")

'DBParametreApplicom = Val(AdresseDB)
' OperandeParametreApplicom = Val(DebutDB)
' NbParametreApplicom = TableauParametres.NbParametres
' KEP
' AdrParametreOPC = S7 & AdresseDB & "," & DebutDB & "[" &
(CStr(TableauParametres.NbParametres)) & "]"
' S7
' AdrParametreOPC = S7 & AdresseDB & "," & DebutDB & "," &
(CStr(TableauParametres.NbParametres))

```

```

Catch ex As Exception
    MsgBox("Erreur lecture fichier " & FICH_PARAM, vbOKOnly, "IHM")
End Try

```

End Sub

Sub LectureFichierOperateurs()

```
Dim i As Integer
```

```
Dim j As Integer
```

```
Try
```

```
DIR_PARAM = TableauParametres.Parametres(0).Données(0).Adr & "\"
FICH_PARAM = "FichierOperateurs.ini"
```

```
'TableauParametres.NbParametres = 0
```

```
'Lecture du nombre de section
```

```
StrNbSection = LireINI(DIR_PARAM & FICH_PARAM, "Entete", "NbSection")
```

```
NbSection = CInt(StrNbSection)
```

```
'Redimensionnement du tableau en fonction du Nb de section
```

```
ReDim TableauOperateurs.Operateurs(NbSection - 1)
```

```
'Redimensionnement du tableau en fonction du Nb de section
```

```
For i = 0 To NbSection - 1
```

```
    ReDim TableauOperateurs.Operateurs(i).Donnees(CInt(LireINI(DIR_PARAM &
FICH_PARAM, "Section" & CStr(i + 1), "NbCle"))) - 1)
```

```
Next i
```

```
'Affectation du nombre de groupe
```



```

TableauOperateurs.NbGroup = NbSection
TableauOperateurs.NbOperateurs = 0

' Lecture des valeurs du fichier parametre
For i = 0 To TableauOperateurs.NbGroup - 1
    ' On redimensionne le tableau en fonction du nombre de langue
    ReDim TableauOperateurs.Operateurs(i).NomGroupOperateursL(NbLangue -
1)

    For k = 0 To NbLangue - 1
        ' On récupère les libellés des sections dans les différentes
langues
        TableauOperateurs.Operateurs(i).NomGroupOperateursL(k) =
LireINI(DIR_PARAM & FICH_PARAM, "Section" & CStr(i + 1), "L" & CStr(k + 1) &
"NomSection")
    Next k
    ' On récupère le type de la section : sécurisée par mot de passe ou non
    TableauOperateurs.Operateurs(i).CodeSecret = LireINI(DIR_PARAM &
FICH_PARAM, "Section" & CStr(i + 1), "TypeSection")
    ' On récupère le nombre de parametres de la section
    TableauOperateurs.Operateurs(i).NbOperateursGroup = LireINI(DIR_PARAM
& FICH_PARAM, "Section" & CStr(i + 1), "NbCle")
    ' On incrémente le nombre de parametres total de la cellule
    ' TableauParametres.NbParametres = TableauParametres.NbParametres +
TableauParametres.Parametres(i).NbParametreGroup

    ' On balaie tous les parametres de la section
    For j = 0 To TableauOperateurs.Operateurs(i).NbOperateursGroup - 1
        ' On récupère l'adresse du parametre
        TableauOperateurs.Operateurs(i).Donnees(j).Adr = LireINI(DIR_PARAM
& FICH_PARAM, "Section" & CStr(i + 1), "AdrCle" & CStr(j + 1))
        ' On redimensionne le tableau en fonction du nombre de langue
        ReDim TableauOperateurs.Operateurs(i).Donnees(j).Param(NbLangue -
1)

        For k = 0 To NbLangue - 1
            ' On récupère le libellé du parametre dans les différentes
langues
            TableauOperateurs.Operateurs(i).Donnees(j).Param(k) =
LireINI(DIR_PARAM & FICH_PARAM, "Section" & CStr(i + 1), "L" & k + 1 & "NomCle" &
CStr(j + 1))
        Next k
        ' If (TableauParametres.NbParametres <
TableauParametres.Parametres(i).Données(j).Adr) Then
            ' TableauParametres.NbParametres =
TableauParametres.Parametres(i).Données(j).Adr
        ' End If

    Next j
Next i

' TableauParametres.NbParametres = TableauParametres.NbParametres + 1
' Lecture Adresse du DB des parametres
' AdresseDB = LireINI(DIR_PARAM & FICH_PARAM, "Entete", "AdresseDB")
' DebutDB = LireINI(DIR_PARAM & FICH_PARAM, "Entete", "DebutDB")

' DBParametreApplicom = Val(AdresseDB)
' OperandeParametreApplicom = Val(DebutDB)
' NbParametreApplicom = TableauParametres.NbParametres
' KEP
' AdrParametreOPC = S7 & AdresseDB & "," & DebutDB & "[" &
(CStr(TableauParametres.NbParametres)) & "]"

```

```
'S7
'AdrParametreOPC = S7 & AdresseDB & "," & DebutDB & "," &
(CStr(TableauParametres.NbParametres))

Catch ex As Exception
MsgBox("Erreur lecture fichier " & FICH_PARAM, vbOKOnly, "IHM")
End Try

End Sub
End Module
```

VariablesGlobales

Module VariablesGlobales

'déclaration des variables globales

```
Public NumLangueEnCours As Integer
Public NumLangueDem As Integer
Public DIR_PARAM As String
Public FICH_LANGUE As String
Public FICH_AFFICHAGE As String
Public FICH_PARAM As String
Public Const Nblangue = 4
Public PassageCleForm As String
Public variable As Integer

Public Utilisateur As String
Public NoUtilisateur As Int16

Public LangueUtilisateur As Int16
Public SaisieAutorisee As Int16
Public GestionAutorisee As Int16
Public EditionsAurisees As Int16
Public StocksAutorises As Int16
Public SaisieEntreesAutorisee As Int16
Public SuperviseurReceptionAutorise As Int16
Public ConnectionEDIAutorisee As Int16
Public SuperviseurExpeditionsAutorise As Int16
Public superviseurProductionAutorise As Int16
Public VisuStatistiquesAutorise As Int16
```

End Module

VariablesINI

```
Module FichierINI
'Déclaration variables pour fichier ini
Public StrNbSection As String
Public NbSection As Integer

Public StrNbCle As String
Public NbCle As Integer

Public AdresseDB As String
Public DebutDB As String
Public NbWordDB As String

'Declaration du tableau Parametre
Structure TypParam
    Public LibelleL() As String
    Public Adr As String
End Structure

Structure TypGroupParametre
    Public NomGroupParametreL() As String
    Public NbParametreGroup As Integer
    Public Securise As Boolean
    Public Données() As TypParam
End Structure

Structure TypTableauParametre
    Public NbGroup As Integer
    Public NbParametres As Integer
    Public Parametres() As TypGroupParametre
End Structure

Public TableauParametres As TypTableauParametre

Structure TypOperateur
    Public Param() As String
    Public Adr As String
End Structure

Structure TypGroupOperateur
    Public NomGroupOperateursL() As String
    Public NbOperateursGroup As Integer
    Public CodeSecret As String
    Public Donnees() As TypOperateur
End Structure

Structure TypTableauOperateur
    Public NbGroup As Integer
    Public NbOperateurs As Integer
    Public Operateurs() As TypGroupOperateur
End Structure

Public TableauOperateurs As TypTableauOperateur

'Déclaration du tableau d'affichage
Structure TypAffichage
```

```

    Public Libelle() As String
End Structure

Structure TypGroupAffichage
    Public NomGroup() As String
    Public NbCle As Integer
    Public Donnees() As TypAffichage
End Structure

Structure TypTableauAffichage
    Public NbGroup As Integer
    Public GroupAffichage() As TypGroupAffichage
End Structure

Structure TypGroupRerefence
    Public LblRef As String
    Public Param As String

End Structure

Structure TypeReference
    Public Nbgroup As Integer
    Public NbReference As Integer
    Public NbParametre As Integer
    Public Reference As TypGroupRerefence
End Structure

' Structure TypeAffichage
'Public LibelleL() As String
'End Structure
' Structure TyepGroupAffichage
'Public NomGroupL() As String
'Public NbCle As Integer
'Public Donnees As TypeAffichage
' End Structure
' Structure typeTableazuAffichage
'Public NbGroup As Integer
' Public GroupeAffichage As TypGroupAffichage
'End Structure
Structure TypLiaisonEntrepriseSite
    Public NoLiaisonEntrepriseSite As String
    Public NoSite As String
    Public NoEntreprise As String
    Public NbSite As Integer
    Public NomSite As String

End Structure
Structure TypLiaisonSiteAtelier
    Public NoLiaisonSiteAtelier As String
    Public NoAtelier As String
    Public NoSite As String
    Public NbAtelier As Integer
    Public NomAtelier As String

End Structure

Structure TypEntreprise
    Public NoEntreprise As String
    Public NomEntreprise As String
    Public NoSIRET As String

```

```

Public Adr1 As String
Public adr2 As String
Public CP As String
Public Ville As String
Public Pays As String
Public Description As String
Public NbVariable As Int16
Public NbLiaison As Int16
Public Email As String
Public NoTel As String
Public NoFax As String
Public CheminLogo As String
Public SiteWeb As String
Public RechercheNomEntreprise As String
Public MentionLegale As String
Public LiaisonEntrepriseSite() As TypLiaisonEntrepriseSite
End Structure
Structure TypAdresse
Public NoAdresse As Int16
Public Adr1 As String
Public Adr2 As String
Public CP As String
Public Ville As String
Public Pays As String
Public Email As String
Public NoTel As String
Public NoFax As String

' Public CheminLogo As String
Public SiteWeb As String
Public Contact As String
End Structure

Structure TypSite
Public NoSite As String
Public NomSite As String
Public NbVariable As Int16
Public NbLiaison As Int16
Public Description As String
Public RechercheNomSite As String
Public NoDisponibilite As Int16

Public LiaisonSiteAdresse() As TypAdresse
' Public NoAdresse() As Int16

Public LiaisonSiteAtelier() As TypLiaisonSiteAtelier

End Structure

Public TableauAffichage As TypTableauAffichage
Public TableauMessageBox As TypTableauAffichage
Public TableauReference As TypGroupRerefence

'Declaration des structures base de donnée
Public TableauEntreprise As TypEntreprise
' Public TableauLiaisonEntrepriseSite As TypLiaisonEntrepriseSite
Public TableauListeSite As TypEntreprise
Public TableauSite As TypSite
Public TableauListeAtelier As TypSite

```

End Module

RequetesExploitation

Module RequetesExploitation

```
Sub SelectEntreprise1(ByVal TypeSQL, ByVal ValeurSQL)
    Dim i As Integer

    sql = "SELECT * FROM Entreprise WHERE " & TypeSQL & " = " & ValeurSQL
    NomTable = "Entreprise"

    Requete()
    TableauEntreprise.NbVariable = dtt.Rows.Count
    For i = 0 To (TableauEntreprise.NbVariable - 1)
        TableauEntreprise.NoEntreprise = dtt.Rows(i).Item("NoEntreprise")
        TableauEntreprise.NomEntreprise = dtt.Rows(i).Item("NomEntreprise")
        TableauEntreprise.NoSIRET = dtt.Rows(i).Item("NoSIRET")
        TableauEntreprise.Adr1 = dtt.Rows(i).Item("Adr1")
        TableauEntreprise.adr2 = dtt.Rows(i).Item("Adr2")
        TableauEntreprise.CP = dtt.Rows(i).Item("CP")
        TableauEntreprise.Ville = dtt.Rows(i).Item("Ville")
        TableauEntreprise.Pays = dtt.Rows(i).Item("Pays")
        TableauEntreprise.Description = dtt.Rows(i).Item("Description")
        TableauEntreprise.Email = dtt.Rows(i).Item("Email")
        TableauEntreprise.NoTel = dtt.Rows(i).Item("NoTel")
        TableauEntreprise.NoFax = dtt.Rows(i).Item("NoFax")
        TableauEntreprise.SiteWeb = dtt.Rows(i).Item("SiteWeb")
        TableauEntreprise.MentionLegale = dtt.Rows(i).Item("MentionLegale")
        TableauEntreprise.RechercheNomEntreprise =
dtt.Rows(i).Item("RechercheNomEntreprise")
        ' TableauEntreprise.CheminLogo = dtt.Rows(i).Item("CheminLogo")

        ' TableauEntreprise.NoLiaisonEntrepriseSite =
dtt.Rows(i).Item("NoLiaisonEntrepriseSite")

    Next i
End Sub
Sub SelectEntreprise2(ByVal TypeSQL, ByVal ValeurSQL)
    Dim i As Integer

    sql = "SELECT * FROM LiaisonEntrepriseSite WHERE " & TypeSQL & " = " &
ValeurSQL
    NomTable = "Entreprise"

    Requete()
    ReDim TableauEntreprise.LiaisonEntrepriseSite(dtt.Rows.Count - 1)
    TableauEntreprise.NbLiaison = dtt.Rows.Count
    For i = 0 To (dtt.Rows.Count - 1)

        TableauEntreprise.LiaisonEntrepriseSite(i).NoEntreprise =
dtt.Rows(i).Item("NoEntreprise")

        TableauEntreprise.LiaisonEntrepriseSite(i).NoLiaisonEntrepriseSite =
dtt.Rows(i).Item("NoLiaisonEntrepriseSite")
        TableauEntreprise.LiaisonEntrepriseSite(i).NoSite =
dtt.Rows(i).Item("NoSite")
    Next i
End Sub
```



```

    Next i
End Sub

Sub SelectSite()
    Dim i As Integer

    sql = "SELECT NoSite, NomSite FROM Site"
    NomTable = "Site"

    Requete()

    ReDim TableauListeSite.LiaisonEntrepriseSite(dtt.Rows.Count - 1)
    TableauListeSite.NbLiaison = dtt.Rows.Count
    For i = 0 To (dtt.Rows.Count - 1)

        TableauListeSite.LiaisonEntrepriseSite(i).NoSite =
dtt.Rows(i).Item("NoSite")

        TableauListeSite.LiaisonEntrepriseSite(i).NomSite =
dtt.Rows(i).Item("NomSite")

    Next i
End Sub

Sub SelectSite1(ByVal TypeSQL, ByVal ValeurSQL)
    Dim i As Integer
    Dim j As Integer
    ReDim TableauSite.LiaisonSiteAdresse(2)

    For j = 0 To 2
        TableauSite.LiaisonSiteAdresse(j).NoAdresse = 0
    Next
    sql = "SELECT * FROM Site WHERE " & TypeSQL & " = " & ValeurSQL
    NomTable = "Site"

    Requete()

    TableauSite.NbVariable = dtt.Rows.Count
    For i = 0 To (TableauSite.NbVariable - 1)
        TableauSite.NoSite = dtt.Rows(i).Item("NoSite")
        TableauSite.NomSite = dtt.Rows(i).Item("NomSite")
        TableauSite.RechercheNomSite = dtt.Rows(i).Item("RechercheNomSite")
        TableauSite.LiaisonSiteAdresse(0).NoAdresse =
CInt(dtt.Rows(i).Item("NoAdresseSite"))
        TableauSite.LiaisonSiteAdresse(1).NoAdresse =
CInt(dtt.Rows(i).Item("NoAdresseFacturation"))
        TableauSite.LiaisonSiteAdresse(2).NoAdresse =
CInt(dtt.Rows(i).Item("NoAdresseLivraison"))

        TableauSite.Description = dtt.Rows(i).Item("Description")
        TableauSite.NoDisponibilite = dtt.Rows(i).Item("NoDisponibilite")

    Next i
    For j = 0 To 2
        If TableauSite.LiaisonSiteAdresse(j).NoAdresse <> 0 Then
            sql = "SELECT * FROM Adresse WHERE NoAdresse = " &
TableauSite.LiaisonSiteAdresse(j).NoAdresse

```

```

        NomTable = "Site"

        Requete()

        For i = 0 To (dtt.Rows.Count - 1)
            TableauSite.LiaisonSiteAdresse(j).NoAdresse =
CInt(dtt.Rows(i).Item("NoAdresse"))
            TableauSite.LiaisonSiteAdresse(j).Adr1 = dtt.Rows(i).Item("Adr1")
            TableauSite.LiaisonSiteAdresse(j).Adr2 = dtt.Rows(i).Item("Adr2")
            TableauSite.LiaisonSiteAdresse(j).CP =
CInt(dtt.Rows(i).Item("CP"))
            TableauSite.LiaisonSiteAdresse(j).Ville =
dtt.Rows(i).Item("Ville")
            TableauSite.LiaisonSiteAdresse(j).Pays = dtt.Rows(i).Item("Pays")
            TableauSite.LiaisonSiteAdresse(j).Email =
dtt.Rows(i).Item("Email")
            TableauSite.LiaisonSiteAdresse(j).NoTel =
dtt.Rows(i).Item("NoTel")
            TableauSite.LiaisonSiteAdresse(j).NoFax =
dtt.Rows(i).Item("NoFax")
            TableauSite.LiaisonSiteAdresse(j).SiteWeb =
dtt.Rows(i).Item("SiteWeb")
            TableauSite.LiaisonSiteAdresse(j).SiteWeb =
dtt.Rows(i).Item("SiteWeb")
            TableauSite.LiaisonSiteAdresse(j).Contact =
dtt.Rows(i).Item("Contact")

            Next i

        End If

    Next j
End Sub
Sub SelectSite2(ByVal TypeSQL, ByVal ValeurSQL)
    Dim i As Integer

    sql = "SELECT * FROM LiaisonSiteAtelier WHERE " & TypeSQL & " = " & ValeurSQL
    NomTable = "LiaisonSiteAtelier"

    Requete()
    ReDim TableauSite.LiaisonSiteAtelier(dtt.Rows.Count - 1)
    TableauSite.NbLiaison = dtt.Rows.Count
    For i = 0 To (dtt.Rows.Count - 1)

        TableauSite.LiaisonSiteAtelier(i).NoSite = dtt.Rows(i).Item("NoSite")

        TableauSite.LiaisonSiteAtelier(i).NoAtelier =
dtt.Rows(i).Item("NoAtelier")

        Next i
    End Sub

Sub SelectAtelier()
    Dim i As Integer

    sql = "SELECT NoAtelier, NomAtelier FROM Atelier"
    NomTable = "Atelier"
    Requete()

```

```
ReDim TableauListeAtelier.LiaisonSiteAtelier(dtt.Rows.Count - 1)
TableauListeAtelier.NbLiaison = dtt.Rows.Count
For i = 0 To (dtt.Rows.Count - 1)

    TableauListeAtelier.LiaisonSiteAtelier(i).NoAtelier =
dtt.Rows(i).Item("NoAtelier")

    TableauListeAtelier.LiaisonSiteAtelier(i).NomAtelier =
dtt.Rows(i).Item("NomAtelier")

Next i

End Sub

End Module
```

MenuGeneral

```
Public Class Menu
```

```
    Private Sub MenuStrip1_Load(sender As System.Object, e As System.EventArgs)  
Handles MyBase.Load
```

```
    LectureFichierParametres()  
    ConnexionDB()  
    LectureFichierLangue()  
    LectureFichierAffichage()  
    LectureFichierOperateurs()
```

```
    InitParametresUsers()
```

```
    GestionLangueMenu()
```

```
End Sub
```

```
    Private Sub EntrepriseToolStripMenuItem_Click(sender As Object, e As EventArgs)  
Handles EntrepriseToolStripMenuItem.Click
```

```
    SaisieEntreprise.Show()
```

```
End Sub
```

```
    Private Sub SiteToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles  
SiteToolStripMenuItem.Click
```

```
    SaisieSites.Show()
```

```
End Sub
```

```
    Private Sub AtelierToolStripMenuItem_Click(sender As Object, e As EventArgs)  
Handles AtelierToolStripMenuItem.Click
```

```
    SaisieAtelier.Show()
```

```
End Sub
```

```
    Private Sub ArticleToolStripMenuItem_Click(sender As Object, e As EventArgs)  
Handles ArticleToolStripMenuItem.Click
```

```
    SaisieArticle.Show()
```

```
End Sub
```

```
    Private Sub DescriptifModèleToolStripMenuItem_Click(sender As Object, e As  
EventArgs)
```

```
End Sub
```

```
    Private Sub DescriptifTâcheToolStripMenuItem_Click(sender As Object, e As  
EventArgs) Handles DescriptifTâcheToolStripMenuItem.Click
```

```
    SaisieTache.Show()
```

```
End Sub
```

```
    Private Sub DescriptifGammeToolStripMenuItem_Click(sender As Object, e As  
EventArgs) Handles DescriptifGammeToolStripMenuItem.Click
```

```
    SaisieDescriptifGamme.Show()
```

```
End Sub
```

```

Private Sub ApprovisionnementsToolStripMenuItem_Click(sender As Object, e As
EventArgs) Handles ApprovisionnementsToolStripMenuItem.Click

End Sub

Private Sub SaisieDesCommandesToolStripMenuItem_Click(sender As Object, e As
EventArgs) Handles SaisieDesCommandesToolStripMenuItem.Click
    SaisieCommandeClient.Show()
End Sub

Private Sub ToolStripMenuItem2_Click(sender As Object, e As EventArgs) Handles
ToolStripMenuItem2.Click
    SaisieFournisseur.Show()
End Sub

Private Sub ToolStripMenuItem1_Click(sender As Object, e As EventArgs) Handles
ToolStripMenuItem1.Click
    SaisieClient.Show()
End Sub

Private Sub SaisieDesEntréesToolStripMenuItem_Click(sender As Object, e As
EventArgs) Handles SaisieDesEntréesToolStripMenuItem.Click
    Saisiedesentrees.Show()
End Sub

Private Sub StocksToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles StocksToolStripMenuItem.Click
    GestiondesStocks.Show()
End Sub

Private Sub SaisieDuneProductionToolStripMenuItem_Click(sender As Object, e As
EventArgs) Handles SaisieDuneProductionToolStripMenuItem.Click
    SaisieduneFabrication.Show()
End Sub

Private Sub PosteToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
PosteToolStripMenuItem.Click
    SaisiePoste.Show()
End Sub

Private Sub ToolStripMenuItem3_Click(sender As Object, e As EventArgs) Handles
ToolStripMenuItem3.Click
    SaisieCommandeFournisseur.Show()

End Sub

Private Sub SuperviseurDapprovisionnementToolStripMenuItem_Click(sender As Object,
e As EventArgs)

End Sub

Private Sub ToolStripMenuItem4_Click(sender As Object, e As EventArgs) Handles
ToolStripMenuItem4.Click
    Saisiedunelivraison.Show()
End Sub

Private Sub SuperviseurDeProductionToolStripMenuItem_Click(sender As Object, e As
EventArgs)

```

```

End Sub

Private Sub SuperviseurDexpeditionToolStripMenuItem_Click(sender As Object, e As
EventArgs)

End Sub

Private Sub ParametrageBaseDeDonneeToolStripMenuItem_Click(sender As
System.Object, e As System.EventArgs)
    ParametrageDB.Show()
End Sub

Private Sub DeconnexionToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs)
    Utilisateur = ""
    InitParametresUsers()

End Sub

Private Sub SaisieMachineToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs)
    SaisieMachines.Show()
End Sub

Private Sub ToolStripMenuItem7_Click(sender As System.Object, e As
System.EventArgs) Handles ToolStripMenuItem7.Click
    SaisieTypePoste.Show()

End Sub

Private Sub ConnexionToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles ConnexionToolStripMenuItem.Click
    SaisieCodeUtilisateur.Show()
End Sub

Private Sub QuitterToolStripMenuItem1_Click(sender As System.Object, e As
System.EventArgs) Handles QuitterToolStripMenuItem1.Click
    cnx = Nothing
    dts = Nothing

    Me.Close()
End Sub

Private Sub SaisieDesDonneesToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles SaisieDesDonneesToolStripMenuItem.Click
    SaisieDesDonneesToolStripMenuItem.Enabled = False
End Sub

Private Sub Timer1_Tick(sender As System.Object, e As System.EventArgs) Handles
Timer1.Tick

    EnableUsers()
    GestionLangueMenu()

End Sub

Private Sub StatistiquesToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles StatistiquesToolStripMenuItem.Click

```

```

End Sub

Private Sub ApprovisionnementsToolStripMenuItem1_Click(sender As System.Object, e
As System.EventArgs) Handles ApprovisionnementsToolStripMenuItem1.Click

End Sub

Private Sub SuperToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles SuperToolStripMenuItem.Click

End Sub

Private Sub ToolStripMenuItem5_Click(sender As System.Object, e As
System.EventArgs) Handles ToolStripMenuItem5.Click

End Sub

Private Sub CommandesToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles CommandesToolStripMenuItem.Click

End Sub

Private Sub GestionDeProductionToolStripMenuItem_Click(sender As System.Object, e
As System.EventArgs) Handles GestionDeProductionToolStripMenuItem.Click

End Sub

Private Sub SuperviseurDexpéditionsToolStripMenuItem_Click(sender As
System.Object, e As System.EventArgs) Handles
SuperviseurDexpéditionsToolStripMenuItem.Click

End Sub

Private Sub SuperviseurDeProductionToolStripMenuItem1_Click(sender As
System.Object, e As System.EventArgs) Handles
SuperviseurDeProductionToolStripMenuItem1.Click
    AccueilSuperviseur.Show()
End Sub

Private Sub _Click(sender As System.Object, e As System.EventArgs) Handles
ConnectionEDIToolStripMenuItem.Click

End Sub
Sub InitParametresUsers()
    SaisieDesDonnéesToolStripMenuItem.Enabled = False
    ToolStripMenuItem5.Enabled = True
    GestionToolStripMenuItem.Enabled = False
    EtatsEtÉditionsToolStripMenuItem.Enabled = False
    ToolStripMenuItem5.Enabled = False
End Sub

Private Sub FichierToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles FichierToolStripMenuItem.Click
    InitParametresUsers()
End Sub
Sub EnableUsers()
    If StocksAutorises = 0 Then

        StocksToolStripMenuItem1.Enabled = False
    Else
        GestionToolStripMenuItem.Enabled = True

```

```

    CommandesToolStripMenuItem.Enabled = False
    ApprovisionnementsToolStripMenuItem.Enabled = False
    GestionDeProductionToolStripMenuItem.Enabled = False
    StocksToolStripMenuItem1.Enabled = True
End If
'
' SaisieEntreesAutorisee = 0
'   ToolStripMenuItem5.Enabled = True
If SuperviseurReceptionAutorise = 1 Then
    ToolStripMenuItem5.Enabled = True
    SuperToolStripMenuItem.Enabled = True
End If

If SuperviseurExpeditionsAutorise = 1 Then
    ToolStripMenuItem5.Enabled = True
    SuperviseurDexpéditionsToolStripMenuItem.Enabled = True
End If

If superviseurProductionAutorise = 1 Then
    ToolStripMenuItem5.Enabled = True
    SuperviseurDeProductionToolStripMenuItem1.Enabled = True
End If

'   EtatsEtÉditionsToolStripMenuItem.Enabled = True
'   StocksToolStripMenuItem.Enabled = False
If SaisieAutorisee = 0 Then
    SaisieDesDonnéesToolStripMenuItem.Enabled = False
Else
    SaisieDesDonnéesToolStripMenuItem.Enabled = True
End If
If GestionAutorisee = 0 Then
    GestionToolStripMenuItem.Enabled = False
Else
    GestionToolStripMenuItem.Enabled = True
    ConnectionEDIToolStripMenuItem.Enabled = False
End If
If EditionsAurisees = 0 Then
    EtatsEtÉditionsToolStripMenuItem.Enabled = False
Else
    EtatsEtÉditionsToolStripMenuItem.Enabled = True
    StatistiquesToolStripMenuItem.Enabled = False
End If

If VisuStatistiquesAutorise = 0 Then
    StatistiquesToolStripMenuItem.Enabled = False
Else
    StatistiquesToolStripMenuItem.Enabled = True
End If
If ConnectionEDIAutorisee = 1 Then
    ConnectionEDIToolStripMenuItem.Enabled = True
End If
'   ApprovisionnementsToolStripMenuItem1.Enabled = False

'   ProductionToolStripMenuItem.Enabled = False

'   CommandesToolStripMenuItem1.Enabled = False
'   ExpéditionsToolStripMenuItem.Enabled = False
End Sub
Sub GestionLangueMenu()

```



```

MenuStrip1.Text = TableauAffichage.GroupAffichage(0).NomGroup(NumLangueDem -
1)
Me.Text = TableauAffichage.GroupAffichage(1).Donnees(16).Libelle(NumLangueDem
- 1)
FichierToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(0).Libelle(NumLangueDem - 1)
ConnexionToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(1).Libelle(NumLangueDem - 1)
' DroitsToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(2).Libelle(NumLangueDem - 1)
' LangageToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(3).Libelle(NumLangueDem - 1)
' AdministrateurToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(4).Libelle(NumLangueDem - 1)
' UtilisateurToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(5).Libelle(NumLangueDem - 1)
SaisieDesDonneesToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(6).Libelle(NumLangueDem - 1)
EntrepriseToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(7).Libelle(NumLangueDem - 1)
AtelierToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(8).Libelle(NumLangueDem - 1)
PosteToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(9).Libelle(NumLangueDem - 1)
ArticleToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(10).Libelle(NumLangueDem - 1)
ModèleToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(11).Libelle(NumLangueDem - 1)
SiteToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(12).Libelle(NumLangueDem - 1)
GammeToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(13).Libelle(NumLangueDem - 1)
' TypeModèleToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(14).Libelle(NumLangueDem - 1)
' DescriptifModèleToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(15).Libelle(NumLangueDem - 1)
StocksToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(16).Libelle(NumLangueDem - 1)
ApprovisionnementsToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(17).Libelle(NumLangueDem - 1)
GestionDeProductionToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(18).Libelle(NumLangueDem - 1)
CommandesToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(19).Libelle(NumLangueDem - 1)
EtatsEtÉditionsToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(20).Libelle(NumLangueDem - 1)
StocksToolStripMenuItem1.Text =
TableauAffichage.GroupAffichage(0).Donnees(21).Libelle(NumLangueDem - 1)
ApprovisionnementsToolStripMenuItem1.Text =
TableauAffichage.GroupAffichage(0).Donnees(22).Libelle(NumLangueDem - 1)
ProductionToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(23).Libelle(NumLangueDem - 1)
GammeToolStripMenuItem1.Text =
TableauAffichage.GroupAffichage(0).Donnees(24).Libelle(NumLangueDem - 1)
BonDeFabricationToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(25).Libelle(NumLangueDem - 1)
CommandesToolStripMenuItem1.Text =
TableauAffichage.GroupAffichage(0).Donnees(26).Libelle(NumLangueDem - 1)
ExpéditionsToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(27).Libelle(NumLangueDem - 1)

```

```
StatistiquesToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(28).Libelle(NumLangueDem - 1)
'ParamètresToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(29).Libelle(NumLangueDem - 1)
'ParamétrageBaseDeDonnéeToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(30).Libelle(NumLangueDem - 1)
' SaisieTabletteToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(31).Libelle(NumLangueDem - 1)
' SaisieOrdinateurToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(32).Libelle(NumLangueDem - 1)
ToolStripMenuItem2.Text =
TableauAffichage.GroupAffichage(0).Donnees(107).Libelle(NumLangueDem - 1)
ToolStripMenuItem1.Text =
TableauAffichage.GroupAffichage(0).Donnees(108).Libelle(NumLangueDem - 1)
DescriptifTâcheToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(1).Donnees(15).Libelle(NumLangueDem - 1)
DescriptifGammeToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(1).Donnees(6).Libelle(NumLangueDem - 1)
ToolStripMenuItem3.Text =
TableauAffichage.GroupAffichage(1).Donnees(5).Libelle(NumLangueDem - 1)
SaisieDesEntréesToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(1).Donnees(7).Libelle(NumLangueDem - 1)
```

```
SuperToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(109).Libelle(NumLangueDem - 1)
SaisieDesCommandesToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(1).Donnees(4).Libelle(NumLangueDem - 1)
ToolStripMenuItem4.Text =
TableauAffichage.GroupAffichage(0).Donnees(110).Libelle(NumLangueDem - 1)
SaisieDuneProductionToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(1).Donnees(8).Libelle(NumLangueDem - 1)
ToolStripMenuItem4.Text =
TableauAffichage.GroupAffichage(1).Donnees(9).Libelle(NumLangueDem - 1)
SuperviseurDeProductionToolStripMenuItem1.Text =
TableauAffichage.GroupAffichage(0).Donnees(111).Libelle(NumLangueDem - 1)
SuperviseurDexpéditionsToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(112).Libelle(NumLangueDem - 1)
ToolStripMenuItem5.Text =
TableauAffichage.GroupAffichage(0).Donnees(118).Libelle(NumLangueDem - 1)
ConnectionEDIToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(119).Libelle(NumLangueDem - 1)
BonDeTransportToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(120).Libelle(NumLangueDem - 1)
StatistiquesDeDirectionToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(121).Libelle(NumLangueDem - 1)
StatistiquesCommercialesToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(122).Libelle(NumLangueDem - 1)
StatistiquesDeProductionToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(123).Libelle(NumLangueDem - 1)
ChargePostesToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(124).Libelle(NumLangueDem - 1)
RécapitulatifDuneJournéeToolStripMenuItem1.Text =
TableauAffichage.GroupAffichage(0).Donnees(125).Libelle(NumLangueDem - 1)
CarnetDeCommandesToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(126).Libelle(NumLangueDem - 1)
ProduitsVedetteToolStripMenuItem.Text =
TableauAffichage.GroupAffichage(0).Donnees(127).Libelle(NumLangueDem - 1)
ToolStripMenuItem7.Text =
TableauAffichage.GroupAffichage(0).Donnees(135).Libelle(NumLangueDem - 1)
ToolStripMenuItem6.Text =
TableauAffichage.GroupAffichage(0).Donnees(149).Libelle(NumLangueDem - 1)
```

```

        QuitterToolStripMenuItem1.Text =
TableauAffichage.GroupAffichage(0).Donnees(116).Libelle(NumLangueDem - 1)

    End Sub

    Private Sub BonDeTransportToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles BonDeTransportToolStripMenuItem.Click

    End Sub

    Private Sub StatistiquesDeDirectionToolStripMenuItem_Click(sender As
System.Object, e As System.EventArgs) Handles
StatistiquesDeDirectionToolStripMenuItem.Click

    End Sub

    Private Sub StatistiquesCommercialesToolStripMenuItem_Click(sender As
System.Object, e As System.EventArgs) Handles
StatistiquesCommercialesToolStripMenuItem.Click

    End Sub

    Private Sub StatistiquesDeProductionToolStripMenuItem_Click(sender As
System.Object, e As System.EventArgs) Handles
StatistiquesDeProductionToolStripMenuItem.Click

    End Sub

    Private Sub ChargePostesToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles ChargePostesToolStripMenuItem.Click

    End Sub

    Private Sub RécapitulatifDuneJournéeToolStripMenuItem1_Click(sender As
System.Object, e As System.EventArgs) Handles
RécapitulatifDuneJournéeToolStripMenuItem1.Click

    End Sub

    Private Sub CarnetDeCommandesToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles CarnetDeCommandesToolStripMenuItem.Click

    End Sub

    Private Sub ProduitsVedetteToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles ProduitsVedetteToolStripMenuItem.Click

    End Sub

    Private Sub TypeModèleToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs)

    End Sub

    Private Sub ModèleToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles ModèleToolStripMenuItem.Click
        SaisieModele.Show()
    End Sub

    Private Sub ToolStripMenuItem6_Click(sender As System.Object, e As
System.EventArgs) Handles ToolStripMenuItem6.Click

```

End Sub

```
Private Sub EtatsEtÉditionsToolStripMenuItem_Click(sender As System.Object, e As System.EventArgs) Handles EtatsEtÉditionsToolStripMenuItem.Click
```

End Sub

```
Private Sub AvailabilityToolStripMenuItem_Click(sender As System.Object, e As System.EventArgs) Handles AvailabilityToolStripMenuItem.Click  
    Availability.Show()
```

End Sub

End Class

SaisieEntreprise

```
Public Class SaisieEntreprise
```

```
    Private Sub SaisieEntreprise_Load(sender As Object, e As EventArgs) Handles MyBase.Load
```

```
        variable = 1
        Me.Text = TableauAffichage.GroupAffichage(1).Donnees(10).Libelle(NumLangueDem - 1)
        Label17.Text =
TableauAffichage.GroupAffichage(0).Donnees(96).Libelle(NumLangueDem - 1)
        Label11.Text =
TableauAffichage.GroupAffichage(0).Donnees(95).Libelle(NumLangueDem - 1)
        Label12.Text =
TableauAffichage.GroupAffichage(0).Donnees(53).Libelle(NumLangueDem - 1)
        Label14.Text =
TableauAffichage.GroupAffichage(0).Donnees(61).Libelle(NumLangueDem - 1)
        Label13.Text =
TableauAffichage.GroupAffichage(0).Donnees(62).Libelle(NumLangueDem - 1)
        Label15.Text =
TableauAffichage.GroupAffichage(0).Donnees(66).Libelle(NumLangueDem - 1)
        Label2.Text =
TableauAffichage.GroupAffichage(0).Donnees(106).Libelle(NumLangueDem - 1)
        Label16.Text =
TableauAffichage.GroupAffichage(0).Donnees(64).Libelle(NumLangueDem - 1)
        Label19.Text =
TableauAffichage.GroupAffichage(0).Donnees(113).Libelle(NumLangueDem - 1)
        Enregistrer.Text =
TableauAffichage.GroupAffichage(0).Donnees(114).Libelle(NumLangueDem - 1)
        Supprimer.Text =
TableauAffichage.GroupAffichage(0).Donnees(115).Libelle(NumLangueDem - 1)
        Quitter.Text =
TableauAffichage.GroupAffichage(0).Donnees(116).Libelle(NumLangueDem - 1)
        NoSite.HeaderText =
TableauAffichage.GroupAffichage(0).Donnees(57).Libelle(NumLangueDem - 1)
        NomSite.HeaderText =
TableauAffichage.GroupAffichage(0).Donnees(58).Libelle(NumLangueDem - 1)
        Label10.Text =
TableauAffichage.GroupAffichage(0).Donnees(128).Libelle(NumLangueDem - 1)
        Label11.Text =
TableauAffichage.GroupAffichage(0).Donnees(129).Libelle(NumLangueDem - 1)
        Label12.Text =
TableauAffichage.GroupAffichage(0).Donnees(130).Libelle(NumLangueDem - 1)
        ' Label13.Text =
TableauAffichage.GroupAffichage(0).Donnees(131).Libelle(NumLangueDem - 1)
        Label14.Text =
TableauAffichage.GroupAffichage(0).Donnees(131).Libelle(NumLangueDem - 1)
        Label17.Text =
TableauAffichage.GroupAffichage(0).Donnees(133).Libelle(NumLangueDem - 1)
        Label18.Text =
TableauAffichage.GroupAffichage(0).Donnees(134).Libelle(NumLangueDem - 1)
```

```
End Sub
```

```

Private Sub NoEntreprise_TextChanged(sender As System.Object, e As
System.EventArgs) Handles NoEntreprise.TextChanged
    Dim i As Integer

    Dim TypeEntreprise As String
    Dim ValeurEntreprise As Integer
    If Me.NoEntreprise.Text <> "" And IsNumeric(Me.NoEntreprise.Text) Then

        ' sql = "SELECT NoEntreprise, NomEntreprise, NoSIRET, Adr1, Adr2, CP,
Ville, Pays, Description FROM Entreprise WHERE NoEntreprise = " & Me.NoEntreprise.Text
'NomTable = "Entreprise"

        TypeEntreprise = "NoEntreprise"
        ValeurEntreprise = Me.NoEntreprise.Text
        SelectEntreprise1(TypeEntreprise, ValeurEntreprise)
        SelectEntreprise2(TypeEntreprise, ValeurEntreprise)
        SelectSite()

    If (TableauEntreprise.NbVariable = 0) Then
        ' Me.NoEntreprise.Text = ""
        Me.NomEntreprise.Text = ""
        ReinitialiserSaisieEntreprise()
        'Me.NoSIRET.Text = ""
        ' Me.Adr1.Text = ""
        'Me.Adr2.Text = ""
        'Me.CP.Text = ""
        'Me.Ville.Text = ""
        ' Me.Pays.Text = ""
        'Me.Description.Text = ""
    Else
        For i = 0 To (TableauEntreprise.NbVariable - 1)
            If CInt(Me.NoEntreprise.Text) =
CInt(TableauEntreprise.NoEntreprise) Then
                ' Me.NoEntreprise.Text = dtt.Rows(i).Item("NoEntreprise")
                Me.NomEntreprise.Text = TableauEntreprise.NomEntreprise
                MAJEntreprise()
            End If
        Next i
    End If

    Else
        Me.NoEntreprise.Text = ""
        Me.NomEntreprise.Text = ""
        ReinitialiserSaisieEntreprise()
    End If

End Sub

' Private Sub NomEntreprise_TextChanged(sender As System.Object, e As
System.EventArgs) Handles NomEntreprise.TextChanged
' Dim i As Integer
' Dim TypeEntreprise As String
' Dim ValeurEntreprise As String

```

```

' If Me.NoEntreprise.Text = "" Then
'     If Me.NomEntreprise.Text <> "" Then
'         sql = "SELECT NoEntreprise, NomEntreprise, NoSIRET, Adr1, Adr2, CP, Ville,
Pays, Description FROM Entreprise WHERE NoEntreprise = " & Me.NoEntreprise.Text
'NomTable = "Entreprise"
'         TypeEntreprise = "NomEntreprise"
'         ValeurEntreprise = "'" & Me.NomEntreprise.Text & "'" '
'         SelectEntreprise1(TypeEntreprise, ValeurEntreprise)
'     If (TableauEntreprise.NbVariable = 0) Then
'         Me.NoEntreprise.Text = ""
'Me.NomEntreprise.Text = ""
'         ReinitialiserSaisieEntreprise()
'     Else
'         For i = 0 To (TableauEntreprise.NbVariable - 1)
'             Me.NoEntreprise.Text = TableauEntreprise.NoEntreprise
'             Me.NomEntreprise.Text = TableauEntreprise.NomEntreprise
'             MAJEntreprise()
'         Next i
'     End If
' Else
'     Me.NoEntreprise.Text = ""
'     Me.NomEntreprise.Text = ""
'     ReinitialiserSaisieEntreprise()
' End If
' End If

' End Sub
Private Sub TrouverRapide_TextChanged(sender As System.Object, e As
System.EventArgs) Handles TrouverRapide.TextChanged
    Dim i As Integer
    Dim TypeEntreprise As String
    Dim ValeurEntreprise As String

    If Me.TrouverRapide.Text <> "" Then
        '         sql = "SELECT NoEntreprise, NomEntreprise, NoSIRET, Adr1, Adr2, CP,
Ville, Pays, Description FROM Entreprise WHERE NoEntreprise = " & Me.NoEntreprise.Text
'NomTable = "Entreprise"
TypeEntreprise = "RechercheNomEntreprise"
ValeurEntreprise = "'" & Me.TrouverRapide.Text & "'" '

        SelectEntreprise1(TypeEntreprise, ValeurEntreprise)

        If (TableauEntreprise.NbVariable = 0) Then
            Me.NoEntreprise.Text = ""
            Me.NomEntreprise.Text = ""
            ReinitialiserSaisieEntreprise()
        Else

```

```

        For i = 0 To (TableauEntreprise.NbVariable - 1)
            Me.NoEntreprise.Text = TableauEntreprise.NoEntreprise
            ' Me.NomEntreprise.Text = TableauEntreprise.NomEntreprise
            MAJEntreprise()
        Next i
    End If

End If

```

```
End Sub
```

```
Private Sub Enregistrer_Click(sender As System.Object, e As System.EventArgs)
Handles Enregistrer.Click

```

```
    EnregistrementProcedure()

```

```
End Sub
```

```
Private Sub InsérerEntreprise(ByVal CleAttribuee)

```

```
    Dim ValeurInsert As String

```

```
    Dim Sp As String

```

```
    ' Dim CleAttribuee As String

```

```
    Dim NomInsert As String

```

```
    Sp = "','"

```

```
    ValeurInsert = ""

```

```
    ValeurInsert = CStr(CleAttribuee) & Sp

```

```
    NomInsert = "NoEntreprise,"

```

```
    ValeurInsert = ValeurInsert & Me.NomEntreprise.Text & Sp

```

```
    NomInsert = NomInsert & "NomEntreprise,"

```

```
    ValeurInsert = ValeurInsert & Me.NomEntrepriseAbrege.Text & Sp

```

```
    NomInsert = NomInsert & "RechercheNomEntreprise,"

```

```
    ValeurInsert = ValeurInsert & Me.NoSIRET.Text & Sp

```

```
    NomInsert = NomInsert & "NoSIRET,"

```

```
    ValeurInsert = ValeurInsert & Me.Adr1.Text & Sp

```

```
    NomInsert = NomInsert & "Adr1,"

```

```
    ValeurInsert = ValeurInsert & Me.Adr2.Text & Sp

```

```
    NomInsert = NomInsert & "Adr2,"

```

```
    ValeurInsert = ValeurInsert & Me.CP.Text & Sp

```

```
    NomInsert = NomInsert & "CP,"

```

```
    ValeurInsert = ValeurInsert & Me.Ville.Text & Sp

```

```
    NomInsert = NomInsert & "Ville,"

```

```
    ValeurInsert = ValeurInsert & Me.Pays.Text & Sp

```

```
    NomInsert = NomInsert & "Pays,"

```

```
    ValeurInsert = ValeurInsert & Me.Email.Text & Sp

```

```
    NomInsert = NomInsert & "Email,"

```

```
    ValeurInsert = ValeurInsert & Me.NoTel.Text & Sp

```

```
    NomInsert = NomInsert & "NoTel,"

```



```

ValeurInsert = ValeurInsert & Me.NoFax.Text & Sp
NomInsert = NomInsert & "NoFax,"
ValeurInsert = ValeurInsert & Me.SiteWeb.Text & Sp
NomInsert = NomInsert & "SiteWeb,"
ValeurInsert = ValeurInsert & Me.MentionLegale.Text & Sp
NomInsert = NomInsert & "MentionLegale,"
'ValeurInsert = ValeurInsert & Me.CheminLogo.Text & Sp &
'(NomInsert = NomInsert & "CheminLogo,"
ValeurInsert = ValeurInsert & Me.Description.Text
NomInsert = NomInsert & "Description"
sql = "INSERT INTO Entreprise (" & NomInsert & ") VALUES ('" & ValeurInsert &
""")"

```

```
NomTable = "Entreprise"
```

```
Requete()
```

```
End Sub
```

```
Private Sub ModifierEntreprise(ByVal CleAttribuee)
```

```
Dim Sp1 As String
```

```
Dim Sp2 As String
```

```
Dim NomUpdate As String
```

```
Sp1 = " = '"
```

```
Sp2 = "','"
```

```
NomUpdate = "NomEntreprise" & Sp1 & Me.NomEntreprise.Text & Sp2
```

```
NomUpdate = NomUpdate & "RechercheNomEntreprise" & Sp1 &
```

```
Me.NomEntrepriseAbrege.Text & Sp2
```

```
NomUpdate = NomUpdate & "NoSIRET" & Sp1 & Me.NoSIRET.Text & Sp2
```

```
NomUpdate = NomUpdate & "Adr1" & Sp1 & Me.Adr1.Text & Sp2
```

```
NomUpdate = NomUpdate & "Adr2" & Sp1 & Me.Adr2.Text & Sp2
```

```
NomUpdate = NomUpdate & "Ville" & Sp1 & Me.Ville.Text & Sp2
```

```
NomUpdate = NomUpdate & "Pays" & Sp1 & Me.Pays.Text & Sp2
```

```
NomUpdate = NomUpdate & "NoTel" & Sp1 & Me.NoTel.Text & Sp2
```

```
NomUpdate = NomUpdate & "NoFax" & Sp1 & Me.NoFax.Text & Sp2
```

```
NomUpdate = NomUpdate & "Email" & Sp1 & Me.Email.Text & Sp2
```

```
NomUpdate = NomUpdate & "SiteWeb" & Sp1 & Me.SiteWeb.Text & Sp2
```

```
' NomUpdate = NomUpdate & "CheminLogo" & Sp1 & Me.CheminLogo.Text & Sp2
```

```
NomUpdate = NomUpdate & "MentionLegale" & Sp1 & Me.MentionLegale.Text & Sp2
```

```
NomUpdate = NomUpdate & "Description" & Sp1 & Me.Description.Text
```

```
sql = "UPDATE Entreprise SET " & NomUpdate & "' WHERE NoEntreprise = " &
```

```
CleAttribuee
```

```
NomTable = "Entreprise"
```

```
Requete()
```

```
End Sub
```

```
Private Sub Supprimer_Click(sender As System.Object, e As System.EventArgs)
```

```
Handles Supprimer.Click
```

```
Dim msg As String
```

```
Dim res As Integer
```

```
If Me.NoEntreprise.Text <> "" Then
```

```
msg = TableauAffichage.GroupAffichage(2).Donnees(1).Libelle(NumLangueDem -
```

```
1)
```

```
res = MsgBox(msg, MsgBoxStyle.OkCancel)
```

```
If res = vbOK Then
```

```

        sql = "SELECT NoEntreprise FROM Entreprise WHERE NoEntreprise = " &
Me.NoEntreprise.Text
        NomTable = "Entreprise"

        Requete()

        If dtt.Rows.Count <> 0 Then
            sql = "DELETE FROM Entreprise WHERE NoEntreprise = " &
Me.NoEntreprise.Text
            NomTable = "Entreprise"
            Requete()
        End If
        sql = "DELETE FROM LiaisonEntrepriseSite WHERE NoEntreprise = " &
Me.NoEntreprise.Text
        NomTable = "LiaisonEntrepriseSite"
        Requete()

        Me.NoEntreprise.Text = ""
        Me.NomEntreprise.Text = ""
        ReinitialiserSaisieEntreprise()
    End If

End If

End Sub

Private Sub Quitter_Click(sender As System.Object, e As System.EventArgs) Handles
    Quitter.Click
        QuitterProcedure()

End Sub

Private Sub Site_CellContentClick(sender As System.Object, e As
    System.Windows.Forms.DataGridViewCellEventArgs) Handles Site.CellContentClick

End Sub

Private Sub Button1_Click(sender As System.Object, e As System.EventArgs)

End Sub

Private Sub Site_DragDrop(sender As Object, e As
    System.Windows.Forms.DragEventArgs) Handles Site.DragDrop

End Sub

Private Sub Site_KeyDown(sender As Object, e As System.Windows.Forms.KeyEventArgs)
    Handles Site.KeyDown
        If e.KeyCode = Keys.Tab Or e.KeyCode = Keys.Enter Then

```

```

        MAJSite()

    End If

    If e.KeyCode = Keys.Delete Then
        Site.Rows.RemoveAt(Site.CurrentRow.Index)
    End If
End Sub
Private Sub ReinitialiserSaisieEntreprise()
    ' Dim i As Integer
    'Dim RowToDelete As Int32 =
Me.Site.Rows.GetFirstRow(DataGridViewElementStates.)
    ' Dim NbToDelete As Int32 = Me.Site.Rows.GetRowCount(datagridviewelementstat

    ' Me.NoEntreprise.Text = ""
Me.NomEntreprise.Text = ""
Me.NoSIRET.Text = ""
Me.Adr1.Text = ""
Me.Adr2.Text = ""
Me.CP.Text = ""
Me.Ville.Text = ""
Me.Pays.Text = ""
Me.NoTel.Text = ""
Me.NoFax.Text = ""
Me.Email.Text = ""
    ' Me.CheminLogo.Text = ""
Me.Description.Text = ""
Me.SiteWeb.Text = ""
Me.MentionLegale.Text = ""
Me.NomEntrepriseAbrege.Text = ""
LogoEntreprise.Image =
Image.FromFile(TableauParametres.Parametres(0).Données(4).Adr & "\Default.jpg")
    Site.Rows.Clear()

    ' For i = (Site.Rows.Count - 1) To 0 Step -1
    ' Me.Site.rows[i]["NoSite"].value = ""

    ' Next

    ' Me.Site.CurrentRow(index) = ""

End Sub

Private Sub Site_MouseClick(sender As Object, e As
System.Windows.Forms.MouseEventArgs) Handles Site.MouseClick

    MAJSite()

End Sub
Sub MAJEntreprise()
    Dim j As Integer

    Dim i As Integer
Me.NomEntreprise.Text = TableauEntreprise.NomEntreprise
Me.NoSIRET.Text = TableauEntreprise.NoSIRET
Me.Adr1.Text = TableauEntreprise.Adr1
Me.Adr2.Text = TableauEntreprise.adr2

```

```

Me.CP.Text = TableauEntreprise.CP
Me.Ville.Text = TableauEntreprise.Ville
Me.Pays.Text = TableauEntreprise.Pays
Me.Description.Text = TableauEntreprise.Description
Me.NoTel.Text = TableauEntreprise.NoTel
Me.NoFax.Text = TableauEntreprise.NoFax
Me.Email.Text = TableauEntreprise.Email
Me.SiteWeb.Text = TableauEntreprise.SiteWeb
' Me.CheminLogo.Text = TableauEntreprise.CheminLogo
Me.NomEntrepriseAbrege.Text = TableauEntreprise.RechercheNomEntreprise
Me.MentionLegale.Text = TableauEntreprise.MentionLegale
If NomEntrepriseAbrege.Text <> "" And
(My.Computer.FileSystem.FileExists(TableauParametres.Parametres(0).Données(4).Adr &
"\ " & NomEntrepriseAbrege.Text & ".jpg")) = True Then

    'TableauParametres.Parametres(0).Données(4).Adr & "\ " &
NomEntrepriseAbrege.Text & ".jpg")) Then
    LogoEntreprise.Image =
Image.FromFile(TableauParametres.Parametres(0).Données(4).Adr & "\ " &
NomEntrepriseAbrege.Text & ".jpg")
    End If
    Site.Rows.Clear()
    For j = 0 To CInt(TableauEntreprise.NbLiaison - 1)
        If TableauEntreprise.NoEntreprise =
TableauEntreprise.LiaisonEntrepriseSite(j).NoEntreprise Then
            For i = 0 To (TableauListeSite.NbLiaison - 1)
                If TableauEntreprise.LiaisonEntrepriseSite(j).NoSite =
TableauListeSite.LiaisonEntrepriseSite(i).NoSite Then
                    'TableauLiaisonEntrepriseSite.ListeSite(i).NoSite =
dtt.Rows(i).Item("NoSite")
                    TableauEntreprise.LiaisonEntrepriseSite(j).NomSite =
TableauListeSite.LiaisonEntrepriseSite(i).NomSite
                    End If
                Next i

                'If TableauEntreprise.LiaisonEntrepriseSite(j).NoSite =
TableauLiaisonEntrepriseSite.ListeSite(k) Then
                    Me.Site.Rows.Add(TableauEntreprise.LiaisonEntrepriseSite(j).NoSite,
TableauEntreprise.LiaisonEntrepriseSite(j).NomSite)

            End If
        Next j

    End Sub

Sub MAJSite()
Dim i As Integer
If Me.Site.Rows(Site.CurrentRow.Index).Cells(0).ColumnIndex = 0 Then

    If Me.Site.Rows(Site.CurrentRow.Index).Cells(0).Value <> 0 Then
        For i = 0 To (TableauListeSite.NbLiaison - 1)
            If Me.Site.Rows(Site.CurrentRow.Index).Cells(0).Value =
TableauListeSite.LiaisonEntrepriseSite(i).NoSite Then
                'TableauLiaisonEntrepriseSite.ListeSite(i).NoSite =
dtt.Rows(i).Item("NoSite")
                Me.Site.Rows(Site.CurrentRow.Index).Cells(1).Value =
TableauListeSite.LiaisonEntrepriseSite(i).NomSite
                End If
            Next i
        End Sub
    End Sub

```

```

        End If
    End If
End Sub
Sub EnregistrerSite(ByVal CleAttribuee)
    Dim j As Integer
    Dim k As Integer
    Dim typeEntreprise As String
    Dim ValeurEntreprise As String
    Dim i As Integer
    Dim Tableau() As Integer
    Dim Sp As String
    Dim ValeurInsert As String
    Dim TableauASupprimer() As Integer
    Dim TableauSauve(100) As Integer
    Dim Indice As Integer
    Dim indice2 As Integer
    Dim l As Integer
    Dim m As Integer
    Dim trouve As Boolean
    'sql = "SELECT NoLiaisonEntrepriseSite FROM LiaisonEntrepriseSite WHERE
NoEntreprise = " & Me.NoEntreprise.Text
    'NomTable = "LiaisonEntrepriseSite"

    ' Requete()
    Sp = "','"
    If CleAttribuee <> "" Then
        typeEntreprise = "NoEntreprise"
        ValeurEntreprise = CleAttribuee

        SelectEntreprise2(typeEntreprise, ValeurEntreprise)
        indice2 = 0
        ReDim Tableau(100)
        ReDim TableauASupprimer(100)
        If Site.RowCount - 1 > TableauEntreprise.NbLiaison - 1 Then
            indice2 = 0
            Indice = 0
            For i = 0 To Site.RowCount - 1
                TableauSauve(i) = Me.Site.Rows(i).Cells(0).Value
            Next
            For i = 0 To Site.RowCount - 1

                TableauASupprimer(Indice) = 0
                Tableau(indice2) = 0
                Tableau(Indice) = Me.Site.Rows(i).Cells(0).Value
                ' Tableau(indice2) = Me.Site.Rows(i).Cells(0).Value
                ' TableauASupprimer(Indice) = Me.Site.Rows(i).Cells(0).Value
                For j = 0 To TableauEntreprise.NbLiaison - 1
                    If Me.Site.Rows(i).Cells(0).Value <>
TableauEntreprise.LiaisonEntrepriseSite(j).NoSite Then

                        ' Tableau(indice2) = 0

                        ' Else
                        ' TableauASupprimer(Indice) = 0
                        ' End If

                        'TableauASupprimer(Indice) =
Me.Site.Rows(i).Cells(0).Value
                        ' Tableau(Indice) = Me.Site.Rows(i).Cells(0).Value

```

```

        For l = 0 To Indice
            If TableauEntreprise.LiaisonEntrepriseSite(j).NoSite =
Tableau(l) Then
                Tableau(l) = 0
            End If

            Next l
            trouve = False
            For l = 0 To Site.RowCount - 1
                For m = 0 To indice2
                    If
(TableauEntreprise.LiaisonEntrepriseSite(j).NoSite = TableauSauve(l)) Or
TableauEntreprise.LiaisonEntrepriseSite(j).NoSite = TableauASupprimer(m) Then

                        trouve = True
                    End If
                Next m
            Next l

            If trouve = False Then
                TableauASupprimer(indice2) =
TableauEntreprise.LiaisonEntrepriseSite(j).NoSite
            End If
        End If

        Next j
        indice2 = indice2 + 1
        Indice = Indice + 1
    Next i

    For k = 0 To Indice - 1
        If TableauASupprimer(k) <> 0 Then
            sql = "DELETE FROM LiaisonEntrepriseSite WHERE NoSite = " &
TableauASupprimer(k) & " AND NoEntreprise = " & CleAttribuee
            NomTable = "LiaisonEntrepriseSite"

            Requete()
        End If

    Next k

    For k = 0 To indice2 - 1
        If Tableau(k) <> 0 Then

            ValeurInsert = Tableau(k) & Sp & CleAttribuee
            sql = "INSERT INTO LiaisonEntrepriseSite (NoSite,
NoEntreprise) VALUES ('" & ValeurInsert & "')"

            NomTable = "LiaisonEntrepriseSite"

            Requete()
        End If

    Next k

Else
    ReDim TableauASupprimer(100)

```



```

        If Tableau(k) <> 0 Then
            ValeurInsert = Tableau(k) & Sp & CleAttribuee
            sql = "INSERT INTO LiaisonEntrepriseSite (NoSite,
NoEntreprise) VALUES ('" & ValeurInsert & "'"")"

            NomTable = "LiaisonEntrepriseSite"

            Requete()
        End If
    Next
    For k = 0 To Indice - 1
        If TableauASupprimer(k) <> 0 Then
            sql = "DELETE FROM LiaisonEntrepriseSite WHERE NoSite = " &
TableauASupprimer(k) & " AND NoEntreprise = " & CleAttribuee
            NomTable = "LiaisonEntrepriseSite"

            Requete()
        End If
    Next k

    End If

End If

End Sub
Sub EnregistrementProcedure()

```

```

Dim i As Integer
Dim j As Integer
Dim CleAttribuee As String

```

```

If Me.NoEntreprise.Text = "" Then
    sql = "SELECT NoEntreprise FROM Entreprise"
    NomTable = "Entreprise"

    Requete()
    j = CInt(dtt.Rows(0).Item("NoEntreprise"))
    CleAttribuee = CStr(j + 1)
    For i = 0 To dtt.Rows.Count - 1

        If CInt(dtt.Rows(i).Item("NoEntreprise")) < j Then

            CleAttribuee = (dtt.Rows(i).Item("NoEntreprise") + 1)
        End If
    Next i

Else
    CleAttribuee = Me.NoEntreprise.Text

End If

```



```

        sql = "SELECT NoEntreprise FROM Entreprise WHERE NoEntreprise = " &
CleAttribuee
        NomTable = "Entreprise"

        Requete()
        If Me.CP.Text = "" Then
            Me.CP.Text = "0"
        End If

        If dtt.Rows.Count = 0 Then
            InserirEntreprise(CleAttribuee)

        Else
            ModifierEntreprise(CleAttribuee)

        End If
        EnregistrerSite(CleAttribuee)
        Me.NoEntreprise.Text = ""
        Me.NomEntreprise.Text = ""

        ReinitialiserSaisieEntreprise()
    End Sub
    Sub QuitterProcedure()

        Dim res As Integer
        Dim msg As String
        If Me.NomEntreprise.Text <> "" Then

1)         msg = TableauAffichage.GroupAffichage(2).Donnees(0).Libelle(NumLangueDem -

            res = MsgBox(msg, MsgBoxStyle.YesNoCancel)
            Select Case res
                Case vbYes
                    EnregistrementProcedure()
                    Me.Hide()

                Case vbNo
                    Me.NoEntreprise.Text = ""
                    Me.TrouverRapide.Text = ""
                    ReinitialiserSaisieEntreprise()
                    Me.Hide()
                Case vbCancel

            End Select
        Else
            Me.Hide()
        End If
    End Sub

    Private Sub LogoEntreprise_Click(sender As System.Object, e As System.EventArgs)
Handles LogoEntreprise.Click
        ' If OFGSelectImage.ShowDialog = Windows.Forms.DialogResult.OK Then
        ' LogoEntreprise.Image = Image.FromFile(OFGSelectImage.FileName)
        ' End If

    End Sub

```

```
Private Sub Label10_Click(sender As System.Object, e As System.EventArgs) Handles  
Label10.Click  
  
End Sub  
End Class
```

SaisieSites

```
Public Class SaisieSites
```

```
    Private Sub SaisieSites_Load(sender As Object, e As EventArgs) Handles MyBase.Load
```

```
        Me.Text = TableauAffichage.GroupAffichage(1).Donnees(14).Libelle(NumLangueDem  
- 1)  
        Label17.Text =  
TableauAffichage.GroupAffichage(0).Donnees(57).Libelle(NumLangueDem - 1)  
        Label11.Text =  
TableauAffichage.GroupAffichage(0).Donnees(58).Libelle(NumLangueDem - 1)  
  
        Label16.Text =  
TableauAffichage.GroupAffichage(0).Donnees(64).Libelle(NumLangueDem - 1)  
        Label18.Text =  
TableauAffichage.GroupAffichage(0).Donnees(65).Libelle(NumLangueDem - 1)  
        Label15.Text =  
TableauAffichage.GroupAffichage(0).Donnees(66).Libelle(NumLangueDem - 1)  
        Label14.Text =  
TableauAffichage.GroupAffichage(0).Donnees(61).Libelle(NumLangueDem - 1)  
        Label13.Text =  
TableauAffichage.GroupAffichage(0).Donnees(62).Libelle(NumLangueDem - 1)  
  
        Label139.Text =  
TableauAffichage.GroupAffichage(0).Donnees(157).Libelle(NumLangueDem - 1)  
        ' Label21.Text =  
TableauAffichage.GroupAffichage(0).Donnees(158).Libelle(NumLangueDem - 1)  
  
        Label12.Text =  
TableauAffichage.GroupAffichage(0).Donnees(106).Libelle(NumLangueDem - 1)  
  
        Label110.Text =  
TableauAffichage.GroupAffichage(0).Donnees(64).Libelle(NumLangueDem - 1)  
        Label19.Text =  
TableauAffichage.GroupAffichage(0).Donnees(65).Libelle(NumLangueDem - 1)  
        Label111.Text =  
TableauAffichage.GroupAffichage(0).Donnees(66).Libelle(NumLangueDem - 1)  
        ' Label11.Text =  
TableauAffichage.GroupAffichage(0).Donnees(65).Libelle(NumLangueDem - 1)  
        Label112.Text =  
TableauAffichage.GroupAffichage(0).Donnees(61).Libelle(NumLangueDem - 1)  
        Label113.Text =  
TableauAffichage.GroupAffichage(0).Donnees(62).Libelle(NumLangueDem - 1)  
  
        'Label20.Text =  
TableauAffichage.GroupAffichage(0).Donnees(68).Libelle(NumLangueDem - 1)  
        Label17.Text =  
TableauAffichage.GroupAffichage(0).Donnees(64).Libelle(NumLangueDem - 1)  
        Label116.Text =  
TableauAffichage.GroupAffichage(0).Donnees(65).Libelle(NumLangueDem - 1)  
        Label118.Text =  
TableauAffichage.GroupAffichage(0).Donnees(66).Libelle(NumLangueDem - 1)  
        Label114.Text =  
TableauAffichage.GroupAffichage(0).Donnees(61).Libelle(NumLangueDem - 1)
```

```
Label115.Text =
TableauAffichage.GroupAffichage(0).Donnees(62).Libelle(NumLangueDem - 1)

Label137.Text =
TableauAffichage.GroupAffichage(0).Donnees(133).Libelle(NumLangueDem - 1)
Label138.Text =
TableauAffichage.GroupAffichage(0).Donnees(134).Libelle(NumLangueDem - 1)

Label134.Text =
TableauAffichage.GroupAffichage(0).Donnees(136).Libelle(NumLangueDem - 1)
Label135.Text =
TableauAffichage.GroupAffichage(0).Donnees(136).Libelle(NumLangueDem - 1)
Label136.Text =
TableauAffichage.GroupAffichage(0).Donnees(136).Libelle(NumLangueDem - 1)

Label24.Text =
TableauAffichage.GroupAffichage(0).Donnees(128).Libelle(NumLangueDem - 1)
Label32.Text =
TableauAffichage.GroupAffichage(0).Donnees(128).Libelle(NumLangueDem - 1)
Label28.Text =
TableauAffichage.GroupAffichage(0).Donnees(128).Libelle(NumLangueDem - 1)

Label23.Text =
TableauAffichage.GroupAffichage(0).Donnees(129).Libelle(NumLangueDem - 1)
Label31.Text =
TableauAffichage.GroupAffichage(0).Donnees(129).Libelle(NumLangueDem - 1)
Label27.Text =
TableauAffichage.GroupAffichage(0).Donnees(129).Libelle(NumLangueDem - 1)

Label22.Text =
TableauAffichage.GroupAffichage(0).Donnees(130).Libelle(NumLangueDem - 1)
Label30.Text =
TableauAffichage.GroupAffichage(0).Donnees(130).Libelle(NumLangueDem - 1)
Label26.Text =
TableauAffichage.GroupAffichage(0).Donnees(130).Libelle(NumLangueDem - 1)

Label25.Text =
TableauAffichage.GroupAffichage(0).Donnees(131).Libelle(NumLangueDem - 1)
Label33.Text =
TableauAffichage.GroupAffichage(0).Donnees(131).Libelle(NumLangueDem - 1)
Label29.Text =
TableauAffichage.GroupAffichage(0).Donnees(131).Libelle(NumLangueDem - 1)

NoAtelier.HeaderText =
TableauAffichage.GroupAffichage(0).Donnees(55).Libelle(NumLangueDem - 1)
NomAtelier.HeaderText =
TableauAffichage.GroupAffichage(0).Donnees(56).Libelle(NumLangueDem - 1)

Enregistrer.Text =
TableauAffichage.GroupAffichage(0).Donnees(114).Libelle(NumLangueDem - 1)
Supprimer.Text =
TableauAffichage.GroupAffichage(0).Donnees(115).Libelle(NumLangueDem - 1)
Quitter.Text =
TableauAffichage.GroupAffichage(0).Donnees(116).Libelle(NumLangueDem - 1)
' Me.NoSite.Text = SaisieEntreprise.CP.Text
```

End Sub

```

Private Sub NoSite_TextChanged(sender As System.Object, e As System.EventArgs)
Handles NoSite.TextChanged
    Dim i As Integer

    Dim TypeSite As String
    Dim ValeurSite As Integer
    If Me.NoSite.Text <> "" And IsNumeric(Me.NoSite.Text) Then

        ' sql = "SELECT NoEntreprise, NomEntreprise, NoSIRET, Adr1, Adr2, CP,
Ville, Pays, Description FROM Entreprise WHERE NoEntreprise = " & Me.NoEntreprise.Text
'NomTable = "Entreprise"

        TypeSite = "NoSite"
        ValeurSite = Me.NoSite.Text
        SelectSite1(TypeSite, ValeurSite)
        SelectSite2(TypeSite, ValeurSite)
        SelectAtelier()

    If (TableauSite.NbVariable = 0) Then

        Me.NomSite.Text = ""
        ReinitialiserSaisieSite()

    Else

        For i = 0 To (TableauSite.NbVariable - 1)
            If CInt(Me.NoSite.Text) = CInt(TableauSite.NoSite) Then
                ' Me.NoEntreprise.Text = dtt.Rows(i).Item("NoEntreprise")
                Me.NomSite.Text = TableauEntreprise.NomEntreprise
                MAJSite()
            End If
        Next i
    End If

    Else
        Me.NoSite.Text = ""
        Me.NomSite.Text = ""
        ReinitialiserSaisieSite()
    End If

End Sub

Private Sub TrouverRapide_TextChanged(sender As System.Object, e As
System.EventArgs) Handles TrouverRapide.TextChanged
    Dim i As Integer
    Dim TypeSite As String
    Dim ValeurSite As String

    If Me.TrouverRapide.Text <> "" Then

        ' sql = "SELECT NoEntreprise, NomEntreprise, NoSIRET, Adr1, Adr2, CP,
Ville, Pays, Description FROM Entreprise WHERE NoEntreprise = " & Me.NoEntreprise.Text
'NomTable = "Entreprise"
        TypeSite = "RechercheNomSite"
        ValeurSite = "" & Me.TrouverRapide.Text & "" '

```

```

SelectSite1(TypeSite, ValeurSite)

If (TableauSite.NbVariable = 0) Then
    Me.NoSite.Text = ""
    'Me.NomEntreprise.Text = ""
    '    ReinitialiserSaisieEntreprise()
Else
    For i = 0 To (TableauSite.NbVariable - 1)
        Me.NoSite.Text = TableauSite.NoSite
        '    Me.NomEntreprise.Text = TableauEntreprise.NomEntreprise
        MAJSite()
    Next i
End If

End If

End Sub

' Private Sub NomSite_TextChanged(sender As System.Object, e As System.EventArgs)
Handles NomSite.TextChanged
'Dim i As Integer
'Dim TypeSite As String
'Dim ValeurSite As String

'    If Me.NoSite.Text = "" Then

'        If Me.NomSite.Text <> "" Then

'            TypeSite = "NomSite"
'            ValeurSite = "" & Me.NomSite.Text & "" '

'            SelectSite1(TypeSite, ValeurSite)

'            If (TableauSite.NbVariable = 0) Then
'                Me.NoSite.Text = ""

'                ReinitialiserSaisieSite()
'            Else

'                For i = 0 To (TableauSite.NbVariable - 1)

'                    Me.NoSite.Text = TableauSite.NoSite
'                    Me.NomSite.Text = TableauSite.NomSite
'                    MAJSite()
'                Next i
'            End If

'        Else

'            Me.NoSite.Text = ""
'            Me.NomSite.Text = ""
'            ReinitialiserSaisieSite()

```

```

' End If
' End If

'End Sub

Private Sub Enregistrer_Click(sender As System.Object, e As System.EventArgs)
Handles Enregistrer.Click

    EnregistrementProcedure()

    'Dim CleAttribuee As String

    '      If Me.NoSite.Text = "" Then

    '   sql = "SELECT MAX(NoSite) as MaxNoSite FROM Site"
    '   NomTable = "Adresse"

    'Requete()

    'CleAttribuee = dtt.Rows(0).Item("MaxNoSite") + 1
    ' sql = "SELECT NoSite FROM Site"
    '   NomTable = "Site"

    '   Requete()
    '   j = CInt(dtt.Rows(0).Item("NoSite"))
    '   CleAttribuee = CStr(j + 1)
    '   For i = 0 To dtt.Rows.Count - 1

    'If CInt(dtt.Rows(i).Item("NoSite")) < j Then

    'CleAttribuee = (dtt.Rows(i).Item("NoSite") + 1)
    ' End If

    '   Next i

    'Else
    'CleAttribuee = Me.NoSite.Text

    '      End If

    'sql = "SELECT NoSite FROM Site WHERE NoSite = " & CleAttribuee
    'NomTable = "Site"

    'Requete()
    'If Me.CPSite.Text = "" Then
    'Me.CPSite.Text = "0"
    'End If
    'If Me.CPLivraison.Text = "" Then
    'Me.CPLivraison.Text = "0"
    'End If
    'If Me.CPFacturation.Text = "" Then
    ' Me.CPFacturation.Text = "0"
    ' End If

```

```

' If dtt.Rows.Count = 0 Then

'InsererSite(CleAttribuee)

'Else
'ModifierSite(CleAttribuee)

'End If
'EnregistrerAtelier(CleAttribuee)
'Me.NoSite.Text = ""
'Me.NomSite.Text = ""
'ReinitialiserSaisieSite()

```

End Sub

```

Private Sub InsererSite(ByVal CleAttribuee)
    Dim ValeurInsert As String
    Dim Sp As String
    Dim CleAdresseSite As Integer
    Dim CleAdresseLivraison As Integer
    Dim CleAdresseFacturation As Integer
    Dim NbAdresse As Integer
    ' Dim CleAttribuee As String
    Dim NomInsert As String

    sql = "SELECT MAX(NbAdresse) as MaxNbAdresse FROM Adresse"
    NomTable = "Adresse"

    Requete()

    NbAdresse = CInt(dtt.Rows(0).Item("MaxNbAdresse"))
    CleAdresseSite = NbAdresse + 2
    CleAdresseLivraison = NbAdresse + 3
    CleAdresseFacturation = NbAdresse + 4
    ' For i = 0 To dtt.Rows.Count - 1

    'If CInt(dtt.Rows(i).Item("NoSite")) < j Then

    'CleAttribuee = (dtt.Rows(i).Item("NoSite") + 1)
    ' End If

    'Next i

    Sp = "','"
    ValeurInsert = ""
    ValeurInsert = CStr(CleAttribuee) & Sp
    NomInsert = "NoSite,"
    ValeurInsert = ValeurInsert & Me.NomSite.Text & Sp
    NomInsert = NomInsert & "NomSite,"
    ValeurInsert = ValeurInsert & Me.NomSiteAbrege.Text & Sp
    NomInsert = NomInsert & "RechercheNomSite,"
    ValeurInsert = ValeurInsert & CleAdresseSite & Sp
    NomInsert = NomInsert & "NoAdresseSite,"
    ValeurInsert = ValeurInsert & CleAdresseLivraison & Sp
    NomInsert = NomInsert & "NoAdresseLivraison,"
    ValeurInsert = ValeurInsert & CleAdresseFacturation & Sp
    NomInsert = NomInsert & "NoAdresseFacturation,"

    ValeurInsert = ValeurInsert & Me.NoDisponibilite.Text & Sp

```



```
NomInsert = NomInsert & "NoDisponibilite,"
```

```
ValeurInsert = ValeurInsert & Me.Description.Text
```

```
NomInsert = NomInsert & "Description"
```

```
sql = "INSERT INTO Site (" & NomInsert & ") VALUES ('" & ValeurInsert & "'")"
```

```
NomTable = "Site"
```

```
Requete()
```

```
If Me.CPLivraison.Text = "" Then
```

```
    Me.CPLivraison.Text = 0
```

```
End If
```

```
ValeurInsert = ""
```

```
NomInsert = ""
```

```
ValeurInsert = ValeurInsert & CleAdresseLivraison & Sp
```

```
NomInsert = NomInsert & "NoAdresse,"
```

```
ValeurInsert = ValeurInsert & Me.Adr1Livraison.Text & Sp
```

```
NomInsert = NomInsert & "Adr1,"
```

```
ValeurInsert = ValeurInsert & Me.Adr2Livraison.Text & Sp
```

```
NomInsert = NomInsert & "Adr2,"
```

```
ValeurInsert = ValeurInsert & Me.CPLivraison.Text & Sp
```

```
NomInsert = NomInsert & "CP,"
```

```
ValeurInsert = ValeurInsert & Me.VilleLivraison.Text & Sp
```

```
NomInsert = NomInsert & "Ville,"
```

```
ValeurInsert = ValeurInsert & Me.PaysLivraison.Text & Sp
```

```
NomInsert = NomInsert & "Pays,"
```

```
ValeurInsert = ValeurInsert & Me.NoTelLivraison.Text & Sp
```

```
NomInsert = NomInsert & "NoTel,"
```

```
ValeurInsert = ValeurInsert & Me.NoFaxLivraison.Text & Sp
```

```
NomInsert = NomInsert & "NoFax,"
```

```
ValeurInsert = ValeurInsert & Me.EmailLivraison.Text & Sp
```

```
NomInsert = NomInsert & "Email,"
```

```
ValeurInsert = ValeurInsert & Me.SiteWebLivraison.Text & Sp
```

```
NomInsert = NomInsert & "SiteWeb,"
```

```
ValeurInsert = ValeurInsert & Me.ContactLivraison.Text
```

```
NomInsert = NomInsert & "Contact"
```

```
sql = "INSERT INTO Adresse (" & NomInsert & ") VALUES ('" & ValeurInsert & "'")"
```

```
NomTable = "Adresse"
```

```
Requete()
```

```
If Me.CPSite.Text = "" Then
```

```
    Me.CPSite.Text = 0
```

```
End If
```

```
ValeurInsert = ""
```

```
NomInsert = ""
```

```
ValeurInsert = ValeurInsert & CleAdresseSite & Sp
```

```
NomInsert = NomInsert & "NoAdresse,"
```

```
ValeurInsert = ValeurInsert & Me.Adr1Site.Text & Sp
```

```
NomInsert = NomInsert & "Adr1,"
```

```
ValeurInsert = ValeurInsert & Me.Adr2Site.Text & Sp
```

```
NomInsert = NomInsert & "Adr2,"
```

```
ValeurInsert = ValeurInsert & Me.CPSite.Text & Sp
```

```
NomInsert = NomInsert & "CP,"
```

```
ValeurInsert = ValeurInsert & Me.VilleSite.Text & Sp
```

```
NomInsert = NomInsert & "Ville,"
```

```
ValeurInsert = ValeurInsert & Me.PaysSite.Text & Sp
```

```
NomInsert = NomInsert & "Pays,"
```

```
ValeurInsert = ValeurInsert & Me.NoTelSite.Text & Sp
```

```

NomInsert = NomInsert & "NoTel,"
ValeurInsert = ValeurInsert & Me.NoFaxSite.Text & Sp
NomInsert = NomInsert & "NoFax,"
ValeurInsert = ValeurInsert & Me.EmailSite.Text & Sp
NomInsert = NomInsert & "Email,"
ValeurInsert = ValeurInsert & Me.SiteWebSite.Text & Sp
NomInsert = NomInsert & "SiteWeb,"
ValeurInsert = ValeurInsert & Me.ContactSite.Text
NomInsert = NomInsert & "Contact"

sql = "INSERT INTO Adresse (" & NomInsert & ") VALUES ('" & ValeurInsert &
"')"

NomTable = "Adresse"

Requete()

If Me.CPFacturation.Text = "" Then
    Me.CPFacturation.Text = 0
End If
ValeurInsert = ""
NomInsert = ""
ValeurInsert = ValeurInsert & CleAdresseFacturation & Sp
NomInsert = NomInsert & "NoAdresse,"
ValeurInsert = ValeurInsert & Me.Adr1Facturation.Text & Sp
NomInsert = NomInsert & "Adr1,"
ValeurInsert = ValeurInsert & Me.Adr2Facturation.Text & Sp
NomInsert = NomInsert & "Adr2,"
ValeurInsert = ValeurInsert & Me.CPFacturation.Text & Sp
NomInsert = NomInsert & "CP,"
ValeurInsert = ValeurInsert & Me.VilleFacturation.Text & Sp
NomInsert = NomInsert & "Ville,"
ValeurInsert = ValeurInsert & Me.PaysFacturation.Text & Sp
NomInsert = NomInsert & "Pays,"
ValeurInsert = ValeurInsert & Me.NoTelFacturation.Text & Sp
NomInsert = NomInsert & "NoTel,"
ValeurInsert = ValeurInsert & Me.NoFaxFacturation.Text & Sp
NomInsert = NomInsert & "NoFax,"
ValeurInsert = ValeurInsert & Me.EmailFacturation.Text & Sp
NomInsert = NomInsert & "Email,"
ValeurInsert = ValeurInsert & Me.SiteWebFacturation.Text & Sp
NomInsert = NomInsert & "SiteWeb,"
ValeurInsert = ValeurInsert & Me.ContactFacturation.Text
NomInsert = NomInsert & "Contact"

sql = "INSERT INTO Adresse (" & NomInsert & ") VALUES ('" & ValeurInsert &
"')"

NomTable = "Adresse"

Requete()

End Sub
Private Sub ModifierSite(ByVal CleAttribuee)

Dim Sp1 As String
Dim Sp2 As String

Dim NomUpdate As String
Sp1 = " = '"
Sp2 = "',"
NomUpdate = "NomSite" & Sp1 & Me.NomSite.Text & Sp2
NomUpdate = NomUpdate & "RechercheNomSite" & Sp1 & Me.NomSiteAbrege.Text & Sp2

```

```
NomUpdate = NomUpdate & "NoAdresseSite" & Sp1 &  
TableauSite.LiaisonSiteAdresse(0).NoAdresse & Sp2
```

```
NomUpdate = NomUpdate & "NoAdresseFacturation" & Sp1 &  
TableauSite.LiaisonSiteAdresse(1).NoAdresse & Sp2  
NomUpdate = NomUpdate & "NoAdresseLivraison" & Sp1 &  
TableauSite.LiaisonSiteAdresse(2).NoAdresse & Sp2  
NomUpdate = NomUpdate & "NoDisponibilite" & Sp1 & Me.NoDisponibilite.Text &  
Sp2
```

```
NomUpdate = NomUpdate & "Description" & Sp1 & Me.Description.Text  
sql = "UPDATE Site SET " & NomUpdate & "' WHERE NoSite = " & CleAttribuee  
NomTable = "Site"
```

```
Requete()
```

```
NomUpdate = ""  
NomUpdate = NomUpdate & "Adr1" & Sp1 & Me.Adr1Site.Text & Sp2  
NomUpdate = NomUpdate & "Adr2" & Sp1 & Me.Adr2Site.Text & Sp2  
NomUpdate = NomUpdate & "CP" & Sp1 & Me.CPSite.Text & Sp2  
NomUpdate = NomUpdate & "ville" & Sp1 & Me.VilleSite.Text & Sp2  
NomUpdate = NomUpdate & "Pays" & Sp1 & Me.PaysSite.Text & Sp2  
NomUpdate = NomUpdate & "NoTel" & Sp1 & Me.NoTelSite.Text & Sp2  
NomUpdate = NomUpdate & "NoFax" & Sp1 & Me.NoFaxSite.Text & Sp2  
NomUpdate = NomUpdate & "Email" & Sp1 & Me.EmailSite.Text & Sp2  
NomUpdate = NomUpdate & "SiteWeb" & Sp1 & Me.SiteWebSite.Text & Sp2  
NomUpdate = NomUpdate & "Contact" & Sp1 & Me.ContactSite.Text  
sql = "UPDATE Adresse SET " & NomUpdate & "' WHERE NoAdresse = " &  
TableauSite.LiaisonSiteAdresse(0).NoAdresse  
NomTable = "Adresse"
```

```
Requete()
```

```
NomUpdate = ""  
NomUpdate = NomUpdate & "Adr1" & Sp1 & Me.Adr1Facturation.Text & Sp2  
NomUpdate = NomUpdate & "Adr2" & Sp1 & Me.Adr2Facturation.Text & Sp2  
NomUpdate = NomUpdate & "CP" & Sp1 & Me.CPFacturation.Text & Sp2  
NomUpdate = NomUpdate & "ville" & Sp1 & Me.VilleFacturation.Text & Sp2  
NomUpdate = NomUpdate & "Pays" & Sp1 & Me.PaysFacturation.Text & Sp2  
NomUpdate = NomUpdate & "NoTel" & Sp1 & Me.NoTelFacturation.Text & Sp2  
NomUpdate = NomUpdate & "NoFax" & Sp1 & Me.NoFaxFacturation.Text & Sp2  
NomUpdate = NomUpdate & "Email" & Sp1 & Me.EmailFacturation.Text & Sp2  
NomUpdate = NomUpdate & "SiteWeb" & Sp1 & Me.SiteWebFacturation.Text & Sp2  
NomUpdate = NomUpdate & "Contact" & Sp1 & Me.ContactFacturation.Text  
sql = "UPDATE Adresse SET " & NomUpdate & "' WHERE NoAdresse = " &  
TableauSite.LiaisonSiteAdresse(1).NoAdresse  
NomTable = "Adresse"
```

```
Requete()
```

```
NomUpdate = ""  
NomUpdate = NomUpdate & "Adr1" & Sp1 & Me.Adr1Livraison.Text & Sp2  
NomUpdate = NomUpdate & "Adr2" & Sp1 & Me.Adr2Livraison.Text & Sp2  
NomUpdate = NomUpdate & "CP" & Sp1 & Me.CPLivraison.Text & Sp2  
NomUpdate = NomUpdate & "ville" & Sp1 & Me.VilleLivraison.Text & Sp2  
NomUpdate = NomUpdate & "Pays" & Sp1 & Me.PaysLivraison.Text & Sp2  
NomUpdate = NomUpdate & "NoTel" & Sp1 & Me.NoTelLivraison.Text & Sp2  
NomUpdate = NomUpdate & "NoFax" & Sp1 & Me.NoFaxLivraison.Text & Sp2  
NomUpdate = NomUpdate & "Email" & Sp1 & Me.EmailLivraison.Text & Sp2  
NomUpdate = NomUpdate & "SiteWeb" & Sp1 & Me.SiteWebLivraison.Text & Sp2
```

```

        NomUpdate = NomUpdate & "Contact" & Sp1 & Me.ContactLivraison.Text
        sql = "UPDATE Adresse SET " & NomUpdate & "' WHERE NoAdresse = " &
TableauSite.LiaisonSiteAdresse(2).NoAdresse
        NomTable = "Adresse"

        Requete()

    End Sub
    Private Sub Supprimer_Click(sender As System.Object, e As System.EventArgs)
Handles Supprimer.Click
        Dim i As Integer
        If Me.NoSite.Text <> "" Then
            sql = "SELECT NoSite FROM Site WHERE NoSite = " & Me.NoSite.Text
            NomTable = "Site"

            Requete()

            If dtt.Rows.Count <> 0 Then
                sql = "DELETE FROM Site WHERE NoSite= " & Me.NoSite.Text
                NomTable = "Site"
                Requete()
            End If
            sql = "DELETE FROM LiaisonSiteAtelier WHERE NoSite = " & Me.NoSite.Text
            NomTable = "LiaisonSiteAtelier"
            Requete()
            For i = 0 To 2
                sql = "DELETE FROM Adresse WHERE NoAdresse = " &
TableauSite.LiaisonSiteAdresse(i).NoAdresse
                NomTable = "Adresse"
                Requete()
            Next
        End If
        Me.NoSite.Text = ""
        Me.NomSite.Text = ""
        ReinitialiserSaisieSite()

    End Sub

    Private Sub Quitter_Click(sender As System.Object, e As System.EventArgs) Handles
Quitter.Click
        QuitterProcedure()

    End Sub

    Private Sub Site_KeyDown(sender As Object, e As System.Windows.Forms.KeyEventArgs)
        If e.KeyCode = Keys.Tab Or e.KeyCode = Keys.Enter Then

            MAJAtelier()

        End If

        If e.KeyCode = Keys.Delete Then
            Atelier.Rows.RemoveAt(Atelier.CurrentRow.Index)
        End If
    End Sub

```

```

Private Sub ReinitialiserSaisieSite()

    Me.NomSite.Text = ""
    Me.NoDisponibilite.Text = ""

    Me.Adr1Site.Text = ""
    Me.Adr2Site.Text = ""
    Me.CPSite.Text = ""
    Me.VilleSite.Text = ""
    Me.PaysSite.Text = ""
    Me.NoTelSite.Text = ""
    Me.NoFaxSite.Text = ""
    Me.EmailSite.Text = ""
    Me.SiteWebSite.Text = ""
    Me.ContactSite.Text = ""

    Me.Adr1Livraison.Text = ""
    Me.Adr2Livraison.Text = ""
    Me.CPLivraison.Text = ""
    Me.VilleLivraison.Text = ""
    Me.PaysLivraison.Text = ""
    Me.NoTellivraison.Text = ""
    Me.NoFaxLivraison.Text = ""
    Me.EmailLivraison.Text = ""
    Me.SiteWebLivraison.Text = ""
    Me.ContactLivraison.Text = ""

    Me.Adr1Facturation.Text = ""
    Me.Adr2Facturation.Text = ""
    Me.CPFacturation.Text = ""
    Me.VilleFacturation.Text = ""
    Me.PaysFacturation.Text = ""
    Me.NoTelFacturation.Text = ""
    Me.NoFaxFacturation.Text = ""
    Me.EmailFacturation.Text = ""
    Me.SiteWebFacturation.Text = ""
    Me.ContactFacturation.Text = ""

    Me.NomSiteAbrege.Text = ""

    ' Me.CheminLogo.Text = ""
    Me.Description.Text = ""
    LogoSite.Image = Image.FromFile(TableauParametres.Parametres(0).Données(4).Adr
& "\Defaut.jpg")
    Atelier.Rows.Clear()

End Sub

Private Sub Atelier_MouseClick(sender As Object, e As
System.Windows.Forms.MouseEventArgs)

    MAJAtelier()

End Sub
Sub MAJSite()
    Dim j As Integer
    Dim i As Integer

    Me.NomSite.Text = TableauSite.NomSite
    Me.NomSiteAbrege.Text = TableauSite.RechercheNomSite

```

```
Me.NoDisponibilite.Text = TableauSite.NoDisponibilite
Me.Adr1Site.Text = TableauSite.LiaisonSiteAdresse(0).Adr1
Me.Adr2Site.Text = TableauSite.LiaisonSiteAdresse(0).Adr2
Me.CPSite.Text = TableauSite.LiaisonSiteAdresse(0).CP
Me.VilleSite.Text = TableauSite.LiaisonSiteAdresse(0).Ville
Me.PaysSite.Text = TableauSite.LiaisonSiteAdresse(0).Pays
Me.NoTelSite.Text = TableauSite.LiaisonSiteAdresse(0).NoTel
Me.NoFaxSite.Text = TableauSite.LiaisonSiteAdresse(0).NoFax
Me.EmailSite.Text = TableauSite.LiaisonSiteAdresse(0).Email
Me.SiteWebSite.Text = TableauSite.LiaisonSiteAdresse(0).SiteWeb
Me.ContactSite.Text = TableauSite.LiaisonSiteAdresse(0).Contact
```

```
Me.Adr1Facturation.Text = TableauSite.LiaisonSiteAdresse(1).Adr1
Me.Adr2Facturation.Text = TableauSite.LiaisonSiteAdresse(1).Adr2
Me.CPFacturation.Text = TableauSite.LiaisonSiteAdresse(1).CP
Me.VilleFacturation.Text = TableauSite.LiaisonSiteAdresse(1).Ville
Me.PaysFacturation.Text = TableauSite.LiaisonSiteAdresse(1).Pays
Me.NoTelFacturation.Text = TableauSite.LiaisonSiteAdresse(1).NoTel
Me.NoFaxFacturation.Text = TableauSite.LiaisonSiteAdresse(1).NoFax
Me.EmailFacturation.Text = TableauSite.LiaisonSiteAdresse(1).Email
Me.SiteWebFacturation.Text = TableauSite.LiaisonSiteAdresse(1).SiteWeb
Me.ContactFacturation.Text = TableauSite.LiaisonSiteAdresse(1).Contact
```

```
Me.Adr1Livraison.Text = TableauSite.LiaisonSiteAdresse(2).Adr1
Me.Adr2Livraison.Text = TableauSite.LiaisonSiteAdresse(2).Adr2
Me.CPLivraison.Text = TableauSite.LiaisonSiteAdresse(2).CP
Me.VilleLivraison.Text = TableauSite.LiaisonSiteAdresse(2).Ville
Me.PaysLivraison.Text = TableauSite.LiaisonSiteAdresse(2).Pays
Me.NoTelLivraison.Text = TableauSite.LiaisonSiteAdresse(2).NoTel
Me.NoFaxLivraison.Text = TableauSite.LiaisonSiteAdresse(2).NoFax
Me.EmailLivraison.Text = TableauSite.LiaisonSiteAdresse(2).Email
Me.SiteWebLivraison.Text = TableauSite.LiaisonSiteAdresse(2).SiteWeb
Me.ContactLivraison.Text = TableauSite.LiaisonSiteAdresse(2).Contact
```

```
If NomSiteAbrege.Text <> "" And
(My.Computer.FileSystem.FileExists(TableauParametres.Parametres(0).Données(4).Adr &
"\ " & NomSiteAbrege.Text & ".jpg")) = True Then
```

```
    'TableauParametres.Parametres(0).Données(4).Adr & "\ " &
NomEntrepriseAbrege.Text & ".jpg")) Then
        LogoSite.Image =
Image.FromFile(TableauParametres.Parametres(0).Données(4).Adr & "\ " &
NomSiteAbrege.Text & ".jpg")
    End If
```

```
    ' Me.CheminLogo.Text = TableauSite.CheminLogo
Me.Description.Text = TableauSite.Description
Atelier.Rows.Clear()
For j = 0 To CInt(TableauSite.NbLiaison - 1)
    If TableauSite.NoSite = TableauSite.LiaisonSiteAtelier(j).NoSite Then
        For i = 0 To (TableauListeAtelier.NbLiaison - 1)
            If TableauSite.LiaisonSiteAtelier(j).NoAtelier =
TableauListeAtelier.LiaisonSiteAtelier(i).NoAtelier Then
                TableauSite.LiaisonSiteAtelier(j).NomAtelier =
TableauListeAtelier.LiaisonSiteAtelier(i).NomAtelier
                End If
        Next i
```

```

        Me.Atelier.Rows.Add(TableauSite.LiaisonSiteAtelier(j).NoAtelier,
TableauSite.LiaisonSiteAtelier(j).NomAtelier)

    End If
Next j

End Sub

Sub MAJAtelier()
    Dim i As Integer
    SelectAtelier()
    If Me.Atelier.Rows(Atelier.CurrentRow.Index).Cells(0).ColumnIndex = 0 Then

        If Me.Atelier.Rows(Atelier.CurrentRow.Index).Cells(0).Value <> 0 Then
            For i = 0 To (TableauListeAtelier.NbLiaison - 1)
                If Me.Atelier.Rows(Atelier.CurrentRow.Index).Cells(0).Value =
TableauListeAtelier.LiaisonSiteAtelier(i).NoAtelier Then

                    Me.Atelier.Rows(Atelier.CurrentRow.Index).Cells(1).Value =
TableauListeAtelier.LiaisonSiteAtelier(i).NomAtelier
                End If

                Next i

            End If

        End If
    End Sub

Sub EnregistrerAtelier(ByVal CleAttribuee)
    Dim j As Integer
    Dim k As Integer
    Dim typeSite As String
    Dim ValeurSite As String
    Dim i As Integer
    Dim Tableau() As Integer
    Dim Sp As String
    Dim ValeurInsert As String
    Dim TableauASupprimer() As Integer
    Dim TableauSauve(100) As Integer
    Dim Indice As Integer
    Dim indice2 As Integer
    Dim l As Integer
    Dim m As Integer
    Dim trouve As Boolean
    'sql = "SELECT NoLiaisonEntrepriseSite FROM LiaisonEntrepriseSite WHERE
NoEntreprise = " & Me.NoEntreprise.Text
    'NomTable = "LiaisonEntrepriseSite"

    ' Requete()
    Sp = "',''"
    If CleAttribuee <> "" Then
        typeSite = "NoSite"
        ValeurSite = CleAttribuee

        SelectSite2(typeSite, ValeurSite)
        indice2 = 0
        ReDim Tableau(100)
        ReDim TableauASupprimer(100)
        If Atelier.RowCount - 1 > TableauSite.NbLiaison - 1 Then
            indice2 = 0
            Indice = 0
            For i = 0 To Atelier.RowCount - 1

```



```

        End If
    Next m
Next l

If trouve = False Then
    Tableau(indice2) = Me.Atelier.Rows(j).Cells(0).Value
End If

End If

Next j
indice2 = indice2 + 1
Indice = Indice + 1

Next i

For k = 0 To indice2 - 1
    If Tableau(k) <> 0 Then
        ValeurInsert = Tableau(k) & Sp & CleAttribuee
        sql = "INSERT INTO LiaisonSiteAtelier (NoAtelier, NoSite)
VALUES ('" & ValeurInsert & "'"")"

        NomTable = "LiaisonSiteAtelier"

        Requete()
    End If
Next
For k = 0 To Indice - 1
    If TableauASupprimer(k) <> 0 Then
        sql = "DELETE FROM LiaisonSiteAtelier WHERE NoAtelier = " &
TableauASupprimer(k) & " AND NoSite = " & CleAttribuee
        NomTable = "LiaisonSiteAtelier"

        Requete()
    End If
Next k

End If

End If

```

```

' Dim k As Integer

' Dim Sp As String
' Dim ValeurInsert As String

' If CleAttribuee <> "" Then
'sql = "DELETE FROM LiaisonSiteAtelier WHERE NoSite = " & CleAttribuee
' NomTable = "LiaisonSiteAtelier"
' Requete()

'For k = 0 To Atelier.RowCount - 2

'Sp = "',''"
' ValeurInsert = Me.Atelier.Rows(k).Cells(0).Value & Sp & CleAttribuee
' sql = "INSERT INTO LiaisonSiteAtelier (NoAtelier, NoSite) VALUES ('" &
ValeurInsert & "'"")"

' NomTable = "LiaisonSiteAtelier"

' Requete()

' Next k
' End If

```

End Sub

Sub EnregistrementProcedure()

```

Dim i As Integer
Dim j As Integer
Dim CleAttribuee As String

```

```

If Me.NoSite.Text = "" Then
    sql = "SELECT NoSite FROM Site"
    NomTable = "Site"

    Requete()
    j = CInt(dtt.Rows(0).Item("NoSite"))
    CleAttribuee = CStr(j + 1)
    For i = 0 To dtt.Rows.Count - 1

        If CInt(dtt.Rows(i).Item("NoSite")) < j Then

            CleAttribuee = (dtt.Rows(i).Item("NoSite") + 1)
        End If

    Next i

Else
    CleAttribuee = Me.NoSite.Text

```

```
End If
```

```
sql = "SELECT NoSite FROM Site WHERE NoSite = " & CleAttribuee  
NomTable = "Entreprise"
```

```
Requete()
```

```
If Me.CPSite.Text = "" Then  
    Me.CPSite.Text = "0"
```

```
End If
```

```
If Me.CPLivraison.Text = "" Then  
    Me.CPLivraison.Text = "0"
```

```
End If
```

```
If Me.CPFacturation.Text = "" Then  
    Me.CPFacturation.Text = "0"
```

```
End If
```

```
If Me.NoDisponibilite.Text = "" Then  
    Me.NoDisponibilite.Text = "0"
```

```
End If
```

```
' If Me.CP.Text = "" Then  
' Me.CP.Text = "0"  
' End If
```

```
If dtt.Rows.Count = 0 Then  
    InserirSite(CleAttribuee)
```

```
Else  
    ModifierSite(CleAttribuee)
```

```
End If
```

```
EnregistrerAtelier(CleAttribuee)
```

```
Me.NoSite.Text = ""
```

```
Me.NomSite.Text = ""
```

```
ReinitialiserSaisieSite()
```

```
End Sub
```

```
Sub QuitterProcedure()
```

```
Dim res As Integer
```

```
Dim msg As String
```

```
If Me.NomSite.Text <> "" Then
```

```
    msg = TableauAffichage.GroupAffichage(2).Donnees(0).Libelle(NumLangueDem -
```

1)

```
    res = MsgBox(msg, MsgBoxStyle.YesNoCancel)
```

```
    Select Case res
```

```
        Case vbYes
```

```
            EnregistrementProcedure()
```

```
            Me.Hide()
```

```
        Case vbNo
```

```
            Me.NoSite.Text = ""
```

```
            Me.TrouverRapide.Text = ""
```

```
            ReinitialiserSaisieSite()
```

```
            Me.Hide()
```

```
        Case vbCancel
    End Select
Else
    Me.Hide()
End If
End Sub
```

```
Private Sub Label137_Click(sender As System.Object, e As System.EventArgs) Handles  
Label137.Click
```

```
End Sub  
End Class
```

SaisieCodeUtilisateur

```
Public Class SaisieCodeUtilisateur
```

```
    Private Sub Connexion_Click(sender As System.Object, e As System.EventArgs)
Handles Connexion.Click
    Dim res As Integer
    Dim ConnectionReussie As Boolean
    Utilisateur = ""
    SaisieAutorisee = 0
    GestionAutorisee = 0
    EditionsAurisees = 0
    StocksAutorises = 0
    SaisieEntreesAutorisee = 0
    SuperviseurReceptionAutorise = 0
    ConnectionEDIAutorisee = 0
    SuperviseurExpeditionsAutorise = 0
    superviseurProductionAutorise = 0
    VisuStatistiquesAutorise = 0
    ConnectionReussie = False

    For i = 0 To TableauOperateurs.NbGroup - 1
        'On redimensionne le tableau en fonction du nombre de langue
        '    ReDim TableauOperateurs.Operateurs(i).NomGroupOperateursL(NbLangue
- 1)
        ' For k = 0 To NbLangue - 1
        'On récupère les libellés des sections dans les différentes langues
        If NomUtilisateur.Text =
TableauOperateurs.Operateurs(i).NomGroupOperateursL(0) Then
            Utilisateur = NomUtilisateur.Text
            NoUtilisateur = i
            LangueUtilisateur =
TableauOperateurs.Operateurs(i).NomGroupOperateursL(1)
        End If
        'Next k
        'On récupère le type de la section : sécurisée par mot de passe ou non

        If MotDePasse.Text = TableauOperateurs.Operateurs(i).CodeSecret Then

            'On récupère le nombre de paramètres de la section
            '    TableauOperateurs.Operateurs(i).NbOperateursGroup =
LireINI(DIR_PARAM & FICH_PARAM, "Section" & CStr(i + 1), "NbCle")
            'On incrémente le nombre de paramètres total de la cellule
            'TableauParametres.NbParametres = TableauParametres.NbParametres +
TableauParametres.Parametres(i).NbParametreGroup

            'On balaie tous les paramètres de la section
            ' For j = 0 To TableauOperateurs.Operateurs(i).NbOperateursGroup - 1
            'On récupère l'adresse du paramètre
            SaisieAutorisee =
TableauOperateurs.Operateurs(NoUtilisateur).Donnees(0).Adr
            GestionAutorisee =
TableauOperateurs.Operateurs(NoUtilisateur).Donnees(1).Adr
            EditionsAurisees =
TableauOperateurs.Operateurs(NoUtilisateur).Donnees(2).Adr
```

```

        StocksAutorises =
TableauOperateurs.Operateurs(NoUtilisateur).Donnees(4).Adr
        SaisieEntreesAutorisee =
TableauOperateurs.Operateurs(NoUtilisateur).Donnees(5).Adr
        SuperviseurReceptionAutorise =
TableauOperateurs.Operateurs(NoUtilisateur).Donnees(6).Adr
        ConnectionEDIAutorisee =
TableauOperateurs.Operateurs(NoUtilisateur).Donnees(7).Adr
        SuperviseurExpeditionsAutorise =
TableauOperateurs.Operateurs(NoUtilisateur).Donnees(8).Adr
        superviseurProductionAutorise =
TableauOperateurs.Operateurs(NoUtilisateur).Donnees(9).Adr
        VisuStatistiquesAutorise =
TableauOperateurs.Operateurs(NoUtilisateur).Donnees(10).Adr
        NumLangueDem = LangueUtilisateur
        ConnectionReussie = True

    End If

        'On redimensionne le tableau en fonction du nombre de langue
        'ReDim TableauOperateurs.Operateurs(i).Donnees(j).Param(NbLangue - 1)
        ' For k = 0 To NbLangue - 1
        'On récupère le libellé du parametre dans les différentes langues
        'TableauOperateurs.Operateurs(i).Donnees(j).Param(k) = LineINI(DIR_PARAM &
FICH_PARAM, "Section" & CStr(i + 1), "L" & k + 1 & "NomCle" & CStr(j + 1))
        ' Next k
        ' If (TableauParametres.NbParametres <
TableauParametres.Parametres(i).Donnees(j).Adr) Then
        'TableauParametres.NbParametres =
TableauParametres.Parametres(i).Donnees(j).Adr
        ' End If

    Next i
    If ConnectionReussie = False Then
        res = MsgBox("Failure conncting User" & NomUtilisateur.Text,
MsgBoxStyle.OkCancel)
        Select Case res
            Case vbOK
                Me.NomUtilisateur.Text = ""
                Me.MotDePasse.Text = ""

                Me.Hide()

            Case vbCancel
                Me.NomUtilisateur.Text = ""
                Me.MotDePasse.Text = ""

        End Select
    Else
        Me.NomUtilisateur.Text = ""
        Me.MotDePasse.Text = ""

        Me.Hide()
    End If
End Sub
Private Sub SaisieCodeUtilisateur_Load(sender As System.Object, e As
System.EventArgs) Handles MyBase.Load

    End Sub
End Class

```

Programmation Orienté Objet

JAVA

C'est un exercice de programmation orientée objet Java que je vous propose-là. C'est un résumé des notions fondamentales (classes, méthodes, attributs, etc...) qui a été réuni. Ce n'est pas vraiment un cours mais l'essentiel expliqué par un exemple de programme : l'utilisation de la boîte à outil par un ouvrier.


```

Public Classe Outil {

Private String Nom ;
Private Utilisateur Ouvrier ;

Private boolean EstUtilise() {
    If (ouvrier == NULL)
        Return false ;
    Else return true ;
}

Public Date utilise(Utilisateur Uti){
    If (this.EstUtilise())
        Return null;
    Else
        Ouvrier = Uti;
        Uti.AjouteOutil(this);
        Return new Date() ;
}

Outil(String o){
    Nom = o ;
    Ouvrier = null ;
}

Void SetNom(string n){
    Nom = n ;
}

```

Constructeur

```

String GetNom(){
Return Nom ;
}

Public Class Utilisateur{
    Private Outil[] boiteoutil ;
    Private int nboutils ;

    Public Void ajouteoutil(outil
out){
        Boiteoutil[nboutils] = out ;
        Nboutils++ ;
    }

    Public Class Gererlaboite{
        Void ajouterdanslaboite(string
nom){
            Outil nouveloutil ;
            Nouveloutil = New outil(nom) ;
            Metdanslaboite(NouvelOutil) ;
            ...
        }
    }

    Public Class Pinceau extend Outil{
        Private string typepinceau();
    }
}

```

Instanciation de
la classe outil

ISBN : 978-2-900794-01-2
Cr   en France

Site : www.frederic-gilet.fr