

Je tiens à remercier le Pr.Guessoum de m'avoir encadré, pour ses efforts fournis, sa patience, et son entière disponibilité. Qu'il trouve ici l'expression de ma gratitude pour ses précieux conseils durant l'élaboration de mon projet.

Un grand merci à mon Co-promoteur Mr.Kazed pour ses conseils et ses informations qui m'ont beaucoup aidé durant mon travail.

Sans oublier un modeste merci pour tous les enseignants du département d'électronique qui ont participé à ma formation.

Enfin mes derniers remerciements vont à toutes les personnes qui nous ont soutenus de près ou de loin durant ce travail.

En dernier je n'oublierai pas ABDELAZIZ qui m'à tellement aider à réaliser le projet qui est entre vos mains

مخلص : هذا المشروع يقوم على انجاز جهاز للتحكم عن بعد يعمل بطريقة ذاتية من اجل الإشراف على نظام في حالة عمل بدون الحضور شخصيا, مهما كان , آلة ,باب أو سيارة. الجهاز مرفق بمتحكم يشرف على عمله بصفة مبرمجة للحصول على تقرير النظام المراقب من اجل التدخل عند الضرورة, و ذلك باستعمال جهاز للاستقبال و إرسال الحرارة المقاسة على شكل رسالة هاتفية قصيرة.

كلمات المفاتيح: متحكم PIC ;وحدة GSM ;جهاز استقبال DS1621 ; اتصال I²C

Résumé : Ce projet consiste à créer un dispositif de contrôle à distance fonctionnant de manière autonome à l'aide d'un module GSM pour superviser un système en état de marche tout en n'étant pas physiquement sur place, qu'il soit machine, porte ou véhicule. Le dispositif est équipé d'un Microcontrôleur qui gère son fonctionnement de façon programmé pour avoir un compte rendu de l'état du système observé afin de pouvoir intervenir si nécessaire.

Mots clés : Microcontrôleur ; Module GSM ; Capteur DS1621 ; Communication I²C.

Abstract : This project purpose is to create a device that allows its user to control and oversee a working system wirelessly like a machine, a vehicle or a door through a GSM module. The device is equipped with a microcontroller which manages the operation already programmed by user for an account of the status and allows its user to intervene when necessary.

Keywords : Microcontroller ; GSM Module ; DS1621 Sensor ; I²C communication.

Table des matière

1	Chapitre 1 : Introduction Generale.....	2
1.1	Introduction.....	2
1.2	Fonctionnement	2
1.3	Bénéfices du projet	2
1.4	Cahier des charges	3
1.4.1	But du projet	3
1.4.2	Utilité du système	3
1.4.3	Structure générale du système	3
1.5	Fonctionnement générale du système	3
1.5.1	Présentation de la carte d'acquisition (contrôle)	4
1.5.2	Présentation de la carte d'interface pour le module GSM	5
1.6	Conclusion	5
2	Chapitre 2 : Le GSM et Commandes AT.....	6
2.1	Historique	6
2.2	Le réseau cellulaire	7
2.2.1	Le radio téléphone de la première génération	7
2.2.2	Architecture cellulaire	8
2.3	La norme GSM	10
2.3.1	présentation de la norme GSM	11
2.3.2	Module GSM MC45	12
a	Caractéristiques techniques	12
b	Les entrées-sorties du MC45	13
c	Structure interne du module MC45	14
2.3.3	Module GSM Télit GM862	16
a	Caractéristiques techniques	17
b	Les entrées sorties du module GM 862	17
c	Structure interne du module GM 862	18
2.4	Les Commandes AT	19
2.5	Fonctionnement	20
2.6	Commandes dédiées au service SMS	20
2.7	Conclusion	21

3	Chapitre 3 : Capteur et bus I ² C.....	22
3.1	Les Capteurs	22
3.2	Classification des capteurs	22
3.2.1	Apport énergétique	22
a	Capteur passif	22
b	Capteur Actif (Direct)	22
3.2.2	Types de sorties	23
a	Capteur Analogique	23
b	Capteur Numérique	23
c	Capteur Logique	23
3.2.3	Type de détection	24
3.3	Principale caractéristique d'un capteur	24
3.4	Le capteur Dallas 1621	24
3.4.1	Caractéristiques du capteur DS1621	25
3.4.2	Description détaillée des PIN du DS1621	26
3.4.3	Diagramme de fonctionnement du DS1621	27
3.5	Le bus I ² C	27
3.6	Caractéristique du bus I ² C	28
3.7	Le protocole I ² C	29
3.7.1	Terminologie d'un transfert sur le bus I ² C	29
3.7.2	La notion d'adresse	31
3.7.3	Adressage des composant I ² C	31
3.7.4	Transfert d'une donnée sur le bus	32
3.8	Conclusion	32
4	Chapitre 4 : Etude et conception	33
4.1	Présentation Générale	33
4.2	Harvard contre Von Neumann	33
4.3	Les points forts de l'architecture RISC	34
4.4	Contenu du microcontrôleur	35
4.5	La jungle des références	37
4.6	Les différentes familles des PIC	37
4.7	Les éléments essentiels du PIC 16F876	38

4.8	Appellations normalisées des diverses pattes disponibles	39
4.9	Architecture interne du PIC 16F87X	42
4.10	Le cœur du PIC	42
4.11	La mémoire du PIC	45
4.11.1	La mémoire programme	45
4.11.2	La mémoire RAM	46
4.11.3	La mémoire EEPROM	47
4.12	Les ports d'entrée / sortie généraux	48
4.13	L'USART.....	49
4.14	Le SSP	50
4.15	Pourquoi programmer en C?.....	51
4.16	Choix du compilateur	52
4.17	Choix du PIC : Le 16F876.....	53
4.18	Programmation du PIC	53
4.19	Présentation du MikroC Pro	54
4.20	Création du Programme	55
4.20.1	Première étape : Crée un nouveau projet.....	55
4.20.2	Deuxième étape : Choisir notre PIC dans la liste.....	55
4.20.3	Troisième étape : Choisir la fréquence de fonctionnement du programme.....	56
4.20.4	Quatrième étape : Choisir d'inclure toutes les bibliothèques ou pas.....	56
4.20.5	Dernière étape : Fin de la création du projet.....	57
4.20.6	Ecriture du Programme C.....	57
4.21	Schéma du projet sur ISIS Proteus	63
4.22	Réalisation du projet.....	64
4.23	Conclusion	64
5.	Conclusion Générale	65

1.1 Introduction :

La communication à distance apparaît de nos jours comme essentielle dans la politique de transfert de données ou encore de contrôle sans fils, donc les équipements des entreprises industrielles modernes doivent être équipés de dispositif de commande et de contrôle à distance.

1.2 Fonctionnement :

Dans le cadre du projet de fin d'études, on se propose la réalisation d'une carte de contrôle à distance autour d'un module GSM ou un téléphone portable permettant à son utilisateur de contrôler des systèmes à distance.

L'utilisation de cette carte est très facile ou l'utilisateur pourra à l'aide de son téléphone portable contrôler ou surveiller une machine ou un système ou n'importe quel autre dispositif tout en n'étant pas physiquement présent, et ceci se fait par la réception d'un simple SMS qui contient une information ou un compte rendu sur l'état d'avancement du dispositif en question.

1.3 Bénéfices du projet:

Notre projet présente des avantages qui s'offrent à l'industrie et tout aussi bien à l'individu et se présentant comme suit :

- Réduction du temps d'arrêt des systèmes en cas de défaillance.
- Contrôle des machines ou systèmes de n'importe quel endroit.
- Gain de temps est un gain d'argent.

1.4 Cahier des charges :

1.4.1 But du projet :

La réalisation du système consiste à mettre en œuvre un microcontrôleur PIC interfacé avec un module GSM, qui recueillera les données d'un capteur de température pour les envoyer à un téléphone portable.

Afin de concrétiser cette opération on passera par les étapes suivantes :

- Choix d'un téléphone portable.
- Conception du système.
- Réalisation des cartes.
- Programmation et test.

1.4.2 Utilité du système :

Le système doit permettre à l'utilisateur le contrôle d'un ou plusieurs dispositifs (machine, porte, usine,...), la surveillance à distance en récupérant, à tout instant, l'état de son système via un SMS. Ce contrôle peut être immédiat ou de façon programmé.

1.4.3 Structure général du système :

Notre système est composé de deux cartes, la première est une carte d'interface pour le module GSM et la deuxième une carte d'acquisition et de contrôle. La liaison entre les deux se fait à travers une communication série (RS232).

1.5 Fonctionnement générale du système :

Le fonctionnement du système est simple, la carte d'acquisition prélève la température qui sera ensuite traitée et transmise à la carte du module GSM à travers le port série et en dernier lieu l'information sera envoyée comme un SMS au téléphone portable.

On recevra les informations que la carte d'acquisition récupérera en SMS dans le téléphone portable.

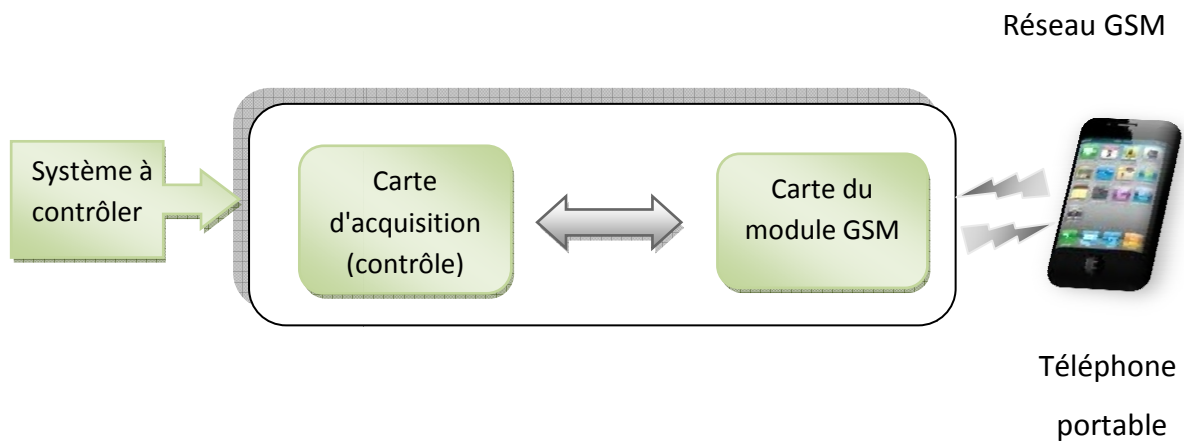


Figure 1. schéma descriptif de fonctionnement général du système.

1.5.1 Présentation de la carte d'acquisition (contrôle) :

C'est une carte qui permet le prélèvement de données dans notre cas température et qui commande un capteur de température grâce à un microcontrôleur PIC 16F876

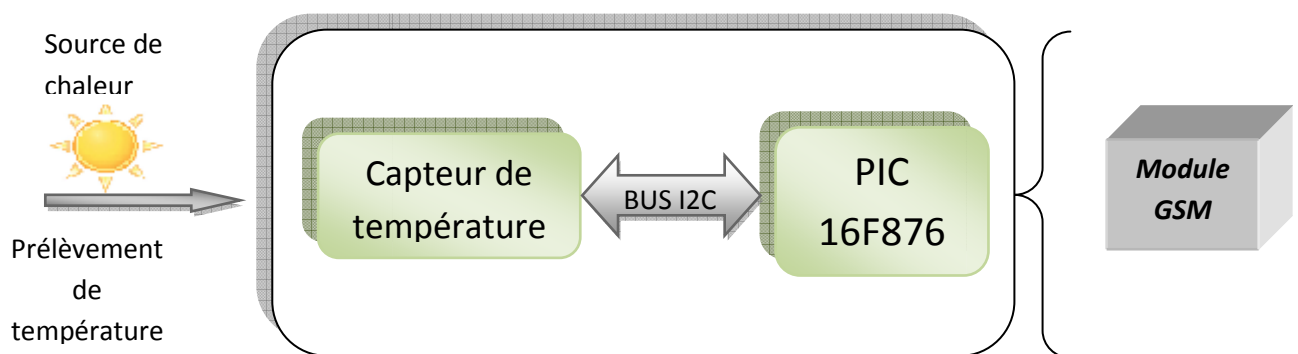


Figure 2. structure de la carte d'acquisition.

1.5.2 Présentation de la carte d'interface pour le module GSM :

la carte d'interface du module GSM est équipée de :

- un module GSM qui joue le rôle d'émetteur des SMS traités au préalable à l'utilisateur.
- un circuit d'interface MAX232 permettant la liaison série RS232 pour l'adaptation des signaux micro power 5V/3V.

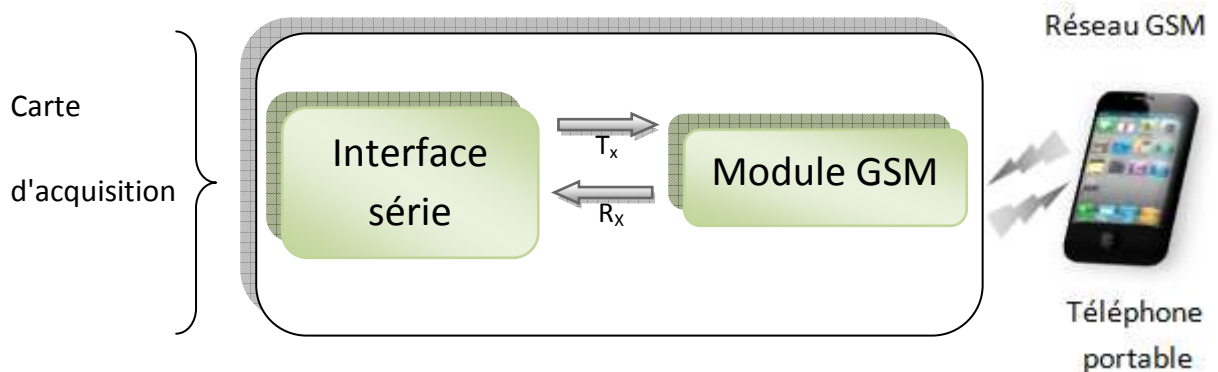


Figure 3. Structure de la carte d'interface du module GSM.

1.6 Conclusion :

Dans ce chapitre on a présenté la structure globale de notre réalisation ainsi que les différentes interfaces entre les cartes qui constituent le montage (Carte d'interface du module GSM et la carte de contrôle). Ces cartes sont un atout majeur dans la réalisation de notre projet.

2.1 Historique : [1]

De l'antiquité jusqu'à la fin du XIX^{ème} siècle, l'écrit a été le seul moyen de communication entre deux personnes séparées par une distance importante. Durant cette période antérieure aux télécommunications, les informations voyagent au rythme des messagers. Ce n'est que depuis le début du XX^{ème} siècle que les télécommunications ont permis d'éliminer la distance entre des personnes qui souhaitent communiquer. Nous allons parcourir rapidement les étapes majeures de l'histoire des télécommunications qui ont précédé la naissance du GSM, avant d'aborder le GSM plus ou moins en détail à travers ses protocoles, les équipements qui composent un réseau et les caractéristiques des terminaux.

L'acquisition des connaissances théoriques et du savoir-faire dans le domaine des télécommunications se situe au XIX^{ème} siècle.

En 1876, le savant canadien Graham Bell (1847-1922) invente le téléphone, premier moyen de télécommunication moderne. Une communication est possible entre deux postes fixes reliés par une paire de fils (la paire téléphonique).

En 1887, le physicien allemand Heinrich Hertz (1857-1894) découvre les «ondes hertziennes» : les ondes radio.

En 1896 à Bologne, le physicien italien Guglielmo Marconi (1874-1937) réalise la première transmission radio. L'expérience a lieu dans son grenier où est installé son laboratoire. Marconi réussit à télécommander une sonnette électrique, la distance entre l'émetteur et le récepteur est de quelques mètres. Marconi brevète son invention, qu'il nomme «Télégraphe sans fil» ; l'histoire retiendra seulement le sigle TSF. Pour perfectionner son invention, Marconi séjourne en Angleterre de 1897 à 1901 où il réalise la première transmission radio transatlantique entre la Cornouaille et Terre-Neuve en 1901. Cette liaison radio ouvre l'ère des télécommunications longues distances. En moins de dix ans, Marconi développe une technique de transmission radio efficace et fiable. Il reçoit en 1909 le prix Nobel de physique pour ses travaux sur les «ondes hertziennes».

Les ondes hertziennes radioélectriques permettent de communiquer entre deux stations fixes, mais surtout, elles offrent la solution idéale pour communiquer avec les mobiles de tous types, les bateaux, les avions, les satellites, les automobiles et les personnes, et cela quelque soit la distance entre les deux correspondants.

2.2 Le réseau cellulaire :

Nous allons présenter la radiotéléphonie de la première génération, puis on introduira les concepts et les composants de l'architecture d'un réseau de radiotéléphone cellulaire de la seconde génération.

2.2.1 Le radio téléphone de la première génération : [1]

L'architecture d'un réseau pour la première génération s'inspire de celle des systèmes de radiodiffusion, c'est-à-dire d'un système de communication point-multipoint. Un point focal est la station de base. Elle diffuse vers un vaste territoire dans lequel des terminaux mobiles se déplacent. Un réseau peut se résumer à une puissante station de base, ses ondes portent jusqu'à 50km. Un faisceau de lignes téléphoniques relie la station de base, avec un commutateur, le centre de commutation du service mobile MSC (Mobil Switching Center). Le commutateur relie au réseau téléphonique commuté (RTC) le réseau de radiotéléphone.

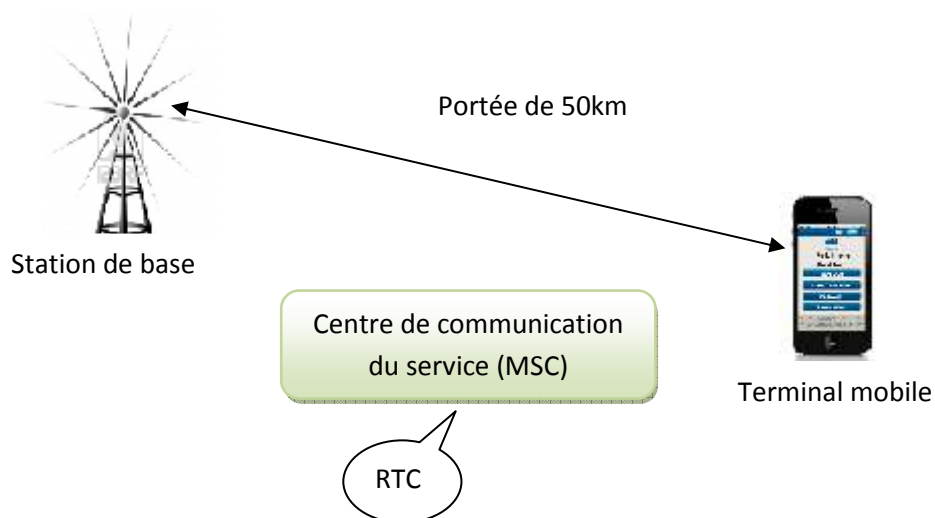


Figure4. Architecture d'un réseau de la première génération.

Malheureusement ce type d'architecture à un défaut de taille, le nombre d'abonnés du réseau est limité par le nombre de canaux radio, donc pour accroître le nombre d'abonnés plus de canaux sont indispensables, ce qui réserve ce service pour une catégorie ciblée de la population.

La première évolution consiste à allouer un canal de communication uniquement quand un terminal en a besoin. De cette façon le nombre d'abonnés peut être supérieur au nombre de canaux radio.

Pour accroître de façon significative le nombre d'abonnés dans un réseau, le concept « cellule » est introduit, ce qui est une évolution majeure. Le nombre de canaux radio est maintenant en relation directe avec le nombre de cellules du réseau. L'accès au radiotéléphone est désormais possible pour tous les abonnés de téléphone filaire.

2.2.2 Architecture cellulaire :

Le concept de base d'un réseau de radiotéléphone cellulaire est d'une part la division du territoire couvert par un réseau en un ensemble d'espaces appelés « cellule » et d'autre part le partage des canaux radio entre les cellules. C'est une rupture avec l'architecture des réseaux de la première génération de radiotéléphone, où il n'y a qu'une cellule de très grande dimension. Dans chaque cellule d'un réseau de seconde génération, il y a une station de base, c'est un émetteur récepteur. Ce dernier interconnecte le réseau mobile fixe avec le réseau commuté filaire(RTC).

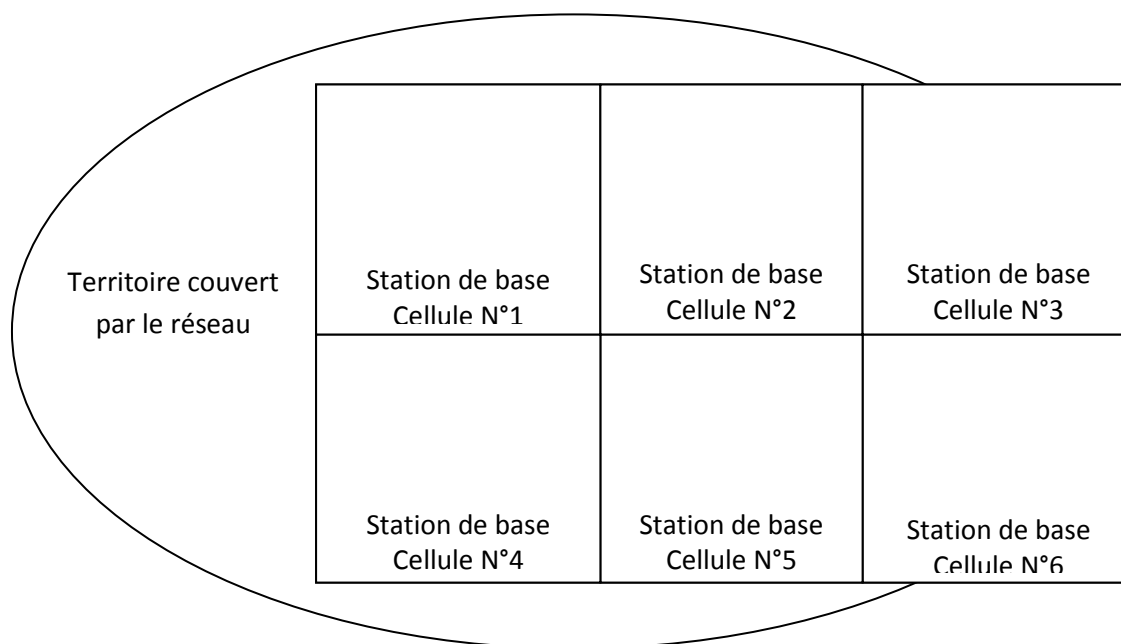


Figure5. Division du territoire d'un réseau en n cellules.

La taille d'une cellule varie en fonction d'un ensemble de contraintes se présentant comme étant :

- le relief du territoire {plaine, montagne}.
- la localisation {urbaine, rurale, suburbaine}.
- la densité d'abonnés.
- la nature des constructions {maison, pavillon, tours}.

Un groupe de fréquences radios définissant les canaux de communication est attribué à une cellule. Deux cellules adjacentes n'ont pas de canaux de communication communs. Pour protéger une cellule des interférences de co-canal, une distance minimale de deux cellules sépare deux cellules utilisant les canaux de communication.

La cellule est l'unité géographique d'un réseau. L'unité d'utilisation des fréquences radio définissant les canaux de communication est un motif de sept cellules (le cluster).

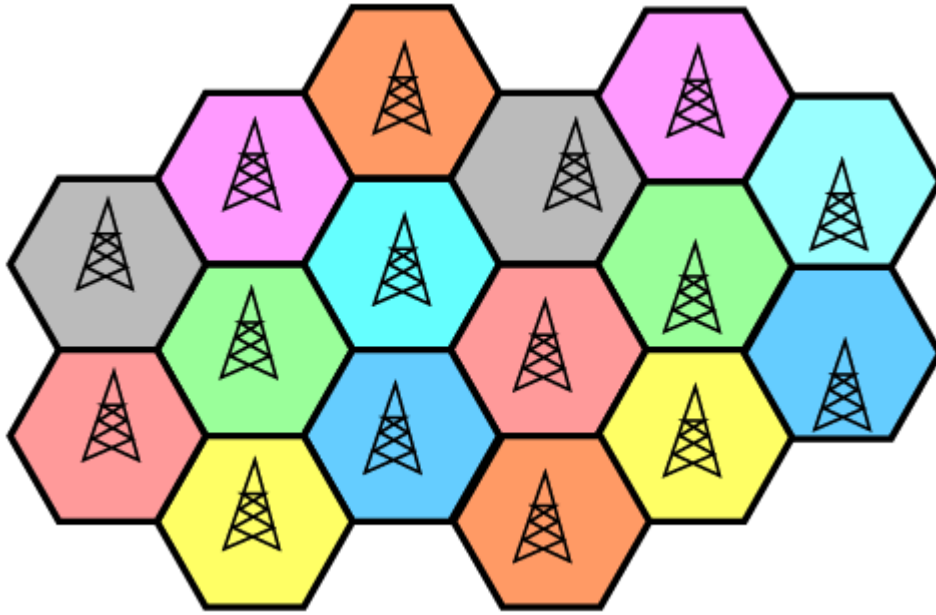


Figure6. Motif de sept cellules.

2.3 La norme GSM :

Un utilisateur considère le radiotéléphone comme un outil de communication à l'utilisation simple et agréable et pour un coût d'utilisation raisonnable. Pour cela le GSM présente des avantages plutôt intéressants comme suit :

- une transmission de la parole avec une qualité semblable au téléphone classique.
- une large couverture territoriale.
- la transmission de données.
- une grande disponibilité du service.
- la transmission des messages.

2.3.1 présentation de la norme GSM :

Le réseau GSM offre à ses abonnés trois catégories de service qui sont :

- les services support.
- les télé-services.
- les services supplémentaires.

La figure ci dessous présente les composants d'un réseau GSM.

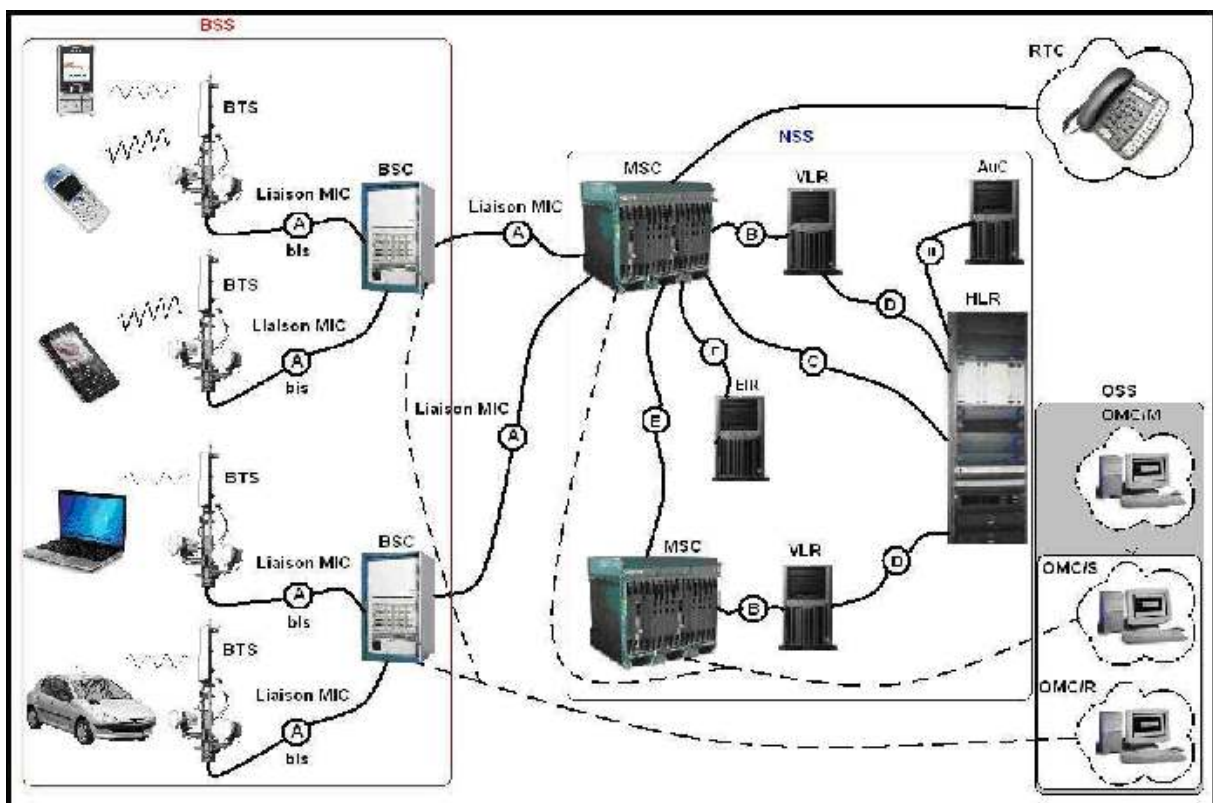


Figure7. structure d'un réseau GSM. [2]

2.3.2 Module GSM MC45 : [3]

Le module MC45 est un module GSM/GPRS qui fonctionne à trois bandes de fréquences GSM 900MHz, GSM 1800MHz et GSM 1900MHz. Ce module nous permet d'envoyer et de recevoir des données numérisées en utilisant la norme GSM et GPRS, l'utilisation du MC45 englobe plusieurs applications.

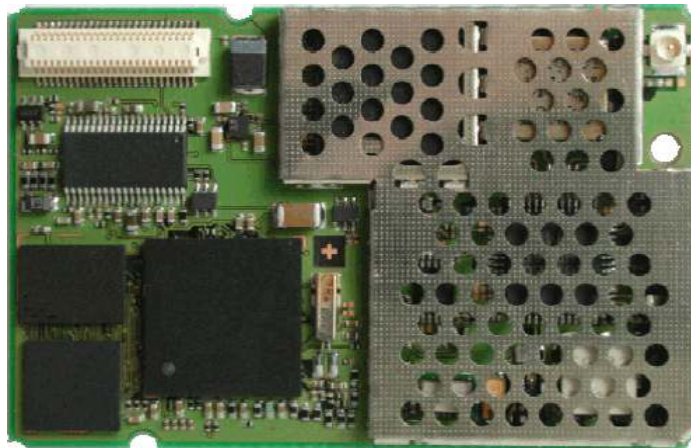


Figure8. Module GSM SIEMENS MC45.

a **Caractéristiques techniques :**

Le module MC45 propose de nombreuses solutions à rendement élevé compte tenu de son architecture qui est principalement constituée d'un microcontrôleur de bande de base, l'alimentation d'énergie ASIC (Application Specific Integrated Circuit) et d'autre part le MC 45 nous offre une flexibilité pour l'intégration facile avec l'interface Home-Machine donc il compte deux interfaces séries (ASCO et ASC1) et aussi le module est économique en énergie. Ses caractéristiques sont données par le tableau suivant :

Alimentation	3.2v – 4.5v
Bandes de fréquence	GSM 900 MHz, GSM 1800 MHz, GSM 1900 MHz
Puissance de transmissions	Classe 4 (2w) à GSM900 MHz Classe 1 (1W) à GSM1800 MHz et à GSM 1900 MHz
Interface de SIM	1.8 ou 3v
Antenne externe	500hm
2 Interfaces séries : ASC0, ASC1	niveau 2.65v, bidirectionnel ASC0 constitué de 8 ports ASC1 constitué de 4 ports
Données GPRS	Débit maximal pour recevoir des données par GPRS 85,6 kbps Débit maximal pour le transfert des données par GPRS 26,6 kbps
Température normale	-20°C +55°C

Tableau 1. Caractéristique technique du module GSM MC45.

b Les entrées-sorties du MC45 :

le module est équipé d'un connecteur bord to bord DF12 qui contient 12 pin d'entrée-sortie divisés en quatre catégories :

- Commande d'alimentation.
- Deux interfaces série.
- Deux interfaces audio analogique et une interface audio numérique.
- Interface de la carte SIM.

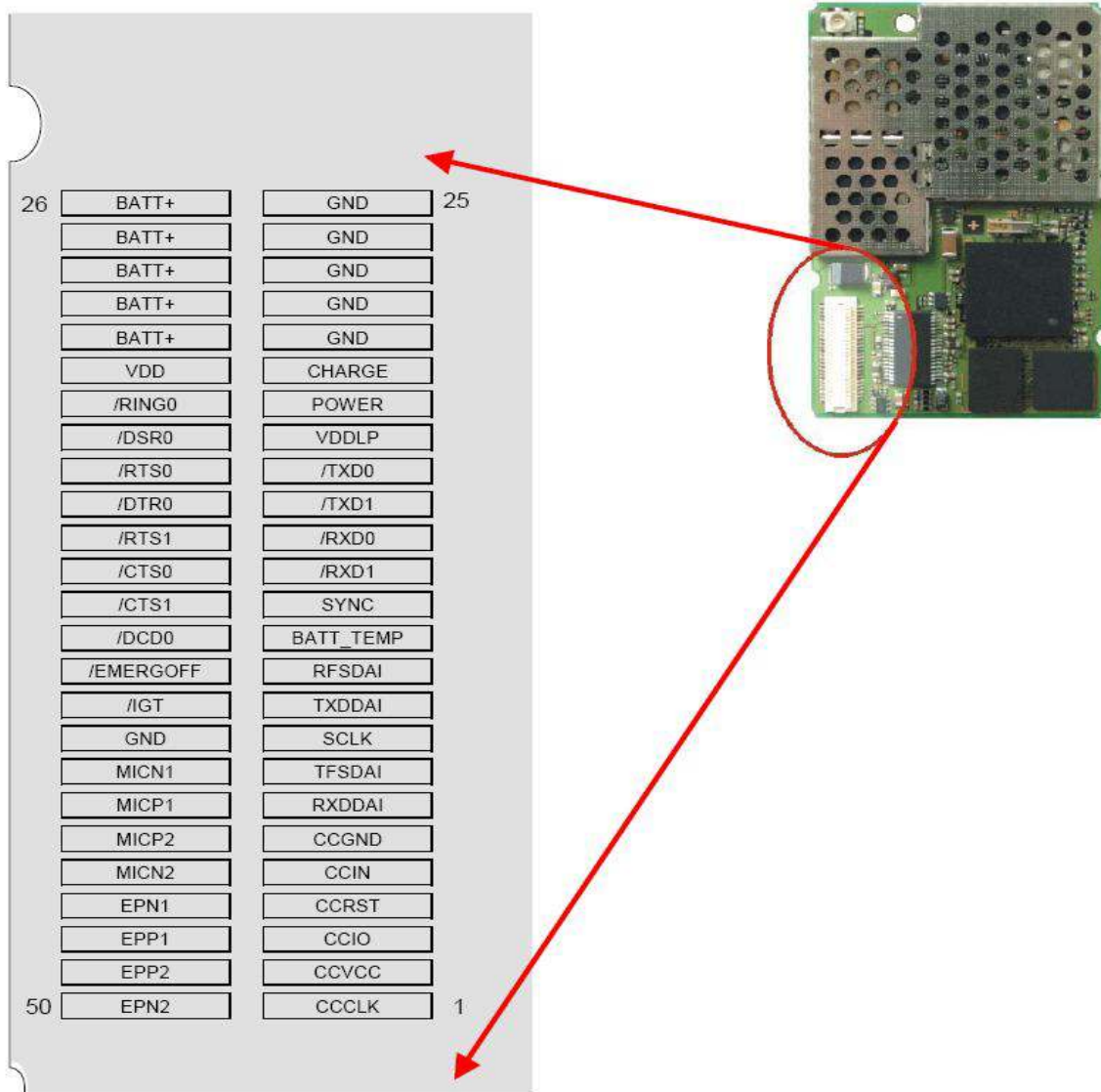


Figure9. Emplacement du PIC.

c Structure interne du module MC45 :

Le MC45 est constitué essentiellement de deux parties, la première est la section bande de base et la seconde est la section Radio Fréquence (RF).

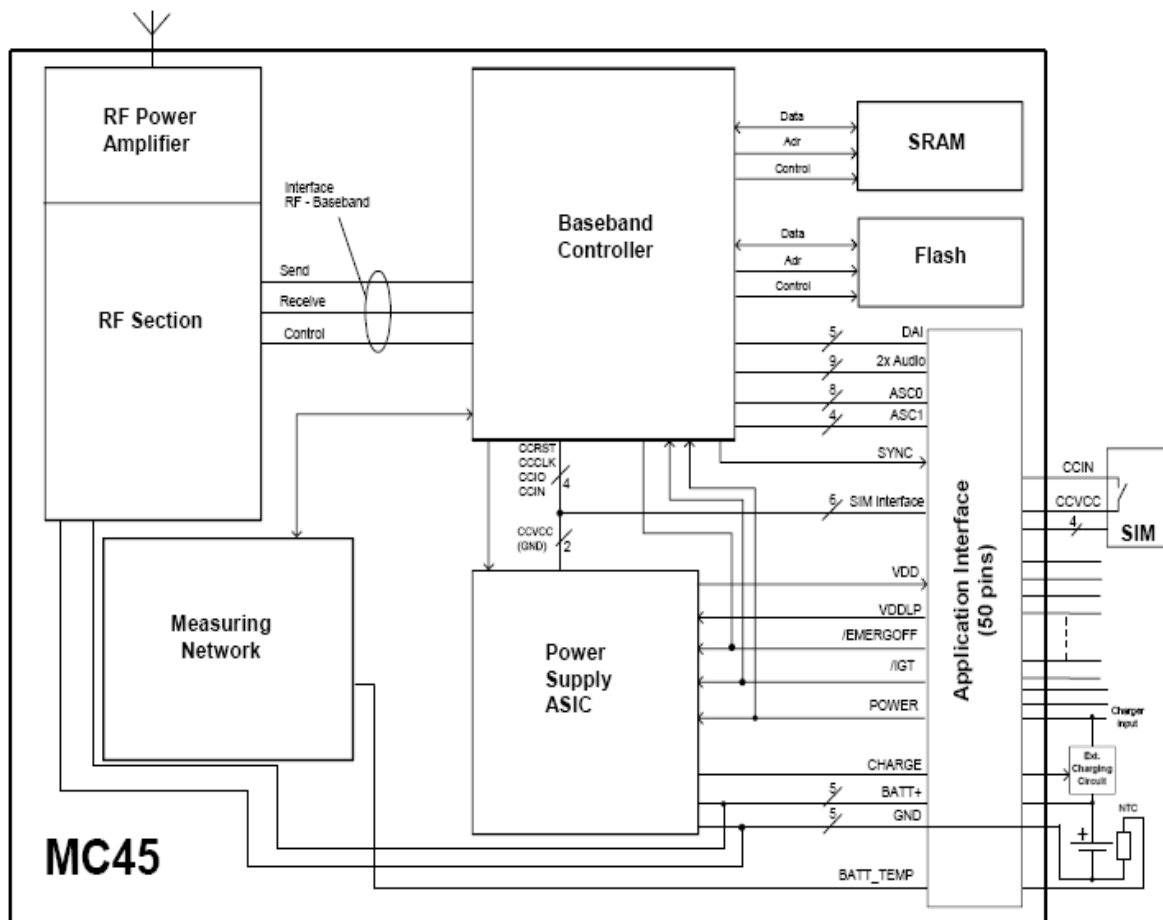


Figure10. Schéma interne du SIEMENS MC45.

1) SECTION RADIO FREQUENCE :

Les émetteurs récepteurs RF sont des dispositifs électroniques qui reçoivent et démodulent des signaux de fréquence radio et qui sont aussi aptes à moduler à leurs tours et envoyer des signaux du même type de fréquence. Il sont employés dans divers applications d'échange de données. Un émetteur récepteur RF dispose d'une antenne pour recevoir et émettre des signaux et aussi il dispose d'un amplificateur pour séparer un signal spécifique des autre signaux que l'antenne perçoit et les modulateurs ou les codeurs récupèrent l'information envoyée à l'origine par l'émetteur.

2) SECTION BANDE DE BASE :

cette partie est le cerveau du module GSM en quelque sorte compte tenu que c'est la partie intelligente de ce dernier, elle gère les données entrantes et sortantes, elle se compose d'un microcontrôleur de bande de base et une alimentation ASIC (Application Specific Integrated Circuit).

2.3.3 Module GSM Télit GM862 :



Figure11. Module GSM GM862.

Le GM862 est un module GSM fonctionnel à quatre bandes de fréquence GSM 850MHz, GSM 900MHz, GSM 1800MHz et GSM 1900MHz. Le GM862 utilise la norme du réseau GSM et GPRS pour la transmission et la réception des données numérisées et son utilisation est présente dans divers domaines comme :

- Télémétrie et Télé-contrôle.
- Systèmes de sécurité.
- Téléphones et Payphones.

a Caractéristiques techniques :

Alimentation	3.4v – 4.2v
Bandes de fréquence	GSM 850MHz, GSM 900MHz, GSM 1800MHz, GSM1900 MHz
Puissance de transmissions	Classe 4 (2w) à GSM900 MHz Classe 1 (1W) à GSM1800 MHz et à GSM 1900 MHz
Interface de SIM intégré	1.8 ou 3v
Antenne externe	500hm
2 Interfaces séries : ASC0, ASC1	niveau 2.8v, bidirectionnel
Données GPRS	Débit maximal pour recevoir des données par GPRS 85,6 kbps Débit maximal pour le transfert des données par GPRS 26,6 kbps
Interface de caméra	
Température normale	-20°c +55°c

Tableau2. Caractéristiques techniques du module GM862.**b Les entrées sorties du module GM 862**

GM862 est équipé d'un connecteur board-to-board CSTP de Molex contient 50 pin d'entres/sorties qu'on peut diviser en quatre types :

- Commande d'alimentation.
- Deux interfaces série.
- Deux interfaces audio analogiques et une interface audio numérique.
- Interface de caméra.
- Interface intégré de la carte SIM.

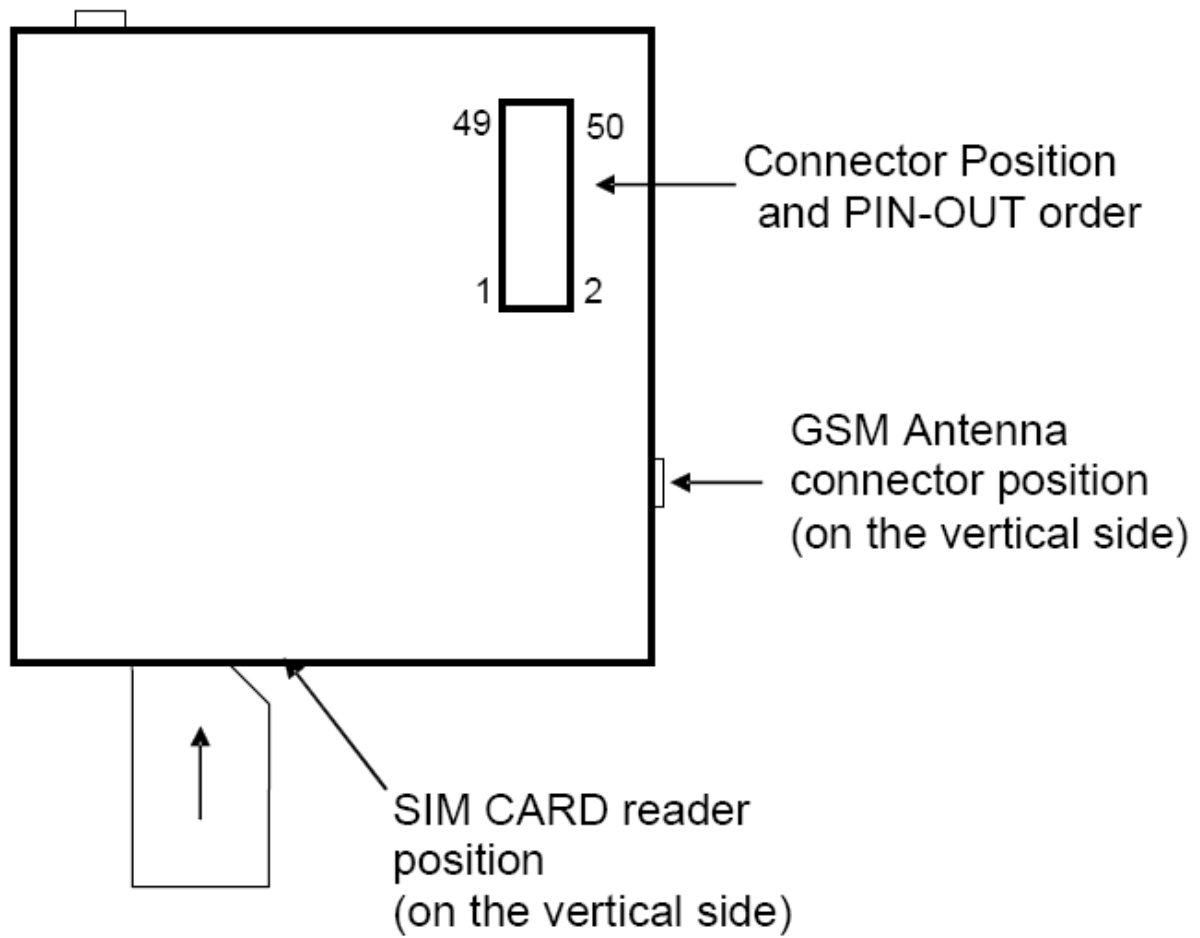


Figure12. Schéma des connexions du GM862.

c Structure interne du module GM 862 :

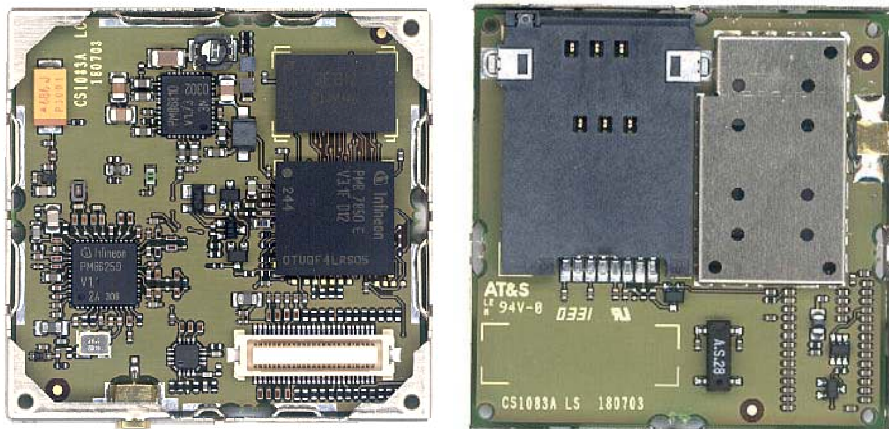


Figure13. Architecture du GM 862.

Le GM862 comme le MC45 est composé de deux sections : section RF (Radio Fréquence) et une section de bande de base sauf que pour le GM862 l'interface de la carte SIM est intégrée.

2.4 Les Commandes AT :

La plupart des équipements modernes disposent de modems qui sont à l'écoute de certaines commandes textuelles qui servent à donner aux modems des instructions à exécuter. Les modèles les plus anciens n'obéissent qu'à des instructions par signaux de contrôle sous la forme de télégramme.

Les commandes les plus courantes sont les commandes AT ou encore "Hayes" du nom de l'entreprise qui a commercialisé les premiers modems dits intelligents. Ces commandes sont simples et sont préfixées par "AT". Ces deux codes sont utilisés pour synchroniser un modem à la vitesse adéquate. Ces commandes agissent sur des registres internes qui sont directement manipulables, et il est possible de sauvegarder la configuration dans une mémoire interne non volatile. On parlera cependant des plus courantes car il n'existe pas de norme proprement dite pour ces commandes, en général les constructeurs fournissent de la documentation suffisante pour le paramétrage.

AT : attention command

c'est le préfixe de ligne de commande qui indique au modem qu'une commande ou une séquence de commandes est entrée.

2.5 Fonctionnement :

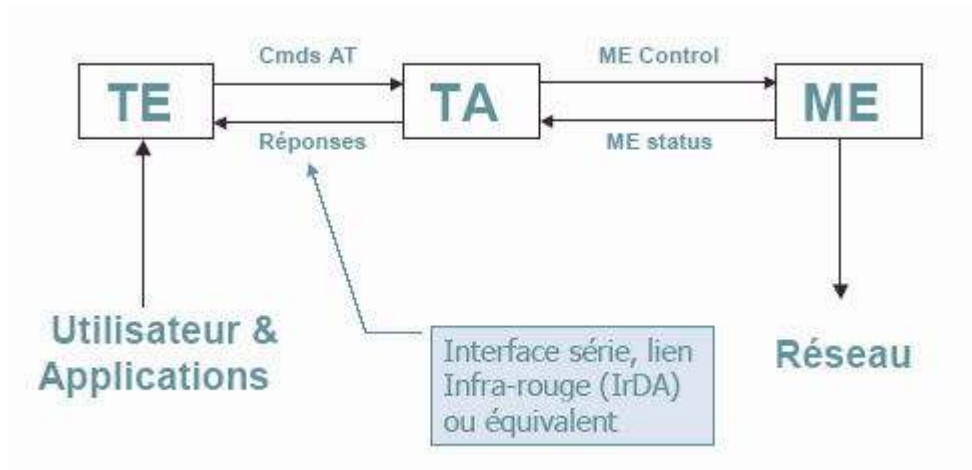


Figure14. Schema de fonctionnement.

- **ME :** (Mobil Equipment) Téléphone Portable.
- **TE :** (Terminal Equipment) ça peut être un ordinateur ou un microcontrôleur.
- **TA :** (Terminal Adaptor) Assure la liaison entre le "ME" et "TE".
- **TA et ME** forment une entité, par exemple un téléphone portable standard ou un terminal GSM contient dans son boîtier à la fois le "TA" et le "ME".
- Le **TE** forme une entité à part, par exemple il peut s'agir d'un ordinateur qui dispose d'un port série ou d'un circuit électronique basé sur microcontrôleur qui implante un port série.

2.6 Commandes dédiées au service SMS :

AT+CSMS	Sélection du service de messagerie
AT+CPMS	Sélection de ta zone mémoire pour le stockage des SMS
AT+CMGF	Sélection du format du SMS (PDU ou TEXT)
AT+CSCA	Définition de l'adresse du centre de messagerie

AT+CSDH	Affiche en mode TEXT le paramétrage des SMS
AT+CSAS	Sauvegarde du paramétrage
AT+CRES	Restauration du paramétrage par défaut
AT+CNMI	Indication concernant un nouveau SMS
AT+CMGL	Liste les SMS stockés en mémoire
AT+CMGR	Lecture d'un SMS
AT+CMGS	Envoie un SMS
AT+CMSS	Envoie d'un SMS stocké en mémoire
AT+CMGW	Écriture d'un SMS.
AT+CMGD	Efface un SMS

Tableau3. Commandes dédiées au service SMS.

2.7 Conclusion :

Les modules GSM facilitent la transmission et la réception des données numérisées en utilisant les réseaux satisfaisants les conditions de la norme GSM. Ils sont utilisés dans divers domaines auxquels fait partie notre système, ses données seront véhiculées dans un port série pour la communication des commandes AT qui nous permettent de communiquer avec le modem.

3.1 Les Capteurs :

Un capteur est un appareil de transformation d'une grandeur physique en une autre utilisable telle une tension ou une intensité électrique ou encore la déviation d'une aiguille, la confusion est souvent présente lorsqu'on parle de capteur et transducteur. Le capteur est au minimum constitué d'un transducteur.

La différence entre un capteur et un instrument de mesure et que ce dernier est indépendant et se suffit à lui même pour afficher l'information concernant la grandeur physique visée, par contre le capteur n'est qu'une simple interface entre une information physique et une autre manipulable dépourvue d'une mémoire pour stocker les données. Néanmoins, les capteurs restent des éléments de base dans les systèmes d'acquisition des données.

3.2 Classification des capteurs :

3.2.1 Apport énergétique :

a Capteur passif :

Dans la majorité des cas, pour fonctionner ces capteurs ont besoin d'énergie extérieure (jauge d'extensomètre, potentiomètre, photorésistance). Ce sont des capteurs modélisables par une impédance, une variation de la grandeur physique mesurée implique une variation de cette impédance. Il reste entendu que l'application d'une tension à l'entrée est nécessaire pour obtenir une sortie.

b Capteur Actif (Direct) :

On parle de capteurs actifs lorsque le phénomène physique utilisé pour la détermination du mesurande effectue directement la transformation en grandeur électrique, c'est une loi physique qui relie mesurande et grandeur électrique de sortie.

Un capteur actif fonctionne assez souvent en électromoteur et dans ce cas la sortie est une différence de potentiel.

Les lois physiques qui permettent une telle transformation sont limitées, on peut alors facilement recenser les capteurs actifs. Toutefois, les domaines d'application de ces capteurs sont très étendues : Température, effet thermoélectrique ou effet Seebeck.

3.2.2 Types de sorties :

Une classification par type de sortie est aussi possible pour les capteurs :

a Capteur Analogique :

La sortie est une grandeur électrique dont la valeur est proportionnelle à celle de la grandeur physique mesurée par le capteur. La sortie peut prendre une infinité de valeurs continues. Le signal analogique de sortie du capteur peut être :

- une tension.
- un courant.
- une règle graduée.
- une jauge (avec aiguille ou liquide).

b Capteur Numérique :

La sortie est une séquence d'états distincts. Elle peut prendre une infinité de valeurs discrètes. Le signal de sortie du capteur numérique peut être du type :

- code numérique binaire.
- train d'impulsions avec un nombre ou une fréquence précise.

c Capteur Logique :

Les capteurs, TOR possèdent une sortie qui ne peut prendre que deux états logiques notés 1 ou 0 et le signal de sortie est du type :

- courant présent/absent dans un circuit électrique.
- potentiel souvent 5V/0V.

- LED allumée ou éteinte.

3.2.3 Type de détection :

La détection peut être faite de deux manières différentes :

- détection avec contact où le capteur doit physiquement être en contact avec un phénomène pour le détecter.
- détection à distance où le capteur détecte le phénomène seulement à proximité.

3.3 Principales caractéristiques d'un capteur :

Les caractéristiques du capteurs sont :

- La gamme de mesure.
- La grandeur physique observée.
- La sensibilité.
- La résolution.
- La précision.
- La reproductibilité.
- La linéarité.
- Le temps de réponse.
- La gamme de température d'utilisation.

3.4 Le capteur Dallas 1621 : [4]

Le DS1621 Thermomètre numérique et thermostat fournit des mesures de température de 9 bits indiquant la température de l'appareil. La sortie d'alarme

thermique, T_{out} , est active lorsque la température de l'appareil dépasse un seuil défini par l'utilisateur de la température T_H . La sortie reste active jusqu'à ce que la température redescend en dessous de la température prédéfinie T_L par l'utilisateur. Les réglages de

température sont tous communiqués de ou vers le DS1621 via une interface série simple à 2 fils. Le DS1621 est représenté dans la figure ci dessous.

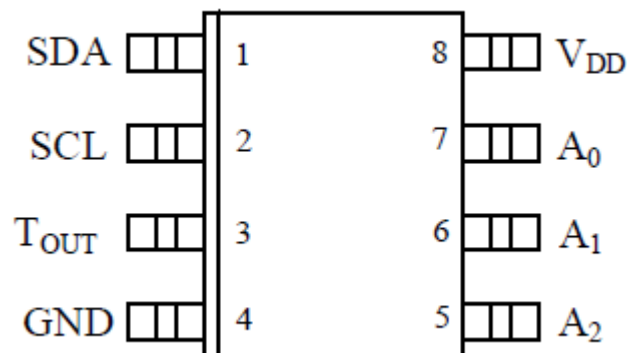


Figure15. Brochage du DS1621.

3.4.1 Caractéristiques du capteur DS1621 :

- La mesure de la température ne nécessite aucun composant externe.
- Bande de mesure de la température allant de -55°C à 125°C avec une précision de 0.5°C . L'équivalent en Fahrenheit est de -67°F à 257°F avec une précision de 0.9°F .
- La température est lue en 9bits.
- Rang de l'alimentation allant de 2.7V à 5.5V .
- Conversion de la température en une valeur digitale en moins d'une seconde.
- Les paramètres thermostatiques sont définies par l'utilisateur.
- Les données sont écrites ou lues via un bus d'un interface série à deux fils.

3.4.2 Description détaillée des PIN du DS1621 :

Cette description est donnée par le tableau suivant :

PIN	SYMBOLE	DESCRIPTION
1	SDA	PIN d'entrée/sortie des données pour la communication série à deux fils
2	SCL	PIN d'entrée/sortie Horloge pour la communication série à deux fils
3	T _{out}	PIN sortie du thermostat. Actif quand la température dépasse TH et Reset quand la température va en dessous de TL
4	GND	Masse
5	A2	PIN d'adresse
6	A1	PIN d'adresse
7	A0	PIN d'adresse
8	V _{DD}	PIN d'alimentation 2.7V à 5.5V

Tableau4. Description des PIN du DS1621.

3.4.3 Diagramme de fonctionnement du DS1621 :

Son fonctionnement est décrit par le tableau suivant :

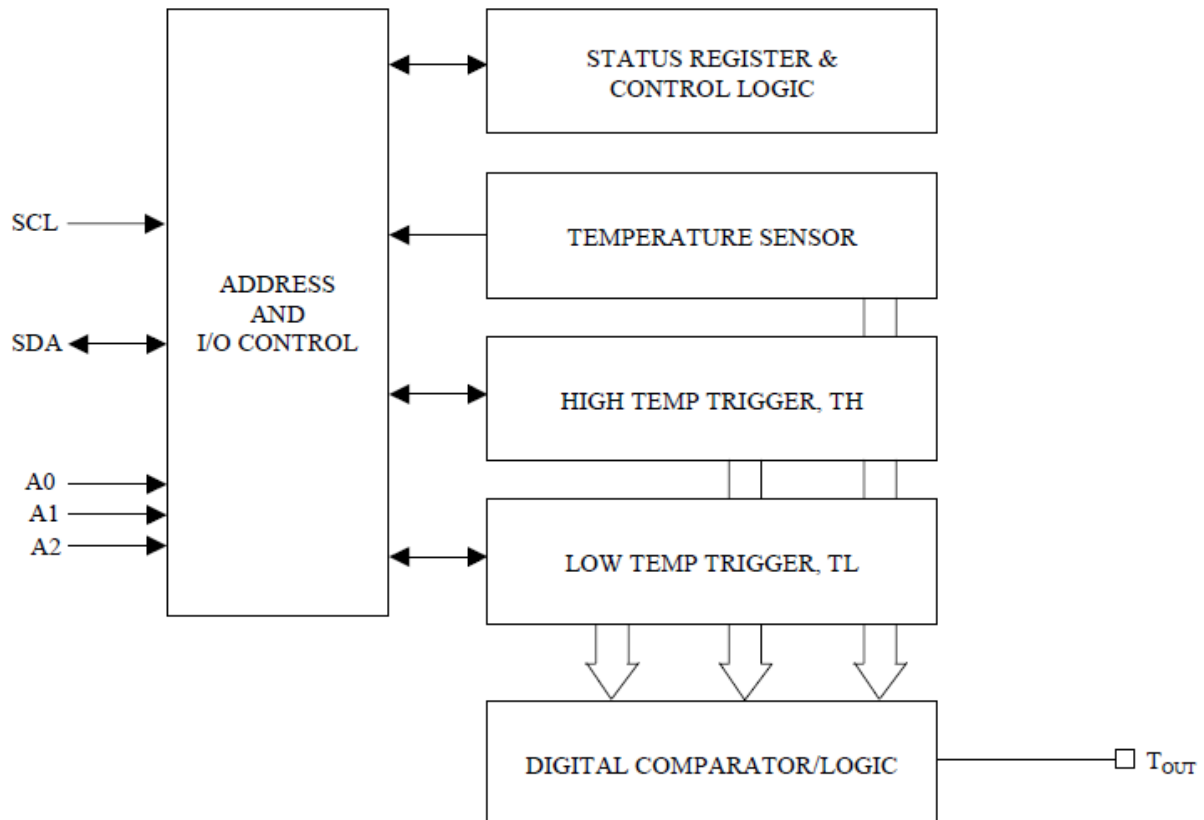


Figure16. Diagramme de fonctionnement du DS1621.

3.5 Le bus I²C :

Le bus I²C est utilisé par tous les périphériques électroniques équipés d'un microcontrôleur ou un microprocesseur parce qu'il est simple à intégrer aux designs les plus complexes. C'est une liaison série qui peut relier un ou plusieurs périphériques à la fois et il a été créé pour permettre la communication entre circuits intégrés se trouvant sur une même plateforme.

L'I²C ou l'Inter Integrated Circuit a été mise au point par la société PHILIPS pour relier facilement des systèmes électroniques prévues pour une utilisation du public et s'est

répandu avec le temps pour devenir une star de la technologie moderne. Il permet la communication avec seulement deux fils en plus de ceux de l'alimentation, un fil pour les données SDA et un autre pour l'horloge SCL. Ceci permet de créer des systèmes avec de très puissantes fonctionnalités.

Il est très rare aujourd'hui de ne pas trouver un appareil qui ne soit pas équipé de son microprocesseur ou de son microcontrôleur. Si ces derniers devaient communiquer par un bus parallèle il faudra un grand nombre de pin et de liaisons qui occuperont une surface importante sur le circuit imprimé, en utilisant le bus I²C le nombre de liaisons à prévoir sur le circuit se trouve nettement réduit de même que la surface occupée sur le circuit imprimé et même le prix de fabrication.

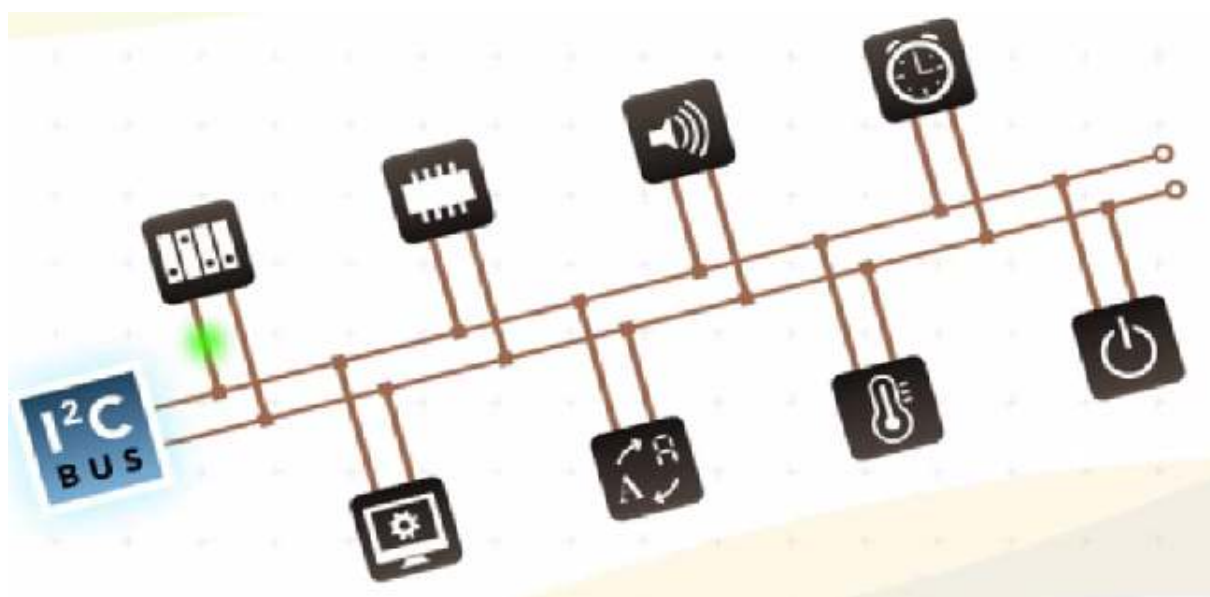


Figure17. Structure du bus I²C.

3.6 Caractéristique du bus I²C :

Le bus I²C présente des similitudes au SMBUS de Intel (system management bus). L'I²C a débuté avec une vitesse maximale de 100KBits/seconde qui est rapidement porté à 400KBits/seconde. Il est constitué de deux lignes bidirectionnelles :

- La ligne SDA (Serial Data Line) qui véhicule les données (bits), gérée à un moment donné par celui qui envoie l'information qu'il soit maître ou esclave.
- La ligne SCL (Serial Clock Line) véhicule l'horloge de synchronisation, gérée par le maître.
- Tous les circuits sont connectés sur ces deux lignes.

En précisant que les circuits maîtres sont ceux qui dirigent le transfert des données et qui pilotent l'horloge alors que les circuits esclaves sont ceux qui subissent l'horloge et répondent aux ordres des maîtres.

3.7 Le protocole I²C :

Comme nous l'avons laissé entrevoir l'I²C procède de nombreuses qualités mais compte tenu du peu de lignes qu'il dispose cela pose le vrai premier problème : comment gérer le bus I²C ?

En effet, afin d'éviter de faire des fautes de compréhension il est nécessaire de bien étudier et analyser le protocole qui régit le bus.

3.7.1 Terminologie d'un transfert sur le bus I²C : [5]

Les notions suivantes sont utilisées :

- F (Free) Libre : le bus est libre; la ligne SDA et SCL sont toutes les deux à l'état haut.
- S (START) Début : le transfert de données commence avec une condition de Start (ne pas confondre avec le bit Start) qui fait que le niveau de la ligne SDA change de haut en bas tandis que la ligne SCL reste à l'état haut.

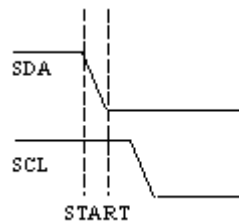


Figure18. Condition START.

- C (CHANGE) Changement : pendant que la ligne SCL est à l'état bas, le bit de donnée à transmettre peut être appliqué à la ligne de SDA et il peut changer de niveau.
- P (STOP) Arrêt : un transfert de données se conclut par une condition d'arrêt ceci se produit lorsque la ligne SDA passe de l'état bas à l'état haut tandis que la ligne SCL reste à l'état haut. Le bus est à nouveau dit libre.

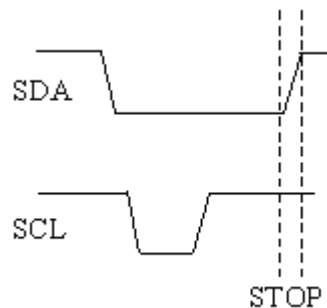


Figure19. Condition STOP.

- ACK (ACKNOWLEDGEMENT) Acquiescement : C'est en effet un accusé de réception de l'octet envoyé. C'est donc le récepteur qui émet ce bit pour signaler qu'il a bien lu l'octet envoyé par l'émetteur. Cet accusé de réception est lu comme un bit classique. Il vaudra 0 si OK et 1 pour une autre raison (récepteur dans l'impossibilité de répondre, par exemple). Pour cela et pendant que la ligne SCL est au niveau bas le maître place sa sortie au niveau haut et génère un cycle d'horloge supplémentaire pendant que l'esclave place sa sortie au niveau bas.

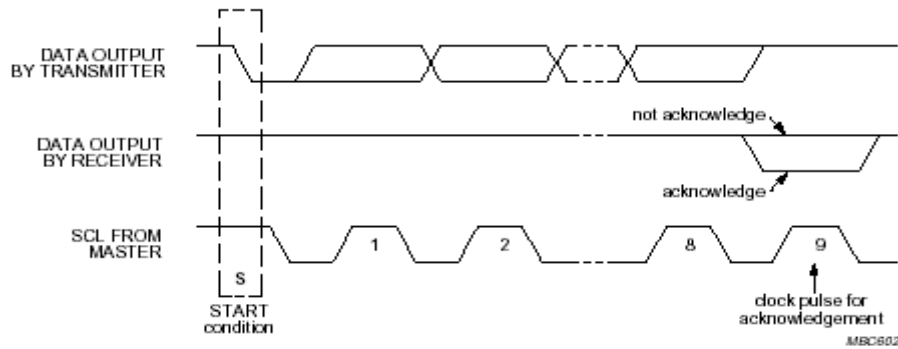


Figure20. Acquittement.

3.7.2 La notion d'adresse :

Un avantage très intéressant offert par le bus I²C réside dans la possibilité de connecter plusieurs composants en même temps, mais il ne dispose d'aucune ligne de sélection, alors que faut-il faire pour choisir lequel utiliser? C'est alors que la notion d'adresse intervient. La première norme I²C limité la taille des adresses à 7Bits, cependant, au fil de son évolution a vue apparaitre la possibilité d'adresser sur 10bits.

Le principe est très simple : le maitre envoie une adresse avant de transmettre et seul l'esclave dont l'adresse correspond à celle envoyée va répondre. Toutes les adresses ne sont pas permise, certaines sont utilisées pour des commandes spécifiques. Parmi celles-ci nous citons :

- '0000000' B : Adresser simultanément tout les périphériques (General Call Adresse).
- '0000001' B : Accès aux composants CBUS.
- '00001XX' B : Accès aux composants à haute vitesse.
- '11110XY' B : Précision d'une adresse à 10Bits.

3.7.3 Adressage des composant I²C :

Chaque circuit intégré susceptible d'être appelé par un maitre doit disposer d'une adresse unique qui le désigne des autres circuits. Bien entendu les composants qui n'ont que la fonction de maitre n'ont pas besoin d'une adresse.

3.7.4 Transfert d'une donnée sur le bus :

Comme dans tout protocole structuré le transfert des informations est spécifique au protocole et il comprend :

- Des définitions pour le niveau électrique haut et bas.
- Des conditions de fonctionnement.
- Des conditions de changement d'état.
- Des conditions de validité des données.
- Des conditions de départ et d'arrêt.

- Des formats de mots.
- Des formats de transmission.
- Des procédures de synchronisation.
- Des procédures d'arbitrage.

3.8 Conclusion :

Dans ce chapitre on vu de manière générale les capteurs qui nous permettent de recueillir une information à l'état physique et nous donne un signal électrique à la sortie que nous pouvons manipuler. On a aussi eu un petit aperçu concernant l'un des protocoles qui nous permettent de communiquer avec ces capteurs et les exploiter avec un microcontrôleur.

4.1 Présentation Générale :

Commercialisés pour la première fois depuis quelque années, les microcontrôleurs PIC auraient pu n'avoir qu'un succès mitigé car ils arrivaient sur un marché déjà bien encombré par des produits déjà remarquables tel que « MOTOROLA » ou bien « INTEL ». Malgré cela, le succès de « Microchip » fut immédiat et fulgurant et on peut affirmer qu'aujourd'hui ces circuits sont devenus de véritables références dans le monde des microcontrôleurs. Ce succès ils le doivent entre autres choses à deux caractéristiques particulières de leurs architectures internes dont nous allons maintenant en parler en guise d'introduction.

4.2 Harvard contre Von Neumann :

LA majorité des microcontrôleurs actuels utilisent une architecture interne dite de « Von Neumann », c'est-à-dire en fait une architecture semblable à celle intégrée dans les microordinateurs. La mémoire, appelée improprement « de programme », contient en fait des instructions et des données placées les unes à la suite des autres et l'on ne dispose que d'un bus, appelé bus de données, pour véhiculer à tour de rôle les codes des instructions et les données qui leurs sont associées, comme le montre la figure(21).

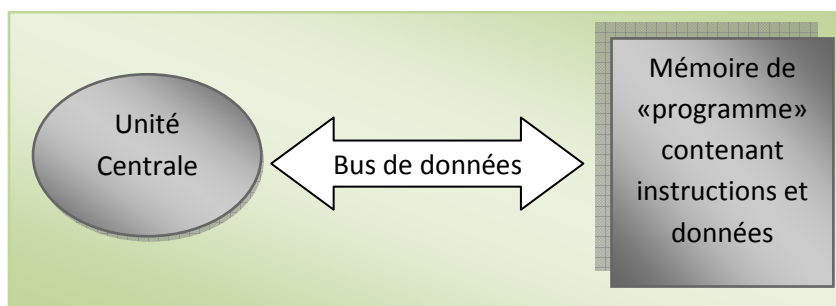


Figure21. Principe de l'architecture Von Neumann.

Si cette architecture donne entière satisfaction chacun en à la preuve chaque jour, elle pose cependant quelques problèmes dès que l'on veut faire fonctionner l'ensemble rapidement. En effet, l'exécution d'une seule instruction nécessite plusieurs échanges de données sur le seul et unique bus consacré à cet usage puisqu'il faut tout d'abord aller chercher le code de l'instruction, puis le ou les données qu'elle doit manipuler.

Il est alors préférable de faire appel à une architecture dite « Harvard » dans laquelle les instructions et les données sont clairement différentes et sont véhiculées sur des bus différents comme le montre la figure(22).

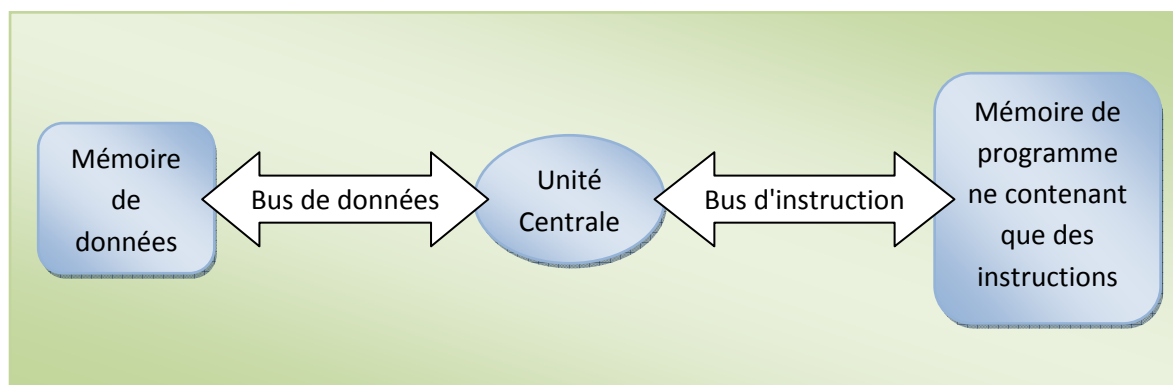


Figure22. Principe de l'architecture Harvard.

Vu de l'utilisateur, cela ne change rien bien sûr et des circuits de ce type s'utilisent exactement de la même façon que les autres. Par contre, les résultats obtenus, en terme de vitesse d'exécution des programmes, peuvent être impressionnants.

En effet, l'exécution d'une instruction ne fait plus appel qu'à un seul cycle machine puisque l'on peut simultanément, grâce aux deux bus, rechercher le code de l'instruction et la ou les données qu'elle manipule.

Rompant avec une tradition bien établie, les microcontrôleurs PIC de Microship utilisent une architecture Harvard et ce n'est pas tout. Ils font également appel à une architecture de type RISC qui, comme on le lit un peu trop souvent, ne se résume pas simplement à disposer d'un jeu d'instructions réduit.

4.3 Les points forts de l'architecture RISC :

"RISC" signifie « Reduced instruction set computer », ce qui veut effectivement dire circuit à jeu d'instructions réduit mais ne se limite pas uniquement à ça. Un vrai circuit de type RISC doit en effet présenter un certain nombre de particularités propres à accroître sa vitesse de fonctionnement.

Les microcontrôleurs à architecture RISC utilisent des instructions codées sur un seul mot de 12 ou 14 bits de long dans le cas des PIC ce qui présente deux avantages. Le premier est que tous les emplacements de la mémoire de programme contiennent une instruction, le second est qu'un seul cycle machine suffit pour lire le code complet d'une instruction, d'où un gain en vitesse d'exécution.

Les circuits RISC utilisent ensuite une structure de type pipe-line qui leur permet d'exécuter une instruction tout en recherchant la suivante en mémoire d'où, là encore, accroissement de la vitesse d'exécution.

Les instructions de type RISC sont simplifiées considérablement de telle façon que toutes les instructions peuvent être exécutées sur tous les registres avec tous les modes d'adressage. Cela simplifie le travail du programmeur, et donc, le notre, car il n'est plus nécessaire de retenir une multitude de cas particuliers d'instructions qui ne fonctionnent que dans un mode ou dans un autre.

Enfin, et c'est le plus important que l'on retient à propos de l'architecture RISC, on n'a pas besoin d'apprendre une centaine d'instructions différentes pour la programmation comme avec la plu part de microcontrôleurs mais seulement "33" ou "35" dans le cas des PIC qu'on découvrira dans la suite de ce mémoire.

Mais tout d'abord que contiennent ces microcontrôleur PIC..?

4.4 Contenu du microcontrôleur :

Un microcontrôleur est un composant électronique doté d'un :

- Microprocesseur.
- Mémoire RAM.
- Mémoire permanant.
- Les interfaces d'entrée sortie parallèles et série (RS232, I2C, SPI.....)
- Les interfaces d'entrée sortie analogiques.
- des " Timers "ou Horloges pour gérer le temps.

- d'autres modules plus ou moins sophistiqués selon la taille du microcontrôleur.

Un microcontrôleur est généralement moins puissant qu'un microprocesseur en terme de rapidité ou de taille mémoire, il se contente le plus souvent d'un bus de 8 à 16 bits. Ceci en fait un composant très bon marché et parfaitement adapté pour piloter des applications dans de nombreux domaines de la vie courante. Je pense qu'on ne se tromperai pas beaucoup si on dit qu'aujourd'hui il y a un microcontrôleur plus ou moins grand dans chaque équipement électronique.

- informatique (souris, modem.....).
- vision (appareils photo et caméras numériques).
- contrôle des processus industriels (régulation, pilotage).
- appareils de mesure (affichage, calcul statistique, mémorisation).
- automobile (ABS, injection, airbag).
- multimédia en général (télévision, carte audio ou vidéo).
- la téléphonie mobile ou le FAX.
- l'électroménager (four micro-onde, lave-vaisselle, lave-linge).

Les microcontrôleurs peuvent être programmés une fois pour toute afin qu'ils effectuent une ou plusieurs tâches au sein d'un appareil électronique. Bien que, les microcontrôleurs actuels peuvent être reprogrammés grâce à leur mémoire permanente de type "Flash".

Plusieurs constructeurs se partagent le marché des microcontrôleurs tel que Intel, MOTOROLA, AMTEL, ZILOG, PHILIPS et en fin Microship qui nous intéresse dans ce mémoire.

4.5 La jungle des références :

Avant d'entamer la présentation des différentes familles des PIC, attardons nous un instant sur les références des circuits qui à première vue nous renseignent sur un certain nombre de paramètres importants

Une référence de microcontrôleur Microchip est toujours de la forme FF-MMM-XXX où :

- FF désigne la famille dont fait partie le circuit et peut valoir :12, 16, 17 ou 18.
- MMM peut être une suite de un, deux ou trois lettres qui indique tout à la fois le type de mémoire de programme contenu dans le circuit et si on est en présence d'un circuit à la plage de tension d'alimentation normale ou étendue.
- XXX est un ensemble de deux ou trois chiffres qui représente la référence du circuit dans la famille en fournissant une idée sur les ressources internes du circuit.

Un suffixe peut également suivre la référence, bien qu'un désordre semble régner chez Microchip puisque ces suffixes évoluent au fil du temps on verra quand on présentera l'horloge, ils précisent quel type d'horloge peut être utilisé avec le circuit et surtout quelle est la fréquence maximale de cette horloge.

4.6 Les différentes familles des PIC :

L'avantage des microcontrôleurs à architecture RISC ou à jeu d'instruction réduit est que plus on réduit le nombre d'instructions et plus le décodage sera rapide ce qui augmente la vitesse de fonctionnement du microcontrôleur.

Les PICs sont réparties en trois grandes familles : La famille des « base-line » qui utilisent des mots de 12 bits, La famille des « Mid-Range », qui utilisent les mots de 14 bits (et dont font partie le 16F876 et 16F877) et enfin la famille « High-End », qui utilisent les mots de 16 bits.

Les PICs sont des composants statiques, ils peuvent fonctionner avec des fréquences d'horloge allant du continu jusqu'à une fréquence maximum spécifique à chaque circuit. Un PIC 16F876-4 peut fonctionner avec une horloge allant jusqu'à 4MHz.

Nous nous limiterons dans ce mémoire à la famille des Mid-Range et particulièrement au PIC 16F876 sachant que si on à tout assimilé, on pourra facilement passer à une autre famille ou un autre microcontrôleur.

PIC	Flash	RAM	EEPROM	I/O	A/D	Port //	Port série
16F870	2K	128	64	22	5	NON	USART
16F871	2K	128	64	33	8	PSP	USART
16F872	2K	128	64	22	5	NON	MSSP
16F873	2K	192	128	22	5	NON	USART/MSSP
16F874	2K	192	128	33	8	PSP	USART/MSSP
16F876	8K	368	256	22	5	NON	USART/MSSP
16F877	8K	368	256	33	8	PSP	USART/MSSP

Tableau5. Différents circuit de la famille 16F87X.

4.7 Les éléments essentiels du PIC 16F876 :

Le PIC 16F876 contient essentiellement :

- Une mémoire programme EEPROM flash 8K mots de 14 bits.
- Une RAM donnée de 368 octets.
- Une mémoire EEPROM 256 octets.
- Trois port d'entrée sortie, A (6 bits), B (8bits), C (8bits).
- Un convertisseur analogique numérique 10bits à 5 canaux.
- USART, port série universel, mode asynchrone (RS232) et mode synchrone.
- SSP, port série synchrone supportant I2C.
- Trois TIMERS : TMR0, TMR1, TMR2.
- Deux modules de comparaison et de capture CCP1, CCP2.

- Un chien de garde.
- 13 sources d'interruption.
- Un générateur d'horloge, à quartz (jusqu'à 20MHZ) ou à oscillateur RC.
- Une protection du code.
- Fonctionnement en mode "sleep" pour la réduction de la consommation d'énergie.
- Une programmation par mode ICSP 12V ou 5V.
- Une possibilité d'accéder à la mémoire programme pour les applications utilisateurs.
- Tension de fonctionnement 2V à 5V.
- Jeu de 35 instructions.

4.8 Appellations normalisées des diverses pattes disponibles :

Les pattes dont on dispose sur le boîtier d'un microcontrôleur PIC varient évidemment en fonction des ressources internes de ce boîtier. Fort heureusement, chez Microship la notion de famille de microcontrôleurs n'est pas un vain mot et tous les circuits utilisent les mêmes désignations des pattes à ressources internes identiques.

La patte PSP2, si présente sur un boîtier, sera ainsi toujours la patte 2 du port parallèle esclave (PSP), quelque soit le type de circuit retenu. Cela va nous simplifier la prise de contact avec les différents circuits de la famille et nous évite d'avoir à se livrer à de fastidieuses conversations de noms.

Dans le but d'avoir plus d'idées des ressources qu'on va rencontrer et des noms de pattes qui leurs sont associer. On retrouve le tableau qui va suivre qui contient la liste complète des ces derniers et de leur appellation, ainsi qu'un résumé de leur fonction et de leurs types électriques. Les noms présents dans ce tableau sont ceux utilisés par Microship dans ces manuels d'utilisation et ces ouvrage techniques. ceux sont aussi ceux qu'on va retrouver tout au long de ce mémoire.

Appellation	Type logique	Type Electrique	Fonction
AN0 à AN15	E	Analogique	Entrées analogiques
AV _{DD}	Alimentation	Alimentation	Alimentation étage analogique
AV _{SS}	Masse	Masse	Masse étage analogique
C1	E	Analogique	Convertisseur à pompe de charge de l'interface LCD
C2	E	Analogique	Convertisseur à pompe de charge de l'interface LCD
CCP1	E/S	Trigger de Schmitt	E/S Module CCP1, PWM1
CCP2	E/S	Trigger de Schmitt	E/S Module CCP2, PWM2
CK	E/S	Trigger de Schmitt	Horloge synchrone SCI
CLKIN/OSC1	E	Trigger de Schmitt ou CMOS	Oscillateur d'horloge
CLKOUT/OSC2	S	-	Oscillateur d'horloge
CMPA	S	-	Comparateur analogique A
CMPB	S	-	Comparateur analogique B
COM0	S	Driver LCD	Electrode commune 0 afficheur LCD
COM1	S	Driver LCD	Electrode commune 1 afficheur LCD
COM2	S	Driver LCD	Electrode commune 2 afficheur LCD
COM3	S	Driver LCD	Electrode commune 3 afficheur LCD
DT	E/S	Trigger de Schmitt	Données synchrones SCI
GP0	E/S	TTL/Trigger de Schmitt	Ligne 0 port GPIO
GP1	E/S	TTL/ Trigger de Schmitt	Ligne 1 port GPIO
GP2	E/S	Trigger de Schmitt	Ligne 2 port GPIO
GP3	E	TTL	Ligne 3 port GPIO
GP4	E/S	TTL	Ligne 4 port GPIO
GP5	E/S	TTL	Ligne 5 port GP0
INT	E	Trigger de Schmitt	Interruption externe
MCLR/V_{pp}	E/Alimentation	Trigger de Schmitt	Entrée de Reset et de tension V _{pp} en mode programmation
PSP0 à PSP7	E/S	TTL	Ligne0 à 7 PSP
RA0 à RA3	E/S	TTL	Ligne 0 à 3 port A
RA4	E/S	Trigger de Schmitt	Ligne 4 port A
RA5	E/S	TTL	Ligne 5 port A
RB0 à RB7	E/S	TTL	Ligne 0 à 7 port B

RC0 à RC7	E/S	Trigger de Schmitt	Ligne 0 à 7 port C
RDO à RD7	E/S	Trigger de Schmitt	Ligne 0 à 7 port D
RE0 à RE7	E/S	Trigger de Schmitt	Ligne 0 à 7 port E
RF0 à RF7	E/S	Trigger de Schmitt	Ligne 0 à 7 port F
RG0 à RG7	E/S	Trigger de Schmitt	Ligne 0 à 7 port G ou sortie segment port LCD
\overline{CS}	E	TTL	\overline{CS} en mode PSP
\overline{RD}	E	TTL	Entrée lecture en mode PSP
\overline{WR}	E	TTL	Entrée écriture en mode PSP
REFA	S	CMOS	Sortie référence A
REFB	S	CMOS	Sortie référence B
RX	E	Trigger de Schmitt	Réception asynchrone SCI
TX	S	-	Emission asynchrone
SCL,SCLA,SCLB	E/S	Trigger de Schmitt	Horloge pour bus I2C
SDA, SDAA, SDAB	E/S	Trigger de Schmitt	Données pour bus I2C
SCK	E/S	Trigger de Schmitt	Horloge en mode SPI
SDI	E	Trigger de Schmitt	Entrée de données en mode SPI
SDO	S	-	Sortie de données en mode SPI
\overline{SS}	E	TTL	Entrée écriture en mode PSP
SEG00 à SEG31	S	LCD	Sortie segment port LCD
TOCKL	E	Trigger de Schmitt	Horloge externe timer 0
T1CKL	E	Trigger de Schmitt	Horloge externe timer 1
T1OSO	S	CMOS	Oscillateur timer 1
T1OSL	E	CMOS	Oscillateur timer 1
V_{LCD1} à V_{LCD3}	Alimentation	-	Alimentation afficheur LCD
VLCDADJ	E	Analogique	Réglage alimentation LCD
V_{REF}	E/S	Analogique	Entrée tension de référence convertisseur A/D, sortie tension de référence du générateur interne
V_{REF+}	E	Analogique	Entrée tension de référence haute convertisseur A/D
V_{REF-}	E	Analogique	Entrée tension de référence basse convertisseur A/D
V_{SS}	Alimentation	-	Masse du circuit
V_{DD}	Alimentation	-	Alimentation positive du circuit

Tableau6. Appellations, fonction, types électrique et logique de tous les pattes pouvant être rencontrées sur les différentes versions de PIC.[6]

4.10 Le cœur du PIC :

Un microcontrôleur est avant tout un microprocesseur, une unité de traitement logique qui effectue l'une après les autres les opérations contenues dans un

microprogramme stocké en mémoire (mémoire flash), on peut le voir dans le schéma ci dessous (assombri). Il est essentiellement constitué de l' « ALU » (unité arithmétique et logique) qui effectue les opérations sur les données, le registre de travail « W reg », le multiplexeur « MUX », le registre de statut « status reg », le registre « FSR reg » utilisé pour l'adressage indirect (en assembleur), le multiplexeur d'adresse « addr mux », le compteur programme « Program Counter » qui pointe les instructions à exécuter, la pile à 8 niveaux « 8 Level Stack », le registre d'instruction « Instruction reg » ainsi que tous les différents bus qui relient tous les éléments entre eux.

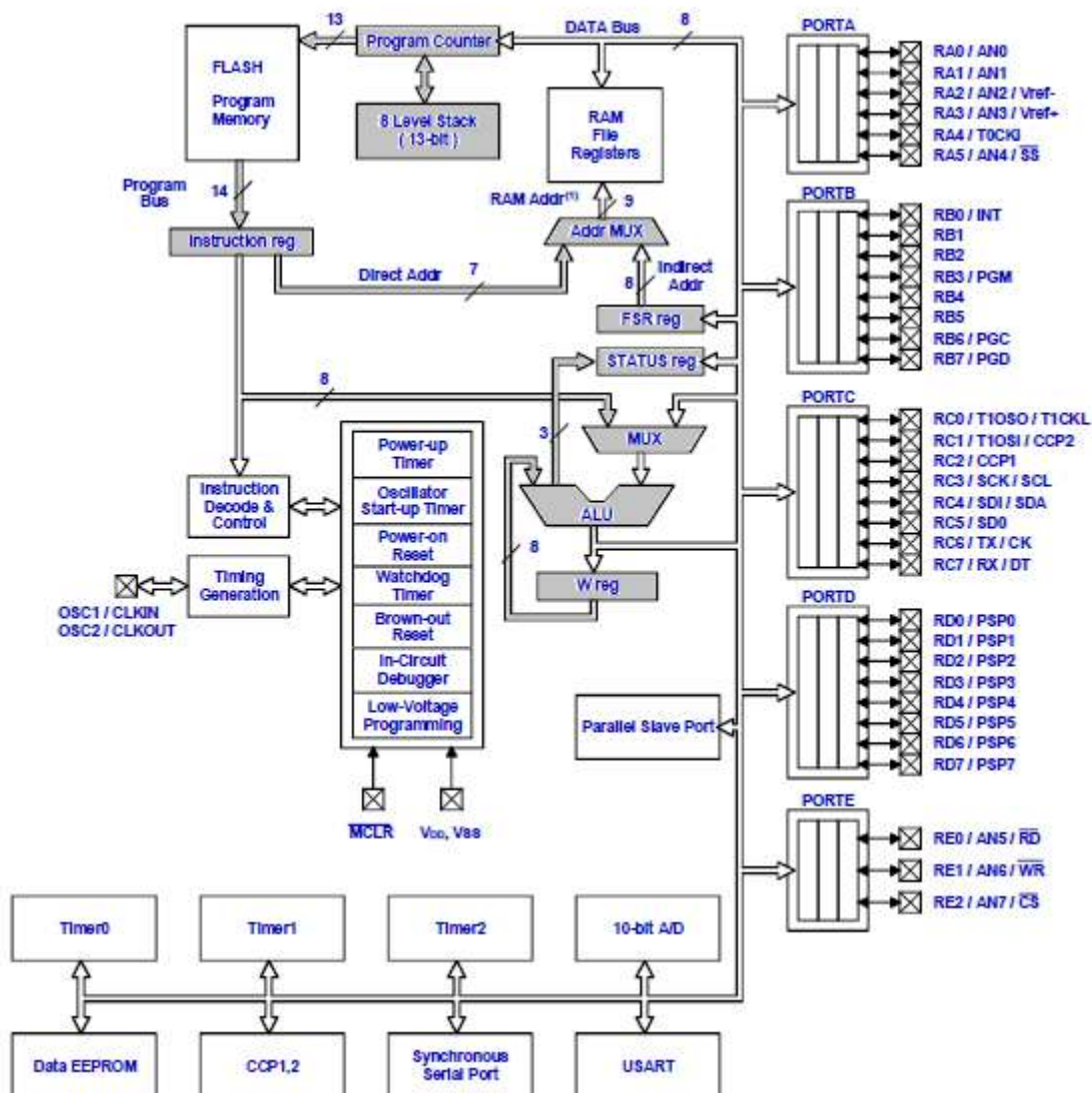


Figure 24. Cœur du PIC.

On ne détaillera pas le fonctionnement d'un microprocesseur mais c'est indispensable si on programme en assembleur. En ce qui nous concerne, on veut programmer en Langage C, et c'est donc le compilateur qui se chargera de traduire notre code source en instructions de bas niveau pour le microprocesseur contenu dans le PIC. C'est là le principal avantage de la programmation en C. On prend du recul par rapport au fonctionnement intime du système et on se concentre davantage sur « ce que fait le programme » que sur « comment fonctionne le programme ».

On va tout de même jeter un petit coup d'œil sur le schéma ci-dessus, histoire de comprendre quelques particularités du PIC. On s'aperçoit que les bus autour de l'ALU est à 8 bits les PIC 16F877/6 travaillent sur des données de 8 bits, c'est donc bien un microcontrôleur 8 bits. Pourquoi alors le « Program Bus » est-il lui, large de 14 bits? C'est simple, certaines instructions peuvent être codées sur plus de 8 bits si ce bus n'était large que de 8 bits, il faudrait tout simplement plus d'un cycle d'horloge pour transmettre ces instructions, alors qu'avec un bus plus large, ça passe en une fois. De plus, la mémoire programme, indépendante du bus de données, est elle-même adressée avec un bus large le « Program Counter », qui pointe sur l'instruction en cours, a une largeur de 13 bits. Et avec 13 bits on peut coder environ 8000 adresses. Autrement dit, on a 8000 cases de mémoire programme pouvant contenir chacune 1 instruction complète. Cette architecture avec bus de données et de programme séparés « Harvard » permet d'optimiser le fonctionnement du PIC. La plupart du temps, le PIC exécute une instruction et charge la suivante simultanément en un seul cycle d'horloge. En comparaison, un microcontrôleur 8 bits construit selon l'architecture concurrente « Von Neumann » devra faire plusieurs cycles pour chercher les instructions en mémoire et les exécuter ensuite. En conséquence, à fréquence d'horloge égale, un microprocesseur « Harvard » sera plus rapide qu'un « Von Neumann ».

4.11 La mémoire du PIC :

Dans le PIC, il n'y a pas une mémoire mais trois.

4.11.1 *La mémoire programme :*

Elles sont de type FLASH sur les 16F87X. Capacité : 8K. C'est dans celles-ci qu'est stocké le programme du PIC. Après compilation fichier « .hex » est créé, une suite de codes hexadécimaux. Celui-ci est transféré ensuite dans la mémoire programme du PIC à l'aide du programmeur. Cette mémoire n'est pas reliée au bus de données, sa vocation est de stocker le programme du PIC, mais pas les variables de votre programme. Le gros avantage de la mémoire FLASH c'est que vous pouvez la réécrire, donc implanter un nouveau programme dans le PIC. Les PIC existent également avec d'autres versions de

mémoire programme (non-FLASH), certaines ne pouvant être programmées qu'une seule fois.

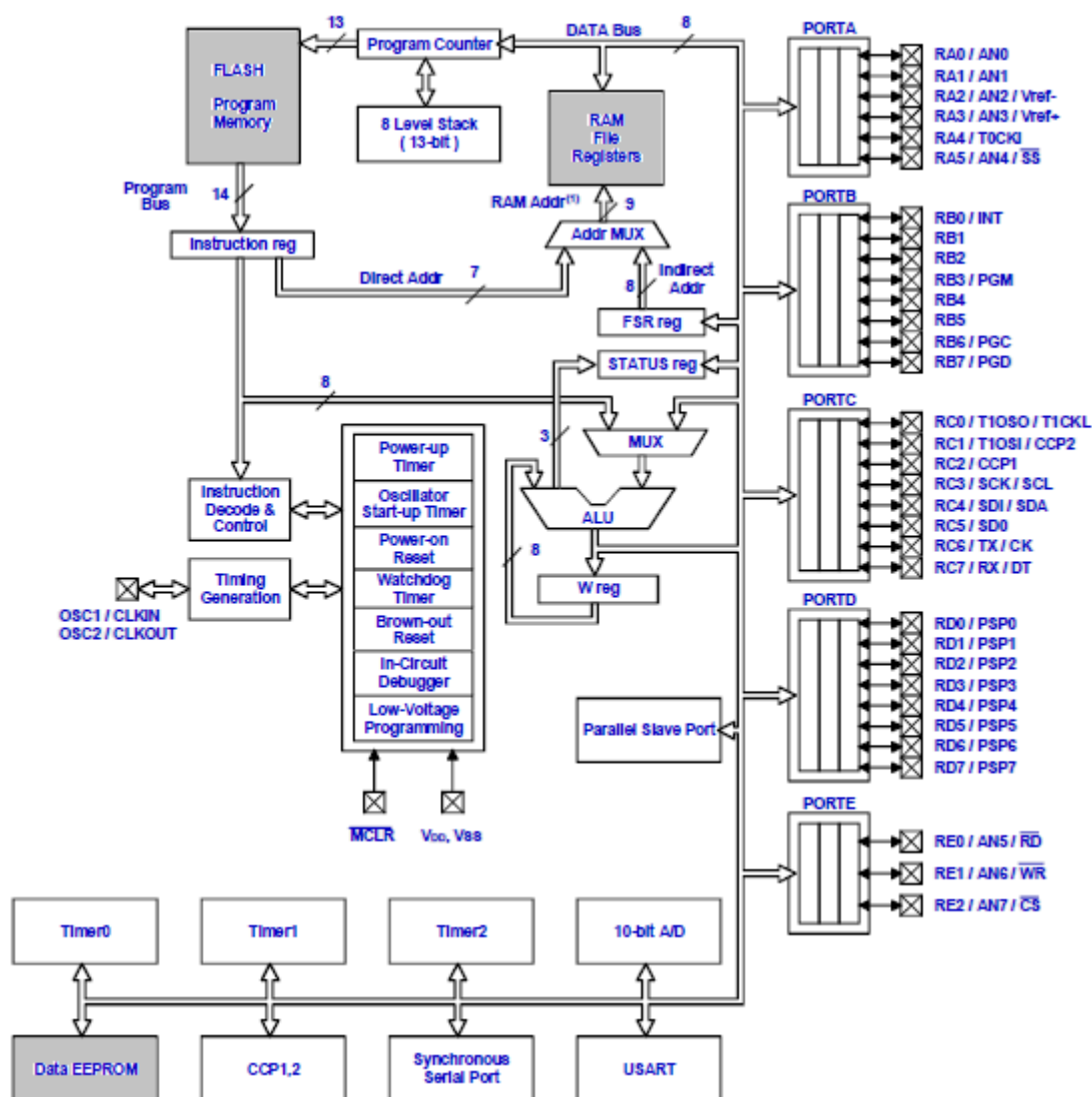


Figure25. Les mémoires du PIC 16F87X.

4.11.2 La mémoire RAM :

Cette mémoire fait partie de la zone d'adressage des données. Elle comprend tous les registres spéciaux permettant de contrôler le cœur du PIC ainsi que ses périphériques. Elle contient également des cases mémoires à usage générique dans lesquelles pourront être stockées les variables de nos futurs programmes.

4.11.3 *La mémoire EEPROM :*

L'EEPROM est plutôt une mémoire de stockage de données à long terme, alors que la RAM est utilisée pour les variables du programme. On a 256 octets d'EEPROM disponibles. Les mémoires de type EEPROM sont limitées en nombre de cycles d'effacement/écriture. Ce nombre de cycle est tout de même de l'ordre du million pour le PIC, mais si on l'utilisait pour stocker des variables modifiées plusieurs milliers de fois par secondes, cette limite pourrait être atteinte plus vite qu'on ne le croît. Par contre, pour stocker toutes les heures une mesure de température, c'est tout bon.

4.12 Les ports d'entrée / sortie généraux :

Le PIC 16F877 est généreusement doté de 5 ports tandis que le 16F876 ne dispose que de 3 ports, ce qui est plutôt confortable. Un examen plus attentif du schéma ci-dessus nous montre cependant que les lignes d'entrées/sorties correspondantes sont également utilisées par d'autres modules du PIC. Ainsi, les pattes du PORTA servent également au convertisseur Analogique/Numérique, les pattes du PORTD au Port Parallèle Esclave, etc. Il faudra faire des choix au moment de la conception du schéma électronique. On voit également que les ports B, C et D ont 8 lignes d'entrée/sortie, alors que le port A n'en a

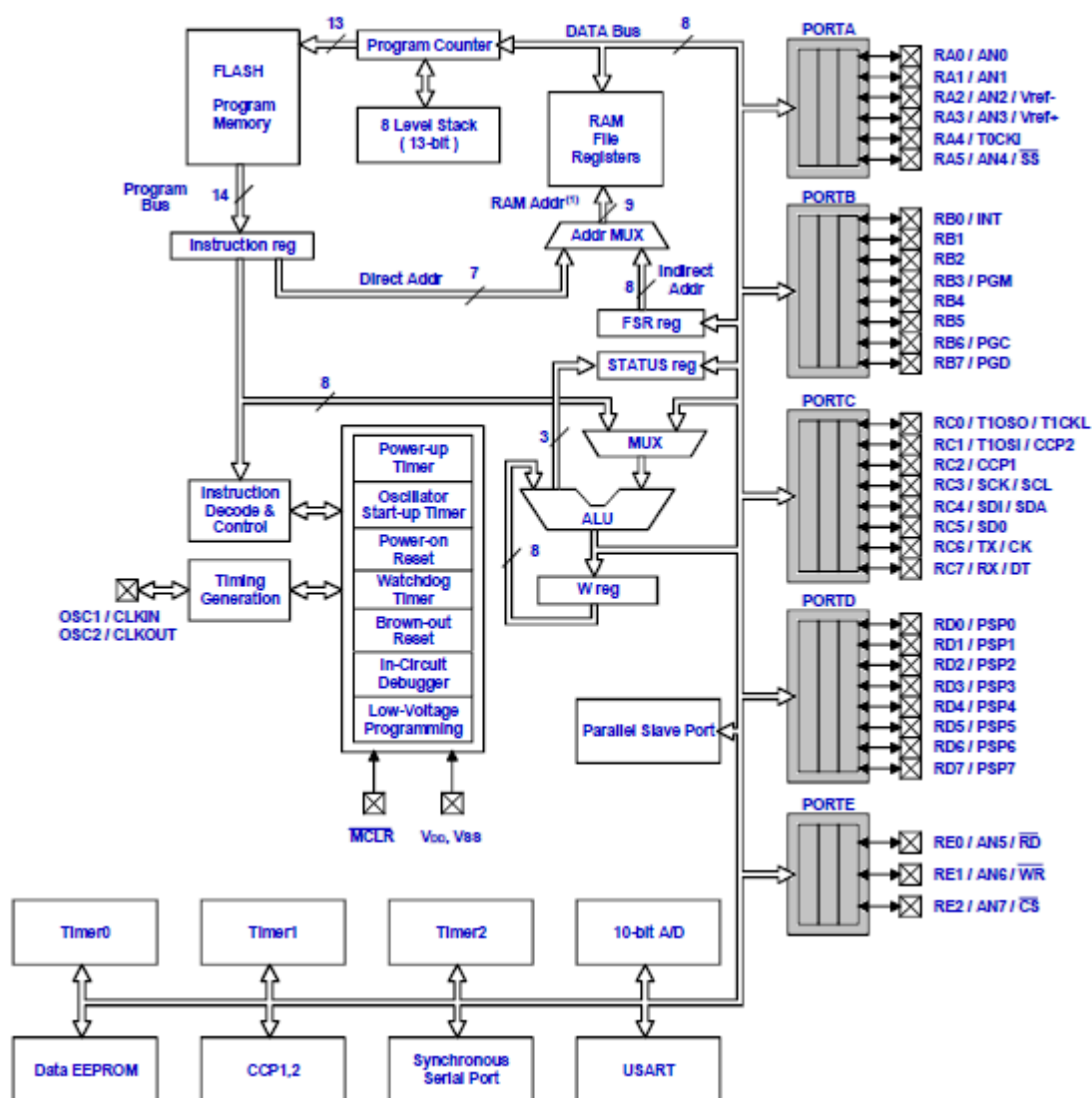


Figure26. : Les ports du PIC.

que 6 et le port E que 3. Ils sont tous connectés au bus de données, on pourra donc librement les adresser pour y lire ou écrire des données. On peut configurer les entrées/sorties de chaque port en entrée ou en sortie, grâce un registre spécial dédié à chaque port. De plus, un des ports (le port B) possède des résistances de « pull-up » internes.

4.13 L'USART

L' Universal Synchronous Asynchronous Receiver Transmitter est l'un des deux modules de communication série du PIC ou « SCI » en anglais (Serial Communication Interface). Comme son nom l'indique, elle peut établir une liaison synchrone ou asynchrone, recevoir et transmettre des données, selon la manière dont elle est configurée. C'est son côté « je peux tout faire » qui lui vaut l'attribut «Universal». Concrètement, l'USART permet de communiquer avec le reste du monde un ordinateur, téléphone ou autre matériel équipé d'une interface série RS232.

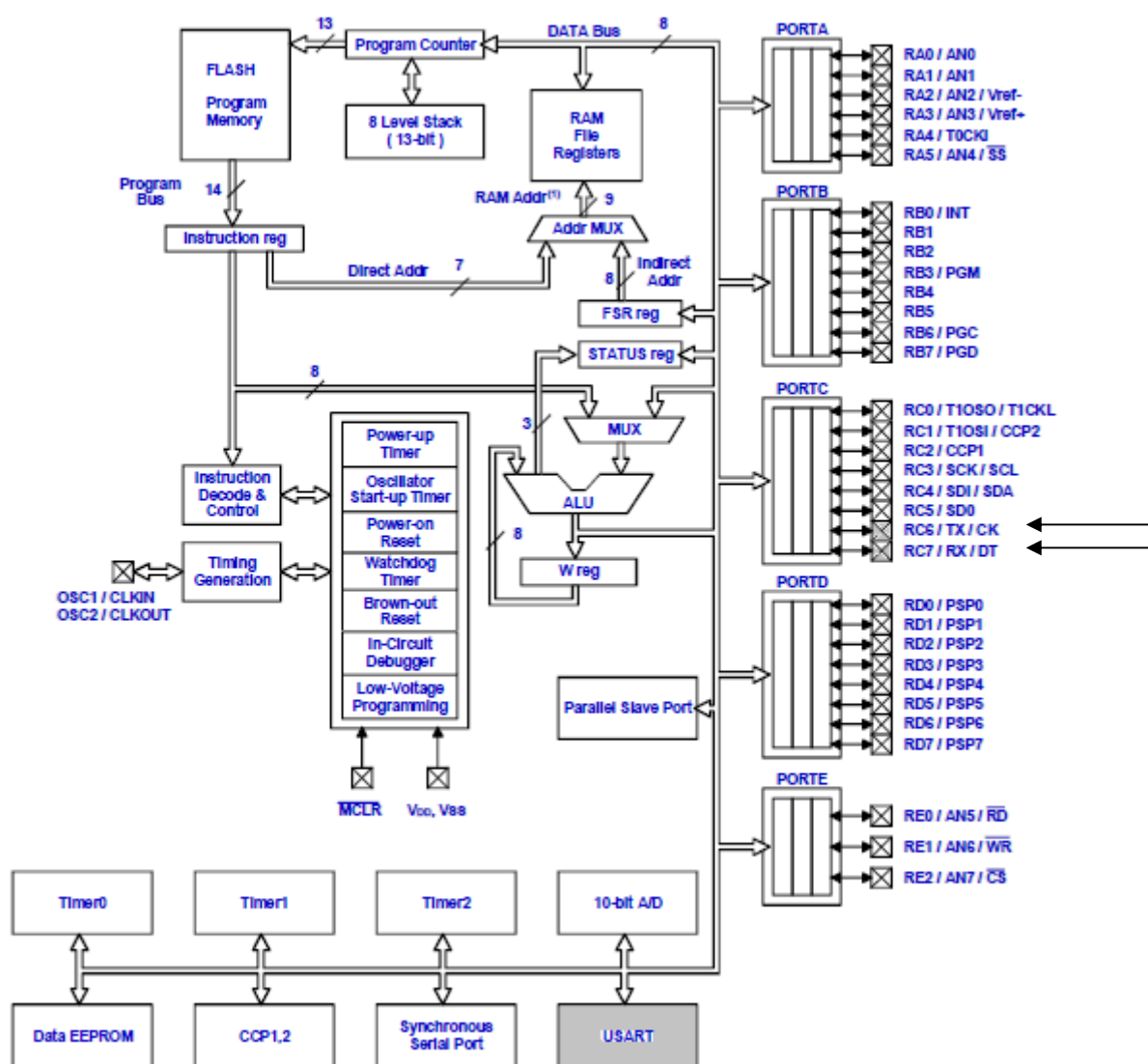


Figure 27. USART (Universal Synchronous Asynchronous Receiver Transmitter).

4.14 Le SSP :

Le « Synchronous Serial Port » est la deuxième interface de communication série du PIC, il s'agit en réalité d'un port Maître, donc l'appellation exacte est plutôt MSSP pour « Master Synchronous Serial Port ». Il est utile pour communiquer avec d'autres modules ou microcontrôleurs, des registres à décalage, des afficheurs ou des convertisseurs A/N.

Il peut fonctionner dans deux modes de communication :

- Le mode SPI (Serial Peripheral Interface).
- Le mode I²C (Inter-Integrated Circuit).

L'utilisation de l'USART ou le MSSP dépend donc essentiellement du protocole de communication nécessaire. Systématiquement, l'USART est bien adapté pour communiquer avec le reste de l'univers via la célèbre interface « RS232 » alors que le MSSP permet de communiquer aisément avec d'autres composants électroniques à interface série. On pourra aussi faire remarquer que par rapport aux interfaces de communication certains PIC intègrent une interface USB ou un module de communication Ethernet.

4.15 Pourquoi programmer en C?

Tout d'abord, il faut savoir ce qu'est un compilateur. C'est tout simplement un programme informatique qui transforme un code source écrit dans un langage de haut niveau facilement compréhensible par l'humain vers un langage de bas niveau pour l'assemblage afin qu'il puisse être exploité par la machine.

Un compilateur C produit un code plus volumineux et moins performant que ce que sait faire un bon programmeur en assembleur. De plus, selon le compilateur employé, on n'accède plus difficilement, voire pas du tout à certaines ressources de bas niveau. Et on risque également de moins approfondir l'étude du microcontrôleur utilisé. Un dernier point, les compilateurs C sont souvent payants. En contrepartie, il y a de sacrés avantages. Connaître la programmation en C permet de créer des programmes sur de multiples plateformes, à commencer par les PC. Autrement dit, les connaissances utilisées sont en grande partie réutilisables. Le C est un langage de « haut niveau », comparé à l'assembleur, qui permet d'écrire des programmes nettement plus intelligibles et donc plus faciles à relire, corriger ou modifier. Le compilateur contrôle la cohérence du code au moment de la compilation et signale bon nombre d'erreurs, qui seront faciles à corriger. Le compilateur prend en charge la gestion d'un certain nombre de mécanismes fastidieux, par exemple, pas besoin de spécifier la page mémoire dans laquelle on veut écrire, le compilateur s'en charge. De plus, certains compilateurs permettent tout de même d'accéder à des ressources de bas niveau, voir même d'insérer dans le code des portions d'assembleur. A vrai dire, à moins d'être un « pro » de l'assembleur, vous pourrez certainement créer avec un bon compilateur C un code plus propre et plus robuste, en nettement moins de temps et le portage d'un de vos programmes sur un nouveau microcontrôleur sera nettement simplifié.

Pour bien fixer les idées sur la différence de niveau de langage entre assembleur et C, on donne un petit exemple:

Soit à décrire une action simple : ouvrir la fenêtre de la pièce dans laquelle vous êtes actuellement.

En assembleur ça donnerait :

- soulever pied droit.
- avancer pied droit.
- poser pied droit.
- soulever pied gauche.
- avancer pied gauche.
- poser pied gauche.
-
- sélectionner bras gauche.
- soulever bras
- avancer bras.
- prendre poignée fenêtre dans main gauche.
- tourner poignée de 90°.
- tirer poignée.
- etc.....

En C ce serait plutôt :

- ouvrir la fenêtre.

Et c'est le compilateur qui se chargera de traduire la fonction « ouvrir la fenêtre » en instructions élémentaires compréhensible par le microcontrôleur.

4.16 Choix du compilateur :

Il y a sur le marché plusieurs compilateurs C. Le microcontrôleur étudié ici étant un PIC de « Microchip », la solution la plus évidente consiste à voir ce que propose Microchip qui nous propose bien des compilateurs, ceux-ci sont payants et ne couvrent que le haut de la gamme des PIC, ce qui est un peu luxueux pour un amateur désirant s'initier à l'utilisation de ces petites bêtes. D'autres éditeurs proposent également des

compilateurs dont certains sont très complets, livrés avec des bibliothèques simplifiant l'utilisation des périphériques du PIC. Malheureusement, ici encore c'est assez cher, pour un environnement de développement intégré couvrant l'essentiel de la gamme des PIC. Heureusement il existe des alternatives moins onéreuses. Par exemple, en cherchant bien sur internet on peut dénicher des compilateurs freeware.

Notre choix s'est porté sur le compilateur « MikroC Pro » un compilateur payant qui offre une simplicité et qui donne accès au code source des bibliothèques. Le langage est du C avec des extensions destinées à tirer pleinement parti de toutes les ressources des microcontrôleurs PIC. Avec bien sûr toute la documentation et aide nécessaire au site de la société « MikroElektronika »

4.17 Choix du PIC : Le 16F876

Pour apprendre, la meilleure solution est de se faire la main sur du concret, on va donc étudier un vrai microcontrôleur, sachant que ce qui aura été vu sera assez facilement transposable à d'autres PIC. Le 16F876 est un PIC de la série « Mid-range » qui se prête particulièrement bien à la programmation en C. Les PIC de la série inférieure sont un peu justes en performance et en capacité mémoire pour accueillir un programme issu d'un compilateur C, mieux vaut les programmer en assembleur. Les gammes supérieures (16 ou 32 bits) supportent sans problème la programmation en C. Le 16F876 convient parfaitement, mémoire programme de taille suffisante (8K), nombreux périphériques intégrés, fréquence de fonctionnement jusqu'à 20 MHz. Tout ce qu'il faut pour créer des applications intéressantes.

4.18 Programmation du PIC :

Comme on l'a mentionné précédemment le compilateur qu'on a choisi est le MikroC Pro de MikroElektronika.

4.19 Présentation du MikroC Pro :

Les Pics sont des composants qui ne sont capables de rien tant qu'on ne leur a pas insérer un programme. C'est comme un ordinateur sans système d'exploitation il ne démarre pas. La compilation du programme va générer un fichier .hex compréhensible par le PIC afin d'être exécuté. Ce fichier a extension *.hex, et doit être transmis dans le PIC à l'aide un appareil appelé programmeur.

Le MikroC Pro est l'un des compilateurs qui facilitent la programmation des PIC. En utilisant ce logiciel on peut faire la compilation ainsi que corriger les fautes dans notre programme.

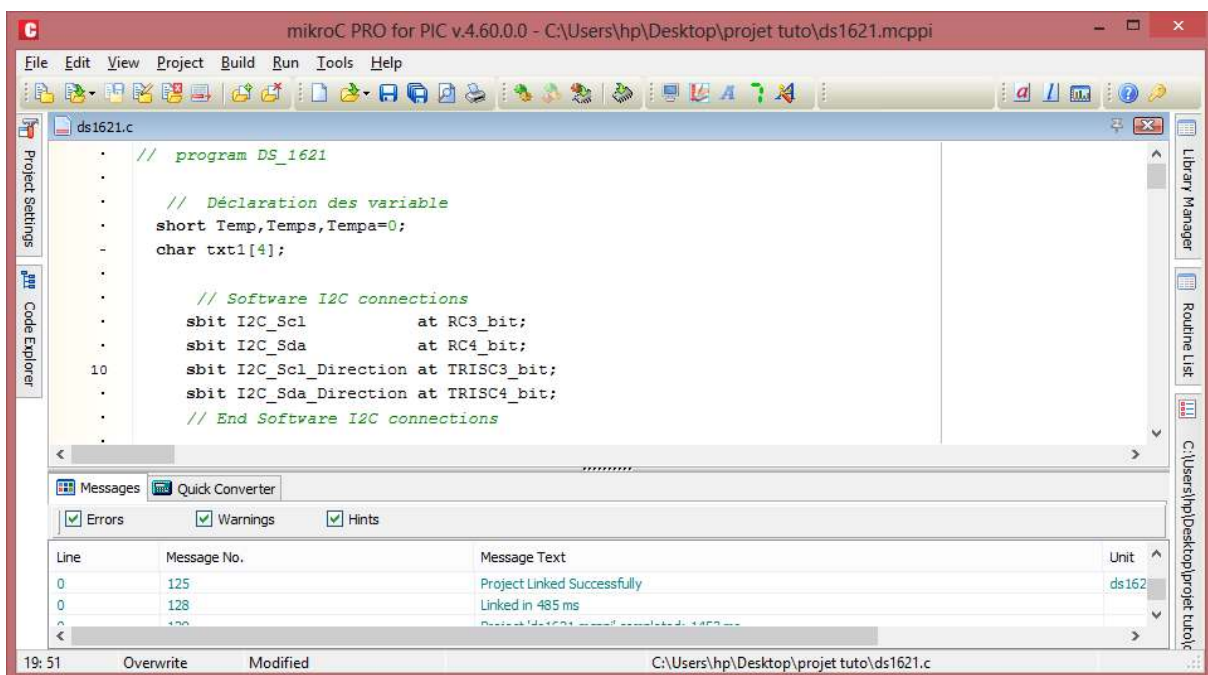


Figure28. MikroC Pro.

4.20 Création du Programme :

4.20.1 *Première étape* : Crée un nouveau projet

Les cinq figures qui suivent montrent comment on procède pour créer un nouveau projet

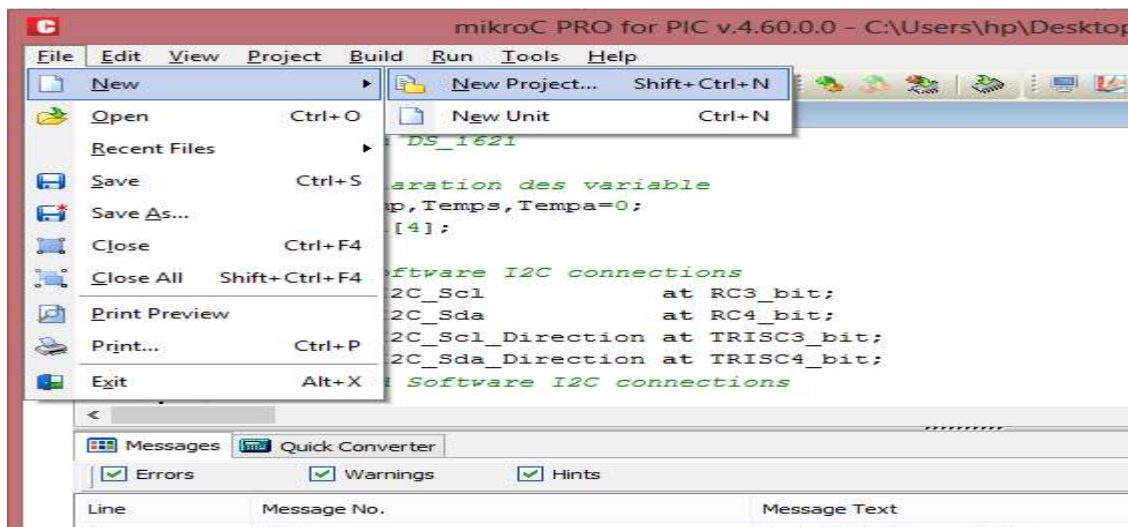


Figure29. Ouvrir un nouveau projet.

4.20.2 *Deuxième étape* : Choisir notre PIC dans la liste

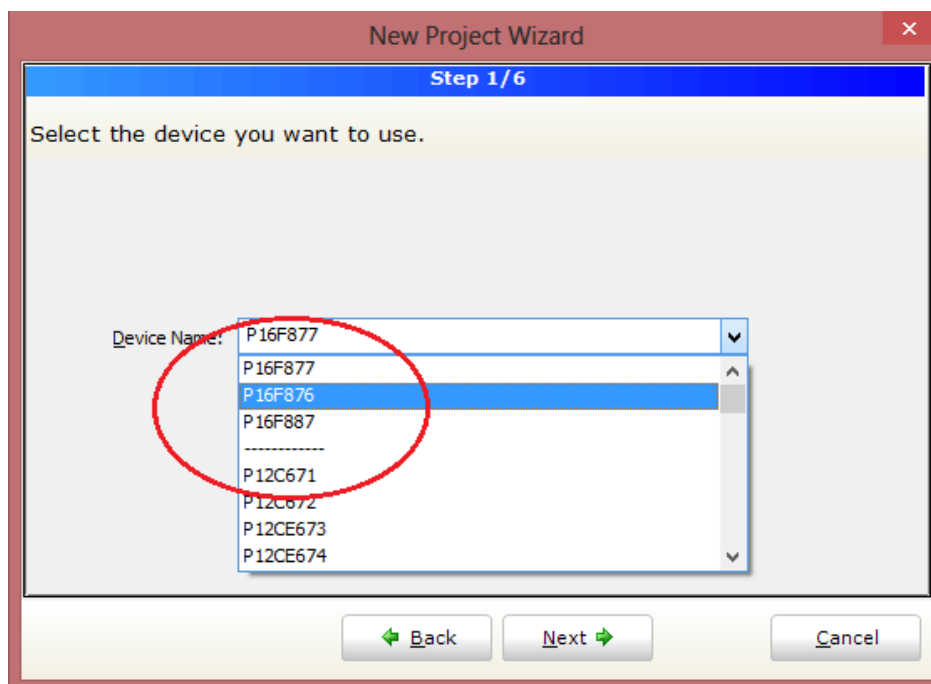


Figure30. Choix du PIC désiré.

4.20.3 Troisième étape : Choisir la fréquence de fonctionnement du programme

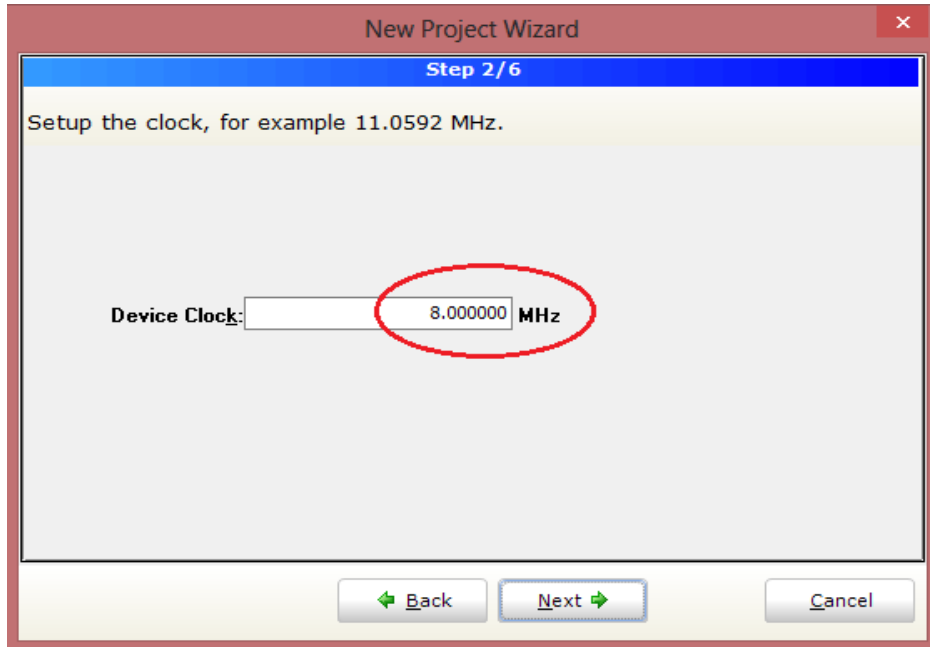


Figure31. Choix de la fréquence.

4.20.4 Quatrième étape : Choisir d'inclure toutes les bibliothèques ou pas

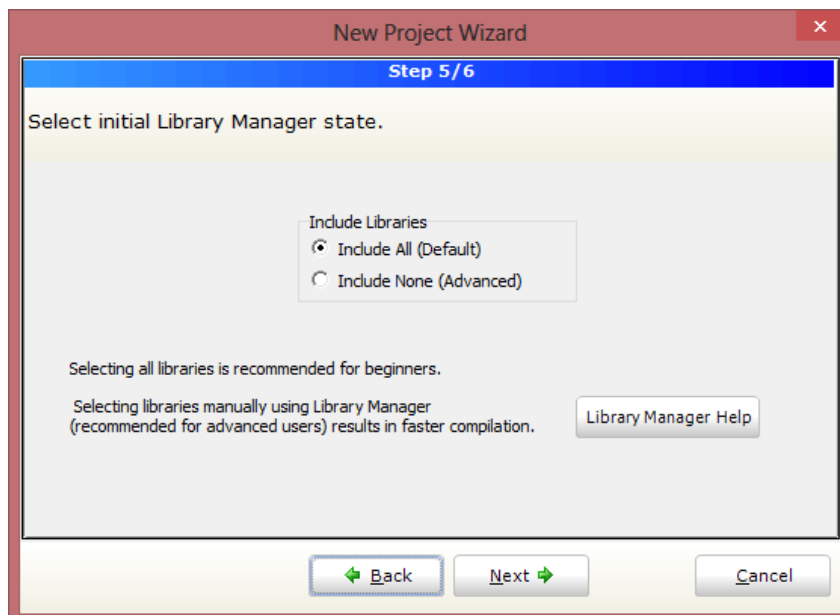


Figure32. Inclure les bibliothèques ou pas.

4.20.5 Dernière étape : Fin de la création du projet

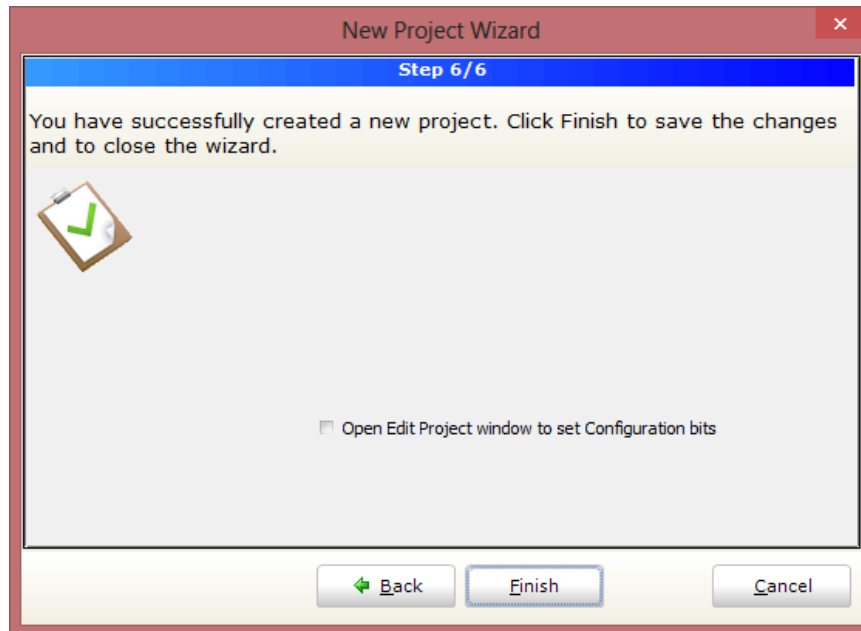


Figure33. Fin de la création d'un nouveau projet.

4.20.6 Ecriture du Programme :

Maintenant on peut écrire notre programme dans l'éditeur qui est apparu et qu'on va ensuite charger dans le PIC

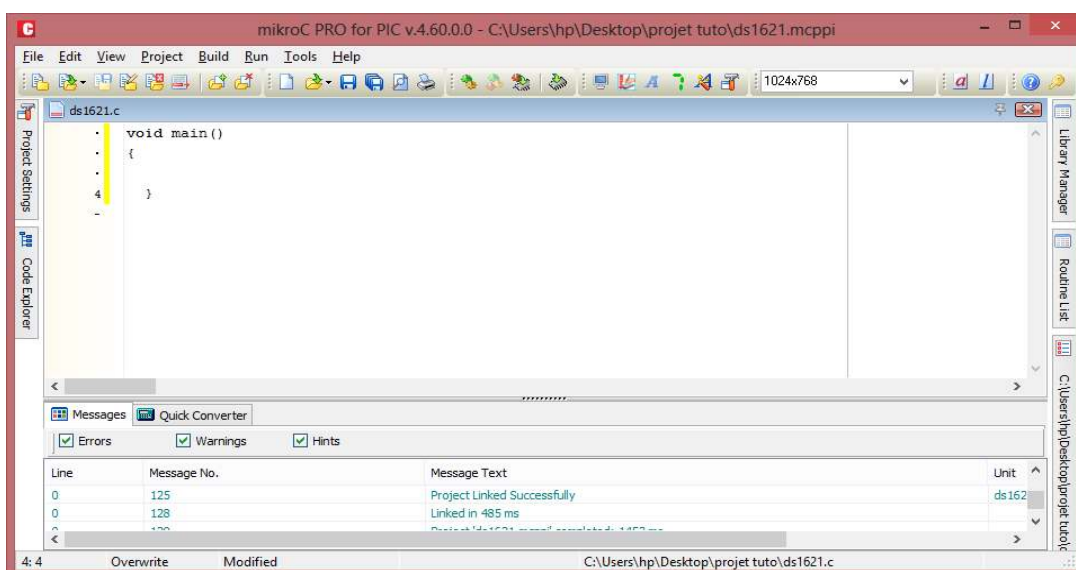


Figure34. Editeur du MikroC Pro.

Et le programme que nous avons développé en langage C est le suivant :

```
// program supervision par réseau GSM avec PIC16F876
```

```
short Temp,Tempa=0;
```

```
signed short temps;
```

```
char txt1[4],txt2[9];
```

```
char txt3[19]= "La Temperature est";
```

```
sbit I2C_Scl      at RC3_bit;
```

```
sbit I2C_Sda      at RC4_bit;
```

```
sbit I2C_Scl_Direction at TRISC3_bit;
```

```
sbit I2C_Sda_Direction at TRISC4_bit;
```

```
sbit LCD_RS at RB0_bit;
```

```
sbit LCD_EN at RB1_bit;
```

```
sbit LCD_D4 at RB2_bit;
```

```
sbit LCD_D5 at RB3_bit;
```

```
sbit LCD_D6 at RB4_bit;
```

```
sbit LCD_D7 at RB5_bit;
```

```
sbit LCD_RS_Direction at TRISB0_bit;
```

```
sbit LCD_EN_Direction at TRISB1_bit;
```

```
sbit LCD_D4_Direction at TRISB2_bit;
```

```
sbit LCD_D5_Direction at TRISB2_bit;
```

```
sbit LCD_D6_Direction at TRISB4_bit;
```

```
sbit LCD_D7_Direction at TRISB5_bit;

void Init()

{

    PORTB=0;

    TRISB=0xf0;

    TRISC.f3=0xf0;

    TRISC.f4=0xf0;

    PORTC=0;

    TRISC.f6=0xf0;

    TRISC.f7=0xf0 ;

    I2c1_Init(100000);

    Lcd_Init();

    Lcd_Cmd(_LCD_CLEAR);

    Lcd_Cmd(_LCD_CURSOR_OFF);

    Lcd_Out_Cp(" La temperature est de ");

    LCD_CMD(196);

    LCD_Out_Cp(" .");

    LCD_Out_Cp("C");

}

void Dyspl_Temp()

{
```



```
ShortToStr(Temp,txt1);

LCD_CMD(193);

Lcd_Out_Cp(txt1);

Lcd_Out(2,2,txt1);

}

void Read_Temp()

{

I2c1_Start();

I2c1_Wr(0x90);

I2c1_Wr(0xAC);

I2c1_Wr(0);

I2c1_Stop();

Delay_ms(10);

I2c1_Start();

I2c1_Wr(0x90);

I2c1_Wr(0xEE);

I2c1_Stop();

Delay_ms(10);

I2c1_Start();

I2c1_Wr(0x90);

I2c1_Wr(0xAA);

I2c1_Stop();
```

```
    Delay_ms(10) ;

    I2c1_Start();

    I2c1_Wr(0x91);

    Temp=I2c1_Rd(0);

    Temps=Temp;

    I2c1_Stop();

    Delay_ms(10);

    I2c1_Start();

    I2c1_Wr(0x90);

    I2c1_Wr(0x22);

    I2c1_Stop();

}

void main()

{

    Init();

    while (1) {

        Read_Temp();

        Tempa=Temps;

        Dyspl_Temp();

        Delay_ms(1000);

        Read_Temp();

        shortToStr(Temps, txt2);
```

```
    if(Tempa!=Temps)
    {
        Delay_ms(1000);

        UART1_Init(9600);

        UART1_Write_Text("AT+CMGS=");

        Delay_ms(500);

        UART1_Write(0x22);

        Delay_ms(500);

        UART1_Write_Text("00213561393708");

        Delay_ms(500);

        UART1_Write(0x22);

        UART1_Write(0x0D);

        Delay_ms(500);

        UART1_Write_Text(txt3);

        UART1_Write_Text(txt2);

        UART1_Write(0x0D);

        UART1_write(26);

        Delay_ms(500);

        UART1_Write(0x0D);

        Delay_ms(500);

    }
}
```

4.21 Schéma du projet sur ISIS Proteus et résultat de la simulation :

La figure suivante représente le schéma de simulation sur le logiciel ISIS Proteus

On remarque que l'afficheur LCD nous affiche la température 12°C celle indiquée par le capteur DS1621 et cette valeur change sur l'afficheur si on change sur le capteur. D'une autre part, on peut lire une tension de 5V sur le Voltmètre brancher à la ligne T_x du PIC qui est la patte RC6 et cela signifie que le microcontrôleur délivre bien une sortie vers le port série RS232 et cela représente l'envoi de l'SMS.

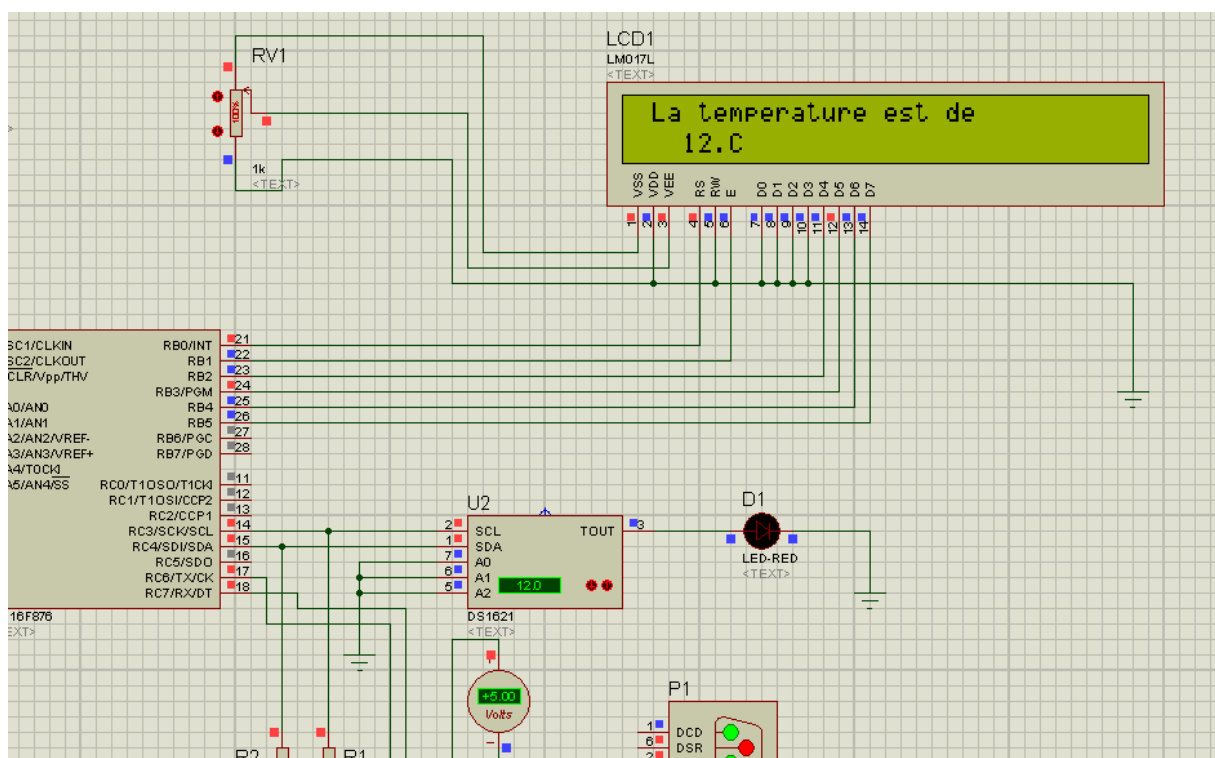


Figure35.Schéma ISIS.

4.22 Réalisation :

La figure suivante représente la réalisation du Projet :

On peut voir dans la figure ci dessus la plaque du circuit imprimé avec un afficheur LCD (placer au dessus de quelques composants pour gagner de l'espace) pour afficher directement la température et aussi un interface pour le module GSM afin d'envoyer les données par SMS.

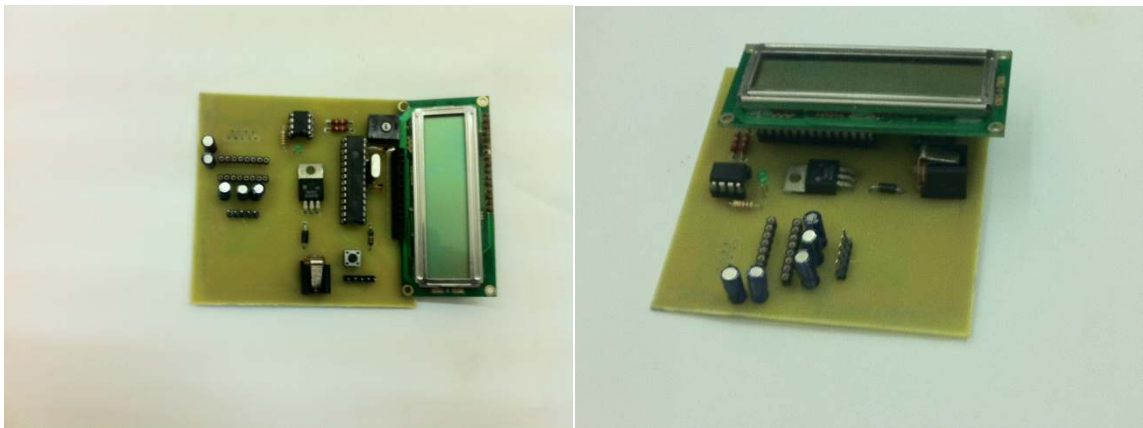


Figure36. Réalisation du Projet.

4.23 Conclusion :

Ce dernier chapitre a été entamé par une simulation et réalisation pratique en exploitant une méthodologie hiérarchique. Cette analyse a été suivie d'une explication plus ou moins détaillée des différentes étapes qu'il faut suivre pour la réalisation.

Enfin, le tout a été couronné par la réalisation pratique d'un dispositif automatique de supervision avec un PIC 16F876.

Intoduction

Les moyens traditionnels de transfert d'information ne répondent plus aux critères d'efficacité et aux contraintes de temps qui deviennent de plus en plus sévères. Le recours aux moyens de communication sophistiqués devient incontournable pour assurer une maîtrise des aléas qui peuvent être rencontrés, ainsi que satisfaire notre besoin en confort.

En pensant à la technologie de télécommunication GSM, il nous vient immédiatement à l'esprit la communication vocale, l'envoi et la réception des SMS et des MMS, l'internet mobile, et tous ce qui fonctionne en utilisant cette technologie moderne introduite dans notre vie quotidienne.

Il est moins évident de penser, qu'en se servant de ces services fournis à travers ce système, il est possible de contrôler et de commander des systèmes à distance en ayant recours au réseau GSM. Des applications peuvent être utilisées dans divers domaines comme le contrôle et la commande à distance des machines, des systèmes d'alarme et de surveillance, de commander des portes ou d'allumer des lampes...

En fait, la supervision par GSM (notre projet à réaliser), se présente comme une solution simple et facile qui peut nous aider à assurer une bonne manipulation de l'existant afin d'aboutir à concrétiser notre objectif.

Conclusion Générale

Dans ce mémoire, nous avons étudié comment on peut utiliser le système de communication mobile GSM pour acquérir des mesures de variables physiques à distance et éventuellement pour commander des systèmes. Le système de mesure et de commande à distance est simulé sur un logiciel de développement utilisant le langage C. La simulation montre que le système est fonctionnel. Une réalisation pratique a également été faite : elle comporte une carte à microcontrôleur et une interface avec le module GSM. Les applications sont nombreuses, par exemple dans le domaine de télésurveillance et des réseaux de capteurs. Le procédé peut être plus autonome plus pratique et évolutif compte tenu des progrès réalisés dans le domaine de la technologie de la communication.

Liste des figures

Figure 1. schéma descriptif de fonctionnement général du système.....	4
Figure 2. structure de la carte d'acquisition.....	4
Figure 3. Structure de la carte d'interface du module GSM.....	5
Figure4. Architecture d'un réseau de la première génération.....	7
Figure5. Division du territoire d'un réseau en n cellules.....	9
Figure6. Motif de sept cellules.....	10
Figure7. structure d'un réseau GSM.....	11
Figure8. Module GSM SIEMENS MC45.....	12
Figure9. Emplacement du PIC.....	14
Figure10. Schéma interne du SIEMENS MC45.....	15
Figure11. Module GSM GM862.....	16
Figure12. Schéma des connexions du GM862.....	18
Figure13. Architecture du GM 862.....	18
Figure14. Schéma de fonctionnement.....	20
Figure15. Brochage du DS1621.....	25
Figure16. Diagramme de fonctionnement du DS1621.....	27
Figure17. Structure du bus I ² C.....	28
Figure18. Condition START.....	30
Figure19. Condition STOP.....	30
Figure20. Acquiescement.....	31
Figure21. Principe de l'architecture Von Neumann.....	33

Figure22. Principe de l'architecture Harvard.....	34
Figure23. Les éléments constitutifs du PIC 16F87X.....	42
Figure24. Cœur du PIC.....	43
Figure25. Les mémoires du PIC 16F87X.....	46
Figure26. : Les ports du PIC.....	48
Figure27. USART (Universal Synchronous Asynchronous Receiver Transmitter).....	50
Figure28. MikroC Pro.....	54
Figure29. Ouvrir un nouveau projet.....	55
Figure30. Choix du PIC désiré.....	55
Figure31. Choix de la fréquence.....	56
Figure32. Inclure les bibliothèques ou pas.....	56
Figure33. Fin de la création d'un nouveau projet.....	57
Figure34. Editeur du MikroC Pro.....	57
Figure35. Schéma ISIS.....	63
Figure36. Réalisation du projet.....	64

Liste des Tableaux

Tableau 1. Caractéristique technique du module GSM MC45.....	13
Tableau2. Caractéristiques techniques du module GM862.....	17
Tableau3. Commandes dédiées au service SMS.....	20
Tableau4. Description des PIN du DS1621.....	26
Tableau5. Différents circuit de la famille 16F87X.....	38
Tableau6. Appellations, fonction, types électrique et logique de tous les pattes pouvant être rencontrées sur les différentes versions de PIC.....	40

Listes des acronymes et abréviations

SMS : *Short Message Service*.

GSM : Global System for Mobile Communications.

PIC : Peripheral Interface Controller.

TSF : Télégraphe sans fil.

MSC : Mobil Switching Center.

RTC : Réseau Téléphonique Commuté.

GPRS : General Packet Radio Service.

ASIC : Application Specific Integrated Circuit.

SIM : Subscriber Identity Module.

AT : attention command.

TOR : tout ou rien.

LED : Light-emitting Diode.

SCL : Serial Clock Line.

GND : Ground.

I²C : Inter Integrated Circuit.

SMBUS : system management bus.

RISC : Reduced instruction set computer.

EEPROM : Electrically Erasable Programmable Memory.

ICSP : In Circuit Serial Programming.

PSP : Parallel Slave Port.

RAM : Random Access Memory.

Cite internet :

- www.datasheet.com. [4]
- www.mikroe.com.
- www.Microchip.com.
- www.Electroniquepratique.com.
- www.tafats.fr [2]

Projet de fin d'étude :

- Réalisation d'une chaine d'acquisition à base d'un microcontrôleur avec l'utilisation des capteurs.
Proposé par : Maamoun Mountassar .
Présenté par : Lalaoui Sofiane & Bensaada Mohamed.
Université Saad dahlab de blida. 2004
- Supervision par GSM. [3]
Proposé par : Hatem Abidi.
Présenté par : Rafik Saied & Alaeddine Mzoughi.
Université de Tunis El Manar. 2010.
- Acquisition de plusieurs températures à travers le bus I2C.
Proposé par : Boualem Kazed.
Présenté par : Souhila Mardoud & Razika Bouziani.
Université Saad Dahlab de Blida. 2002
- Acquisition et Afiichage de l'humidité à travers le bus I²C.
Proposé par : Boualem Kazed.
Présenté par : Boulahres Doudja & Haoua Saliha Session 2004.
Université Sad Dahlab de Blida. 2004

Livres :

- Le manuel du bus I2C théorie et pratique avec des application de Brodier Jean édition. Edition PUBLITRONIC. 1995.

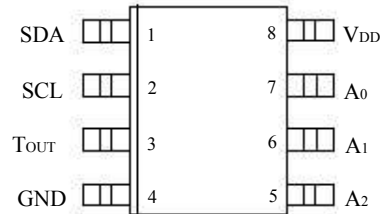
- Le bus I²C Principe et mise en œuvre 2^{ème} édition de Dominique Paret.
Edition DUNOD. 1999.[5]
- Apprendre la programmation des PIC par l'expérimentation et la simulation 3^{ème} édition de Pascal Mayeux. Edition Techniques et Scientifiques Française. 2000.
- Le Réseau GSM et l'évolution GPRS : une étape vers UMTS 3^{ème} édition de Joachim Tisal. Edition DUNOD. 1999 [1]
- Les microcontrôleurs PIC. Description et mise en œuvre 2^{ème} édition de Christian Tavernier. Edition DUNOD. 2002. [6]

Annexes

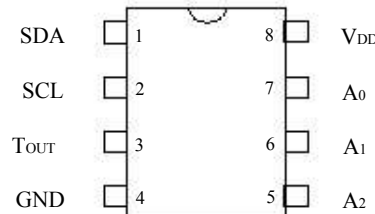
FEATURES

- Temperature measurements require no external components
- Measures temperatures from -55°C to +125°C in 0.5°C increments. Fahrenheit equivalent is -67°F to 257°F in 0.9°F increments
- Temperature is read as a 9-bit value (2-byte transfer)
- Wide power supply range (2.7V to 5.5V)
- Converts temperature to digital word in less than 1 second
- Thermostatic settings are user definable and nonvolatile
- Data is read from/written via a 2-wire serial interface (open drain I/O lines)
- Applications include thermostatic controls, industrial systems, consumer products, thermometers, or any thermal sensitive system
- 8-pin DIP or SO package (150mil and 208mil)

PIN ASSIGNMENT



DS1621S 8-PIN SO (150mil)
 DS1621V 8-PIN SO (208mil)
 See Mech Drawings Section



DS1621 8-PIN DIP (300mil)
 See Mech Drawings Section

PIN DESCRIPTION

- SDA - 2-Wire Serial Data Input/Output
- SCL - 2-Wire Serial Clock
- GND - Ground
- TOUT - Thermostat Output Signal
- A0 - Chip Address Input
- A1 - Chip Address Input
- A2 - Chip Address Input
- VDD - Power Supply Voltage

DESCRIPTION

The DS1621 Digital Thermometer and Thermostat provides 9-bit temperature readings, which indicate the temperature of the device. The thermal alarm output, TOUT, is active when the temperature of the device exceeds a user-defined temperature TH. The output remains active until the temperature drops below user defined temperature TL, allowing for any hysteresis necessary.

User-defined temperature settings are stored in nonvolatile memory so parts may be programmed prior to insertion in a system. Temperature settings and temperature readings are all communicated to/from the DS1621 over a simple 2-wire serial interface.

OPERATION

Measuring Temperature

A block diagram of the DS1621 is shown in Figure 1.

The DS1621 measures temperature using a bandgap-based temperature sensor. A delta-sigma analog-to-digital converter (ADC) converts the measured temperature to a digital value that is calibrated in °C; for °F applications, a lookup table or conversion routine must be used.

The temperature reading is provided in a 9-bit, two's complement reading by issuing the READ TEMPERATURE command. Table 2 describes the exact relationship of output data to measured temperature. The data is transmitted through the 2-wire serial interface, MSB first. The DS1621 can measure temperature over the range of -55°C to +125°C in 0.5°C increments.