

**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE UNIVERSITE MOULOUD MAMMERI, TIZI-OUZOU**



**FACULTE DE GENIE ELECTRIQUE ET DE L'INFORMATIQUE
DEPARTEMENT D'ELECTRONIQUE**

Mémoire de fin d'études

En vue de l'obtention

Du Diplôme de Master II en Electronique

Option : Réseaux et télécommunication

Thème :

**Conception et réalisation d'un système à
microcontrôleur de mesure de CO avec une
transmission ZigBee**

Proposé et dirigé par :

Mr. LAGHROUCHE Mourad.

Présenté par :

M^r BOUAZIZ Yacine

M^{elle} DEHAM Malha

Année universitaire 2010/2011

REMERCIEMENTS

En premier lieu nous remercions notre **DIEU** le tous puissant de nous avoir donné la foi, la santé et nous a permit de bien mener ce travail.

Avant d'entreprendre la rédaction de notre mémoire, nous souhaitons vivement remercier et exprimer notre gratitude à :

Notre promoteur **M. LAGHROUCH Mourad**, à qui nous sommes très reconnaissants pour son aide et ses conseils.

Nous tenons, également, à exprimer notre reconnaissance et notre sincère gratitude à tous les enseignants de bonne foi qui nous ont accompagnés tout au long de notre formation.

Un grand merci à notre ami AKLI et à tous ceux qui ont contribués de prêt ou de loin à la réussite de ce mémoire en particulier à A. Salim et B. Sihem.

YACINE, MALHA.



Dédicaces

Je dédie ce modeste travail à :

Mes très chers parents qui m'ont soutenus tout au long de mes études et qui ont contribué à ma réussite, que dieu les garde et leur donne une longue vie.

Mes sœurs Ferroudja et Sihem que j'aime beaucoup et à qui je souhaite une bonne réussite dans leurs études et dans leurs vie.

Toute ma famille et mes amis.

Tous ceux qui m'aime et que j'aime que je n'ai pas cité, mais que je n'ai pas oublié.

YACINE





Dédicaces

Je dédie ce modeste travail à :

Mes très chers parents ;

Mes sœurs et frères ;

Ma chère belle sœur Fatiha ;

Nana et son mari ;

*Mes précieuses amies : Lwisa et sa sœur Nacima, Myma,
Wiwiz, Haya, Racha, Dihou et à Lynda ;*

Mes amis : Remand, Djamal, Mohamed et Karim ;

Mon binôme et surtout pour sa patience ;

*Toute personne qui a contribué de près ou de loin à la
réalisation de ce travail ;*

Toute personne que j'estime.

LILA.

Liste des figures

Figure 1 : Schéma de principe d'un capteur.....	2
Figure 2 : Le capteur dans la chaîne de mesure.....	3
Figure 3 : Schéma idéal de la réponse d'un capteur	4
Figure 4 : Schéma synoptique de l'architecture VON NEUMANN	6
Figure 5 : Schéma synoptique de l'architecture HARVARD	6
Figure 6 : Architecture protocolaire du ZigBee.....	8
Figure 7 : Bandes de fréquence pour ZigBee.....	8
Figure 8 : Structure d'un paquet de la couche physique IEEE 802.15.4.....	9
Figure 9 : Format général de la trame MAC	10
Figure 10 : Topologie Etoile	12
Figure 11 : Topologie Point à point	13
Figure 12 : Montage d'un module XBee sur une carte avec une interface RS232	15
Figure 13 : Interfaçage du module XBee à un processeur.....	16
Figure 14 : Structure interne du module XBee	16
Figure 15 : Principe d'une liaison RS 232	17
Figure 16 : Structure d' E /S d'un module I2C.....	18
Figure 17 : Le schéma synoptique de fonctionnement	19
Figure 18 : Structure du capteur HS 134	20
Figure 19 : Principe de détection du capteur HS134.....	21
Figure 20 : Principe de mesure.	22
Figure 21 : sensibilité du capteur HS 134.	22
Figure 22 : Liaison du capteur HS 134 au PIC 16F876A.	23
Figure 23 : Broches du capteur DS1621.....	24
Figure 24 : Liaison du capteur DS 1621 au PIC 16F876A.....	25

Figure 25 : Configuration du pic 16F876A.	26
Figure 26 : Broches du pic 16F876A.	27
Figure 27 : L'interruption.	28
Figure 28 : Configuration minimale du PIC 16F876A.	29
Figure 29: Broches du pic 18F452	30
Figure 30: câblage de l'afficheur LCD.....	31
Figure 31 : EEPROM 24C64A	32
Figure 32 : Câblage de l'EEPROM.....	33
Figure 33 : Câblage du capteur DS1302	34
Figure 34 : Brochage des boutons poussoirs avec le PIC 16F876A	34
Figure 35 : Schéma électrique du bloc alimentation	35
Figure 36 : Le module XBee-INT-RS232	36
Figure 37 : Brochage du connecteur DB-9.....	37
Figure 38 : Platine d'interface USB pour modules XBee 990.002	38
Figure 39 : Les différentes platines XBee	38
Figure 40 : Création et enregistrement du programme	55
Figure 41 : Configuration du PIC par logiciel (étape 1).....	56
Figure 42 : Configuration du PIC par logiciel (étape 2).....	57
Figure 43 : Configuration du PIC par logiciel (étape 3).....	57
Figure 44 : fenêtre pour écrire le programme principale.....	58
Figure 45 : Compilation du programme	58
Figure 46 : Programmation du PIC sous WinPic800_3.55G (étape 1).....	59
Figure 47 : Programmation du PIC sous WinPic800_3.55G (étape 2)	60
Figure 48 : Le circuit imprimé de la carte d'émission coté cuivre	62
Figure 49 : le circuit imprimé de la carte d'émission coté composant	63
Figure 50 : Implémentation des composants (carte d'émission)	63

Figure 51 : Le circuit imprimé de la carte de réception coté cuivre.....	64
Figure 52 : le circuit imprimé de la carte de réception coté composant	64
Figure 53 : Implémentation des composants (carte de réception)	65
Figure 54 : Carte d'émission en 3D	65
Figure 55 : Carte de reception en 3D	65
Figure 56 : Carte d'émission après réalisation	66
Figure 57 : Carte de réception après réalisation	66

Liste des tableaux

Tableau 1 : Comparaison des protocoles sans fil.....	13
Tableau 2 : Les différentes broches du module XBee	15
Tableau 3 : Conditions de travail du capteur HS 134	23
Tableau 4 : Broches de l’afficheur LCD	31
Tableau 5 : Broches du support DB-9 sur la platine de l’XBee PRO.....	37

SOMMAIRE

Sommaire

Introduction générale

Chapitre I : Généralités

Introduction.....	1
1) Généralités sur la température	1
2) Généralités sur la mesure du monoxyde de carbone (CO)	1
3) Généralités sur les capteurs	1
3.1) Définition	1
3.2) Classification des capteurs.....	2
a. Capteur actif	2
b. Capteur passif.....	2
3.3) Le capteur dans la chaîne de mesure	3
3.4) Les caractéristiques d'un capteur.....	4
4) Généralités sur les microcontrôleurs	5
4.1) Définition d'un microcontrôleur	5
5) Généralités sur le réseau ZigBee.....	6
5.1) Définition du protocole ZigBee	7
5.2) Architecture protocolaire du ZigBee.....	7
a. La couche physique IEEE 802.15.4	8
b. La sous couche MAC du réseau IEEE 802.15.4	9
c. La couche ZigBee	11
d. La couche application (APL).....	12
5.3) Topologie du réseau Zigbee.....	12
a. Topologie étoile	12
b. Topologie point à point	13
5.4) Comparaison entre les protocoles sans fil	13
6) Le Module XBee	14
6.1) Caractéristiques des modules XBee	14
a. Communication RF	14
b. Gestion réseaux	14

c. Consommation	14
6.2) Description du module XBee	14
6.3) Communication avec le module XBee	16
7) Interface série RS232	16
8) Le bus I2C	17
Conclusion	18

Chapitre II: Conception matérielle

Introduction.....	19
1) Schéma synoptique	19
2) Description des différents éléments de la première carte	20
2.1) Bloc capture.....	20
2.1.1) Le capteur HS 134.....	20
a. Structure du capteur HS 134.....	20
b. Principe de détection	20
c. Caractéristiques du capteur HS 134	22
d. Conditions de travail du capteur HS 134.....	23
e. Liaison du capteur HS 134 au PIC 16F876A.....	23
2.1.2) Le capteur DS 1621.....	24
a. Définition du capteur DS1621	24
b. Transmission de données avec le DS 1621	24
c. Les broches du capteur DS1621.....	24
d. Liaison du capteur DS 1621 au PIC 16F876A.....	24
3) Le pic 16F876A.....	25
3.1) Présentation du PIC 16F876A	25
3.2) Schéma bloc du PIC 16F876 et description de chaque broche.....	26
a. Les ports d'E/S	27
b. L'interruption.....	27
c. Configuration minimal du Pic 16F876A.....	28
4) Description des différents éléments de la deuxième carte	29
4.1) Description du PIC 18F452	29
4.2) Présentation du PIC 18F452	29
4.3) Les broches du PIC 18F452.....	30

5) Bloc affichage	30
5.1) Afficheur LCD.....	30
5.2) Les broches de l’afficheur LCD	31
5.3) Fonctionnement de l’afficheur LCD.....	31
5.4) EEPROM (Electrically-Erasable Programmable Read-Only Memory)	32
5.5) Les broches de l’EEPROM	32
5.6) Brochage de l’EEPROM au PIC 18F452.....	33
5.7) Le dateur DS1302.....	33
5.8) Brochage du DS 1302.....	33
5.9) Brochage des boutons poussoirs.....	34
6) Les composants présents sur les deux cartes.....	35
6.1) Bloc alimentation	35
6.2) Le module XBee.....	36
6.3) Description des modules XBee utilisés	36
a. Platine d'interface RS 232 pour modules XBee 990.006	36
b. Platine d'interface USB pour modules XBee 990.002	37

Chapitre III: Programmation et réalisation pratique

1) Définition d’un organigramme	39
2) Les organigrammes.....	39
2.1) Déclaration des variables.....	39
2.2) Organigramme de la transmission.....	40
2.3) Organigramme de la réception.....	41
2.4) Organigramme du réglage de l’heure et de la date	43
2.5) Organigramme de l’écriture dans l’EEPROM externe	46
2.6) Organigramme de la lecture à partir de l’EEPROM externe.....	50
3) Schéma d’affichage des étapes du réglage de l’heure et de la date.....	54
4) Ecriture et transfert du programme dans le PIC	55
4.1) Utilisation du logiciel CCS	55
4.2) WinPic800_3.55G	59
5) Schémas électrique de la carte d’émission.....	60
6) Schémas électrique de la carte de réception.....	61
7) Réalisation pratique	61

7.1) Les circuits imprimés des deux cartes	61
7.1.1) L'insolation	61
7.1.2) La gravure.....	62
7.1.3) Le perçage	62
7.2) Implantation des composants	62
7.2.1) Implantation des composants de la carte d'émission	62
7.2.2) Implantation des composants de la carte de réception	64
8) Nomenclature des composants	67
8.1) Liste des composants utilisés dans la carte d'émission	67
8.2) Liste des composants utilisés dans la carte de réception	67
Conclusion générale	

Introduction générale

Introduction général:

De nos jours, l'utilisation de la logique programmable se généralise à toutes les conceptions de cartes électroniques. Cela est dû à sa maniabilité qui se traduit dans sa rapidité d'exécution, sa facilité de programmation, sa bonne précision et la possibilité d'apporter des améliorations, en plus l'avènement du microcontrôleur a changé l'aspect des cartes électroniques en réduisant la superficie de celles-ci les rendant beaucoup plus pratiques à l'utilisation, tout en réduisant le coup de fabrication.

On retrouve l'électronique dans la vie quotidienne de l'homme moderne, et dans tous les corps de métiers (médecine, industrie, météorologie...), elle a pour objectif de simplifier l'existence des êtres humains, et de contribuer au développement des différents secteurs d'activité.

En tant qu'électroniciens, nous souhaitons apporter notre contribution à la lutte contre le réchauffement climatique, et cela en limitant les quantités de gaz à effet de serre émises dans l'atmosphère, en ayant des systèmes de mesure performants.

Notre travail consiste en l'étude et la réalisation de deux cartes électroniques, la première carte est gérée par le PIC 16F876A, elle prend en charge la mesure du monoxyde de carbone et de la température, les valeurs mesurées seront transmises via le module XBee à la deuxième carte, qui est gérée par le PIC18F452 et chargée de leur affichage. Cette carte sera installée dans une voiture pour surveiller les émissions en CO du moteur

Ce travail est réalisé au sein du département d'électronique, de l'université MOULOUD MAMMERI de Tizi-Ouzou, il est organisé en trois chapitres.

Le premier chapitre traite les notions générales sur la température, le monoxyde de carbone CO, les capteurs et le protocole de transmission Zig Bee.

Le deuxième chapitre est consacré à la Conception matérielle.

Dans le troisième chapitre nous allons décrire le programme de gestion du système et sa réalisation pratique.

Chapitre 1 :

Généralités

0 Introduction :

Avant d'entamer notre travail, il est primordial d'avoir une approche globale et une idée générale sur les propriétés et les caractéristiques des éléments utilisés dans la conception du système: la température, le monoxyde de carbone, les capteurs, ainsi que le protocole de transmission Zig Bee.

1) Généralités sur la température: [22]

La température est la mesure de l'énergie moyenne des particules de la matière. La température est reliée au mouvement des particules. Plus la vitesse des particules est grande, plus la température de la substance est élevée.

Elle se présente à nous comme une valeur susceptible d'augmenter ou de diminuer, caractérisant l'état d'un corps au point de vue des échanges possibles de chaleur entre ce corps et les corps extérieurs. Cette transmission ne cesse (*si les deux corps sont seuls en présence*) que quand ils sont arrivés, au point de vue calorifique, au même état : on dit qu'ils sont alors à la même température. Un corps plus chaud est dit à température supérieure. On a ainsi défini l'égalité et l'inégalité de température.

Mais pour apprécier cette égalité ou cette inégalité avec quelques précisions, nos sens sont insuffisants, donc on a recours à des instruments spéciaux tels que les thermomètres.

2) Généralités sur la mesure du monoxyde de carbone (CO): [17]

Le monoxyde de carbone est un gaz inodore, incolore et inflammable, il se forme lors de la combustion incomplète de matières organiques (gaz, charbon, fioul ou bois, carburants,...). La source principale est le trafic automobile. Des taux importants de CO peuvent être rencontrés quand un moteur tourne au ralenti dans un espace clos ou en cas d'embouteillages dans des espaces couverts, ainsi qu'en cas de mauvais fonctionnement d'un appareil de chauffage domestique.

Le CO participe aux mécanismes de formation de l'ozone troposphérique. Dans l'atmosphère, il se transforme en dioxyde de carbone CO_2 et contribue à l'effet de serre.

3) Généralités sur les capteurs: [16]

Dans de nombreux domaines (industrie, recherche scientifique, services, loisirs ...), on a besoin de contrôler de nombreux paramètres physiques (température, force, position, vitesse, luminosité, ...).

Le capteur est l'élément indispensable à la mesure de ces grandeurs physiques.

3.1) Définition : [8]

Un capteur est un dispositif électronique qui sert à transformer une grandeur physique, chimique ou biologique (mesurande) en un signal électrique exploitable.

On peut caractériser les capteurs selon deux critères:

- ü En fonction de la grandeur mesurée : on parle alors de capteur de position, de température, de vitesse, de force, de pression, etc.
- ü en fonction du caractère de l'information délivrée : on parle alors de capteurs logiques appelés aussi capteurs tout ou rien (TOR), de capteurs analogiques ou numériques.

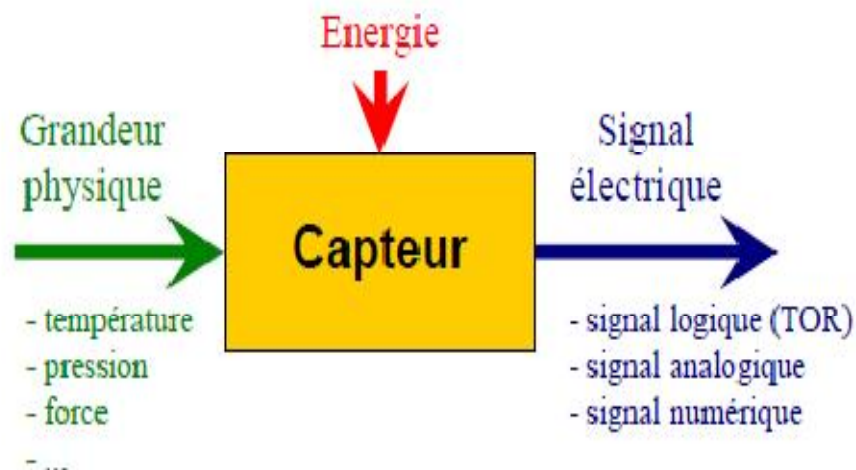


Figure 1: Schéma de principe d'un capteur

3.2) Classification des capteurs: [8]

a. Capteur actif:

Comme son nom l'indique, ce capteur fonctionne en générateur. Son principe repose sur un effet physique qui assure la conversion en énergie électrique de l'énergie propre au mesurande: l'énergie thermique, mécanique ou de rayonnement.

b. Capteur passif:

Ce capteur se comporte en sortie comme un dipôle passif qui peut être résistif, capacitif ou inductif (exemple : thermistance, photorésistance, potentiomètre, jauge d'extensomètre).

Les capteurs qu'on a vu précédemment nécessitent tous un montage d'amplification et de traitement du signal. Avec l'avancé technologique et l'apparition de LSI (large scale integration) et de VLSI (very large scale intégration), les capteurs peuvent être intégrés aux circuits de traitement du signal sur la même puce pour réaliser l'ensemble des fonctions, on les appelle « capteurs intelligents ».

Les capteurs intelligents regroupent bon nombre d'avantages tel que :

- ü Amélioration du rapport signal/bruit par adaptation d'impédance et amplification.
- ü Prétraitement du signal, tel que : compensation en température et en variation d'alimentation, remise à zéro automatique, filtrage de signaux parasites, corrections de non-linéarité.
- ü Traitement du signal, tel que le codage et la modulation des signaux de sortie.
- ü Réduction de la consommation.
- ü Atténuation des perturbations extérieures (grandeurs d'influence).

Exemple :

La famille des LM135/ LM235/ LM335 qui sont plus intéressante en terme de performance/prix, avec une sortie analogique nécessitant tout de même un convertisseur analogique/numérique, de ce fait on s'intéressera à la famille DS75/ DS1620/ DS1621 qui facilite l'interfaçage avec le PIC, on utilisant la norme I2C.

3.3) Le capteur dans la chaîne de mesure :

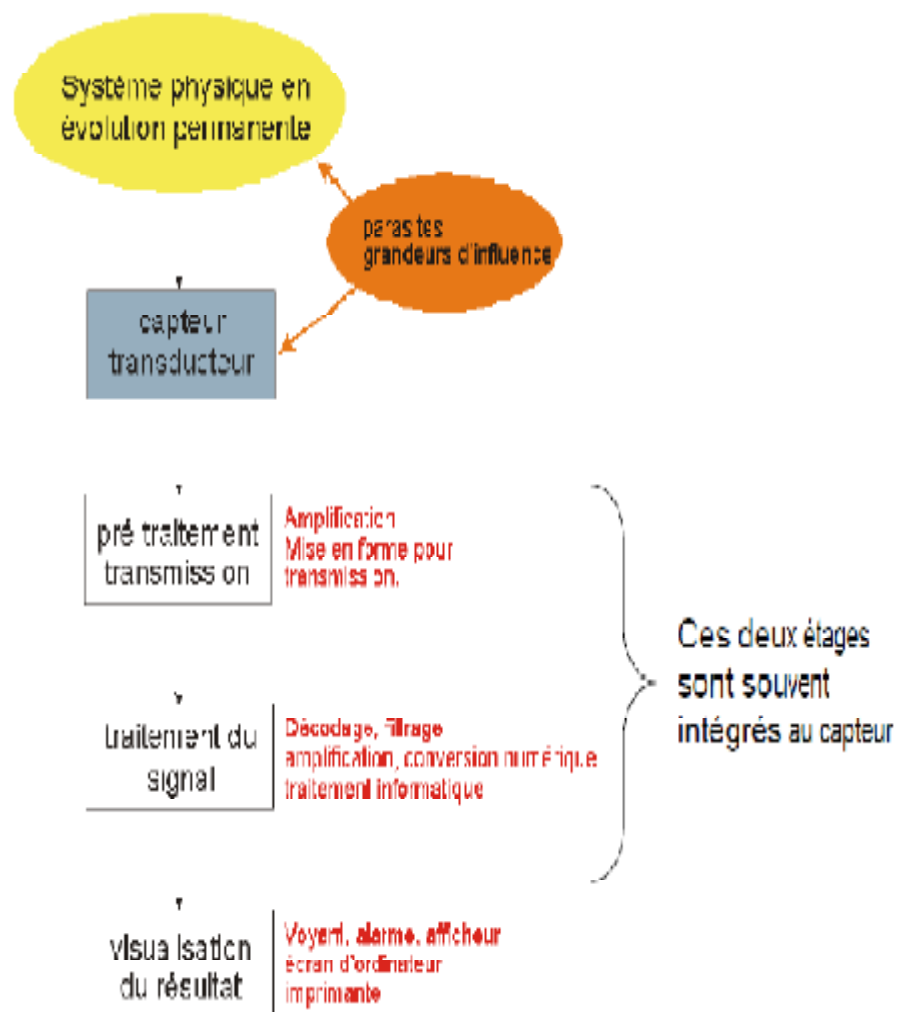


Figure 2: Le capteur dans la chaîne de mesure

3.4) Les caractéristiques d'un capteur: [8]

Les qualités d'un capteur par rapport à la nature de la grandeur à mesurer et aux conditions de mesure sont appréciées en déterminant leur résolution, leur fidélité, leur sensibilité et leur sélectivité.

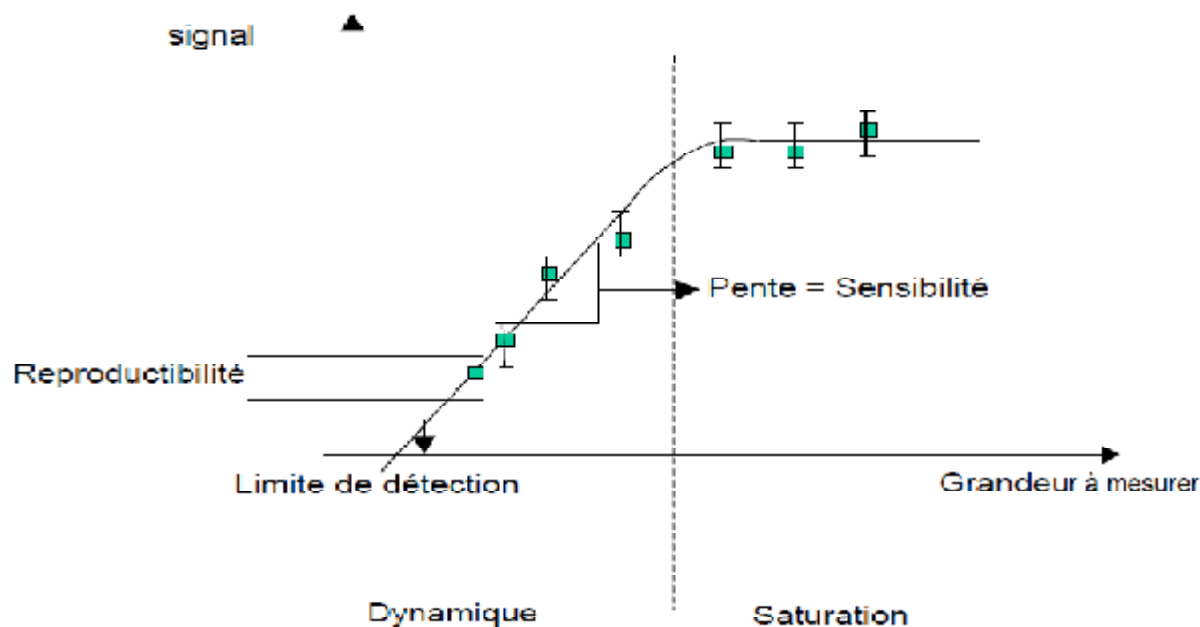


Figure 3 : Schéma idéal de la réponse d'un capteur.

Reproductibilité: Ce paramètre est probablement le plus important, tant pour les capteurs physiques que chimiques. C'est l'aptitude d'un capteur à donner, dans des conditions définies, des réponses très voisines lors de l'application répétée d'un même signal d'entrée.

Résolution: Plus petit incrément de la valeur mesurée qui puisse être significativement mesuré par le capteur.

Sensibilité: Ce paramètre caractérise l'aptitude du capteur à détecter la plus petite variation de la grandeur à mesurer. Il est calculé par le quotient de l'accroissement de la réponse d'un instrument de mesure par l'accroissement correspondant du signal d'entrée.

Sélectivité: C'est la capacité du capteur à ne mesurer qu'une seule grandeur dans le milieu où il est utilisé ou en d'autres termes, d'être le plus insensible aux grandeurs d'influence, grandeurs qui ne font pas l'objet de la mesure, mais influent sur la sortie du capteur.

Limite de détection : C'est la plus petite valeur de la grandeur à mesurer pouvant être détectée, avec une incertitude acceptable, et qui sort significativement du bruit de fond.

4) Généralités sur les microcontrôleurs : [2][18][19][20]**4.1) Définition d'un microcontrôleur :**

Un microcontrôleur est un composant électronique Autonome doté :

- d'un microprocesseur.
- de la mémoire RAM.
- de la mémoire permanente.
- des interfaces d'E/S « parallèles, série (RS232, I2C, SPI...) ».
- des interfaces d'E/S analogique.
- Des Timer pour gérer le temps.
- D'autres modules plus au moins sophistiqués selon la taille des microcontrôleurs.

Il est généralement moins puissant qu'un microprocesseur en terme de rapidité ou de taille mémoire, il se contente le plus souvent d'un bus 8 ou 16 bits. Ceci en fait un composant très bon marché, parfaitement adapté pour piloter les applications embarquées dans de nombreux domaines d'application. On peut même affirmer qu'aujourd'hui il y'a un microcontrôleur (plus ou moins grand) dans chaque équipement électronique que ce soit en :

- Informatique (souris, modem).
- Vidéo (Appareil photos numérique, caméra numérique).
- Contrôle des processus industriels (régulation, pilotage).
- Appareil de mesure (affichage, calcul statistique, mémorisation.)
- Automobile (ABS, injection, GPS, airbag).
- Multimédia (téléviseur, carte audio, carte vidéo, MP3, magnétoscope).
- Téléphones (fax, portable, modem).
- Electroménager (lave-vaisselle, lave-linge, four micro-onde).

Un microcontrôleur peut être programmé une fois pour toutes afin qu'il effectue une ou des tâches précises au sein d'un appareil électronique. Mais les microcontrôleurs récents peuvent être reprogrammés et ceci grâce à leur mémoire permanente de type FLASH (d'où le terme flasher quelque chose).

Plusieurs Constructeurs se partagent le marché des microcontrôleurs, citons INTEL, MOTOROLA, AMTEL, ZILOG, PHILIPS et enfin MICROCHIP avec leurs PICs très populaires qui nous intéressent dans notre travail.

Les microcontrôleurs, quelque soit leurs constructeurs, ont des architectures très similaires et sont constitués de modules fondamentaux assurant les mêmes fonctions : UAL, Ports d'E/S, interfaces de communications série, Interfaces d'E/S analogiques, Timers et horloge temps réels.

Il existe deux architectures des microcontrôleurs :

- Ø L'architecture **VON NEUMANN** employée par la plupart des microcontrôleurs actuels (**INTEL80XX**, **Motorola HC05**, **HC08** et **HC11**, ou **ZILOG Z80**) est basée sur un bus de données unique. Celui-ci véhicule les **instructions** et les **données**.

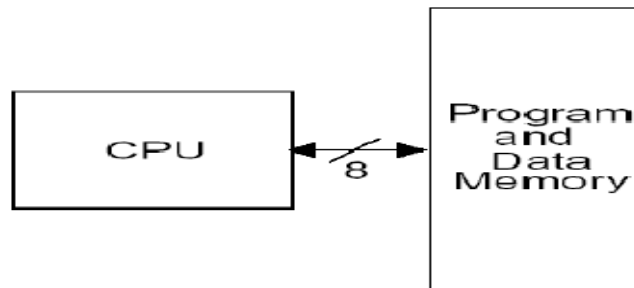


Figure 4 : Schéma synoptique de l'architecture **VON NEUMANN**

- Ø L'architecture **HARVARD** utilisée par les microcontrôleurs **PICS** est basée sur deux bus de données. Un bus est utilisé pour les **données** et un autre pour les **instructions**.

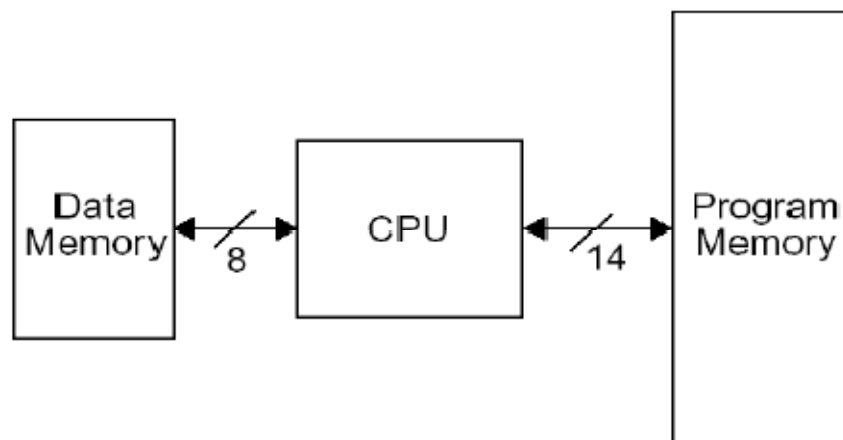


Figure 5 : Schéma synoptique de l'architecture **HARVARD**

5) Généralités sur le réseau ZigBee : [10][13][15]

Avec la convergence de l'informatique, de l'électronique et des télécommunications, il est ainsi possible de mettre en place des réseaux de capteurs sans fil dans un contexte domotique ou de contrôle industriel.

Il est clair qu'il existe un marché que ZigBee permet de combler : surveillance de locaux pour la détection de départ de feu, surveillance de bâtiments contre les intrusions, assistance aux personnes...

Ce standard est promu par l'alliance ZigBee qui regroupe un ensemble d'industriels (40 entreprises) travaillant sur l'élaboration de spécifications afin de pouvoir développer des applications sans fil bon marché, faible consommation et sécurisées.

5.1) Définition du protocole ZigBee : [14]

Le nom ZigBee signifie « Zig Zag like a bee » qui veut dire en français « zigzaguer comme une abeille », c'est un protocole de haut niveau qui permet à de petites radios de communiquer sur de faibles distances.

La technologie Zigbee est un LP-WPAN (Low Power – Wireless Personal Area Network), c'est un réseau sans fil qui présente les caractéristiques suivantes :

- bas débit.
- courte portée.
- équipé d'une batterie et non d'une alimentation classique, à la différence des équipements Wifi.
- Très basse consommation énergétique.

Le protocole Zigbee utilise les ondes hertziennes pour transporter des messages entre deux ou plusieurs entités réseaux visant à pallier les faiblesses des systèmes existant (tel que le Wifi : qui est un protocole lourd et consommateur, le Bluetooth : qui est un protocole de faible portée, limité aux petits réseaux).

Son principal objectif est de mettre au point une technologie qui permet une installation facile avec un coût réduit, une pile protocolaire légère, déclinable sous plusieurs versions en fonction des besoins et de la topologie souhaitée, ce qui la rend particulièrement adaptée pour être directement intégrée dans de petits appareils électroniques (appareils électroménagers, jouets).

La norme ZigBee s'appuie sur la norme IEEE (Institute of Electrical and Electronics Engineers) 802.15.4 pour les niveaux physique et MAC (Medium Access).

5.2) Architecture protocolaire du ZigBee : [11]

L'architecture protocolaire de la norme IEEE 802.15.4 est donnée par la figure suivante :

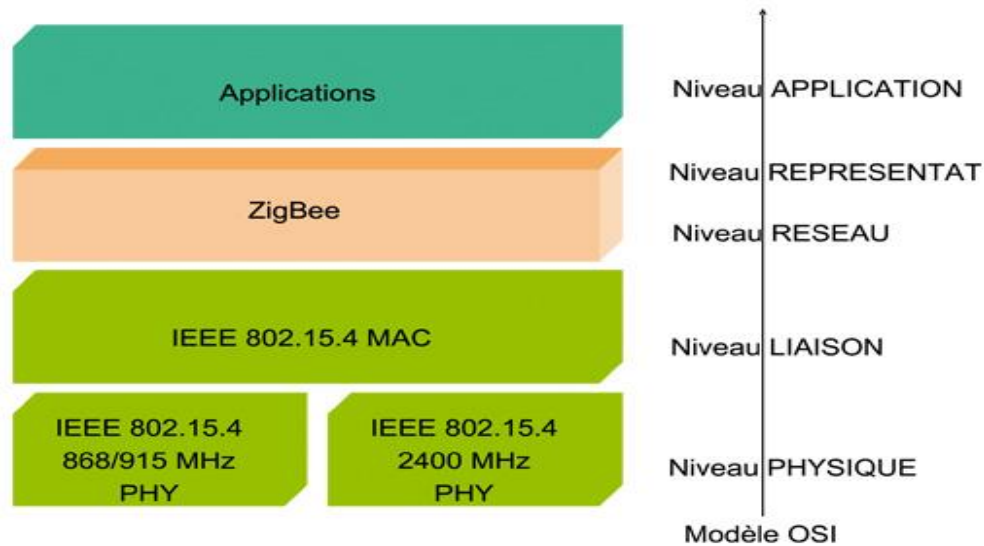


Figure 6 : Architecture protocolaire du ZigBee

a. La couche physique IEEE 802.15.4 : [23]

La norme IEEE 802.15.4 supporte les 3 bandes ISM (Industrial, Scientific, Medical) de 868 MHz, 915 MHz et 2,4 GHz. Les caractéristiques de chaque bande de fréquence est résumé dans la figure suivante :

	Bande	Usage	Débit	Nombre de canaux
2.4 GHz	ISM	Mondial	250kb/s	16
915 MHz	ISM	Amérique	40kb/s	10
868 MHz	ISM	Europe	20kb/s	1

Figure 7: Bandes de fréquence pour ZigBee

Le niveau physique gère les fonctionnalités suivantes :

- ü Activation/désactivation de l'interface radio.
- ü Détection de l'énergie dans le canal (Choix du canal radio).
- ü Qualité du lien radio.
- ü Évaluation du canal pour la mise en œuvre du protocole d'accès CSMA.
- ü Émission/réception des paquets dans le canal radio.

- La structure du paquet PHY de données émis/reçu est illustré par la figure 8 :

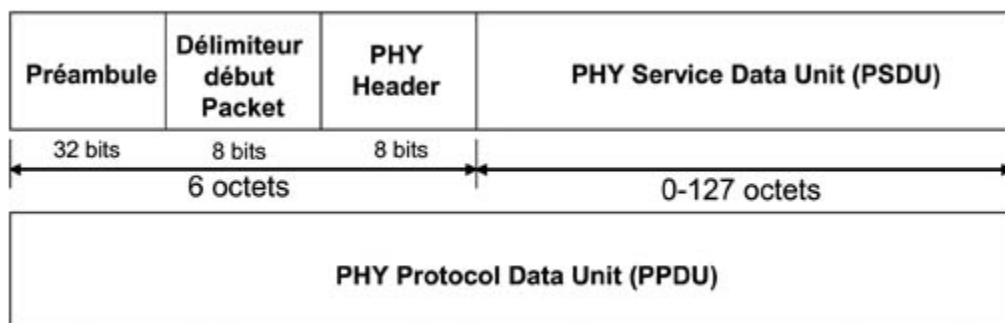


Figure 8: Structure d'un paquet de la couche physique IEEE 802.15.4

- **Le paquet contient un entête de synchronisation qui comprend 6 octets :**
 - Un préambule sur 5 octet qui est conçu pour l'acquisition de symboles et le chronométrage du jeton (chip timing), et dans certains cas, peut être utilisé pour ajuster grossièrement la fréquence et permet au récepteur de parfaire sa synchronisation.
 - Un fanion de délimiteur de début de paquet sur 1 octet qui rompt l'alternance des bits transmis lors du préambule, indiquant ainsi l'imminence de la transmission de données.
 - Après l'entête de synchronisation, vient l'entête PHY dont le rôle est de spécifier la longueur du paquet avec un codage de 7 bit et le bit restant est réservé.
 - Ensuite l'unité de données de service PHY (PSDU : PHY Service Data Unit) supporte une longueur de 0 à 127 octets, bien qu'à cause de la couche MAC au dessus, des paquets de longueur zéro ne pourront pas se produire en pratique.

b. La sous-couche MAC du réseau IEEE 802.15.4 : [23]

La sous-couche MAC possède les caractéristiques suivantes :

- Mise en œuvre des trames avec un adressage IEEE sur 64 bits ou un adressage court sur 16 bits, soit 65536 équipements adressables au plus dans ce dernier cas.
- Structure de trame simple.
- Mode de transmission fiable.
- Gestion de l'association/désassociations d'équipements.
- Mode de transfert de type half duplex.

- **Deux mécanismes d'accès au réseau sont possibles:**

1. Réseau en mode non beacon (balise): On utilise des communications avec une politique d'accès au médium (MAC) de type CSMA/CA. Les trames de données reçues correctement sont acquittées.
2. Réseau en mode beacon: Un coordinateur du réseau émet périodiquement des trames beacon (de 15 ms jusqu'à 252 s) délimitant ainsi une supertrame. Le format de la

supertrame est fixé par le coordinateur, elle se compose de 16 intervalles de temps, et cela permet ainsi de synchroniser tous les équipements (nœuds) du réseau. Un équipement pourra émettre ses données dans un intervalle de temps donné pendant la durée de celui-ci. L'accès aux premiers intervalles de temps est non garanti donc soumis à contention (Contention Access Period), alors que sur les derniers, il est garanti (Contention Free Period), ce qui permet d'assurer un certain déterminisme et une certaine qualité de service.

- **Structure générale de la trame MAC :[23]**

Le format général d'une trame MAC est montré à la figure 9 :

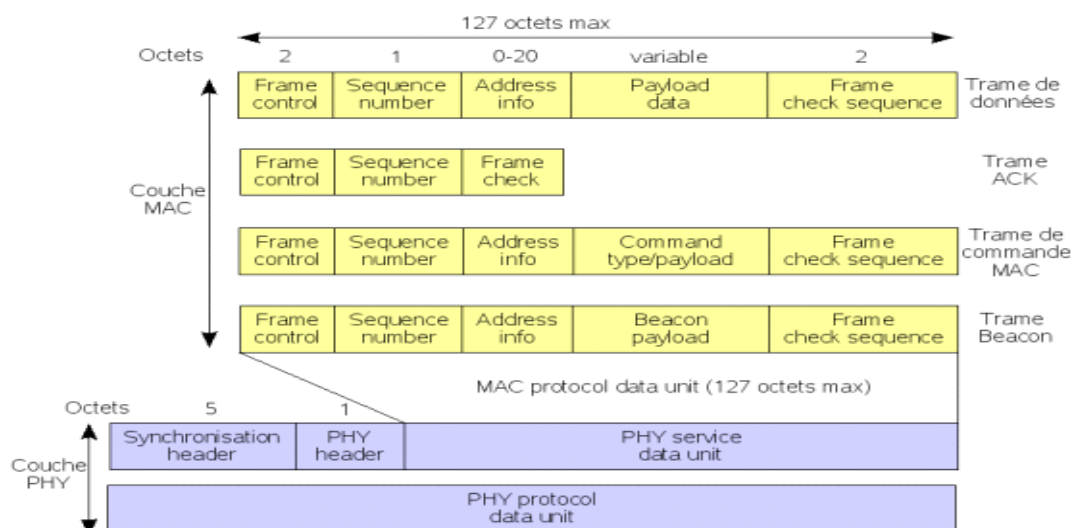


Figure 9 : Format général de la trame MAC.

Les trames MAC de la norme IEEE 802.15.4 peuvent être de quatre types différents qui sont les suivants:

1. Trame de données (Data Frame) : transfert de données.
2. Trame d'acquiescement (Acknowledgment Frame) : confirmation de données bien reçues.
3. Trame beacon (Beacon Frame): émise par le coordinateur de réseau en mode beacon.
4. Trame de commande MAC (MAC Command Frame) : pour le contrôle des nœuds.

La trame générale MAC est composée d'un entête MAC (MHR : MAC Header), d'une unité de données de service MAC (MSDU: MAC Service Data Unit) et pied de page MAC (MFR : MAC Footer).

- ü L'entête MAC (MAC Header) est composé d'un champ de contrôle de trame qui indique le type de trame MAC transmise, spécifie le format du champ adresse et contrôle l'acquiescement. Le champ d'adresse est variable de 0 à 20 octets en fonction

du type de trame, et le champ du numéro de séquence assure l'ordre à la réception et permet l'acquittement de la trame MAC.

- ü MSDU (MAC service data unit) (payload) est de longueur variable. Cependant, la trame MAC complète ne doit pas dépasser 127 octets de long, et les données contenues dans le champ de données "payload" dépendent du type de trame.
- ü MFR (MAC Footer) contient Le champ de la séquence de contrôle de trame FCS (Frame Check Sequence) qui aide à vérifier l'intégrité de la trame MAC. Dans les trames MAC IEEE 802.15.4 le FCS est un control de redondance cyclique CRC (Cyclic Redundancy Check) sur 16 bits de l'Union des Télécommunications Internationales- secteur standardisation des télécommunications ITUT (International Telecommunication Union). Seules les trames de données et de balisage contiennent réellement des données envoyées par les couches supérieures, les trames d'acquittement et de commande MAC proviennent de la couche MAC et sont utilisées pour les communications MAC point à point.

Enfin, la norme IEEE 802.15.4 définit 3 types d'équipements (nœuds du réseau) :

- Ø Le coordinateur du réseau.
- Ø L'équipement à fonctionnalités complètes FFD (Full Function Device).
- Ø L'équipement à fonctionnalités réduites RFD (Reduced Function Device).

L'équipement FFD peut être soit un coordinateur, soit un routeur, soit un équipement terminal (capteur).

L'équipement RFD est un équipement simplifié comme un équipement terminal (End Device) muni de capteurs.

Pour communiquer au sein d'un même réseau, au moins un équipement FFD et des équipements RFD utilisent de concert le même canal radio parmi ceux définis dans la norme.

Un équipement FFD peut dialoguer avec des équipements RFD ou FFD, mais un équipement RFD ne peut dialoguer qu'avec un équipement FFD.

c. La couche ZigBee :[23]

Le niveau ZigBee précise les algorithmes de routage pouvant être mis en œuvre dans le réseau. Le routage mis en œuvre peut être soit direct, soit indirect.

Dans le cas du routage direct, lorsqu'un dispositif voulant transmettre des données connaît l'adresse réseau du destinataire. Cette adresse est donc transmise dans la trame pour atteindre et agir sur le dispositif prévu.

Dans le cas contraire, routage indirect l'équipement source ne connaît pas l'adresse de l'équipement destinataire. Dans ce cas, un équipement routeur ou coordinateur fera la mise en relation avec l'équipement destinataire en utilisant sa table de routage.

d. La couche application (APL) :

C'est la couche la plus élevée dans le réseau et accueille les objets d'application. Elle se compose des éléments APS (Application support Sublayer), AF (Application Framework), et enfin des objets d'application des équipements, que ceux ci soient définis par ZigBee ou par le constructeur des équipements. Jusqu'à 240 objets d'application distincts.

5.3) Topologie du réseau Zigbee : [23]

Les topologies mises en œuvre avec ZigBee dépendent de la complexité de l'application utilisée. Il existe plusieurs types de topologie (Etoile, Point à Point, Maillé), les plus utilisées sont :

a. Topologie étoile:

Dans cette configuration, la pièce maîtresse, celle par qui tous les messages transitent, est le coordinateur (PAN coordinator), il joue en quelque sorte le rôle de passerelle. Si un dispositif "A" souhaite dialoguer avec un dispositif "B", la communication doit impérativement passer par le coordinateur. Dans cette topologie, les dispositifs sont très simples et n'embarquent qu'une couche simplifiée du protocole Zigbee. Le coordinateur est en général fixe et alimenté pour garantir le fonctionnement du réseau.

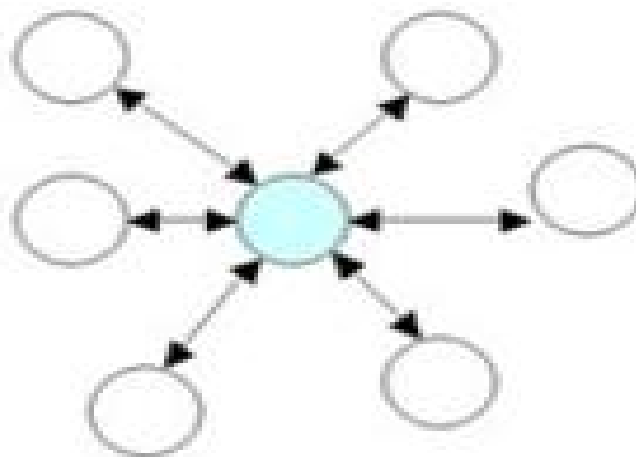


Figure 10: Topologie Etoile

b. Topologie Point à Point:

Les dispositifs sont capables de dialoguer directement entre eux s'ils sont à proximité, ou bien d'utiliser le coordinateur pour contacter un dispositif à plus longue distance. Dans cette topologie, les dispositifs sont plus compliqués et intègrent entièrement le protocole ZigBee. Une mécanique de tables de liaison permet également à chaque dispositif de jouer le rôle de coordinateur, permettant à deux dispositifs éloignés de dialoguer par l'intermédiaire d'un élément à portée.

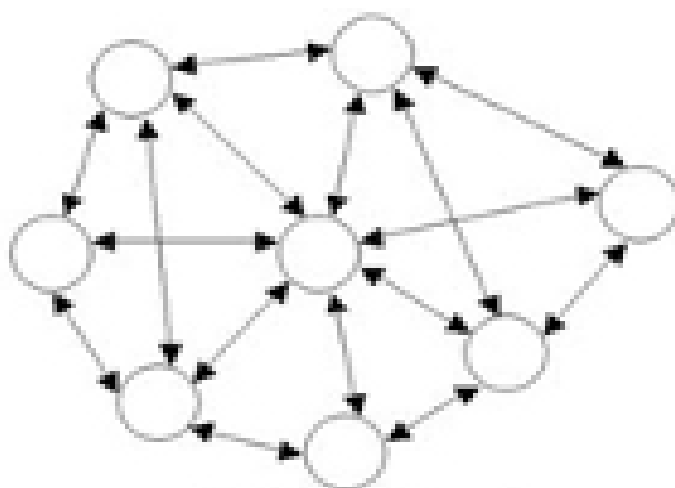


Figure 11 : Topologie Point à point

5.4) Comparaison entre les protocoles sans fil : [11]

Protocole	ZigBee	Bluetooth	Wifi
IEEE	802.15.4	802.15.1	802.11a/b/g
Besoin mémoire	4-32Kb	250Kb et +	1Mb et +
Autonomie de la batterie	Années	Jours	Heurs
Nombre de nœuds	65000 et +	7	32
Vitesse de transfert	250Kb/s	1Mb/s	11-54 Mb/s
Portée	100 m	10 m	100 m

Tableau 1: Comparaison des protocoles sans fil

Toutes les caractéristiques du protocole ZigBee sont bien adaptées aux systèmes embarqués.

En effet, le protocole ZigBee se distingue des autres protocoles par ses faibles besoins en mémoire, ce qui est favorable pour son implémentation. De plus, il présente une durée de vie très importante qui est de l'ordre de plusieurs années, ainsi qu'un très large nombre de nœuds à supporter dans son réseau. Enfin, ce protocole convient parfaitement aux applications nécessitant une faible vitesse de transfert de l'ordre de 250 Kb/s.

6) Le Module XBee : [23]

Vue que le ZigBee ne peut être utilisé directement, la société « Maxstream » propose des modules intégrés ZigBee prêt à l'emploi intitulés « les modules XBee ». Ces modules permettent une conversion de données série TTL vers le réseau ZigBee. Les modules XBee sont des transceivers radio au standard ZigBee™ / IEEE 802.15.4, qui sont spécialement conçus pour la réalisation de systèmes de communication au sein de réseaux de capteurs sans fil. De petites dimensions, ces modules se distinguent par leur grande simplicité d'utilisation et leur coût très compétitif qui les prédestinent également à de très nombreuses autres applications. Les modules XBee opèrent dans la bande ISM des 2.4 GHz, et ils reposent sur une technologie de communication de type DSSS (Direct Sequence Spread Spectrum). Pouvant être utilisés sans aucun paramétrage, les modules disposent également d'une multitude de modes de fonctionnement très facilement accessibles via des commandes AT.

6.1) Caractéristiques des modules XBee : [11]

Les modules XBee présentent des hautes performances en termes de transmission RF, de consommation électrique et de gestion réseaux.

a. Communication RF :

Les modules XBee présentent une puissance de sortie RF de 10 mW (0dbm). Concernant la portée intérieure, elle peut atteindre au minimum les 30 mètres suivant la nature des obstacles, alors que la portée extérieure peut atteindre 1,2 km en champ libre. Le débit RF est de 250 Kbps alors que le débit à l'interface est au maximum de 115.2 Kbps. Quand au récepteur, il possède une sensibilité de -92 dbm.

b. Gestion réseaux :

Les modules XBee présentent des caractéristiques avancées en matière de gestion de réseaux. En effet, il est possible de configurer les modules sur près de 65.000 adresses différentes ainsi que de disposer de communications point à point, multipoints ou broadcast.

c. Consommation :

Le module XBee est alimenté sous une tension comprise entre 2.8 et 3.4 V. Pour une alimentation de 3.3 V, la consommation est de 45 mA en émission et de 50 mA en réception.

6.2) Description du module XBee : [11]

Les modules XBee présentent les propriétés physiques suivantes :

- ü Dimensions : 2.438 cm x 2.761 cm
- ü Poids : 3g
- ü Températures de fonctionnement : -40 °C à 85° C

Le module XBee se monte sur une carte à l'aide d'un support permettant de connecter ses 20 broches comme le montre la figure 12. Ainsi aucune soudure n'est nécessaire pour câbler ce composant.

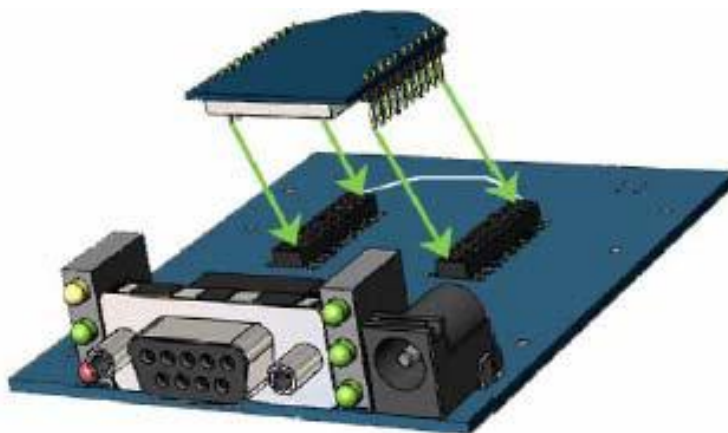


Figure 12 : Montage d'un module XBee sur une carte avec une interface RS232

Le tableau 2 présente le brochage du module XBee :

Broche	Nom	Direction	Description
1	VCC	-	Alimentation
2	DOUT	Out	Sortie UART
3	DIN/CONFIG	In	Entrée UART
4	DO8	Out	Sortie digitale 8
5	RESET	-	Reset (au moins 200 ns)
6	PWM0/RSSI	Out	Sortie PWM0/Indication puissance Rx
7	PWM1	Out	Sortie PWM1
8	Réservé	-	-
9	DTR*/SLEEP_RQ/DI8	In	Contrôle Sleep/Entrée digitale 8
10	GND	-	Ground
11	AD4/DIO4	Inout	Entrée analogique 4 ou E/S digitale 4
12	CTS/DIO7	Inout	Clear To Send/ E/S digitale 7
13	ON/SLEEP	Out	Indicateur état
14	VREF	-	Tension de référence pour conversion
15	Associate/AD5/DIO5	Inout	Indication association/Entrée analogique 5 ou E/S digitale 5
16	RTS/AD6/DIO6	Inout	Reday To Send/Entrée analogique 6 ou E/S digitale 6
17	AD3/DIO3	Inout	Entrée analogique 3 ou E/S digitale 3
18	AD2/DIO2	Inout	Entrée analogique 2 ou E/S digitale 2
19	AD1/DIO1	Inout	Entrée analogique 1 ou E/S digitale 1
20	AD0/DIO0	Inout	Entrée analogique 0 ou E/S digitale 0

Tableau 2: les différentes broches du module XBee

6.3) Communication avec le module XBee : [11][14]

Le module XBee peut être connecté à n'importe quel processeur possédant une interface série UART comme indiqué dans la Figure 13.

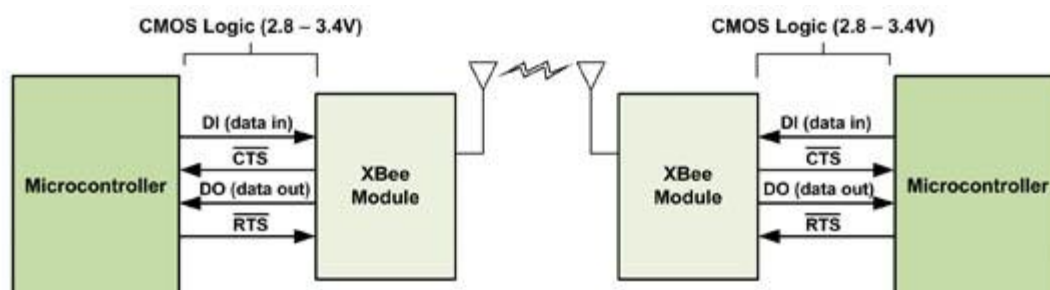


Figure 13: Interfaçage du module XBee à un processeur

Le processeur envoie les données (caractères) sur le port DI en mode série asynchrone. Chaque caractère est composé d'un bit de start (niveau logique 0), suivi de 8 bits de données avec le bit de poids faible en premier et enfin un bit de stop.

La figure 14 présente le mode de fonctionnement interne du module XBee.

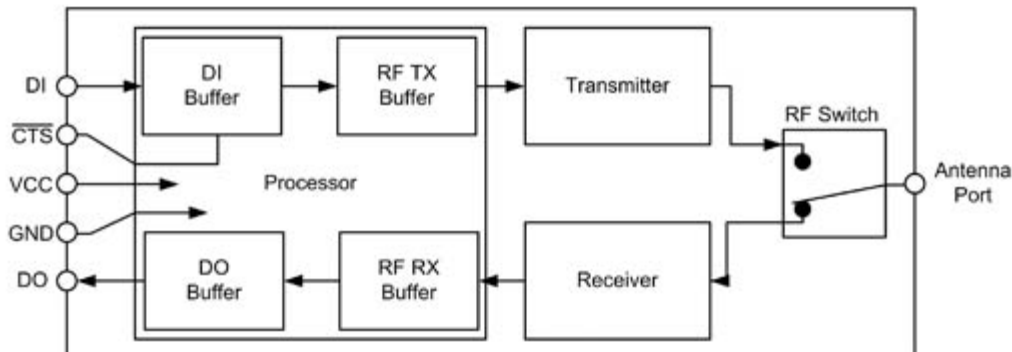


Figure 14: Structure interne du module XBee

Il existe un buffer à l'émission (DI buffer) et un buffer à la réception (DO buffer). Un contrôle de flux matériel peut être mis en place en utilisant le traditionnel couple CTS/RTS pour gérer le flux à l'émission et à la réception des données suivant le taux de remplissage des buffers (comme dans un circuit UART classique).

7) Interface série RS232 : [9]

Une liaison série est une ligne où les bits d'information (1 ou 0) arrivent successivement, soit à intervalles réguliers (transmission synchrone), soit à des intervalles aléatoires (transmission asynchrone). La liaison RS232 est une liaison série asynchrone.

Ø Principe d'une liaison asynchrone:

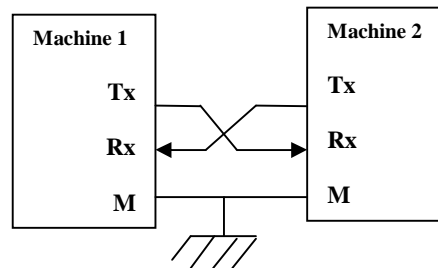


Figure 15: Principe d'une liaison RS232

L'octet à transmettre est envoyé bit par bit (**poids faible en premier**) par l'émetteur sur la ligne Tx vers le récepteur sur la ligne Rx qui le reconstitue. La vitesse de transmission de l'émetteur doit être identique à la vitesse d'acquisition du récepteur. Ces vitesses sont exprimées en **BAUDS** (1 baud correspond à 1 bit / seconde). Il existe différentes vitesses normalisées: 9600, 4800, 2400, 1200... Bauds. La communication peut se faire dans les deux sens, soit émission d'abord puis réception (Half-duplex), soit émission et réception simultanés (full-duplex).

La transmission étant du type asynchrone (pas d'horloge commune entre l'émetteur et le récepteur), des bits supplémentaires sont indispensables au fonctionnement: bit de début de mot (**START**), bit(s) de fin de mot (**STOP**). D'autre part, l'utilisation éventuelle d'un bit de **parité**, permet la détection d'erreurs dans la transmission.

8) Le bus I2C : [3]

Le bus I2C permet de faire communiquer entre eux des composants électroniques divers grâce à seulement trois fils : Un signal de donnée (SDA), un signal d'horloge (SCL), et un signal de référence électrique (Masse).

Ceci permet de réaliser des équipements ayant des fonctionnalités très puissantes (En apportant toute la puissance des systèmes à microprocesseur) en conservant un circuit imprimé très simple, par rapport à un schéma classique (8bits de données, 16 bits d'adresse plus les bits de contrôle).

Les données sont transmises en série à 100Kbits/s en mode standard et jusqu'à 400Kbits/s en mode rapide. Ce qui ouvre la porte de cette technologie à toutes les applications où la vitesse n'est pas primordiale.

Les données sont transmises en série de manière synchrone, cela signifie que les informations sont envoyées à la suite sur le même fil (une donnée par coup d'horloge)

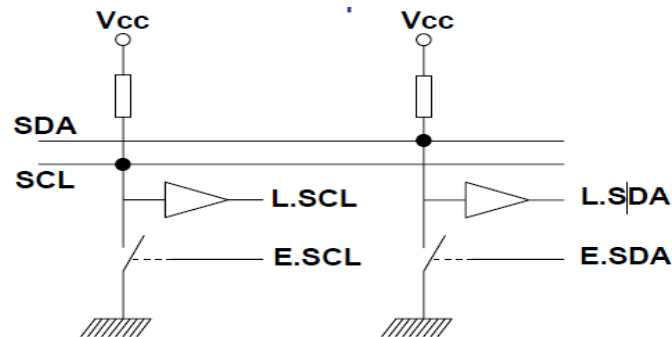


Figure 16: structure d'E/S d'un module I2C

Ø Câblage du bus I2C:

Les lignes SCL et SDA sont à VDD si aucune donnée n'est transmise. Pour mettre un 1 sur SCL ou SDA, le PIC envoie un 0 c'est-à-dire qu'il relie la ligne à la masse, de cette façon il ne peut jamais y avoir de conflit électrique (court-circuit).

Conclusion :

L'étude théorique développée ci-dessus nous permettra d'entamer enfin notre application en ayant suffisamment de connaissance sur les capteurs, la transmission Radio Fréquence (en occurrence la transmission avec le module XBee du protocole Zig Bee).

Chapitre 2 :

Conception matérielle.

Introduction :

Notre conception est basée sur la logique programmable, pour bien montrer notre travail, on a divisé nos deux cartes en plusieurs blocs selon la fonction attribué à chacune d'elles, ensuite on a étudié chaque bloc séparément.

1) Schéma synoptique :

Notre travail consiste en la conception de deux cartes, la première dite carte d'émission, la seconde carte de réception.

La carte d'émission est chargé de la mesure de deux grandeurs, l'une est analogique, elle représente la concentration du monoxyde de carbone (CO) rejeté par l'échappement d'une voiture, cette mesure se fera par le capteur de gaz HS134, quand à la deuxième grandeur elle est numérique, elle représente la température, cette mesure se fera par le capteur de température DS 1621, les mesures effectuées par ces deux capteurs seront continuellement transférées vers le PIC 16F876A via l'entrée analogique pour le HS134 et le bus I2C pour le DS 1621.

La carte de réception est chargée de la réception des données qui lui seront transmises par la carte d'émission via le module XBee, de leur acquisition par le PIC 18F452. Ce dernier sera chargé de gérer un afficheur LCD (4X16) qui affichera les données acquises par la carte d'émission, et qui seront daté par le dateur DS 1302 et enregistré dans une mémoire externe EEPROM (la 24C64A) chaque minute. Dans cette carte on utilisera les ports d'entrée/sortie pour l'afficheur LCD, les ports CCP1 et CCP2 pour le DS 1302, et le port I2C pour l'EEPROM. Le module XBee est relié à chacun des PIC par le bus RS232.

Le schéma synoptique ci-dessous nous donne un aperçu global et général sur notre conception. La liaison entre les deux cartes s'effectue via le module XBee du protocole de transmission sans fil ZigBee qui est basé sur la norme IEEE 802.15.4.

Le système est composé de deux cartes comme suite :

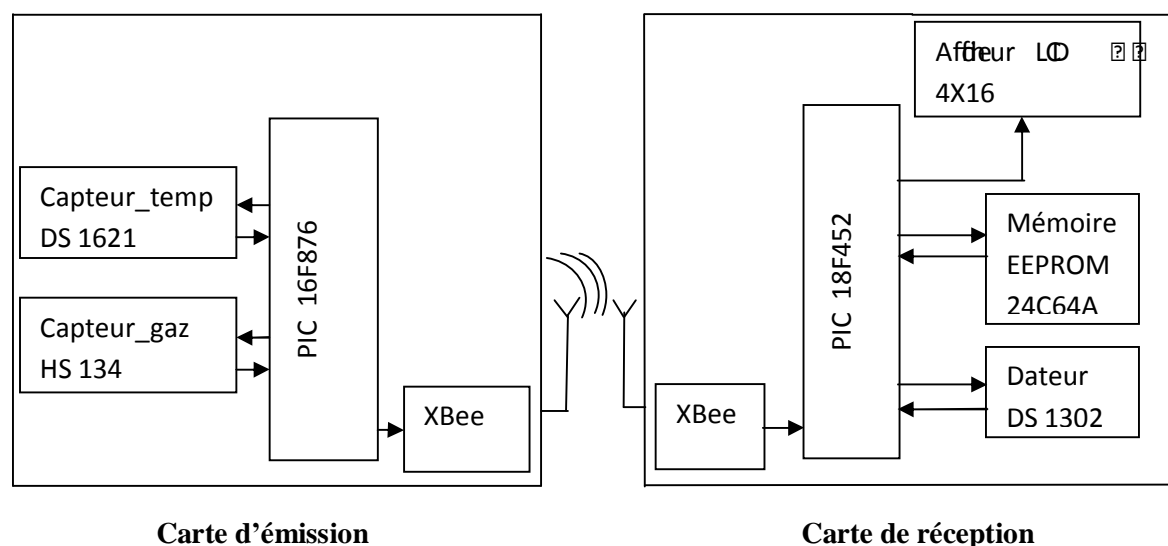


Figure 17 : Le schéma synoptique de fonctionnement

2) Description des différents éléments de la première carte :

2.1) Bloc capture :

On a utilisé le capteur **HS 134** pour mesurer la quantité de **CO** présent dans le gaz d'échappement, et pour mesurer la température on a utilisé le capteur **DS 1621**.

2.1.1) Le capteur HS 134 : [20]

Il existe sur le marché, de nombreux capteurs capables de détecter un gaz dangereux. Pour déceler le taux de monoxyde de carbone (CO) que rejette l'échappement d'une voiture, on a utilisé dans notre système le capteur HS 134.

a. Structure du capteur HS 134:

La structure du capteur HS 134 est donnée par la figure suivante.

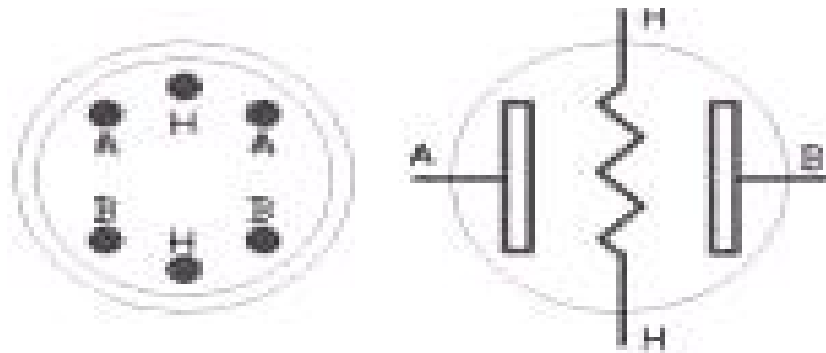


Figure 18 : Structure du capteur HS 134.

Le capteur HS 134 contient 6 broches, 4 entrées sont utilisées pour prendre des signaux (**A**, **B**), et 2 autres (**H**) sont utilisés pour fournir du courant de chauffage.

b. Principe de détection :

Le capteur **HS 134** fait partie des capteurs de gaz à base d'oxyde métallique, en effet, le cœur de ce capteur est un matériau semi-conducteur qui a la propriété d'avoir une conductivité variable selon la composition de l'atmosphère qui l'entoure, comme le montre la figure 19.

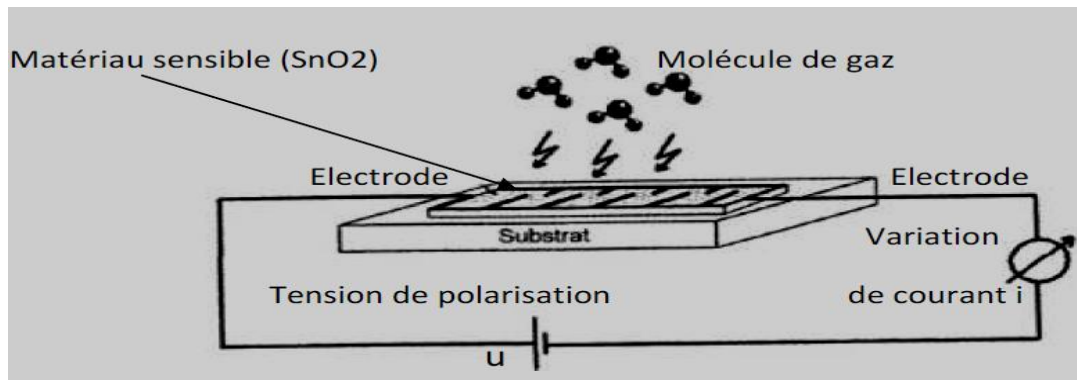


Figure 19 : Principe de détection du capteur HS134.

L'élément sensible de ce capteur est constitué d'une couche semi-conductrice poreuse à base de dioxyde d'étain (SnO₂).

Ce semi-conducteur a la propriété d'offrir une faible conductivité en présence d'un environnement en air propre. Dès qu'il y a présence d'un gaz détectable, la conductivité augmente (cela correspond à la variation ohmique de la couche sensible du capteur) selon la concentration du gaz dans l'air.

La conductivité de la couche semi-conductrice est donc modulée par l'absorption d'éléments chimiques auxquels le capteur est sensible. Afin d'augmenter la sensibilité de la couche sensible à base de SnO₂, il est possible d'introduire des dopants comme la platine (Pt).

Précisons également que ce type de capteurs est fabriqué avec une couche métallique résistive généralement en poly silicium, dont le rôle est de chauffer la couche sensible jusqu'à sa température de fonctionnement, cette couche résistive se trouve sous le matériau semi-conducteur (SnO₂) mais elle est bien sûr isolée de celui-ci.

Pour mesurer la variation de la résistance du matériau semi-conducteur (SnO₂) utilisé pour la fabrication du capteur, le principe consiste à mesurer la résistance de surface R_s du capteur qui est obtenue par la tension de sortie au borne de la résistance de charge R_L . La relation entre eux est donnée par l'équation suivante :

$$R_s \setminus R_L = (V_c - BRV) / BRV$$

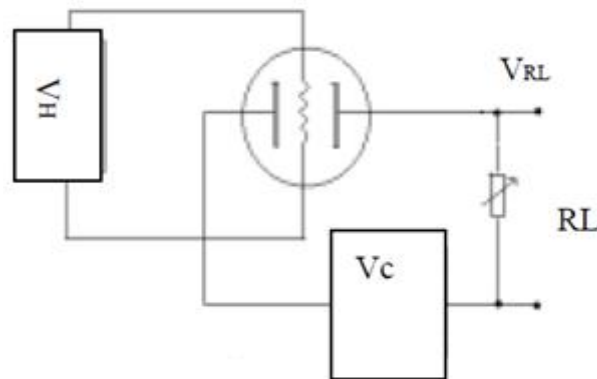


Figure 20 : Principe de mesure.

c. Caractéristiques du capteur HS 134 :

Le capteur HS 134 est conçu pour détecter la concentration dans l'air de différents types de gaz tel que le CO, l'hydrogène H_2 , et l'alcool.

Les courbes de la figure 21, indiquent pour une tension donnée (0-5 V) aux bornes de la résistance R_L la concentration en ppm (partie par million) d'un gaz tel que l'hydrogène (300-500ppm), ces courbes indiquent également un pourcentage entre 20 et 30 % pour la concentration en CO et entre 1 et 5 % pour la détection d'alcool.

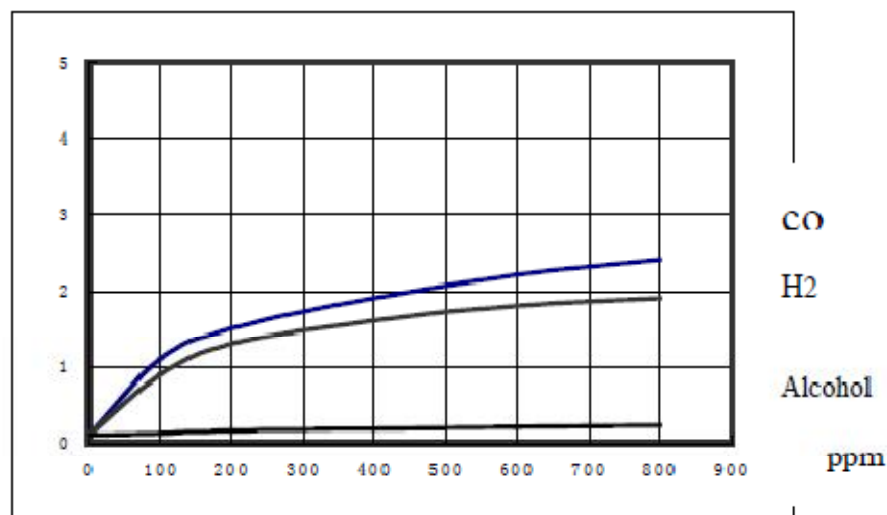


Figure 21 : sensibilité du capteur HS 134.

Remarque : Le ppm (particules par million) est l'unité de mesure des gaz. 1ppm équivaut à 1 cm^3 de particules de CO dans 1 m^3 de gaz.

d. Conditions de travail du capteur HS 134 :

Les conditions de détection standards sont :

- § Température : 20°C
- § Tension d'alimentation, $V_c = 5V \pm 0.1$
- § Humidité : 65 % ± 5 % Rh
- § Tension d'alimentation de la résistance de chauffage $V_h : 4V \pm 0.1$.

Il peut être utilisé dans les gammes indiquées dans le tableau 3.

Le paramètre	Condition technique
Température de fonctionnement	-20°C_50°C
Température de stockage	-20°C_70°C
L'humidité en fonctionnement	Moins de 95% Rh
Concentration d'oxygène (O ₂)	21 % (condition standard) pas moins de 2 %

Tableau 3 : Conditions de travail du capteur HS 134

e. Liaison du capteur HS 134 au PIC 16F876A : [8]

Dans notre réalisation on a relié le capteur HS 134 à la broche RA0/AN0 du PIC, sachant qu'il n'est pas un capteur intelligent, et que la donnée mesurée (le monoxyde de carbone) est analogique.

La transmission de ces données doit être numérique, et c'est là qu'intervient le PIC qui convertit les données analogiques qui lui sont transmises par le capteur en données numériques pour qu'elles soient transmises vers la carte de réception via le module XBee.

Avant d'entamer notre programme on a sélectionné la broche RA0/AN0 du PIC dans le logiciel (PIC C Compiler) comme sera expliqué et détaillé dans le chapitre III.

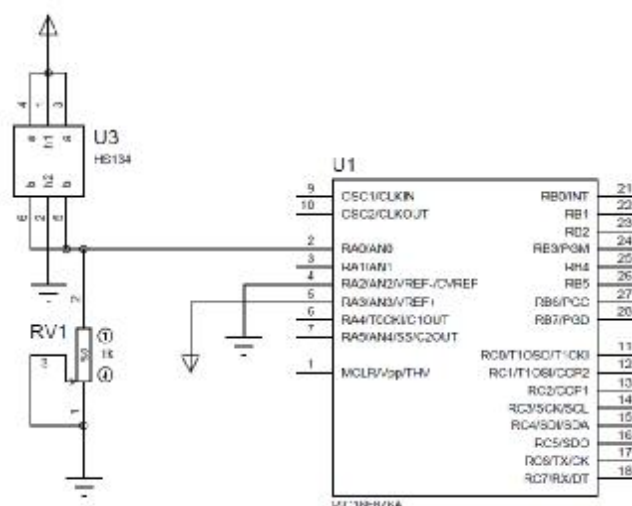


Figure 22: Liaison du capteur HS134 au PIC 16F876A

2.1.2) Le capteur DS 1621 :

a. Définition du capteur DS 1621 : [11] [21]

Le capteur de température DS 1620 (**Dallas Semiconductor**) fait partie de la famille des « capteurs intelligents ». Sur la même puce, il y a un capteur de température classique associé à une électronique d'interface (convertisseur analogique/numérique, contrôleur avec son jeu d'instructions, EEPROM, port série synchrone 2 fils).

Ce capteur peut mesurer une température à l'origine variant de -55°C à 125°C avec une précision de $0,5^{\circ}\text{C}$, mais dans notre application on travaillera avec une précision de 1°C .

b. Transmission de données avec le DS 1621 :

Le DS 1621 communique avec le PIC via le protocole de transmission I2C, l'information qui est sur 9 bits est véhiculée sur la broche SDA. Quand à la broche SCL, elle s'occupe de la synchronisation du transfert des données (signal d'horloge).

c. Les broches du capteur DS1621 : [14] [11]

SDA : chemin des données de port série (Input/Output).

SCL : Horloge du port série

GND : la masse.

T_{OUT} (Thermostat Output Signal) : signal de sortie.

A₀ : Chip Adresse d'entrée "Input".

A₁ : Chip Adresse d'entrée "Input".

A₂ : Chip Adresse d'entrée "Input".

VDD : tension d'alimentation électrique.

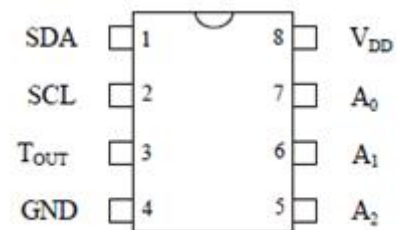


Figure 23 : broches du capteur DS1621

d. Liaison du capteur DS 1621 au PIC 16F876A :

Dans notre conception on a programmé le PIC en maître et le DS 1621 en esclave, et comme on a utilisé un seul capteur, on a relié les broches A₀, A₁ et A₂ à la masse, et les broches SDA et SCL sont reliées au PIC pour la transmission de données via le protocole I2C, comme illustré dans la figure 24.

Remarque : les broches d'adresse d'entrée A₀, A₁ et A₂ nous permettent d'utiliser 8 (c'est-à-dire 2^3) capteurs DS 1621.

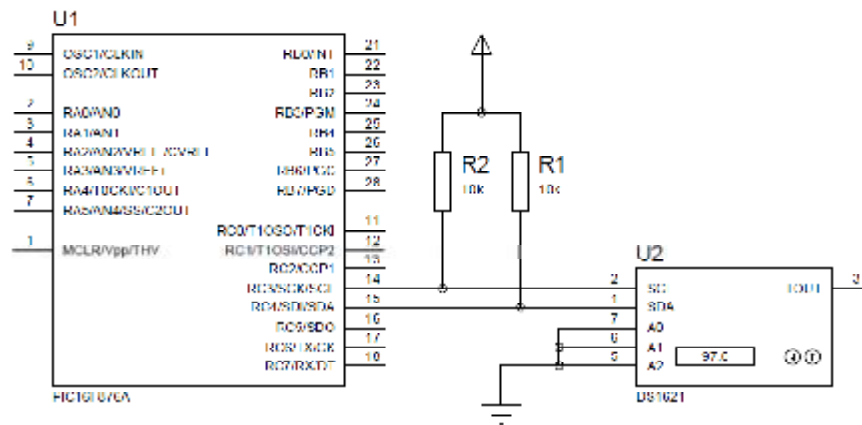


Figure 24 : Liaison du capteur DS 1621 au PIC 16F876A

3) Le PIC 16F876A :

La carte d'émission est architecturé autour du microcontrôleur **PIC 16F876A** (comme on peut le voir sur le schéma synoptique de fonctionnement de la figure 17). En effet, en ayant recours à la logique programmable, son fonctionnement repose essentiellement sur le programme embarqué.

L'intégration du pic 16F876A de l'architecture HAVARD, à notre réalisation est due aux raisons suivantes :

- Ø Le nombre de pin (au nombre de 28) suffisant pour concevoir notre carte.
- Ø La taille (8cm de large et 33mm de long) qui fait qu'il n'est pas trop encombrant, ce qui nous permet de réduire la surface de la carte d'émission.
- Ø La mémoire : une mémoire programme largement suffisante.
- Ø Un prix raisonnable.

3.1) Présentation du PIC 16F876A : [8]

Le pic 16F876A est produit par le constructeur MICROCHIP, conçu selon l'architecture HARVARD, il fait partie de la famille Mid-Range (les mots de la mémoire programme sont codés sur 14 bits).

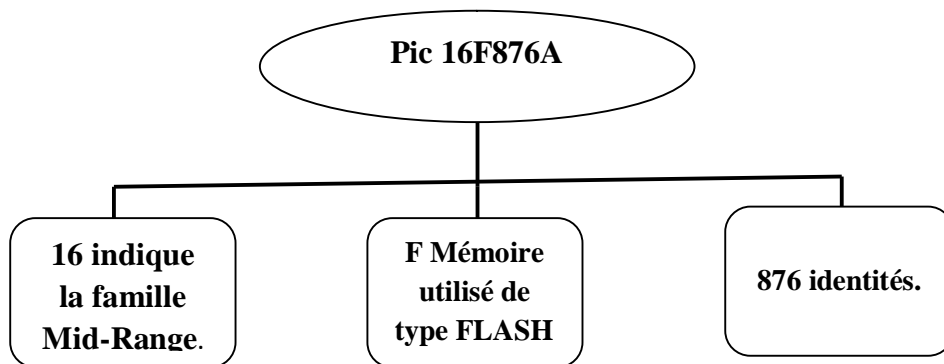


Figure 25 : Configuration du pic 16F876A

Il possède essentiellement les éléments suivants :

- Ü Une mémoire programme de type EEPROM flash de 8K mots de 14 bits.
- Ü Une RAM données de 368 octets.
- Ü Une mémoire EEPROM de 256 octets.
- Ü Fréquence de fonctionnement élevée, jusqu'à 20 Mhz.
- Ü Trois ports d'entrée sortie, A (6 bits, programmable en entrée/sortie numérique ou en entrée analogique sauf RA₄), B (8 bits), C (8 bits).
- Ü Deux modules de comparaison et Capture CCP1 et CCP2
- Ü Trois TIMERS avec leurs Prescalers, TMR0, TMR1, TMR2
- Ü Convertisseur Analogiques/numériques 10 bits à 5 canaux.
- Ü USART, Port série universel, configurable en mode asynchrone (RS232).
- Ü SSP, Port série synchrone supportant I2C.
- Ü Un chien de garde (WDT).
- Ü 13 sources d'interruption.
- Ü Générateur d'horloge, à quartz (20 MHz) ou à Oscillateur RC.
- Ü Fonctionnement en mode sleep pour réduction de la consommation.
- Ü Tension de fonctionnement de 2 à 5.5V.
- Ü Jeux de 35 instructions.

3.2) Schéma bloc du PIC 16F876A et description de chaque broche : [20]

Les entrées/sorties du microcontrôleur peuvent être groupées par fonction comme le montre la figure suivante :

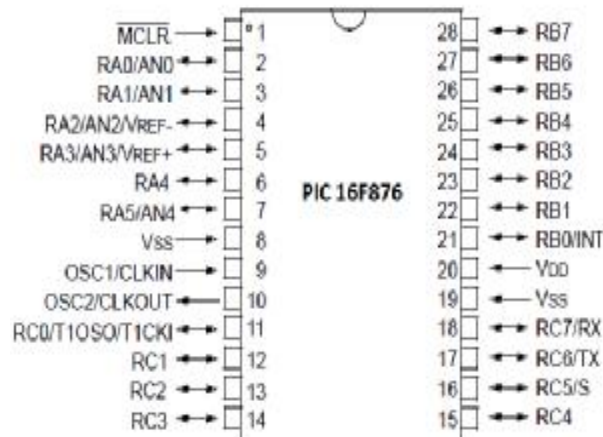


Figure 26 : Broches du PIC 16F876A

Le PIC 16F876A est un circuit intégré de **28 broches** dont : 22 broches d'entrées/sorties, 3 broches pour l'alimentation, 2 broches pour l'oscillateur et une broche pour le reset (MCLR).

a. Les ports d'E/S :

Le PIC 16F876A dispose de **3 ports** (A, B, C) d'entrées/sorties (**I/O**) qui sont parallèles et bidirectionnels.

Ø Le **PORT A** (6 bits) **I/O** pure et/ou convertisseur analogique et/ou **TIMER 0**.

La broche **RA4** du **PORT A** (Entrée du **TIMER 0 T0CKI**) est du type **DRAIN OUVERT**.

Ø Le **PORT B** (8 bits) **I/O** pure et/ou programmation in situ **ICSP/ICD** (Broche **RB3/PGM**, **RB6/PGC** et **RB7/PGD**) et l'entrée d'interruption externe **RB0/INT**.

Ø Le **PORT C** (8 bits) **I/O** pure et/ou **TIMER 1** et/ou **SPI / I2C** et/ou **USART**.

Chacune des lignes de **PORT** peut fournir un courant de 25mA.

b. L'interruption :

L'interruption est une action extérieure à un programme obligeant celui-ci à traiter un sous programme prioritaire logé dans une adresse bien spécifique de la mémoire du microcontrôleur. Cet événement extérieur n'est pas synchronisé avec le programme principal, il peut survenir n'importe quand.

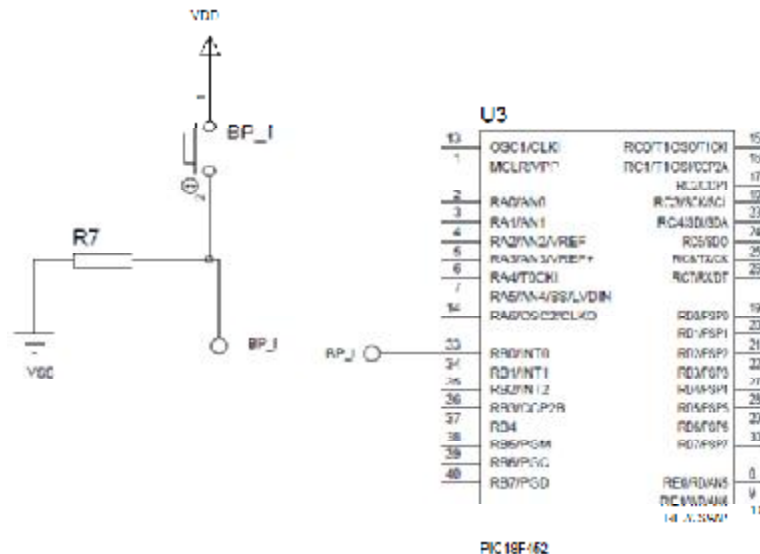


Figure 27 : L'interruption

c. Configuration minimale du PIC 16F876 :

Pour réaliser une carte d'acquisition à base du PIC 16F876A, il faut au moins configurer son alimentation, le MCLR et l'horloge (voir figure 28).

Ü L'alimentation :

Le PIC fonctionne dans la plage de tension qui s'étend de 2.0V à 5.5V.

Ü Le circuit d'initialisation :

Le MCLR est habituellement relié au 5V. Cette broche est utilisée pour effectuer un reset du composant en cas de connexion à la masse et cela en utilisant un bouton poussoir.

Ü L'oscillateur :

L'oscillateur est le cœur du microcontrôleur, c'est lui qui cadence le déroulement du programme. Il fournit un signal carré périodique au microcontrôleur.

Le DATASHEET du composant fournit 3 types d'oscillateurs :

- Ü Un quartz ou résonateur céramique.
- Ü Un oscillateur externe.
- Ü Un réseau RC.

Pour notre cas, nous avons opté pour la méthode avec l'oscillateur à Quartz.

Un quartz se met à osciller lorsqu'il est soumis à une différence de potentiel. La fréquence centrale du quartz est très précise et varie peu dans le temps.

Les broches OSC1 et OSC2 permettent de faire fonctionner l'oscillateur interne du PIC.

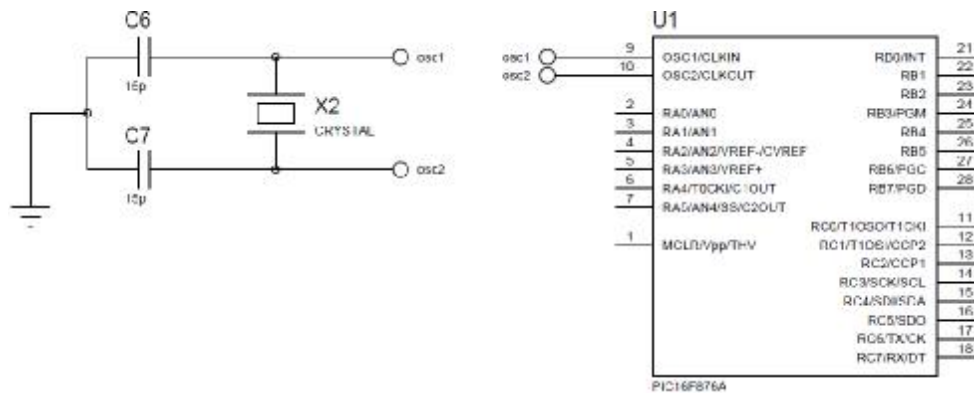


Figure 28 : Configuration minimale du PIC 16F876A.

4) Description des différents éléments de la deuxième carte :

4.1) Description du PIC 18F452 : [9]

Dans la deuxième carte nous avons utilisé le PIC 18F452 pour les raisons suivantes :

- Ø La mémoire de programme est de type FLASH de capacité 32Koctet.
- Ø Nombre de broches (40), largement suffisant pour notre carte.
- Ø Un prix raisonnable.

4.2) Présentation du PIC 18F452 : [9]

Le pic 18F452 est lui aussi produit par le constructeur MICROCHIP, et conçu selon l'architecture HARVARD, il fait partie de la famille Mid-Range.

Ø La taille de la mémoire programme :

Le PIC 18F452 dispose d'une mémoire de programme de type FLASH de 32Koctet de capacité. Chaque adresse est codée sur 2octets ce qui confère à ce PIC une large utilisation lors de la réalisation d'un programme laborieux.

Ø La vitesse de l'horloge :

Notre PIC est piloté par un quartz de 20MHz (T=50 ns), une instruction sera exécutée toutes les 200ns. Dans ce cas là, le PIC a une puissance de calcul de 5 MIPS (5 millions d'instructions par seconde).

Ø Le nombre d'entrées/sorties :

Au nombre de 5 ports (port A, B, C, D et E) ils offrent, en plus de la fonction classique Entrée/Sortie, plusieurs fonctions spéciales ainsi que plusieurs modes de fonctionnement.

4.3) Les broches du PIC 18F452 :

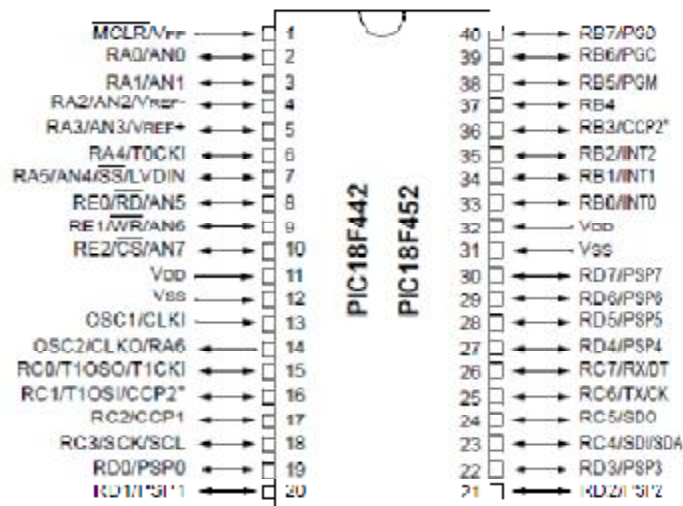


Figure 29: Broches du PIC 18F452

5) Bloc affichage : [9][20]

Ce bloc est constitué d'un afficheur **LCD** (Liquide Crystal Display), d'une EEPROM **24C64A** et d'un dateur **DS1302**.

5.1) Afficheur LCD : [9]

Les afficheurs à cristaux liquides sont des modules compacts intelligents et nécessitent peu de composants externes pour un bon fonctionnement. Ils sont relativement bons marchés et s'utilisent avec beaucoup de facilité.

L'afficheur LCD « 4x16 » (4 lignes de 16 caractères) est directement connecté aux broches du PIC et est alimenté par 5v.

Et comme le PIC ne fait aucune interprétation des codes de commandes des afficheurs, il est compatible à tous les modèles existants (de 1 à 4 lignes de 6 à 80 caractères).

Il existe deux interfaces normalisées : une version « parallèle » et une autre « série », or dans notre conception, l'afficheur communique avec le PIC via le **PORT B** en version parallèle où le transfert de données se fait en deux fois (sur 4 bits).

5.2) Les broches de l'afficheur LCD :

N° de pastille	Appellation	Fonction
1	Vss	Masse
2	Vdd	Alimentation de 5V
3	Vee	Tension a appliqué pour gérer le contraste
4	RS	Registre Selecte (mode instruction ou données)
5	R/W	Ecriture ou Lecture (Read/Write)
6	E	Enable (sélection de l'afficheur)
7-14	DB0-DB7	Data 0 à 7 (Données ou instructions)

Tableau 4: Broches de l'afficheur LCD

5.3) Fonctionnement de l'afficheur LCD :

Dans notre application on a utilisé l'afficheur en écriture (R/W=0), L'afficheur reçoit deux types de commande, soit des instructions permettant de gérer l'afficheur proprement dit (effacement de l'écran, affichage du curseur, clignotement du curseur etc.), soit des données qui seront affichées à la position courante du curseur.

La sélection du mode « instruction » est réalisée en mettant la broche RS à 0 (Registre Select) et en envoyant ensuite l'octet de commande sur le port de données (DB4 à DB7). Lorsque la broche RS est à 1 les valeurs envoyées sur les lignes DB4 à DB7 seront affichées sous forme de caractères ASCII à la position courante du curseur.

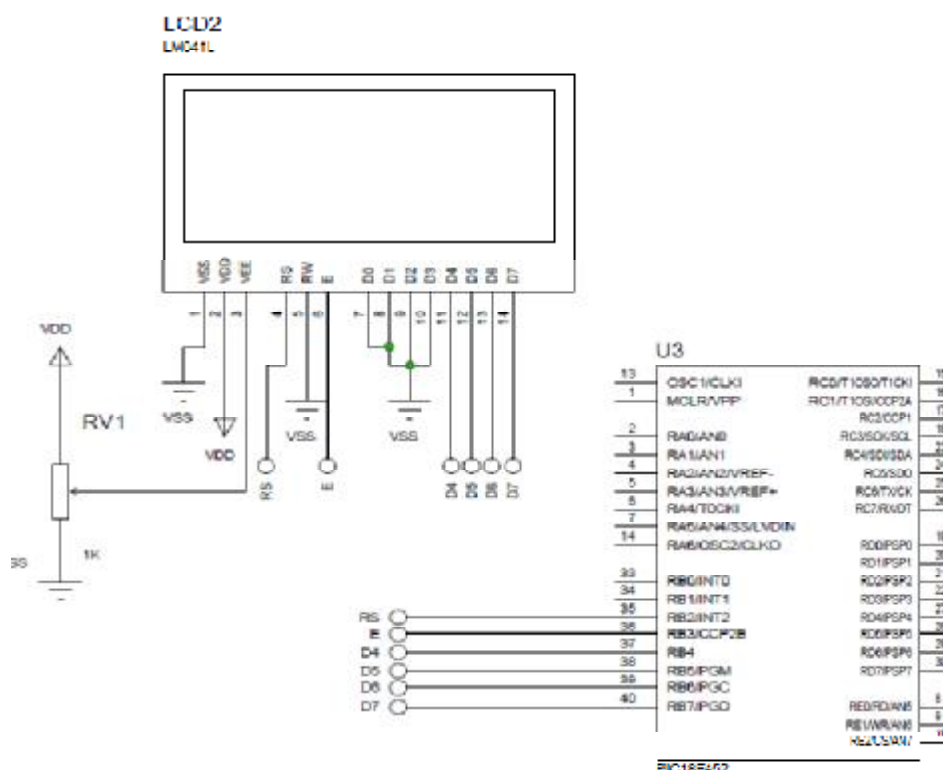


Figure 30: câblage de l'afficheur LCD

5.4) EEPROM (Electrically-Erasable Programmable Read-Only Memory):

Pour certaines applications, il est nécessaire de pouvoir conserver des informations de façon permanente même lorsque l'alimentation électrique est interrompue. On utilise alors des mémoires mortes ou mémoires à lecture seule (ROM : Read Only Memory). Ces mémoires sont non volatiles, contrairement aux RAM.

Il existe plusieurs types de mémoires mortes : ROM, PROM, EPROM, EEPROM, FLASH EPROM.

Pour une éventuelle étude des variations de la température et du taux de monoxyde de carbone, on a besoin de constituer une base de données. A cet effet on a utilisé l'EEPROM « 24C64A ».

La mémoire 24C64A, présente les caractéristiques suivante :

- Programmable et effaçable électriquement à accès série.
- Sa capacité est de 8192 octets ($64 \times 1024 = 65536$ bits), chaque mot mémoire à une taille d'un octet. Au démarrage de notre système, après l'incrémement de la première minute, les quatre données (température, taux de CO, date, heure) seront enregistrées dans les premières adresses de cette EEPROM. Ensuite, à chaque incrémement de minute, seul les deux données suivantes : température, taux de gaz CO, seront enregistrées. Cette mémoire peut sauvegarder des données sur plus de 68 heures tout en ayant la possibilité de connaître l'heure et la date du début et de la fin de sauvegarde, et cela en enregistrant l'heure et la date dans les deux dernières adresses une fois que la mémoire est pleine.
- Elle est reliée au microcontrôleur (PIC) via le bus I2C.

5.5) les broches de l'EEPROM : [20]

A0 - A2 : Entrées de l'adresse

SDA : Données en série

SCL : signal d'Horloge

WP : Protection en écriture

Vss: Masse

Vcc: Alimentation (5v)

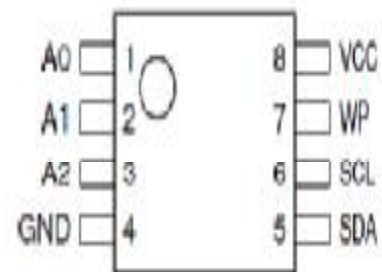


Figure31 : EEPROM 24C64A

5.6) Brochage de l'EEPROM au PIC 18F452:

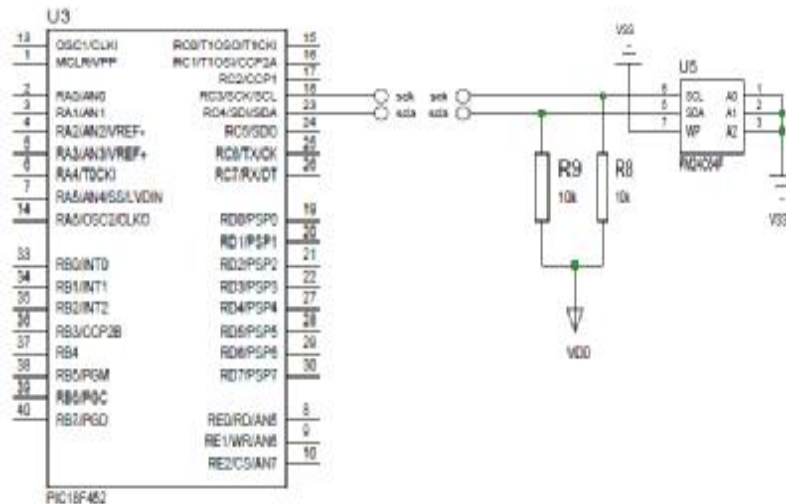


Figure 32: câblage de l'EEPROM

5.7) Le dateur DS1302 : [20]

Le circuit Dallas DS1302 est une horloge temps réel (Real Time Clock), qui fournit : secondes, minutes, heures, jours, dates, mois et années. Il comporte en plus 31 cases mémoire de 8bits.

5.8) Brochage du DS 1302 :

Le boîtier comporte deux accès d'alimentation (les broches 1 et 8), la broche 1 est reliée à l'alimentation de 5v tandis que la broche 8 est reliée à un super condensateur qui est maintenu en charge par le circuit et assure la sauvegarde de l'heure et de la date en cas de coupure de l'alimentation principale, tandis que les broches 5, 6 et 7 sont reliées au PIC via le bus I2C pour assurer le transfert de données, la broche 4 est reliée à la masse, quand aux broches 2 et 3 au quartz de 32.768kHz.

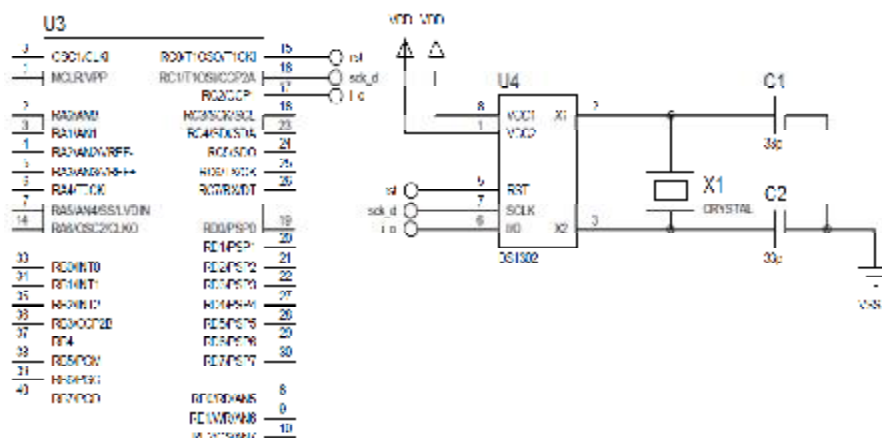


Figure 33 : câblage du capteur DS1302

5.9) Brochage des boutons poussoirs avec le PIC :

Notre montage comporte six boutons poussoirs qui ont les fonctions suivantes : (voir Figure 34).

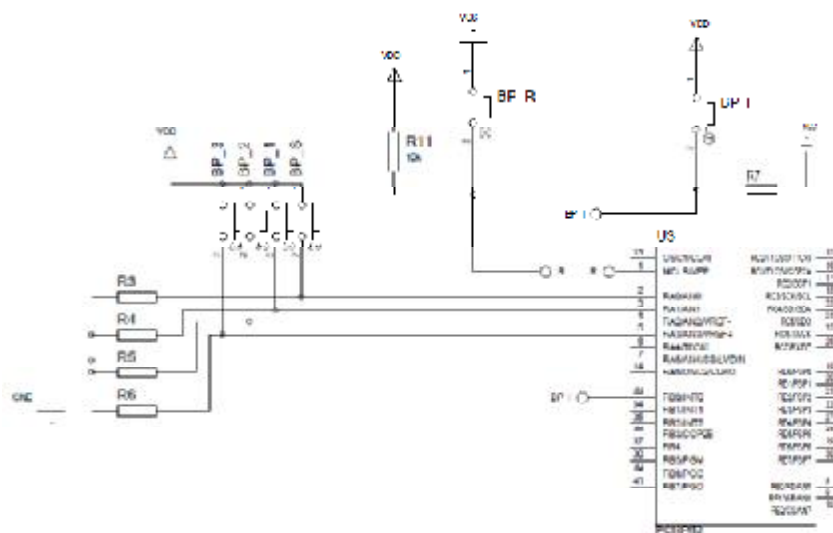


Figure 34 : Brochage des boutons poussoirs avec le PIC 16F876A

BP_R : bouton poussoir qui permet de provoquer le RESET (redémarrage du PIC).

BP_I: bouton poussoir qui permet de provoquer l'interruption et d'en sortir pour régler heure et date.

BP_S: bouton poussoir qui permet de sélectionner le réglage de l'heure ou de la date/ ou bien la lecture à partir de l'EEPROM externe.

BP_1: bouton poussoir qui permet d'incrémenter l'heure ou les jours /ou bien d'incrémenter les données de l'EEPROM externe lus sur l'afficheur LCD.

BP_2: bouton poussoir qui permet d'incrémenter les minutes ou les mois /ou bien décrémenter les données de l'EEPROM externe lus sur l'afficheur LCD.

BP_3: bouton poussoir qui permet d'incrémenter les secondes ou les années /ou bien d'effacer le contenu de l'EEPROM externe.

6) Les composants présents sur les deux cartes :

6.1) Bloc alimentation :

Pour tout équipement électronique, il est indispensable d'avoir une source d'alimentation, qui peut être soit une batterie, soit une tension prélevée d'un secteur.

Notre montage doit être alimenté à partir d'une batterie de 12V, car notre dispositif est destiné à être utilisé comme appareil portatif dans une voiture.

La batterie de la voiture nous fournit une tension de 12V nécessaire à l'alimentation de nos deux cartes. On a placé une diode à la sortie de la borne positive de la batterie pour éviter les tensions inverses (inversion de polarité), quand au condensateurs ils ont pour but le filtrage et le lissage de la tension pour éviter tout pic de tension qui peut nuire au fonctionnement de notre système et dans le pire des cas sa détérioration.

Comme nos composants fonctionnent avec une tension de 5V, et qu'en entrée on a une tension de 12V, on a utilisé le régulateur de tension 7805.

Le schéma électrique du bloc alimentation est donné par la figure 35.

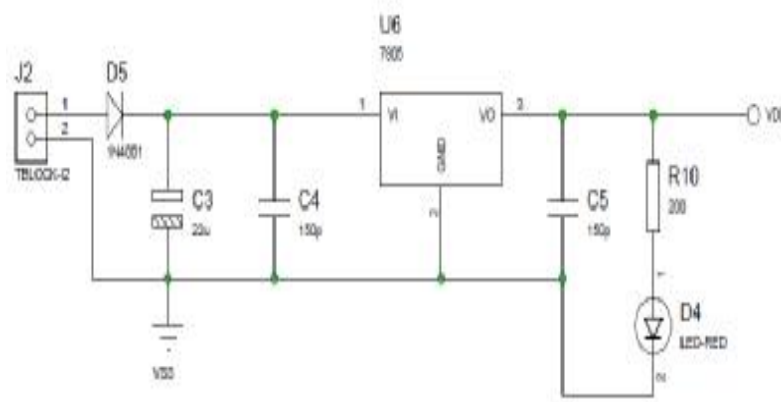


Figure 35 : Schéma électrique du bloc alimentation.

6.2) Le module XBee :

Concernant la technologie ZigBee, notre choix s'est porté sur le produit XBee PRO. Ce module nous est fourni en kit, ce kit contient deux platines, l'une avec support RS232 et l'autre avec un support USB. Sachant que l'alimentation de l'XBee PRO est de 3.3V, ces deux platines intègrent des convertisseurs de tension de 5V à 9V en entrée vers 3.3V en sortie (alimentation de l'XBee).

6.3) Description des modules XBee utilisés: [10]

On a utilisé dans notre système deux types de platine l'une avec support RS232 et l'autre avec support USB, comme détaillé dans ce qui suit :

a. Platine d'interface RS232 pour modules XBee 990.006 :

Le module XBee 990.006 (ou XBEE-INT-RS232) peut être connecté à un périphérique de communication comme un modem, ou à tout autre équipement détenant un connecteur série normalisé DB-9 femelle. Grâce à l'utilisation de cette carte interface intégrée entre un PC et un module de communication XBee, toutes les liaisons câblées seront transformées en une connexion "wireless" avec une vitesse maximale de 115KBaud et une distance pouvant atteindre 1,2 km.

La carte est compatible avec tous les modules XBee 2.4GHz et 868MHz de la série 1 ou de la série 2, en versions standard ou PRO. Sur la platine d'interface RS232 on peut distinguer trois LEDs : la première reporte l'activité de transmission (TX), la seconde la puissance du signal (RSSI), quand à la troisième LED, elle montre l'état associé sur le réseau (ASSOCIATE LED). Le connecteur latéral (9 points) est utilisé pour la programmation du Xbee ou pour l'alimentation de périphériques nécessitant une source externe de 3,3 VDC.

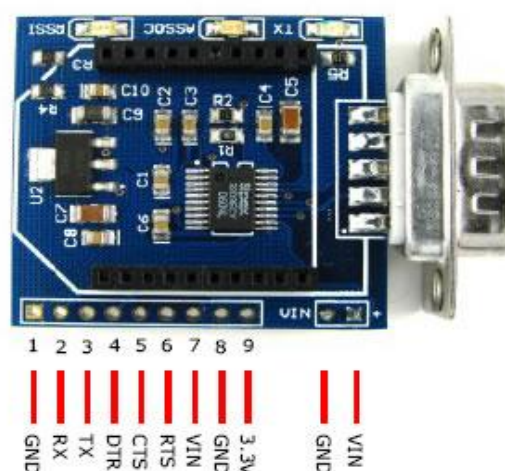


Figure 36 : Le module XBee-INT-RS232

Ø Brochage de DB-9 :

Le brochage de DB-9 est donné dans le tableau qui suit :

DB-9 Pin	RS-232 Name	Description	Implementation
1	DCD	Data-Carrier-Detect	Connected to DSR (pin6)
2	RXD	Receive Data	Serial Data exiting the module assembly (to host)
3	TXD	Transmit Data	Serial Data intering into the module assembly(from host)
4	DTR	Data-Terminal-Ready	Can enable Power-down on the module assembly
5	GND	Ground Signal	Ground
6	DSR	Data -Set- Ready	Connected to DCD (pin1)
7	RTS/CMD	Request-to-Send /Command Mode	Enable $\overline{\text{RTS}}$ flow control or commande mode
8	$\overline{\text{CTS}}$	Clear-to-Send	Provides $\overline{\text{CTS}}$ flow control
9	RI	Ring Indicator	Optional power input that is connected internally to the positive lead of the front power connector

Tableau 5 : Broches du support DB-9 sur la platine de l'XBee PRO.

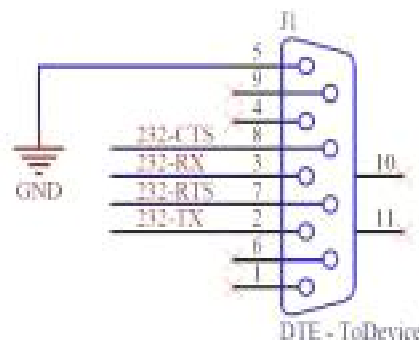


Figure 37 : brochage du connecteur DB-9

b. Platine d'interface USB pour modules XBee 990.002 :

Cette petite platine est idéalement conçue pour raccorder rapidement et simplement un module XBee™ ou XBee PRO™ sur le port USB d'un compatible PC. Cette dernière dispose d'un régulateur 3,3 V intégré, de 4 Leds de visualisation et de 2 connecteurs femelles au pas de 2 mm (sur lesquels les modules XBee™ pourront être insérés).

Une rangée de broches au pas de 2,54 mm présente sur le côté de la platine permet une reprise des principaux signaux du module XBee™. Les 4 Leds restituent l'état des signaux RSSI, "ASSOCIATE", TX et RX. Un mini connecteur USB est destiné à recevoir le câble de raccordement vers le PC.

La platine délivre également une tension de sortie de 3,3 Vcc pouvant être utilisée pour alimenter d'autres dispositifs externes (avec une intensité max. de 140 à 300 mA max. suivant le type de module XBee™ qui aura été inséré sur la platine). Un driver permet de créer un port RS-232 virtuel sur le PC afin de pouvoir piloter le module XBee™ au moyen d'une communication série standard.

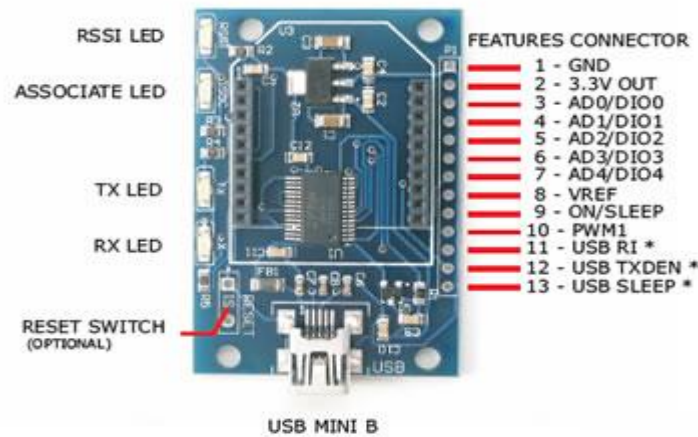


Figure 38 : Platine d'interface USB pour modules XBee 990.002

Le module XBee nous offre une grande souplesse d'utilisation, avec lui on peut relier :

- Ø Plusieurs cartes à microcontrôleurs
- Ø Plusieurs cartes à microcontrôleurs et micro-ordinateurs
- Ø Des micro-ordinateurs

Et cela grâce aux platines qui possèdent différents ports (DB-9, USB,...), et son protocole qui s'adapte à différents environnements. L'image suivante illustre les différentes manières d'utiliser les platines XBee.

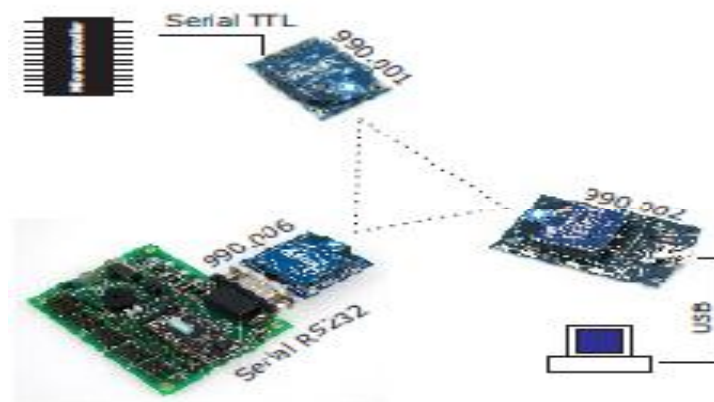


Figure 39 : les différentes platines XBee

Chapitre 3 :

Programmation et réalisation pratique.

1) Définition d'un organigramme :

L'organigramme est une représentation graphique d'un algorithme. C'est donc un diagramme qui montre le cheminement des données dans un programme ou dans un système d'information, ainsi que les opérations pratiquées sur ces données lors des différentes étapes du traitement. Il a pour but de faciliter la compréhension du déroulement du programme.

2) Les organigrammes :

Pour écrire nos programmes on a suivi les organigrammes suivants :

2.1) Déclaration des variables :

Les différentes variables utilisées dans notre programme sont définies ci-dessous :

i : représente les adresses de l'EEPROM (lors de la lecture)

j : représente les adresses (cases mémoires) vierges de l'EEPROM

h : représente les adresses de l'EEPROM (lors de l'écriture)

r : représente une variable qui permet d'incrémenter les minutes

BP_I: bouton poussoir qui permet de provoquer l'interruption et d'en sortir pour régler heure et date.

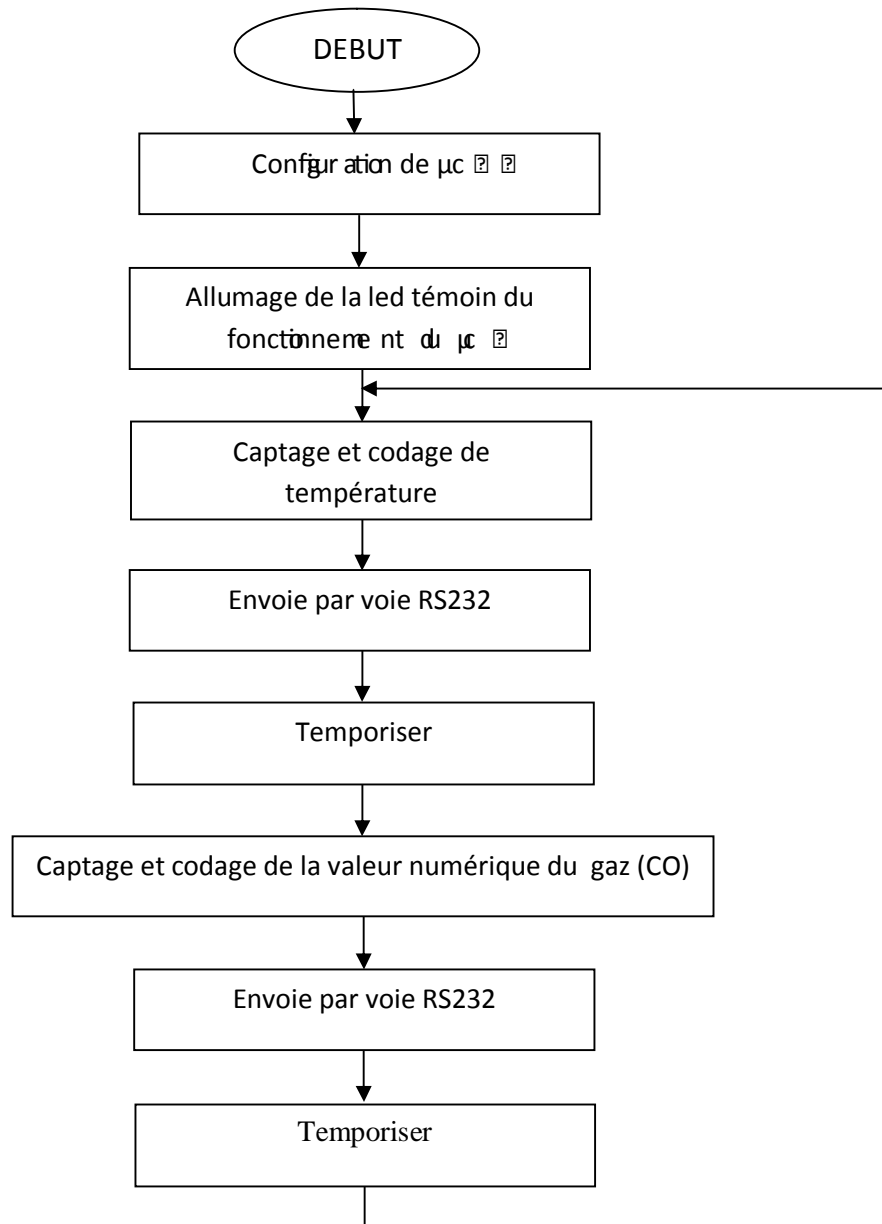
BP_S: bouton poussoir qui permet de sélectionner le réglage de l'heure ou de la date/ ou bien la lecture à partir de l'EEPROM externe.

BP_1: bouton poussoir qui permet d'incrémenter l'heure ou les jours /ou bien d'incrémenter les données de l'EEPROM externe lus sur l'afficheur LCD.

BP_2: bouton poussoir qui permet d'incrémenter les minutes ou les mois /ou bien décrémenter les données de l'EEPROM externe lus sur l'afficheur LCD.

BP_3: bouton poussoir qui permet d'incrémenter les secondes ou les années /ou bien d'effacer le contenu de l'EEPROM externe.

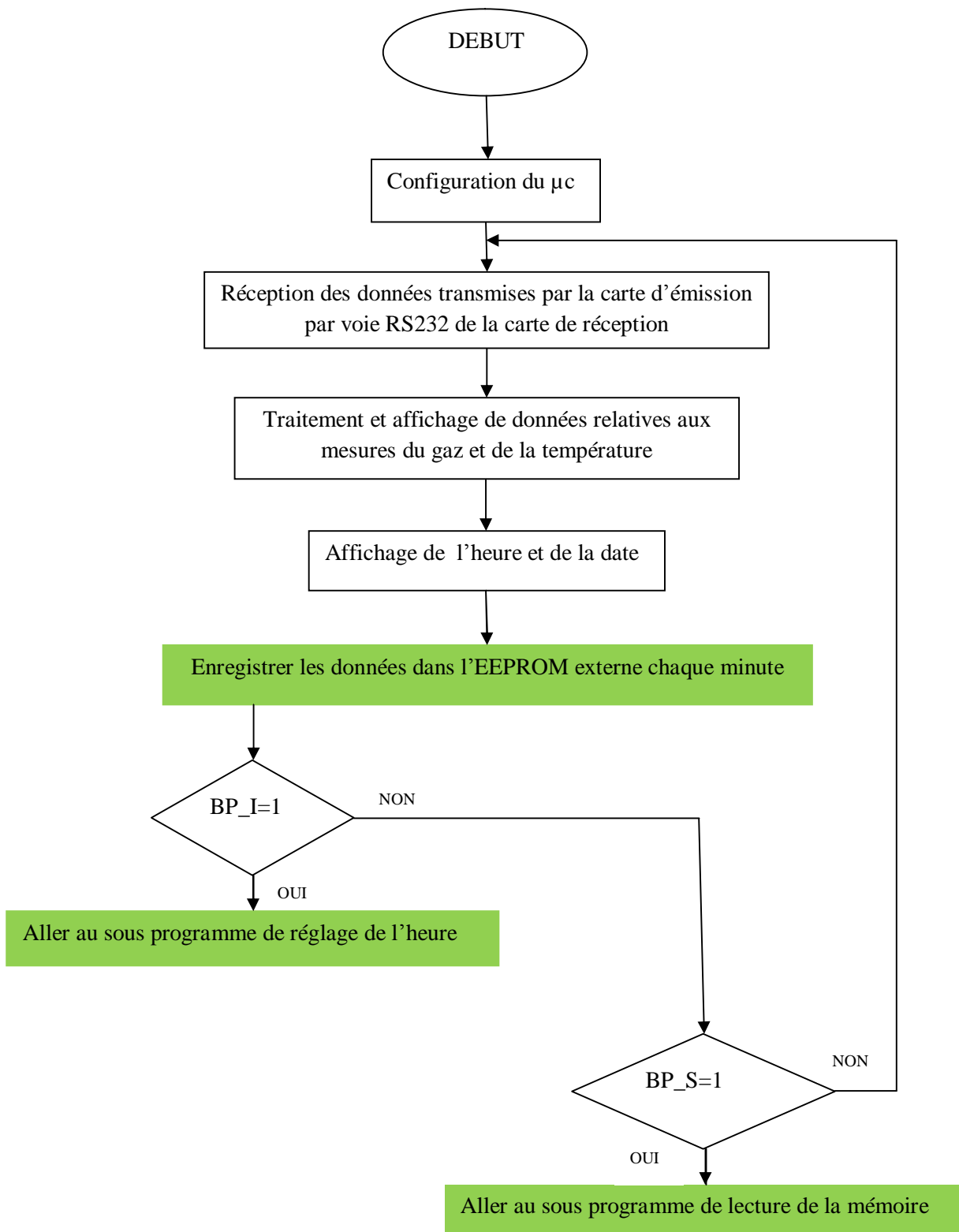
2.2) Organigramme de la transmission:



- **Explication de l'organigramme de transmission :**

La mesure de la température et de la quantité de monoxyde de carbone (CO) à la sortie de l'échappement de la voiture s'effectue d'une manière continue. Ces deux valeurs seront acquises par le microcontrôleur de la carte d'émission et envoyées sur le port RS232 avec une temporisation de 500ms entre chaque donnée et une autre.

2.3) Organigramme de la réception :



- **Explication de l'organigramme de réception :**

Le microcontrôleur de la carte de réception reçoit continuellement sur le port RS232 les données envoyées par la carte d'émission.

Un dateur (le DS 1302) est chargé de donner à temps réel l'heure et la date à laquelle ces deux données sont reçus.

La température, la quantité de CO, l'heure et la date seront affichées sur un afficheur LCD (4x16).

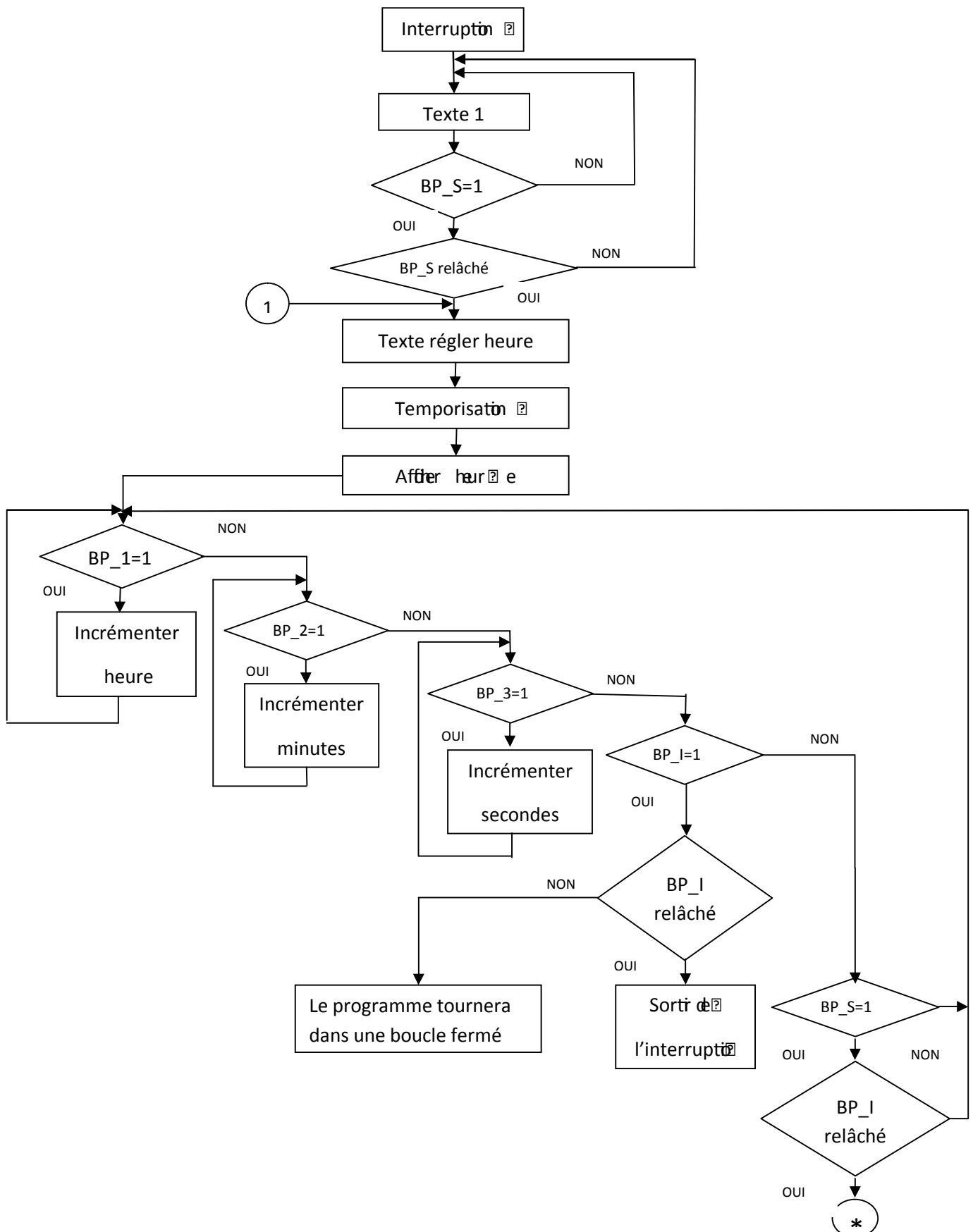
Une mémoire EEPROM externe de 64 Koctet enregistrera les données mesurées (température et taux du monoxyde de carbone) chaque minute à partir de la première minute.

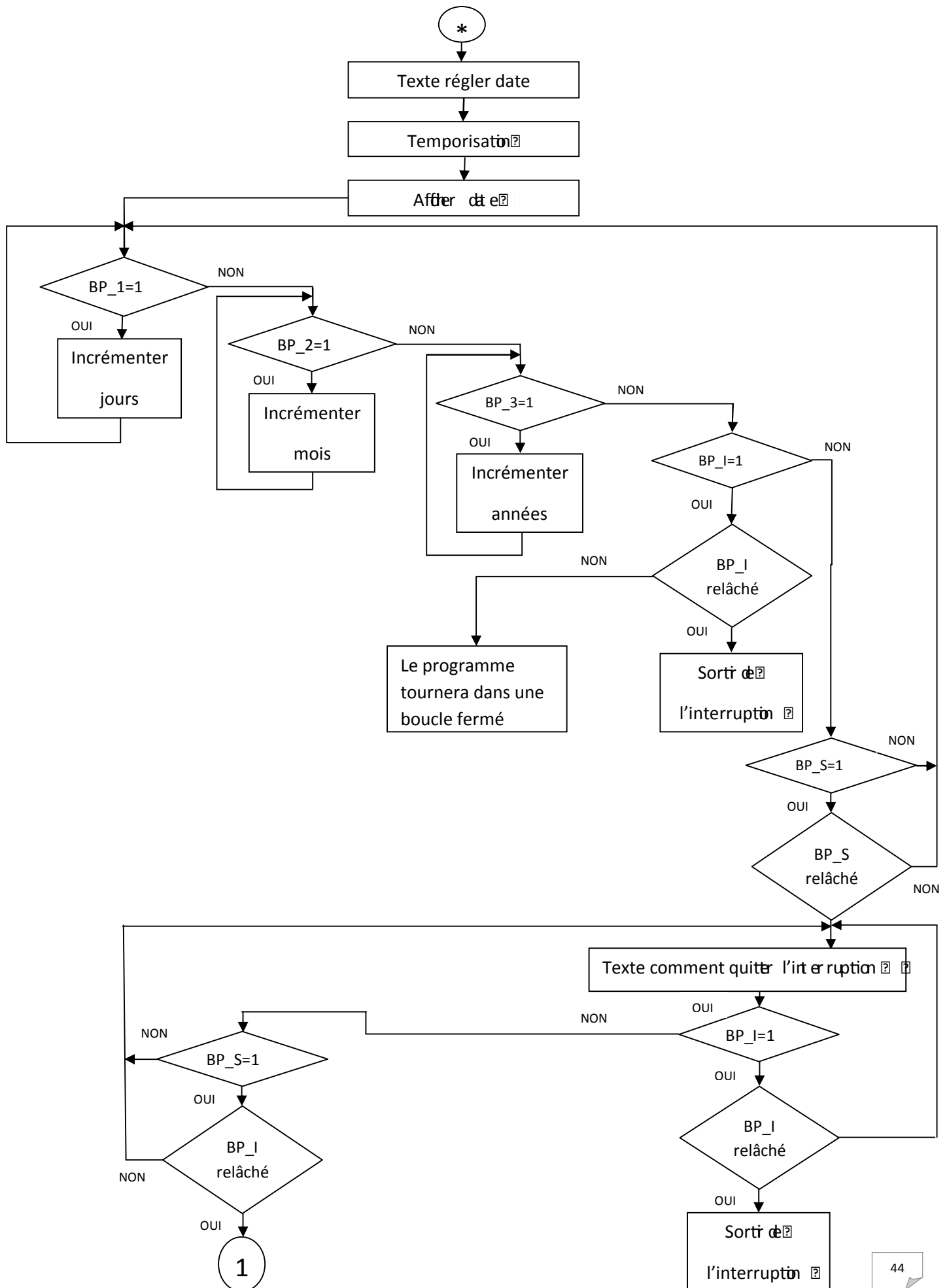
Deux boutons poussoirs nous permettront de provoquer une interruption dans le programme principale du PIC :

- Le premier nommé BP_I nous permettra d'accéder au programme de réglage de l'heure et de la date.
- Le second nommé BP_S nous permettra d'accéder au programme de la lecture à partir de la mémoire.

Les deux interruptions surviendront au moment du relâchement de chaque bouton poussoir, c'est-à-dire que l'interruption survient avec des fronts descendants.

2.4) Organigramme du réglage de l'heure et de la date :





- **Explication de l'organigramme du réglage de l'heure :**

Les boutons poussoir BP_I et BP_S travaillent avec les fronts descendants, c'est-à-dire que la requête ne sera prise en considération par le microcontrôleur qu'une fois le bouton poussoir relâché. Quand aux boutons poussoirs (BP_1, BP_2 et BP_3) travaillent avec les fronts montant.

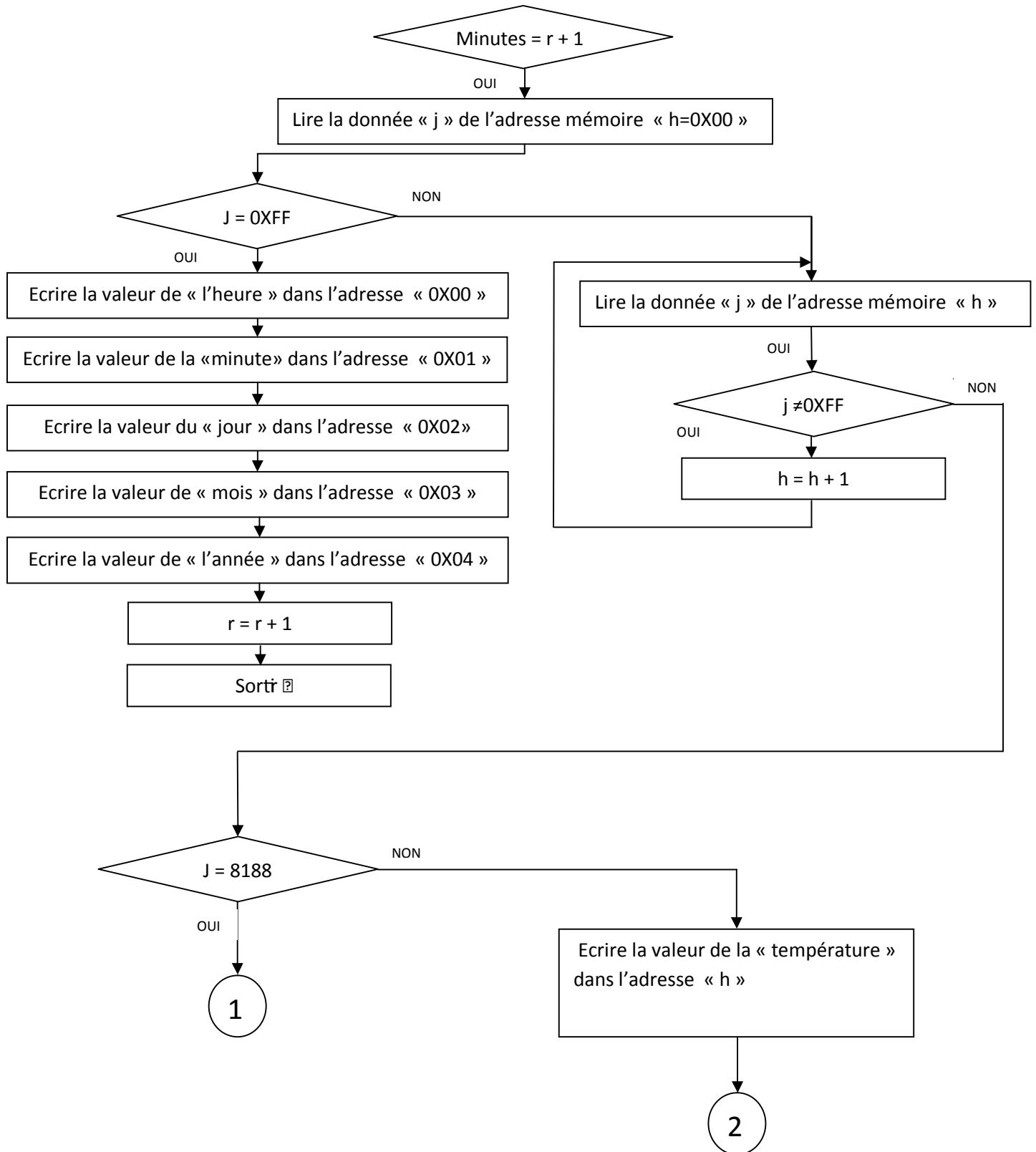
Le bouton poussoir BP_I nous permet de provoquer l'interruption du programme principale pour accéder au programme du réglage de l'heure. Une fois à l'intérieur du programme réglage de l'heure, le fait d'appuyer sur ce bouton (BP_I) nous permet de quitter l'interruption et de retourner au programme principal.

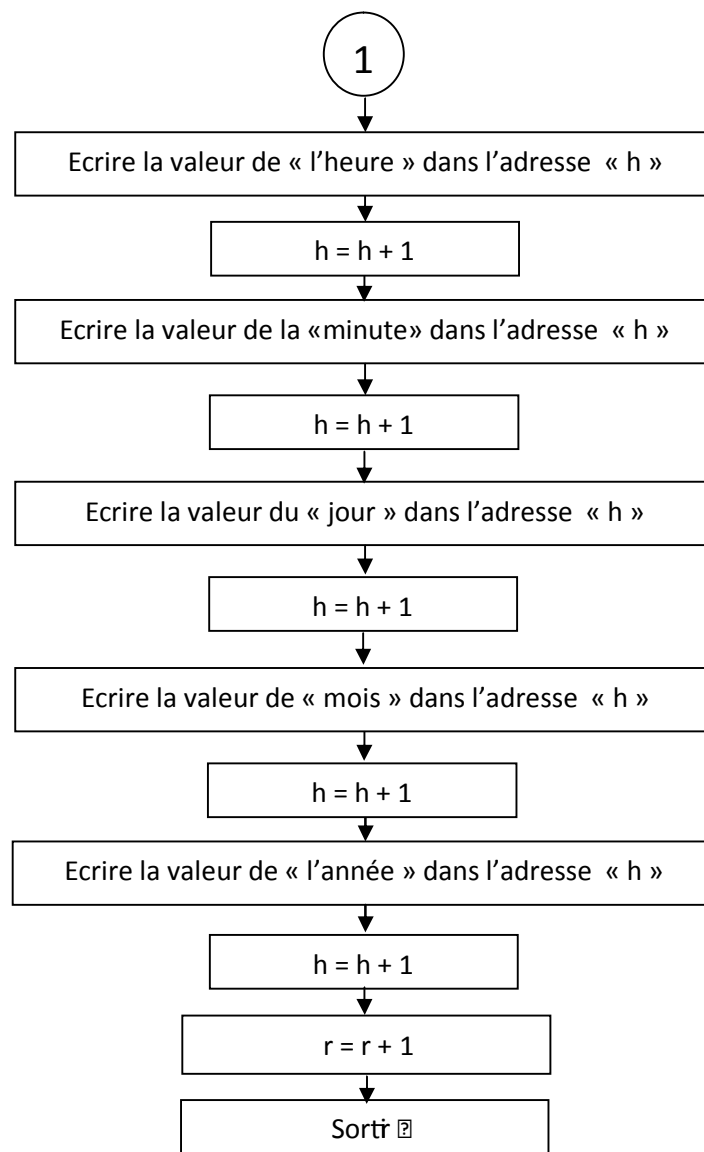
Le bouton poussoir BP_S nous permet de sélectionner soit le réglage de l'heure ou de la date.

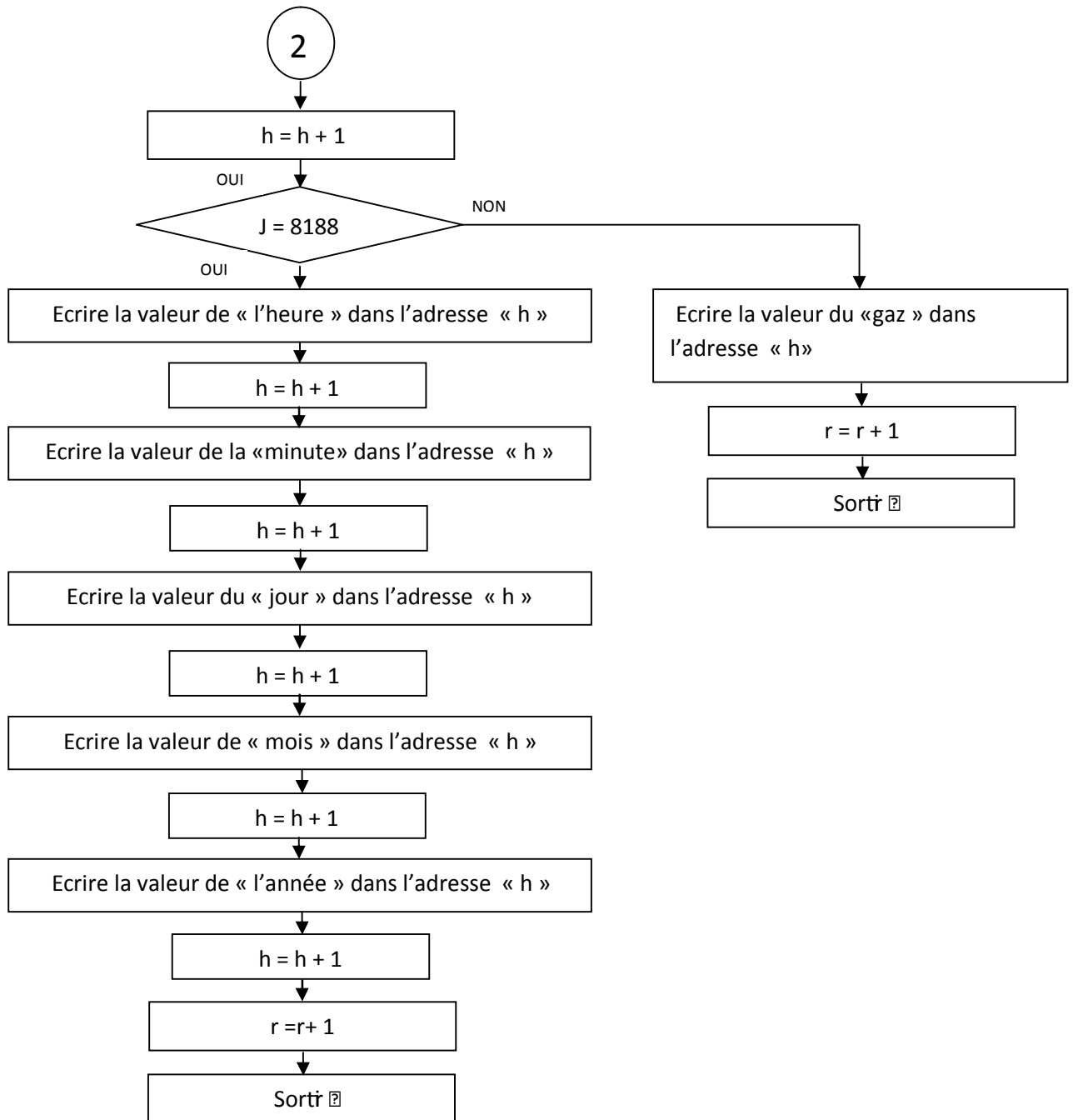
Les boutons poussoir BP_1, BP_2 et BP_3 quand à eux, ils nous permettent d'incrémenter respectivement : « heures », « minutes », « secondes », si le bouton poussoir BP_S sélectionne le réglage de l'heure. Sinon incrémenter « jours », « mois », « années » si le bouton poussoir BP_S sélectionne le réglage de la date.

Dans notre programme du réglage de l'heure, on a introduit des textes qui s'affichent sur notre afficheur LCD, ils nous indiquent à chaque fois qu'un bouton poussoir est sélectionné, l'étape suivante. Comme indiqué sur le « schéma d'affichage des étapes du réglage de l'heure et de la date » (à la page 56).

2.5) Organigramme de l'écriture dans l'EEPROM externe :







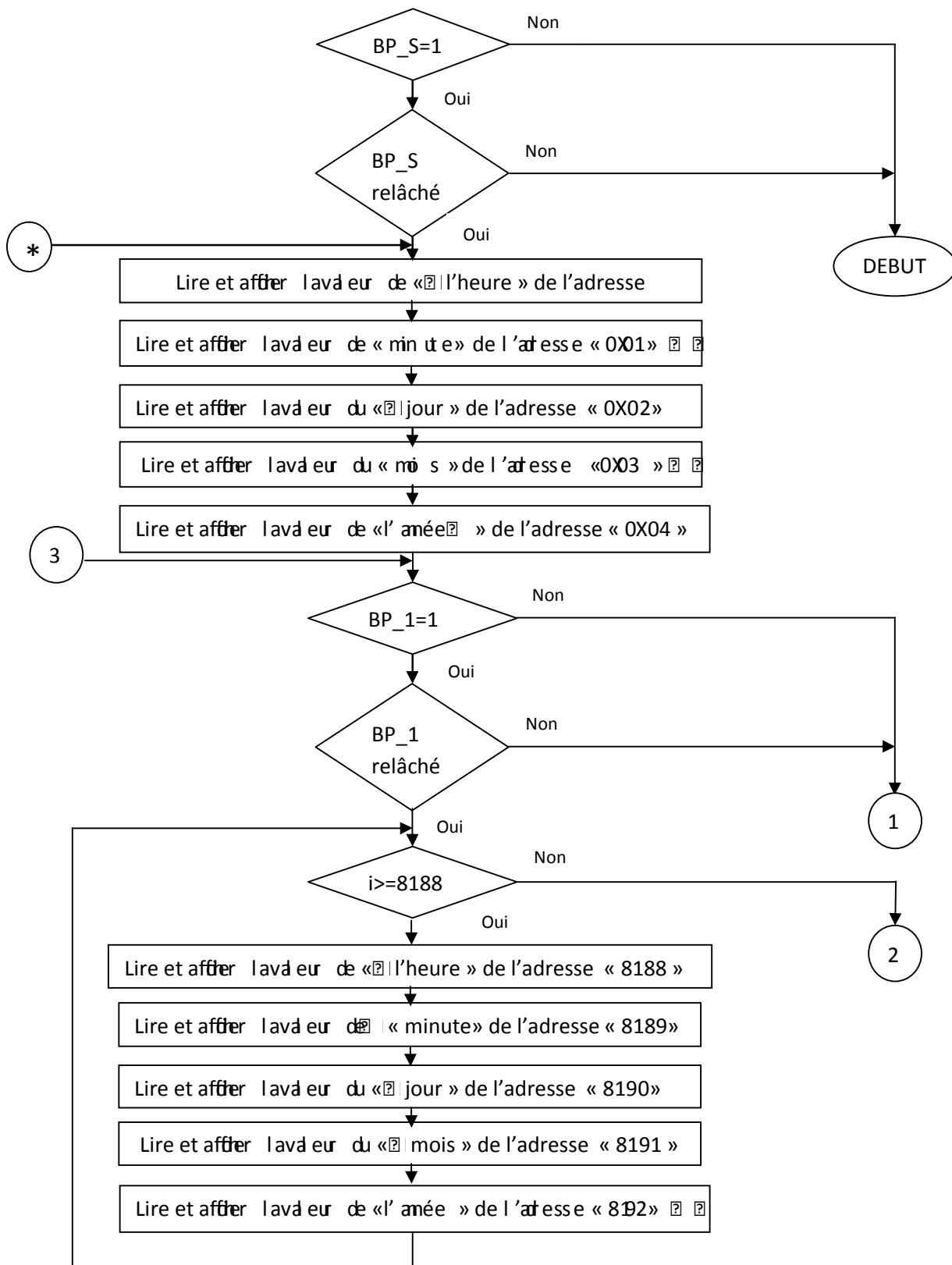
- **Explication du programme de l'écriture dans l'EEPROM externe :**

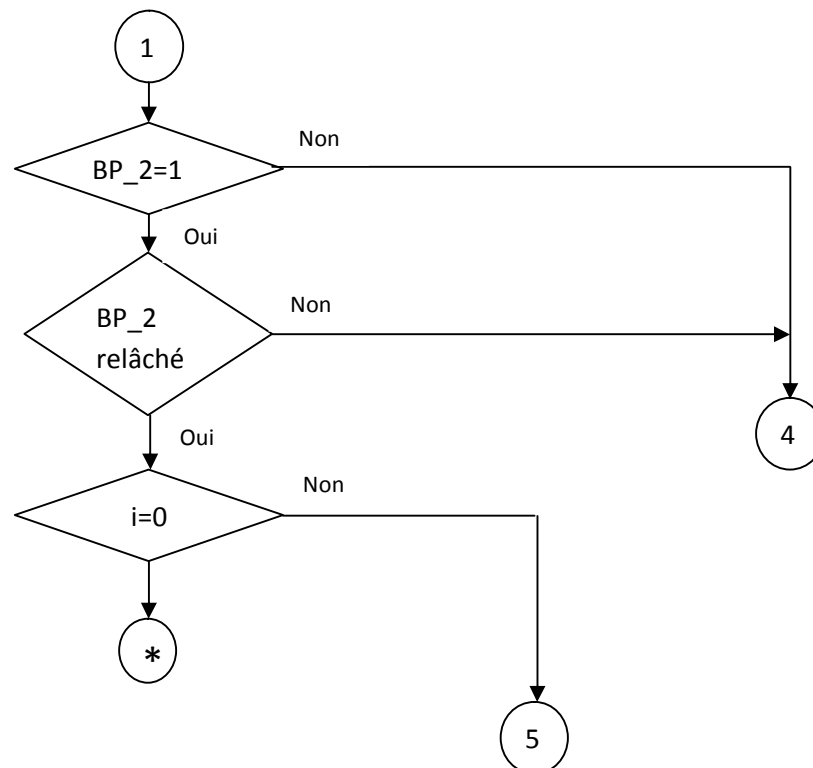
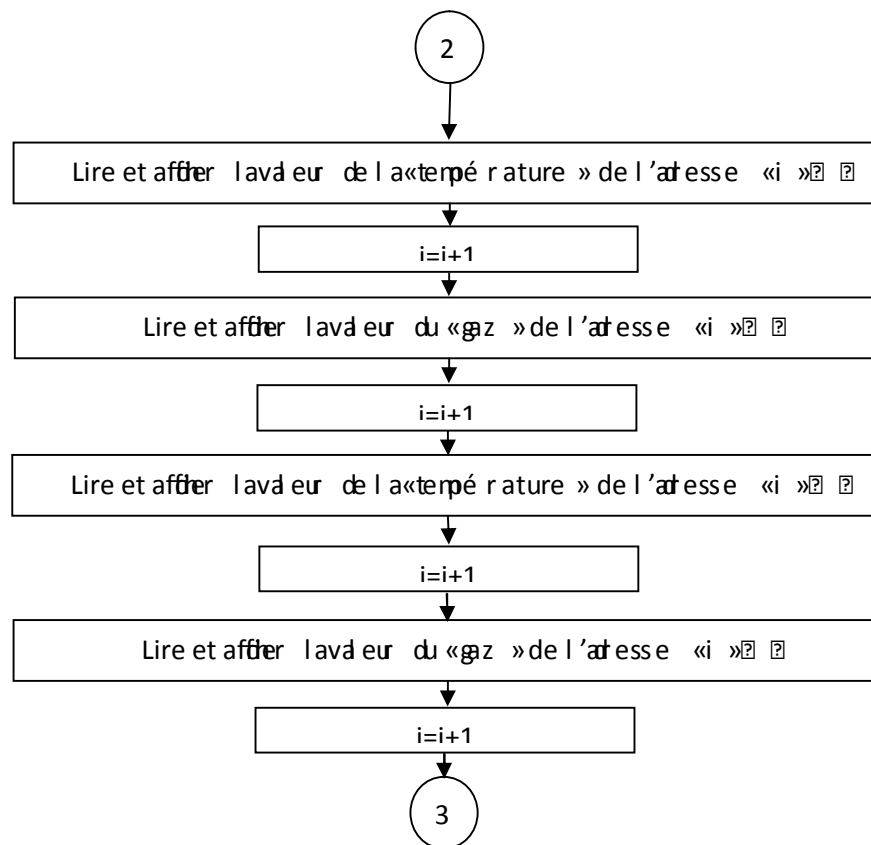
A chaque incrémentation de minute, les données acquises par le microcontrôleur de la carte de réception et affichées sur l'afficheur LCD, seront enregistrées dans l'EEPROM externe comme suite :

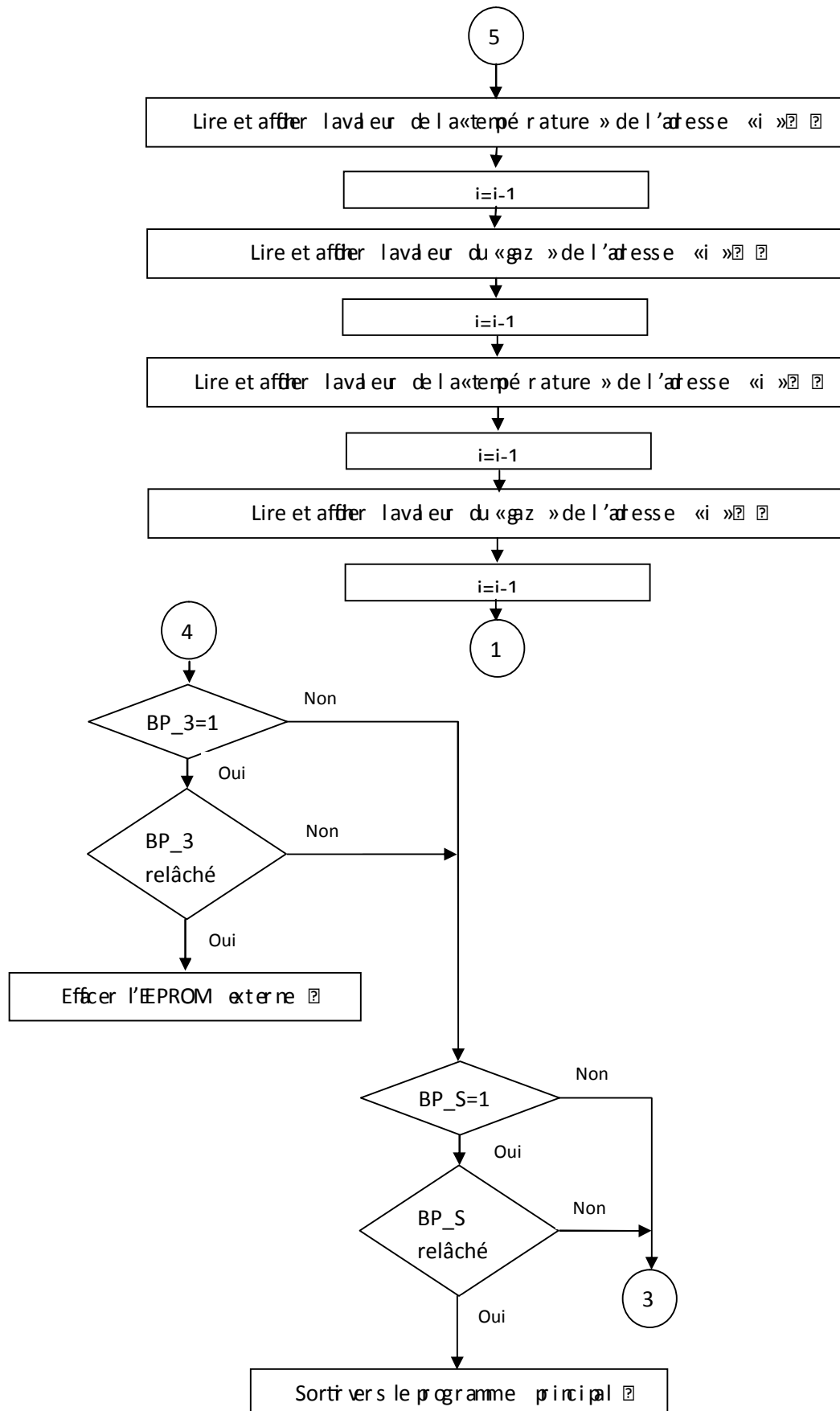
- Lors de la première incrémentation : la température, la quantité de gaz (CO), l'heure et la date seront enregistrées dans les quatre premières adresses de l'EEPROM externe.
- A partir de la deuxième minute et à chaque incrémentation de minute, la température et la quantité de gaz (CO) seront enregistrées dans deux adresses mémoire.
- A l'adresse 8188 (sachant que notre EEPROM externe possède 8192 adresses mémoires), on enregistre les quatre données suivantes : la température, la quantité de gaz (CO), l'heure et la date.
- Une fois l'EEPROM externe pleine, un message d'avertissement sera affiché sur l'afficheur LCD.

Pour ne pas écraser les données déjà enregistrées dans l'EEPROM externe, à chaque incrémentation on vérifie les adresses à partir de la première, si cette dernière est vierge on enregistre les données, sinon on passe à l'adresse suivante.

2.6) Organigramme de la lecture à partir de l'EEPROM externe :







- **Lecture à partir de l'EEPROM externe :**

Tous les boutons poussoir (BP_S, BP_1, BP_2 et BP_3) travaillent avec les fronts descendants.

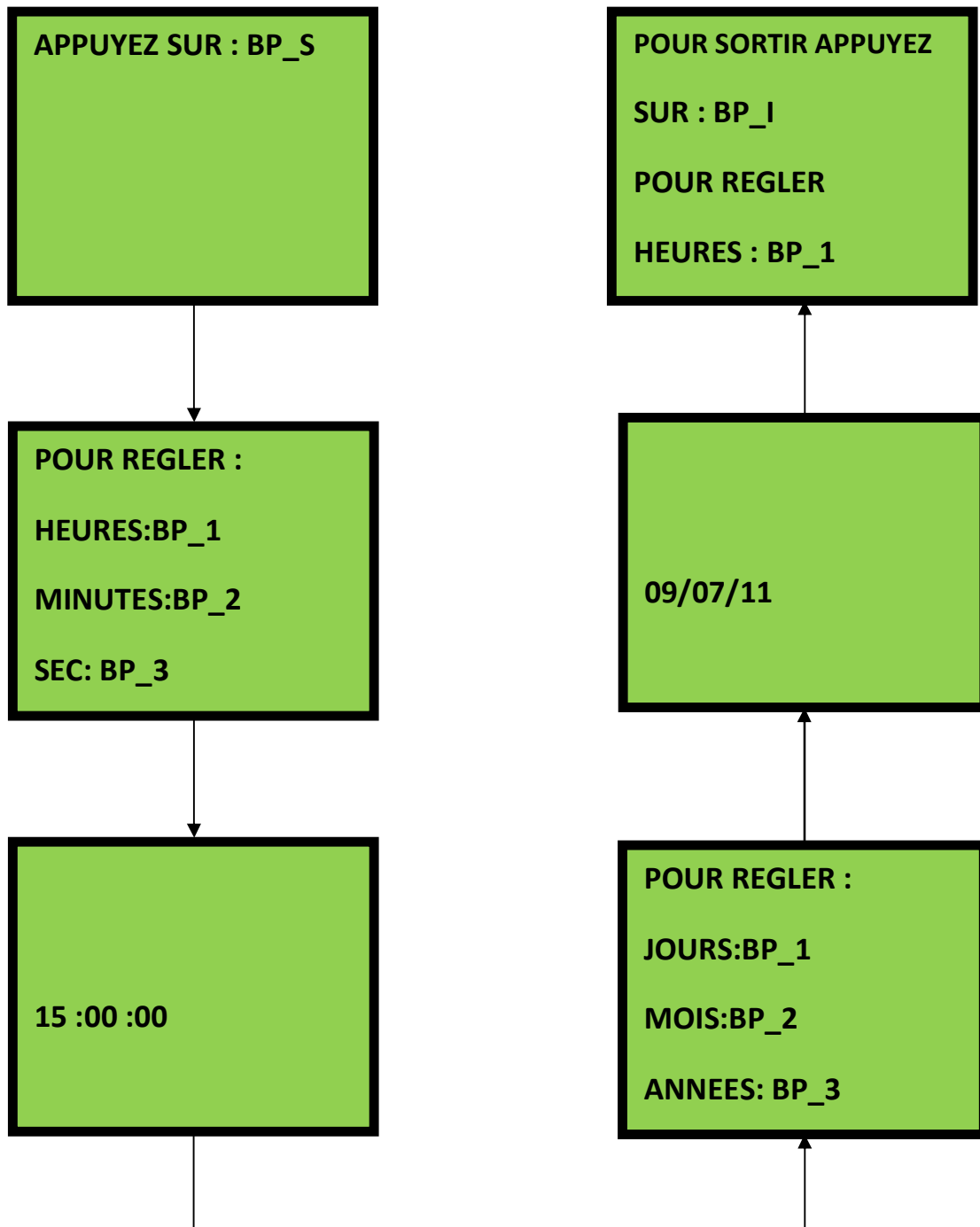
Le bouton poussoir BP_S nous permet de provoquer l'interruption du programme principale pour accéder au programme de la lecture de l'EEPROM externe. Une fois à l'intérieur du programme de la lecture de l'EEPROM externe, le fait d'appuyer sur ce bouton (BP_S) nous permet de quitter l'interruption et de retourner au programme principal.

Le bouton poussoir BP_1 nous permet de lire les données enregistrées dans l'EEPROM externe et de les afficher sur l'afficheur LCD quatre par quatre dans l'ordre croissant des adresses.

Le bouton poussoir BP_2 nous permet de lire les données enregistrées dans l'EEPROM externe et de les afficher sur l'afficheur LCD quatre par quatre dans l'ordre décroissant des adresses.

Le bouton poussoir BP_S nous permet d'effacer l'EEPROM externe.

3) Schéma d'affichage des étapes du réglage de l'heure et de la date :



4) Ecriture et transfert du programme dans le PIC :

L'écriture du programme pour le PIC se fait dans le compilateur CCS avec le langage C, nous l'écrivons d'abord en langage évolué, après la compilation, un fichier hexadécimal sera créé automatiquement et enfin on le transfère vers la mémoire du PIC en utilisant le logiciel WinPic800.

4.1) Utilisation du logiciel CCS :

Nous utilisons l'éditeur de texte du logiciel CCS pour l'écriture du code source du PIC. Ce logiciel, contient un fichier intégré d'aide à la mise en route (un Wizard) bien pratique. La figure 40 donne le premier écran qu'on obtient après le lancement du logiciel.

L'utilisation du logiciel CCS est très simple, il suffit de suivre les étapes comme indiqué sur les figures suivantes. Pour la création de notre programme on a procédé comme suite :

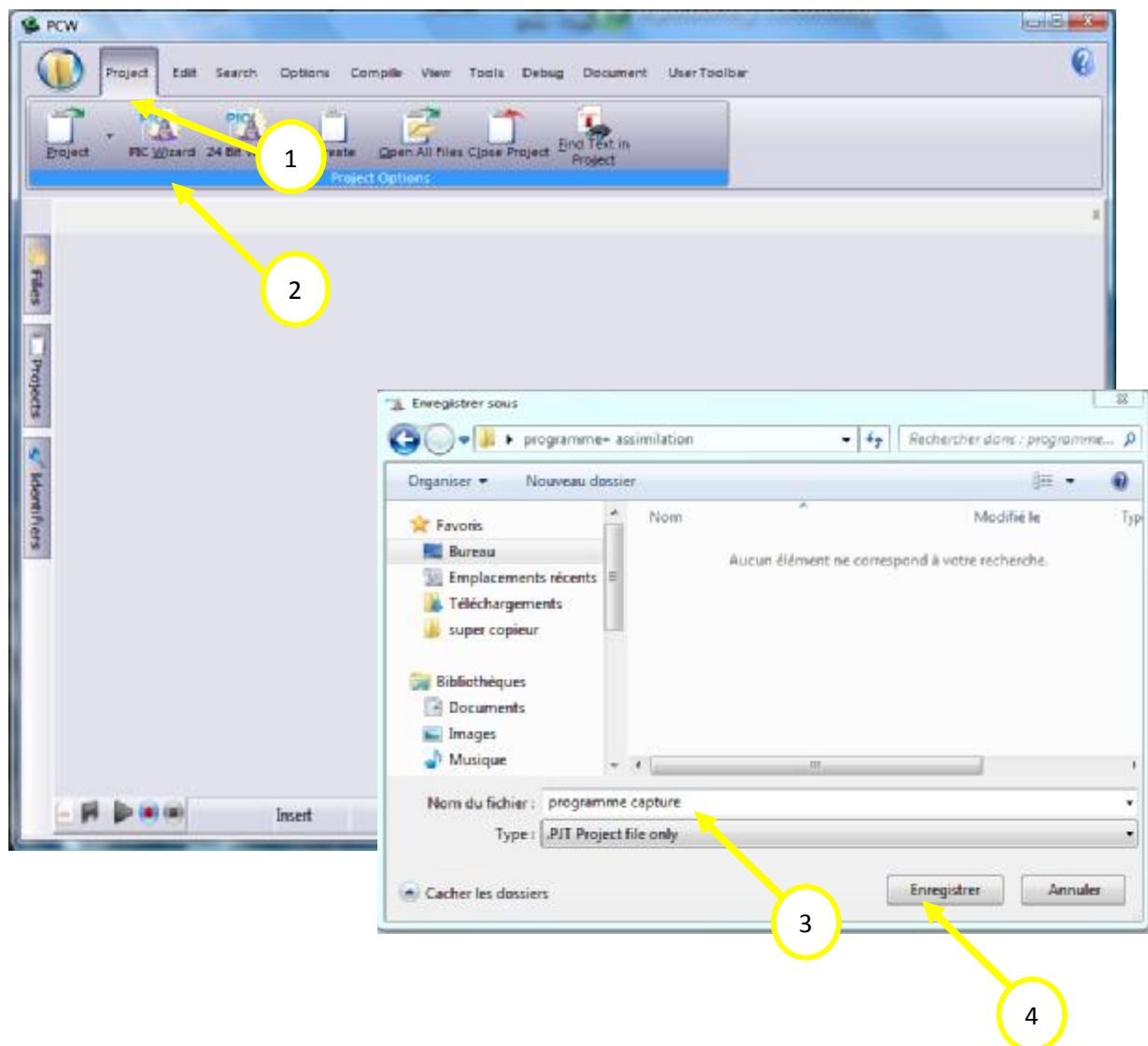


Figure 40: Création et enregistrement du programme.

On a commencé par la création d'un dossier vide dans lequel on a enregistré notre programme, en suite on a lancé le logiciel, et on a choisi l'interface « Projet », en appuyant sur « PIC Wizard » une fenêtre apparaît, à partir de cette dernière on a ouvert le dossier vide qu'on a créé au par avant, on a nommé le fichier de notre programme, et on l'a enregistré.

Une fois qu'on a appuyé sur « Enregistrer » la fenêtre suivante s'ouvre :

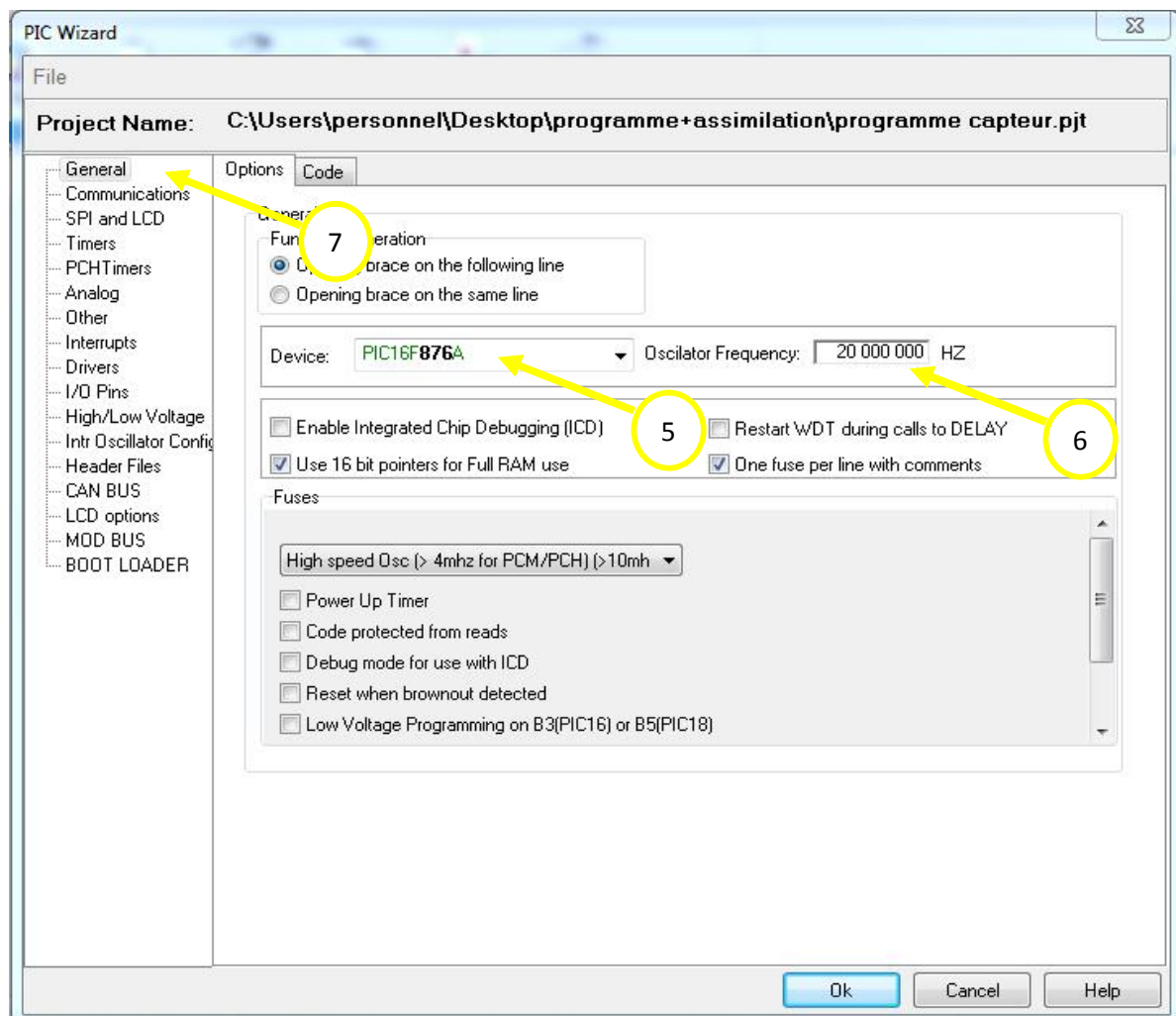


Figure 41 : Configuration du PIC par logiciel (étape 1).

Cette fenêtre nous permet de configurer le PIC par le logiciel. Elle est constituée de deux parties : l'une (à gauche) pour choisir le bloc à configuration du PIC, et l'autre (à droite) nous permet de choisir les configurations à faire ;

Dans notre cas pour le choix du PIC on a pris le PIC 16F876A, avec une fréquence de l'oscillateur de 20MHZ.

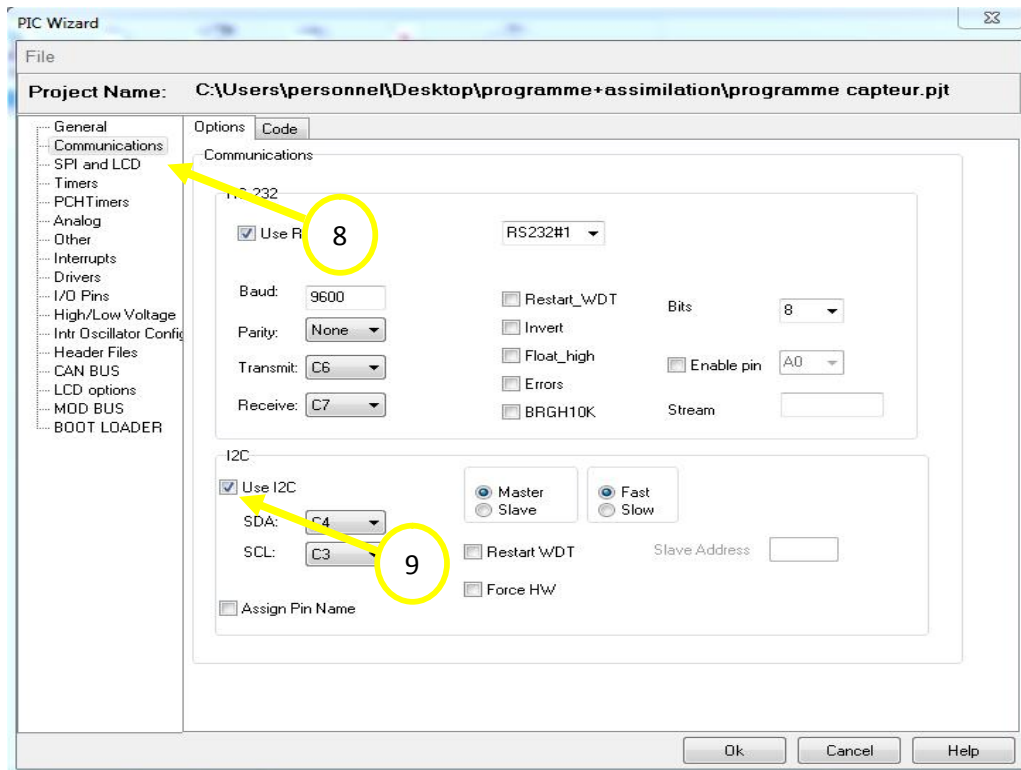


Figure 42 : Configuration du PIC par logiciel (étape 2).

Dans le bloc Communication on a coché les deux modes de transmission « RS232 » et « I2C » tout en gardant leur configuration par défaut.

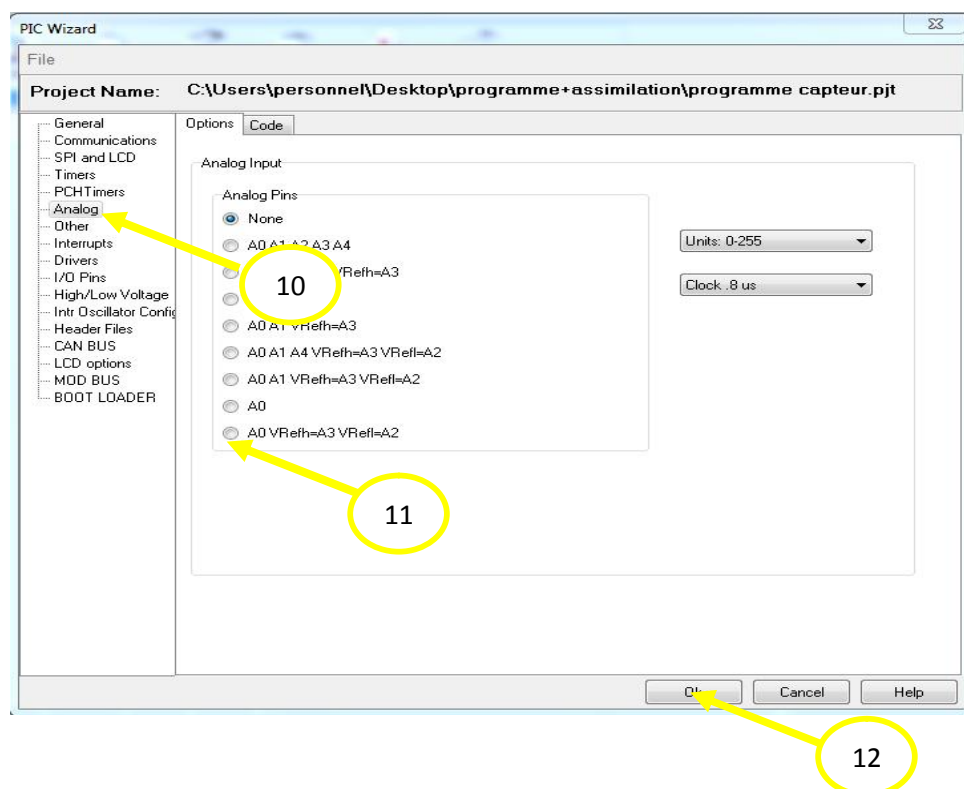


Figure 43 : Configuration du PIC par logiciel (étape 3).

Dans le bloc « Analog » on choisit la pin A0 comme étant l'entrée analogique, quand à A3 et A2 comme étant des entrées de référence.

En appuyant sur « OK » la fenêtre suivante apparaît :

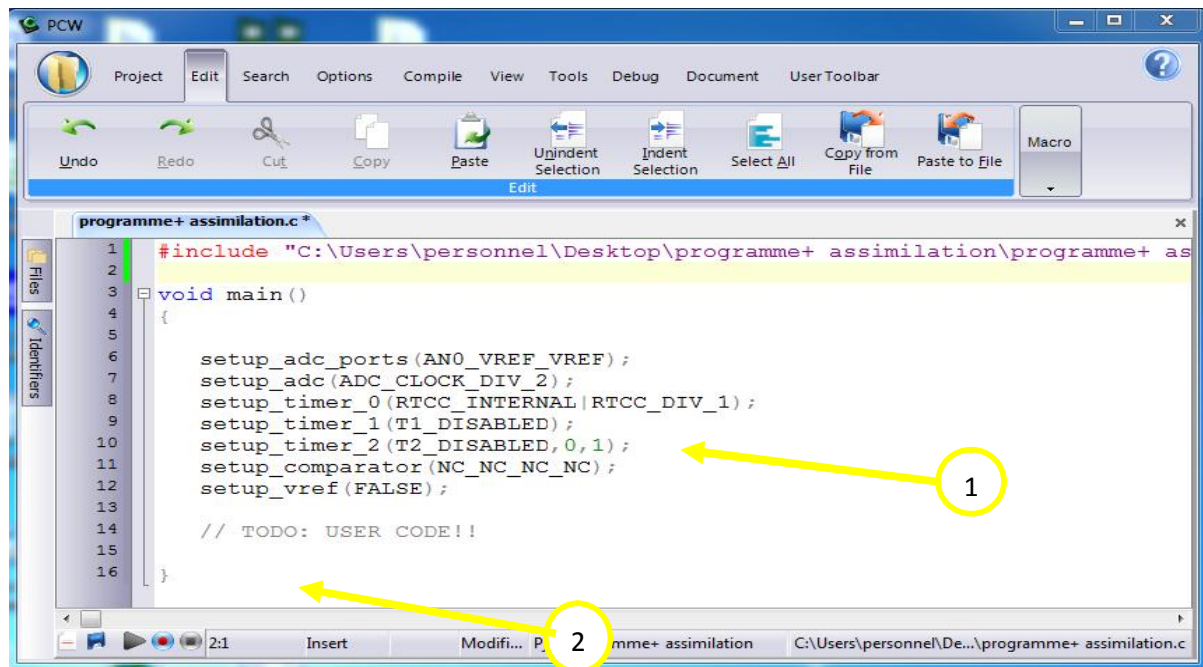


Figure 44 : fenêtre pour écrire le programme principal.

Cette fenêtre comporte les différentes configurations qu'on a faites précédemment dans le logiciel, et notre programme en langage avancé.

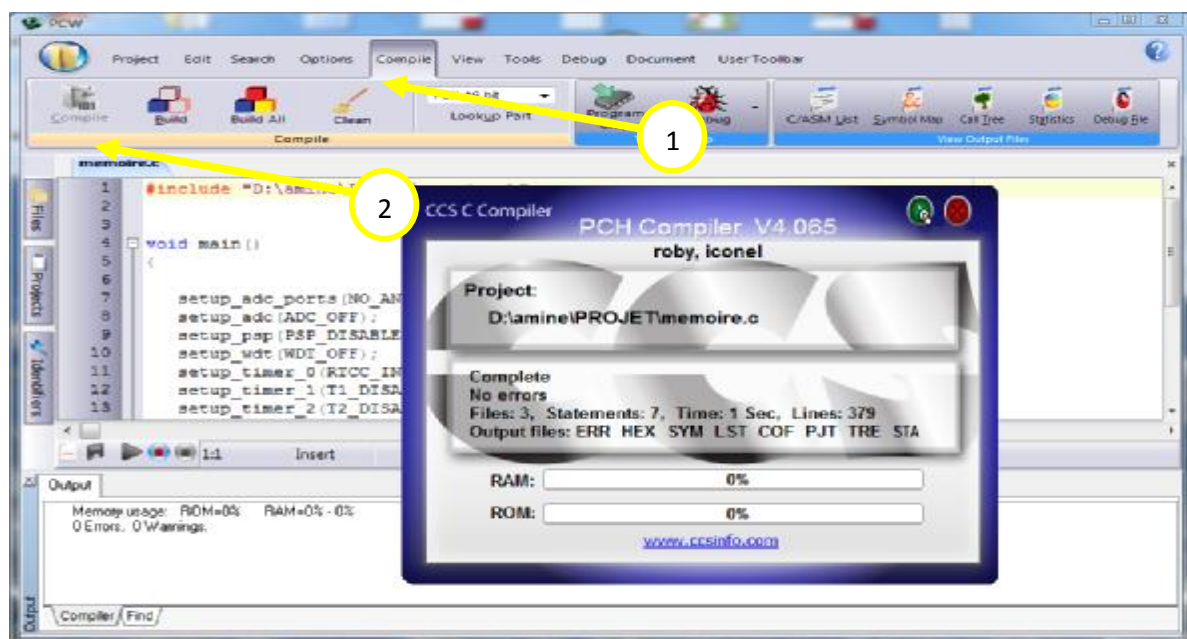


Figure 45 : Compilation du programme.

La dernière fenêtre nous indique les méthodes à suivre pour la compilation de notre programme et la création du fichier Hexadécimal.

4.2) WinPic800_3.55G :

Pour le flashage du PIC on suit les étapes suivantes :

- ü On place le PIC 16F876A sur le support du programmeur universel.
- ü On branche le programmeur à l'unité centrale de micro-ordinateur et on le met sous tension.
- ü On lance le logiciel WinPic800_3.55G.

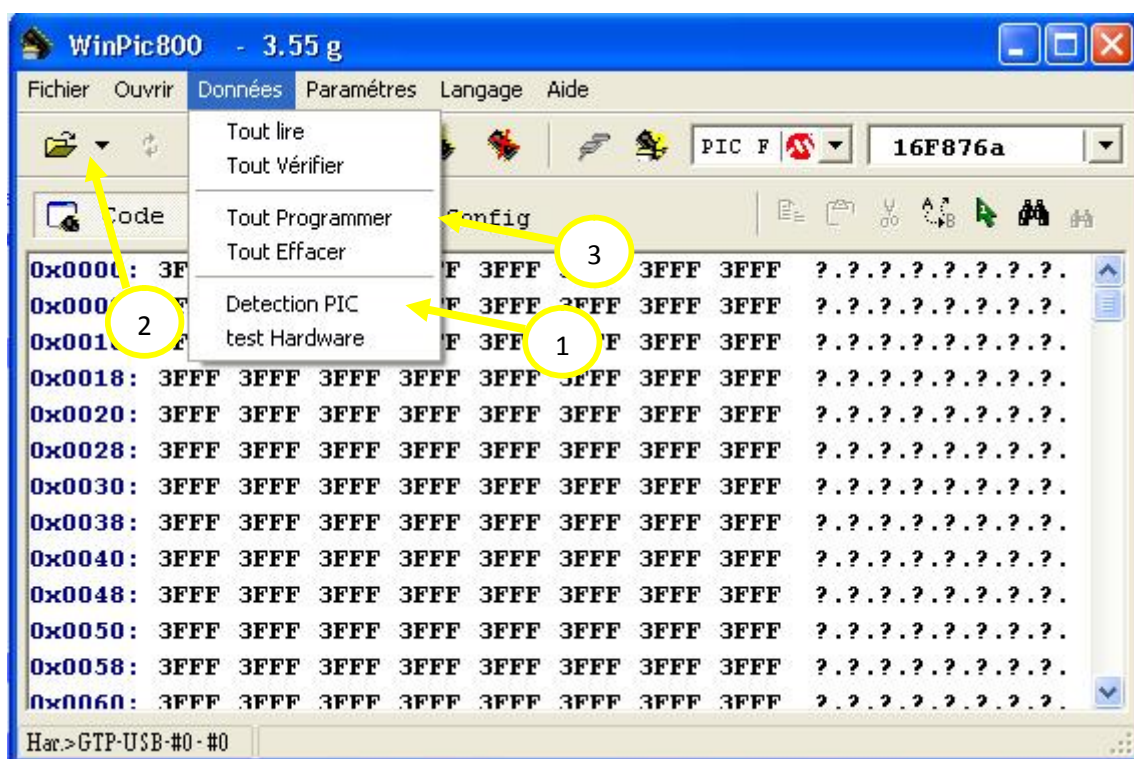


Figure 46: Programmation du PIC sous WinPic800_3.55G

Les étapes à suivre pour charger le fichier Hexadécimal dans le PIC sont comme indiqué sur la figure :

1. On commence par détecter le PIC.
2. On charge le fichier hexadécimal à partir du dossier qui contient notre programme.
3. On appui sur « Tout programmer » pour charger le programme dans le PIC
4. Une fois le programme chargé, on appui sur « Accepter », comme indiqué sur la figure si dessous.

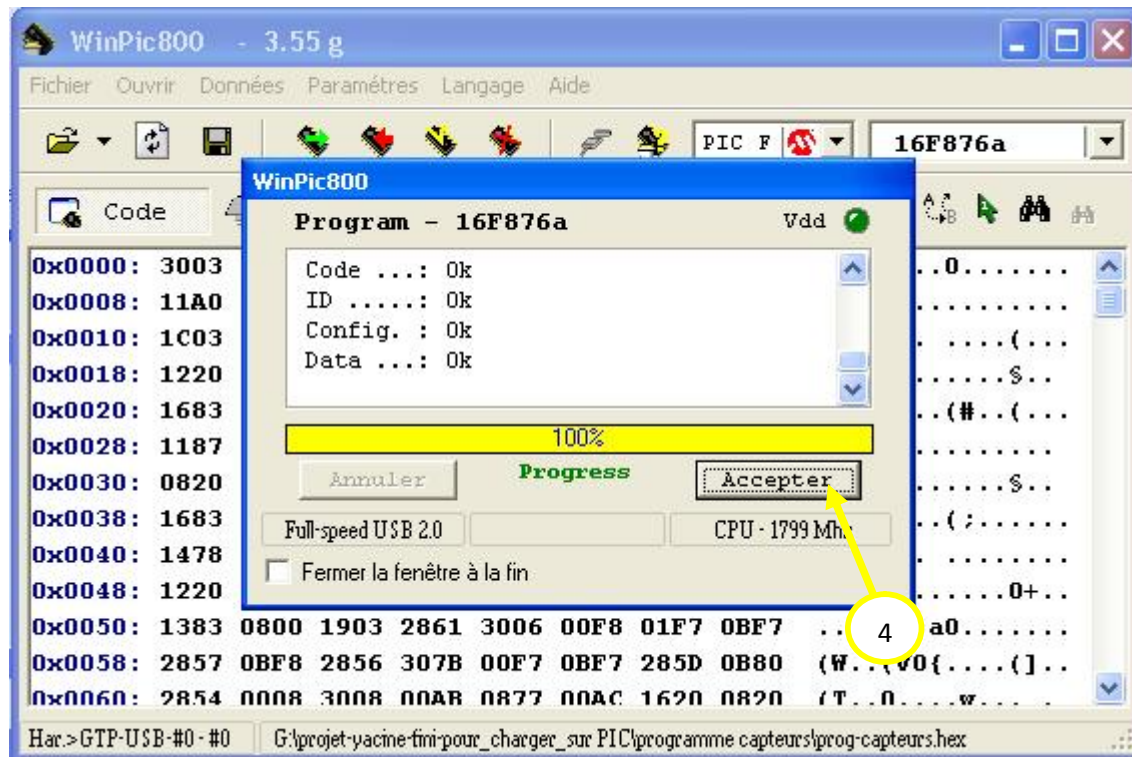
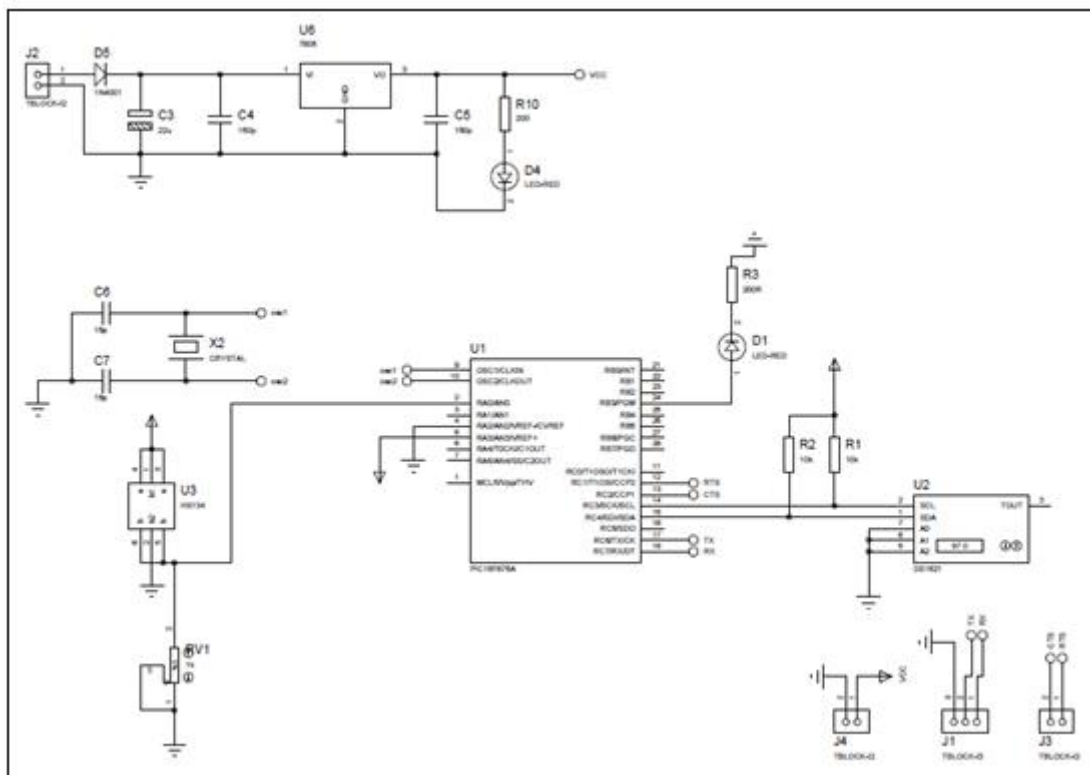
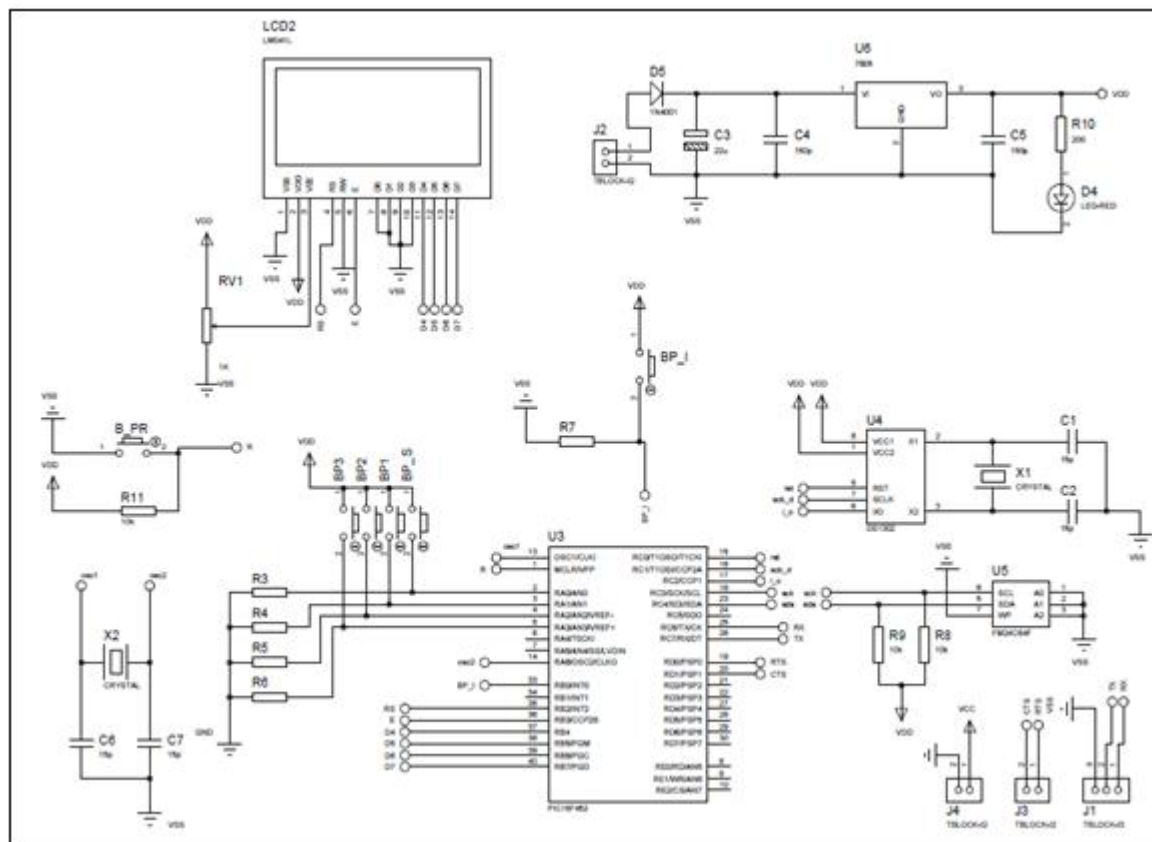


Figure 47 : Programmation du PIC sous WinPic800_3.55G

5) Schémas électrique de la carte d'émission :



6) Schémas électrique de la carte de réception :



7) Réalisation pratique :

Dans la réalisation pratique on est passé par deux étapes essentielles :

- Ø Réalisation des circuits imprimés.
- Ø L'implantation des composants.

7.1) Les circuits imprimés des deux cartes:

Les circuits imprimés représentés par les figures 48 et 51 sont réalisés par le logiciel de routage et de simulation électronique « PROTEUS 7.6 ».

7.1.1) L'insolation :

C'est une phase importante du processus de la réalisation d'un circuit imprimé. Sa réussite nécessite la vérification des points suivants :

- Ø Le bon plaquage du typon sur la face sensible de la plaque.
- Ø Le sens du typon (Face Cuivrée ou non).
- Ø La durée de l'insolation.

7.1.2) La gravure :

Cette étape consiste à plonger la plaque obtenue après insolation dans le révélateur pour éliminer la résine brûlée par les UV et après lavage de la plaque avec de l'eau, on la plonge dans un bain de perchlorure de fer afin d'éliminer le cuivre non protégé par la résine. Après avoir terminé on la rince abondamment avec de l'eau.

7.1.3) Le perçage :

On fixe la plaque sur une planche en bois afin d'éviter tous mouvements de cette dernière pendant le perçage, et pour chaque diamètre des trous on choisit le foret qui convient.

7.2) Implantation des composants :

Cette partie représente le circuit imprimé, et l'emplacement des composants.

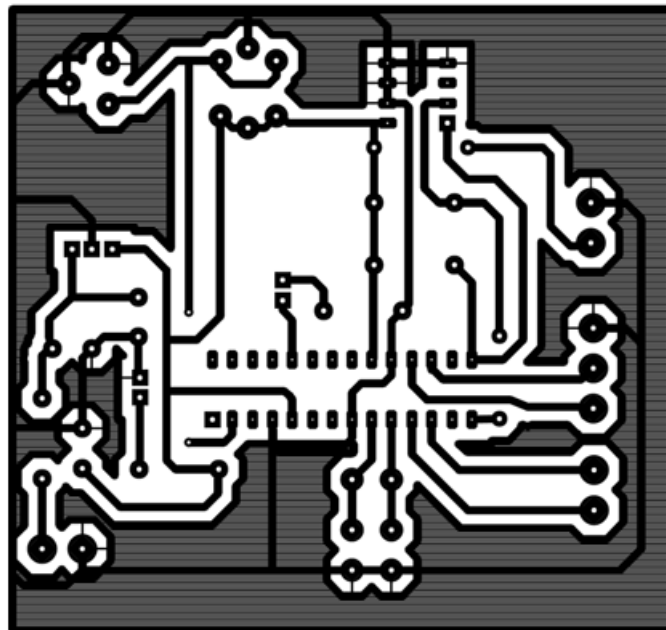
7.2.1) Implantation des composants de la carte d'émission :

Figure 48 : Le circuit imprimé de la carte d'émission coté cuivre.

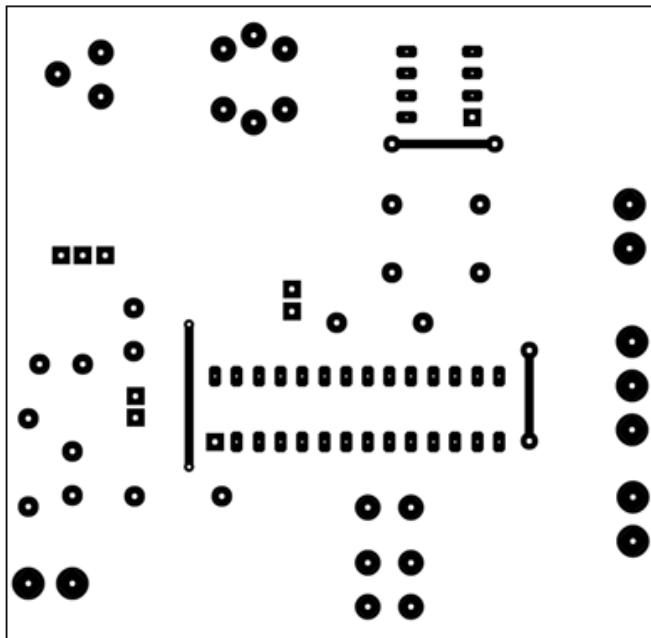


Figure 49 : le circuit imprimé de la carte d'émission coté composant.

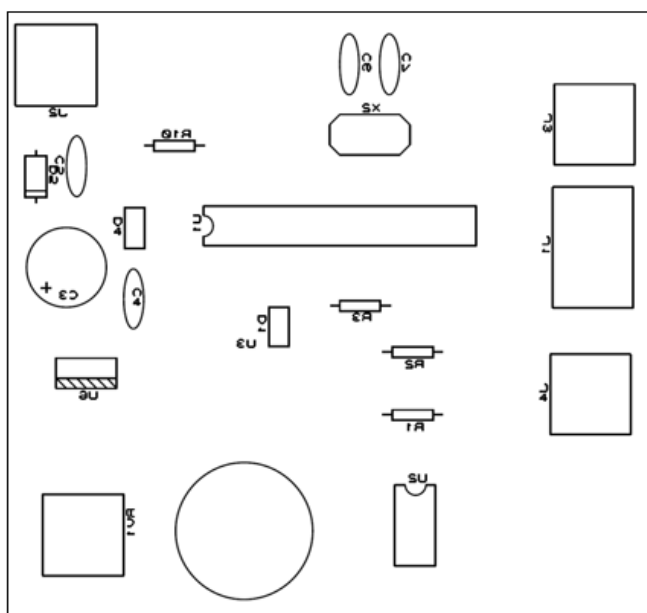


Figure 50: Implémentation des composants (carte d'émission).

7.2.2) Implantation des composants de la carte de réception :

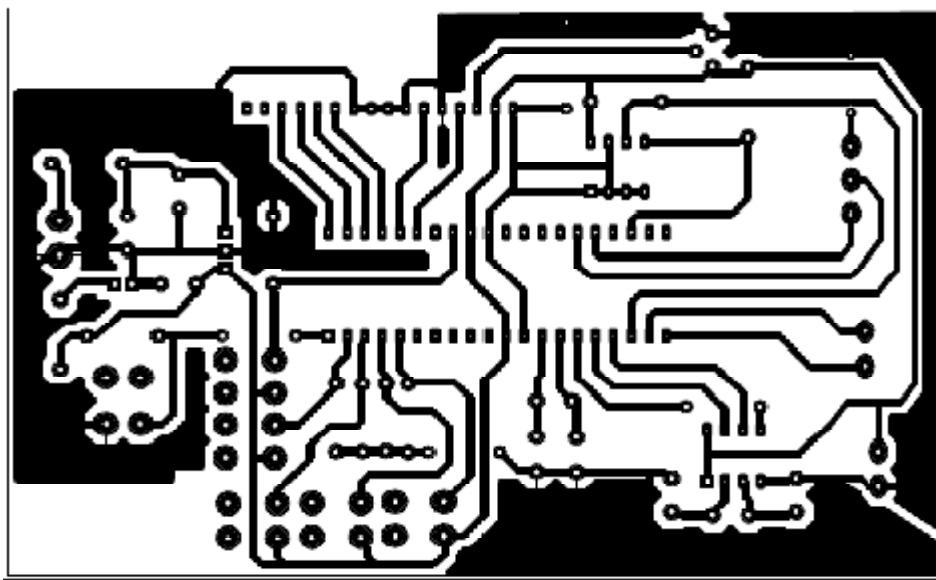


Figure 51 : Le circuit imprimé de la carte de réception coté cuivre.

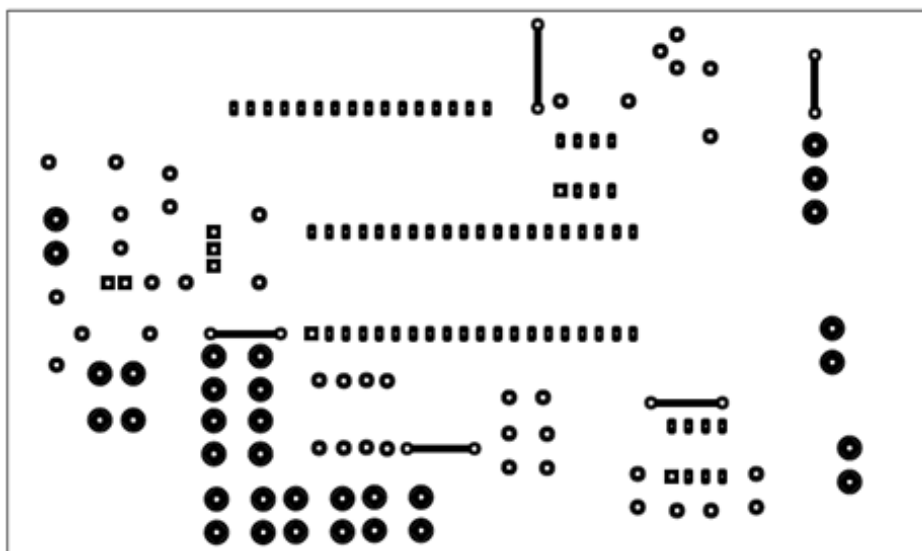


Figure 52: le circuit imprimé de la carte de réception coté composant.

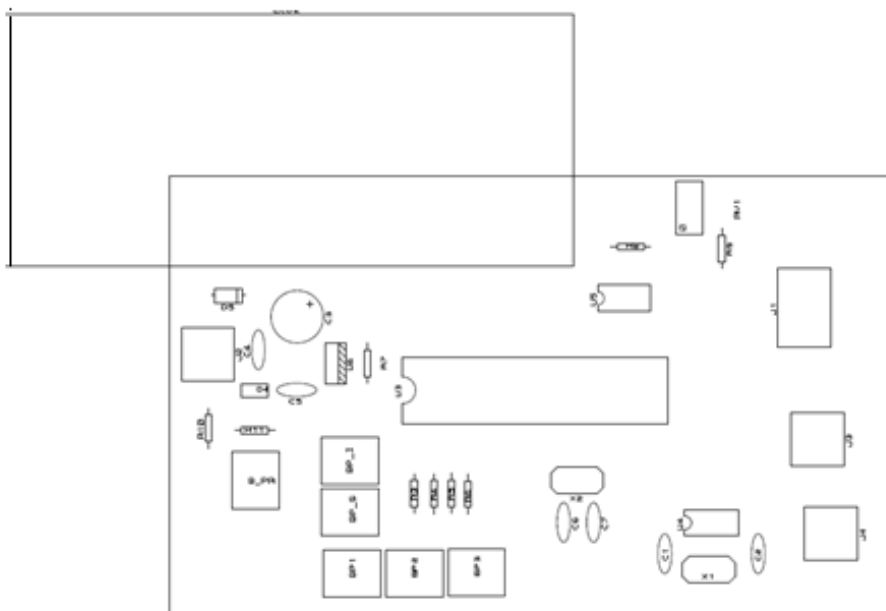


Figure 53: Implémentation des composants (carte de réception).

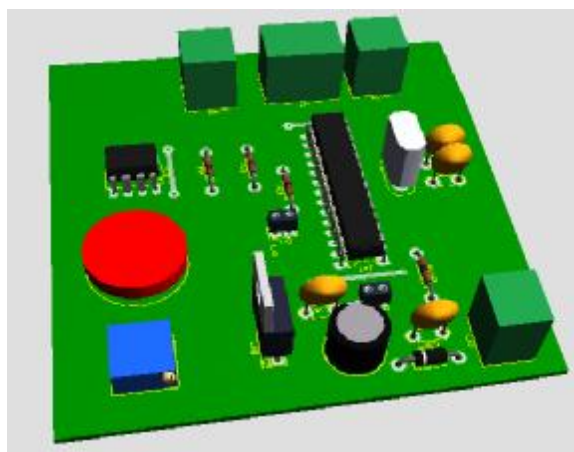


Figure 54 : carte d'emission en 3D

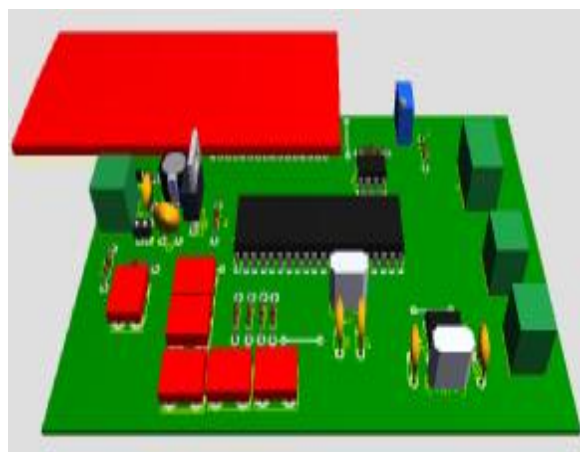


Figure 55: carte de reception en 3D

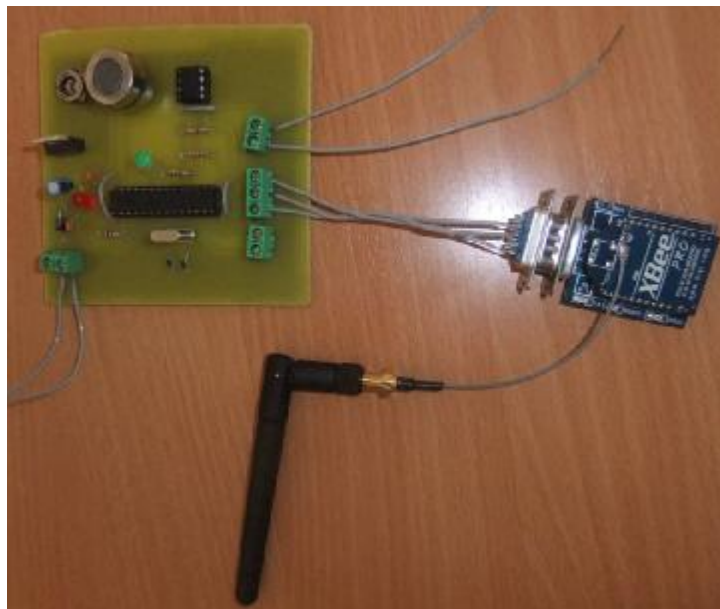


Figure 56 : Carte d'émission après réalisation



Figure 57 : Carte de réception après réalisation

8) Nomenclature des composants :**8.1) Liste des composants utilisés dans la carte d'émission :****Résistance**

- 2 Résistances (R1, R2) de 10K.
- 1 Résistance R3 de 200 Ω .
- 1 Résistance R10 de 200 Ω .

Capacité

- 1 capacité C3 de 22u.
- 2 capacités (C4, C5) de 150p.
- 2 capacités (C6, C7) de 15p.

Circuit intégrés

- 1 microcontrôleur 16F876A.
- 1 capteur de température DS1621.
- 1 capteur de gaz CO HS134.
- 1 régulateur de tension 7805.

Diodes

- 2 Led (D1, D4) rouge.
- 1 diode (D5) 1N4001.

Autre composant

- 3 bornes à vices (J1, J2, J4).
- 1 potentiomètre (RV1) 1K.
- 1 quartz (X1) 20 MHZ.

8.2) Liste des composants utilisés dans la carte de réception :**Résistance**

- 8 Résistances (R3- R9, R11) de 10k.
- 1 Résistance (R10) de 200 Ω .

Capacité

- 4 capacités (C1, C2, C6, C7) de 15p.
- 1 capacité de C3 de 22u.

- 2 capacités (C4, C5) de 150p.

Circuit intégrés

- 1 microcontrôleur 18F452.
- 1 dateur DS1302.
- 1EEPROM 24C64A.
- 1 régulateur de tension 7805.

Diodes

- 1 Led (D4) rouge.
- 1 diode (D5) 1N4001.

Autre composants

- 6 boutons poussoir (BP_1, BP_2, BP_3, BP_I, BP_S, BP_R).
- 3 bornes à vices (J1, J2, J4).
- 1 afficheur LCD (4x16).
- 1 résistance variable (RV1) de 1K.
- 1 quartz X1 de 32 kHz.
- 1 quartz X2 de 20 MHz.

Conclusion générale

Conclusion générale:

Au terme de ce travail, nous avons pu réaliser deux cartes, la première est une carte de mesure et d'acquisition de gaz et de température, ces valeurs seront transmises à la seconde soit par liaison filaire (un câble avec deux ports DB-9), soit avec une liaison Radio Fréquence via le module de transmission XBee Pro, Cette carte est chargée de l'acquisition de ces deux données et les afficher sur un afficheur LCD, tout en les sauvegardant dans une mémoire externe.

En plus des difficultés qu'on a rencontrés, en particulier avec la transmission via le module XBee Pro, nous avons acquis une expérience pratique non négligeable dans le domaine des circuits électroniques programmables et on s'est familiarisé avec des logiciels de simulation et de conception (Proteus Professionnel 7.6, CCS PIC C Compiler, Mplab et Winpic800) et surtout avec la programmation des microcontrôleurs.

Comme toute conception, il reste toujours des améliorations à lui apporter telle qu'une transmission half duplex via le module XBee Pro, soit entre deux microcontrôleurs, ou bien un microcontrôleur et un PC, et la sécurisation de la liaison entre ces deux derniers (reconnaissance du destinataire avant chaque transmission). En plus, l'utilisation du port USB pour l'acquisition des données par le PIC apportera une grande maniabilité, et une adaptabilité des différentes réalisations aux nouvelles technologies.

On espère bien que notre travail sera repris par les générations à venir, et qu'il constituera un point de départ pour des travaux beaucoup plus élaborés, et qu'il leur apportera la base dont ils auront besoin pour la réalisation de leurs projets.

Programme

Ø Programme en CCS injecté dans le PIC de la carte d'émission :

• Programme principal :

```
#include "C:\Users\personnel\Desktop\dossier projet yacine avec pic 18F452\programme-  
capteurs\prog-capteurs.h"  
#include "DS1621_Lib.c"  
#include "init.c"  
  
void main()  
{  
  
    setup_adc_ports(AN0_VREF_VREF);    // sélectionné le PORTA.0 (ANALOGIQUE)  
    setup_adc(ADC_CLOCK_DIV_64);  
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);  
    setup_timer_1(T1_DISABLED);        // désactiver timer_1  
    setup_timer_2(T2_DISABLED,0,1);    // désactiver timer_2  
    setup_comparator(NC_NC_NC_NC);  
    setup_vref(FALSE);  
  
    debut:  
        output_high (pin_b2); // allumer une led qui indique l'alimentation du PIC  
        d = read_temp_DS1621(0);    // la temperature mesuré par le DS1621  
        set_adc_channel (0);  
        g=read_adc();                // gaz mesuré  
  
    //////////////////////////////////////  
  
        printf("%c",d);            // envoi de la valeur de la température par RS232  
        delay_ms(30);  
        printf("%c",g);            // envoi de la valeur du gaz par RS232  
        delay_ms(30);  
  
        goto debut;  
  
}
```

• Déclaration des variables :

```
char d=0;        // d represente la temperature mesuré par le DS1621  
char g=0;        // d represente la valeur du gaz mesuré par le HS135
```

Ø Programme en CCS injecté dans le PIC du programme de réception:

Programme principal :

```
#include "C:\Users\personnel\Desktop\dossier projet yacine avec pic 18F452\programme-
affichage\prog-affichage.h"
#include "Flex_LCD_4x16.c"
#include "2464.c"
#include "ds1302.c"
#include "init.c"
#include "fonctions.c"
//=====
=====
//+++++programme réglage heure et date+++++
//=====
=====
#int_EXT
EXT_isr()          //provoquer l'interruption en appuyant sur BP_I
{
    texte_regler_heur();
star_1:
    if(BP_S==1)      // appui sur BP_S pour selectonner reglage temps
    {
        delay_ms(30);
        re_bp_s();
        afficher_comentaire();
        delay_ms(2000);
        afficher_heur ();
        goto saut_1;
    }
    else
    {
        goto star_1;
    }

saut_1:
    if(BP_1==1)      // appui sur BP_1 pour reglage heur
    {
        inc_heur ();
        afficher_heur ();
        lcd_gotoxy(1,1);
        lcd_putc("incrémenter heur");
    }

    if(BP_2==1)      // appui sur BP_2 pour reglage minutes
    {
        inc_min ();
        afficher_heur ();
        lcd_gotoxy(1,1);
        lcd_putc("incrémenter\n minute");
    }

    if(BP_3==1)      // appui sur BP_3 pour reglage secondes
```

```

{
inc_sec ();
afficher_heur ();
lcd_gotoxy(1,1);
lcd_putc("incrémenter\n  seconde");
}
ds1302_set_datetime();
if (BP_I==1)          //appui BP_I pour quitter interruption
{
delay_ms(30);
re_BP_I();
goto sortir;
}
if(BP_S==1)          // appui sur BP_S pour sélectionner réglage date
{
delay_ms(30);
re_bp_s();
afficher_coment_date();
delay_ms(2000);
afficher_date ();
goto saut_2;
}
else
{

goto saut_1;
}
saut_2:
if(BP_1==1)          // appui sur BP_1 pour réglage jours
{
inc_jour ();
afficher_date ();
lcd_gotoxy(1,1);
lcd_putc("incrémenter\n  jour");
}

if(BP_2==1)          // appui sur BP_2 pour réglage mois
{
inc_mois ();
afficher_date ();
lcd_gotoxy(1,1);
lcd_putc("incrémenter mois");
}

if(BP_3==1)          // appui sur BP_3 pour réglage années
{
inc_annee ();
afficher_date ();
lcd_gotoxy(1,1);
lcd_putc("incrémenter\n  année");
}
ds1302_set_datetime();

if (BP_I==1)          //appui BP_I pour quitter interruption
{

```

```

    delay_ms(30);
    re_BP_I();
    goto sortir;
}
if(BP_S==1)          // appui sur BP_S pour selectionner texte
                    //comment quitter l'interruption
{
    delay_ms(30);
    re_bp_s();
    afficher_commentaire2 ();
    goto saut_3;
}
else
{
    goto saut_2;
}
saut_3:

    if (BP_I==1)          //appui BP_I pour quitter interruption
    {
        delay_ms(30);
        re_BP_I();
        goto sortir;
    }
    else
    {
        if(BP_S==1)
        {
            goto star_1;
        }
        else
        {
            goto saut_3;
        }
    }

sortir:

    lcd_putc('\f');
}

//:.....
//=====programme principale=====
//:.....

void main()
{

    enable_interrupts(INT_EXT);
    enable_interrupts(GLOBAL);
    lcd_init();
    ds1302_init();          // initialisation le dateur DS1302
    init_ext_eeprom ();

```

```

ds1302_set_datetime();    // mise à l'heure (initialisation des compteurs)

debut:

d=getc();    // la donnée d represente la temperature mesuré par la carte 1
g=getc();    //la donnée d represente la quantité de gaz mesuré par la carte 1

lcd_gotoxy(1,1);
printf(lcd_putc,"temp=%03i celsius",d);          //affichage sur LCD
lcd_gotoxy(1,2);
printf(lcd_putc,"gaz=%03u ppm",g);              //affichage sur LCD

fonction_write_m ();      // fonction d'ecriture dans l'EEPROM externe

ds1302_get_datetime();    // lire les registres du dateur
lcd_gotoxy(1,3);
printf(LCD_PUTC,"%02u:%02u:%02u", heures, minutes, secondes);
lcd_gotoxy(1,4);
printf(LCD_PUTC,"%02u/%02u/%02u", jours, mois, annee);

//=====
=====
//+++++programme pour lire de la memoire externe+++++
//=====
=====

if(BP_s==1)                // lecture de l'EEPROM
{
    delay_ms(30);
    re_bp_s();
    goto prog_read_memo;
}

goto debut;

//=====
//:.....prog_read_memo:.....:
//=====
dar:
prog_read_memo:            //lire à partire de l' EEPROM
heures = read_ext_eeprom(0x00);
minutes = read_ext_eeprom(0x01);
jours = read_ext_eeprom(0x02);
mois = read_ext_eeprom(0x03);
annee= read_ext_eeprom(0x04);
d = read_ext_eeprom(0x05);
g = read_ext_eeprom(0x06);
lcd_putc("\f");
lcd_gotoxy(1,1);
printf(lcd_putc,"%02u:%02u",heures ,minutes);
lcd_gotoxy(1,2);

```

```

printf(lcd_putc,"%02u/%02u/%02u",jours ,mois,annee);
lcd_gotoxy(1,3);
printf(lcd_putc,"temp=%u c",d );
lcd_gotoxy(1,4);
printf(lcd_putc,"gaz=%u ppm",g );

i=4;
sav:
if(BP_1==1) // le bouton qui nous permet de lire les données (incremente)
{
re_bp_1();
lcd_putc('\f');
delay_ms(100);

d = read_ext_eeprom(i++);
lcd_gotoxy(1,1);
printf(lcd_putc,"temp=%u c",d );

g = read_ext_eeprom(i++);
lcd_gotoxy(1,2);
printf(lcd_putc,"gaz=%u ppm",g );

d = read_ext_eeprom(i++);
lcd_gotoxy(1,3);
printf(lcd_putc,"temp=%u c",d );

g = read_ext_eeprom(i++);
lcd_gotoxy(1,4);
printf(lcd_putc,"gaz=%u ppm",g );

if(i==8188) // lire les dernieres adresses de l'EEPROM
{
lire_de_EEPROM();
i=8188; // arreter incrementation à l'adresse 8188
}
}

if(BP_2==1) //le bouton qui nous permet de lire les données (decremente)
{
re_bp_2();
lcd_putc('\f');
delay_ms(100);
d = read_ext_eeprom(i--);
g = read_ext_eeprom(i--);
lcd_gotoxy(1,1);
printf(lcd_putc,"temp=%u c",d );
lcd_gotoxy(1,2);
printf(lcd_putc,"gaz=%u ppm",g );
d = read_ext_eeprom(i--);
g = read_ext_eeprom(i--);
lcd_gotoxy(1,3);
printf(lcd_putc,"temp=%u c",d );
lcd_gotoxy(1,4);
printf(lcd_putc,"gaz=%u ppm",g );
}

```

```

    if(i==0)                // arreter decrementation à l'adresse 0
    {
        lcd_putc('\f');
        goto dar;
    }
}

if(BP_S==1)//le bouton qui nous permet de sortire de la lecture de l'EEPROM
{
    delay_ms(30);
    lcd_putc('\f');
    re_bp_s();
    delay_ms(300);
    goto debut;
}
if (BP_3==1)                // le bouton qui nous permet d'effacer l'EEPROM
{
    delay_ms(30);
    re_bp_3();
    lcd_putc('\f');
    eeprom_reset();
    goto debut;
}

goto sav;

}

```

· **les fonctions :**

```

//=====
=====
//:.....fonctions pour incrimmentation heur:.....
//=====
=====
void inc_heur ()
{
    delay_ms(30);
    heures++;
    delay_ms(300);
    if (heures==24)
    {
        heures=0;
    }
    ds1302_set_datetime();
}
//-----
void inc_min ()
{
    delay_ms(30);
    minutes++;
}

```



```

delay_ms(300);
    if (minutes==60)
    {
        minutes=0;
    }
    ds1302_set_datetime();
}
//-----
void inc_sec ()
{
    delay_ms(30);
    secondes++;
    delay_ms(300);
    if (secondes==60)
    {
        secondes=0;
    }

    ds1302_set_datetime();
}

//=====
===
//:.....fonctions pour incrimmentation DATE:.....
//=====
=====

void inc_jour ()
{
    delay_ms(30);
    jours++;
    delay_ms(300);
    if (jours==32)
    {
        jours=0;
    }
    ds1302_set_datetime();
}
//-----
void inc_mois ()
{
    delay_ms(30);
    mois++;
    delay_ms(300);
    if (mois==13)
    {
        mois=0;
    }
    ds1302_set_datetime();
}
//-----
void inc_annee ()
{
    delay_ms(30);

```

```

annee++;
delay_ms(300);
if (annee==100)
{
    annee=0;
}

ds1302_set_datetime();
}

//=====
//:::fonctions texts::::::::::::::::::::::::::::::::::
//=====
=====
void texte_regler_heur()
{
    lcd_putc('\f');
    lcd_gotoxy(1,1);
    lcd_putc("APUIEZ SUR BP_S");
}

void afficher_comentaire()
{
    lcd_putc('\f');
    lcd_gotoxy(1,1);
    lcd_putc("POUR REGLER:");
    lcd_gotoxy(1,2);
    lcd_putc("HEURS:BP_1");
    lcd_gotoxy(1,3);
    lcd_putc("MINUTES:BP_2");
    lcd_gotoxy(1,4);
    lcd_putc("SEC: BP_3");
}
void afficher_commentaire2 ()
{
    lcd_putc('\f');
    lcd_gotoxy(1,1);
    lcd_putc("POUR SORTIR APPUI\nSUR BP_I\npour regler\nheur BP_1");
}
void afficher_coment_date()
{
    lcd_putc('\f');
    lcd_gotoxy(1,1);
    lcd_putc("POUR REGLER");
    lcd_gotoxy(1,2);
    lcd_putc("JOUR:BP_1");
    lcd_gotoxy(1,3);
    lcd_putc("MOIS:BP_2");
    lcd_gotoxy(1,4);
    lcd_putc(" ANNEES:BP_3");
}

```

```

void afficher_heur ()
{
    lcd_putc('\f');
    lcd_gotoxy(1,3);
    printf(LCD_PUTC,"%02u:%02u:%02u", heures, minutes, secondes);
}

void afficher_date ()
{
    lcd_putc('\f');
    lcd_gotoxy(1,3);
    printf(LCD_PUTC,"%02u/%02u/%02u", jours, mois, annee);
}

//=====
=====
//::::::::::::::::fonctions pour relacher BP::::::::::::::::::::::::::
//=====
=====

void re_bp_s()
{
    delay_ms(50);
    while (BP_S==1)
    {
    }
}

void re_bp_1()
{
    delay_ms(50);
    while (BP_1==1)
    {
    }
}

void re_bp_2()
{
    delay_ms(50);
    while (BP_2==1)
    {
    }
}

void re_bp_3()
{
    delay_ms(50);
    while (BP_3==1)
    {
    }
}

void re_BP_I()
{
    delay_ms(50);
    while (BP_I==1)

```

```

    {
    }
}
//=====
=====
//::::::::::::fonctions pour ecriture dans la memoire::::::::::::
//=====
=====
void fonction_write_m ()
{

if (minutes==r+1) // à chaque incrementation de minutes,enregistrer les données
{

    j=read_ext_eeprom(0);//à la premiere incrementation ecrire
    if(j==0XFF)        //à l'adresse (0) de l'EEPROM
    {
        write_ext_eeprom(0X00,heures);
        write_ext_eeprom(0X01,minutes);
        write_ext_eeprom(0X02,jours);
        write_ext_eeprom(0X03,mois);
        write_ext_eeprom(0X04,annee);
        write_ext_eeprom(0X05, d);
        write_ext_eeprom(0X06, g);
        goto ger;
    }
    s:

    j=read_ext_eeprom(h);    //verifier si l'adresse h est vierge,
    if(j!=0XFF)              //avant d'ecrire
    {
        h++;
        goto s;
    }

    j=read_ext_eeprom(h);
    if(j==0XFF)
    {
        if(h=8188) //4 dernieres adresse de l'EEPROM
        {
            write_ext_eeprom(h,heures);
            write_ext_eeprom(h++,minutes);
            write_ext_eeprom(h++,jours);
            write_ext_eeprom(h++,mois);
            write_ext_eeprom(h++,annee);
            lcd_putc('\f');
            lcd_gotoxy(1,1);
            printf(lcd_putc,"la memoire\nest pleine");
            lcd_putc('\f');
            delay_ms(8000);
            goto ger;
        }
        write_ext_eeprom(h, d);
        h++;
    }
}

```

```

        if(h=8188) //4 dernieres adresse de l'EEPROM
        {
            write_ext_eeprom(h,heures);
            write_ext_eeprom(h++,minutes);
            write_ext_eeprom(h++,jours);
            write_ext_eeprom(h++,mois);
            write_ext_eeprom(h++,annee);
            lcd_putc("\f");
            lcd_gotoxy(1,1);
            printf(lcd_putc,"la memoire\nest pleine");
            lcd_putc("\f");
            delay_ms(8000);
            goto ger;
        }
        write_ext_eeprom(h, g);
        goto ger;
    }

    ger:{ }
        r++;
    }

}

```

```

//=====
=====
//::::::::::::::::::::initialisation d'EEPROM::::::::::::::::::::
//=====
=====
void eeprom_reset()
{
    for (i=0;i<=8000;i++)
    {
        write_ext_eeprom(i,0xFF);
    }

}

//=====
=====
//::::::::::::::::::::lecture de l'EEPROM::::::::::::::::::::
//=====
=====
void lire_de_EEPROM()
{
    lcd_putc("\f");
    heures = read_ext_eeprom(8188);
    minutes = read_ext_eeprom(8189);
    lcd_gotoxy(1,1);
    printf(lcd_putc,"%02u:%02u",heures ,minutes);
    jours = read_ext_eeprom(8190);
    mois = read_ext_eeprom(8191);
}

```

```

annee = read_ext_eeprom(8192);
lcd_gotoxy(1,2);
printf(lcd_putc,"%02u/%02u/%02u",jours ,mois,annee);
}

```

· **Les variables :**

```

char d=0;      // d represente la temperature mesuré par le DS1621
char g=0;      // g represente la valeur du gaz mesuré par le HS135
int16 i=0;     // i represente les adresses de l'EEPROM(lors de la lecture)
int16 j=0;     // j represente les données vierges de l'EEPROM
int16 h=0;     // h represente les adresse de l'EEPROM (lors de l'écriture)
int16 r=minutes;//r represente une variable qui permet d'incrémenté les minutes

```

```

//=====
===
//+++++variables pour bouton poussoires+++++
//=====
===
#byte PORTA=0xF80      // adresse PORTA dans le PIC
#byte PORTB=0xF81      // adresse PORTB dans le PIC
#byte PORTC=0xF82      // adresse PORTC dans le PIC
#byte PORTD=0xF83
#byte PORTE=0xF84
#bit BP_S =PORTA.0      // BP_s relier à PORTA.0
#bit BP_1= PORTA.1      // BP_1 relier à PORTA.1
#bit BP_2= PORTA.2      // BP_2 relier à PORTA.2
#bit BP_3 =PORTA.3      // BP_3 relier à PORTA.3
#bit BP_I =PORTB.0      // BP_I relier à PORTB.0

```

Annexe

Plan Mémoire pour les données et registres internes (SFR : Special Function Register).

Le plan mémoire des données et des registres internes est découpé en 4 zones ou bank de 128 octets, pour accéder à une zone il faut positionner les bits RP0 (bit 5) et RP1 (bit 6) du registre STATUS.

RP1 : RP0	Zone sélectionnée (BANK)
0 0	De 00h à 7Fh : BANK 0
0 1	De 80h à FFh : BANK 1
1 0	De 100h à 17Fh : BANK 2
1 1	De 180h à 1FFh : BANK 3

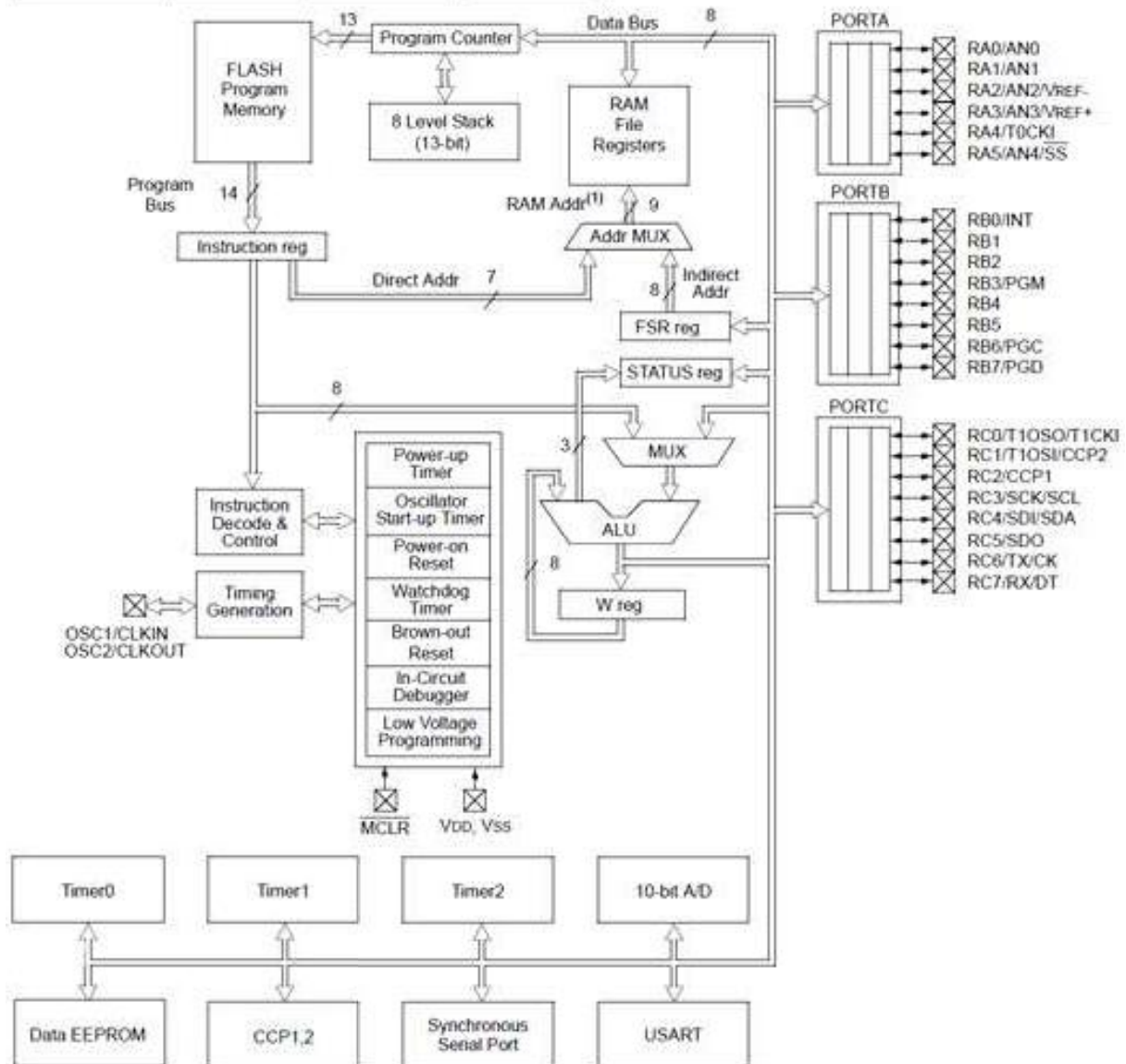
File Address	File Address	File Address	File Address
Indirect addr. ⁽¹⁾ 00h	Indirect addr. ⁽¹⁾ 80h	Indirect addr. ⁽¹⁾ 100h	Indirect addr. ⁽¹⁾ 180h
TMR0 01h	OPTION_REG 81h	TMR0 101h	OPTION_REG 181h
PCL 02h	PCL 82h	PCL 102h	PCL 182h
STATUS 03h	STATUS 83h	STATUS 103h	STATUS 183h
FSR 04h	FSR 84h	FSR 104h	FSR 184h
PORTA 05h	TRISA 85h		
PORTB 06h	TRISB 86h	PORTB 106h	TRISB 186h
PORTC 07h	TRISC 87h		
PORTD ⁽¹⁾ 08h	TRISD ⁽¹⁾ 88h		
PORTE ⁽¹⁾ 09h	TRISE ⁽¹⁾ 89h		
PCLATH 0Ah	PCLATH 8Ah	PCLATH 10Ah	PCLATH 18Ah
INTCON 0Bh	INTCON 8Bh	INTCON 10Bh	INTCON 18Bh
PIR1 0Ch	PIE1 8Ch	EEDATA 10Ch	EECON1 18Ch
PIR2 0Dh	PIE2 8Dh	EEADR 10Dh	EECON2 18Dh
TMR1L 0Eh	PCON 8Eh	EEDATH 10Eh	Reserved ⁽²⁾ 18Eh
TMR1H 0Fh		EEADRH 10Fh	Reserved ⁽²⁾ 18Fh
TICON 10h			
TMR2 11h	SSPCON2 91h		
T2CON 12h	PR2 92h		
SSPBUF 13h	SSPAD 93h		
SSPCON 14h	SSPSTAT 94h		
CCPR1L 15h			
CCPR1H 16h			
CCP1CON 17h			
RCSTA 18h	TXSTA 98h	General Purpose Register 16 Bytes	General Purpose Register 16 Bytes
TXREG 19h	SPBRG 99h		
RCREG 1Ah			
CCPR2L 1Bh			
CCPR2H 1Ch			
CCP2CON 1Dh			
ADRESH 1Eh	ADRESL 9Eh		
ADCON0 1Fh	ADCON1 9Fh		
General Purpose Register 96 Bytes	General Purpose Register 80 Bytes	General Purpose Register 80 Bytes	General Purpose Register 80 Bytes
	accesses 70h-7Fh	accesses 70h-7Fh	accesses 70h - 7Fh
Bank 0 7Fh	Bank 1 FFh	Bank 2 17Fh	Bank 3 1FFh

- Unimplemented data memory locations, read as '0'.
 * Not a physical register.

Note 1: These registers are not implemented on the PIC16F876.
 2: These registers are reserved, maintain these registers clear.

ORGANISATIONS INTERNES

Device	Program FLASH	Data Memory	Data EEPROM
PIC16F876	8K	368 Bytes	256 Bytes



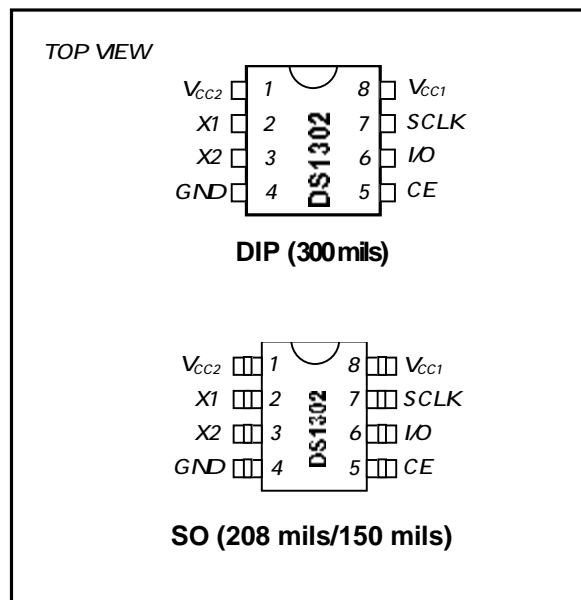
DS1302 Trickle-Charge Timekeeping Chip

www.maxim-ic.com

FEATURES

- Real-Time Clock Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the Week, and Year with Leap-Year Compensation Valid Up to 2100
- 31 x 8 Battery-Backed General-Purpose RAM
- Serial I/O for Minimum Pin Count
- 2.0V to 5.5V Full Operation
- Uses Less than 300nA at 2.0V
- Single-Byte or Multiple-Byte (Burst Mode) Data Transfer for Read or Write of Clock or RAM Data
- 8-Pin DIP or Optional 8-Pin SO for Surface Mount
- Simple 3-Wire Interface
- TTL-Compatible ($V_{CC} = 5V$)
- Optional Industrial Temperature Range: $-40^{\circ}C$ to $+85^{\circ}C$
- DS1202 Compatible
- Underwriters Laboratories (UL®) Recognized

PIN CONFIGURATIONS



ORDERING INFORMATION

PART	TEMP RANGE	PIN-PACKAGE	TOP MARK*
DS1302+	$0^{\circ}C$ to $+70^{\circ}C$	8 PDIP (300 mils)	DS1302
DS1302N+	$-40^{\circ}C$ to $+85^{\circ}C$	8 PDIP (300 mils)	DS1302
DS1302S+	$0^{\circ}C$ to $+70^{\circ}C$	8 SO (208 mils)	DS1302S
DS1302SN+	$-40^{\circ}C$ to $+85^{\circ}C$	8 SO (208 mils)	DS1302S
DS1302Z+	$0^{\circ}C$ to $+70^{\circ}C$	8 SO (150 mils)	DS1302Z
DS1302ZN+	$-40^{\circ}C$ to $+85^{\circ}C$	8 SO (150 mils)	DS1302ZN

+Denotes a lead-free/RoHS-compliant package.

*An N anywhere on the top mark indicates an industrial temperature grade device. A + anywhere on the top mark indicates a lead-free device.

UL is a registered trademark of Underwriters Laboratories, Inc.

Features

- Low-Voltage and Standard-Voltage Operation
 - 5.0 ($V_{CC} = 4.5V$ to $5.5V$)
 - 2.7 ($V_{CC} = 2.7V$ to $5.5V$)
 - 2.5 ($V_{CC} = 2.5V$ to $5.5V$)
 - 1.8 ($V_{CC} = 1.8V$ to $5.5V$)
- Low-Power Devices ($I_{SB} = 2 \mu A$ @ $5.5V$) Available
- Internally Organized 4096 x 8, 8192 x 8
- 2-Wire Serial Interface
- Schmitt Trigger, Filtered Inputs for Noise Suppression
- Bidirectional Data Transfer Protocol
- 100 kHz (1.8V, 2.5V, 2.7V) and 400 kHz (5V) Clock Rate
- Write Protect Pin for Hardware Data Protection
- 32-Byte Page Write Mode (Partial Page Writes Allowed)
- Self-Timed Write Cycle (10 ms max)
- High Reliability
 - Endurance: 1 Million Write Cycles
 - Data Retention: 100 Years
 - ESD Protection: >3,000V
- Automotive Grade and Extended Temperature Devices Available
- 8-Pin JEDEC PDIP, 8-Pin JEDEC SOIC, 8-Pin EIAJ SOIC, and 8-pin TSSOP Packages

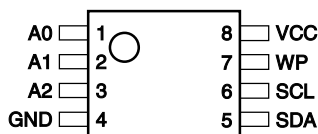
Description

The AT24C32/64 provides 32,768/65,536 bits of serial electrically erasable and programmable read only memory (EEPROM) organized as 4096/8192 words of 8 bits each. The device's cascadable feature allows up to 8 devices to share a common 2-wire bus. The device is optimized for use in many industrial and commercial applications where low power and low voltage operation are essential. The AT24C32/64 is available in space saving 8-pin JEDEC PDIP, 8-pin JEDEC SOIC, 8-pin EIAJ SOIC, and 8-pin TSSOP (AT24C64) packages and is accessed via a 2-wire serial interface. In addition, the entire family is available in 5.0V (4.5V to 5.5V), 2.7V (2.7V to 5.5V), 2.5V (2.5V to 5.5V) and 1.8V (1.8V to 5.5V) versions.

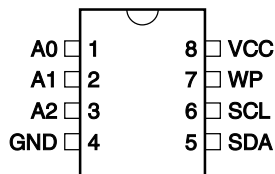
Pin Configurations

Pin Name	Function
A0 - A2	Address Inputs
SDA	Serial Data
SCL	Serial Clock Input
WP	Write Protect

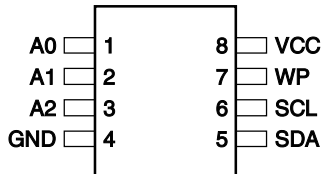
8-Pin TSSOP



8-Pin PDIP



8-Pin SOIC



2-Wire Serial EEPROM

32K (4096 x 8)

64K (8192 x 8)

AT24C32 AT24C64

Rev. 0336G-04/01



Détecteur de monoxyde de carbone HS134 réf : REHS134

Electronique-Diffusion

<http://www.elec dif.com>

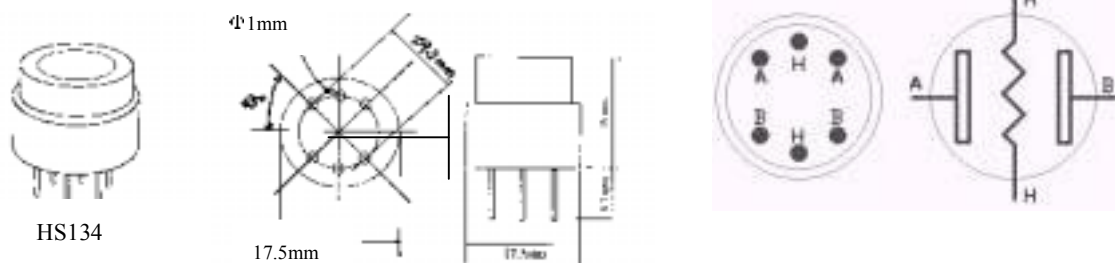


1. Characteristics

1. High sensitive and good selectivity to low carbon monoxide concentration.
2. Long life and reliable stability.

2. Structure, configuration, symbol of components.

1. Structure and configuration of HS-134 is shown as Fig. 1, sensitive components composed by micro Al_2O_3 ceramic tube, Tin Dioxide (SnO_2) sensitive layer, active carbon filtering layer include and made by plastic and stainless steel net.
- 2.2. Active carbon Particle reduces interference from gases of NO_x , Alkane, etc.
- 2.3 The enveloped HS-134 gas sensitive components have 6 pin, 4 of them are

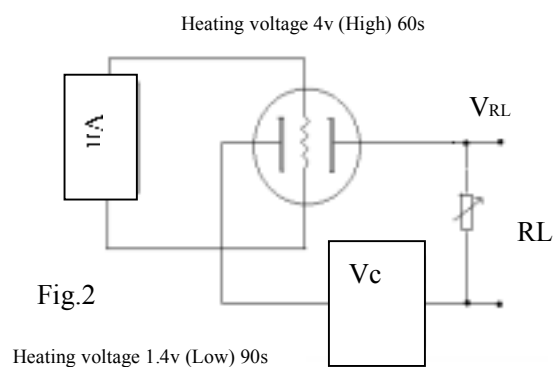


used to take signals(A, B) , and other 2 (H) are used for providing heating current.

3. Working Principle

A. Standard circuit

As shown in Fig 2, standard measuring circuit of HS-134 include 2 parts. one is heating circuit. Have time control function (the high voltage and the low voltage work circularly.) The second is the signal output circuit, it can accurately respond changes of surface resistance of the sensor.



- The surface resistance of the sensor R_s is obtained through effected voltage signal output of the load resistance R_L which series-wound. The relationship between them as below:

$$R_s R_L = (V_c - V_{RL}) / V_{RL}$$

BIBLIOGRAPHIE

Bibliographie

- [1] **YVES NOYELLE**, Belle programmation et langage C, (ellipses), 2001.
- [2] **CHRISTIAN TAVERNIER**, Programmation en c des pics, (DUNOD), 2005.
- [3] **DOGANE IBRAHIM**, Advanced pic microcontroller projects in c, from usb to zigbee with the PIC 18F Series, (*Elsevier Ltd*), Edition *Copyright # 2008*.
- [4] **CHRISTIAN DUPATY**, C minimum connaissances nécessaires à la programmation des microcontrôleurs en langage c, (une introduction au langage C A.N.S.I maj n°3).
- [5] **NIGEL GARDNER** et **PETER BIRNIE**, A collection of Working Application Ideas. (MICROCHIP).
- [6] **SID KATZEN**, The Quintessential PIC Microcontroller, (Springer-Verlag), le 8 novembre 2000.
- [7] **KERBALLIM** et **NAIT ACHOUR M.O**, Conception et réalisation d'une carte d'acquisition et de transmission des températures à base du pic 16F876 pour la raffinerie de HMD, *UMMTO, promotion 2005*.
- [8] **ZERIOUL LOUNES**, Etude et réalisation d'une chaine de mesure de co₂, *UMMTO, promotion 2010*.
- [9] **BABACI MOUNIR** et **BEN SALEM AHMED AMINE**, Mesure de la température de fusion des plastiques à base d'un microcontrôleur PIC18F452, *UMMTO, promotion 2010*.
- [10] **Manuel CAMUS**, Architecture de réception RF très faible coût et très faible puissance, Application aux réseaux de capteurs et au standard ZigBee, *En vue de l'obtention du DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE, soutenu le 29 Février 2008*.
- [11] **Cedric BEAUSSE** et **Mohamed BOUGUERRA** et **Hong Yu GUAN** et **El Ayachi MOKTAD**, Réalisation d'un réseau de capteurs de température sans fil basé sur le protocole ZigBee, *Projet avancé en systèmes embarqués, 2006 / 2007*.
- [12] **Adrien VAN DEN BOSSCHE**, Proposition d'une nouvelle méthode d'accès déterministe pour un réseau personnel sans fil à fortes contraintes temporelles, à l'*UNIVERSITÉ de TOULOUSE II, Ecole Doctorale Systèmes, Spécialité : Génie Informatique, Automatique et Traitement du Signal, Soutenue le 6 juillet 2007*.

[13] **Timothy William Arrowsmith**, A network processing node for light unmanned aircraft, *The Graduate School University of Kentucky*, 2007.

[14] **Willy Aubry, Mikel Azkarate-Askasua, Fabien Marteau, Han en Sabeur**, Développement d'une pile SIP et la création d'un capteur de température sur protocole ZigBee, réalisé à l'ECOLE NATIONAL SUPERIEUR "ELECTRONIQUE, INFORMATIQUE ET RADIOCOMMUNICATIONS" BORDEAU, le 7 février 2008.

[15] **Johan Lönn, Jonas Olsson**, ZigBee for wireless networking, at *Linköping University*, 15th March 2005.

[16] **Saloua HELALI**, Conception et réalisation de matériaux biofonctionnels pour des dispositifs capteurs impédimétrique, à l'Ecole centrale de lyon et de la faculté des sciences de Tunis, le 15 décembre 2005.

[17] **Nathalie Guichard**, Développement durable et hyperconsommation, *Université de Toulouse II*, le 2007/2008.

[18] <http://www.abcelectronique.com/>
Forum de discussion sur le domaine de l'électronique.

[19] www.microchip.com
Documentation du constructeur du pic (16F876).

[20] www.alldatasheet.net
Datasheet des différents composants électroniques utilisé.

[21] [http:// www.circuitDS1621-MicEle-troLinGenMetee.mht](http://www.circuitDS1621-MicEle-troLinGenMetee.mht).

[22] www.Wikipedia.fr

[23] [www.etudeetrealisationd'uncapteursansfilZigBee.<<open silicium>>.mht](http://www.etudeetrealisationd'uncapteursansfilZigBee.<<open%20silicium>>.mht)

Glossaire

Glossaire

B

BF (Beacon Frame) : Est l'une des adresses de gestion dans la norme IEEE 802.11 WLAN. Elle contient toutes les informations sur le réseau. Les Trames Beacon sont transmises périodiquement pour annoncer la présence d'un réseau LAN sans fil. Les Trames Beacon sont transmises par le Point d'Accès (AP) dans une infrastructure BSS .

C

CSMA/CA (Carrier Sens Multi Acces-Avidance collusion) : Le protocole CSMA/CA utilise un mécanisme d'esquive de collision basé sur un principe d'accusé de réception réciproque entre l'émetteur et le récepteur.

D

DSSS (Direct Sequence Spread Spectrum): Est l'acronyme de *Direct Sequence Spread Spectrum*, en français étalement de spectre à séquence directe, est une technique d'étalement de spectre utilisée dans les communications par satellite, les réseaux sans fil et plus précisément le Wi-Fi. Le but du DSSS est d'une part, de rendre les signaux occupant une bande de fréquences réduite, comme un signal de parole, plus résistant aux brouillages rencontrés lors de la transmission; d'autre part permettre à plusieurs liaisons de partager la même fréquence porteuse (Accès multiple par répartition par code).

F

FCS (Frame Check Sequence): Est le code de détection d'erreurs ajouté à la fin d'une trame. C'est le reste de la division modulo 2 de la trame par un polynôme de degré N (exemple HDLC, polynôme de degré 16, FCS de 16 bits). Le récepteur effectue la division de la trame reçue (trame + FCS) par le même polynôme. Si le reste de cette division est 0, il n'y a pas eu d'erreur dans la transmission.

I

IEEE (Institute of Electrical and Electronics Engineers): L'Institute of Electrical and Electronics Engineers ou IEEE (que l'on peut prononcer « i trois e ») (en français : l'« Institut des ingénieurs électriciens et électroniciens ») est une organisation à but non lucratif. L'IEEE compte plus de 390 000 membres et possède différentes branches dans plusieurs parties du monde. L'IEEE est constituée d'ingénieurs électriciens, d'informaticiens, de professionnels du domaine des télécommunications, etc. L'organisation a pour but de promouvoir la connaissance dans le domaine de l'ingénierie électrique (électricité et électronique).

ITU (International Télécommunication Union) : L'Union internationale des télécommunications (UIT, ou en anglais International Telecommunication Union ou ITU) est la plus ancienne organisation intergouvernementale de technique de coordination, puisqu'elle a été créée sous le nom d'Union internationale du télégraphe en 1865. Elle adopte son nom actuel en 1932 et se voit rattachée directement aux Nations unies en 1947.

Glossaire

M

MAC (Medium Access): Est une sous-couche de la couche liaison de données spécifiés dans les sept couches du modèle OSI (couche 2).Il fournit l'adresse et le canal d'accès des mécanismes de contrôle qui permettent à plusieurs terminaux ou des nœuds du réseau de communiquer au sein d'un réseau multi-points, généralement un réseau local (LAN) ou un réseau métropolitain.

P

PPDU (PHY Protocol Data Unit): L'ensemble des informations échangées entre niveaux dans le système de couches.

V

VLSI (very large scale intégration) : Est le processus de création de circuits intégrés en combinant des milliers de transistors dans une puce unique. VLSI a commencé dans les années 1970, lorsque les complexes semi-conducteurs de la communication et de la technologies étaient développées. Le microprocesseur est un dispositif VLSI.

.