



Université de Ghardaia

N°
d'ordre :

Faculté des Sciences et Technologies
Département des Sciences et Technologie

Mémoire présenté en vue de l'obtention du diplôme de

MASTER

Domaine : *Sciences et Technologies*

Filière : *Automatique*

Spécialité : *Automatique*

Par : Hammou AOUF

Thème

**Etude, Conception et Réalisation d'un Compteur de
Production et de Rendement Assisté par Ordinateur**

Soutenu publiquement le : 19/09/2017

Devant le jury :

CHOUIA Fayçal	MAA	Univ. Ghardaïa	Président
BENAOUICHA Karime	MAA	Univ. Ghardaïa	Examinateur
MOUAT Soufiane	MAA	Univ. Ghardaïa	Examinateur
BEKKAR Belgacem	MAA	Univ. Ghardaïa	Encadreur
SADOUNI Rdouane	MAA	Univ. Ghardaïa	Invité

Année universitaire 2016/2017

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Dédicace

*Je dédie ce modeste travail à ceux qui ont tout sacrifié
pour moi et ceux qui, sans eux, je n'aurais jamais pu en*

Arriver là aujourd'hui : mes chers parents

*A mes frères et à ma sœurs chacun à son nom à qui je souhaite beaucoup
de réussite dans la vie*

A ma femme et sa famille qui m'ont soutenu

A mes chers enfants

A mes oncles et chères tantes

A tout les membres des deux familles AOUF et NACER

A tout mes chers enseignants

A ceux qui ont été là pour moi et que j'ai oublié de les citer

Et à la mémoire de mon cher frère défunt

« Abdellghani » qu'Allah ait pitié de lui.

Hammou AOUF.



Remerciements

Avant tout nous devons remercier Allah le tout puissant de nous avoir accordé la patience et la force pour arriver au terme de ce modeste travail.

Tous mes gratitudes pour mon promoteur : Mr: B. BAKKAR qui m'aide par ses conseils et ses orientations.

Je tiens à remercier tous les enseignants de L'université de Ghardaïa.

Mes vifs remerciements aux responsables de l'entreprise ITBM pour leur accueil et leur confiance.

Je remercie très chaleureusement mon frère Djabre qui m'a incité à terminer ce travail.

Merci à toute personne ayant contribué de près ou de loin à l'élaboration de ce travail, que ce soit par son amitié, son encouragement ou ses conseils

...Hammou.

Résumé :

La Gestion de la Production assisté par Ordinateur (GPAO) est un outil indispensable pour le suivi et l'amélioration de la production des entreprises, le challenge c'est de mettre à jour la base de données de cette application, ce travail propose une solution pratique comporte le développement de compteurs basés sur un système micro-programmé fonctionne avec le microcontrôleur PIC18F45K22, ces compteurs seront embarqués sur les machines et liés avec l'ordinateur de gestion via un réseau RS485.

Le système est géré par une application développée avec Microsoft Visual Studio, qui permet au gestionnaire de communiquer avec les compteurs et de sauvegarder les données dans une base de données.

Ce mémoire présente en détail les outils et les méthodes utilisées pour développer et réaliser la solution proposée.

Mots Clés :

GPAO, Base de données, Microcontrôleur, PIC 18F45K22, Entrée numérique, Sortie Numérique, Conception, Circuit imprimé, PCB, EasyEDA, Relais, VDC, Alimentation, Programmation, FLOWCODE, Flashage, PICKIT2, communication, Réseau RS485, Max 485, Développement, Application, Microsoft Visual Studio, Access.

ملخص :

إدارة الإنتاج بمساعدة الحاسوب (GPAO) أداة لا غنى عنها لرصد وتحسين إنتاج الشركات، والتحدي هو تحديث قاعدة البيانات لهذا التطبيق، ويقترح هذا العمل حلاً تطبيقياً للمساعدة في هذه العملية، ويشمل على تطوير عدادات على أساس نظام ميرمج صغير يعمل ب PIC18F45K22، وسوف تتركب هذه العدادات على الآلات ويتم ربطها مع كمبيوتر الإدارة عبر شبكة RS485. يدار النظام بواسطة تطبيق مطور ب: ميكروسوفت فيسوال ستوديو، والذي يسمح للمستخدم التواصل مع العدادات وحفظ البيانات في قاعدة بيانات.

ويوضح هذا العمل الأدوات والأساليب المستخدمة لتطوير وتنفيذ الحل المقترح.

كلمات مفتاحية :

إدارة الإنتاج بالحاسوب، قاعدة بيانات، رقاقة التحكم، حاسوب، نظام، برنامج، تطوير، دائرة كهربائية، برمجة، شبكة الاتصال، ميكروسوفت، ترقية.

Abstract :

Computer Assisted Production Management (CAPM or GPAO in French) is an essential tool for monitoring and improving the production in companies, the challenge is to update the database of this application and this work proposes a practical solution involves development of Counter based on a micro-programmed system works on PIC18F45K22 microcontroller.

Each counter will be embedded on a machine and the series of counters are linked with the managing computer via an RS485 network.

The system is managed by an application developed with Microsoft Visual Studio, which allows to the manager to communicate with the counters and save the data in a database.

This thesis presents in detail the tools and the methods used to develop and implement the proposed solution.

Keywords :

GPAO, Database, Microcontroller, PIC 18F45K22, Digital Input, Digital Output, Design, PCB, PCB, EasyEDA, Relay, VDC, Power Supply, Programming, FLOWCODE, Flashing, PICKIT2, Communication, RS485 Network, Max 485, Development, Application, Microsoft Visual Studio, Access.

Table des matières

Titre :	Page :
Résumé :	ii
Table des matières	iv
Liste des figures.....	viii
Liste des tableaux	xi
Liste des abréviations	xii
Introduction Générale	1
Chapitre 1 : Généralité sur la GPAO et Présentation De Problématique.....	3
1.1 Introduction.....	3
1.2 Présentation système GPAO (Gestion de la Production Assistée par Ordinateur) ...	3
1.2.1 GPAO ET ERP	3
1.3 Gestion de production : les différentes approches	4
1.4 Organisation du système de Gestion de Production.....	5
1.4.1 Les Grands Types De Gestion De Production.....	5
1.4.2 Exemple Des Systèmes Existants :	6
1.5 Implémentation dans le terrain industriel.....	7
1.5.1 Facteurs Clés De Succès d'une GPAO :	8
1.6 Identification de la problématique :	9
1.7 Conclusion	10
Chapitre 2 : Les Systèmes micro-programmé à base de microcontrôleurs.....	12
2.1 Introduction.....	12
2.2 Généralités sur les chaines d'acquisition des données.....	13
2.3 Architecture d'une chaine d'acquisition de données :	13
2.3.1 Les capteurs :	14
2.3.2 Caractéristiques déterminantes dans le choix d'un capteur :	14

2.3.3	Conditionnement des signaux.....	15
2.3.4	Conversion analogique/numérique :	17
2.4	Modèle de base d'un microcontrôleur :	18
2.4.1	L'architecture de Von Neumann :	19
2.4.2	L'architecture de type Harvard :	20
2.4.3	Rôle d'un système à microcontrôleur :	20
2.4.4	Avantages :	20
2.4.5	Inconvénients :	20
2.5	Le bus d'interface I2C (Inter-Integrated Circuit).....	21
2.5.1	Propriétés physiques du bus I2C.....	21
2.5.2	Le Protocole :	22
2.6	Le module RTC: Real Time clock	25
2.6.1	Exemple d'écriture du DS1307	25
2.6.2	Exemple de lecture du DS1307	26
Les réseaux de communication (RS485) :		28
2.6.3	Constitution d'une liaison	28
2.6.4	Caractéristiques électriques du bus RS485.....	30
2.6.5	Règles de câblage :	31
2.7	Les protocoles de communication utilisés dans le réseau RS485 :	32
2.7.1	Structure des messages :	33
2.7.2	Adressage.....	34
2.7.3	Echange du maître avec un esclave	34
2.7.4	Echange Maître vers tous les esclaves.....	34
2.7.5	Format général d'une trame.....	34
2.7.6	Trame d'échange question/réponse	35
2.8	L'Interface Homme-Machines (IHM) :	36
2.8.1	Les différentes techniques	36

2.8.2	Les périphériques IHM	36
2.9	Conclusion :	38
Chapitre 3 : La Conception Du Système (Hardware & software)		40
3.1	Introduction :	40
3.2	Définition de cahier de charges :	40
3.2.1	Cahier de charges du compteur :	40
3.2.2	Cahier de charges du logiciel(IHM) :	41
3.2.3	Cahier de charges du réseau d'alimentation et de communication :	42
3.3	Recensement des entrées sorties du système :	42
3.1	L'algorithme d'acquisition et de traitement au niveau de la machine :	42
3.2	Choix de protocole de communication :	45
3.2.1	Opération de lecture d'un octet de l'espace mémoire EEPROM :	45
3.2.2	Opération d'écriture d'un octet dans un espace mémoire EEPROM :	46
3.3	Choix de microcontrôleur :	47
3.3.1	Récapitulatif des exigences sur le microcontrôleur	47
3.3.2	Caractéristiques du PIC 18F45K22 :	48
3.4	Conception du système d'acquisition (Compteur) :	49
3.4.1	Alimentation :	49
3.4.2	Les Entrées Numériques (Digital Inputs) :	50
3.4.3	Les Sorties Numériques (Digital Outputs) :	51
3.4.4	Afficheur Alphanumérique LCD 2x16 :	53
3.4.5	Clavier matriciel 4x4 (16 touches) :	54
3.4.6	Module d'horloge (Real Time Circuit –RTC-):	55
3.4.7	Communication avec le Host (PC) :	55
3.5	Le circuit global du système d'acquisition (Compteur) :	56
3.6	Conception d'une application d'interfaçage Homme – Machine IHM :	57
3.6.1	Configuration Manuel :	57

3.6.2	La communication et l'actualisation des données coté PC :.....	58
3.7	Conclusion :	60
Chapitre 4 :	Réalisation Et Mise En Service :.....	62
4.1	Les outils utilisés pour la conception des circuits électroniques :	62
4.2	Outils et méthodes de conception des circuits imprimés (PCB) :.....	63
4.3	Développement du programme pour le microcontrôleur :.....	67
4.4	Simulation de programme et génération fichier HEX :	70
4.5	Flashage et programmation du microcontrôleur :	71
4.6	Développement de l'interface IHM :	73
4.6.1	Microsoft Visual Studio (MVS):	74
4.6.2	Développement des interfaces graphiques :	74
4.6.3	Programme de communication est de mise à jour	75
4.6.4	Base de données :.....	77
4.7	Essais et mise en service :	78
4.8	Conclusion :	79
Chapitre 5 :	Conclusion Générale	81
	Bibliographie :	82

Liste des figures

Figure :	Page :
Figure 1-1 : Schéma Descriptif de l'ERP et la GPAO.	4
Figure 1-2 : Schéma Descriptif du projet.	10
Figure 2-1 : Organisation fonctionnelle d'un Système à Microcontrôleur.	13
Figure 2-2 : Schéma de la chaîne d'acquisition d'une mesure.	14
Figure 2-3 : Etendue de mesure d'un capteur.	15
Figure 2-4 : Etendue Domaine de linéarité d'un capteur.	15
Figure 2-5 : Capteur générant un courant associé à un convertisseur courant-tension à amplificateur opérationnel.	16
Figure 2-6 : Echantillonnages d'un signal de mesure (a) avec une fréquence d'échantillonnage plus élevée dans le cas (b) que le cas (c), $f_{eb} > f_{ec}$	17
Figure 2-7 : Construction typique d'un signal numérique.	18
Figure 2-8 : Architecture interne du microcontrôleur.	19
Figure 2-9 : Architecture VON NEUMAN	19
Figure 2-10 : Architecture Harvard	20
Figure 2-11 : Synoptique d'une liaison I2C.	21
Figure 2-12 : les signaux d'un bus I2C.	21
Figure 2-13 : caractéristiques du bus I2C.	22
Figure 2-14 : Adressage des esclaves en Bus I2C.	23
Figure 2-15 : Début de la communication en I2C.	23
Figure 2-16 : Transmission d'une adresse en I2C.	23
Figure 2-17 : Confirmation de réception d'adresse en I2C.	24
Figure 2-18 : Chronologie d'écriture de données sur I2C.	24
Figure 2-19 : Chronologie de lecture de données sur I2C.	24
Figure 2-20 : Fin de la communication en I2C.	25
Figure 2-21: Redémarrage de la communication.	25
Figure 2-22: Exemple d'écriture de données sur le DS1307 via I2C.	25
Figure 2-23: Exemple de lecture de données de DS1307 via I2C.	26
Figure 2-24 : Chronogramme complet de l'échange avec l'esclave DS1307 sur I2C.	26
Figure 2-25 : Eléments d'une liaison entre systèmes informatiques.	28
Figure 2-26 : Circuit de jonction équivalent d'un bus RS485.	29

Figure 2-27 : Niveaux significatifs de la norme	30
Figure 2-28 A : Architecture générale d'un réseau RS 485 2 fils	30
Figure 2-29 B : Architecture générale d'un réseau RS 485 4 fils.....	31
Figure 2-30 : Relation entre la longueur et la vitesse du réseau RS485.....	31
Figure 2-31 : Type des câbles à utiliser dans le réseau RS485.....	32
Figure 2-32 : Polarisation et résistance de fin de ligne RS-485.	32
Figure 2-33 : Architecteur hiérarchique dans le réseau RS-485.....	33
Figure 2-34 : Structure des messages « protocole standard ».....	33
Figure 2-35 : Echange du maître avec un esclave	34
Figure 2-36 : Echange maître vers tous les esclaves	34
Figure 2-37 : Format d'une trame en mode RTU.....	35
Figure 2-38 : Format d'une trame en mode ASCII.	35
Figure 2-39 : schéma descriptif des composantes d'un IHM.	36
Figure 2-40 A : Architecture fondamentale :.....	37
Figure 2-41 : Architecture de Seeheim.....	37
Figure 2-42 : Architecture de lArch :	37
Figure 3-1 : Organigramme général de fonctionnement du compteur.	44
Figure 3-2 : Format d'un message de lecteur.	45
Figure 3-3 : Format d'un message d'écriture.	46
Figure 3-4 : Signaux du Microcontrôleur PIC18F45K22.....	48
Figure 3-5 : Circuit d'alimentation.....	49
Figure 3-6 : Résistances de tirage « Pull Down Resistor ».....	50
Figure 3-7 : Circuit de conditionnement des entrées (Digital inputs).	51
Figure 3-8 : Caractéristiques électrique des LED.....	52
Figure 3-9 : Circuit sorties numérique (commande LED).....	52
Figure 3-10 : Circuit sorties numérique (commande Relais)	52
Figure 3-11 : Circuit d'interconnexion de l'afficheur LCD 2x16 avec le μ C.	53
Figure 3-12 : Clavier matriciel 4X4.	54
Figure 3-13 : Circuit d'interconnexion du Clavier 4x4 avec le μ C.	54
Figure 3-14 : Circuit d'interconnexion du module RTC avec le μ C via I2C.....	55
Figure 3-15 : Circuit de communication MAX485	55
Figure 3-16 : Convertisseur USB –RS485.	55
Figure 3-17 : Circuit global du compteur.....	56
Figure 3-18 : Gestionnaire de périphériques.	57

Figure 3-19 : interface de configuration d'un port série.....	58
Figure 3-20 : Organigramme de communication et d'actualisation des données coté PC...	59
Figure 4-1 : Interface d'accueil de l'application EasyEDA.....	62
Figure 4-2 : Espace de conception des circuits de EasyEDA.....	63
Figure 4-3 : Menue de conversion d'un schéma à un circuit imprimé.	63
Figure 4-4 : Espace d'édition des circuits imprimés.	64
Figure 4-5 : Schéma final du circuit imprimé.....	65
Figure 4-6 : Le circuit imprimé double faces.	65
Figure 4-7 : Compteur Semifini.....	66
Figure 4-8 : Espace de développement de l'application FLOWCODE.....	67
Figure 4-9 : Module Clavier de la bibliothèque de FLOWCODE.	68
Figure 4-10 : Module LCD de la bibliothèque de FLOWCODE.	68
Figure 4-11 : Boutons de la bibliothèque de FLOWCODE.	68
Figure 4-12 : LED de la bibliothèque de FLOWCODE.....	69
Figure 4-13 : Module I2C de la bibliothèque de FLOWCODE.	69
Figure 4-14 : Lancement de la simulation dans FLOWCODE.	70
Figure 4-15 : Kit de flashage PICKIT2.	71
Figure 4-16 : PIC18F45K22 placé dans le socle de flashage.....	72
Figure 4-17 : Importation du fichier HEX pour flasher le PIC18F45K22.	72
Figure 4-18 : Lancement de flashage du PIC18F45K22.	73
Figure 4-19 : Espace de développement de Microsoft Visual Studio.	74
Figure 4-20 : la forme principale de l'application.....	75
Figure 4-21 : Forme secondaire de l'application.....	75
Figure 4-22 : le composant de communication de MVS.....	76
Figure 4-23 : Schéma de montage de l'ensemble du système.....	78

Liste des tableaux

Tableau :	Page :
Tableau 1-1 : Récapitulatif entre Les différents types de la GPAO	7
Tableau 2-1 : Caractéristiques du réseau RS485.....	29
Tableau 3-1 : Liste des entrées et des sorties du microcontrôleur.....	42
Tableau 3-2 : Adressage des variables dans l'EEPROM.	46
Tableau 3-3 : Tableau récapitulatif des caractéristiques du μ C à choisir.....	47
Tableau 3-4 : les entrées numériques.	50
Tableau 3-5 : Les sorties numériques.....	51
Tableau 4-1 : structure de la base de données.	77

Liste des abréviations

GPAO : Gestion de la Production Assistée par Ordinateur.

ERP: Enterprise Resource Planning.

MRP: Manufacturing Resources Planning

OF : Ordre de Fabrication

μC : Microcontrôleur.

CPU : (Central Processing Unit) Unité de calcul centrale

LCD : Affichage à cristaux liquides

EEPROM : Electrically-Erasable Programmable Read-Only Memory ou mémoire morte effaçable électriquement et programmable

RAM : mémoire morte de donnée

ROM : mémoire programme.

CAN : Convertisseur Analogique Numérique

CNA : Convertisseur Numérique Analogique

E.M : Etendue de mesure

m: mesure

S: Sensibilité

I2C: (Inter-Integrated Circuit)

SCL : (Serial Clock) : horloge de cadencement de la communication.

SDA : (Serial Data) : signal de données

MSB : (most significant bit) le bit de gauche dans la représentation d'un Octet.

LSB : (least significant bit) le bit de droit dans la représentation d'un Octet.

NL: Niveau logique

ACK: Acquiescement

ETTD : Equipement Terminal de Traitement de données

ETCD : Equipement de Terminaison de Circuit de Données

RS485: Réseau de communication Industriel à deux fils

RTC: Real Time clock

IHM : L'Interface Homme-Machines

PCB : (Printed circuit board) circuit imprimé

PC (Host) : Ordinateur de de commandement et de gestion.

LAN : Local Area Network

Introduction Générale

Dans un marché ouvert où les petites entreprises cherchent leurs places et les grandes entreprises luttent à garder leurs parts et avec les circonstances de la crise économique l'enjeu devient grand.

La capacité de connaître précisément les marges de manœuvre économique et de maîtriser ses processus logistiques et productifs déterminent donc la capacité de prendre des marchés, de les conserver et de soigner la relation avec ses clients. La **GPAO** (Gestion de la Production Assistée par Ordinateur) est une des briques essentielles dans la maîtrise de cette capacité.

En cours de cette thèse nous présentons en quoi consiste la gestion de production assistée par ordinateur et précisément la gestion des commandes et le rendement des agents producteurs lors de la production à l'aide des compteurs de production et de rendement embarqués sur les machines et liés au bureau de gestion par un réseau RS485.

Ce compteur est développé à la base d'un microcontrôleur fabriqué pour ce genre de rôles (MICROCHIP) et l'application d'interfaçage est développée à la base du logiciel puissant et reconnu (Microsoft Visual studio Community)

Suite à des simulations et des tests réels dans des conditions de production réelles et après plusieurs améliorations le compteur a donné des très bons résultats.

Ce mémoire est structuré comme suit :

Partie Théorique : cette partie est composée de deux chapitres, dans le premier Chapitre nous donnons une présentation générale du système de gestion et leurs différentes approches avec présentation de la problématique.

Le deuxième chapitre est une présentation générale des systèmes micro-programmés à base de microcontrôleur et les détails d'une chaîne d'acquisition et les méthodes de communication utilisées dans ce projet.

Partie Etude et conception : cette partie est détaillée dans le troisième chapitre, qui présente les aspects techniques de l'acquisition et de traitement des données au niveau du compteur et au niveau de l'ordinateur Master clôturée par le circuit global du compteur.

Partie Réalisation, essais et Mise en service : dans le quatrième chapitre nous démontrons les étapes suivies pour réaliser la plaque PCB, la programmation et le flashage du microcontrôleur, l'élaboration du programme, les essais et le résultat final de notre travail

CHAPITRE 01

Généralité sur la GPAO

Et

**Présentation De
La Problématique**

Chapitre 1 : Généralité sur la GPAO et Présentation De Problématique.

1.1 Introduction

La maîtriser des modèles économiques et leur apport de valeur ajoutée est une des compétences clés d'une entreprise forte et organisée.

Dans les entreprises industrielles les progiciels de GPAO sont très répandus. Ils font partie du système d'information de l'entreprise et participent à l'amélioration de la compétitivité. Ils permettent aussi de quantifier et de planifier les besoins en composants ainsi que les ordres de fabrication. Des progiciels de gestion intégrés remplissent maintenant les fonctions de la GPAO en intégrant aussi d'autres applications telles que la gestion commerciale, la comptabilité et d'autres modules. Ils facilitent les échanges d'information entre les différentes parties de l'entreprise et comme ils sont modulaires ils sont évolutifs et adaptables

1.2 Présentation système GPAO (Gestion de la Production Assistée par Ordinateur)

Un logiciel de GPAO est un programme de gestion permettant de piloter l'ensemble des activités liées à la production. Il est généralement organisé par modules :

Gestion des stocks et des achats

Gestion de commandes

Gestion des produits engendrés par ces commandes

Gestion des articles entrant dans la fabrication de ces produits et de leurs nomenclatures-gammes, Expédition des produits et la Facturation

Ce type de progiciel est utilisé par des entreprises qui fabriquent, assemblent, ou modifient des produits ou des installations pour leur apporter de la valeur ajoutée. Il s'agit d'entreprises purement industrielles ou d'organisations fonctionnant en mode projet et en gestion à l'affaire.

1.2.1 GPAO ET ERP

La GPAO se distingue de l'ERP (Enterprise Resource Planning) par un fait essentiel : la comptabilité générale et analytique. L'ERP (Enterprise Resource Planning) est un outil qui intègre la totalité des fonctions de l'entreprise, qu'elles soient des fonctions financières (comptabilité générale, auxiliaire, analytique, budgétaire), des fonctions de gestion du personnel (paie, formation) et des fonctions logistiques (gestion des ventes, des achats, de la production, de la maintenance, de la logistique...).

Cette couverture fonctionnelle moins large n'est pas forcément un problème. Toutes les GPAO du marché savent s'intégrer avec des logiciels de comptabilité standard ou des logiciels de paie.

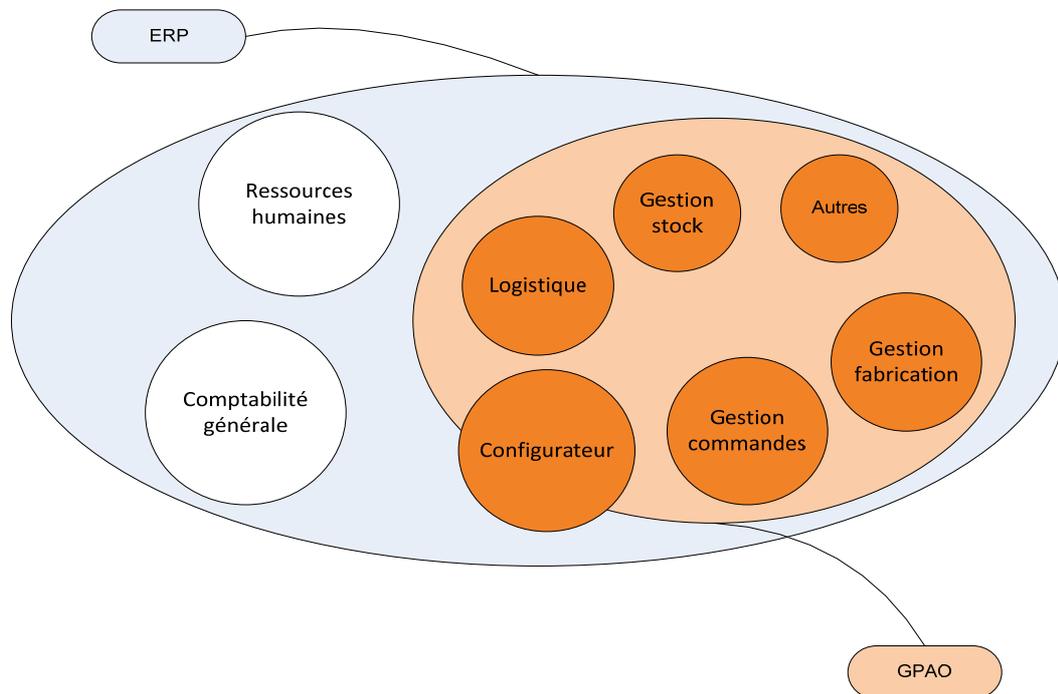


Figure 1-1 : Schéma Descriptif de l'ERP et la GPAO.

1.3 Gestion de production : les différentes approches

La GPAO permet de piloter l'apport de valeur ajoutée dans l'entreprise. Son objectif principal est de connaître de façon précise le coût de revient industriel, de connaître les niveaux de stocks (matière première, produits finis et encours) et d'approvisionner les lignes de production avec le bon composant au bon moment.

MAITRISER LES STOCKS

La GPAO permet de calculer les besoins de l'entreprise en temps réel et d'envisager précisément dans quelle mesure :

Elle peut encore fabriquer des produits

Elle devra s'approvisionner en matière première pour répondre aux commandes

Cette gestion des stocks repose sur la maîtrise des données de base de la gestion industrielle : les nomenclatures, les gammes et la description de la structure industrielle (sections, centre de charges, poste de charge, ressources, calendrier...)

MAITRISER LES COÛTS DE REVIENT

Dans un contexte économique où les coûts des matières fluctuent parfois beaucoup, la hausse du coût de l'énergie et la raréfaction des capacités d'investissement impactent la

rentabilité des entreprises. La marge brute des entreprises étant faible (en règle générale) dans l'entreprise industrielle, il est très rapide de perdre de l'argent sur des productions. La maîtrise des coûts de revient apparaît donc impérative.

La comptabilité industrielle est donc un des pivots de la GPAO.

MAITRISER LA PRODUCTION

Une GPAO est le support :

Des processus de maîtrise de la demande (client)

Des processus de maîtrise des achats / approvisionnements

Des processus de planification

De calcul des besoins

De lancement-ordonnancement

De suivi de la production

De traçabilité industrielle

De déclaration de production

Ces processus livrent à leur tour les données qui sont ensuite utilisée pour le pilotage des temps de cycle (cadences standards, cadences réelles, analyse des dérives ...).

Remarque :

Des outils spécifiques et des accessoires permettent aujourd'hui de suivre la production tout en minimisant les saisies et les surcoûts : le RFID, les douchettes code à barre, les terminaux d'ateliers et dans notre cas le compteur développé permis à relever la production à distance.

1.4 Organisation du système de Gestion de Production.

1.4.1 Les Grands Types De Gestion De Production

Il existe trois grands types de production selon le degré de standardisation et le volume de produits à fabriquer sont :

Production a l'Affaire :

La conception de machines spéciales, la livraison de produits uniques (usines clés en main) ou de prestation de service particulière (maintenance d'équipements industriels) amènent des modes d'organisation très différents des modes de production en série. Plus proche de la gestion par projets que de la gestion industrielle, ces modes de production sont donc soutenus par des GPAO particulières spécialisées dans ces types d'industrie

La Production A La Commande Et Petite Série :

Un peu plus soutenus, les rythmes de production à la commande sont particuliers. Il s'agit par exemple de la fabrication de composants modulaires (fenêtres, portes, cuisines). Ce mode

de production dans lequel le produit n'est déterminé finalement qu'assez tard dans la chaîne de production demande une GPAO capable de répondre de façon intelligente à la spécialisation retardée et à la configuration des produits à la commande. Le focus sera donc mis sur la capacité de maintenir de façon adaptée des données de base qui ne sont pas forcément statiques et répétitives.

La Production De Masse :

Les entreprises industrielles produisent également des séries plus ou moins importantes de produits dont le procédé de fabrication est plus ou moins répétitif. De façon générale, la taille des séries diminue. Aujourd'hui, la capacité de réactivité, le changement de séries, et la modularité de l'équipement industriel sont bien souvent un facteur clé de succès. Dans ce monde de la série, il faut encore considérer des facteurs de différenciation : un logiciel de GPAO ne conviendra pas de la même façon à un industriel dans la plasturgie, dans la mécanique ou dans l'agro alimentaire.

1.4.2 Exemple Des Systèmes Existants :

Les progiciels de GPAO se distinguent les uns des autres par le système qu'il propose. Il existe plusieurs modèles de gestion de la production selon le type de fabrication souhaitée.

Les progiciels classiques de gestion de production (MRP, MRP II)

Les **MRP** (Manufacturing Resources Planning) ont l'avantage d'avoir évolué vers un outil de gestion globale de l'entreprise (gestion commerciale, plan directeur, gestion des matières et des charges (de travail), calcul des coûts de revient, interfaçages avec les comptabilités...). Ils répondent aux principes de flux poussé et de production anticipée, nécessitant des prévisions. Ils calculent, en fonction des besoins et des délais d'obtention (temps de cycle ou délai d'approvisionnement), les Ordres de Fabrication et les Ordres d'Achat à lancer et les dates de lancement.

- Les progiciels de gestion d'atelier et/ou d'ordonnancement ne gèrent que les charges, le planning, lesancements, le suivi de fabrication. Ils permettent l'ordonnancement des opérations sur les machines, en optimisant les temps de cycles. Ils ne gèrent pas globalement l'entreprise et doivent donc être adjoints à une gestion commerciale ou une GPAO.
- Les progiciels de gestion Juste A Temps

Ils sont souvent des modules de GPAO complètes qui gèrent les productions répétitives. Ils ne travaillent pas par ordre de fabrication, mais par programme de fabrication journalier. Les flux dans l'atelier sont souvent contrôlés par l'utilisation de Kanbans.

Les progiciels de gestion à l'affaire

Ils sont adaptés à la gestion de fabrication d'un objet unique, comprenant un grand nombre de tâches. Le problème est le calcul du délai obtenu par le chemin critique, et l'attribution des ressources aux différentes tâches. Il ne faut pas confondre progiciel de gestion de projet et progiciel de gestion d'atelier. En effet, ces derniers incorporent l'environnement nécessaire (notion de gammes, îlots, cellules, machines...) et les algorithmes de placement (optimisation en fonction de critères tels que retard mini, charge maxi...), ce qui n'est pas le cas des progiciels de gestion de projet.

Les différences majeures entre les types suscités sont récapitulées dans le **Tableau 1-1**

Le MRP privilégie la gestion des matières, la gestion d'atelier privilégie la gestion des charges.

Les sous-fabricants de produits propres à nomenclatures complexes utilisent plutôt le MRP.

Les constructions unitaires utiliseront la gestion de projet.

Tableau 1-1 : Récapitulatif entre Les différents types de la GPAO

Typologie de Production	Mode de lancement	Type de commande	Type de système
- Produits «catalogue» - Produits complexes et personnalisés avec nomenclatures	- OF composants communs planifiés - OF assemblage à la commande	Aléatoires ou prévisibles par famille de produits	MRP II
Produits stables et répétitifs type automobiles	Programme journalier	Commandes ouvertes	JAT KANBAN
Produits +/- complexes fabriqués à la commande	Par lots ou unitaire. Lancement à la commande	Aléatoires et ponctuelles	GESTION D'ATELIER
Produits unitaires fabriqués sur spécification	Unitaire à la commande	Aléatoires et ponctuelles	MULTI-PROJET

1.5 Implémentation dans le terrain industriel

Le marché propose deux principales catégories de logiciels :

Des grandes GPAO génériques le plus souvent transformées de façon plus ou moins heureuse en ERP.

Des GPAO très verticales et spécialisées dans des secteurs d'activité précis. Elles sont portées par des éditeurs et éditeurs-intégrateurs de taille plus modeste.

1.5.1 Facteurs Clés De Succès d'une GPAO :

Voici quelques bonnes pratiques pour intégrer une GPAO. Premier conseil : l'intégration d'une GPAO impacte l'organisation et les façons de travailler de l'entreprise. Il est donc fortement conseillé de se faire accompagner par un consultant spécialisé dans le domaine.

L'expression des besoins

C'est pour beaucoup à ce stade que se joue la simplification ou la complexité de la solution. Un bon consultant aidera l'entreprise à distinguer l'indispensable, l'utile et le superflu.

La rédaction d'un cahier des charges «progiciel»

Il ne s'agit pas d'un cahier des charges englobe les besoins des utilisateurs et les performances attendues

L'existence d'un contrat

Il doit bien identifier les responsabilités, les limites de fourniture, les conditions de désengagement (en particulier la possibilité de récupérer les sources si le fournisseur est un petit développeur, la possibilité de récupérer les données en cas de changement de progiciel...).

La mise en concurrence de fournisseurs

Cela permet d'éviter d'être séduit par le premier vendeur venu, et pour comparer solutions, prix et services.

Les conditions de recette du progiciel

Elles doivent être prévues au contrat, par exemple avec un jeu d'essai préalablement mis au point, ou pour vérifier l'interfaçage avec la comptabilité ou d'autres logiciels utilisés par l'entreprise.

Maintenance et formation

Il est important de prendre en compte dès le départ la maintenance du progiciel (et du matériel) ainsi que la formation des utilisateurs. Un bon outil c'est surtout un outil bien utilisé.

Exploitation et évolution

La pertinence d'une GPAO se mesure sur la qualité des données de base de gestion de production : gammes/nomenclatures/ressources/postes de charges... La réussite du projet dans le temps dépendra donc également de la capacité de l'entreprise à gérer de façon rigoureuse l'ensemble de ces informations.

1.6 Identification de la problématique :

Il existe plusieurs méthodes pour alimenter la base de données d'un progiciel, il y a la saisie manuelle, les douchettes à code à barre, RF ID, ...etc.

Dans ce projet nous présentons une solution propre qui facilite le suivi aux responsables de la production, tout en réduisant le risque des erreurs humains, basé sur la conception et la réalisation des compteurs déportés avec des entrées et des sorties liées à la machine sur laquelle le compteur est déporté, tous les compteurs sont liés entre eux et à l'ordinateur de gestion sous un réseau industriel de type RS485. La communication est géré par un Protocole développé pour ce projet.

Le compteur est composé d'un afficheur (LCD 2x16), un clavier (4X3), quelques boutons et des LED pour afficher l'état du compteur.

Le gestionnaire envoie la commande à fabrication (OF) de son ordinateur vers le compteur de la machine ou par une insertion locale via l'entrée d'un code.

Le compteur demande à l'agent de faire entrer son code et le code de son équipe, et dès que les codes sont validés le compteur enregistre sur sa mémoire (EEPROM) le temps de démarrage et la date récupérée du module d'horloge (RTC) en suite il libère la machine pour commencer la production.

Le compteur récupère l'état de la machine, et au temps que la machine est en service le compteur incrémente le temps de travail de l'agent.

La production est suivie par le compteur via un détecteur de proximité, au moment où la quantité du produit ciblée est atteinte le compteur bloque la machine.

Lors de changement des agents et le compteur bloque la machine enregistre l'heure et le compteur de fin de production.

Le compteur redemande le code d'agent et le code de l'équipe pour recommencer un autre cycle de production.

L'ordinateur Master adresse chaque compteur dans la boucle pour récupérer les informations enregistrées sur leur mémoire.

La **Figure 1-2** représente un schéma descriptif de l'ensemble de la solution.

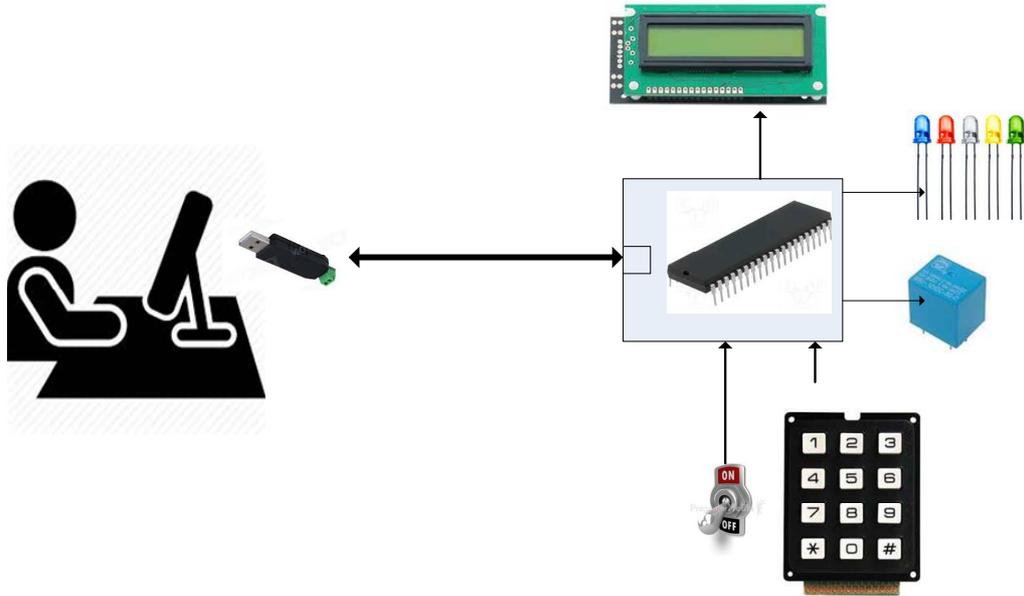


Figure 1-2 : Schéma Descriptif du projet.

1.7 Conclusion

La gestion de la production assistée par Ordinateur (GPAO) que se soit intégré dans un ERP ou comme une application autonome est un outil indispensable pour le suivi et la gestion dans les entreprises de moyenne et de grande production, ce progiciel permis aux gestionnaires de savoir à tous moment la situation des chaines de production.

La base de données de ces progiciels est mise à jour généralement par des saisies manuelles avec un grand risque d'erreur, la solution proposée dans ce projet facilite et automatise la récolte des données pour démunir le risque d'erreur et augmenter le suivi de la production en temps réel.

CHAPITRE 02

**Les Systèmes micro-programmé à
base de microcontrôleurs.**

Chapitre 2 : Les Systèmes micro-programmé à base de microcontrôleurs.

2.1 Introduction

Les systèmes à microprocesseurs ou microcontrôleurs sont omniprésents dans la vie de tous les jours. On les rencontre dans des systèmes complexes (ordinateurs individuels) mais également dans des systèmes simples (machine à laver, pompe à essence, store automatique, ...) où leur grande souplesse est appréciée

Le développement de l'électronique numérique a suscité l'apparition de plusieurs types de composants très puissants en particulier les systèmes micro-programmés.

Leur aptitude à s'adapter aux contraintes technologiques de plus en plus complexes, leur capacité de gérer un grand nombre de fonctionnalités variées et leur coût de revient faible a encouragé leur utilisation dans plusieurs applications tant domestiques qu'industrielles.

Le développement de ces composants programmables, qui ont remplacé l'électronique classique vu que les circuits intégrés analogiques ou logiques ne peuvent plus résoudre des fonctions de plus en plus complexes.

Historiquement, les constructeurs ont développé d'abord les systèmes micro-programmés intégrés dans les calculateurs de bureau ou de poche, avec des codes d'ordre orientés vers le calcul numérique. Puis maîtrisant cette technique ils offrirent des circuits d'usage généraux : les microprocesseurs.

Les systèmes micro-programmés ont vu leur importance progresser au rythme de l'importance prise par les microprocesseurs. C'est d'ailleurs l'élément de base pour de tels systèmes.

==> 1971 : premier microprocesseur 4 bits 4004 d'Intel.

La miniaturisation des transistors a permis d'augmenter considérablement la capacité d'intégration sur silicium. On est passé rapidement du processeur 4 bits au processeur de 8 bits 16 bits, 32 bits et 64 bits.

Cette miniaturisation a offert des possibilités de réaliser des systèmes embarqués.

2.2 Généralités sur les chaînes d'acquisition des données

L'acquisition de données est une définition assez générale d'un ensemble des dispositifs permettant de transformer l'information contenue dans une grandeur physique en une information exploitable par l'observateur.

Un système d'acquisition de données doit donc être non seulement capable de faire la mesure ou plutôt l'ensemble des mesures, et de rendre l'information contenue dans celle-ci disponible pour une exploitation immédiate ou future. Cela suppose donc que la mesure traduise fidèlement la grandeur physique, que sa représentation conserve elle aussi cette information et enfin que cette dernière puisse être éventuellement sauvegardée sur un support pérenne en vue d'une analyse ultérieure.

La **Figure 2-1** représente une organisation interne de base d'un microcontrôleur avec les différents éléments intégrés dans un seul composant.

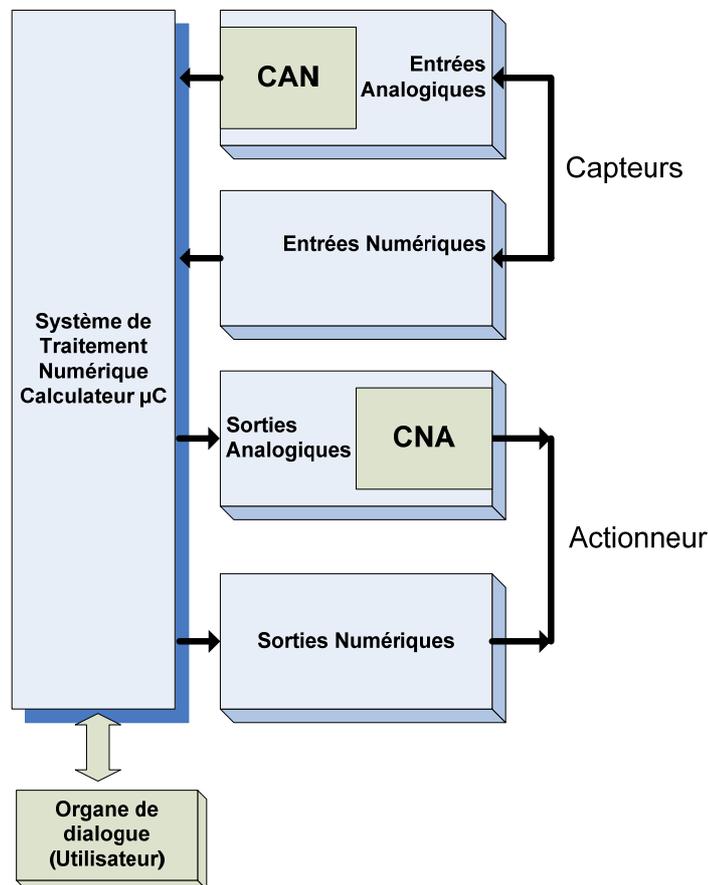


Figure 2-1 : Organisation fonctionnelle d'un Système à Microcontrôleur.

2.3 Architecture d'une chaîne d'acquisition de données :

La chaîne d'acquisition a pour fonction de recueillir et transformer la grandeur à mesurer sous une forme adaptée à son exploitation.



Figure 2-2 : Schéma de la chaîne d'acquisition d'une mesure.

2.3.1 Les capteurs :

Premier élément de la chaîne d'acquisition, le capteur a pour fonction de délivrer un signal électrique en fonction du mesurant, autrement dit il transforme la mesure de la grandeur physique en grandeur électrique (**Figure 2-2**). Il est à noter que certains capteurs sont passifs (i.e. fonctionnent sans alimentation électrique) et d'autres actifs (nécessitent une alimentation électrique).

Il existe plusieurs types de capteurs à savoir :

Capteur de champ magnétique à effet Hall.

Capteur de contrainte mécanique à effet piézoélectrique.

Thermistance

Magnétorésistance

2.3.2 Caractéristiques déterminantes dans le choix d'un capteur :

Pour une application donnée, il est fréquent que plusieurs technologies de capteur puissent convenir. Leur choix dépendra des performances visées en termes de caractéristiques de mesure, dont les principales sont définies ci-après.

Etendue de mesure : (E.M.) différence entre la valeur minimale m_{\min} et maximale m_{\max} du mesurant à obtenir : $E.M. = m_{\max} - m_{\min}$. L'étendue de mesure est définie par la courbe d'étalonnage du capteur (**Figure 2-3**) qui à chaque valeur du mesurant m fait correspondre un signal de sortie s unique.

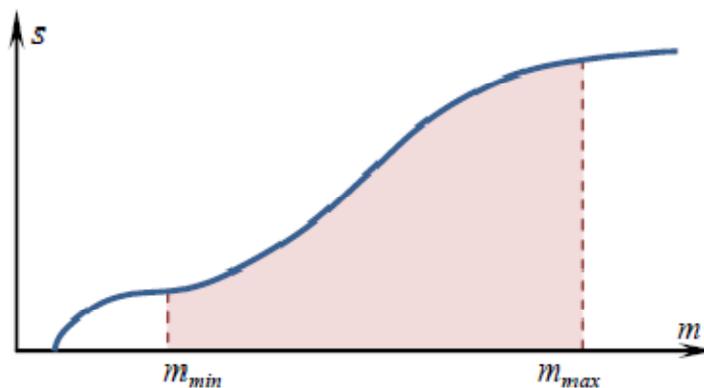


Figure 2-3 : Etendue de mesure d'un capteur.

Dynamique de mesure : C'est la différence entre les valeurs extrêmes mesurables par le capteur pour une marge d'erreur fixée. Les mesures ne sont pas entachées d'une erreur supérieure à celle tolérée, elles sont données avec la notion de précision.

Résolution : Plus petite valeur que le capteur est en mesure d'identifier. La résolution est dépendante du niveau de bruit.

Sensibilité : Facteur de proportionnalité entre le signal de sortie du capteur s et la grandeur mesurée. Pour une valeur donnée m du mesurant, la sensibilité $S(m)$ du capteur est égale au rapport entre la variation de la sortie électrique et la variation du mesurant :

$$S(m) = \left(\frac{\Delta s}{\Delta m} \right)_m$$

Si $S(m)$ est, dans l'étendue de mesure, une fonction linéaire du mesurant m , le capteur est dit linéaire. Sa sensibilité $S(m)$ est alors constante sur l'étendue de mesure

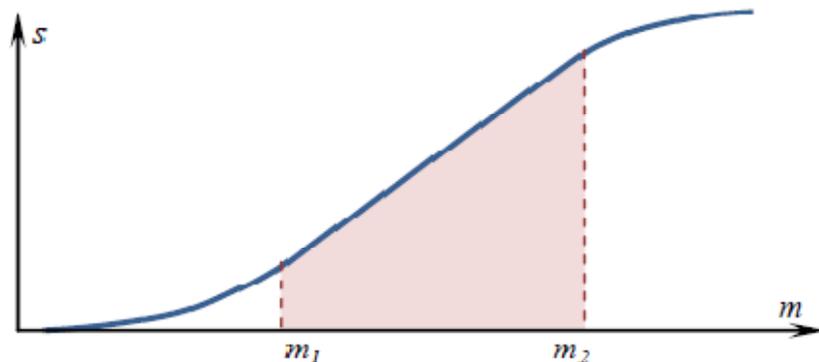


Figure 2-4 : Etendue Domaine de linéarité d'un capteur.

Précision : L'incertitude sur chaque résultat de mesure M doit être quantifiée par une estimation de l'erreur possible exprimée par $\pm \delta M$. On sait alors que $m = M \pm \delta M$. L'erreur de précision est une erreur relative ϵ_p ramenée à l'étendue de mesure :

$$\epsilon_p = \frac{\delta M}{E.M}$$

Bande passante : Gamme de fréquence pour laquelle le capteur fonctionne. Elle est caractéristique de la rapidité du capteur. La rapidité est la capacité de réponses aux variations du mesurant dans le temps.

2.3.3 Conditionnement des signaux

Le conditionnement permet de mettre en forme le signal mesuré en vue d'un traitement et d'une transmission éventuelle.

Conversion courant-tension : Lorsque le signal de sortie du capteur est un courant, le rôle du convertisseur courant-tension est de transformer celui-ci en une tension. Le montage (Figure 2-5) est réalisé de telle sorte qu'aucun courant ne parcourt R_c , et que tout le courant fourni par la source $i(m)$ représentant le courant à convertir traverse R et détermine la tension de sortie :

$$v_{out} = - R.i(m)$$

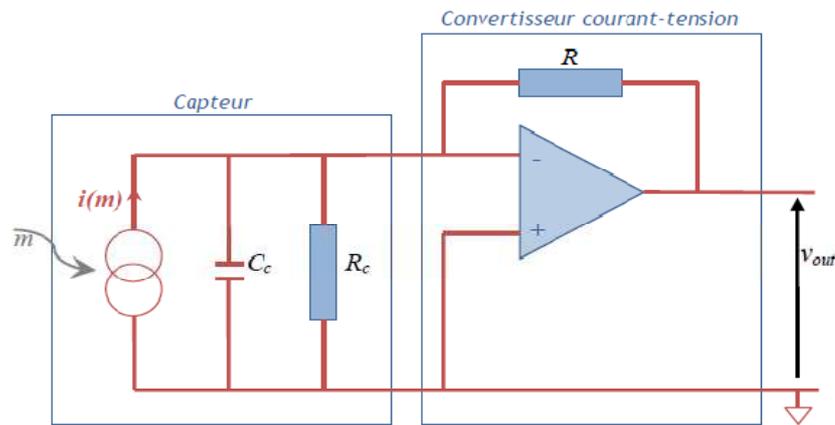


Figure 2-5 : Capteur générant un courant associé à un convertisseur courant-tension à amplificateur opérationnel.

Amplification : Lorsque les signaux électriques issus des capteurs sont de faible amplitude, il peut être nécessaire de les amplifier pour les adapter à la chaîne de transmission. Il faut savoir que l'amplification (en tension ou en puissance) du signal électrique issu du capteur est un phénomène bruyant : elle s'accompagne d'une dégradation du rapport signal sur bruit. Cela signifie que si l'amplitude du signal utile issue du capteur se trouve augmentée, les parasites (bruit) le sont également mais dans des proportions plus grandes encore.

Les amplificateurs d'instrumentation sont conçus de manière à optimiser le rapport signal sur bruit, c'est à dire à « peu » le dégrader. Ils sont caractérisés par un gain d'amplification (en tension ou en puissance) ratio du signal électrique de sortie de l'amplificateur sur le signal d'entrée, ainsi que par un facteur fruit $F > 1$ qui quantifie la dégradation du rapport signal sur bruit entre l'entrée et la sortie.

Filtrage : Le filtrage peut avoir différentes applications. Il peut en particulier être pratiqué afin de réduire le bruit (signal parasite « large bande » ou haute fréquence) entachant le signal utile. Ainsi un filtrage passe-bas éliminera le bruit haute fréquence et produira un effet de lissage utile. Placé avant l'échantillonnage de la conversion analogique/numérique (voir le titre suivant) le filtre d'entrée appelé filtre anti-repliement, contraint le signal à avoir un spectre limité tel que

$f_{\max} < f_e/2$. Placé en sortie de la conversion analogique/numérique, le filtrage lisse le signal de sortie pour restituer le signal utile.

2.3.4 Conversion analogique/numérique :

La conversion analogique/numérique consiste à transformer la tension analogique (issue du capteur) en un code binaire (numérique) adapté à son exploitation dans un processus de régulation, de contrôle, de calculs ou encore de stockage. La conversion analogique/numérique n'est pas systématique, un stockage ou une régulation pouvant également être réalisés à partir de données analogiques.

Le Convertisseur Analogique Numérique (CAN) transforme le signal analogique, signal continûment variable pouvant prendre une infinité de valeurs, en un signal numérique, signal discontinu pouvant être représenté aux moyens de données binaires (0 et 1). La conversion analogique/numérique comporte deux étapes, l'échantillonnage et la conversion proprement dite. L'échantillonnage est une opération qui doit satisfaire un juste équilibre entre précision et rapidité.

La rapidité à laquelle sont prélevés les échantillons doit permettre une reconstruction fidèle du signal, elle est représentée par la fréquence d'échantillonnage f_e qui doit être suffisamment grande pour retranscrire les variations rapides du signal (**Figure 2-6**). Le théorème d'échantillonnage aussi dénommé théorème de Shannon-Nyquist, permet de déterminer la fréquence d'échantillonnage d'un signal donné. Il énonce que la reconstruction d'un signal de sortie fidèle au signal d'entrée, requiert de choisir une fréquence d'échantillonnage qui soit au moins deux fois supérieure à la fréquence maximale contenue dans le signal d'entrée, soit :

$$f_e \geq 2.f_{\max}.$$

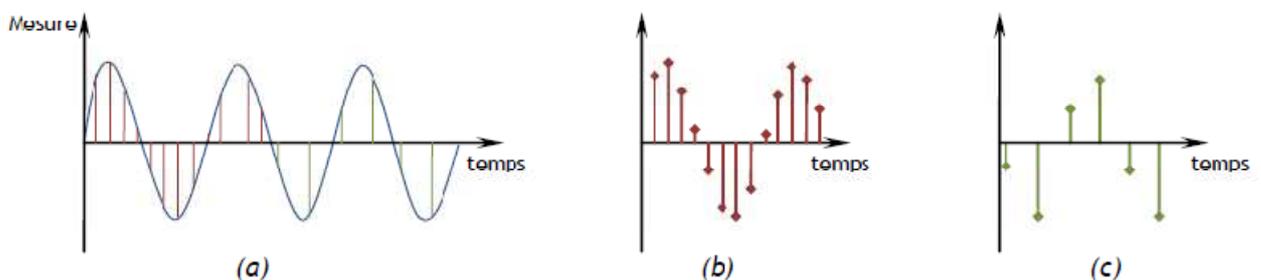


Figure 2-6 : Echantillonnages d'un signal de mesure (a) avec une fréquence d'échantillonnage plus élevée dans le cas (b) que le cas (c), $f_{eb} > f_{ec}$

La précision du codage du signal numérique dépend du nombre de bits sur lequel s'effectue ce codage. Chaque peut prendre 2 valeurs (0 ou 1), un codage sur 2 bits peut prendre 4 valeurs, un codage sur n bits peut prendre 2^n valeurs.

Signal numérique : L'association capteur-conditionneur détermine les caractéristiques du signal de sortie. La chaîne d'acquisition doit être considérée dans son ensemble, ses caractéristiques sont déterminées de manière à répondre à l'application visée. Les divers dispositifs constituant la chaîne d'acquisition (représentés dans la **Figure 2-7**) et sa structure doivent permettre le mesurage, le traitement et la restitution de la mesure avec les caractéristiques nécessaires à l'application : résolution, précision, rapidité et immunité aux parasites.

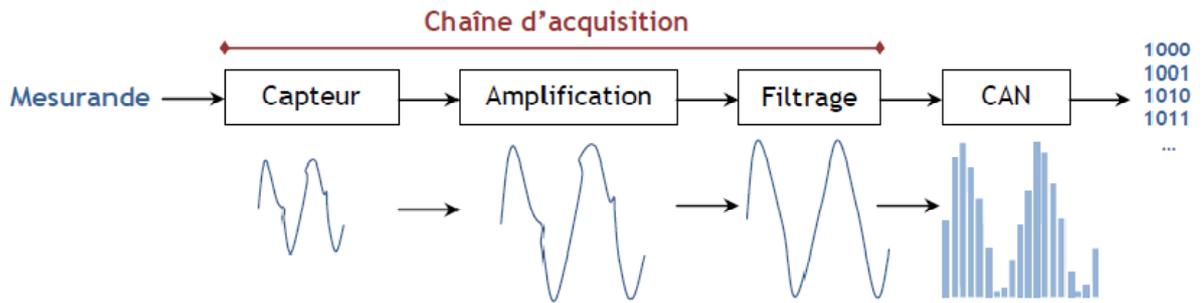


Figure 2-7 : Construction typique d'un signal numérique.

2.4 Modèle de base d'un microcontrôleur :

Un microcontrôleur se présente sous la forme d'un circuit intégré réunissant tous les éléments d'une structure à base de microprocesseur démontrée dans la **Figure 2-8**.

Voici généralement ce que l'on trouve à l'intérieur d'un tel composant :

- Un microprocesseur (C.P.U.),
- De la mémoire de donnée (RAM et EEPROM),
- De la mémoire programme (ROM, OTPROM, UVPROM ou EEPROM),
- Des interfaces parallèles pour la connexion des entrées / sorties,
- Des interfaces séries (synchrone ou asynchrone) pour le dialogue avec d'autres unités,
- Des Timers pour générer ou mesurer des signaux avec une grande précision temporelle,
- Des convertisseurs analogique / numérique pour le traitement de signaux analogiques.

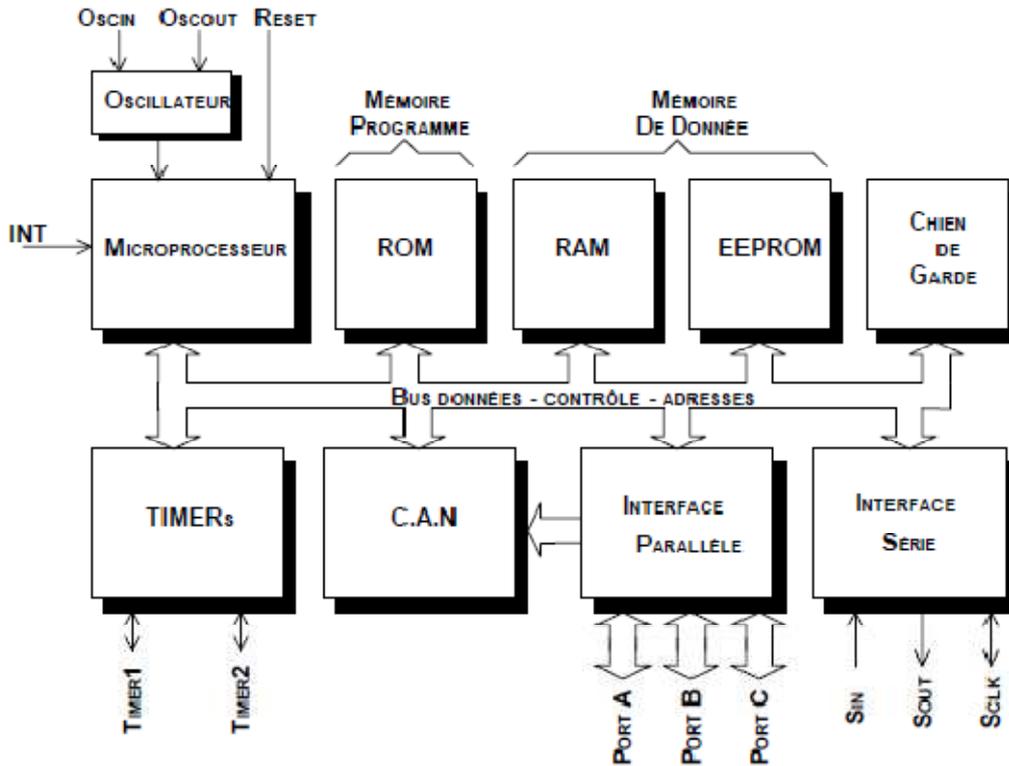


Figure 2-8 : Architecture interne du microcontrôleur.

2.4.1 L'architecture de Von Neumann :

Cette architecture (Figure 2-9) est un modèle pour microcontrôleur qui utilise une structure de stockage unique pour conserver à la fois les instructions et les données requises ou générées par le calcul. La séparation entre le stockage et le processeur est implicite dans ce modèle. Une même instruction permet l'accès au programme ou aux données, cependant pas simultanément. Cette architecture est maintenant principalement utilisée pour la conception des processeurs d'ordinateurs (PC, MAC)

Ce type d'architecture nécessite plusieurs cycles d'horloge pour exécuter une instruction (recherche d'instruction, décodage, lecture opérande, exécution).

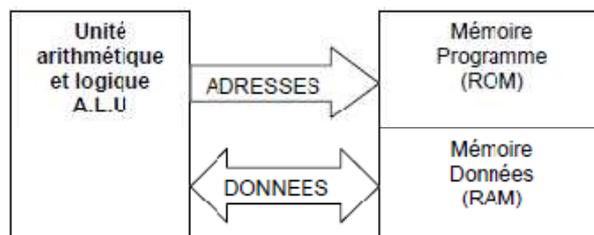


Figure 2-9 : Architecture VON NEUMAN

2.4.2 L'architecture de type Harvard :

Une autre conception (**Figure 2-10**) de microprocesseurs qui sépare physiquement la mémoire de données et la mémoire programme. L'accès à chacune des deux mémoires s'effectue via deux bus distincts. Cette structure permet un accès simultané aux données et aux instructions l'exécution des programmes est plus rapide. En revanche elle nécessite des instructions différentes pour accéder à la mémoire programme et à la mémoire de données.

Cette architecture très employée pour la conception des processeurs de traitement de signal (DSP) est de plus en plus utilisée pour les microcontrôleurs d'usage.

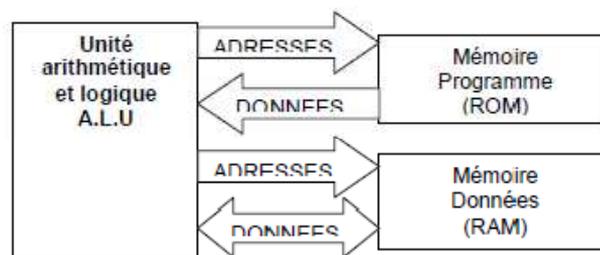


Figure 2-10 : Architecture Harvard

2.4.3 Rôle d'un système à microcontrôleur :

Un système à microcontrôleur permet :

- d'acquérir des entrées logiques et analogiques représentant l'état du système technique,
- d'interpréter, la signification de ces entrées,
- de calculer, mémoriser, récupérer des variables logicielles intermédiaires,
- de gérer le temps et d'agir sur des sorties logiques et analogiques en fonction des entrées et des calculs réalisés de manière à modifier le fonctionnement du système technique (commande moteur, affichage d'informations,...),
- de communiquer par des liaisons séries avec d'autres systèmes techniques et/ou un ordinateur,

2.4.4 Avantages :

- Encombrement réduit,
- Circuit imprimé peu complexe,
- Faible consommation,
- Coût réduit.

2.4.5 Inconvénients :

- Système de développement onéreux,
- Programmation nécessitant un matériel adapté.

2.5 Le bus d'interface I2C (Inter-Integrated Circuit)

Dans les systèmes grands publics (téléviseurs, appareils hifi, lecteur CD...), les échanges de données entre les composants se faisait sur des bus parallèles.

Du fait de la sophistication des appareils toujours plus importante, ces liaisons parallèles devenaient de plus en plus nombreuses, dans le but de minimiser ces liaisons, par conséquent pour augmenter la fiabilité, a été crée le bus série I2C au début des années 80 par Philips.

D'autres fabricants ont adoptés le protocole I2C et de nombreux composants sont apparus (mémoires, convertisseurs, capteurs, micro - contrôleurs ...)

Le bus I2C fait partie des réseaux LAN. Il été créée pour fournir un moyen simple de transférer des informations numériques entre des capteurs et des microcontrôleurs.

I2C à l'avantage de n'avoir besoin que de deux connexions de signalisation (l'emploi de plusieurs périphériques sur les deux connexions est assez simple et on reçoit une confirmation que les signaux ont été correctement reçus).

L'inconvénient est que la vitesse des données est inférieure aux autres bus et qu'elles ne peuvent voyager que dans un seul sens à la fois (Simplex), ce qui abaisse encore plus le débit si des communications bidirectionnelles sont nécessaires (Half Duplex).

2.5.1 Propriétés physiques du bus I2C

- **Le support physique :**

Le bus I2C est de faible longueur. Il est réalisé par des pistes de cuivre sur circuit imprimé ou bien de manière filaire. La transmission se fait sur 3 fils illustrés dans la **Figure 2-11** et détaillées dans la **Figure 2-12**.

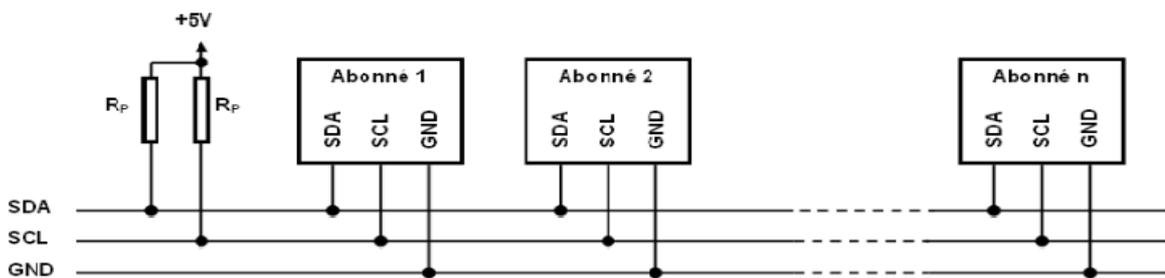


Figure 2-11 : Synoptique d'une liaison I2C.

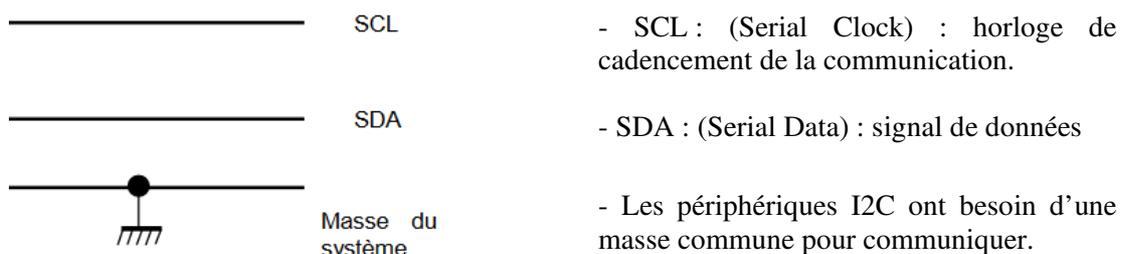


Figure 2-12 : les signaux d'un bus I2C.

Les caractéristiques physiques et électriques du bus I2C sont résumées dans la **Figure 2-13**.

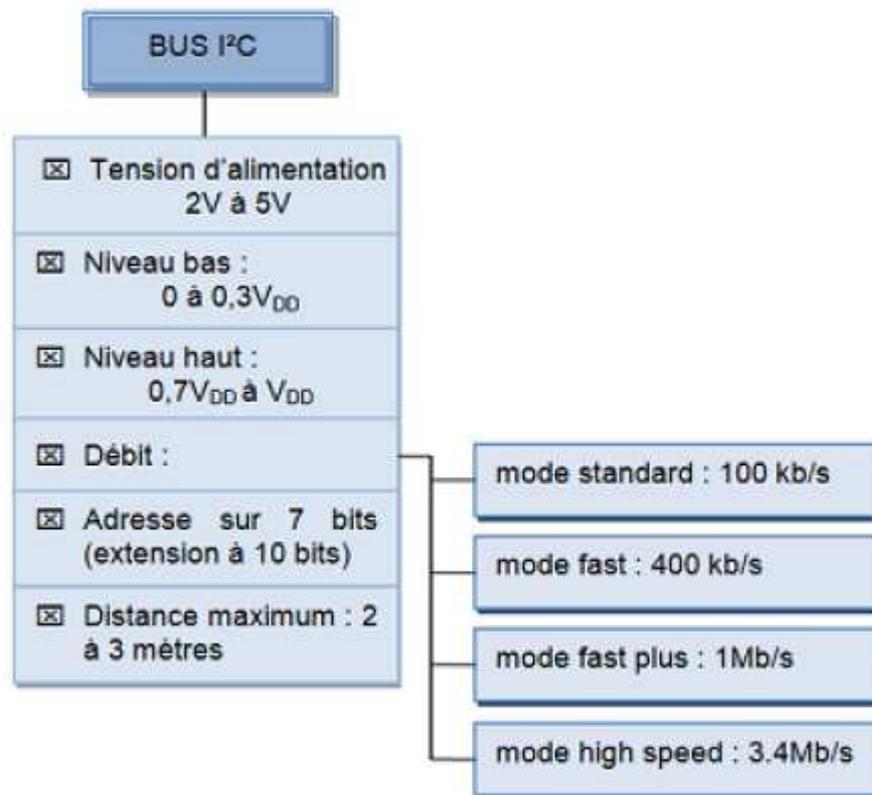


Figure 2-13 : caractéristiques du bus I2C.

2.5.2 Le Protocole :

La communication sur le bus I2C ne peut se faire qu'entre 2 abonnés.

Lorsqu'un abonné prend le contrôle du bus, il devient le maître de la communication. Il génère le signal d'horloge SCL et communique avec un esclave. Selon le sens de la communication, il sera l'émetteur ou le récepteur.

Lorsqu'aucun abonné n'émet sur le bus, les lignes SDA et SCL sont au niveau haut qui est leur état de repos.

Quand une ligne (SDA ou SCL) est au repos (niveau 1), on peut la forcer à 0.

Quand une ligne (SDA ou SCL) est au niveau 0, on ne peut pas la forcer à 1.

Une donnée n'est considérée comme valide sur le bus que lorsque le signal SCL est à l'état haut.

L'émetteur doit donc positionner la donnée à émettre lorsque SCL est à l'état bas et la maintenir tant que SCL est à l'état haut.

Comme la transmission s'effectue sous forme série, une information de début et de fin doit être prévue. L'information de début se nomme START et l'information de fin STOP.

L'en-tête :

La **Figure 2-14** montre le contenu du premier octet qui est toujours présent en début d'échange. Ses sept bits de poids forts contiennent l'adresse du destinataire, ce qui autorise 128 combinaisons différentes. Le bit de poids faible indique si le maître va réaliser une lecture ou une écriture. Si ce bit est à zéro, le maître va écrire dans l'esclave ou lui envoyer des données. Si ce bit est à un, le maître va recevoir des données de l'esclave.

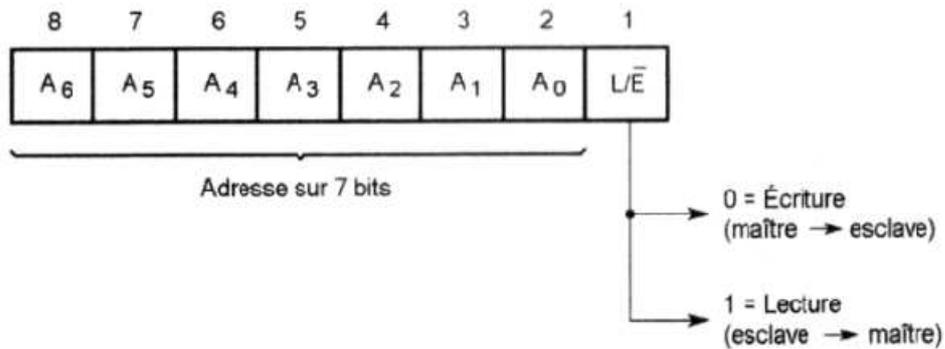


Figure 2-14 : Adressage des esclaves en Bus I2C.

Début de communication :

Un abonné prend le contrôle du bus I2C en émettant une condition de départ :

1. Niveau haut sur SCL
 2. Front descendant sur SDA
- Cet abonné devient le maître.

(**Figure 2-15**)

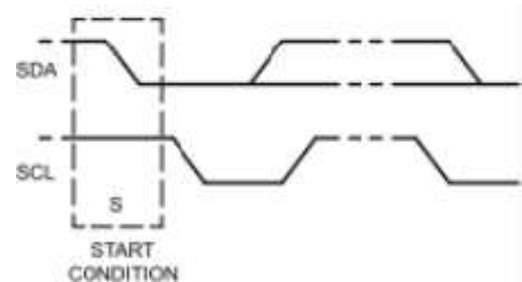


Figure 2-15 : Début de la communication en I2C.

Transmission de l'adresse :

Après avoir pris le contrôle, le maître transmet un octet contenant l'adresse de l'esclave (sur 7 bits) ainsi que l'opération effectuée (écriture ou lecture).

Lecture (NL 1), Écriture (NL 0)

La transmission d'un bit se fait lorsque le signal SCL est au niveau haut :

(**Figure 2-16**).

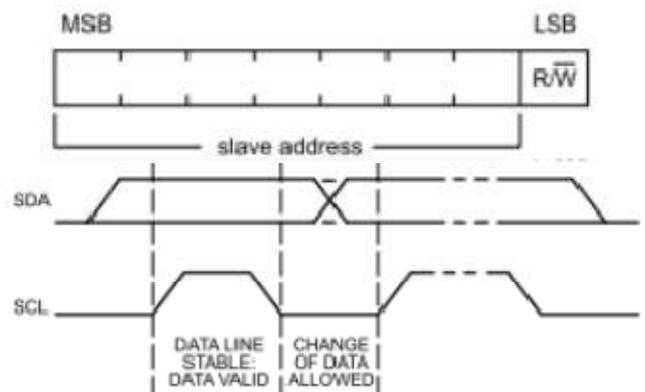


Figure 2-166 : Transmission d'une adresse en I2C.

Confirmation de réception par l'esclave :

Lorsque l'esclave a détecté son adresse, il émet un bit d'acquiescement (ACK) au niveau logique bas comme mentionné dans la **Figure 2-17**.

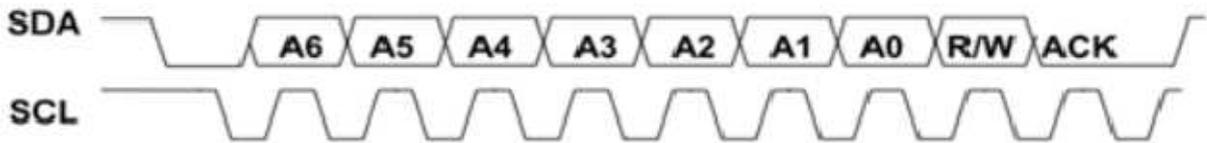


Figure 2-17 : Confirmation de réception d'adresse en I2C.

Transmission des données :

Deux cas se présentent :

1. Le maître envoie des données à l'esclave. A la fin de la transmission de chaque octet, l'esclave émet un acquiescement.

Dans la Figure 2-18 une représentation chronologique d'opération d'écriture de données sur un espace mémoire de l'esclave.

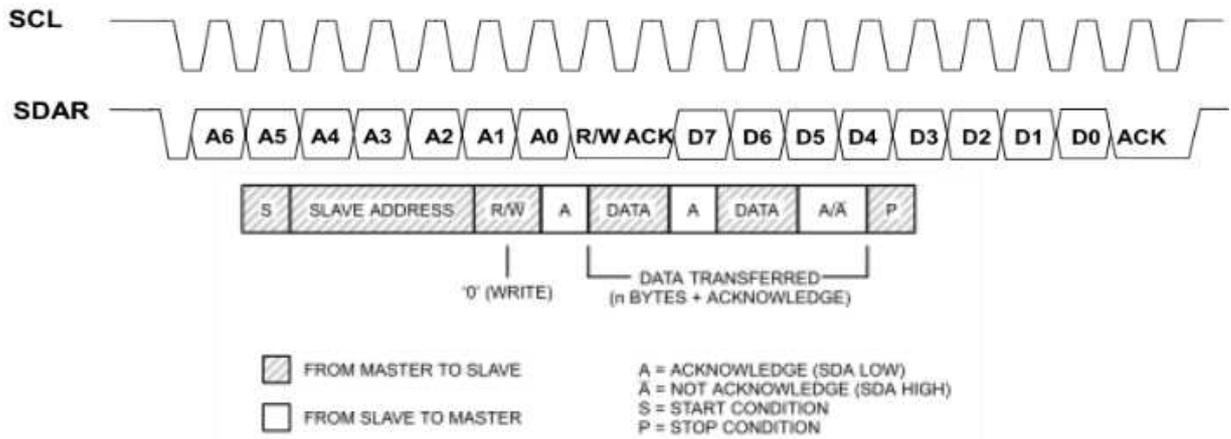


Figure 2-18 : Chronologie d'écriture de données sur I2C.

2. L'esclave envoie des données au maître. A la fin de la transmission d'un octet, le maître émet un acquiescement s'il veut recevoir encore un octet ou bien un non acquiescement (NL 1) s'il a terminé de recevoir.

Dans la Figure 2-19 une représentation chronologique d'opération de lecture de données d'un espace mémoire de l'esclave.

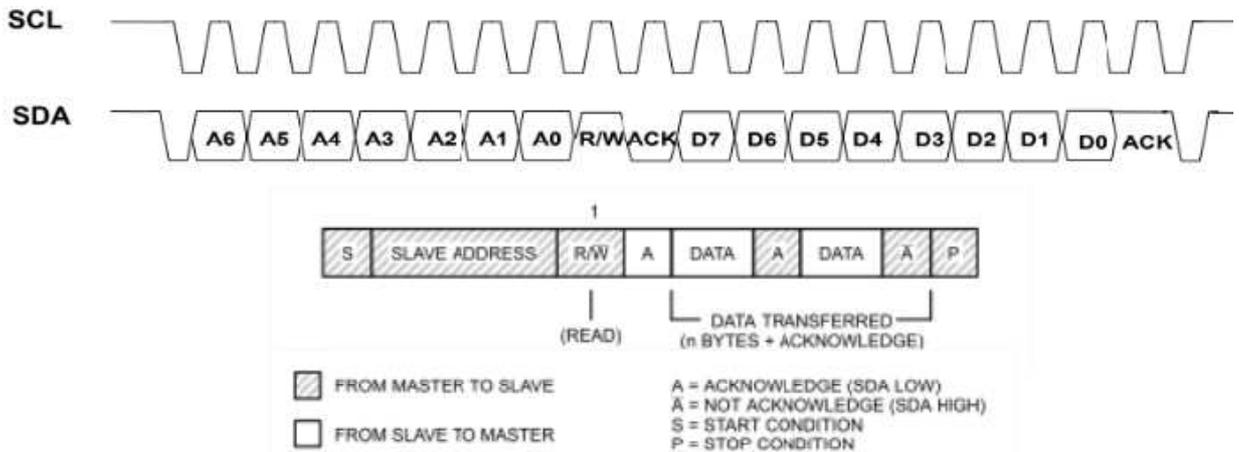


Figure 2-19 : Chronologie de lecture de données sur I2C.

Fin de communication :

Pour terminer la communication, le maître émet une condition d'arrêt.

1. Niveau haut sur SCL
2. Front montant sur SDA

Tous les abonnés sont alors déconnectés du bus. SDA et SCL sont au niveau Haut.

Opération schématisée dans la **Figure 2-20**.

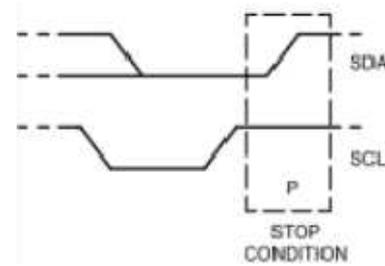


Figure 2-20 : Fin de la communication en I2C.

Lorsqu'un maître désire effectuer plusieurs échanges à destination d'esclaves d'adresses différentes, il n'est pas obligé de terminer le premier échange par une condition d'arrêt mais peut les enchaîner en générant une condition de départ dès la fin d'un échange comme représenté dans la **Figure 2-21**.

Repeated Start

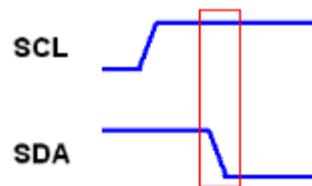


Figure 2-21: Redémarrage de la communication.

2.6 Le module RTC: Real Time clock

Le circuit Dallas DS1307 est une horloge temps réel (Real Time Clock), qui fournit secondes, minutes, heures, jours, dates, mois et années. Les années bissextiles sont prises en compte (jusqu'en 2100).

Le DS1307 travaille dans le mode standard (fréquence d'horloge de 100 kHz). L'adresse I2C (7 bits) du DS1307 est : 1101000 (adresse fixée par le constructeur et non modifiable).

2.6.1 Exemple d'écriture du DS1307

L'émetteur est le maître et le récepteur est l'esclave.

Le registre d'adresse 0x04 du DS1307 contient la date (voir datasheet du DS1307).

Pour régler le calendrier au 27 du mois, il faut écrire la donnée 27 (en code BCD) dans le registre d'adresse 0x04 du DS1307.

Le bus I2C utilise le protocole démontré dans la **Figure 2-22**.



Figure 2-22: Exemple d'écriture de données sur le DS1307 via I2C.

1. Pour initier le dialogue, le maître crée une condition Start.

2. Le maître envoie l'adresse de l'esclave (1101000) suivi du bit 0 (bit Write).
 3. L'esclave répond (accusé de réception : bit ACKnowledge).
 4. Le maître envoie l'adresse du registre (04h) à écrire.
 5. L'esclave répond (accusé de réception : bit ACKnowledge).
 6. Le maître envoie la donnée (27) à écrire.
 7. L'esclave écrit la donnée puis envoie un accusé de réception (bit ACKnowledge).
 8. Le maître termine le dialogue avec une condition Stop.
- Le bus I2C est maintenant libre (SCL = 1, SDA = 1 = niveaux de repos).

2.6.2 Exemple de lecture du DS1307

L'émetteur est l'esclave et le récepteur est le maître.

Les registres d'adresses 00h à 06h du DS1307 contiennent respectivement les secondes, minutes, heures, jours, dates, mois et années (voir datasheet du DS1307).

La **Figure 2-23** démontre comment lire, d'une seule traite, le contenu des registres d'adresses 00h à 06h du DS1307 :

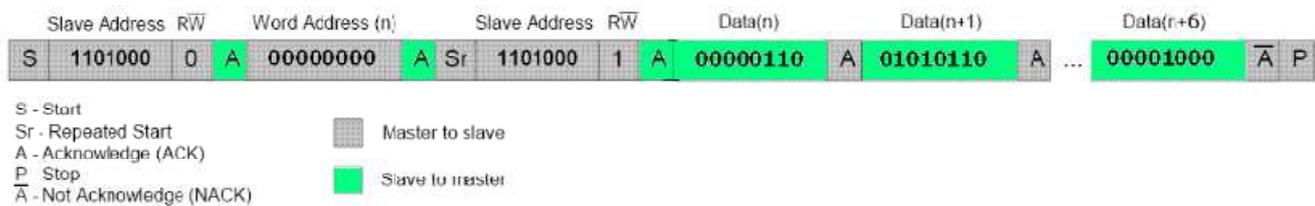


Figure 2-23: Exemple de lecture de données de DS1307 via I2C.

Chronogramme correspondant :

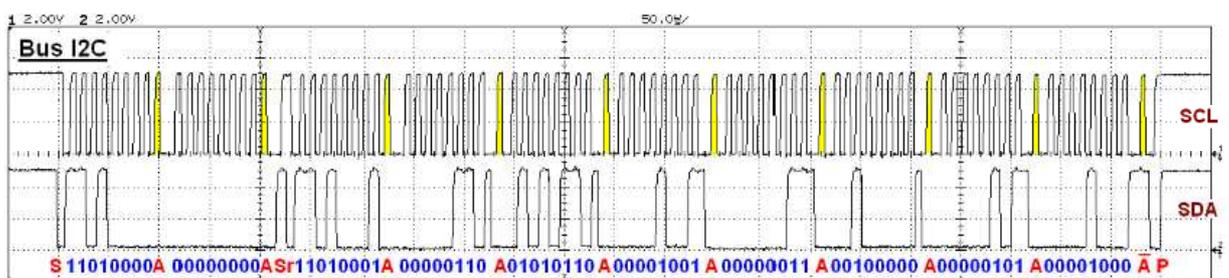


Figure 2-24 : Chronogramme complet de l'échange avec l'esclave DS1307 sur I2C.

1. Pour initier le dialogue, le maître crée une condition Start.
2. Le maître envoie l'adresse de l'esclave (1101000) suivi du bit 1 (bit Write).
3. L'esclave répond (accusé de réception : bit ACKnowledge).
4. Le maître envoie l'adresse du registre (0x00) à lire.
5. L'esclave répond (accusé de réception : bit ACKnowledge).

6. Le maître émet une condition Repeated Start.
 7. Le maître envoie l'adresse de l'esclave (1101000) suivi du bit 1 (bit Read).
 8. L'esclave répond (accusé de réception : bit ACKnowledge).
 9. L'esclave envoie le contenu du registre d'adresse 0x00 au maître.
 10. Le maître répond (accusé de réception : bit ACKnowledge).
 11. L'esclave envoie le contenu du registre d'adresse 0x01 (automatiquement incrémenté) au maître.
 12. Le maître répond (accusé de réception : bit ACKnowledge).
 13. L'esclave envoie le contenu du registre d'adresse 0x02 (automatiquement incrémenté) au maître.
 14. Le maître répond (accusé de réception : bit ACKnowledge).
 -
 -
 21. L'esclave envoie le contenu du registre d'adresse 0x06 (automatiquement incrémenté) au maître.
 22. Le maître répond (accusé de réception : bit Not ACKnowledge).
 23. Le maître termine le dialogue avec une condition Stop.
- Le bus I2C est maintenant libre (SCL = 1, SDA = 1 = niveaux de repos).
- Retrouver précisément la date et l'heure à l'aide du chronogramme complet de l'échange sur bus I2C avec l'esclave DS1307.

Les réseaux de communication (RS485) :

2.6.3 Constitution d'une liaison

La communication entre systèmes informatiques s'effectue grâce à des liaisons dont les principaux éléments sont définis par des normes dans le cadre du CCITT (Comité Consultatif International des Téléphones et Télégraphes).



Figure 2-25 : Eléments d'une liaison entre systèmes informatiques.

ETTD : Equipement Terminal de Traitement de données (ou DTE : Data Terminal Equipment).

ETCD : Equipement de Terminaison de Circuit de Données (ou DCE : Data Communication Equipment).

L'ETTD est un élément qui ne se connecte pas directement à la ligne de transmission. Par exemple : un ordinateur, un terminal, une imprimante...

L'ETCD assure la transmission des données. Par exemple : un modem, un multiplexeur... Il a deux fonctions principales :

- l'adaptation du signal de l'ETTD à la ligne (codage et modulation en émission, démodulation et décodage en réception) ;
- la gestion de la liaison (établissement, maintien et libération de la ligne).

Modes d'exploitation

Il existe trois modes d'exploitation d'une liaison. L'un est choisi en fonction de l'application.

- **Simplex** : Un système est émetteur, l'autre récepteur. C'est par exemple le cas des émissions radio ou télévision.

- **Semi-duplex (half duplex)** : Les deux systèmes peuvent être à la fois émetteur et récepteur, mais les émissions ne peuvent avoir lieu en même temps.

- **Duplex (full duplex)** : Les deux systèmes peuvent être à la fois émetteur et récepteur, les émissions pouvant avoir lieu en même temps.

Normalisation des jonctions

Ils existent Trois (03) organismes de normalisation :

- **CCITT** : Comité Consultatif International des Téléphones et Télégraphes.
- **ISO** : International Standardisation Organisation.
- **EIA** : Electrical Industry Association.

Un bus RS485 est défini par la norme EIA-TIA-485-A et le guide d'application TSB-89-A

Le support de transmission est différentiel. Deux fils correspondant à des niveaux complémentaires sont donc utiles pour chaque signal ce qui limite l'influence des bruits extérieurs et des masses. Pour la RS485, des circuits trois états permettent des liaisons multipoints comme présenté dans la **Figure 2-26**.

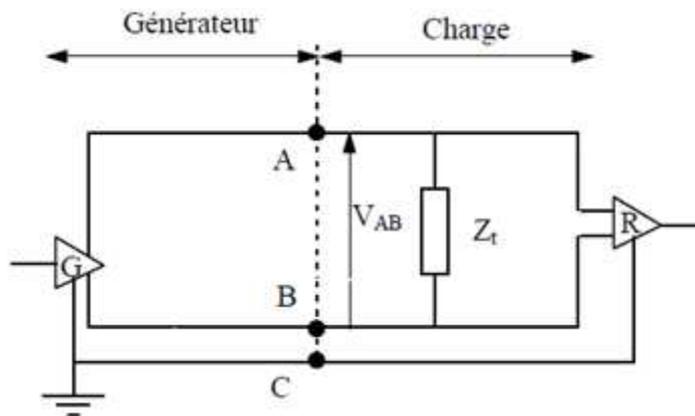


Figure 2-26 : Circuit de jonction équivalent d'un bus RS485.

Les caractéristiques physiques et électriques du bus RS485 récapitulées dans le **Tableau 2-1** doivent être respectés lors de la conception d'un système communicant via RS485.

Tableau 2-1 : Caractéristiques du réseau RS485.

Caractéristiques	Min	Max
Tension du générateur en circuit ouvert (V_{AB})		± 6 V
Courant de sortie en court-circuit (I_0)		150 mA
Résistance interne du générateur	50 Ω	100 Ω
Résistance de charge	(R_L) 100 Ω	
Capacité parallèle		(C_L) 2500 pF
Support	Paire différentielle	
Vitesse	10 Mbit/s	
Longueur de câble	10m - 1km	

2.6.4 Caractéristiques électriques du bus RS485.

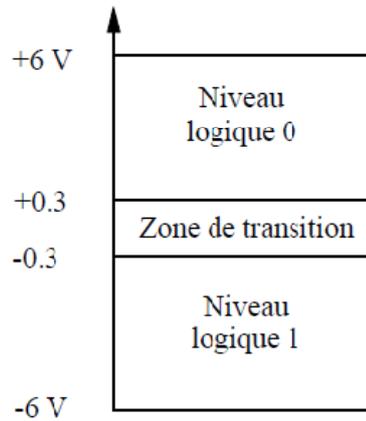


Figure 2-27 : Niveaux significatifs de la norme

Un réseau RS-485 peut avoir jusqu'à 32 charges unitaires selon l'architecture présentée dans les **Figures 2-28 A et B**. Pour installer plus de 32 appareils sur le même segment de réseau, vous devez installer des routeurs ou des répéteurs RS-485 qui régénèrent les signaux et recommencent un autre segment de réseau.

Le réseau nécessite 2 terminaisons de 120Ω connectées en parallèle avec ou juste derrière le dernier appareil situé à chaque extrémité du segment de réseau. Les terminaisons réduisent les réflexions de signaux qui peuvent causer de mauvaises interprétations de niveaux logiques par les récepteurs.

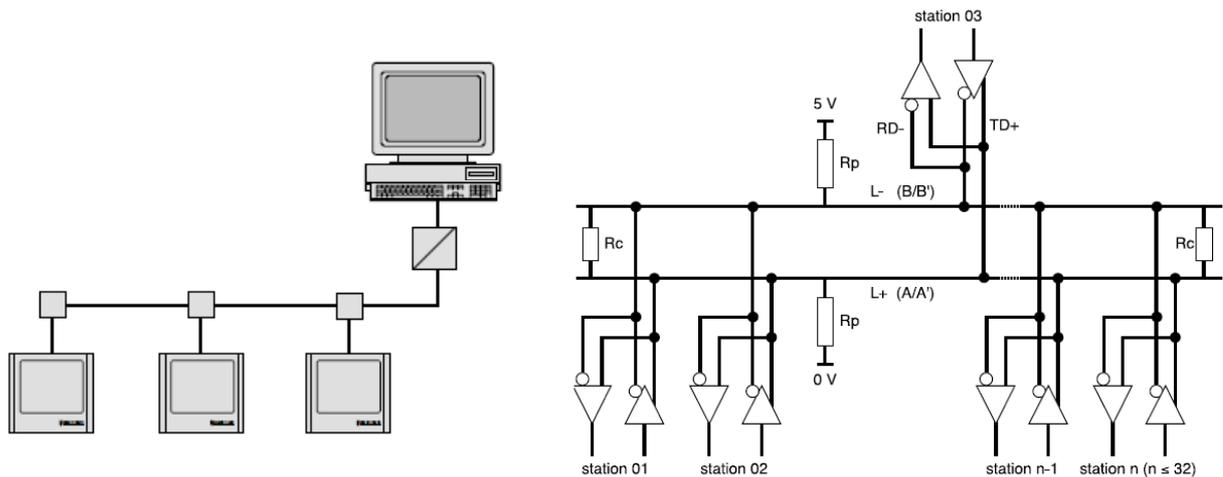


Figure 2-28 A : Architecture générale d'un réseau RS 485 2 fils

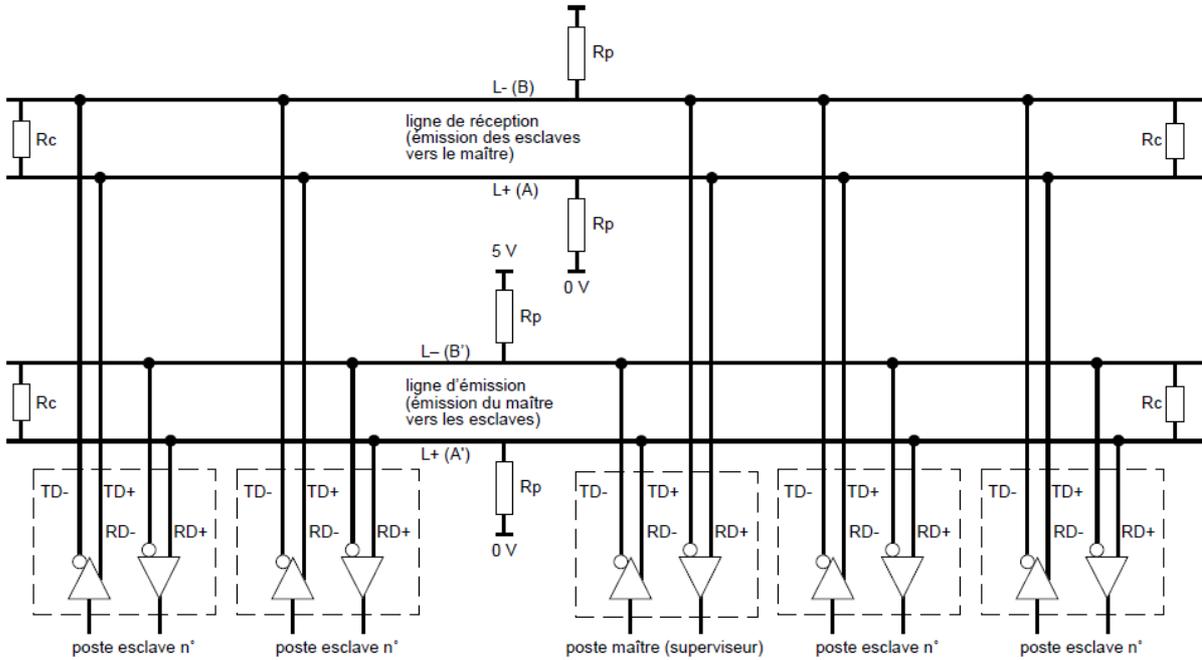


Figure 2-29 B : Architecture générale d'un réseau RS 485 4 fils

2.6.5 Règles de câblage :

Longueur maximale de câble : La longueur maximale d'un réseau est en fonction de sa vitesse de communication et de son nombre d'appareils. Plus le segment de réseau est long, plus la vitesse sera lente. En utilisant du câble et une installation appropriée, la longueur maximale recommandée d'un réseau RS-485 est d'environ 1200 m (4000 pieds). Voir la courbe caractéristique présentée dans la **Figure 2-30**.

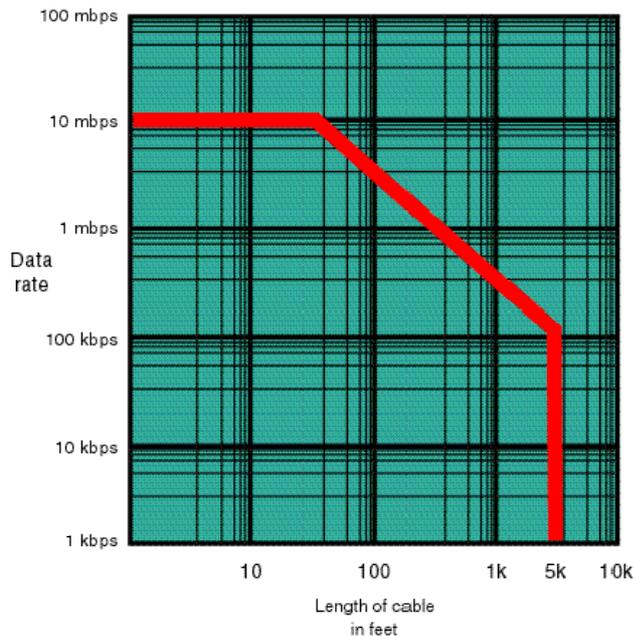


Figure 2-30 : Relation entre la longueur et la vitesse du réseau RS485.

Type de câble : Les constructeurs recommandent l'utilisation de câble EIA RS-485 22-24 AWG torsadé et balancé, avec une impédance caractéristique de 100-130 ohms, une capacitance de moins de 100 pF par mètre (30 pF par pied), avec un blindage tressé. Comparé au blindage en feuille, le blindage tressé offre une meilleure protection contre les perturbations électromagnétiques, en particulier pour les hautes fréquences, quelque type des câbles torsadés présentés dans la **Figure 2-30**.



Figure 2-31 : Type des câbles à utiliser dans le réseau RS485.

Une des extrémités de réseau nécessite aussi deux résistances de polarisation de 604 Ω .

Les résistances de polarisation forcent un état neutre sur le réseau et évitent de faux bits de départ lorsqu'aucun appareil RS-485 n'est en transmission.

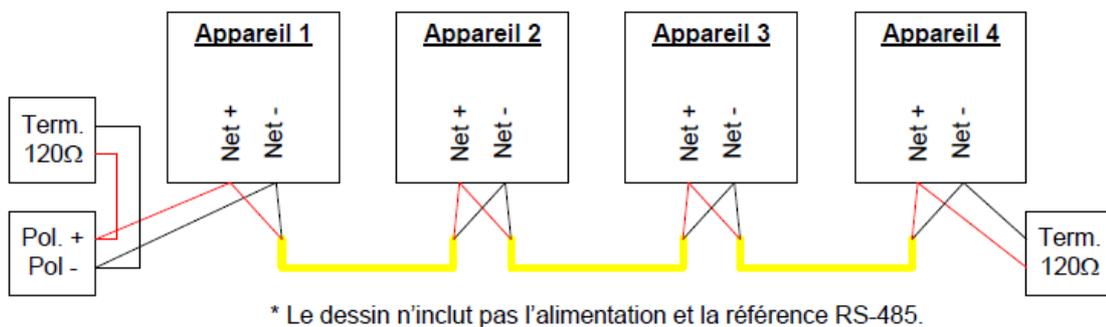


Figure 2-32 : Polarisation et résistance de fin de ligne RS-485.

La polarité présentée dans la **Figure 2-32** des branchements d'un réseau RS-485 est importante. Il faut connecter le réseau positif et réseau négatif aux autres bornes correspondantes des autres composants.

2.7 Les protocoles de communication utilisés dans le réseau RS485 :

Un protocole consiste en la définition de trames d'échange. Plusieurs protocoles en été définies. Le protocole le plus connu est le protocole Modbus (marque déposée par MODICON) et Profibus (Siemens) qui sont deux protocoles standards de dialogue basé sur une structure

hiérarchisée entre un maître et plusieurs esclaves (**Figure 2-33**). Néanmoins nous sommes libres de définir notre propre protocole.

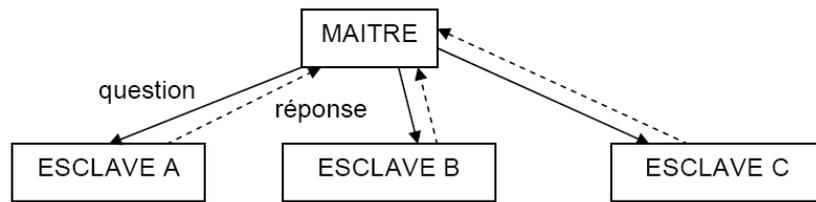


Figure 2-33 : Architecteur hiérarchique dans le réseau RS-485.

Ces protocoles sont caractérisés comme suit :

- Le maître envoie une demande et attend une réponse.
- Deux esclaves ne peuvent dialoguer ensemble.
- Le dialogue maître – esclave peut être schématisé sous une forme successive de liaisons point à point.
- Mode de communication : half-duplex. (2 fils ou 4 fils)
- le maître parle à l'ensemble des esclaves, sans attente de réponse (diffusion générale).

Il ne peut y avoir sur la ligne qu'un seul équipement en train d'émettre. Aucun esclave ne peut envoyer un message sans une demande préalable du maître. Le dialogue direct entre les esclaves est impossible.

2.7.1 Structure des messages :

Le maître envoie un message constitué de la façon présentée dans la **Figure 2-34** :

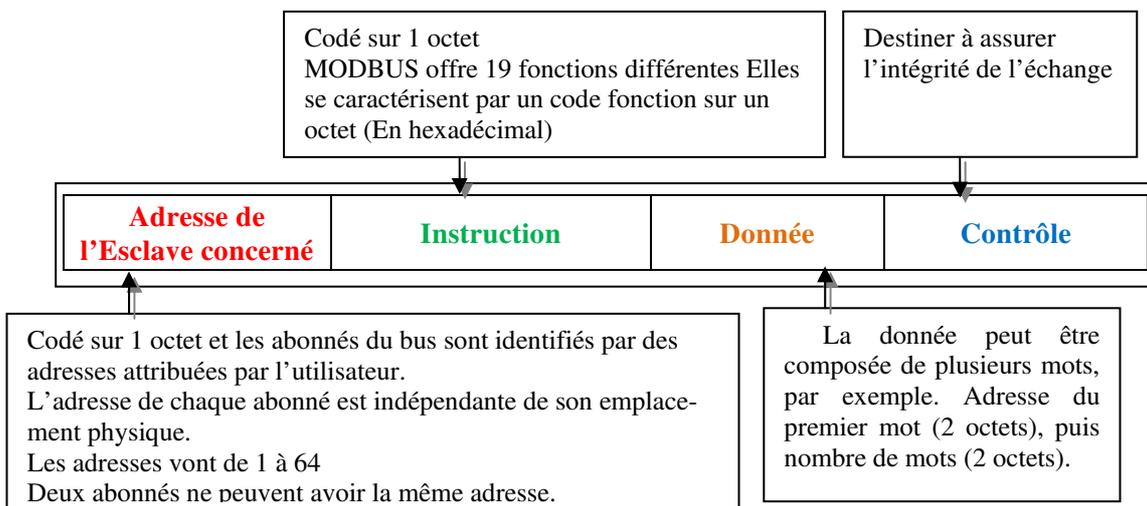


Figure 2-34 : Structure des messages « protocole standard ».

La détection de fin de message est réalisée sur un silence de durée supérieure ou égale à la durée de transmission de 3 octets.

- L'esclave répond par un message du même type que le message question.

2.7.2 Adressage

Les abonnés du bus sont identifiés par des adresses attribuées par l'utilisateur.

L'adresse de chaque abonné est indépendante de son emplacement physique.

Les adresses vont de 1 à 64 pour le protocole Modbus et de 1 à 255 en général et ne doivent pas obligatoirement être attribuées de manière séquentielle et deux abonnés ne peuvent avoir la même adresse.

2.7.3 Echange du maître avec un esclave

Le maître interroge un esclave de numéro unique sur le réseau et attend de la part de cet esclave une réponse (**Figure 2-35**).

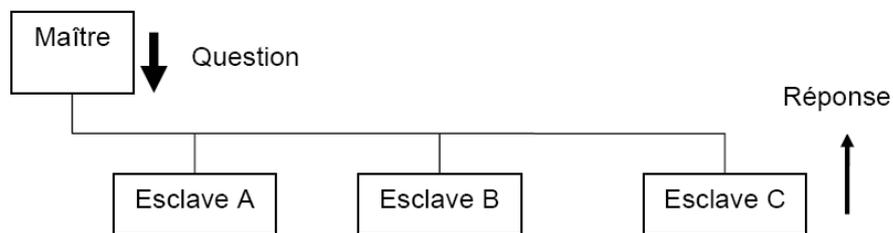


Figure 2-35 : Echange du maître avec un esclave

2.7.4 Echange Maître vers tous les esclaves

Le maître diffuse un message à tous les esclaves présents sur le réseau, ceux-ci exécutent l'ordre du message sans émettre une réponse (**Figure 2-36**).

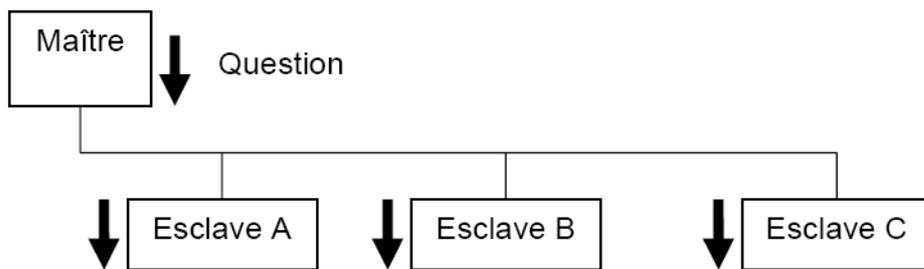


Figure 2-36 : Echange maître vers tous les esclaves

2.7.5 Format général d'une trame

Deux types de codage peuvent être utilisés pour communiquer sur un réseau Modbus. Tous les équipements présents sur le réseau doivent être configurés selon le même type.

- **Mode RTU** (Unité terminale distante) :

La trame ne comporte ni octet d'en-tête de message, ni octets de fin de message.

Sa définition est la suivante :

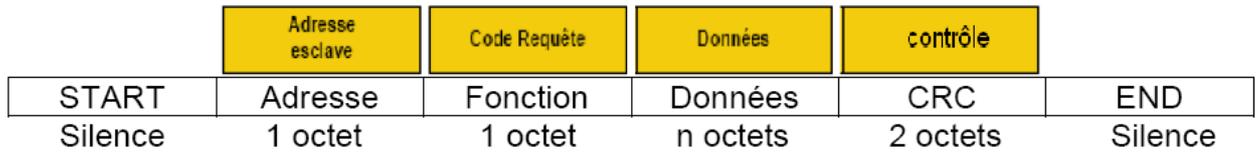


Figure 2-37 : Format d'une trame en mode RTU.

CRC16 : paramètre de contrôle polynomial (cyclical redundancy check).

La détection de fin de trame est réalisée sur un silence supérieur ou égal à 3 octets.

- **Mode ASCII :**

Chaque champ composant une trame est codé avec 2 caractères ASCII (2 fois 8 bits).

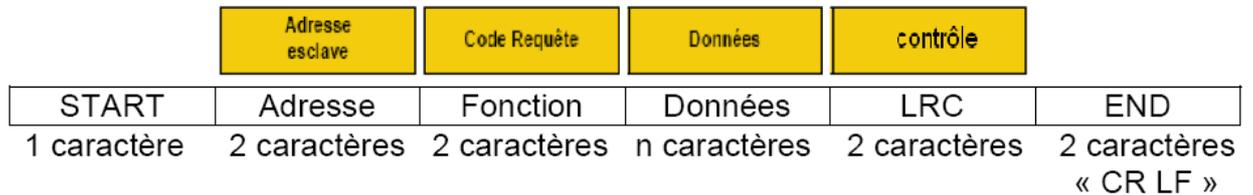


Figure 2-38 : Format d'une trame en mode ASCII.

LRC : C'est la somme en hexadécimal modulo 256 du contenu de la trame hors délimiteurs, complémentée à 2 et transmise en ASCII.

2.7.6 Trame d'échange question/réponse

- **La question :**

Elle contient un code fonction indiquant à l'esclave adressé quel type d'action est demandé.

Les données contiennent des informations complémentaires dont l'esclave a besoin pour exécuter cette fonction.

Le champ octets de contrôle permet à l'esclave de s'assurer de l'intégralité du contenu de la question.

- **La réponse :**

Si une erreur apparaît, le code fonction est modifié pour indiquer que la réponse est une réponse d'erreur.

Les données contiennent alors un code (code d'exception) permettant de connaître le type d'erreur.

Le champ de contrôle permet au maître de confirmer que le message est valide.

2.8 L'Interface Homme-Machines (IHM) :

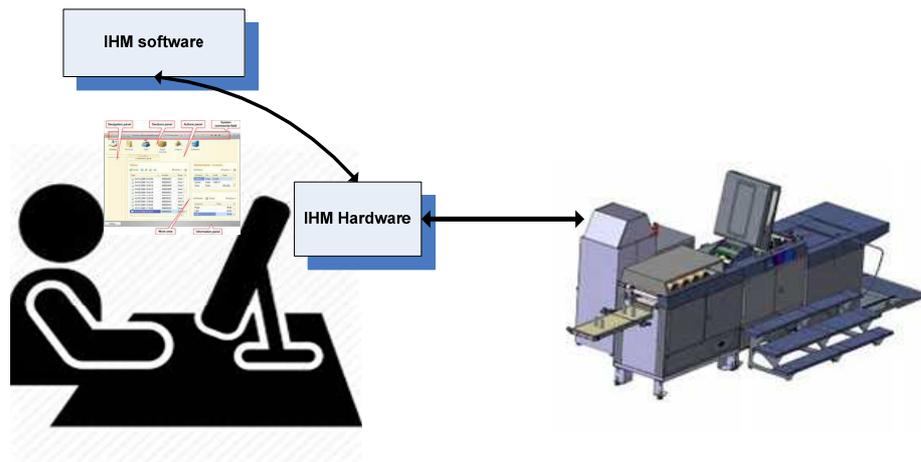


Figure 2-39 : schéma descriptif des composants d'un IHM.

Le domaine d'interactions homme-machine, appelé IHM, s'intéresse à la conception et au développement de systèmes interactifs en prenant en compte ses impacts sociétaux et éthiques². L'homme interagit avec les ordinateurs qui l'entourent, cette interaction nécessite des médiatrices qui facilitent la communication entre l'homme et la machine. La facilitation de l'utilisation de dispositifs devient de plus en plus importante avec le nombre croissant d'interfaces numériques dans la vie quotidienne³. L'IHM a pour but de trouver les moyens le plus efficaces, les plus accessibles et les plus intuitifs pour les utilisateurs afin de compléter une tâche le plus rapidement et le plus précisément possible. L'IHM est pluridisciplinaire, elle bénéficie des sciences cognitives, linguistiques et de psychologie aussi bien que de la vision par ordinateur et que de l'électronique.

L'IHM, -Interface Homme Machine- établit une communication entre un utilisateur et une machine, et agit comme un interprète. L'IHM adapte la logique de la machine à celle de l'utilisateur pour éviter toute mauvaise interprétation lors des échanges d'informations.

2.8.1 Les différentes techniques

Il existe de nombreuses manières pour qu'un humain puisse interagir avec les machines qui l'entourent. Ces manières sont très dépendantes des dispositifs d'interactions et des forces ou compétences que l'être humain ne peut étendre qu'extérieurement.

2.8.2 Les périphériques IHM

D'un point de vue organique, on peut distinguer trois types d'IHM :

- **Les interfaces d'acquisition** : boutons, molettes, joysticks, clavier d'ordinateur, clavier MIDI, télécommande, capteur de mouvement, microphone avec la reconnaissance vocale, etc.

- **Les interfaces de restitution** : écrans, témoins à LED, voyants d'état du système, haut parleur, etc.
- **Les interfaces combinées** : écrans tactiles, multitouches et les commandes à retour d'effort.

Au début la conception des IHM été difficile, par conséquent itérative (itérativité implique modification du logiciel) avec une complexité par rapport à la taille croissante des IHM et l'outil de développement des IHM imparfaits (boîtes à outils, squelettes 'application, générateurs d'interface) par conséquent on a besoin d'un cadre de pensée.

L'architecture est abstraite, elle décrit des composants et leurs relations sans présumer de leur réalisation. Il existe des architectures éprouvées qui sont des modèles de référence quelques exemples sont ulustrés dans les **Figures 2-40, 41 et 42.**

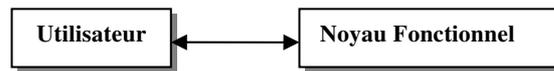


Figure 2-40 A : Architecture fondamentale :

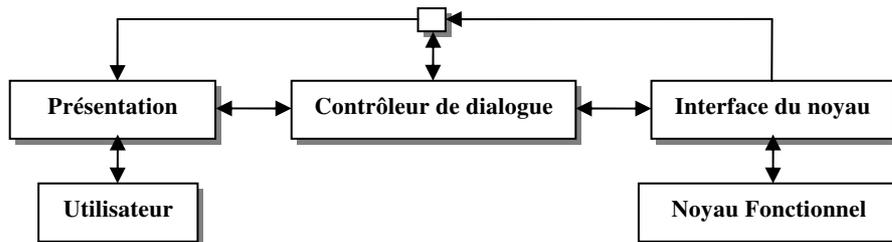


Figure 2-41 : Architecture de Seeheim.

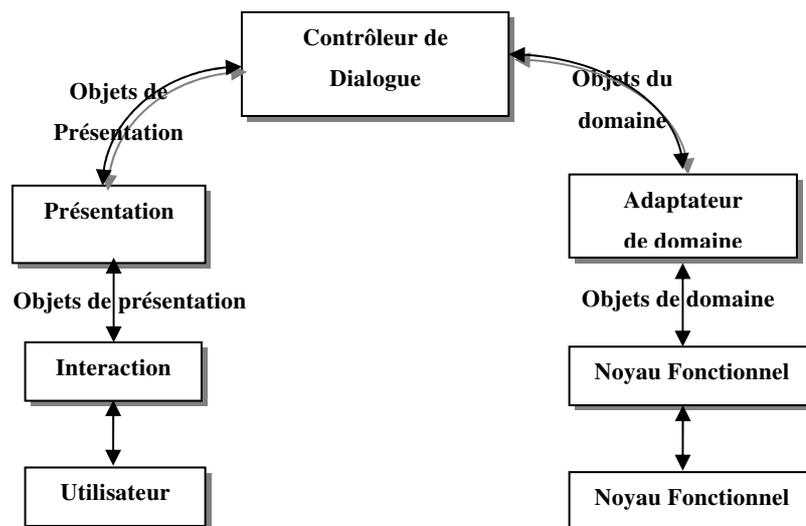


Figure 2-42 : Architecture de l'Arch :

2.9 Conclusion :

Le développement des systèmes microprogrammes à base des microcontrôleurs contient plusieurs points forts et avantages, il suffit d'examiner la spectaculaire évolution et leur présence dans les différents domaines de notre vie quotidienne.

D'une part le microcontrôleur intègre dans un seul et même package plusieurs éléments, il en résulte une diminution de l'encombrement de matériel et de circuit imprimé et Parmi les conséquences de cette intégration la simplification du tracé des circuits imprimé puisqu'il n'est plus nécessaire de véhicule des bus d'adresse et de données d'un autre composant (mémoire par exemple).

D'autre part le développement des modules complémentaires pour les microcontrôleurs comme RTC, LCD, les circuits de communication (MAX485),...etc, interconnectés par des bus comme I2C a facilité le développement de ce genre de système de commande et a réduit les coûts à plusieurs niveaux.

Les réseaux de communication industriels comme le RS485 ont rendu le développement des interfaces homme machine (IHM) plus efficaces et moins coûteux par le tirage d'un seul câble à une seule paire pour plus de 1000 mètres.

CHAPITRE 03

**La Conception Du
Système (Hardware & software).**

Chapitre 3 : La Conception Du Système (Hardware & software)

3.1 Introduction :

Les systèmes micro-programmé à bas de microcontrôleur fonctionnent généralement à faible tension de 0 à 5 Volts, et comme le milieu industriel est très nuisible de coté électromagnétique, ce genre de système doit être bien protégé contre les champs électromagnétiques (EMF) surtout ceux générés par les moteurs électriques.

Dans ce chapitre nous allons présenter les critères sur les quelles notre système est développé pour fonctionner dans un milieu industriel sous une grande influence de l'EMF et exécuter des opérations locales au niveau de la machine et d'autres opérations à distance.

Le microcontrôleur est l'élément crucial du système pour cela le choix de cet item se fait avec une grande précision, d'une façon ou le microcontrôleur choisi ne doit pas être ni sur dimensionné pour éviter le gaspillage et le coût très élevé de la solution, ni sous dimensionné.

Cette opération de choix passe après les calculs et la détermination de tous les modules intégrés.

3.2 Définition de cahier de charges :

Ce paragraphe définit les attendus d'un système de comptage de production sur plusieurs machines et le rendement des agents sur chaque machine, avec la gestion des commandes de production à distance via un seul ordinateur, la distance entre l'ordinateur et les machines est plus de deux cent mètres (200 m).

Le système se compose des parties suivantes :

- 1 - Un compteur pour chaque machine (environ 20 machines).
- 2- logiciel d'interface, de configuration et de gestion.
- 3- Réseau d'alimentation et de communication entre les compteurs et l'ordinateur de gestion.

3.2.1 Cahier de charges du compteur :

Le compteur doit reprendre à ce qui suit :

- Il comporte :
 - Un moyen de saisie des données.
 - Un moyen d'affichage et de dialogue.
 - Des Lampes témoins pour démontrer l'état du compteur.
- avec les fonctionnalités suivantes :
 - la possibilité de by-passer le compteur en cas de besoin.

- Comptage de la production en état de service ou en état de by-pass.
- Bloquer la machine en cas ou la commande est atteinte.
- pointage de l'agent (heures de début et de fin de travail).
- comptage et affichage du temps de travail réel.
- saisie du code de l'agent et la rotation de l'équipe (nombre d'agents est plus de cent (100) et le nombre des équipes est de trois (03))
- la possibilité de sauvegarder les dix derniers accès des agents avec leurs informations de production sur le compteur.
- la possibilité de consulter la production localement ou à distance
- la possibilité de remise à zéro du compteur localement ou à distance
- Le réglage locale ou à distance de :
 - le temps réel (date, heures, minutes).
 - l'adresse de la machine.
 - la commande à produire.

3.2.2 Cahier de charges du logiciel(IHM) :

Le développement d'un logiciel d'interface homme-machine, son premier but est d'assister le gestionnaire pour savoir la situation exacte de chaque machine et l'avancement des commandes à tous moment, il contient trois volets :

- 1- Volet configuration.
- 2- Volet gestion.
- 3- Volet rapport et exportation des données.

I. Le volet Configuration est composé de :

- Insertion et Edition de la liste des Machines, chaque machine est caractérisée par un numéro (adresse de la machine).
- Insertion et Edition de la liste des Agents (code de trois chiffres).

II. Le volet Gestion consiste de :

- Insertion et Edition de la liste des Produits.
- Insertion et Edition de la liste des commandes.
- Lier les commandes aux compteurs physiques.
- Vérification de l'avancement sur chaque compteur.

III. Le volet Reporting comprend :

- 1- Impression d'un état des machines avec leurs caractéristiques.
- 2- Impression d'un état des agents.
- 3- Impression d'un état des produits.
- 4- Impression d'un état des commandes avec leurs états d'avancement.

3.2.3 Cahier de charges du réseau d'alimentation et de communication :

La totalité du System (ordinateur de gestion et les compteurs) doit être alimentés sans coupure.

La distance entre les compteurs et l'ordinateur est de plus de deux cent (200) mètres.

L'ordinateur est le maître et les compteurs sont des esclaves.

3.3 Recensement des entrées sorties du système :

Pour répondre aux exigences du compteur citées dans le paragraphe précédent, nous proposons les solutions suivantes :

Tableau 3-1 : Liste des entrées et des sorties du microcontrôleur.

Item	Le Besoin	La Solution	Nombre des Entrées Numériques	Nombre des Sorties Numériques
1	Moyen de saisie des données	Clavier matriciel 4X4	8	0
2	Moyen d'affichage et de dialogue	Afficheur LCD 16X2	0	6
3	Lampes témoins état du compteur	LED (Service, Arrêt et By-pass)	0	3
4	By-pass du compteur	Interrupteur (ON-OFF)	1	0
5	Comptage de la production	Détecteur de proximité	1	0
6	Bloquer la machine commande atteinte	Relais.	0	1
7	Comptage du temps de travail réel	Contact auxiliaire	1	0
8	Consultation la production localement	Bouton poussoir	1	
9	reset du compteur localement pour agent	Deux boutons poussoir en série	1	0
10	Communication avec circuit temps réel (RTC) via I2C	Module RTC (DS1307)	2	0
Total			15	10

3.1 L'algorithme d'acquisition et de traitement au niveau de la machine :

Les systèmes de commande et de contrôle se basent sur des philosophies et des étapes d'acquisition et de traitement des données pour développer le programme à exécuter par le microcontrôleur afin de répondre aux besoins exprimés dans le cahier des charges. Ces philosophies sont propres à chaque développeur, mais la différence entre une philosophie et une autre reflète l'expérience et la maîtrise du développeur.

À la base du cahier des charges illustré auparavant nous proposons l'algorithme illustré dans la **Figure 3-1**.

Ils existent plusieurs opérations de configuration locale, ces opérations sont enclenchées par la saisie des codes prédéfinis au début (lors de la demande du code agent), ces opérations sont :

- Consultation et changement de l'adresse de la machine, Code 999.
- Consultation et Réglage de la date et l'horloge, Code 998.
- Consultation et Réglage de la commande à produire, Code 997.

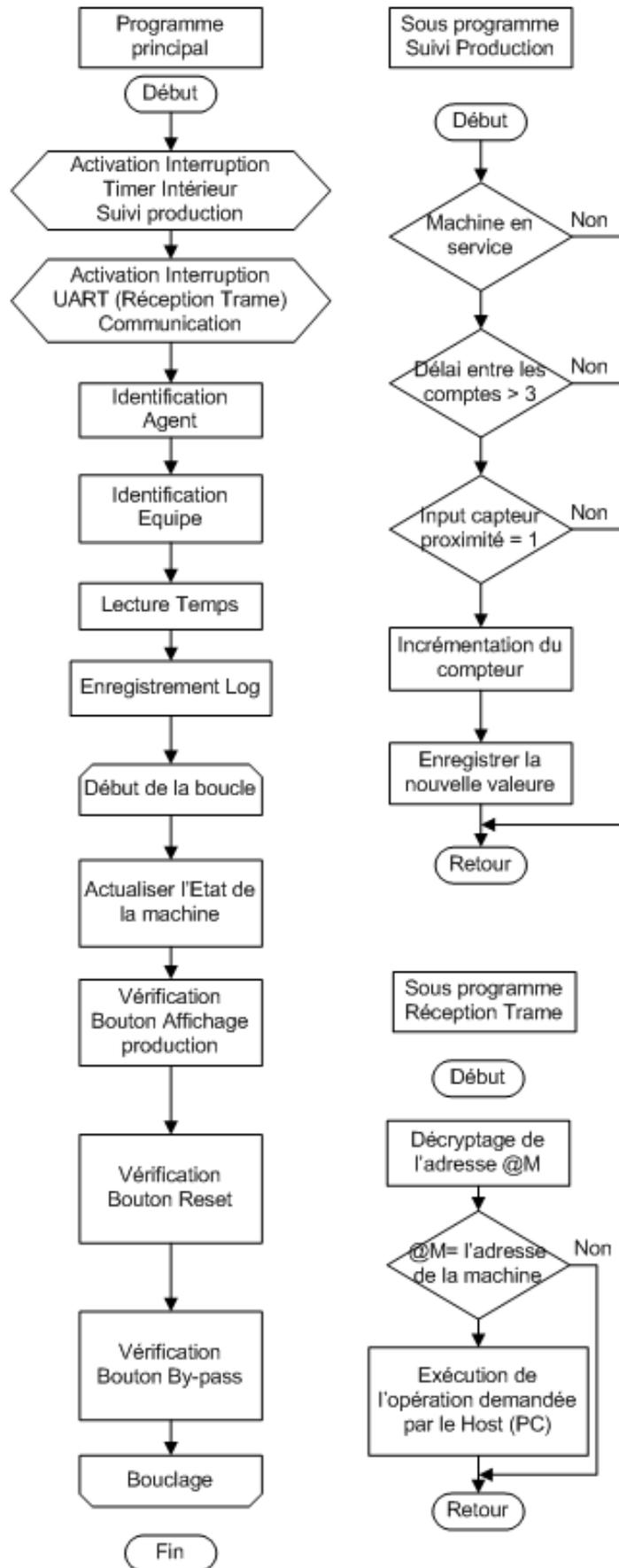


Figure 3-1 : Organigramme général de fonctionnement du compteur.

3.2 Choix de protocole de communication :

La communication dans ce projet se divise en deux types, communication entre circuits intégrés et communication entre le compteur (microcontrôleur) et le Host (PC de gestion).

- **Communication entre circuits intégrés :**

Vue les avantages du bus I2C démontrés dans la partie 2.5 du précédent chapitre nous utilisons ce bus pour interconnecter le module d'horloge (Real Time Circuit –RTC-) avec le micro contrôleur, les fonctionnalités du module et les méthodes de sa communication est bien détaillée dans la partie suscitée.

- **Communication entre le Host (PC) et le compteur (Microcontrôleur) :**

Le support physique de ce type de communication (RS485) et les protocoles standards sont détaillé dans les parties 2.7 et 2.8 du Chapitre précédent.

Par ailleurs, nous allons décrire à ce niveau notre propre protocole de communication.

Pour que le Host (PC Maitre) communique avec un des compteurs (Esclave) il envoie un des messages prédéfinis suivants :

3.2.1 Opération de lecture d'un octet de l'espace mémoire EEPROM :

- Nous avons choisi le format du message de lecture comme suit : « **rXXyyy** » :
 - o r : opération de lecture.
 - o **XX** : adresse du compteur désiré, valeur entre 0 et 32 décimale.
 - o **yyy** : adresse de l'espace mémoire dans le microcontrôleur, valeur entre 0 et 255 décimale selon l'espace mémoire disponible dans le microcontrôleur.

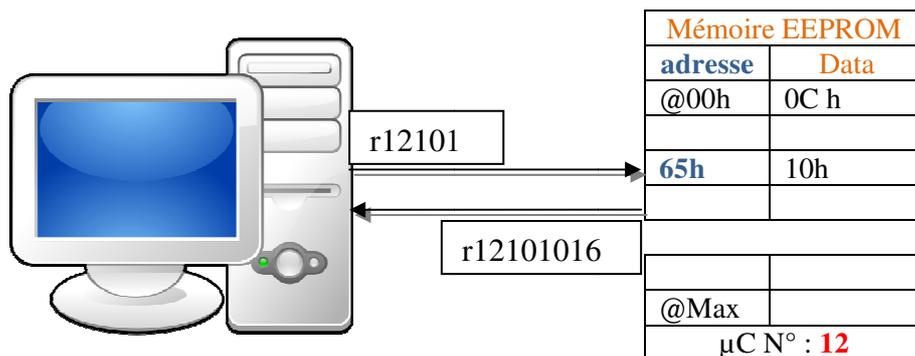


Figure 3-2 : Format d'un message de lecteur.

Si le compteur N° 12 existe sa réponse sera comme présenté dans la **Figure 3-2**

« r12101016 » et la repense est en décimale comme texte.

3.2.2 Opération d'écriture d'un octet dans un espace mémoire EEPROM :

- Nous avons choisi le format du message d'écriture comme suit : « wXXyyyZZ » :
 - o w : opération d'écriture.
 - o **XX** : adresse du compteur désiré, valeur entre 0 et 32 décimale.
 - o **yyy** : adresse de l'espace mémoire dans le microcontrôleur, valeur entre 0 et 255 décimale selon l'espace mémoire disponible dans le microcontrôleur.
 - o **ZZ** : la valeur à écrire dans l'espace mémoire, valeur entre 0 et 255 décimale.

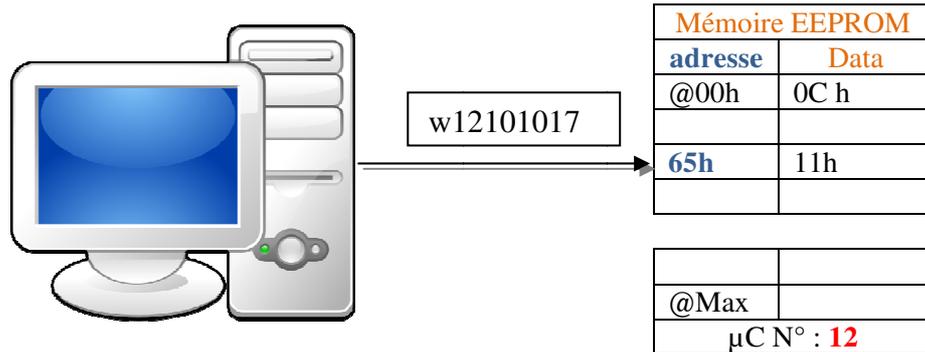


Figure 3-3 : Format d'un message d'écriture.

Si le compteur N° 12 existe va actualiser l'emplacement adressé 101 (65h) par la valeur 17 (11h) opération expliquée dans la **Figure 3-3** et pour vérifier l'accomplissement de l'opération il suffit de relire la même adresse.

Nous avons choisie ces deux opérations parce que tous les évènements sont enregistrés au niveau de l'EEPROM de microcontrôleur et chaque information est enregistrée dans un emplacement de l'EEPROM, le **Tableau 3-2** démontre les variable et leurs adresses.

Tableau 3-2 : Adressage des variables dans l'EEPROM.

Mémoire EEPROM		Description	Mémoire EEPROM		Description
Adresse 0x00h	Data		Adresse 0x00h	Data	
00	0C h	Adresse du compteur	10		
01		La commande à produire dcba :	11		
02		a : les deux petits chiffres	12		
03		d : les trois grands chiffres	13		Numéro auto d'accès (Log)
04			14		La dernière position du Log (n)
05		Le compteur actuel	15		Code Machine Log0
06		d1c1b1a1 :	16		Code Equipe Log0
07		a1: les deux petits chiffres	17		Code Agent Log0
08		d1: les trois grands chiffres	18		Heure Démarrage Log0
09			19		Minute Démarrage Log0
0A			1A		Jour Démarrage Log0
0B			1B		Mois Démarrage Log0
0C			1C		Année Démarrage Log0
0D			1D		Nombre Heure travaillée Log0
0E			1E		Nombre minutes travaillée Log0
0F			1F		Heure Fin Log0

Mémoire EEPROM		Description	Mémoire EEPROM		Description
Adresse 0x00h	Data		Adresse 0x00h	Data	
20		Minute Fin Log0	30		Année Démarrage Log1
11		Compteur de démarrage Log0 (dcba) : a : les deux petits chiffres d : les trois grands chiffres	31		Nombre Heure travaillée Log1
22			32		Nombre minutes travaillée Log1
23			33		Heure Fin Log1
24			34		Minute Fin Log1
25		Compteur de fin Log0 (d1c1b1a1) : a1: les deux petits chiffres d1: les trois grands chiffres	35		Compteur de démarrage Log1 (dcba) : a : les deux petits chiffres d : les trois grands chiffres
26			36		
27			37		
28			38		
29		Code Machine Log1	39		Compteur de fin Log1 (d1c1b1a1) : a1: les deux petits chiffres d1: les trois grands chiffres
2A		Code Equipe Log1	3A		
2B		Code Agent Log1	3B		
2C		Heure Démarrage Log1	3C		
2D		Minute Démarrage Log1	3D		Code Machine Log2
2E		Jour Démarrage Log1	3E		Code Equipe Log2
2F		Mois Démarrage Log1	3F		Code Agent Log2

Jusqu'à l'enregistrement du dixième accès, donc l'espace mémoire peut être calculé comme suit :

$$\text{Espace EEPROM} = (10 \times 20) + 21 = 221 \text{ Octets}$$

3.3 Choix de microcontrôleur :

3.3.1 Récapitulatif des exigences sur le microcontrôleur

Après le recensement de toutes les fonctionnalités et caractéristiques attendues du système développé dans les sections précédentes de ce chapitre nous récapitulons les critères du microcontrôleur comme suit :

Tableau 3-3 : Tableau récapitulatif des caractéristiques du µC à choisir.

Critère	Exigence selon l'étude	Observations
Package	A pin (PDIP)	Pour facilité le montage
Architecture	Harvard /8bits	
Fréquence	En minimum 8 Mhz	16 Mhz c'est meilleur
Oscillateur	Intérieur ou extérieur	Préférable qu'il soit intégré
Nombre des I/O numériques	Plus de 25	
Espace mémoire pour le programme	Plus de 20 KB	Selon le programme principal et les sous programmes.
Espace mémoire de sauvegarde EEPROM	Plus de 221 Octets	Pour assurer la sauvegarde de 10 accès selon le cahier des charges
Voltage de service	5Volts	Préférable
Communication avec les circuits	I2C	
Communication les autre périphériques	UART	
Flashage du programme	ICSP	Pour utiliser le PICKIT
Cycle de vie	En production	Pour éviter son remplacement
Température de service	0 à +70°C	

Le microcontrôleur disponible sur le marché et le plus favorable pour répondre aux critères et aux exigences précédentes est le PIC 18F45K22.

3.3.2 Caractéristiques du PIC 18F45K22 :

Le PIC18F45K22 est un microcontrôleur, avec de hautes performances de calcul et une forte endurance, mémoire de programme Flash. En plus de ces caractéristiques, il introduit des améliorations de conception qui font de ces microcontrôleurs un choix logique pour de nombreuses applications sensibles de haute performance et de puissance.

La **Figure 3-4** démontre les signaux d'entrées et sorties de microcontrôleur sélectionné pour notre application.

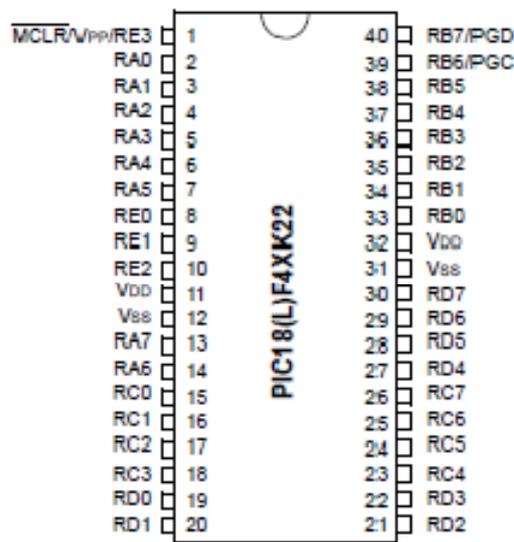


Figure 3-4 : Signaux du Microcontrôleur PIC18F45K22.

Caractéristiques du PIC18F45K22 :

- CPU RISC Haute Performance
- Gestion de faible puissance externe avec nanoWatt XLP™
- Mode Sleep 100nA, Typ.
- Timer Watchdog 500nA Typ.
- 500 nA à 32 kHz typique Timer 1 Oscillateur
- Structure de l'oscillateur flexible
- Precision du bloc oscillateur interne 16 MHz
- Quatre modes de Quartz jusqu'à 64 MHz
- Deux modes d'horloge externe jusqu'à 64 MHz
- PLL (Phase Lock Loop)
- Oscillateur secondaire utilise le Timer 1 à 32 kHz

- Moniteur Fail-safe Clock, permet un arrêt sécurisé des périphériques Horloge Stops
- Oscillateur Start-up à deux vitesses
- Fonctionnement 5.5V complet
- Auto-programmable sous contrôle logiciel
- Power-on Reset (POR), Power-up Timer (PWRT) et Oscillateur Timer Start-up (OST)
- Brown-out Reset (BOR) programmable
- Timer Watchdog (WDT) étendue avec oscillateur et Software Enable
- Code de protection programmable
- Programmation Série In-Situ (ICSP™) via deux broches
- Applications : Electronique Grand Public, Computers & Computer Peripherals, Automobile

Pour plus d'information sur ce microcontrôleur consultez la fiche technique dans les annexes.

3.4 Conception du système d'acquisition (Compteur) :

3.4.1 Alimentation :

Pour assurer le bon fonctionnement du système, il faut fournir une alimentation suffisante pour tous les éléments du système comme suit :

- Détecteur de proximité PNP qui fonction avec une tension entre 6 et 36 VDC et 200mA.
- Relais avec une bobine qui fonction avec 12 VDC et 30mA
- Le reste du system (microcontrôleur et ses accessoires) fonction à 5VDC et 120mA et pour assurer cette alimentation stable de 5VDC nous avons utilisé le régulateur de tension **L7805C-V**.

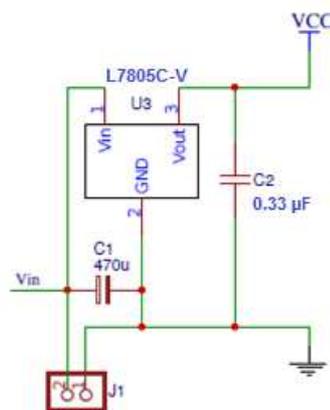


Figure 3-5 : Circuit d'alimentation.

Alors le système doit être alimenté par $V_{in} \geq 12VDC$ et **350 mA**.

Pour améliorer la résistance du circuit à la force électromagnétique il faut installer un condensateur de 0.33 μ F à la sortie du régulateur.

3.4.2 Les Entrées Numériques (Digital Inputs) :

Pour stabiliser le signale d'entrée et éviter le fennomane de « floating Pin », nous utilisons des résistances de tirage « pull down résistance » au niveau de chaque entrée numériques comme suit :

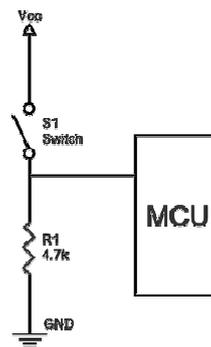


Figure 3-6 : Résistances de tirage « Pull Down Resistor ».

Les résistances de tirage « Pull Down Resistor » ne sont pas spécifiques et la valeur typique est de 4.7 K Ω mais peut varier d'une application à l'autre, dans notre cas est de 10 K Ω pour éviter tous types de fuite ou de consommation de courant.

Selon le cahier des charges nous avons les entrées numériques suivantes :

Tableau 3-4 : les entrées numériques.

Entrée	Port	Détails	Observations
IN1	RB0	Le changement de 0 à 5 VDC à l'entrée du port indique une incrémentation de production.	Opto-coupleur pour protection
IN2	RC0	La présence de 5VDC à l'entrée du port indique que la machine est en service.	Opto-coupleur pour protection
IN3	RC1	La présence de 5VDC au bout du port force l'arrêt de la machine pour changer l'utilisateur.	
IN4	RC2	La présence de 5VDC au bout du port active l'affichage de la production sur le LCD.	
IN5	RD0	La présence de 5VDC à l'entrée du port indique l'état actif du by-pass de la machine.	

Les Opto-coupleurs sont utilisés pour les entrées qui viennent de l'armoire de la machine pour éviter tout éventuel court circuit ou une haute tension.

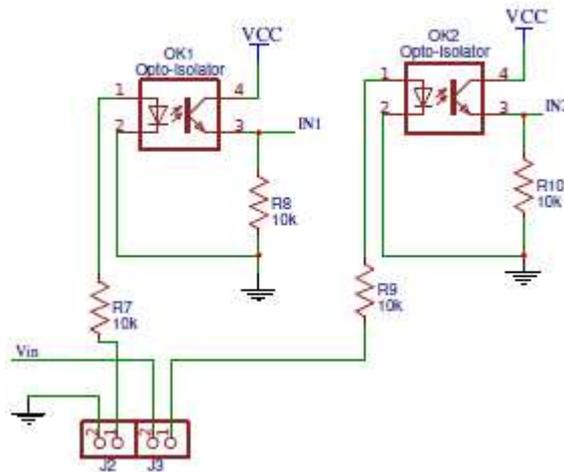


Figure 3-7 : Circuit de conditionnement des entrées (Digital inputs).

3.4.3 Les Sorties Numériques (Digital Outputs) :

Dans notre conception nous avons prévus les sorties suivante :

Tableau 3-5 : Les sorties numériques.

Sortie	Port	Détails	Observations
OUT1	RD1	Commande le blocage et la mise en disponibilité de la machine pour production avec l'allumage de la LED verte	
OUT2	RD2	Commande l'allumage de la LED Orange indiquant le by-pass de la machine	
OUT3	RD3	Commande l'allumage de la LED rouge indiquant l'arrêt de la machine quand la commande est atteinte.	

Les valeurs des résistances peuvent être fixées facilement par l'utilisation de la loi des mails et la loi d'ohm.

Exemple :

La valeur de la résistance d'alimentation d'une LED ?

Selon le graphe présenté dans la **Figure 3-8**, les LED rouge, orange, jaune (ambre) s'activent entre 1.55 V et 2 V.

Verte standard (vert clair) : 1.8V à 2.2V.

Vout : 5Volt

Alors $V = R \cdot i$

$R = V/i$

$R = (5 - 1.8)/8$

$R = 400 \rightarrow$ Valeur standard $R = 470$

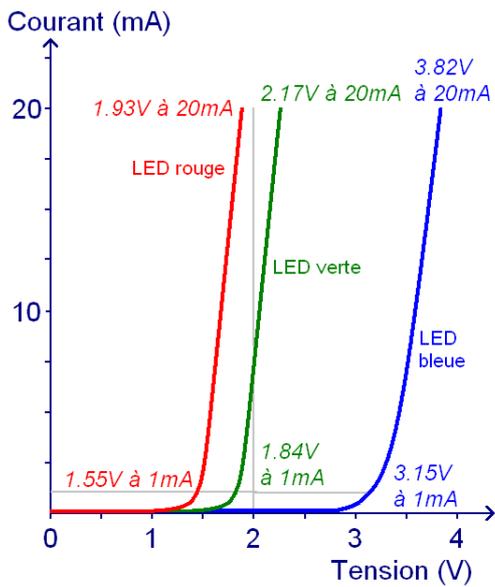


Figure 3-8 : Caractéristiques électrique des LED

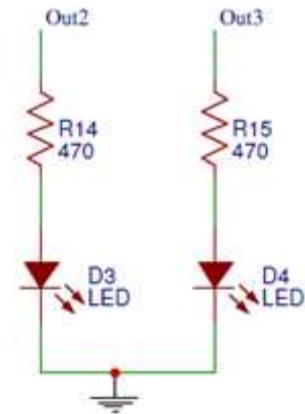


Figure 3-9 : Circuit sorties numérique (commande LED)

- **Commande d'un relais électromécanique**

Un relais peut commuter une tension qui est différente de la tension de commande. Dans notre projet la tension de commande est de 5V et le relais fonctionne avec 12VDC.

pour compenser cette différence de voltage nous utilisons un transistor NPN.

Les résistances R11 et R12 sont installées pour générer un voltage V_{be} de 0.6 Volte, lors de présence d'un voltage de 5VDC à la sortie Out1, pour permettre un passage de courant I_{ce} entre le collecteur et l'émetteur du transistor.

La diode D1 est placée en parallèle du relais (en sens inverse) pour protéger le transistor contre les surtensions provoquées par la bobine du relais lors de sa mise hors fonction.

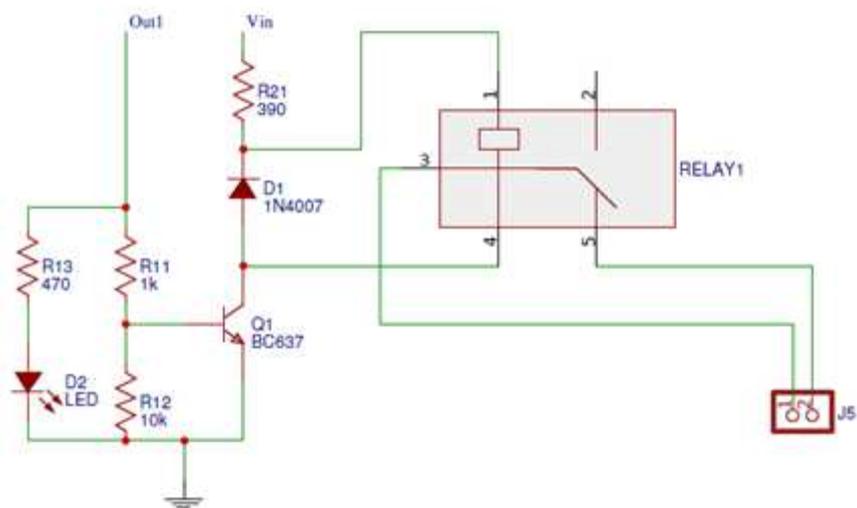


Figure 3-10 : Circuit sorties numérique (commande Relais)

3.4.4 Afficheur Alphanumérique LCD 2x16 :

Afficheur est constitué de deux lames de verre, distantes de 20 μm environ, sur lesquelles sont dessinées les mantisses formant les caractères. L'espace entre elles est rempli de cristal liquide normalement réfléchissant (pour les modèles réfléchitifs). L'application entre les deux faces d'une tension alternative basse fréquence de quelques volts (3 à 5 V) le rend absorbant. Les caractères apparaissent sombres sur fond clair. N'émettant pas de lumière, un afficheur à cristaux liquides réfléchitif ne peut être utilisé qu'avec un bon éclairage ambiant. Sa lisibilité augmente avec l'éclairage. Les modèles transmissifs fonctionnent différemment : normalement opaque au repos, le cristal liquide devient transparent lorsqu'il est excité ; pour rendre un tel afficheur lisible, il est nécessaire de l'éclairer par l'arrière, comme c'est le cas pour les modèles rétro-éclairés.

- L'intensité de l'afficheur est réglée par le rapport entre R1 et R2.
- pour allumer l'éclairage d'arrière il suffit d'appliquer une tension à la résistance R22.
- pour afficher un texte sur l'afficheur il faut configurer les ports du microcontrôleur liés au pin de l'afficheur (RS, E, DB4, DB5, DB6 et DB7) et le sous programme LCD prend en charge l'affichage du texte.

dans notre cas les pin sont configurés respectivement comme suit :

(D0, D1, D3, D4, D5).

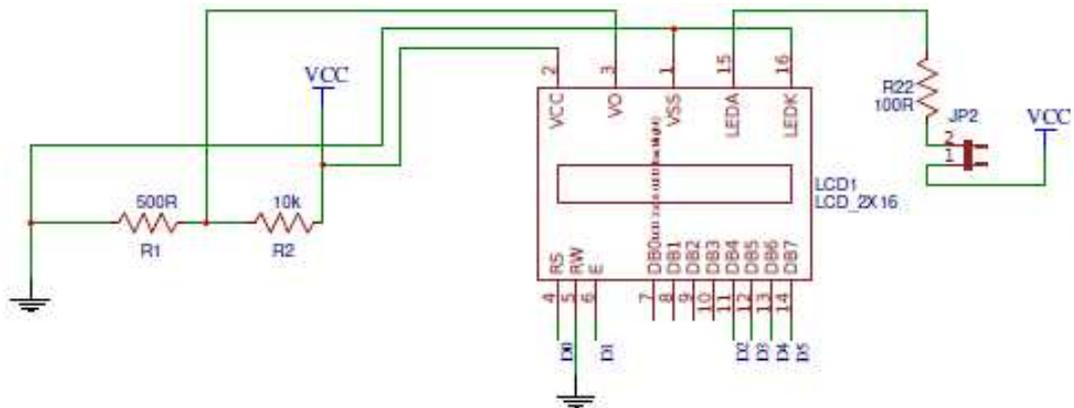


Figure 3-11 : Circuit d'interconnexion de l'afficheur LCD 2x16 avec le μC .

3.4.5 Clavier matriciel 4x4 (16 touches) :

Ce type de clavier comprend 16 touches disposées en 4 lignes et 4 colonnes interconnectées selon la **Figure 3-10**, L'appui sur une touche fait communiquer une ligne avec une colonne.

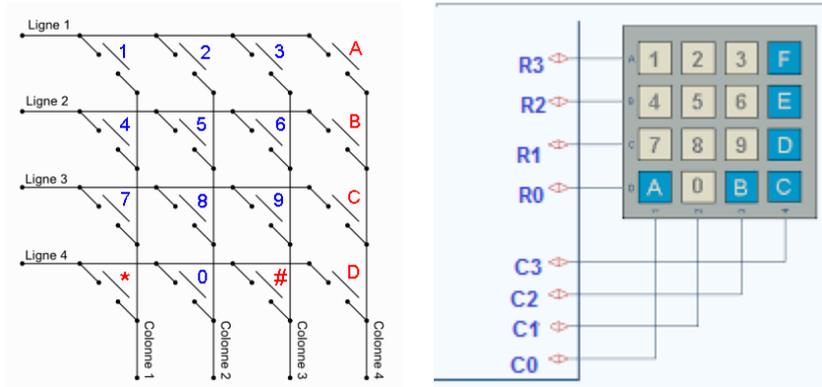


Figure 3-12 : Clavier matriciel 4X4.

Les lignes sont des sorties. Les colonnes sont des entrées maintenues au niveau haut par le microcontrôleur (**Figure 3-11**).

Le système envoie par balayage un niveau bas sur chaque ligne (1 seule à la fois) et balaye les colonnes en lecture.

Quand il lit un niveau bas, c'est que la colonne est reliée par une touche appuyée à la ligne qui est basse à ce moment.

On n'a pas besoin de programmer ce balayage : c'est la bibliothèque qui le fait.

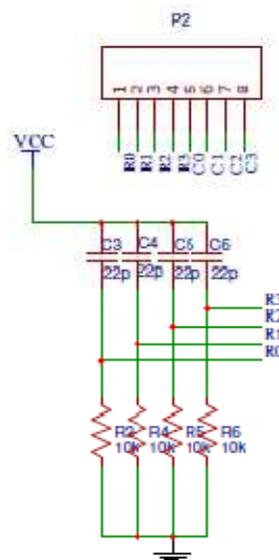


Figure 3-13 : Circuit d'interconnexion du Clavier 4x4 avec le µC.

3.4.6 Module d'horloge (Real Time Circuit –RTC-):

Les pins I2C du module RTC (à savoir : SCL et SDA présentés dans la **Figure 3-12**) doivent être attachés avec les pins I2C du microcontrôleur (RC3 et RC4).

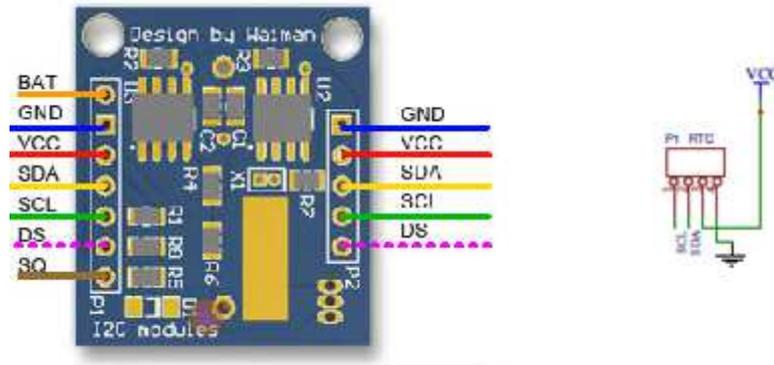


Figure 3-14 : Circuit d'interconnexion du module RTC avec le µC via I2C.

3.4.7 Communication avec le Host (PC) :

➤ Coté Computeur

La communication entre les compteurs (microcontrôleur) et le PC ce fait via un adaptateur RS485 pour convertir le signal physique A-B au signal TTL exploitable par le microcontrôleur.

Cet adaptateur est un circuit électronique appelé MAX485 interconnecté avec le microcontrôleur selon la **Figure 3-15**.

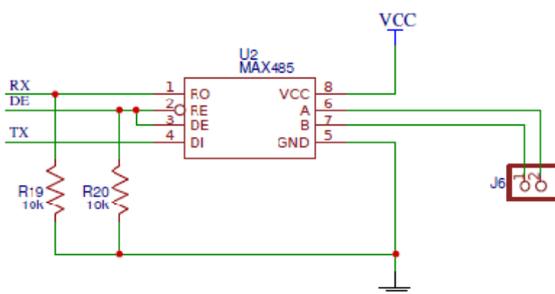


Figure 3-15 : Circuit de communication MAX485



Figure 3-16 : Convertisseur USB –RS485.

➤ Coté Ordinateur

Il faut utiliser un convertisseur USB –RS485 (**Figure 3-16**) et après l'installation du driver de ce convertisseur au niveau de l'ordinateur de commandement (Host) un port série virtuel sera créé et pour communiquer avec le UART du micro contrôleur il suffit de configurer le port avec la même configuration du UART.

3.5 Le circuit global du système d'acquisition (Compteur) :

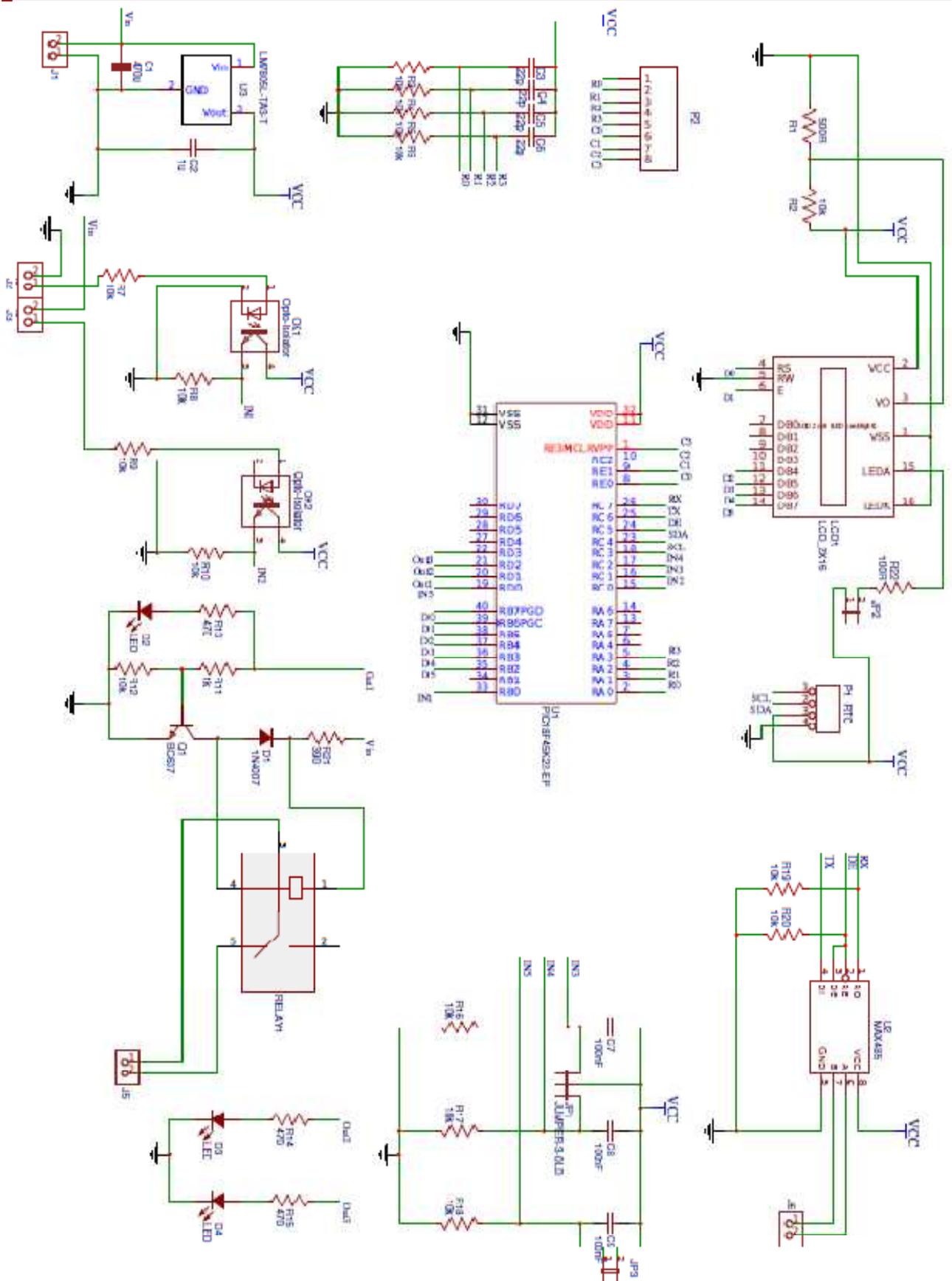


Figure 3-17 : Circuit global du compteur.

3.6 Conception d'une application d'interfaçage Homme – Machine IHM :

Une application au niveau de l'ordinateur de commande et de gestion (Host) doit prendre en charge la communication avec les compteurs pour rendre la collecte et la mise à jour de la base de données fiables, faciles et rapides.

3.6.1 Configuration Manuel :

Des opérations de configuration manuelle avant le développement d'une application sont indispensables.

- nous commençons par l'installation et la configuration du DRIVER du convertisseur USB-RS485.
- en suite il faut vérifier la création d'un nouveau port série de communication au niveau de Gestionnaire de périphériques comme illustrer dans la figure suivante :

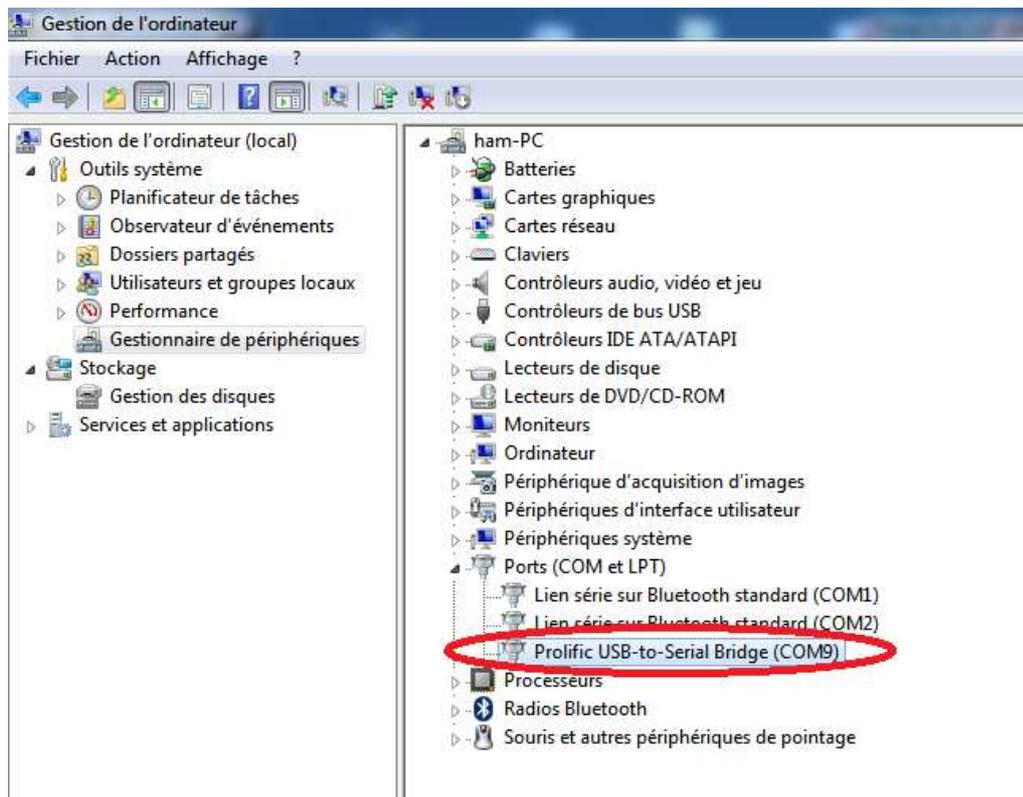


Figure 3-18 : Gestionnaire de périphériques.

- En fin il faut donner la même configuration des UART des microcontrôleurs à notre port de communication comme suit :

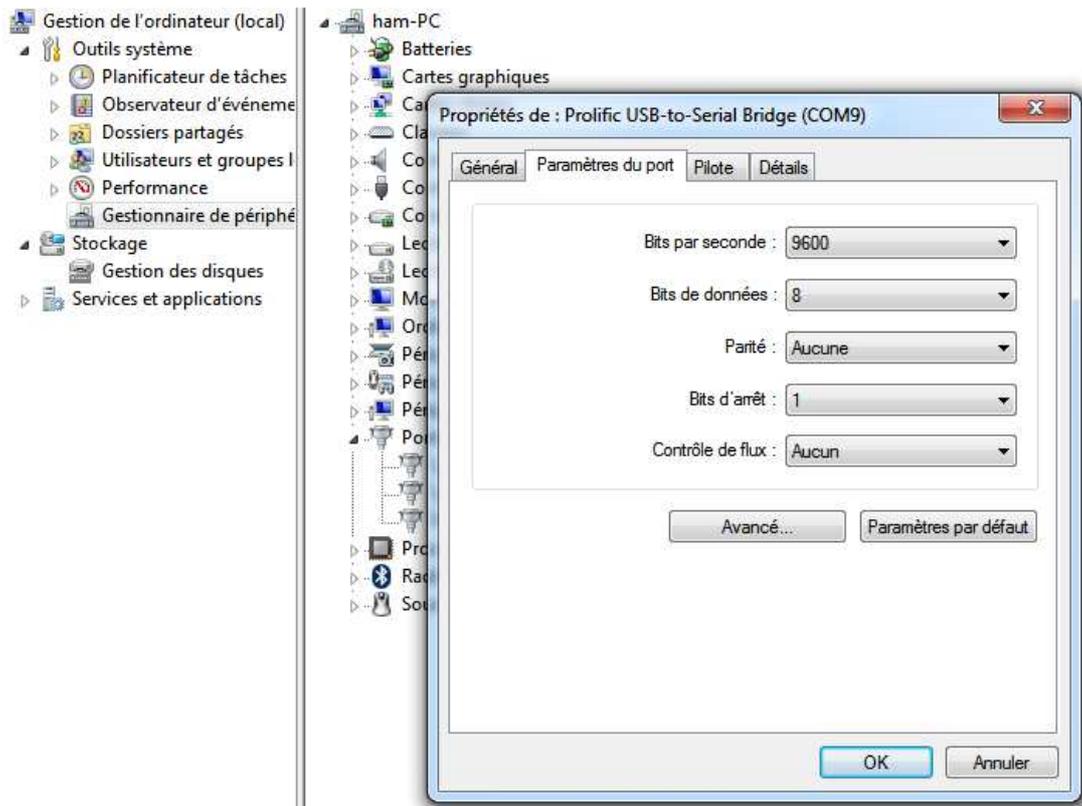


Figure 3-19 : interface de configuration d'un port série.

À ce moment la voie de communication entre l'ordinateur (Host) et les compteurs (microcontrôleurs) est ouvert reste l'envoi et la récupération des messages.

3.6.2 La communication et l'actualisation des données coté PC :

Pour mettre à jour la base de données au niveau du PC une série de requêtes est envoyées périodiquement aux compteurs installés sur les machines.

L'organigramme présenté dans la figure 3-20 montre les étapes à suivre par le PC afin de récupérer les données sauvegardées sur chaque machine.

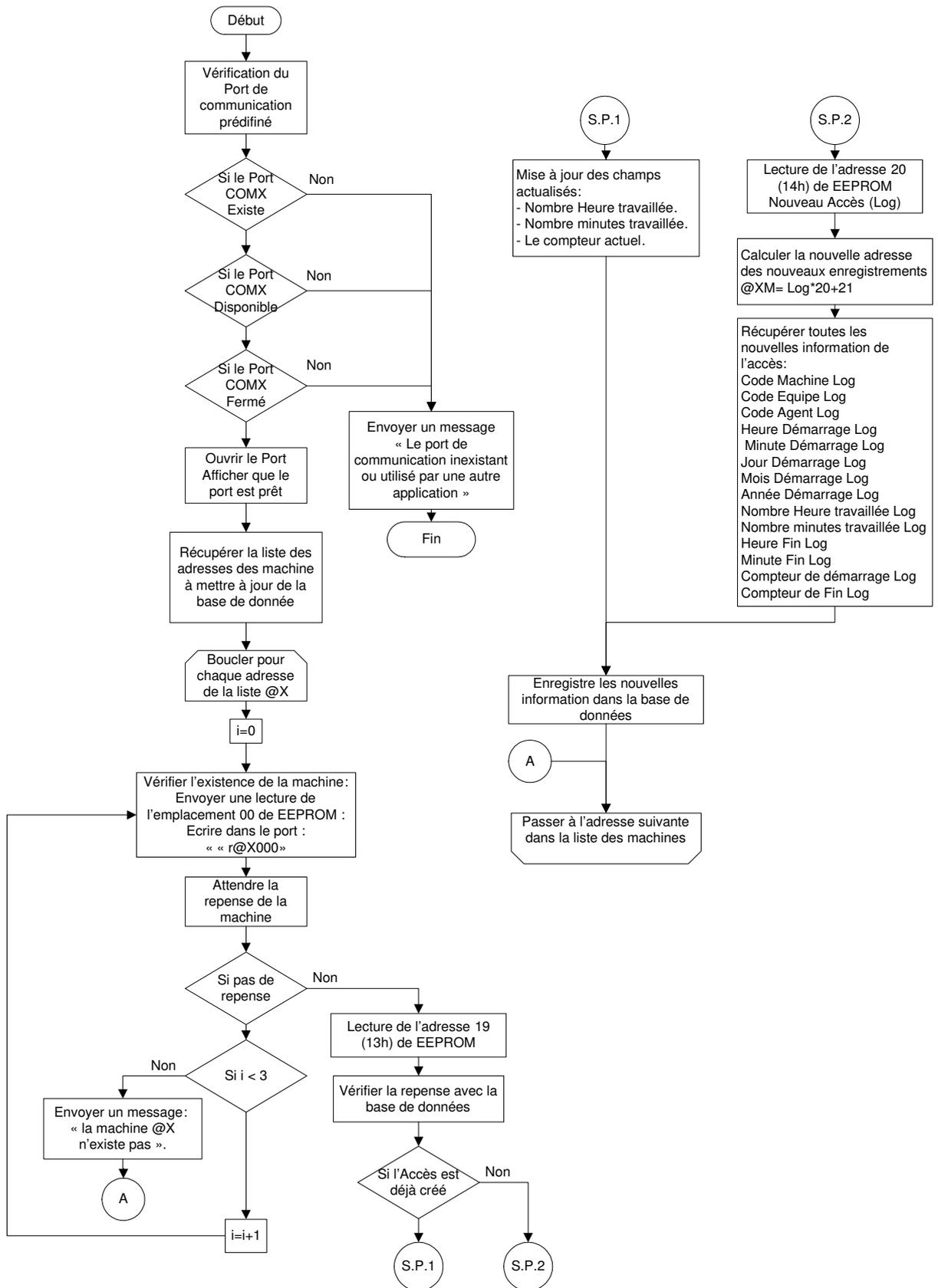


Figure 3-20 : Organigramme de communication et d’actualisation des données coté PC.

3.7 Conclusion :

Tout au long de ce chapitre, nous avons passé par différentes étapes de conception d'un système de gestion de production à partir d'un cahier des charges réparti en deux parties :

➤ Conception du Compteur :

Dans cette partie nous avons fait le design et la conception d'un système d'acquisition de données basé sur le microcontrôleur PIC18F45K22 vu qu'il répond à tous nos exigences.

Le circuit global se compose de plusieurs circuits :

- Des circuits de conditionnement des signaux ou d'entrées ou de sortie (LED et Relais).
 - Des circuits d'interface avec les éléments d'affichage et de dialogue comme le Clavier matriciel 4X4 et l'afficheur 2x16 LCD.
 - Un circuit de communication basé sur le MAX485.
 - Développement d'un algorithme pour le microcontrôleur.
- Elaboration d'un algorithme pour la conception d'une application d'interfaçage entre le gestionnaire et les compteurs à travers un IHM pour lui faciliter la gestion et le contrôle des compteurs à distance via le réseau RS485.

CHAPITRE 04

Réalisation

Et

Mise En Service.

Chapitre 4 : Réalisation Et Mise En Service :

Dans ce chapitre nous allons démontrer les méthodes, les outils et les moyens utilisés pour réaliser notre solution proposée et convertir la conception théorique détaillée dans le chapitre précédant à un produit concret.

4.1 Les outils utilisés pour la conception des circuits électroniques :

Pendant la conception des circuits électronique nous avons utilisé une solution en ligne et gratuite appeler EasyEDA sur le lien : www.easyeda.com

Cette application a plusieurs avantages comme :

- l'accès à l'espace de travail de n'importe quel endroit il suffit d'avoir une connexion à l'internet.
- Garder une copie du projet contre tout endommagement.
- une base de données contienne des milliers de composants standards et d'autres personnalisés.
- un espace de travail personnel ou en public pour partager les taches.

Et beaucoup d'autres avantages.

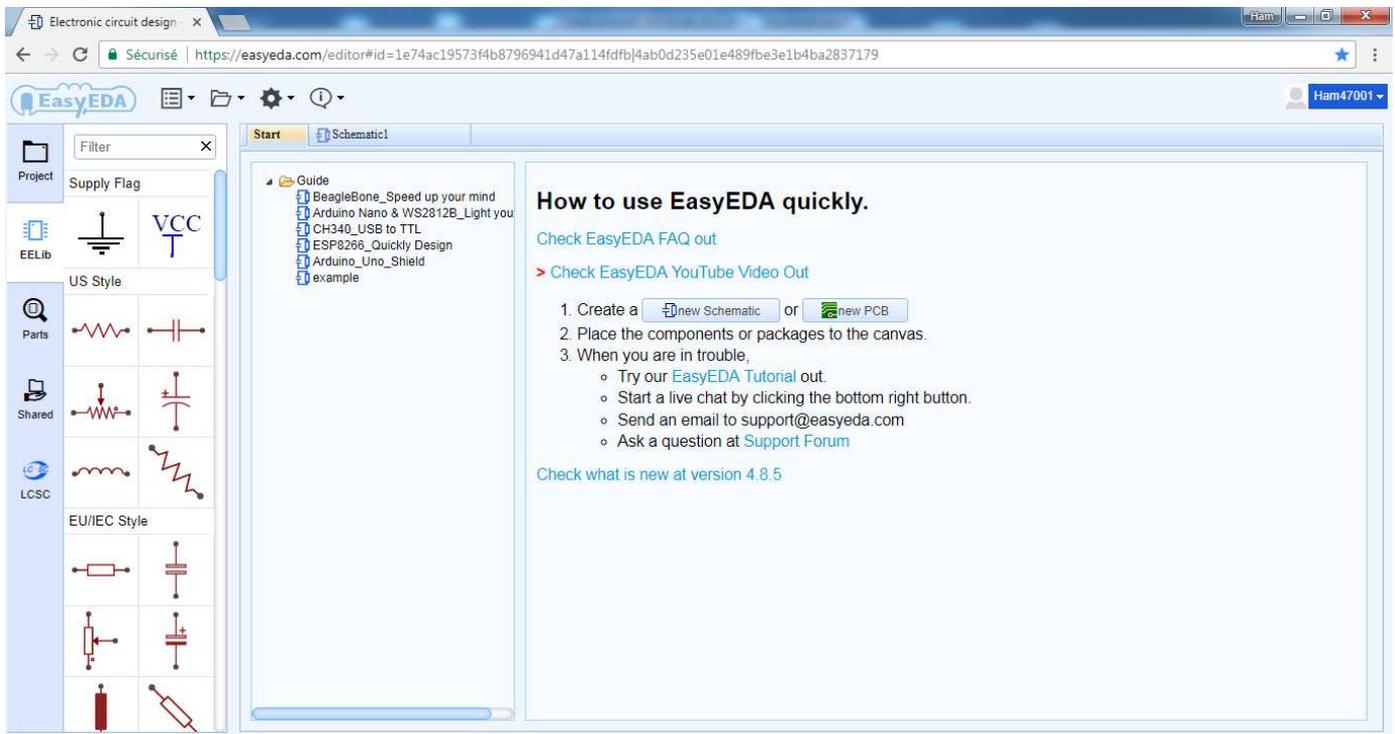


Figure 4-1 : Interface d'accueil de l'application EasyEDA.

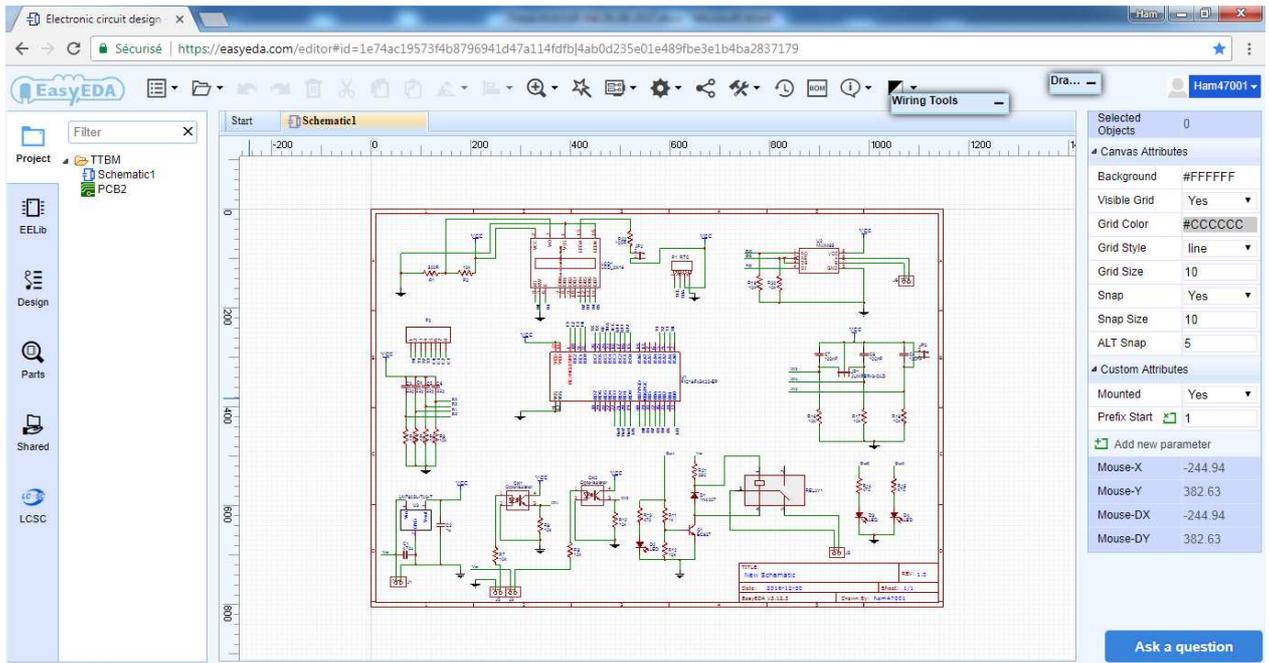


Figure 4-2 : Espace de conception des circuits de EasyEDA.

Pour travailler avec cette application il suffit de créer un compte et en suite créer un projet et c'est au développeur (Ingénieur) le reste.

4.2 Outils et méthodes de conception des circuits imprimés (PCB) :

Après la finalisation de la conception du circuit électronique (schéma) et après la vérification de toutes les connexions, il suffit de cliquer sur (Convert Project to PCB)

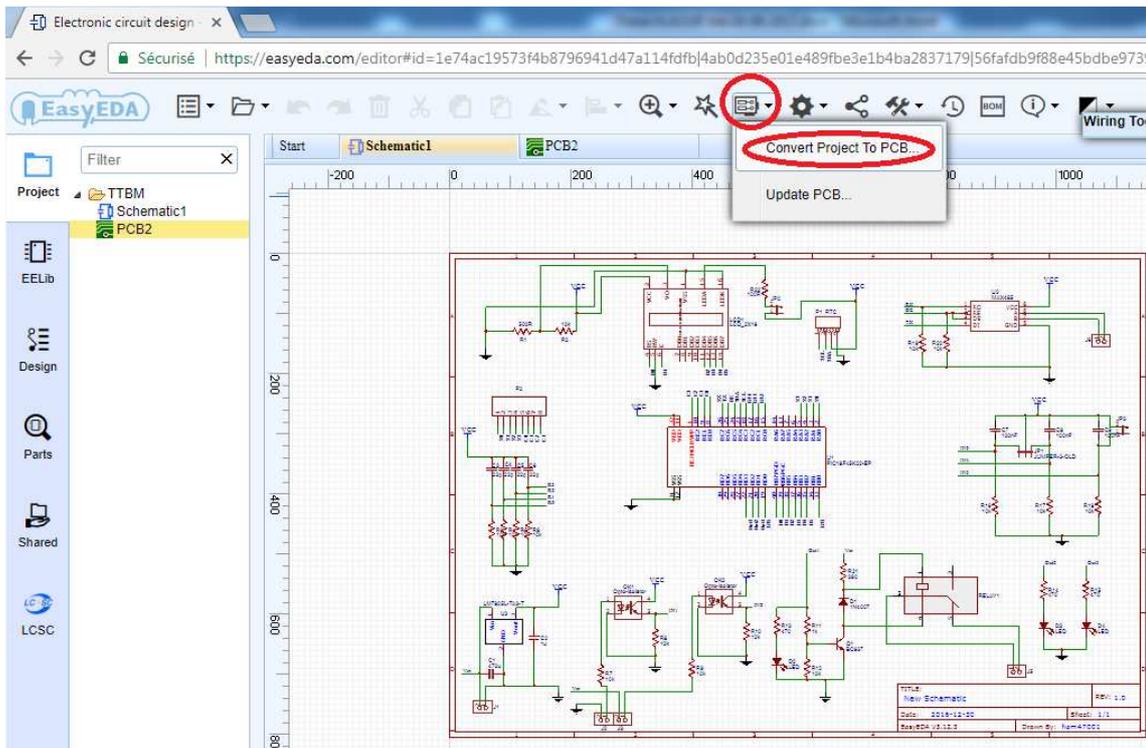


Figure 4-3 : Menue de conversion d'un schéma à un circuit imprimé.

Et avec un peu d'assistance de l'application le circuit imprimé sera prêt.

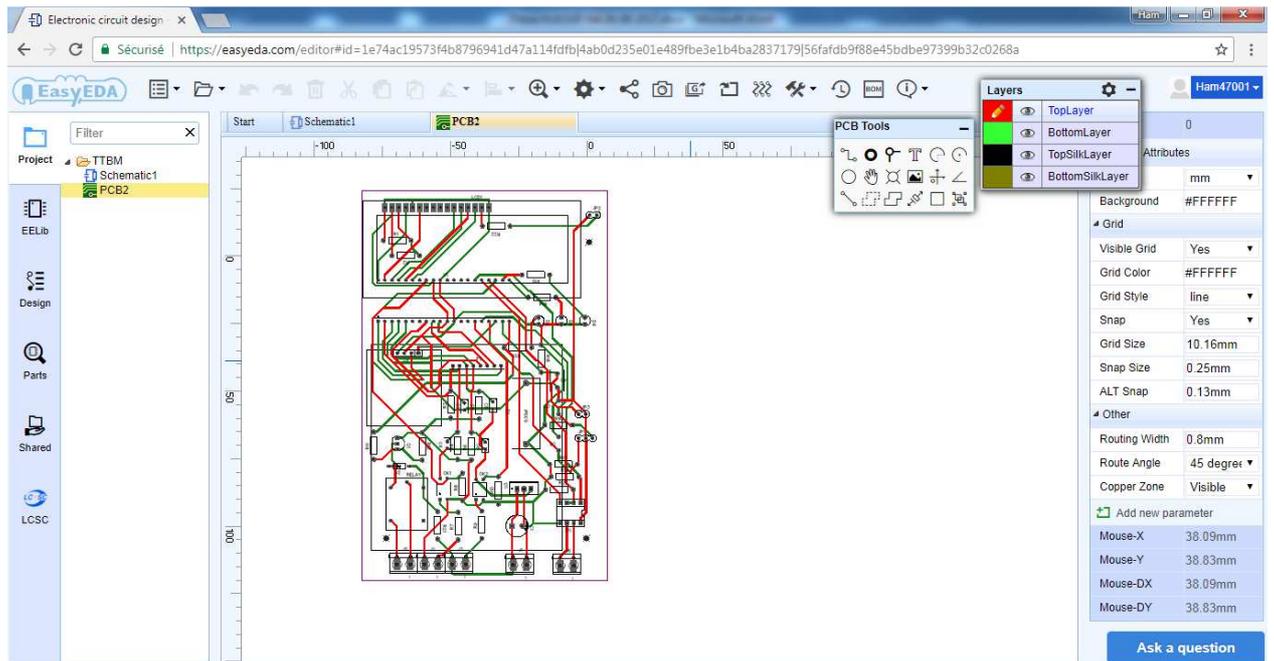


Figure 4-4 : Espace d'édition des circuits imprimés.

Et avec l'utilisation des outils d'exportation nous pouvons facilement extraire notre projet et l'envoyer pour réalisation par des professionnels.

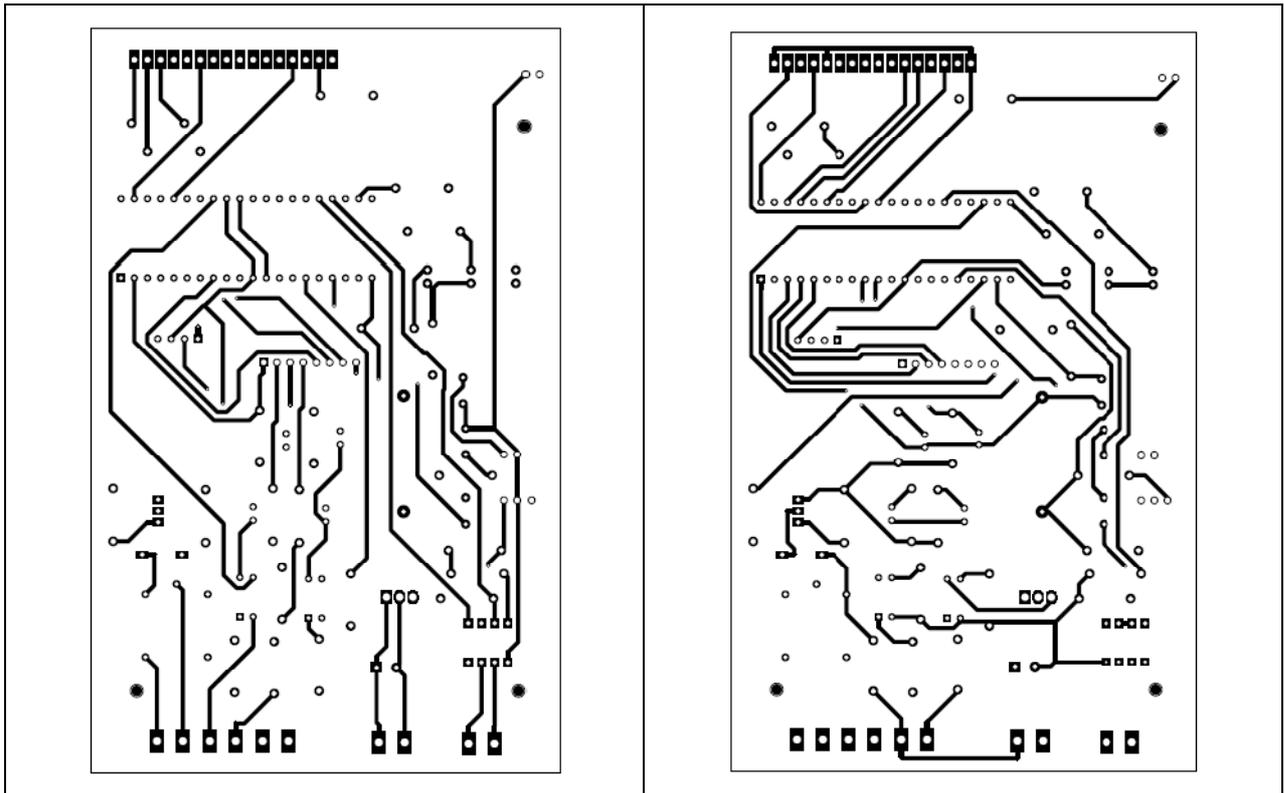


Figure 4-5 : Schéma final du circuit imprimé.

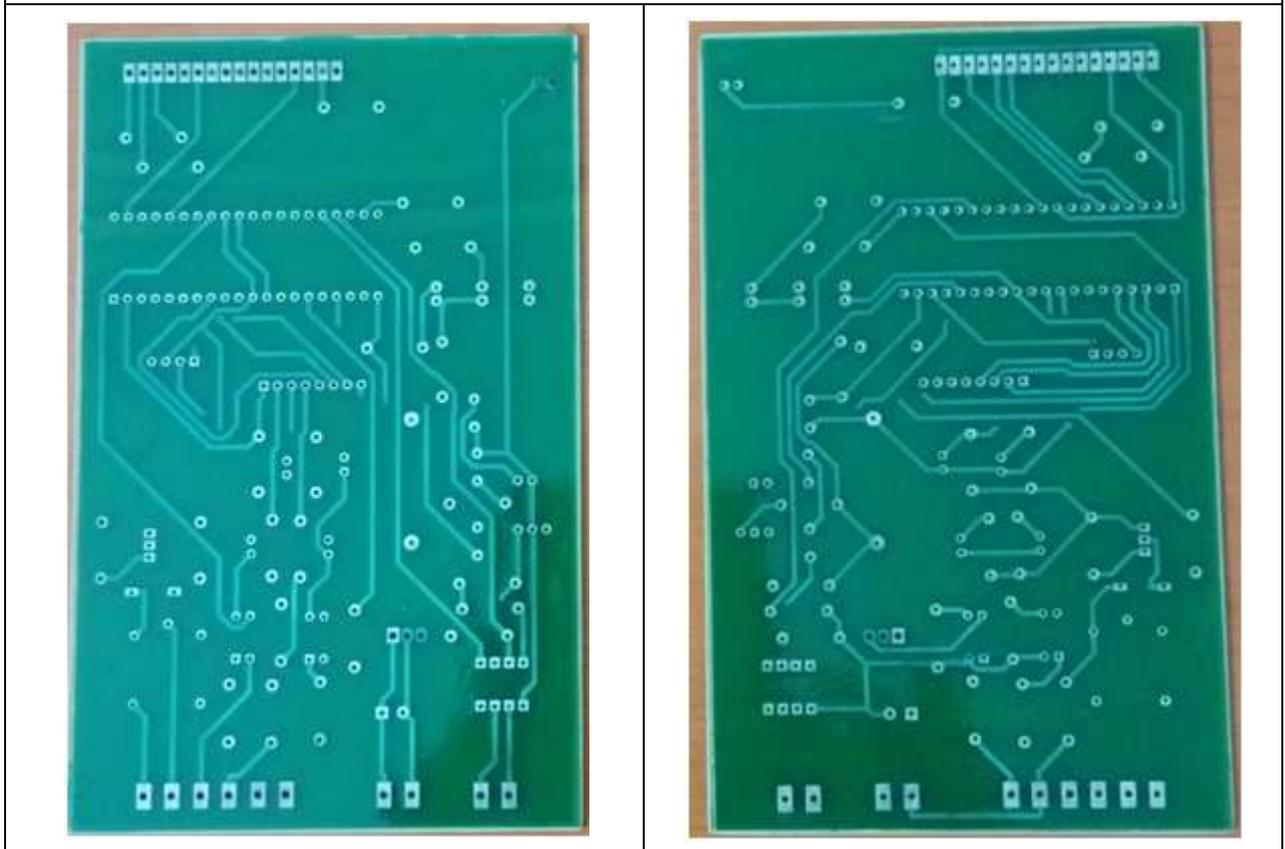
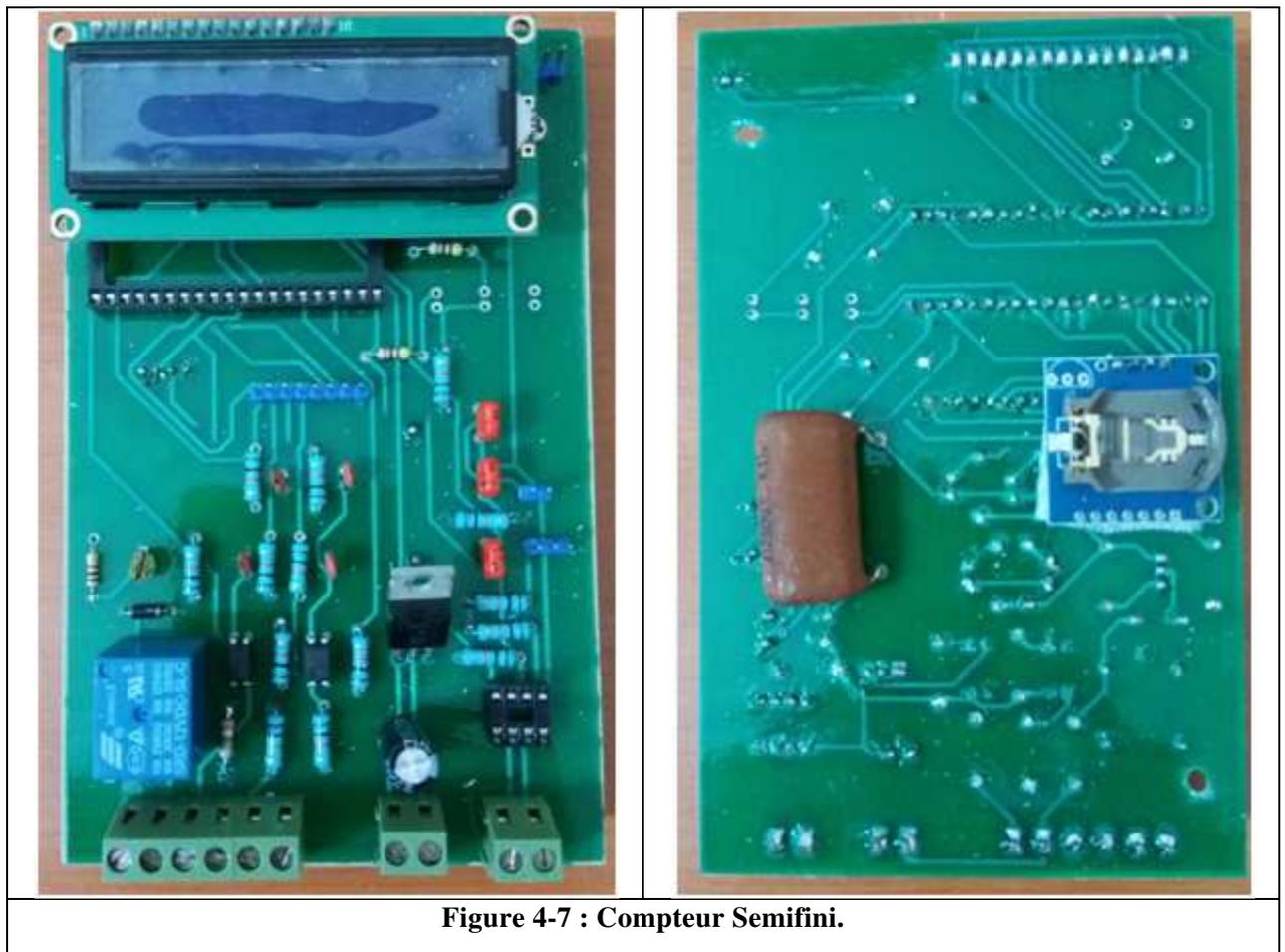


Figure 4-6 : Le circuit imprimé double faces.



4.3 Développement du programme pour le microcontrôleur :

Dans cette partie nous allons présenter le logiciel que nous avons développé avec notre programme pour le microcontrôleur.

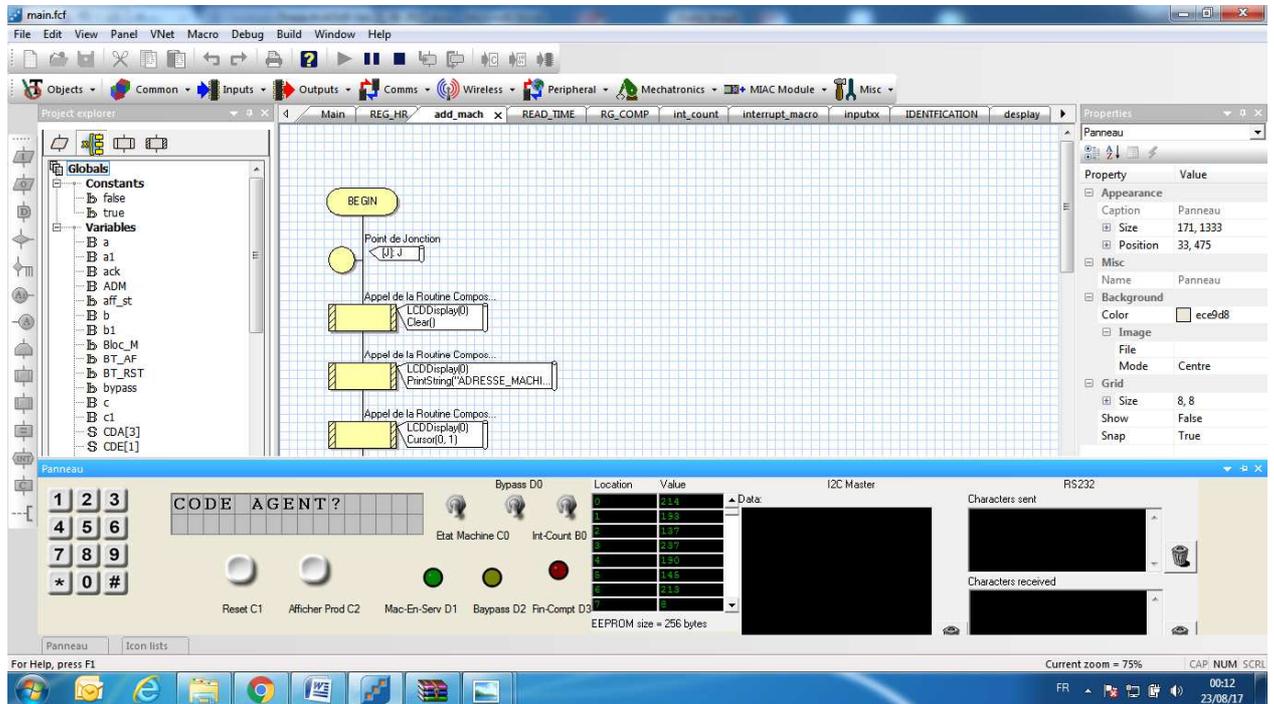


Figure 4-8 : Espace de développement de l'application FLOWCODE.

FLOWCODE est un logiciel dédié au développement de programmes pour une gamme très large de microcontrôleurs, PICFLOWCOD est utilisé précisément pour la famille PIC.

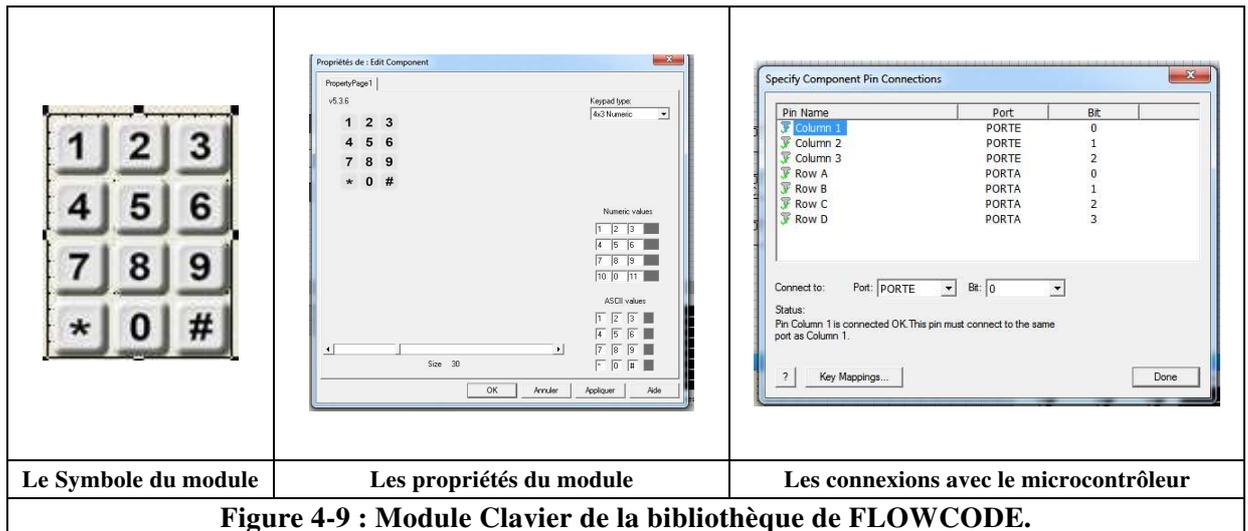
Il utilise le standard international des symboles d'organigrammes ISO5807, Il permet d'enfour du code C ou de l'assembleur sous forme de macro.

Il supporte :

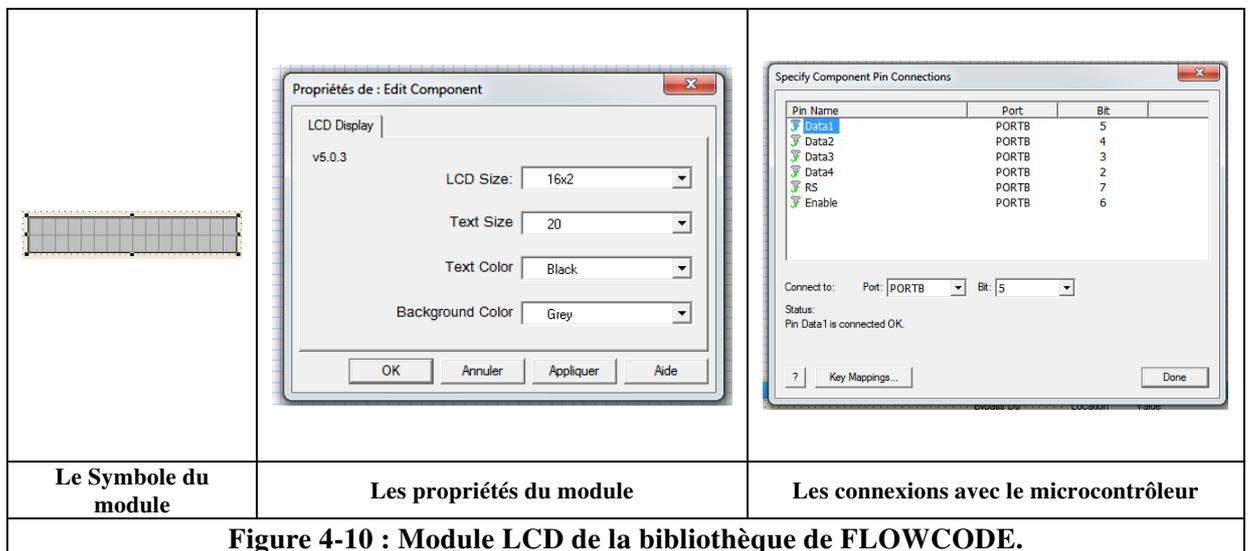
- le paramétrage du Timer 0,
- le paramétrage des convertisseurs A/N 10 bits,
- le paramétrage des interruptions INT, PORTB4-B7 et Timer 0 et d'autres opérations.
- il contient une bibliothèque pour les composants et les modules standards.

Nous avons utilisé les modules suivants :

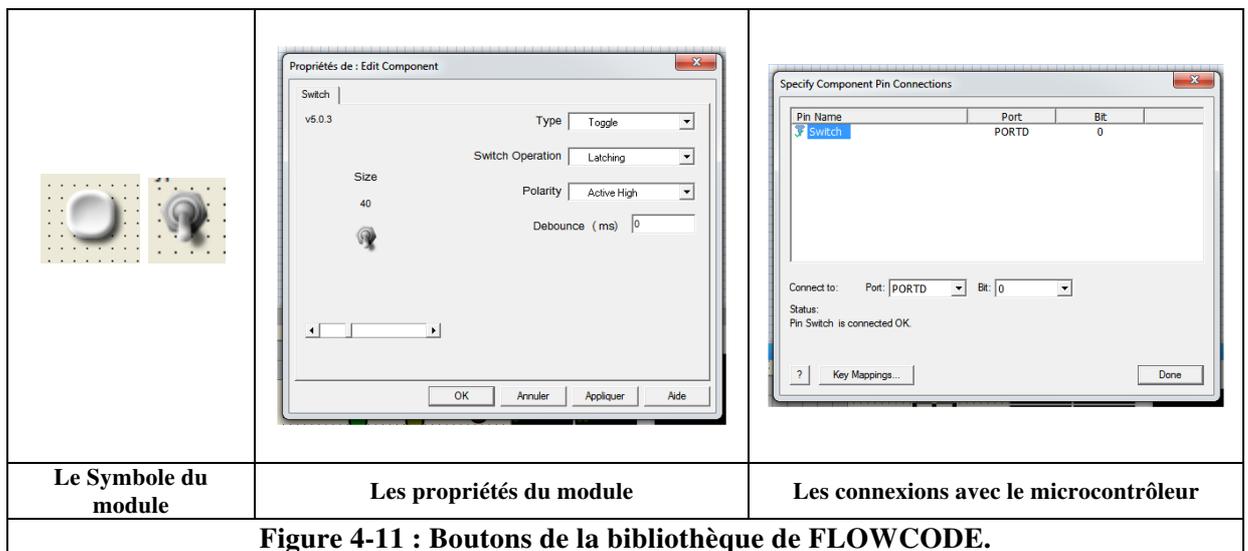
- Le module Clavier :



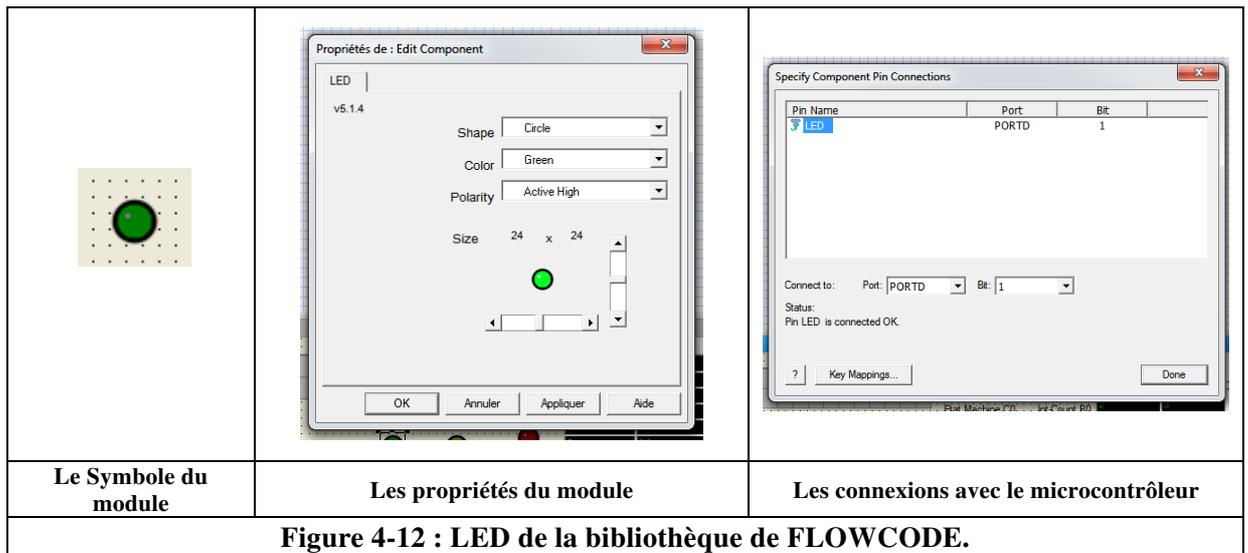
➤ Le module Afficheur LCD :



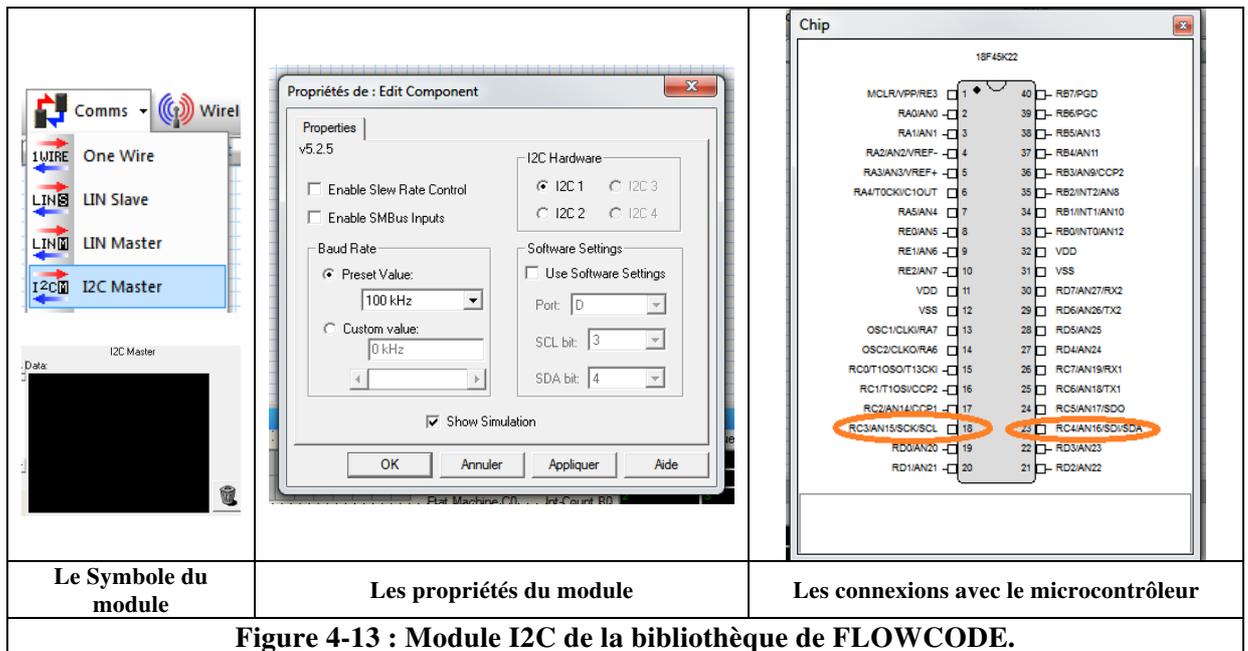
➤ Les boutons :



➤ Les LED :



➤ Les BUS I2C :



Notre programme se compose de :

- Un programme principal.
- Douze (12) sous-programmes à savoir :
 - Initialisation,
 - Actualisation de l'affichage sur le LCD,
 - Récupération des données du Clavier,
 - Identification,
 - Réglage du temps du circuit d'horloge (RTC),

- Réglage de la quantité à produire,
- Changement de l'adresse de la machine,
- lecture de temps du circuit d'horloge (RTC),
- Communication (UART),
- lecture comptage de la EEPROM,
- incrémentation compteur,
- Timer intérieur.

4.4 Simulation de programme et génération fichier HEX :

Le programme développé par le logiciel FLOWCODE peut être simulé dans le même espace de travail, il suffit de cliquer sur Run(F5), s'il n'y a aucune erreur le logiciel exécute le programme comme c'est il est exécuté par le microcontrôleur, comme ça on peut suivre l'action et la réaction de notre programme pour améliorer notre programme et éviter tout éventuel problème avant son téléchargement dans le microcontrôleur.

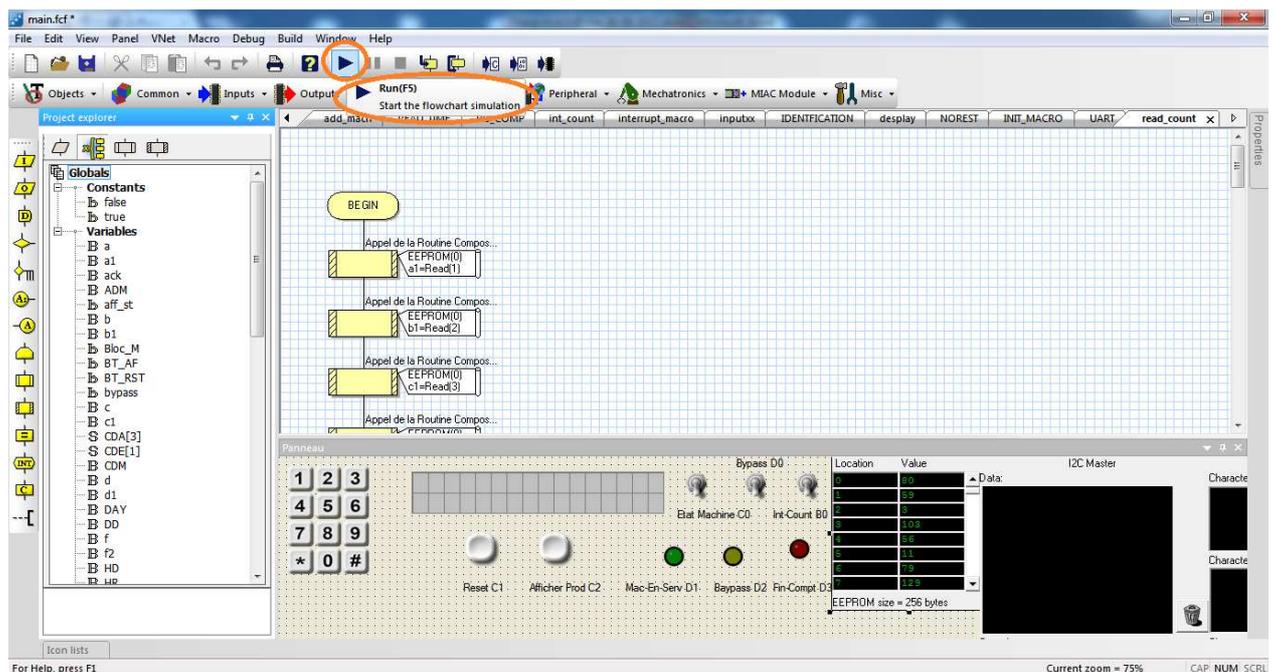


Figure 4-14 : Lancement de la simulation dans FLOWCODE.

Le fichier HEX contient le programme développé en format binaire, c'est le langage machine, et avec ce fichier on peut flasher notre microcontrôleur.

Pour générer ce fichier il suffit de suivre les étapes suivantes :

- Sauvegarder tous les changements, dans l'onglet Build cliquer sur Compile to HEX.

- Un message de fin de compilation apparaitre avec des informations très importante, comme les espaces mémoires dan la RAM et la ROM utilisés par le programme, avec la mention terminer avec succès.

Notre fichier HEX se trouvera dans le même dossier de notre projet FLOWCODE avec une extension « .hex ».

4.5 Flashage et programmation du microcontrôleur :

Le flashage c'est de copie le programme dans la mémoire du microcontrôleur, ce programme déjà compilé et adapté à notre microcontrôleur sous forme d'un fichier HEX.

et pour flasher un microcontrôleur il faut y avoir un pack de flashage compatible avec le microcontrôleur, et comme notre microcontrôleur est le PIC 18F45K22, nous utilisons le PIC KIT2.

Le PICKit2 contient un PIC18F2550 qui possède un bus USB FullSpeed. Le dernier firmware du PICKit2 permet à l'utilisateur de programmer et de déboguer la plupart des PICmicro et des dsPIC 8 et 16 bits de la gamme de Microchip.



Figure 4-15 : Kit de flashage PICKIT2.

Le pack est composé de :

- **Le programmeur :**

Il faut installer les driver du circuit intégré FTD pour pouvoir utiliser le programmeur USB.

- **Le logiciel de programmation (Loader) :**

Permet de télécharger le fichier HEX et les instructions de programmation dans le microcontrôleur.

- **les étapes de flashage sont comme suit :**

- Placer le microcontrôleur à programmer dans le socle de flashage dans notre cas c'est lui de 40 Pins.



Figure 4-16 : PIC18F45K22 placé dans le socle de flashage.

- puis lancer le loader, si tous va bien le loader détectera automatiquement le microcontrôleur.
- Cliquer sur Import Hex dans l'onglet File et parcourir le fichier à télécharger dans le microcontrôleur et si le fichier est conforme le logiciel affiche : « Hex file successfully imported »

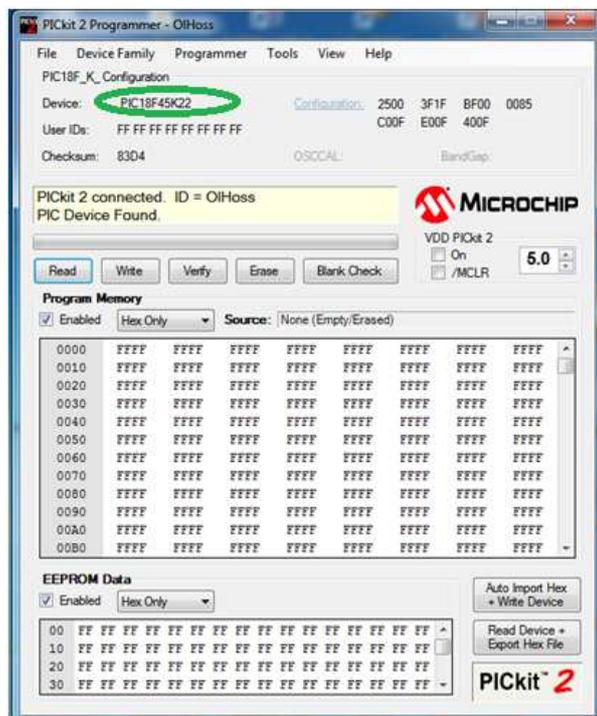


Figure 4-17 : Importation du fichier HEX pour flasher le PIC18F45K22.

- à ce moment on clique sur « write »

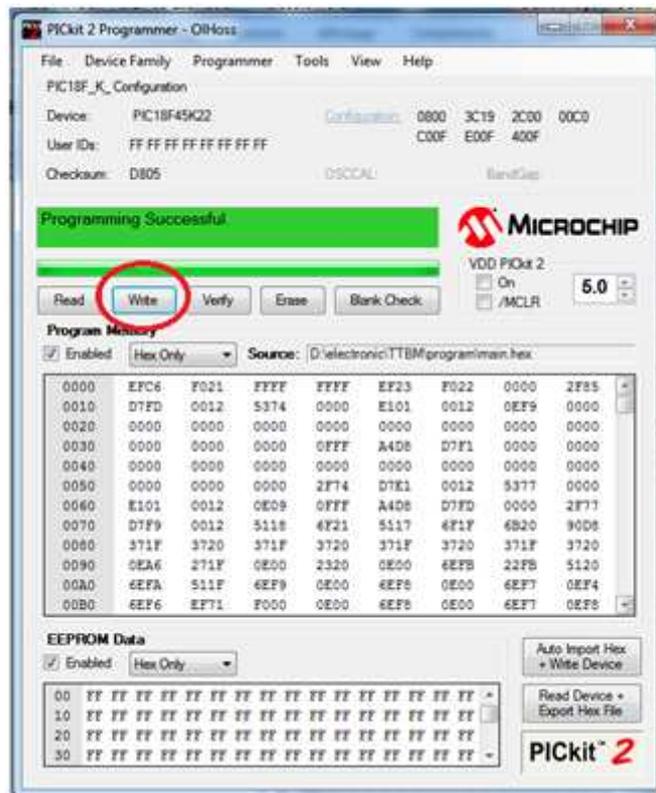


Figure 4-18 : Lancement de flashage du PIC18F45K22.

Si tout va bien le Loader affiche : «Programming Successful» indiquant que notre microcontrôleur est programmé et est prêt à être installé dans le compteur.

4.6 Développement de l'interface IHM :

L'interface homme machine est une application développée afin d'établir la communication entre le PC de gestionnaire (Host) et les compteurs (microcontrôleurs) via un réseau industriel d'une manière automatique après la configuration manuelle des ports de communication comme mentionné dans la section 3.5 du chapitre précédent.

L'ordinateur de gestion (Host) récolte les données des compteurs et les sauvegarde dans une base de données.

Dans cette partie nous allons développer une application composée de trois niveaux :

- **Interface Graphique de consultation et de saisie.**
- **programme de communication est de mise à jour.**
- **base de données.**

Ces trois niveaux sont développés pour que l'application reprenne aux exigences du cahier des charge.

4.6.1 Microsoft Visual Studio (MVS):

Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications web ASP.NET, des services web XML, des applications bureautiques et des applications mobiles. Visual Basic, Visual C++, Visual C# utilisent tous le même environnement de développement intégré (IDE), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages. Par ailleurs, ces langages permettent de mieux tirer parti des fonctionnalités du framework .NET, qui fournit un accès à des technologies clés simplifiant.

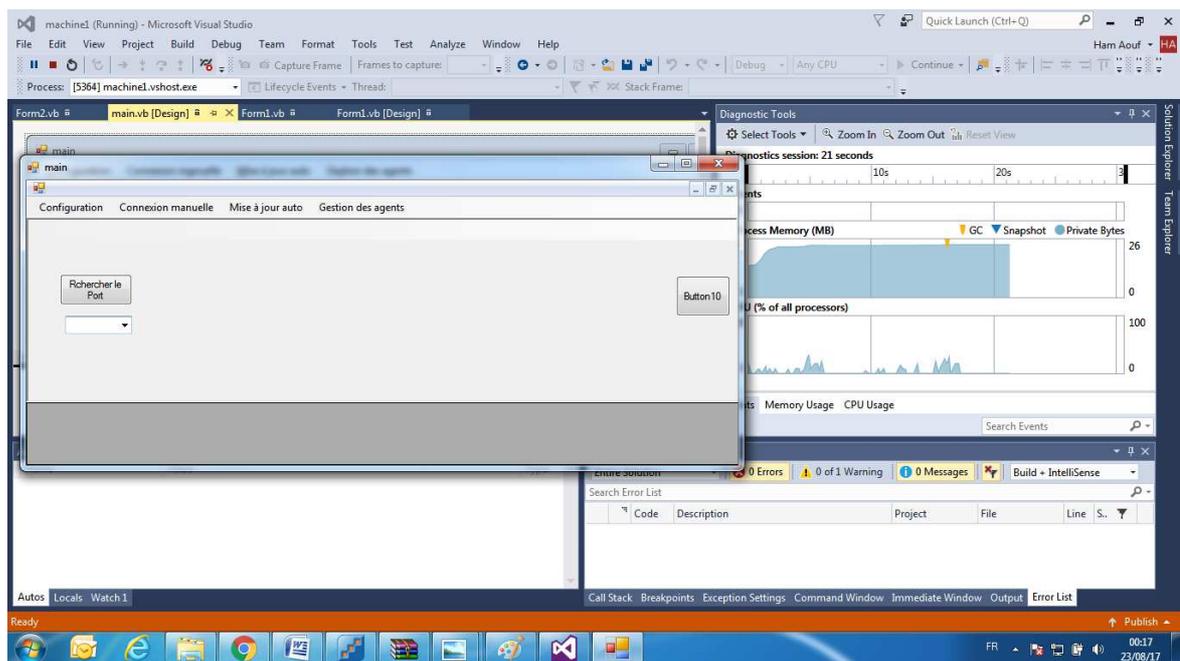


Figure 4-19 : Espace de développement de Microsoft Visual Studio.

4.6.2 Développement des interfaces graphiques :

Nous avons créé une forme principale (main form) sert comme un récipient pour les autre formes de type fils.

- La forme principale : Sur cette forme les menus principaux de l'application.



Figure 4-20 : la forme principale de l'application.

- Formes de type fils : sur ces forme on trouve les boutons les zones de texte et les menus secondaires pour les opérations de la saisie et de consultation.

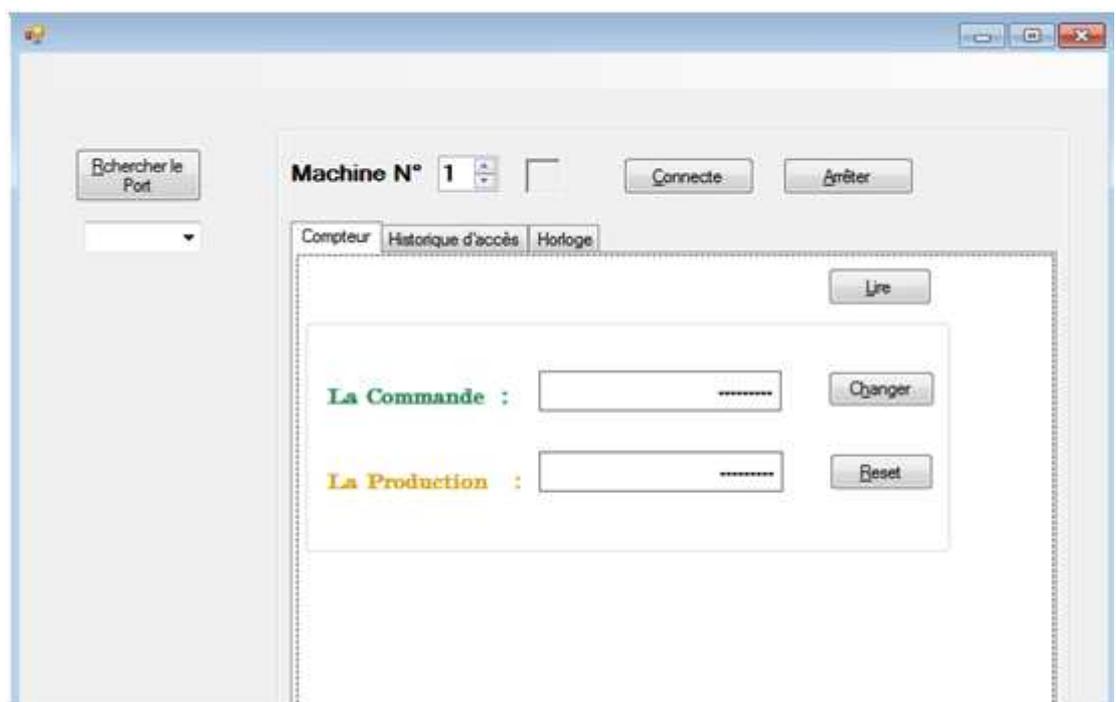


Figure 4-21 : Forme secondaire de l'application

4.6.3 Programme de communication est de mise à jour

La communication est gérée par un sous-programme qui se base sur le composant de communication et de gestion des ports de Microsoft.

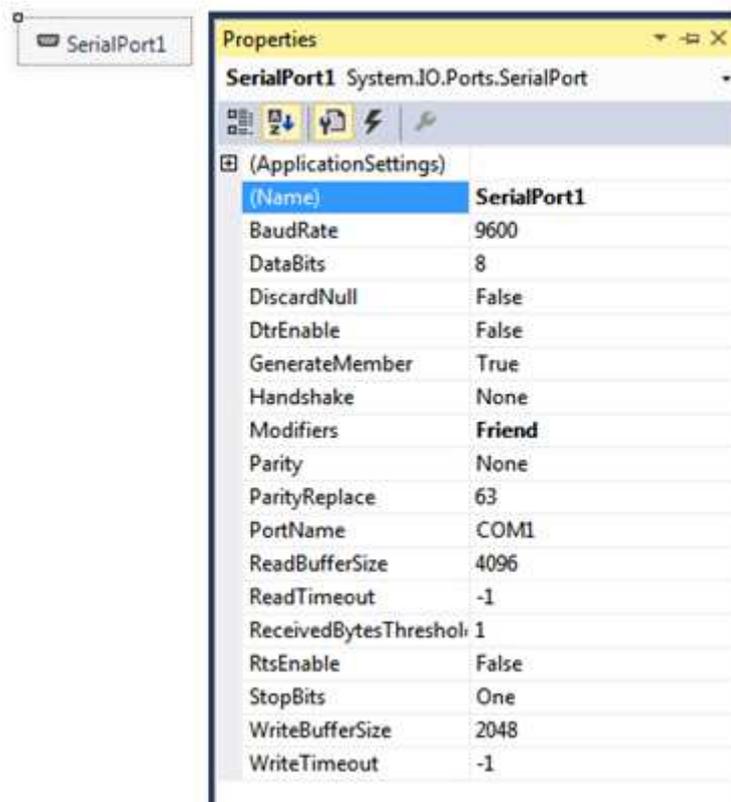


Figure 4-22 : le composant de communication de MVS.

Pour configurer le port du programme il faut un code par exemple :

```

59         With SerialPort1
60
61             .ParityReplace = &H3B           ' replace ";" when parity error occurs
62             .PortName = ComboBox1.Text
63             .BaudRate = 9600
64             .Parity = IO.Ports.Parity.None
65             .DataBits = 8
66             .StopBits = IO.Ports.StopBits.One
67             .Handshake = IO.Ports.Handshake.None
68             .RtsEnable = True
69             .ReceivedBytesThreshold = 1     'threshold: one byte in buffer > event is fired
70             .NewLine = "|"                ' CR must be the last char in frame. This terminates the SerialPort.readLine
71             .ReadTimeout = 10000
72
73         End With
74
75         ' check whether device is available:
76         Try
77             SerialPort1.Open()
78             comOpen = SerialPort1.IsOpen
79         Catch ex As Exception
80             comOpen = False
81             MsgBox("Error Open: " & ex.Message)
82             picOpen.BackColor = Color.Red
83         End Try

```

Et on peut envoyer un message avec le code suivant :

```

256         instr = "r" + Format(NumericUpDown1.Value, "00") + Format(i, "000")
257         If comOpen Then SerialPort1.WriteLine(instr)
258

```

En fin pour recevoir il faut vérifier le registre du port comme suit :

```

109 Private Sub SerialPort1_DataReceived(sender As Object, e As SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived
110     If comOpen Then
111         Try
112             ' read entire string until .Newline
113             readBuffer = SerialPort1.ReadLine
114             If readBuffer <> vbNullString Then
115                 Me.Invoke(New EventHandler(AddressOf Doupdata))
116             End If
117         Catch ex As Exception
118             'If (ex.HResult <> -2147023901) And (ex.HResult <> -2146233883) Then
119                 MsgBox("read " & ex.Message)
120             'End If
121         End Try
122     End If
123 End Sub

```

4.6.4 Base de données :

Nous avons utilisé une base de données Access, avec les tables suivantes :

Tableau 4-1 : structure de la base de données.

Nom de table	Description	structure																														
agent	Liste des agents et leurs codes	<table border="1"> <thead> <tr> <th colspan="2">agents</th> </tr> <tr> <th>Nom du champ</th> <th>Type de données</th> </tr> </thead> <tbody> <tr> <td>CDA</td> <td>Numérique</td> </tr> <tr> <td>Nom</td> <td>Texte</td> </tr> <tr> <td>Prenom</td> <td>Texte</td> </tr> <tr> <td>obs</td> <td>Mémo</td> </tr> </tbody> </table>	agents		Nom du champ	Type de données	CDA	Numérique	Nom	Texte	Prenom	Texte	obs	Mémo																		
agents																																
Nom du champ	Type de données																															
CDA	Numérique																															
Nom	Texte																															
Prenom	Texte																															
obs	Mémo																															
log	Enregistrement des accès	<table border="1"> <thead> <tr> <th colspan="2">log</th> </tr> <tr> <th>Nom du champ</th> <th>Type de données</th> </tr> </thead> <tbody> <tr> <td>N°</td> <td>NuméroAuto</td> </tr> <tr> <td>OPN</td> <td>Numérique</td> </tr> <tr> <td>OPDATE</td> <td>Date/Heure</td> </tr> <tr> <td>LOGN</td> <td>Numérique</td> </tr> <tr> <td>CDM</td> <td>Numérique</td> </tr> <tr> <td>CDE</td> <td>Numérique</td> </tr> <tr> <td>CDA</td> <td>Numérique</td> </tr> <tr> <td>HRD</td> <td>Date/Heure</td> </tr> <tr> <td>HRF</td> <td>Date/Heure</td> </tr> <tr> <td>HRT</td> <td>Numérique</td> </tr> <tr> <td>MNT</td> <td>Numérique</td> </tr> <tr> <td>PRODD</td> <td>Numérique</td> </tr> <tr> <td>PRODF</td> <td>Numérique</td> </tr> </tbody> </table>	log		Nom du champ	Type de données	N°	NuméroAuto	OPN	Numérique	OPDATE	Date/Heure	LOGN	Numérique	CDM	Numérique	CDE	Numérique	CDA	Numérique	HRD	Date/Heure	HRF	Date/Heure	HRT	Numérique	MNT	Numérique	PRODD	Numérique	PRODF	Numérique
log																																
Nom du champ	Type de données																															
N°	NuméroAuto																															
OPN	Numérique																															
OPDATE	Date/Heure																															
LOGN	Numérique																															
CDM	Numérique																															
CDE	Numérique																															
CDA	Numérique																															
HRD	Date/Heure																															
HRF	Date/Heure																															
HRT	Numérique																															
MNT	Numérique																															
PRODD	Numérique																															
PRODF	Numérique																															
machine	Liste des machines et leurs adresses.	<table border="1"> <thead> <tr> <th colspan="2">machine</th> </tr> <tr> <th>Nom du champ</th> <th>Type de données</th> </tr> </thead> <tbody> <tr> <td>CDM</td> <td>Numérique</td> </tr> <tr> <td>NMACH</td> <td>Texte</td> </tr> <tr> <td>OPNBR</td> <td>Numérique</td> </tr> <tr> <td>SCE</td> <td>Oui/Non</td> </tr> <tr> <td>OBS</td> <td>Mémo</td> </tr> <tr> <td>LOG_ACT</td> <td>Numérique</td> </tr> <tr> <td>LAST_UPD</td> <td>Date/Heure</td> </tr> </tbody> </table>	machine		Nom du champ	Type de données	CDM	Numérique	NMACH	Texte	OPNBR	Numérique	SCE	Oui/Non	OBS	Mémo	LOG_ACT	Numérique	LAST_UPD	Date/Heure												
machine																																
Nom du champ	Type de données																															
CDM	Numérique																															
NMACH	Texte																															
OPNBR	Numérique																															
SCE	Oui/Non																															
OBS	Mémo																															
LOG_ACT	Numérique																															
LAST_UPD	Date/Heure																															
port	Le port actif	<table border="1"> <thead> <tr> <th colspan="2">port</th> </tr> <tr> <th>Nom du champ</th> <th>Type de données</th> </tr> </thead> <tbody> <tr> <td>port</td> <td>Texte</td> </tr> </tbody> </table>	port		Nom du champ	Type de données	port	Texte																								
port																																
Nom du champ	Type de données																															
port	Texte																															
user	Liste des utilisateurs de l'application et leurs mots de passe	<table border="1"> <thead> <tr> <th colspan="2">user</th> </tr> <tr> <th>Nom du champ</th> <th>Type de données</th> </tr> </thead> <tbody> <tr> <td>userID</td> <td>Texte</td> </tr> <tr> <td>Username</td> <td>Texte</td> </tr> <tr> <td>Password</td> <td>Texte</td> </tr> </tbody> </table>	user		Nom du champ	Type de données	userID	Texte	Username	Texte	Password	Texte																				
user																																
Nom du champ	Type de données																															
userID	Texte																															
Username	Texte																															
Password	Texte																															

La communication entre la base de données et l'application se fait à l'intermédiaire d'un composant de Microsoft Visual Studio appelé : « Data Base Data Set ».

4.7 Essais et mise en service :

Après la fabrication et le montage du compteur sur la machine modèle de fabrication de bandes en tissu selon le schéma d'installation présenté dans la **Figure 4-23**.

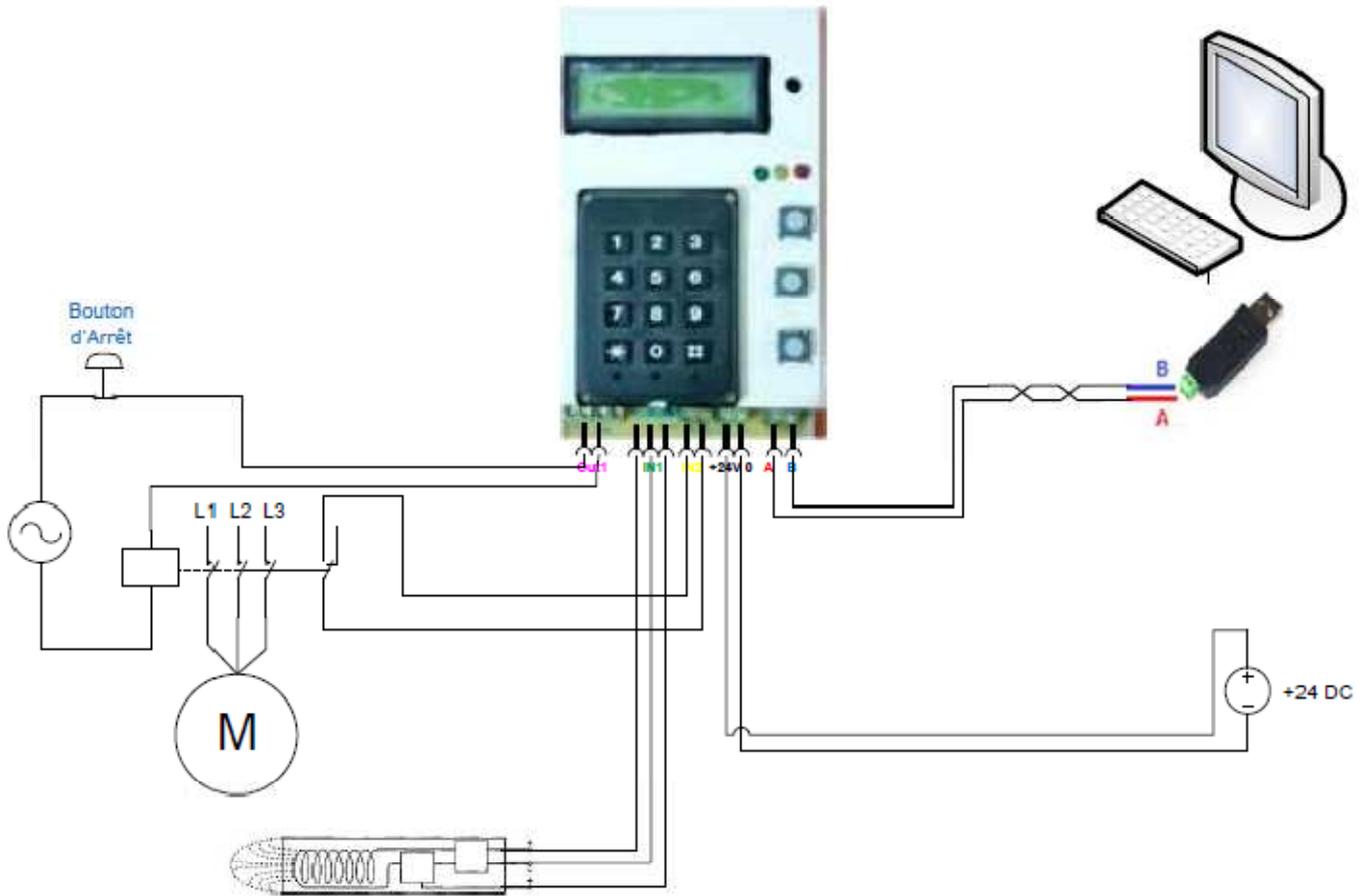


Figure 4-23 : Schéma de montage de l'ensemble du système.

Nous avons constaté un dysfonctionnement lors du démarrage d'un moteur électrique puissant environ 3 KW.

Après analyse nous avons constaté qu'il faut protéger le régulateur L7805CV contre le champ électromagnétique par l'installation d'un condensateur céramique de $0.33\mu\text{F}$ entre les bornes de la sortie du régulateur.

La distance entre le PC de gestion et le compteur a été plus de 100 mètres.

Les résultats des essais ont été très satisfaisants, le projet a répondu parfaitement aux exigences du cahier des charges.

4.8 Conclusion :

Le choix des outils et les méthodes de travail est un facteur principal dans la partie des travaux de pratique et de réalisation, pour cela nous avons choisi des solutions qui ont fait preuve dans le milieu des professionnels, à savoir :

Au début la conception et la réalisation du circuit imprimé nous avons utilisé EasyEDA, une solution simple et efficace avec une base de données de composants très large,

En suite nous avons utilisé pour la programmation du microcontrôleur le FLOWCODE, un autre logiciel très puissant avec une grande base de données des modules et des préprogrammes utiles au développeur pour gagner du temps.

En fin le développement de l'application coté PC (Host) on utilisant Microsoft Visual Studio.

Le seul challenge c'est de maîtriser l'anglais.

CHAPITRE 05

Conclusion Générale

Chapitre 5 : Conclusion Générale

Ce travail démontre le passage d'une idée abstraite à une solution concrète et pratique pour une problématique industrielle.

Nous avons commencé par la présentation de généralité sur la gestion de la production, le suivi des commandes de production et les contraintes rencontrées d'où la nécessité d'une application pour suivre ces commandes, mais comme, dans les moyennes et les grandes entreprises les données à mettre à jour chaque minute pour avoir une situation exacte à tout moment sont très volumineuses, cette mise à jour devienne un grand challenge, et ça, c' été notre problématique.

La fabrication d'un système basé sur microcontrôleurs embarqué sur chaque machine et lié avec l'ordinateur de gestion via le réseau industriel RS485 géré par une application.

Ensuite dans la partie de l'étude nous avons présenté les aspects technique de l'acquisition est de traitement des données, et la conception de divers circuits du compteur comme les entrée et les sorties numérique, l'algorithme de fonctionnement du microcontrôleur et celui de l'application au niveau de l'ordinateur (Host).

Enfin la concrétisation de notre solution proposée à la base d'un microcontrôleur PIC 18F45K22, utilisons l'application FLOWCODE pour la programmation et le PICKIT2 pour le flashage de ce dernier.

Et du côté de l'ordinateur nous avons utilisé le logiciel Microsoft Visual Studio (MVS) pour le développement d'une application de communication et de gestion des données.

La partie de la mise en service et les essais a met en preuve notre travail, d'où nous avons constaté une légère anomalie au niveau de circuit d'alimentation, qu'été remédiée après analyse et après cette opération les résultats des essais ont été concluantes.

Nous tenons à mentionner que le coût de développement et de la réalisation est très confortable si on sait que ce coût ne représente que 50% d'un seul contrôleur programmable et en plus vierge sans aucun programme à l'étranger sans les frais d'importation.

Vue les résultats concluantes de ce travail et le coût abordable de notre travail, nous invitons les industries et les laboratoires de développement Algérien d'encourager les développeurs locaux et de confier à eux une partie de leurs projets.

Bibliographie :

1. **A. COURTOIS, C. MARTIN-BONNEFOUS, M. PILLET.** *Gestion de production*. France : Éditions d'Organisation, 2003. ISBN : 2-7081-2986-4.
2. **Casanova, Gérard.** *Gestion des flux*. France : Université de Lorraine, AUNEGE, 2011. pp. http://www.cetice.u-psud.fr/aunege/gestion_flux/co/Module_gestion_flux.html.
3. **Tavernier, Christian.** *Microcontrôleurs : principes et applications*. Paris : s.n., 1991.
4. **LILEN, Henri.** *Microprocesseurs du CISC au RISC*. s.l. : Dunod, 1998.
5. **MONTAGNY, Sylvain.** *Microprocesseurs & Microcontrôleurs*. chablais, : s.n. pp. www.master-electronique.com.
6. **Kammoun, Yosra Rkhissi.** pp. <http://www.technologuepro.com/cours-genie-electrique/cours-32-microcontrolleurs/>.
7. **B&B Electronics Mfg. Co. Inc.** *RS-422 and RS-485*. Ottawa : s.n., October 1997.
8. **Merlin Gerin - Schneider Electric.** *Guide réseau MODBUS*. 2000.
9. **Microsoft.** MSDN (Microsoft Developer Network). <https://msdn.microsoft.com/fr-fr/aa537127.aspx>. [En ligne] 2015.
10. **Romarc, Hinault.** <https://www.developpez.com/actu/87887/Visual-Studio-2015-permet-de-developper-tout-type-d-application/>. <https://www.developpez.com>. [En ligne]