UNIVERSITÉ MOULOUD MAMMERI TIZI-OUZOU

Faculté de Génie Électrique et D'informatique

Département d 'électronique

Mémoire de fin d'études

En vue de l'obtention du diplôme de MASTER II en Electronique

Option : Réseaux et Télécommunication

Thème

Étude et réalisation d'un banc de mesure pour la surveillance d'un moteur à base de la carte de développement OLIMEX.

Proposé et dirigé par :

Mr M .LAGHROUCHE

Réalisé par :

NECHICHE Sofiane

REMICHI Hichem

Promotion 2012-2013

Remerciements

Nous remercions le Bon Dieu de nous avoir donné la force, le courage et la volonté pour accomplir ce travail.

Nous remercions notre promoteur Mr. LAGHROUCHE pour son aide et conseils tout au long du projet.

Nous remercions également le président de jurys ainsi que les membres de jurys d'avoir accepter d'évaluer et de juger notre travail.

Nos remerciements s'adressent aussi pour la responsable du laboratoire projet Mme SLIMANI pour son soutien et sa confiance.

Dédicaces

Je dédie ce modeste travail

À Toute ma famille, mon père ma mère ainsi que mes frères Idir et Massi, ma sœur Samira et son mari Arezki, mon cousin Loucif et Mhand.

À mon binôme Hichem

À Tous mes ami(e)s de la promotion : Rafik, Mhenna, Rabah, Nasser, Djamal , Belkacem, Titi, Fatima, Lidia, Lamia, Dihia, Ibtissem, Lila, Malika.

À tous mes amis : Mouloud, sofiane, Abdenour,Rabah, Farid, Idris, Gaya, Samy, Belaid, Chabane, Chafaa, Khelifa, Moh, Koucy, Khaled, Wahab, Lyes ,Lotfi

À toutes les personnes qui me sont chères,

Sofiane

Dédicaces

*Nous remercions **ALLAH** le tous puissant de nous avoir donné la force et le courage afin d'accomplir ce travail.*

Je dédie ce travail

A mon cher père et ma chère mère. Pour le grand amour dont ils m'ont entouré depuis ma naissance et pour leurs patiences et leurs sacrifices. J'espère que j'ai atteins leurs souhaits. ***Que dieu me les protègent.***

A mon cher frère Sofiane et à ma petite et chère sœur Meriem.

A toute ma famille.

A la mémoire de mes tantes DAHBIA et ZOUHRA.

Aux petites anges Ryma et Wissal.

A mon binôme Sofiane.

A tous mes ami(e)s de la promotion : Djamal, Rafik, Mhena, Rabah, Nacer, Titi, Fatima et Dihia.

A tous mes amis : Alilou, Koucy, Elhadi, Smail, Ammar, Farid, Mounir, Fateh, Yacine, Lyas, lounes et Mahdi.

A tous ceux qui m'ont aidé de près ou de loin pour réaliser ce travail.

A toute la promotion ELN 2013.

HICHEM

Table des matières

Chapitre III : Description des capteurs

Chapitre IV : Conception logicielle et réalisation

Introduction générale

De nos jours, les machines électriques tournantes sont utilisées dans des domaines aussi variés tel que le transport (trains, véhicules motorisés, etc.), la production électrique (Alternateurs) l'industrie de production, ou encore l'électroménager. De plus, bien que leurs commandes par des équipements contacteurs soient parfaitement adaptées pour un grand nombre d'applications, l'emploi des systèmes électroniques est en constante progression ce qui élargit leurs champs d'applications.

Les principales défaillances rencontrées dans un moteur électrique sont dû aux vibrations ou à des températures qui excèdent les valeurs admissibles. A titre indicatif, si la température de fonctionnement est supérieure à la température permanente admissible la plus élevée, la durée de vie du moteur est réduite de moitié.

Les conséquences des défauts des moteurs se manifestent par des performances médiocres au démarrage, des vibrations excessives, et des échauffements. Tous ces éléments contribuent à la détérioration du rotor, et engendrent des effets secondaires dans le stator qui peuvent provoquer ainsi sa panne.

Afin de diminuer les risques des défaillances d'une machine, on doit mettre en œuvre un dispositif de surveillance et de diagnostique continue qui suit l'évolution de son l'état.

Notre travail consiste à réalisé un système d'acquisition de trois grandeurs essentielles pour le diagnostique des défauts: la température, la vitesse de rotation, ainsi que les vibrations engendrées. Cette tâche est réalisée en utilisant un système à base d'un microcontrôleur PIC18F4550. Le banc de mesure réalisé doit assurer :

- L'acquisition des ces trois grandeurs.
- L'affichage sur un LCD des différentes mesures acquises.
- Le transfert des données vers un PC via le port RS232.

Une interface graphique sous LabVIEW nous permettra d'afficher ces valeurs ainsi qu'une alarme pour signaler le dépassement des seuils prédéfinis.

Notre travail est divisé en quatre chapitres, présenté comme suit:

- Introduction générale.
- Chapitre I : Description des défaillances des moteurs électriques.
- Chapitre II : Présentation de la carte de développement d'OLIMEX ainsi que ces différents composants.
- Chapitre III : Description des différents capteurs utilisés.
- Chapitre IV : Présentation des différents logiciels utilisés ainsi que la réalisation pratique de notre système.
- Conclusion.

Chapitre I

Défaillances dans les moteurs électriques

Introduction :

La plupart des entreprises de réparation de moteurs s'entendent à reconnaitre que la majorité des pannes de moteurs se présente tout d'abord comme des défaillances mécaniques ou électriques d'une ou de plusieurs parties du moteur.

Les origines des défaillances peuvent êtres :

- Un défaut de fabrication de la machine.
- Le résultat d'une application inadéquate.
- Des phénomènes physiques qui apparaissent lors du fonctionnement.

Un défaut dans un composant peut être le résultat d'une usure, d'une mauvaise conception, d'un mauvais montage (désalignement), d'une mauvaise utilisation, ou d'une combinaison de ces différentes causes. Si un défaut n'est pas détecté suffisamment tôt, il peut entraîner la dégradation complète de la machine.

La figure ci-dessous récapitule la distribution des défauts dans un moteur électrique.

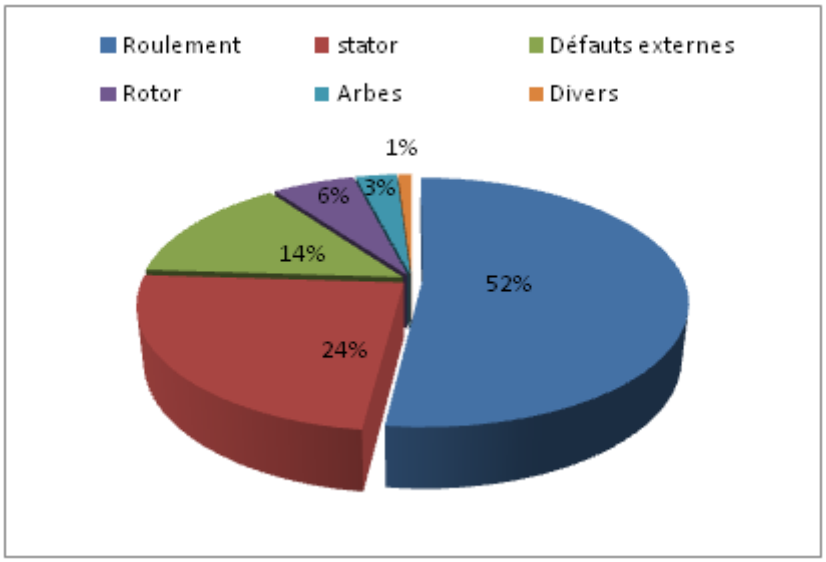


Figure I.1 Distribution des défauts dans un moteur électrique

Les défaillances rencontrées dans les moteurs électriques sont souvent liées à des phénomènes physiques qui apparaissent durant leurs fonctionnements. Une partie de l'énergie consommer par le moteur se dissipe par effet joule entrainant des fortes températures pouvant nuire aux structures les plus fragiles comme les isolations électriques.

L'enroulement statorique d'une machine électrique est soumis à des efforts induits par une variété de facteurs, parmi les plus importants, une surcharge thermique, les vibrations mécaniques [1][2], les pics de tension provoqués par le réglage de fréquence...etc.

La rotation des moteurs peut engendrer des vibrations qui se transmettre aux structures avoisinantes par les point de fixation, provocant ainsi des anomalies de fonctionnement qui peuvent conduire a l'arrêt complet de la machine.

I.1 Influence de la température sur les moteurs électriques :

Lors du fonctionnement des moteurs électriques il faut tenir compte des contraintes thermiques. En effet, les différents matériaux utilisés dans la machine doivent être adaptés aux températures prévues, cependant il arrive parfois que la température dépasse les limites admissibles pour des raisons de surcharge. L'utilisateur doit être en mesure de détecter les signes précurseurs de cette anomalie pour prendre les mesures nécessaires.

Une température élevée, maintenue et prolongée peut endommager le système d'isolation des enroulements. De plus, le point chaud est généralement atteint dans le bobinage ce qui accroît la défaillance des isolants électriques.

I.2 Influence de la vitesse de rotation sur les moteurs électriques :

Un moteur est utilisé pour fonctionner à son point nominal, lequel est indiqué sur sa plaque signalétique, exemple pour un moteur asynchrone classique :

- la vitesse : (1500 tr/min).
- la puissance en kW.

Le variateur de vitesse est un dispositif de réglage de la vitesse de rotation du moteur. Il est choisi selon le point de fonctionnement nominal [3], les constructeurs de machine doivent ainsi fournir à leurs clients la courbe caractéristique du couple résistant / vitesse de la machine.

De ce fait, le variateur est capable de fournir un couple important c'est-à-dire une puissance croissante jusqu'à atteindre sa puissance nominale plaquée.

Au-delà de cette puissance le moteur peut entrer dans un état d'instabilité, entrainant un déséquilibre du rotor, roulement défectueux...etc.

I.3 Influence des vibrations sur les moteurs électriques :

I.3 .1 Les sources de vibrations :

Les principales sources de vibrations dommageables pour les moteurs électriques sont le comportement dynamique des ces composants. Ces vibrations peuvent être de forte amplitude qui peuvent être transmises au reste de la machine suivant les points de fixation, générant des mouvements en translation comme en rotation, et parfois de fortes dissipations de chaleur.

Par exemple dans le processus de fabrication par enlèvement de matière sur machines-outils, l'apparition des vibrations est inévitable, ce qui accélère l'usure, la dégradation de l'outil. De plus, elles risquent aussi d'engendrer des dommages, voire même des ruptures, dans les composants de la machine.

La figure suivante représente les vibrations engendrées et transmises dans une machine tournante.

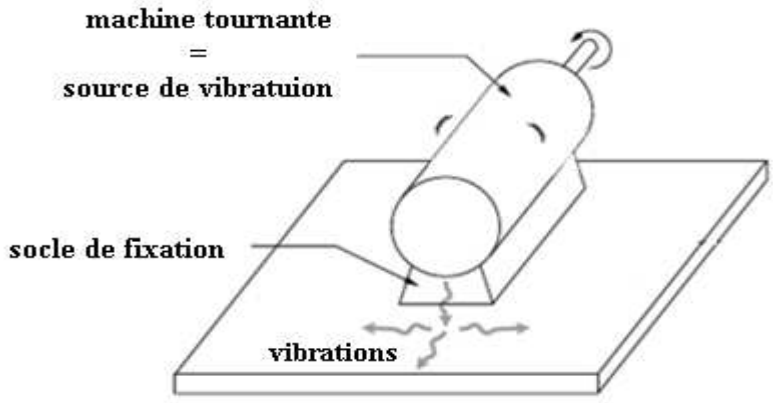


Figure I.2 Les vibrations engendrées et transmises dans une machine tournante

Le comportement dynamique des moteurs est identifié par l'utilisation des capteurs d'accélérations, qui transforment les vibrations engendrées, en signaux électriques. Ces signaux sont conditionnés pour les rendre exploitables par les appareils d'analyses.

Les signaux enregistrés par différents matériels tels que les analyseurs de spectre, les collecteurs de données ou les cartes d'acquisition associés à des logiciels de traitement, serons utilisés à des fins d'études.

La figure ci-dessous illustre un exemple l'emplacement des capteurs d'accélérations dans un moteur électrique.

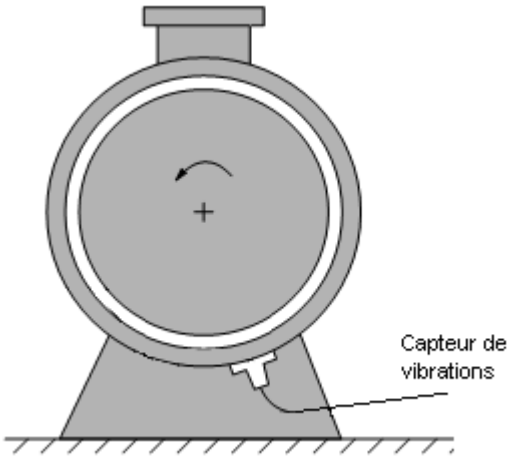


Figure I.3 Capteur d'accélération dans un moteur électrique

I.3.2 L'analyse vibratoire :

L'analyse des vibrations permet de mieux gérer les défaillances, en détectant les défauts à un stade précoce, avant qu'ils ne deviennent critiques.

C'est un outil nécessaire, son emploi vise à servir trois niveaux d'analyse :

- la surveillance.
- le diagnostique.
- le suivi de l'état d'endommagement des composants.

I.3.3 Le but de l'analyse vibratoire :

Le but étant d'identifier le comportement vibratoire des machines ou le comportement dynamique d'un ensemble de pièces prédéfinies. Sur la base des informations obtenues, nous déterminons et nous caractérisons la zone des vibrations susceptible d'intérêt [4]. Cette identification va permettre d'établir un diagnostique fiable en vue d'une maintenance planifiée.

L'intérêt est de séparer la contribution des différentes sources vibratoires généralement et directement liées à une défaillance plus ou moins importante d'un composant mécanique à partir de plusieurs mesures réalisées par l'accéléromètre.

La séparation permettra non seulement de localiser les défauts sur les composants mais également de suivre l'évolution de l'endommagement pour amélioré les conditions de fonctionnement mécanique et électrique.

I.3.4 Structure des signaux vibratoires :

Les signaux vibratoires relevés sur les systèmes mécaniques en fonctionnement contiennent l'information nécessaire relative à l'état des composants de la machine. L'isolation de l'information relative à chaque composant est un sujet très important pour le diagnostique.

La figure ci-dessous représente un exemple de la DSP de l'enveloppe carré d'un signal vibratoire.

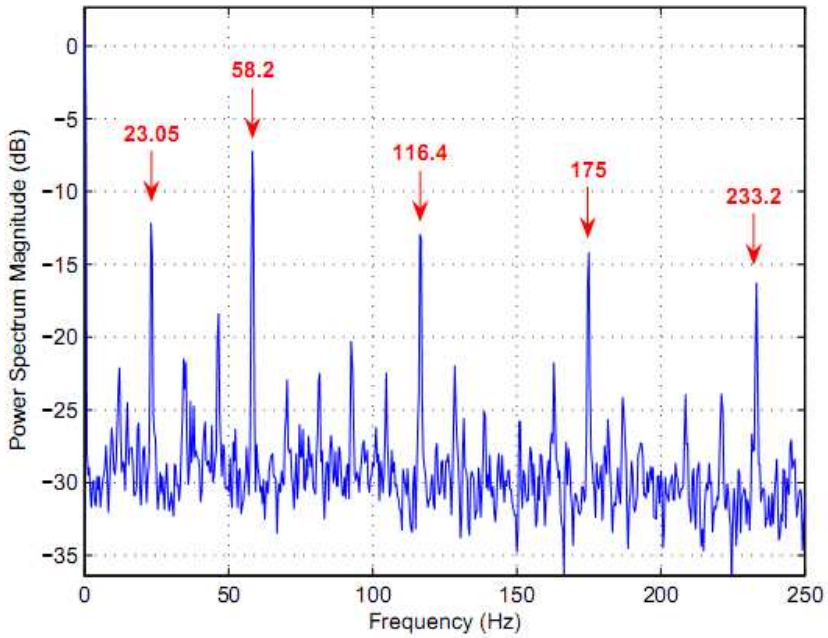


Figure I.4 DSP de l'enveloppe carrée du signal vibratoire

Chapitre II

Présentation de la carte de développement

Introduction :

PIC-USB-STK est une carte de développement d'**OLIMEX**, elle nous permet d'exploiter toutes les caractéristiques du **PIC 18F4550** de Microchip qui représente le noyau de la carte.

II.1 Contenue de la carte :

La carte de développement d'OLIMEX est composée des éléments suivants :

- CPU : PIC 18F4550.
- ICSP /ICD : connecteur pour la programmation du PIC.
- USB 2.0 type B pour la connexion de la carte avec le PC.
- Connecteur RS 232.
- Connecteur SD-MMC pour le stockage des données.
- Connecteurs entres /sorties audio.
- Cinq LEDs.
- Un potentiomètre.
- Quatre boutons.
- Un quartz de 20Mhz.
- Un bouton de Reset.
- Connecteur d'alimentation (9V-DC) ou (6V-AC).
- Des ports libres pour faire une extension de la carte (faire d'autres applications).

II.2 Les modes de programmation de la carte :

La carte de développement d'OLIMEX peut être programmée:

- Avec un câble USB A-B connecté directement avec le port USB de la carte en utilisent le Bootloader et le compilateur C18.
- Avec un câble RS232 mal-femelle, connecté avec le port DB9 femelle de la carte.
- Avec un programmateur externe, dans notre projet en utilise le mikroC comme compilateur.

Remarque :

Pour utiliser la carte en mode USB, il est nécessaire de charger le Bootloader pour le PIC (téléchargeable gratuitement sur le site d'OLIMEX) en utilisant les connecteurs ICSP de la carte et un programmateur PIC-ICD2 ou en utilisant directement un programmateur externe.

II.3.1 Vue globale de la carte :

La figure suivante représente la carte avec ces différents composants :

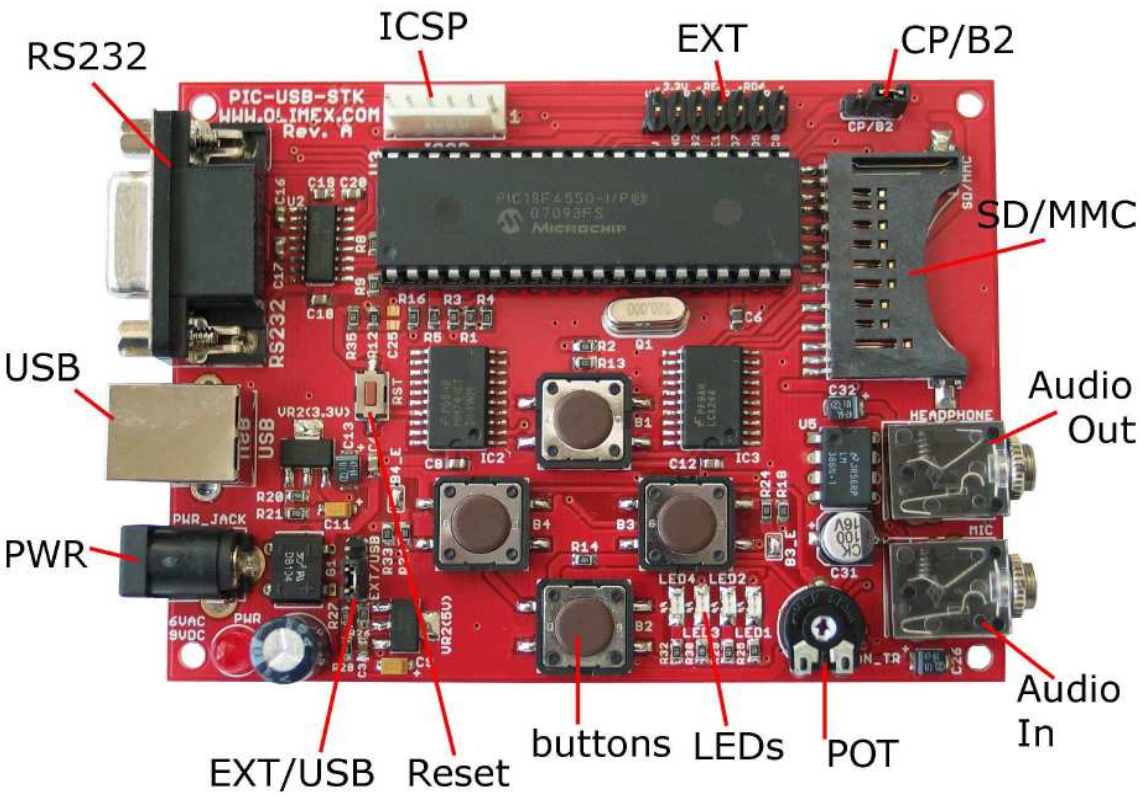


Figure II.1 La carte de développement d'OLIMEX

II.3.2 Circuit électrique de la carte :

Le circuit électrique de la carte OLIMEX utilisé dans notre projet est représenté dans la figure II.2.

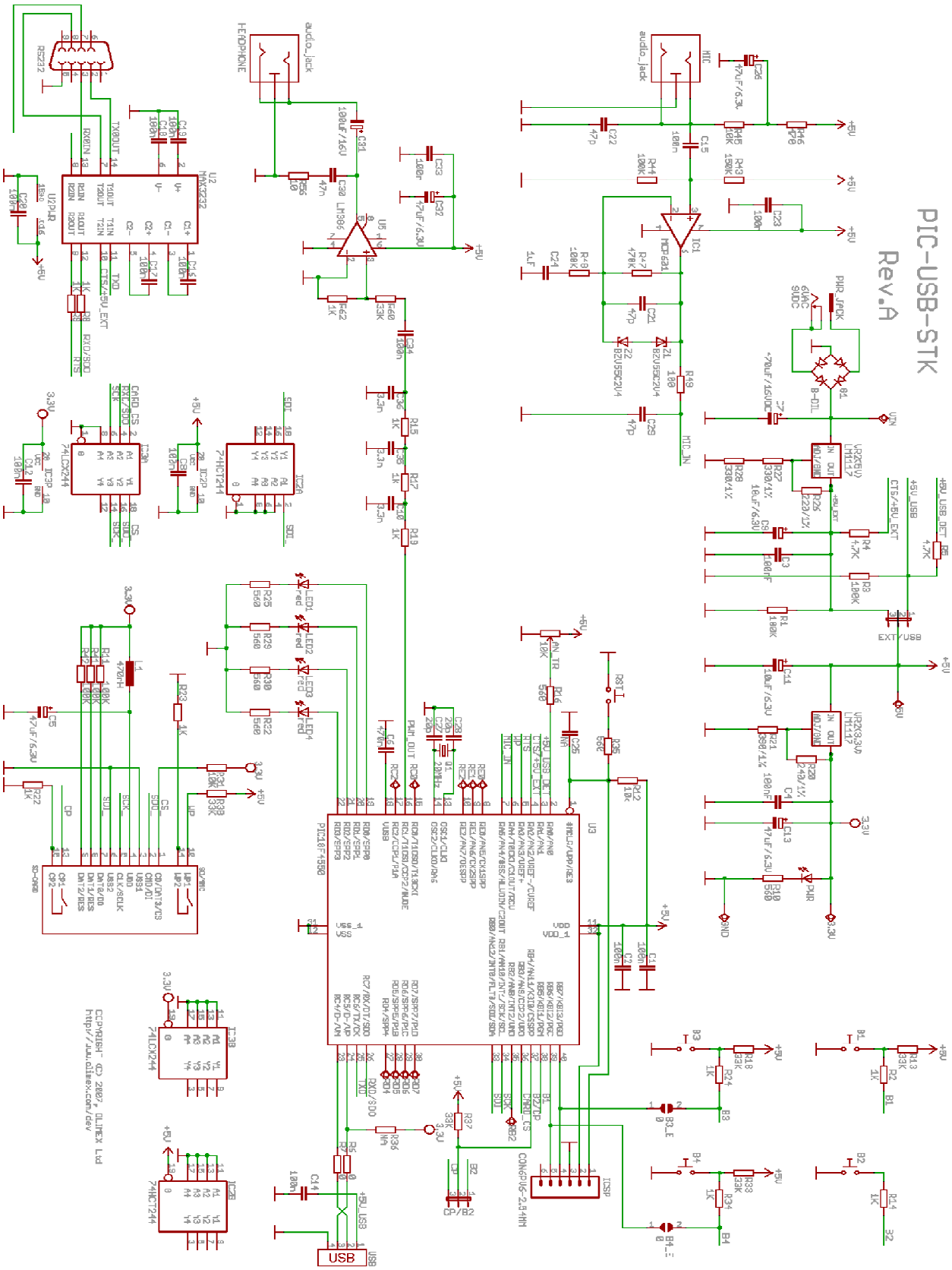


Figure II.2 Circuit électrique de la carte

II.4 Description du PIC18F4550 (CPU de la carte) :

C'est un microcontrôleur de la société « Microchip ». Il est classé dans la famille «High Performance » et cela grâce à sa richesse en ressources matérielles internes et de la capacité très importante de sa mémoire vive et de sa mémoire de programme. Il fait partie des PICs qui sont mieux adaptés à la programmation en langage C.

Le PIC 18F4450 est sous forme d'un boitier DIL(Dual In-Line Package) opaque (en plastique) qui comporte 40 pines , dont 33 pins d'entrée/sortie , deux paires de pins pour l'alimentation (V_{SS} et V_{DD}) , une entrée reset et deux pins pour l'oscillateur, une comme entrée et l'autre comme sortie . Cette dernière peut être configurée comme Entrée/Sortie, dans le cas ou elle n'est pas utilisée par l'oscillateur externe.

La figure suivante représente le brochage du microcontrôleur :

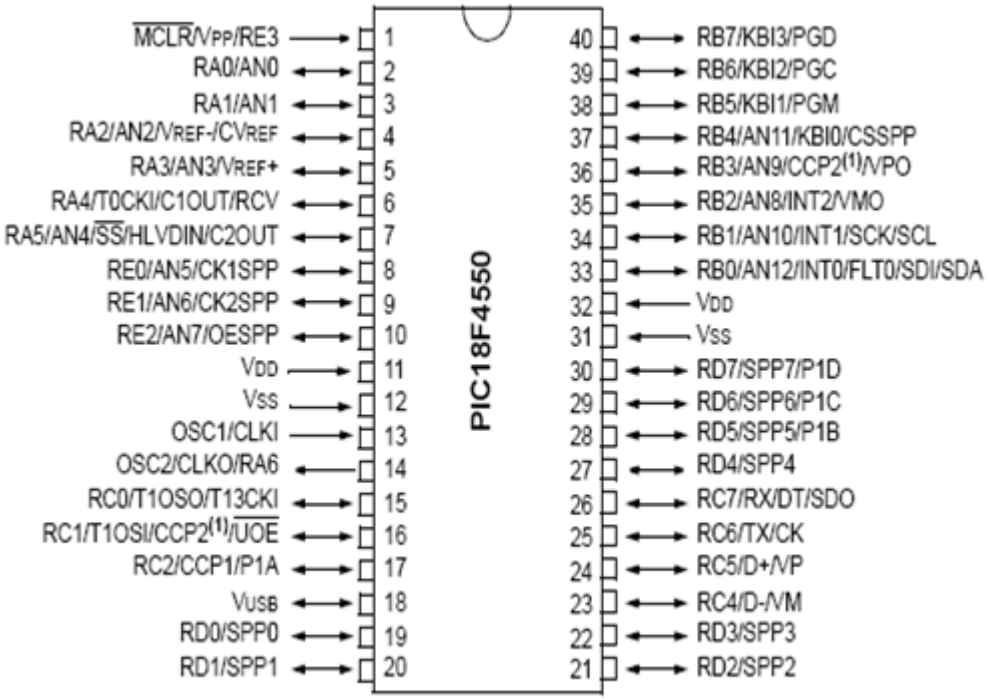


Figure II.3 Brochage du PIC 18F4550

II.4.2 Caractéristiques principales du PIC :

Ce microcontrôleur possède une mémoire de programme de type **FLASH** qui peut supporter jusqu'à 100000 cycles d'effacement / écriture, et une mémoire de donnés qui supporte 1000000 cycles d'effacement / écriture. La durée de rétention minimale de l'information est de 40 ans. Ce Pic peut fonctionner avec un oscillateur qui peut atteindre 48MHz.

Il dispose de trois sources d'interruptions externes (INTCON, INTCON2, INTCON3) qui peuvent être classé en niveaux de priorité.

Il est muni de treize(13) entrées analogiques qui permettent la conversion des signaux analogiques en numériques.

Il est muni de trois modules de transmission série :

- MSSP (Master synchronous Serial Port) qui possède deux modes de fonctionnement :
  - SPI (4 Fils).
  - I2C (3 Fils).
- USART (RS232 et RS485).
- USB.

Afin de protéger le programme contre la relecture et la copie frauduleuse à partir d'un programmateur externe, le PIC est muni d'un code de protection qui peut être activé ou non. Il supporte la programmation en circuit sous forme série (ICSP), ce qui permet de modifier le contenu de sa mémoire de programme sans devoir l'enlever de l'application ou il est installé.

II.4.3 Les ports d'entrées/sorties parallèles :

Le port **A** dispose de **7** pins d'entrées/sorties de **RA0 à RA6**. Chacun des ports **B, C** et **D** comportent **8** pins d'entrées/sorties, tandis que le port **E** est doté de **3** pins.

Les ports **A** et **E** peuvent être programmés en entrées analogiques (sauf les pins **RA4** et **RA6**). Le port B contient 3 entrées d'interruptions externes (**RB0, RB1, RB2**). Le port D peut être utilisé comme un port parallèle esclave (SPP).

II.4.4 Organisation de la mémoire :

Le PIC possède une mémoire de programme et de données. La structure Harvard des PICs fournit un accès séparé à chacune d'elle. Ainsi un accès aux deux est possible pendant le même cycle machine.

II.4.4.1 La mémoire de programme : C'est elle qui contient le programme à exécuter, de taille de 32 Kbytes, une instruction est codée sur deux octets, ce qui lui permet d'enregistrer 16K de mots d'instructions. Le programme est implanté dans la mémoire flash à l'aide d'un programmateur.

II.4.4.2 La mémoire de données : Elle se décompose en deux parties : une RAM de 2048 Bytes et une zone EEPROM de 256 Bytes. La première contient les SFRs (Special Function Registers) qui permettent de contrôler les opérateurs sur le circuit, la deuxième contient les registres généraux, libre pour l'utilisation.

La figure suivante montre le MAP mémoire du PIC 18F4550 que nous avons utilisé.

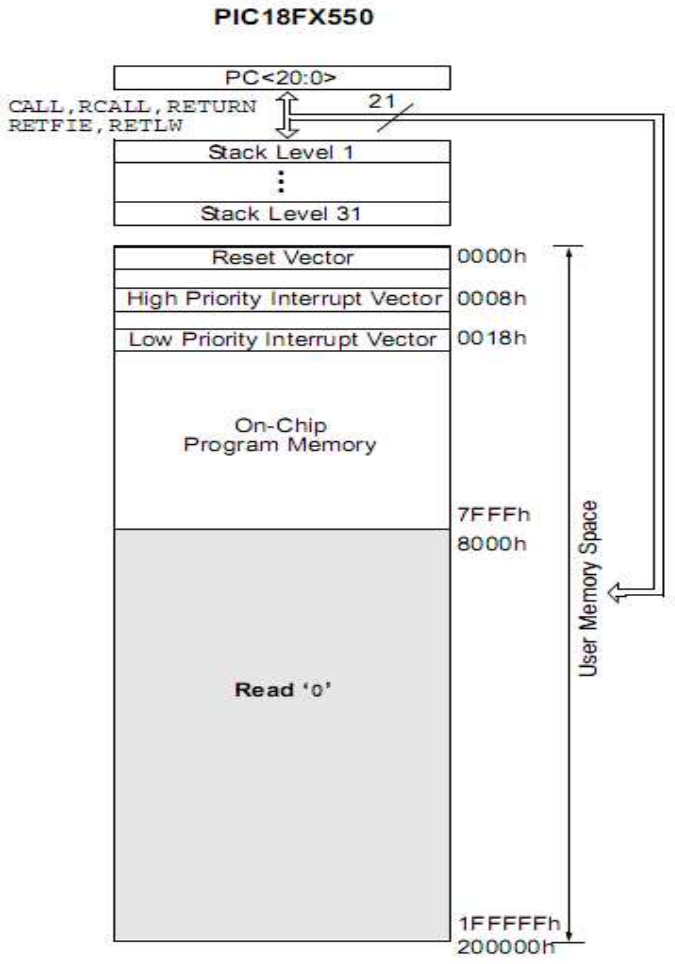


Figure II.4 MAP mémoire du pic 18F4550

La figure II.5 représente la structure interne du PIC 18F4550.

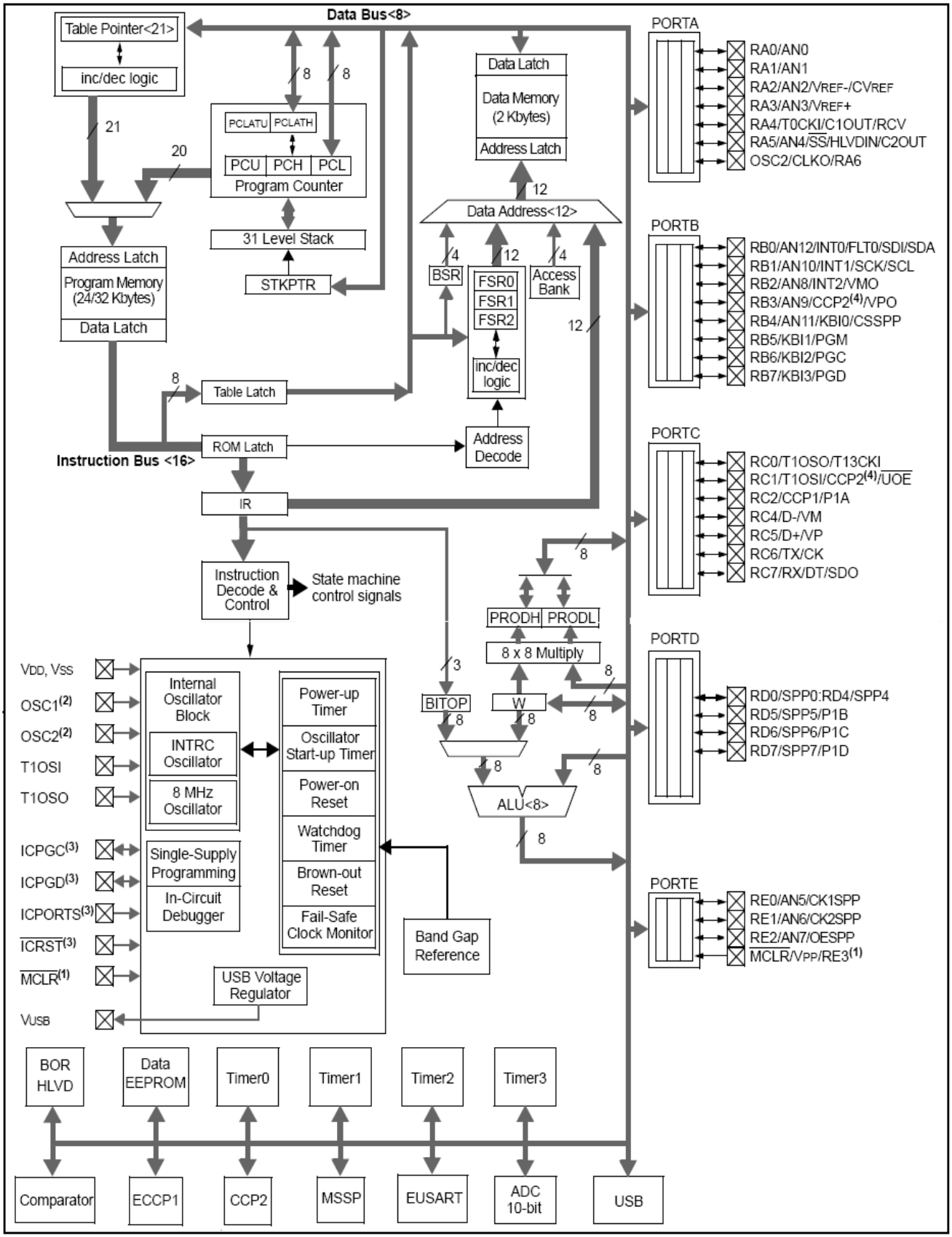


Figure II.5 Architecture interne du PIC 18F4550

II.4.5 Alimentation du PIC 18F4550 :

Pour un fonctionnement normal du PIC, il faut appliquer sur son entrée V_{DD} une tension d'alimentation qui doit être comprise entre 4,2 et 5,25 V. Il peut consommer moins de 1,6 mA sous 5V avec une horloge de 4MHz et 25µA sous 3V avec une horloge 32KHz. Sa mémoire vive ou RAM peut conserver son contenu lorsque le circuit est placé en mode sommeil (une tension d'alimentation qui peut aller jusqu'à 1.5 V). Le positif de la tension d'alimentation est appliqué aux pins V_{dd} du boitier (11 et 32) et la masse aux pins V_{ss} (12 et 31). Les pins d'alimentation doivent être découplés au plus près du circuit au moyen d'un condensateur de 22 à 100 nF de type céramique multicouche.

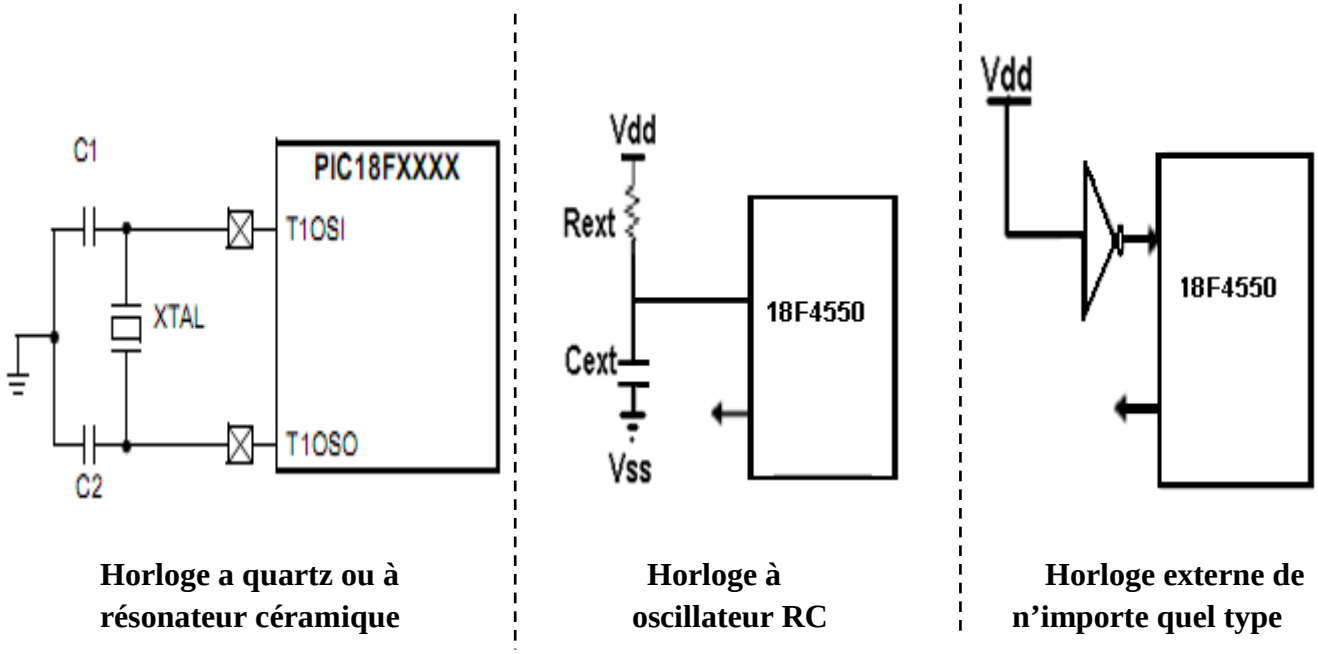
II.4.6 L'horloge :

Le PIC dispose d'une horloge interne qui peut être associé à une horloge externe, utilisable séparément ou les deux au même temps.

II.4.6.a L'horloge interne : associée à un bloc oscillateur, qui peut générer deux horloges de fréquence différentes. Pouvant servir comme horloge principale dans le cas de non utilisation de l'USB.

- La première utilise un oscillateur de fréquence 8 MHz, elle peut être utilisée telle qu'elle est, ou réduite jusqu'à 31 KHz, par un deviseur programmable.
- La deuxième, indépendante de la première, elle fonctionne à 31 KHz et utilise un oscillateur RC interne.

II.4.6.b L'horloge externe : est reliée à OSC1/CLK1 (pin13) et OSC2/CLK2 (pin14) et compte tenu des différents types d'horloges, il existe trois schémas de principes :



Horloge a quartz ou à résonateur céramique | **Horloge à oscillateur RC** | **Horloge externe de n'importe quel type**

Figure II.6 Brochage de l'horloge externe du pic 18F4550

Le PIC 18F4550 peut fonctionner avec huit types d'horloges différentes qui sont :

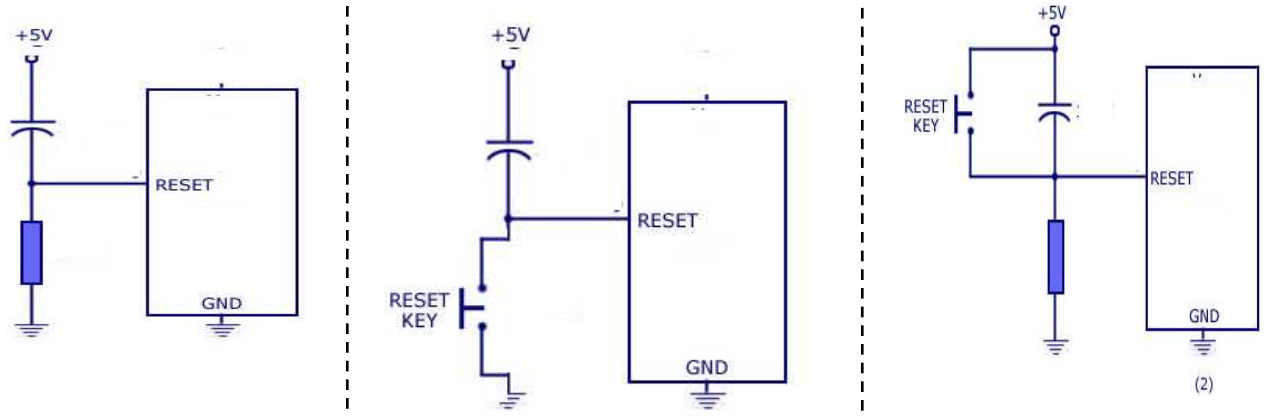
- **LP** (*Low Power*) : horloge à quartz basse vitesse. Elle fait consommer le moins de courant au circuit en raison de sa lenteur.
- **XT** : pour tout quartz de fréquence comprise entre une centaine de KHz et 4MHz.
- **HS** (*High Speed*) : horloge à quartz ou à résonateur céramique haute vitesse. Pour toute fréquence >4MHz.
- **HSPPL :** horloge à quartz ou à résonateur céramique haute vitesse avec utilisation de la boucle de verrouillage de phase PLL interne. (PLL : *phase lockedloop*).
- **RC** ou External RC : Pilotée par un ensemble de résistances et capacités connecté sur l'entrée CLKI. La sortie du signal d'horloge à une fréquence devisé par 4 sur la sortie CLKO.
- **RCIO :** pilotée par un ensemble résistance capacité externe connecté sur l'entrée CLKI tandis que **CLK0 (RA6)** sert d'entrée/sortie à usage générale.
- **EC :** pour l'utilisation d'un signal d'horloge externe appliqué à CLKI avec sortie du signale d'horloge à une fréquence devisée par 4 sur la sortie **CLK0 (RA6).**
- **ECIO :** une source d'horloge externe est appliquée sur CLKI.CLKO (RA6) pin d'E/S du port parallèle.

Pour sélectionner une d'entre elles, l'utilisateur doit configurer 3 bits (FOSC2, FOC1, FOSC0) du registre de configuration (CONFIGIH). La manipulation de ces bits de configuration par l'utilisateur est assurée par tous les logiciels de programmation de Pics sous un panneau de configuration qui demandent de choisir le type d'horloge souhaité (LP,XT,..).

II.4.7 Le circuit Reset :

Le Reset est relativement important. En effet, lorsque le système se plante, il est utile de pouvoir le réinitialiser. Il est également possible et utile parfois d'effectuer un Reset lors de la mise sous tension. Il s'effectue en mettent un «**0**» logique sur la patte 1 du microcontrôleur. Il existe plusieurs types : le Reset manuel, automatique et le mixte.

La figure suivante illustre les différents circuits de Reset :



Reset automatique | **Reset manuel** | **Reset mixte**

Figure II.7 Les différents types de Reset

Le Pic dispose de huit sources potentielles de reset distinctes qui sont :

- Un reset à la mise sous tension du circuit appelé POR (*Power On Reset*).
- Un reset par action sur la pin MCLR alors que le circuit est en mode normal.
- Un reset par action sur la pin MCLR alors que le circuit est en mode sommeil.
- Un reset de débordement du timer chien de garde (*WDT*).
- Un reset par détection d'une chute anormale de la tension d'alimentation, même temporairement, appelé **BOR** (Brown Out reset).
- Un reset suite à une exécution de l'instruction de même nom (reset).
- Un reset suite à une saturation de la pile.
- Un reset suite à un débordement inférieur de la pile (tentative de retirer une donnée de la pile alors qu'elle est vide).

II.4.8 Le Timer chien de garde (WDT) :

C'est un système de surveillance du bon déroulement du programme et de protection contre son blocage. Il s'agit d'un compteur incrémenté au rythme d'une horloge RC interne indépendante de l'horloge système. Ce compteur est réinitialisé régulièrement dans le cas d'un fonctionnement normal du programme. Mais dans le cas d'un dysfonctionnement ou blocage, le compteur va jusqu'au bout et provoque alors un débordement. Deux situations sont possibles :

- Si le µC est en fonctionnement normale, le WDT provoque un RESET.
- Si le µC est en mode SLEEP, le Watch dog Timer provoque un WAKE-UP, l'exécution du programme continue normalement là ou il s'est arrêter.

II.4.9 Le mode SLEEP :

Grâce à l'instruction SLEEP, le PIC est mis en sommeil, afin de limiter sa consommation. Dans ce mode, l'horloge système est arrêtée, ce qui arrête l'exécution du programme. Et pour sortir, il faut provoquer un WAKE-UP, pour cela il y a trois possibilités :

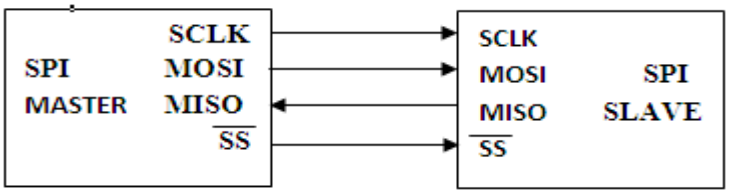
- RESET externe dû à la mise à 0 de l'entrée MCLR du PIC.
- Time out du chien de garde (débordement) si celui-ci est validé.
- Interruption sur RB0, RB1 ou RB2 (si l'interruption est validée).

II.4.10 Le port séries synchrone maître ou MSSP :

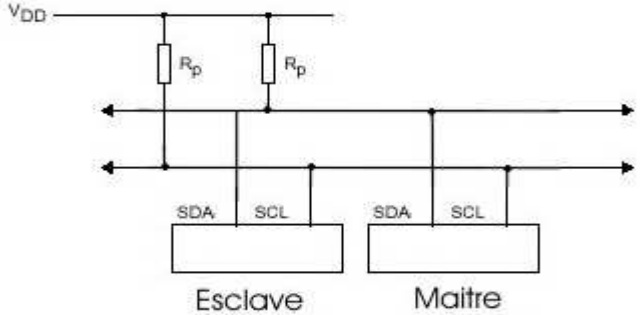
Le mode MSSP (*Master Synchronous Serial Port*) est une interface de communication série avec des périphériques. Ces périphériques peuvent être des mémoires EEPROMs série, des convertisseurs A/D,…. Le module MSSP peut fonctionner soit en mode SPI ou I2C.

II.4.10.a Le MSSP pour SPI (*Serial Peripheral Interface***) :** Il permet d'envoyer et de recevoir des données sur 8 bits d'une manière synchrone et simultanée. Pour assurer la communication, trois pins sont utilisés :

- Sortie série de données (SD0) – RC7.
- Entrée série de données (SDI) – RB0/SDA.
- Horloge série (SCK) – RB1/SCL/LVDIN.



II.4.10.b Le MSSP pour I2C ***(Inter Integrated Circuit)*** **:** Il permet de faire des transmissions au format I2C en tant que maître ou esclave. Il utilise pour cela seulement deux fils, le premier (SDA) pour le transfert de données (pin 33) et l'autre (SCL) pour l'horloge (pin 34).

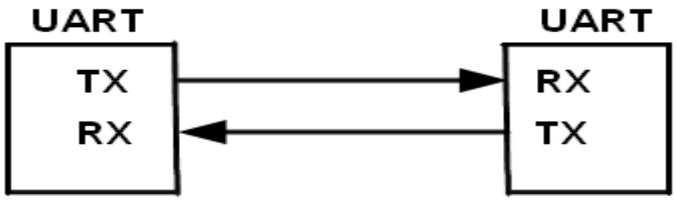


II.4.11 L'USART (Universal Synchronous Asynchronous Receiver Transmitter) :

Appelé aussi SCI (Serial Communication Interface). C'est un module de transmission série qui peut fonctionner comme une interface série asynchrone classique en full duplex, afin de communiquer avec des dispositifs utilisant le même mode comme par exemple un micro-ordinateur, ou comme une interface série synchrone en half-duplex afin de communiquer avec des dispositifs externe comme les mémoires série EEPROMs, il utilise la pin RC6/TX pour l'émission de données et RC7/DX pour la réception.

L'USART peut être configuré comme une interface :

- Asynchrone (*Full-duplex*).
- Synchrone-Master (*Half-duplex*).
- Synchrone-Slave (*Half-duplex*).



II.4.12 L'USB (*Universal Serial Bus)*:

L'USB, Universal Serial Bus est la norme fonctionnant en mode série, permettant la connexion de périphérique externe, compatible plug-and-play. La version 2.0 permet des débits jusqu'à 480 Mbps. Elle utilise la pin 24 (DATA+) pour la l'émission et la pin 23 (DATA-) .pour la réceptions des données.

II.4.13 Convertisseur Analogique/Numérique :

Il est constitué de 13 entrées analogiques, les 8 premières sont sur le **port A** (RA0-RA7), les 5 autres sont sur le **port E (**RB0-RB4**).**

Le résultat de la conversion est codé sur 10 bits. Le convertisseur A/D fonctionne même si le PIC et en mode sommeil, la conversion s'effectue en utilisant une horloge RC interne.

II.4.14 Les Timers:

Tous les microcontrôleurs ont un ou plusieurs Timers pour la gestion du temps. Le PIC18F4550 possède 04 Timers (8bits/16bits) : timer0, timer1, timer2 et timer3.

Le Timer est un circuit périphérique (inclut dans le PIC) lancé par le programme et qui compte un nombre prédéfini d'impulsions du quartz. Ces impulsions peuvent provenir :

• Du quartz cadençant le microcontrôleur : on parle alors d'un fonctionnement en compteur de temps.

• D'une entrée TOR du pic (exemple RA4 pour le Timer0, RC0 pour le Timer 1) on parle alors d'un fonctionnement en compteur d'événement extérieur.

Quand ce nombre d'impulsion est atteint, un **drapeau** (flag en anglais) se **lève**.

Les Timers ont presque tous le même fonctionnement. Ils ont tous un **compteur** (counter en anglais), un **pré-compteur** (prescaler en anglais).

II.5 Le rôle des jumpers de la carte :

II.5.1 Les jumpers SMD :

- **B4_E** : se jumper connecte le bouton 4 avec la pin RB6 du PIC 18F4550, par défaut le jumper et fermer.
- **B3_E** : se jumper connecte le bouton 3 avec la pin RB7 du PIC 18F4550, par défaut le jumper et fermer.

II.5.2 Les jumpers PTH :

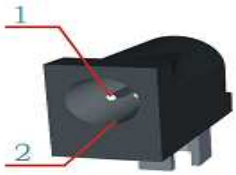
- **EXT/USB** : se jumper définie la source d'alimentation de la carte :
  - La position EXT, la carte sera alimentée par 6V AC ou par 9V DC.
  - La position USB, la carte sera alimentée directement par 5V DC.
- **CP/B2 :** se jumper définie le fonctionnement de la pin 37 (RB4) :
  - La position **CP** connecte le pin 37 du Pic avec le périphérique de stockage (carte mémoire) SD/MMC.
  - La position **B2** connecte la pin 37 du Pic avec le bouton 2 de la carte, se dernier permet de booté la carte dans le car d'utilisation de Bootloader. Par défaut le jumper est sur la position B2.

II.6 Description des connecteurs externe de la carte :

❖ **POWER JACK :**

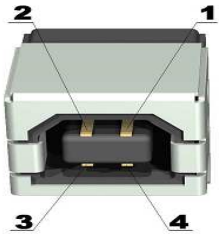
C'est un connecteur d'alimentation pour la carte (9V DC / 6V AC).

PIN #	Nom du signal
1	input
2	GND



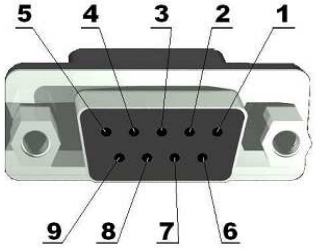
❖ **USB type B :**

PIN #	Nom du signal
1	+5V
2	Data +
3	Data -
4	GND



❖ **RS232 (femelle) :**

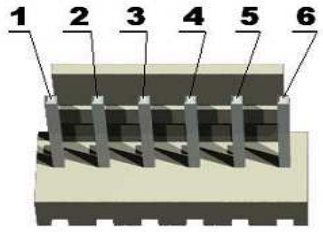
NOM #	Nom du signal
1	Non connecter
2	TX Out
3	RX In
4	Non connecter
5	GND
6	Non connecter
7	Connecter à R2IN
8	Connecter à T2Out
9	Non connecter



Dans notre projet on a utilisé la liaison RS232 pour relier la carte de développement à un PC.

- **ICSP :**

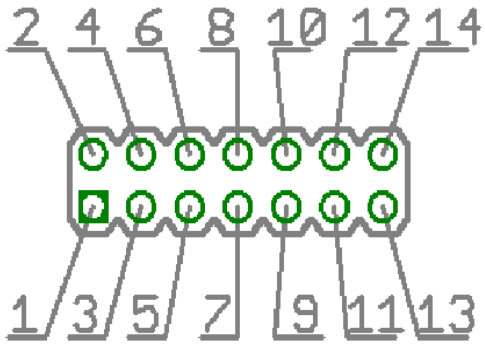
PIN#	Nom du signal
1	Reset
2	+5V
3	GND
4	PGD
5	PGC
6	Non connecter



Le connecteur standard ICSP est utilité dans toutes les cartes d'OLIMEX, généralement pour charger le Bootloader.

- **Les broches d'extension :**

PIN #	Nom du signal
1	5V
2	VIN
3	GND
4	3.3V
5	RB2
6	RE0
7	RE1
8	RE2
9	RD7
10	RD6
11	RD5
12	RD4
13	RC0
14	RC2



Chapitre III

Description des capteurs

Introduction :

Les capteurs sont les premiers éléments d'une chaine d'acquisition. Ils transforment une grandeur physique d'un processus ou d'une installation en signaux électriques exploitables. Dans notre projet nous avons utilisés trois capteurs numériques pour l'acquisition des données.

- Un capteur de température de type DS18B20.
- Un capteur de vitesse de type photo-détecteur.
- Un capteur d'accélération de type accéléromètre BMA180.

Le choix de ces capteurs numériques dans notre projet revient pour leurs basses consommations en courant, la simplicité d'implantation (moins de fils et sans circuit de conditionnement), leurs temps de réponse appréciable par rapport aux capteurs analogiques [5].

III.1 Capteur de température DS18B20 :

Le DS18B20 de DALLAS est un capteur de température numérique, qui peut mesurer une température comprise entre -55°C et +125°C, il possède une résolution de 9, 10, 11, ou 12 bits, ce qui correspond à une précision de 0.5°C, 0.25°C, 0.125°C, et 0.0625°C respectivement, les données sont converties directement en numérique avec un temps de réponse de 0,75 secondes.

Une autre particularité du DS18S20 est de pouvoir fonctionner sans aucune alimentation externe. L'alimentation est fournie par la résistance de tirage par la broche DQ (voir la figure III.3) quand le bus est au niveau haut. Ce niveau du bus charge également une capacité interne (Cpp) qui alimente ensuite le composant quand le bus est au niveau bas. Cette méthode de dériver l'alimentation en énergie est appelée mode « parasite power », il peut aussi être alimenté de façon traditionnelle par sa broche V_{CC}, la plage d'alimentation est de 3 à 5,5V. Il démarre à sa mise sous tension par un état de veille en basse consommation et la valeur du registre température est fixée à +85°C.

Le DS18B20 utilise le bus exclusif ONE-Wire qui permet une communication sur un seul fil.

III.1.1 Brochage du DS18B20 :

Les figures suivantes montrent les différents boitiers existant du DS18B20 .

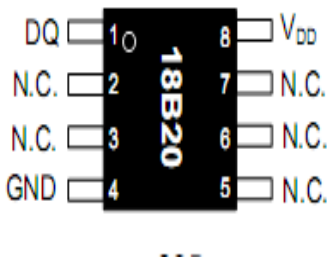


Figure III.1 Boitier 8 pins

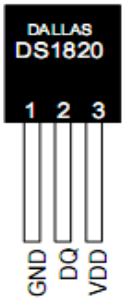


Figure III.2 Boitier TO92

Le tableau suivant montre le branchement des pins du DS18B20.

	Boitier TO92	Boitier 8 pins	symbole	Description
	1	**4**	**GND**	**la masse**
N.C: Non connecter	**2**	**1**	**DQ**	**DATA Input/ output**
	3	**8**	**V_{DD}**	**Source de tension**

Tableau III.1 Branchement des pins du DS18B20

III.1.2 Le schémas bloc du DS18B20 :

La figure suivante montre l'architecture interne du capteur :

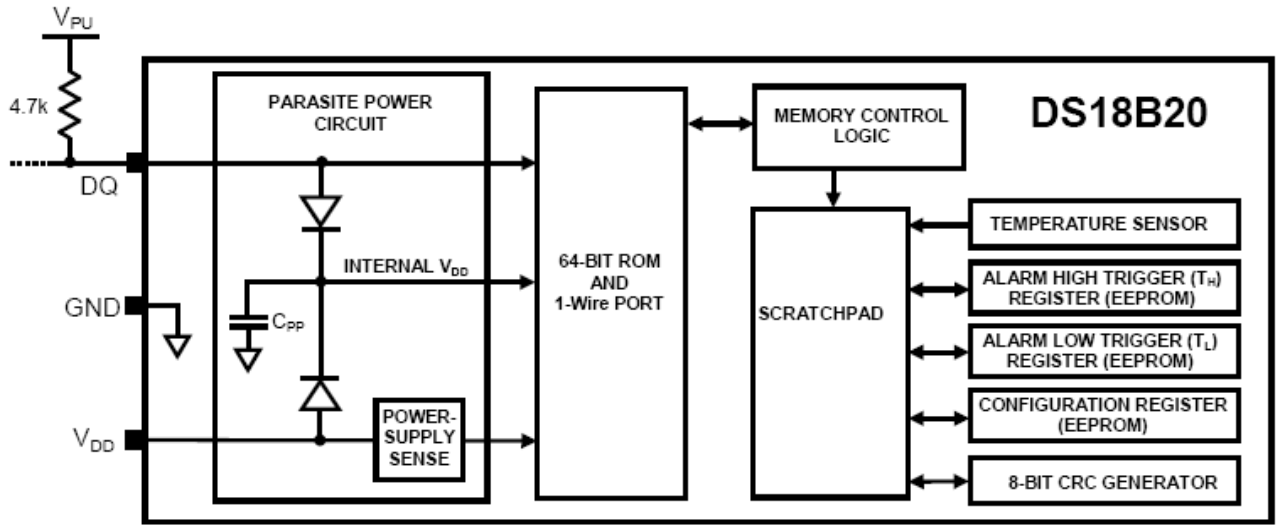


Figure III.3 Architecture interne du DS18B20

III.1.3 Opération de mesure de la température :

La fonctionnalité essentielle du DS18S20 est de fournir directement une valeur numérique de la température. Cette donnée est stockée dans le registre de température sous la forme d'un nombre en complément à deux au format 16 bits. Les bits de signe < S > indiquent si la température est positive (S= NL0) ou bien négative (S=NL1).

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LS BYTE	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
	BIT 15	**BIT 14**	**BIT 13**	**BIT 12**	**BIT 11**	**BIT 10**	**BIT 9**	**BIT 8**
MS BYTE	S	S	S	S	S	2^6	2^5	2^4

S = SIGN

Figure III.4 Registre de température du DS18B20

Le tableau suivant montre les relations entre la température et les données :

TEMPERATURE (°C)	DIGITAL OUTPUT (BINARY)	DIGITAL OUTPUT (HEX)
+125	0000 0111 1101 0000	07D0h
+85*	0000 0101 0101 0000	0550h
+25.0625	0000 0001 1001 0001	0191h
+10.125	0000 0000 1010 0010	00A2h
+0.5	0000 0000 0000 1000	0008h
0	0000 0000 0000 0000	0000h
-0.5	1111 1111 1111 1000	FFF8h
-10.125	1111 1111 0101 1110	FF5Eh
-25.0625	1111 1110 0110 1111	FE6Fh
-55	1111 1100 1001 0000	FC90h

Tableau III.2

III.1.4 La mémoire du DS18B20 :

La mémoire est constituée de ram statique SRAM pour la partie bloc-notes (Scratchpad) et de mémoire non volatile EEPROM pour les registres de seuils d'alarme TH et TL. A noter que si la fonctionnalité d'alarme n'est pas utilisée, ces registres peuvent servir de mémoire à usage général.

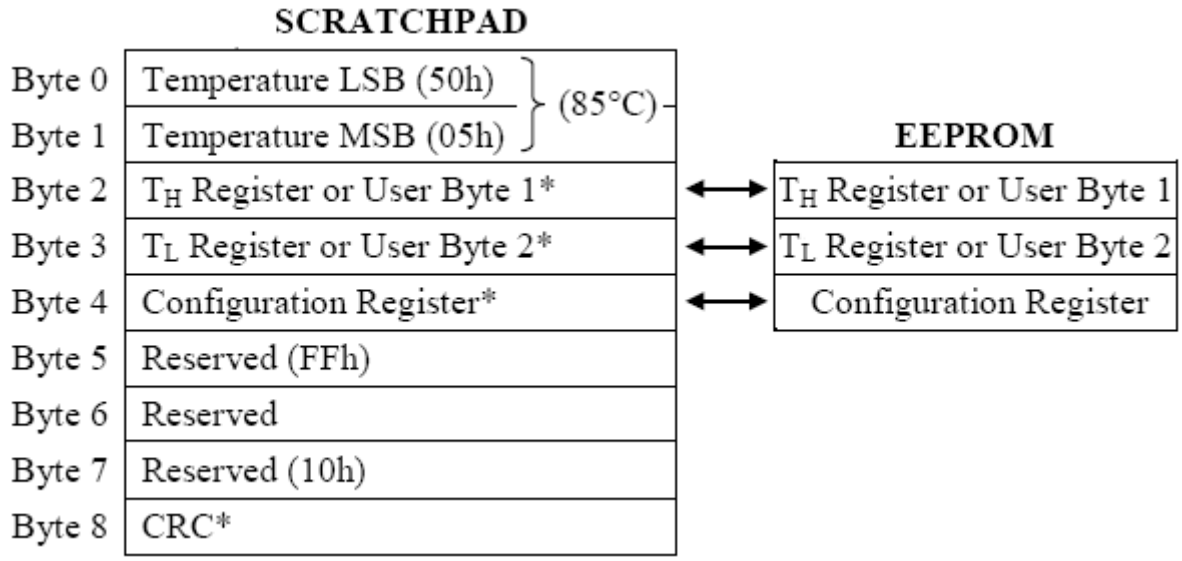


Figure III.5 Organisation de la mémoire du DS18B20

- Le bit 0 et 1 de la mémoire bloc-notes constituent les poids faibles et forts (respectivement) du registre 16 bits de température.
- Le bit 2 et 3 de la mémoire bloc-notes constituent les registres d'alarme TH et TL.
- Le bit 4 contient les données de registre de configuration.
- Le bit 5, 6 et 7 sont réservés pour l'usage interne par le dispositif.
- Le bit 8 est en lecture seule et contient le Code de Redondance Cyclique (CRC) calculé en fonction du contenu des octets 0 à 7 de la mémoire bloc-notes.

III.2 Le bus One-Wire :

Le bus 1-Wire (One-Wire) est conçu par Dallas Semi-conducteur, il permet de connecter des composants avec seulement deux fils. Le niveau de tension utilisé sur ce bus est +5V. Il supporte une topologie série, parallèle ou en étoile, et fonctionne suivant le principe maître / esclave.

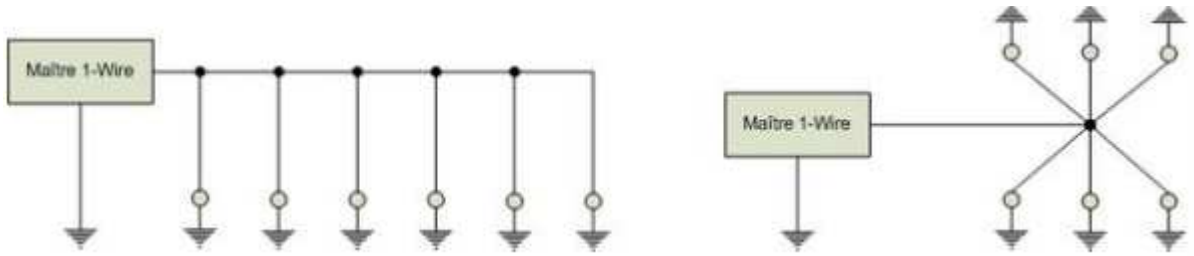


Figure III.6 Les différentes topologies utilisé par 1-Wire

L'avantage de ce bus est qu'il peut être utilisé en mode « parasite » (alimentation à partir du fil de données). Cela permet d'utiliser seulement 2 fils (et non un seul comme le nom le laisse supposer), un fil de données et un fil de masse.

III.2.1 Protocole one-Wire :

La communication sur le bus One-Wire est caractérisée par un ensemble d'impulsions «changement d'état du bus ». Sachant que l'état par défaut de la ligne data est +5V, ce qui permet d'alimenter les différents composants à partir de la ligne data en mode parasite.

Avant toute communication, le maître met le bus à 0 pendant 480µs, tous les composants sur le bus sont remis à zéro, c'est l'impulsion d'initialisation ou de reset. Après un délai de 15 à 60µs, le ou les esclaves raccordés, forcent le bus à l'état bas pendant 60 à 240µs pour signaler leur présence.

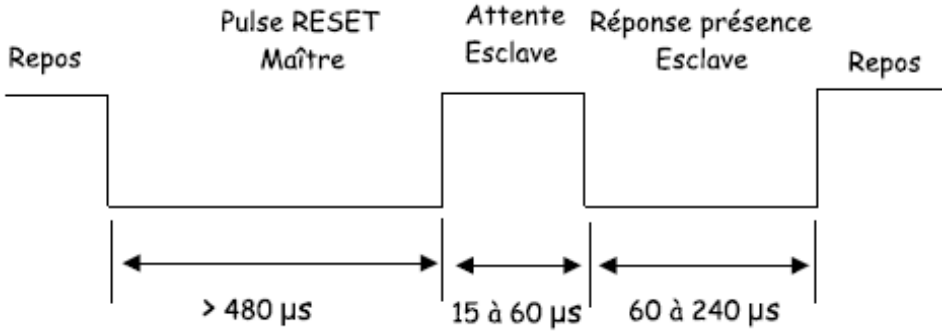


Figure III.7 Chronogramme du bus One-Wire

III.2.2 Emission d'un bit du maître vers l'esclave :

Le maître force le bus à "0" pendant 1 à 15µs. L'esclave va lire le bus entre 15 et 45µs après le front descendant. Si on veut émettre un "1", il faut repasser le bus à "1" immédiatement, et ne plus rien faire jusqu'à t = 60µs. Pour émettre un "0" il faut laisser le bus à "0" jusqu'à t = 60µs, puis repasser le bus à "1". La durée du bit est donc de 60µs, ce qui donne un débit de 16 kbits/sec.

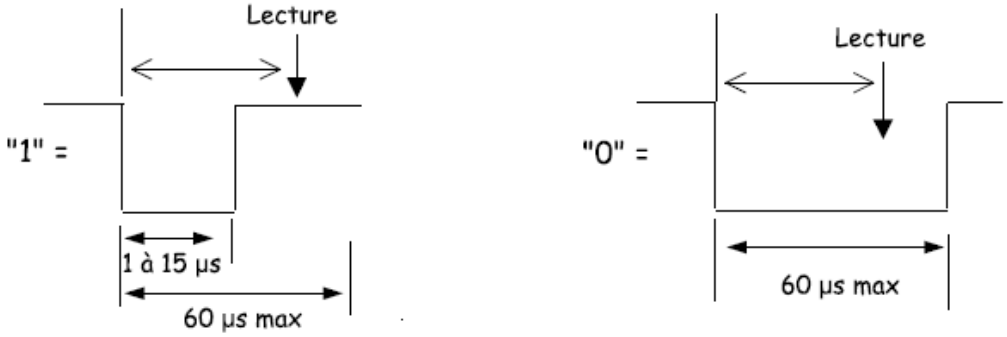


Figure III.8 Emission d'un bit du maître One-Wire

III.2.3 Réception d'un bit par le maître :

Le maître force le bus à "0" pendant au moins 1µs. Si l'esclave veut émettre un "1", il laisse le bus libre donc tiré à "1". Pour émettre un "0", l'esclave doit tirer le bus à "0" pendant 15µs. Le maître devra donc dans tous les cas lire le bus 15µs maximum après avoir tiré le bus à "0" pendant 1µs. L'état du bus donnera alors le bit.

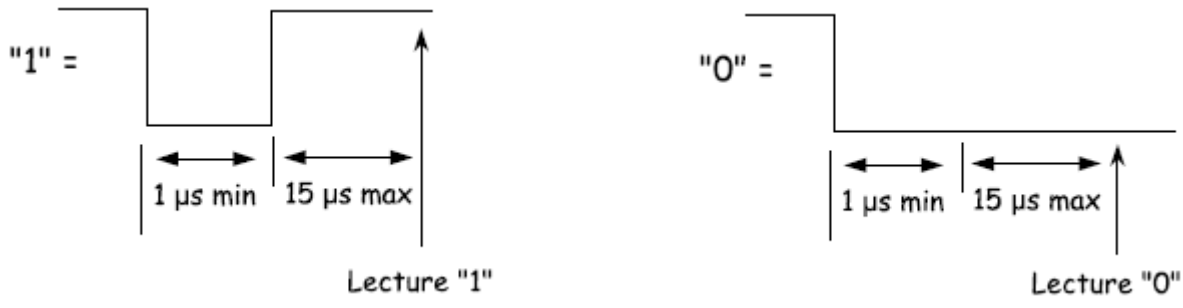


Figure III.9 Réception d'un bit du maître One-Wire

III.3 Capteur d'accélération BMA180 :

L'accéléromètre BMA180 de Bosch est un capteur triaxial numérique d'ultra haute performance, destiné à des applications à faible consommation de puissance. Le BMA180 permet la mesure précise des accélérations dans les trois axes perpendiculaires (X, Y, Z) et par conséquent détecte le sens d'inclinaison, le mouvement, chocs et vibrations. Comme il permet la mesure de la température comprise entre -40° et +85°C. Il fournit une sortie de 14 bits numérique soit par une interface SPI à 4 fils ou une interface I2C à 3 fils.

Le BMA180 peut être alimenté avec une tension comprise entre 1,62 et 3,6 V avec une consommation en courant de 650µA en mode normale.

Dans notre projet nous avons utilisé l'interface I2C pour la communication entre le capteur et la carte d'acquisition d'OLIMEX (*paragraphe III.4*).

III.3.1 Brochage du BMA180 :

La figure III.10 montre le boitier utilisé pour le BMA180.

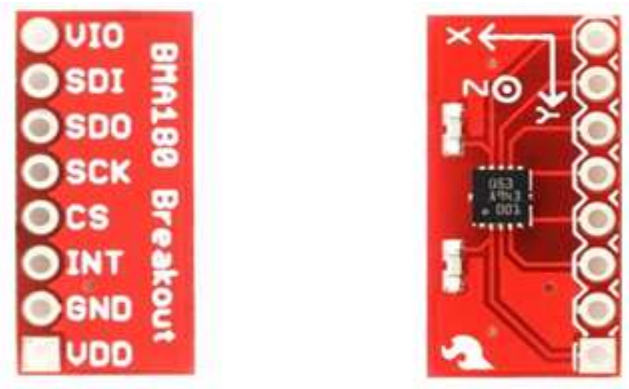


Figure III.10 Face avant (à droite)et face arriere (à gauche) du BMA180

La figure suivante représante le circuit éléctrique du BMA180.

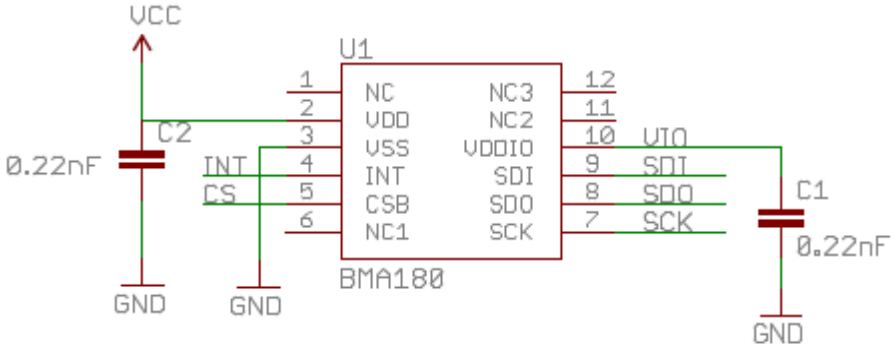


Figure III.11 Circuit éléctrique du BMA180

On remarque la présence de deux condensateurs de 22 nf (C1 et C2) pour le découplage de l'alimentation.

III.3.2 Bloc diagramme du BMA180 :

Le bloc diagramme du BMA180 est composé de :

- Deux capteurs le premier pour la température et le second pour l'accélération.
- Deux préamplificateurs du signal d'entré (analogique).
- Un multiplexeur.
- Un convertisseur analogique/numérique de 14 bits.
- Un bloc numérique responsable du filtrage numérique, calibrage.
- Deux interfaces de sorties (SPI et I2C).
- Une sortie d'interruption.

La figure suivante illustre le bloc diagramme de l'accéléromètre BMA180.

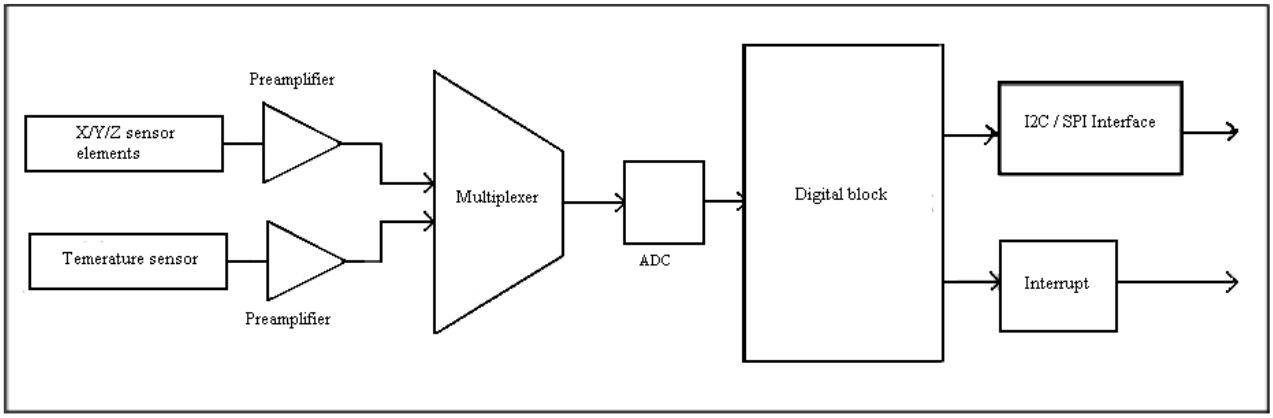


Figure III.12 Bloc diagramme du BMA180

III.3.3 Map mémoire global du BMA180 :

Le map global fournie trois niveaux d'accès à la mémoire. Le tableau suivant représente les différents principaux registres de mémoire ainsi que leurs contenus.

La région mémoire	***Contenue***	***Niveau d'accès***
Registres opérationnels	Registre de données, de control et de statu. Paramètre d'interruption	L'interface série (SPI ou I2C)
Registre par défauts	Réglage de la valeur de température et d'accélération, valeur par défaut du registre opérationnels.	Accès fermer par défaut, mais on peut lui accéder par l'interface série.
Registre réserver au constructeur	Réglage interne	Protéger

Tableau III.13 Map mémoire global du bma180

la mémoire de BMA180 est réalisée dans des architectures physiques diverses. Fondamentalement le BMA180 utilise les registres de mémoire volatile. La partie non_volatile de la mémoire (EEPROM) est utilité pour charger les valeurs par defaut du capteur.

Après chaque reset du BMA180 les valeurs du registre EEPROM sont copiées vers le registre de la mémoire volatile.

III.4 Le bus I2C :

Le bus I2C (**I*nter* I*ntegrated* C*ircuit***) fait partie des bus série avec 2 fils de données et un fil de masse. Il a été développé au début des années 80 par ***Philips*** pour minimiser les liaisons entre les circuits intégrés numérique, et pour cela il utilise :

- Un fil de données (SDA).
- Un fil d'horloge (SCL).
- Un fil de référence électrique (masse).

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.

Le nombre de composants qu'il est ainsi possible de relier est essentiellement limité par la charge capacitive des lignes SDA et SCL (400 pF).

III.4.1 Le protocole I2C :

Le protocole I2C définit la succession des états logiques possibles sur SDA et SCL, et la façon dont doivent réagir les circuits en cas de conflits.

Avant de prendre le contrôle du bus I2C un circuit doit vérifier que les lignes SDA et SCL sont au repos, c'est dire à l'état haut '1'. Pour transmettre des données sur le bus, il faut surveiller deux conditions particulières : La condition du départ et la condition d'arrêt.

- La condition de départ : SDA passe à '0' alors que SCL reste à '1'.
- La condition d'arrêt : SDA passe à '1' alors que SCL reste à '1'.

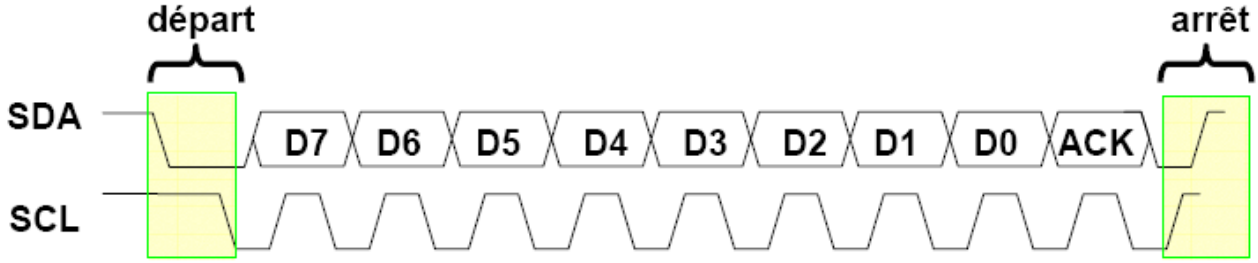


Figure III.14 Condition de départ et d'arrêt pour le bus I2C

III.4.2 La transmission d'un octet :

Après avoir imposé la condition de départ, le maître applique sur SDA le bit de poids fort D7. Il valide ensuite la donnée en appliquant pendant un instant un niveau '1' sur la ligne SCL. Lorsque SCL revient à '0', il recommence l'opération jusqu'à ce que l'octet complet soit transmis. Il envoie alors un bit ACK à '1' tout en scrutant l'état réel de SDA. L'esclave doit alors imposer un niveau '0' pour signaler au maître que la transmission s'est effectuée correctement.

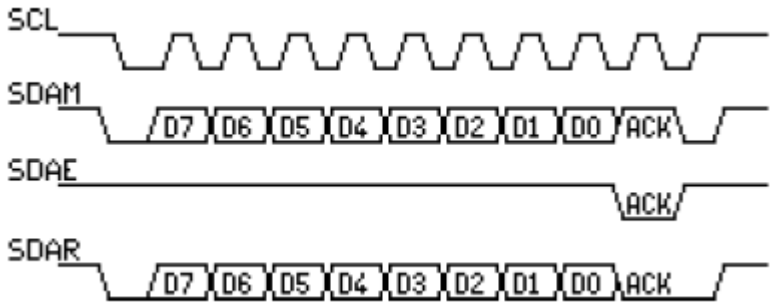


Figure III.15 Chronogramme de transmission d'un octet

III.4.3 La transmission d'une adresse :

Le nombre de composants qu'il est possible de connecter sur un bus I2C étant largement supérieur à deux, il suffit juste de définir pour chacun une adresse unique.

L'adresse du circuit est codée sur sept bits, et définie d'une part par son type et d'autre part par l'état appliqué à un certain nombre de ces broches. Cette adresse est transmise sous la forme d'un octet au format particulier.

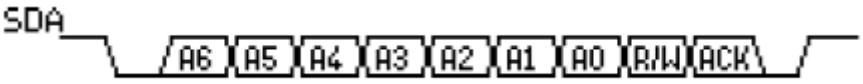


Figure III.16 Chronogramme de transmission d'une adresse

On remarque ici que les bits D7 à D1 représentent les adresse A6 à A0, et que le bit D0 et remplacé par le bit de R/W qui permet au maître de signaler s'il veut lire ou écrire une donnée. Le bit d'acquittement ACK fonctionne comme pour une donnée, ceci permet au maître de vérifier si l'esclave est disponible.

III.5 Mesure de la vitesse de rotation du moteur :

III.5.1 Principe de la manipulation :

Le but étant de mesurer la vitesse de rotation du moteur grâce à un compteur. Des impulsions récupérées seront additionnées, dans un intervalle de temps bien déterminé, pour mesurer la vitesse du moteur correspondante.

Ces impulsions proviennent d'un circuit que nous avons conçu, il est basé sur l'utilisation d'une LDR (photorésistance) comme récepteur optique, avec un émetteur à base d'une diode led. La maquette de la figure III.17 illustre le principe.

Sur l'axe du moteur on accroche un disque troué permettant, une fois aligner au dispositif émetteur/récepteur, de laisser passer le faisceau de lumière issue de la led, pour rejoindre le récepteur de l'autre côté.

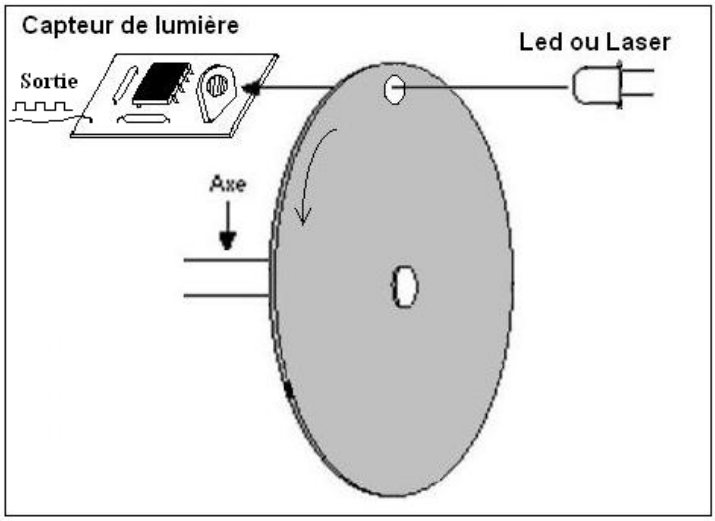


Figure III.17 Principe du compteur d'impulsions

III.5.2 Circuit utilisé :

Le circuit réalisé pour permettre la mesure de la vitesse du moteur est représenté sur la figure ci-dessous, on remarque la présence du Circuit intégré LM311 qui joue le rôle d'un comparateur pour délivrer la tension adéquate à la sortie. La résistance R5 est ajustée pour augmenter la sensibilité de la LDR pour avoir un meilleur temps de réponse.

La figure III.18 représente le circuit électrique utilisé pour le récepteur.

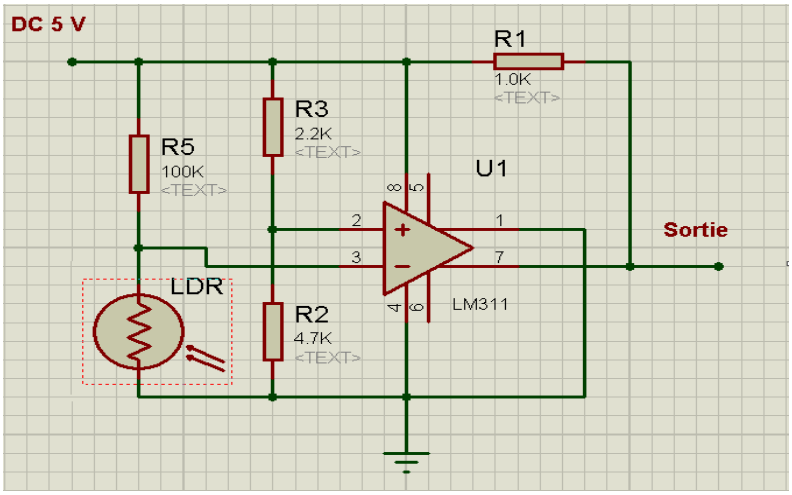


Figure III.18 Circuit récepteur

III.5.3 Principe de fonctionnement :

Le circuit est construit autour du circuit intégré LM311. C'est un circuit intégré spécialisé dans la comparaison analogique de tensions.

La fonction comparaison permet de fournir en sortie un signal dont la valeur est fonction de la comparaison des deux signaux d'entrées issues des broches 2 et 3.

La tension dans la broche 2 étant constante et vaut 3,13 V, celle de la broche 3 varie en fonction de l'état de la LDR comme suit :

- Si elle reçoit de la lumière sa tension sera de 2,57 V.
- Si elle ne reçoit pas de la lumière elle vaudra 3,6 V.

Le signal de sortie alors, peut prendre 2 états différents, selon l'état de la LDR. Dans notre cas les deux tensions de sorties sont 0,17 et 4,6 V.

Si V(2) > V(3) ⟹ Vs = 4,6 V.

Si V(2) < V(3) ⟹ Vs = 0,17 V.

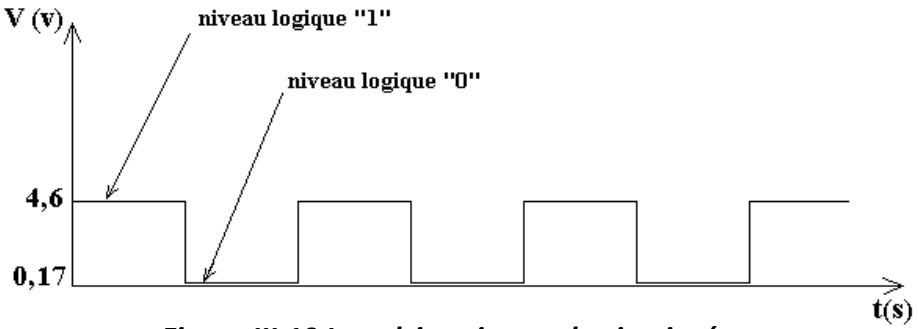


Figure III.19 Impulsions issues du circuit récepteur

Chapitre IV

Conception logicielle et réalisation

IV.1 Environnement de développement :

IV.1.1 Présentation de l'environnement de développement :

Mikroelektronika propose un environnement de développement adapté aux microcontrôleurs PICs de Microchip. Il existe actuellement sous Trois versions différentes, selon le compilateur incorporé :

- MikroBasic : Avec comme langage de programmation le Basic.
- Mikropascal: Avec le langage pascal.
- MikroC : Avec le langage C.

Afin de faciliter la tâche à l'utilisateur (programmeur). Ces compilateurs contiennent des librairies et des drivers qui peuvent être inclus dans le programme ainsi que des fonctions structurelles qui permettent l'accès aux périphériques du microcontrôleur. Un support d'aide très riche et des exemples d'application sont aussi présentés pour permettre aux utilisateurs de développer des applications complexes.

Lors de l'installation, le compilateur nous propose d'installer les pilotes des cartes de développement Easypic ainsi que « mikroProg Suite » pour leurs programmations.

La figure ci-dessous représente la fenêtre principale du compilateur MikroC utilisé dans notre projet.

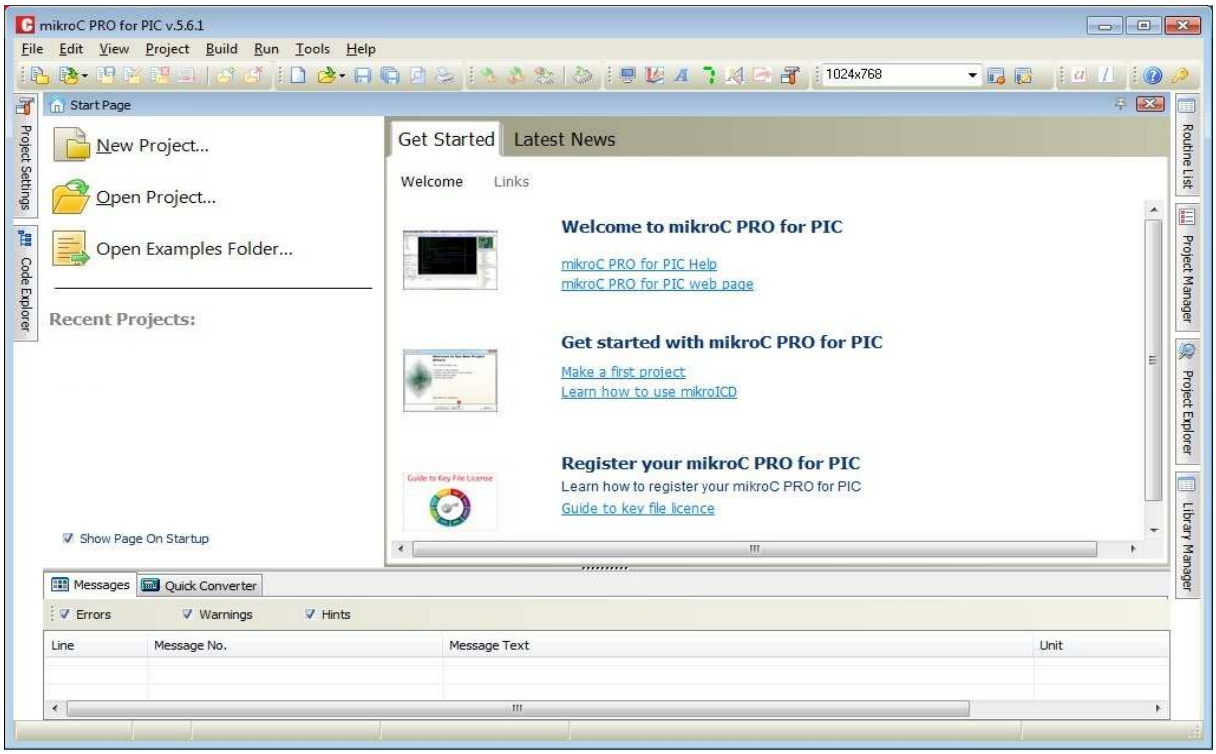


Figure IV.1 La fenêtre principale du compilateur MikroC

IV.1.2 Création d'un projet sous MikroC :

Dans le menu principale de MikroC on sélectionne le menu PROJECT >NEW PROJECT, une fenêtre apparait. Pour démarrer un nouveau projet il suffit d'un simple clic sur le bouton NEXT.

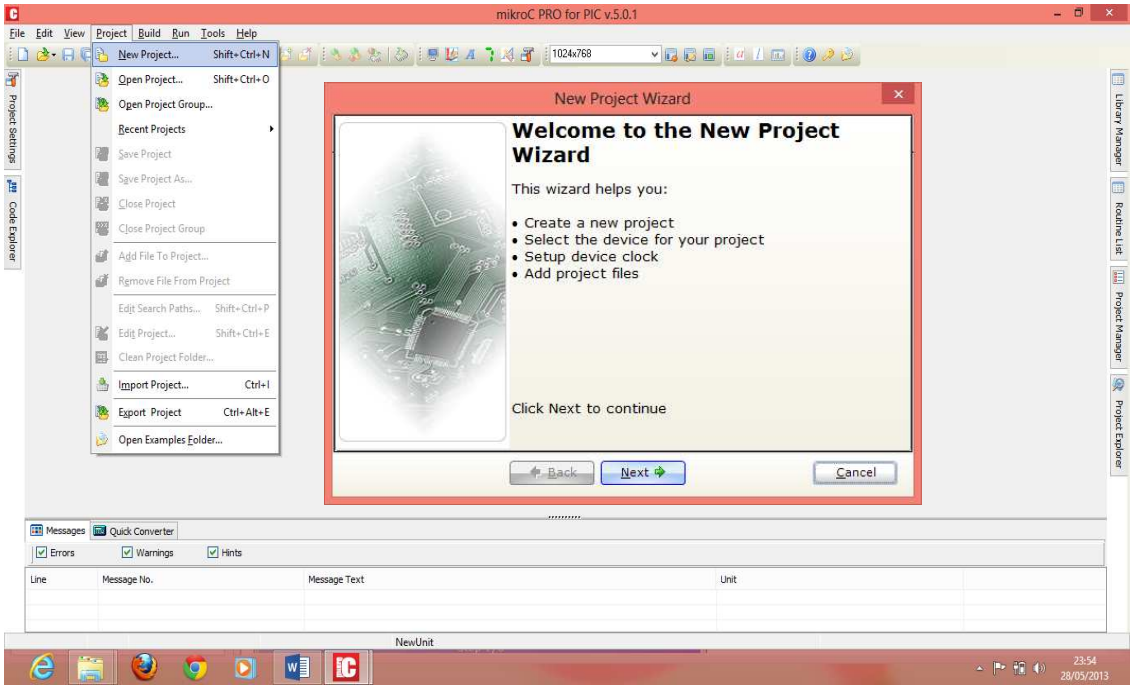


Figure IV.2 Création d'un projet sous mikroC

La fenêtre suivante nous propose de saisir un nom au projet créer, son l'emplacement, le pic à utiliser ainsi que la fréquence de l'oscillateur externe.

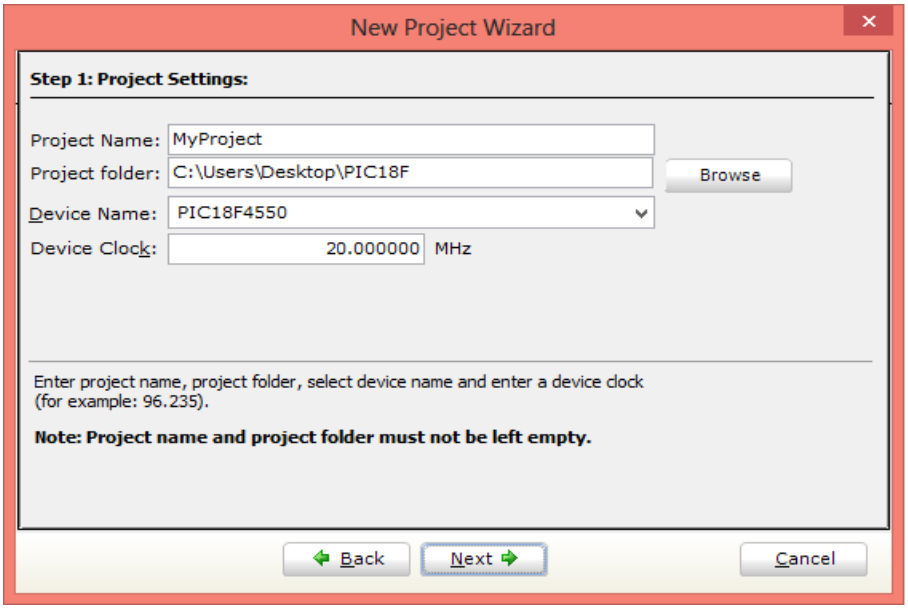


Figure IV.3 Les paramètres d'un nouveau projet

En cliquons sur Next jusqu'à la fin, on obtient ainsi un nouveau projet près et avec toutes les librairies (Bibliothèques) disponible sur le compilateur.

On aperçoit l'apparition des barres d'outils latérales proposant :

- le choix des bibliothèques à utilisées et la gestion du projet,
- la configuration du projet.
- l'explorateur de code qui permet de structurer une hiérarchie des fonctions saisie par l'utilisateur pour permettre la gestion efficace du déroulement de programme.

IV.1.3 Les Principaux Boutons utilisés :


Permet de compiler le programme.


Programmer le PIC (Cas d'une carte de développement Easypic.)

L'onglet « MESSAGE » en bas de la fenêtre nous permet de visualiser l'évolution du processus de compilation et le rapport de fin, ainsi que les éventuelles erreurs commises lors de la programmation.

Une fois les erreurs corrigées, le compilateur génère un fichier d'extension **.hex** dans le dossier du projet.

On utilisera un programmateur externe pour programmer le pic avec le fichier hex générer.

Dans le cas d'une carte de développement Easypic, une fois connecté, il suffit de presser le bouton PROGRAM pour programmer la carte.

L'onglet «Quick Converter » nous permet de convertir les données d'un format vers un autre par exemple un format binaire vers hexadécimal, décimale…etc.

Le menu «Tools » nous donne une variété de choix pour accéder à des fonctionnalités tel que la table ASCII pour la gestion de la communication série RS232, l'édition du contenue de l'EEPROM, le terminal virtuel etc.…

IV.2 L'Interface graphique sous LabVIEW:

IV.2.1 Présentation du logiciel LabVIEW :

LabVIEW (***La**boratory **V**irtual **I**nstrument **E**ngineering **W**orkbench*) est un logiciel de développement graphique (***langage G***) qui fait appel à des symboles (icônes) pour décrire les Opérations. Il répondant à la même finalité que les nombreux systèmes de développement en langage C ou Basic.

C'est un logiciel de développement d'applications appelées "Instruments Virtuels" ***(VI)*** car leur apparence et fonctionnement ressemblent aux instruments réels.

La programmation avec LabVIEW est **intuitive** et s'apprend rapidement.

IV.2.2 Créer un nouveau programme sous LabVIEW:

Lorsqu'on lance LabVIEW  , l'écran de démarrage permet d'utiliser toutes les opérations proposées par cet environnement de développement. La figure suivante montre la fenêtre principale de LabVIEW :

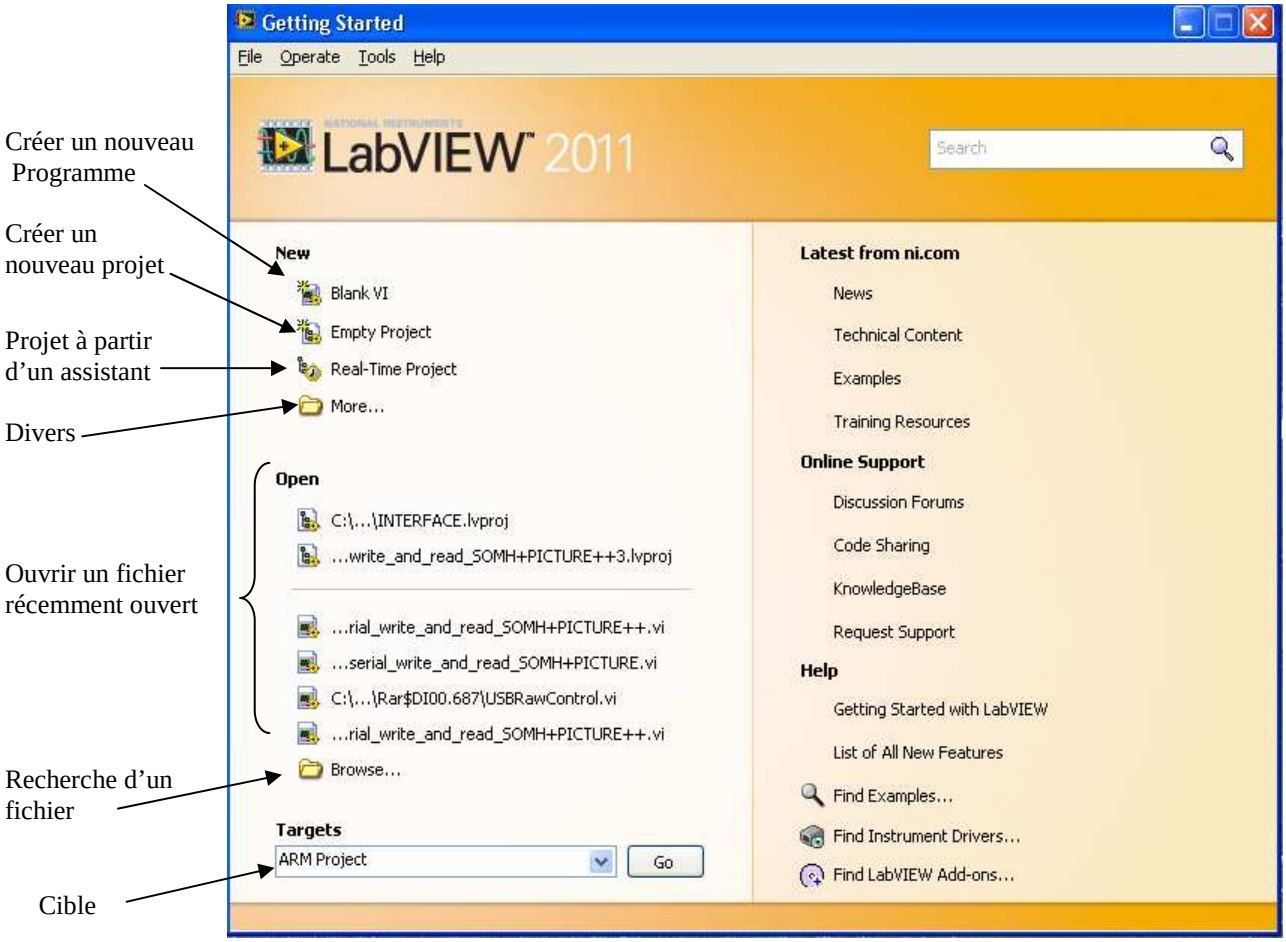


Figure IV.4 Fenêtre principale de LabVIEW

Pour créer un nouveau programme on sélectionne « Blank VI ». Les programmes auront l'extension « .vi ».

Un ***vi*** est composé d'un diagramme (fenêtre blanche) et d'une face avant (fenêtre grise) et représenté par une icône. La figure suivante montre la face avant et la face diagramme d'un programme sous LabVIEW.

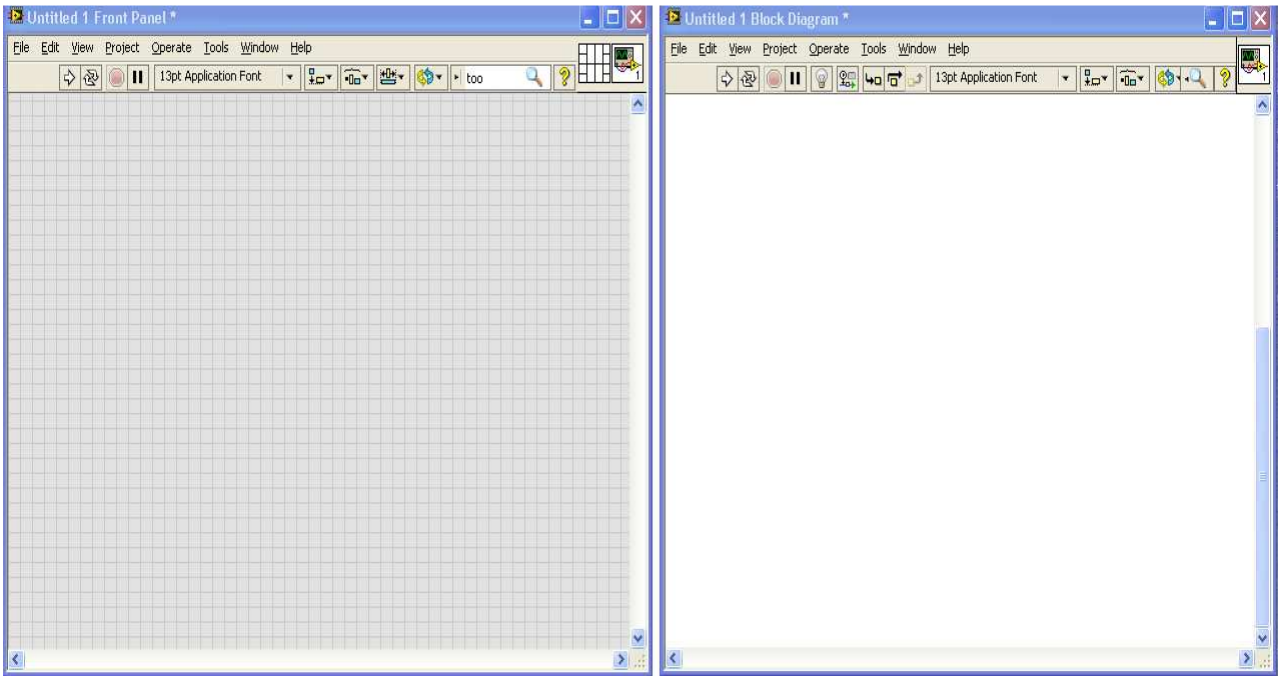


Figure IV.5 Face avant (à gauche) et Diagramme (à droite)

Sur la face-avant on place les éléments graphiques et sur le diagramme on place la logique du programme, en général la logique qui relie les entrées aux sorties.

IV.2.3 Les palettes :

En LabVIEW, toute la programmation se passe de façon graphique, il n'y a pas de syntaxe à saisir. On commence souvent par créer la face-avant, puis on passe au diagramme pour représenter la logique du programme. Et pour cela le logiciel propose trois types de palettes différentes.

- **Palette d'outils :**

La palette d'Outils est disponible sur la face-avant et le diagramme. Un outil est un mode de fonctionnement spécial du curseur de la souris. Lorsqu'on sélectionne un outil, l'icône du curseur est transformée en icône d'outil. Il est utilisé pour placer et modifier les objets de la face-avant et du diagramme.

La figure suivante montre la palette d'outils avec ces différents éléments :

Figure IV.6 Palette d'outils

- **Palette de commandes :**

La palette de commandes est disponible uniquement sur la face-avant. Elle contient les commandes et les indicateurs qu'on utilise pour créer la face-avant. Elle s'obtient par un simple clic sur le bouton droit de la souris dans l'espace de travail ou dans la barre de menu **fenêtre → Palette de commandes**.

La figure IV.7 montre la palette de commandes avec les éléments (commandes et indicateurs) nécessaires à la création de la face avant.

- **Palette de fonctions :**

La palette de fonctions est disponible uniquement sur le diagramme. Elle contient tous les éléments (fonctions de base,…) nécessaires à la création du code graphique. Elle s'obtient par un simple clic sur le bouton droit dans l'espace de travail du diagramme ou dans le menu **fenêtre → Palette de fonctions.**

La figure IV.8 illustre la palette de fonctions avec ces différents éléments :

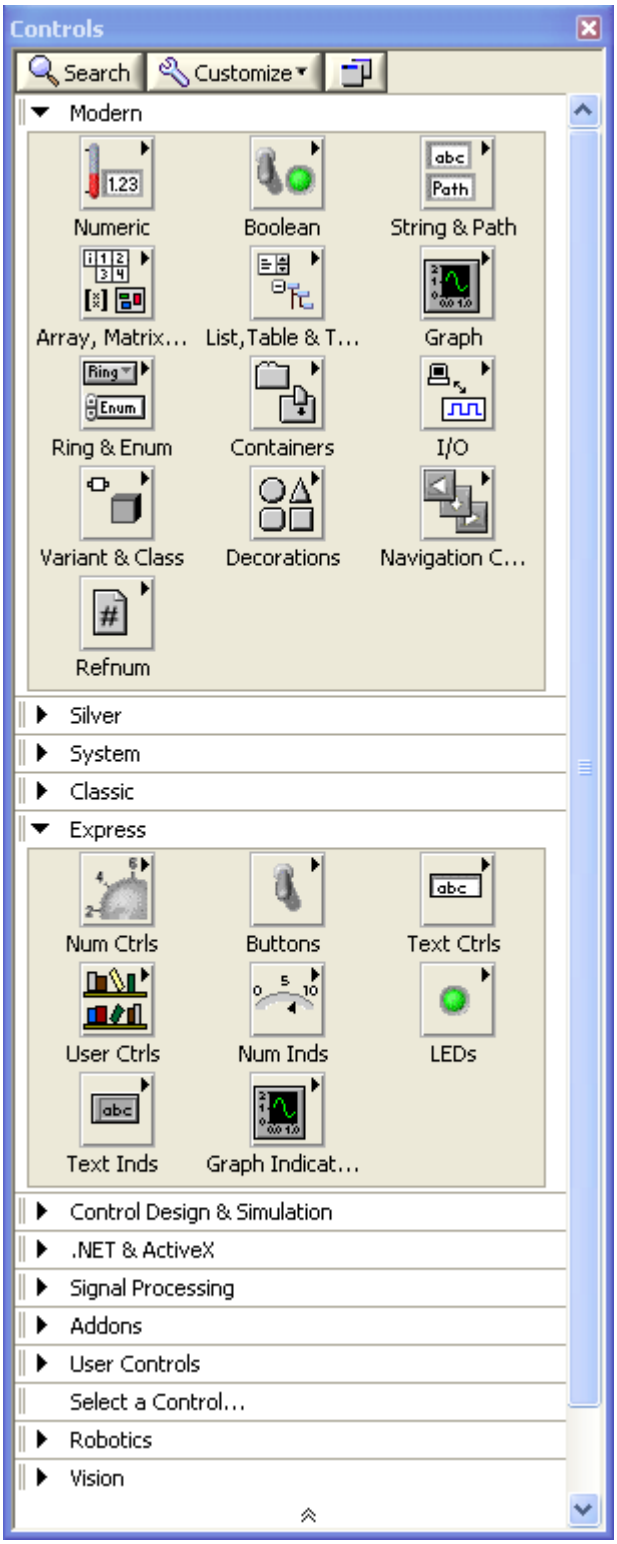


Figure IV.7 Palette de commandes

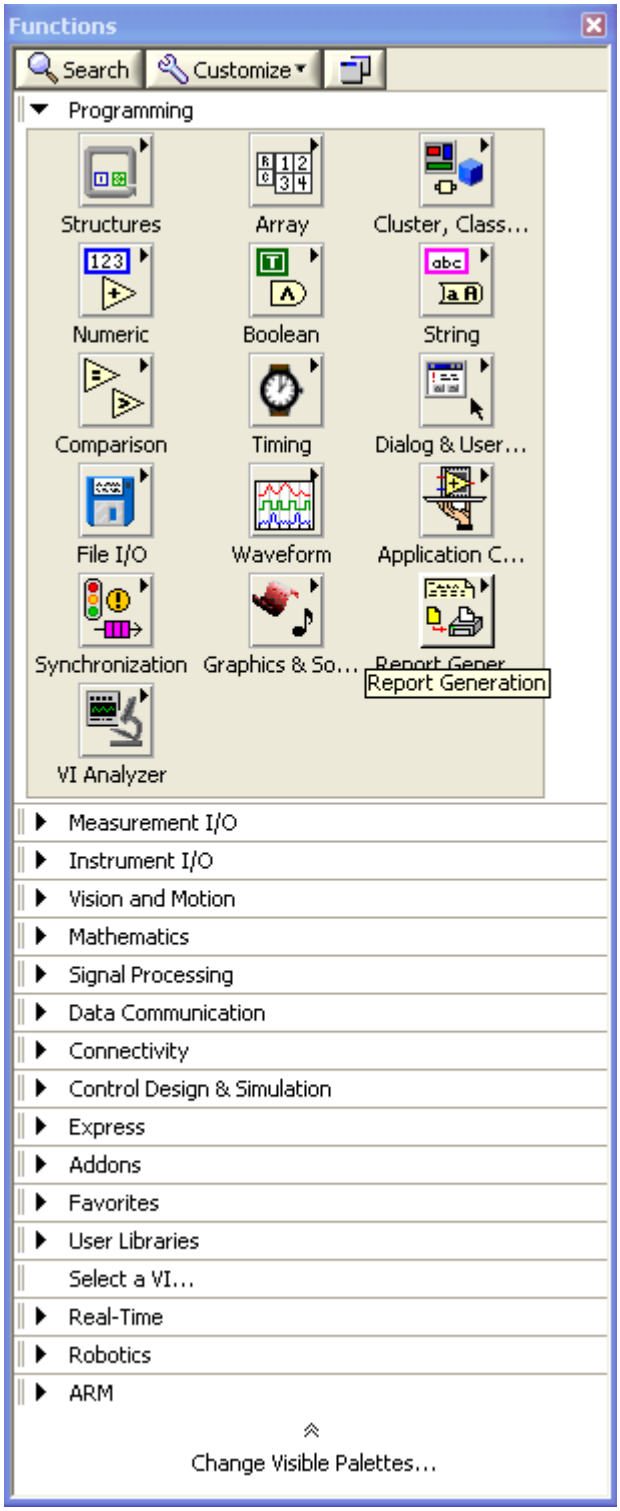


Figure IV.8 Palette de fonctions

IV.2.4 Structure de données sous LabVIEW :

LabVIEW utilise un langage fortement typé et toutes données ou structures de données ne peuvent être manipulées qu'avec des fonctions admettant ce type, en faite dans LabVIEW on trouve plusieurs types (entiers, réels, booléens et chaînes de caractères). Il est important de citer que les icônes représentantes ces données ainsi que les liaisons issues de ces éléments, sont de formes et de couleurs différente.

Booléen.

Entier.

Chaine de caractère.

Réel.

IV.2.5 Structure de programmation :

LabVIEW utilise la programmation graphique qui contient quatre types de structure : la séquence, deux structures d'itération (la boucle « Pour » et la boucle « Tant Que »)et la structure de choix.

- **Structure de séquence :**

La structure de « séquence » permet de spécifier l'ordre d'exécution du flux de donnée. Cette structure se présente sous forme d'un cadre.

- **Les structures itératives :**

La boucle *«Pour»* permet d'exprimer la répétition (ou itération) pour un nombre (N) de fois prédéterminé défini par une connexion d'entrée obligatoire. A l'intérieur de la boucle se trouve un terminale d'entrée local générant l'entier indiquant l'indice d'itération de la boucle

La boucle *« Tant Que »* permet d'exprimer la répétition pour un nombre de fois indeterminer. Un terminale de sortie de type booléen permet d'arrêter cette boucle lorsque la valeur « False » lui est envoyée.

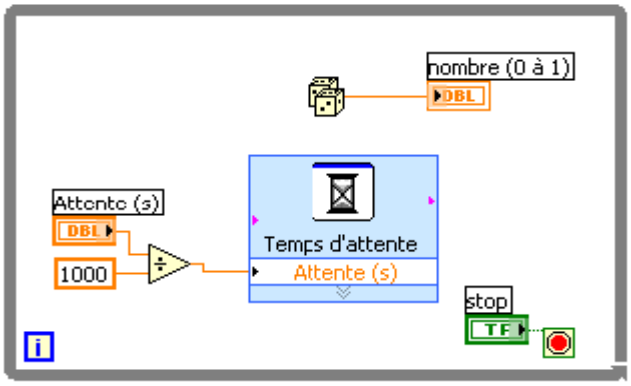


Figure IV.9 Exemple d'utilisation de la boucle « Tant Que »

- **La structure de choix :**

La dernière structure est celle qui permet d'exprimer le choix. La sélection du choix a exécuté est faite par la valeur de la variable associée à la structure, représentée par un point d'interrogation. L'indicateur du choix est représenté en haut de la structure.

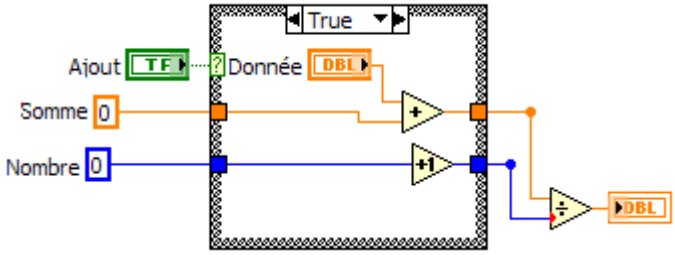


Figure IV.10 Exemple d'utilisation de la structure de choix.

IV.2.6 Bibliothèque de commande :

LabVIEW possède des bibliothèques de fonctions spécialisées dans le domaine du contrôle et de commande.

Dans notre projet nous avons utilisé la bibliothèque *VISA serial port* qui permet de configurer le port RS232. La figure IV.11 illustre les différents paramètres a réglés.

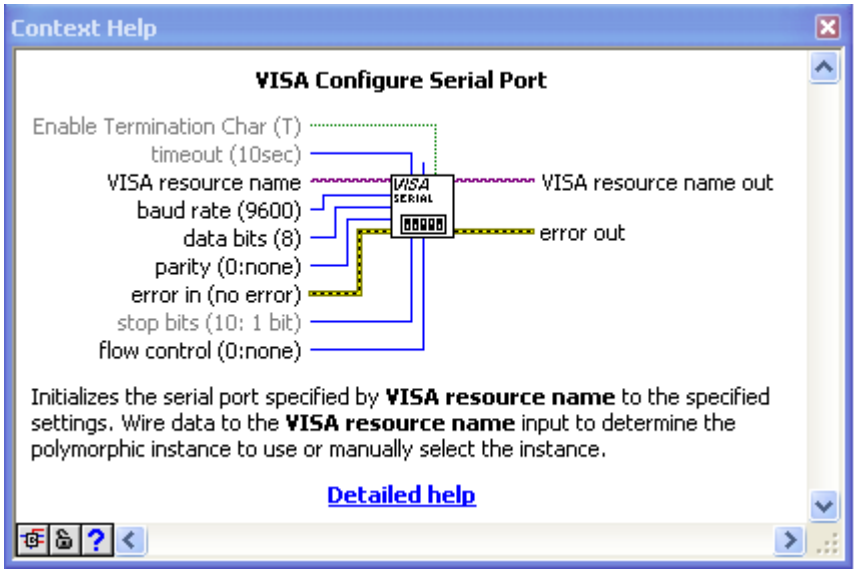


Figure IV.11 Fenétre d'aide VISA serial port

IV.2.7 L'interface de commande réalisé :

L'interface réalisé dans notre projet a pour but l'affichage des differents paramètres mesurés, ainsi qu'un système d'alarme pour avertir l'utilisateur de disfonctionnement du moteur dû au dépassement des seuils prédefinis.

La fenêtre suivante représente le diagramme de l'interface réalisé :

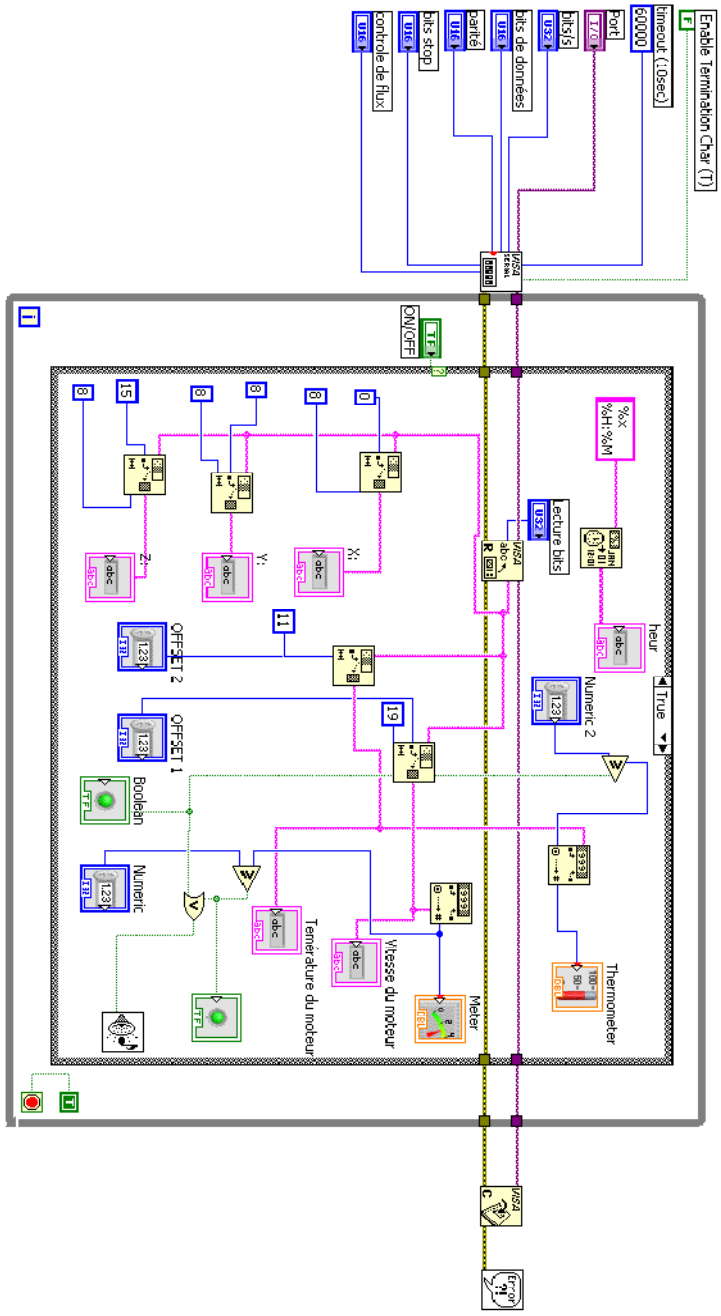


Figure IV.12 Diagramme de l'interface réalisé

Remarque :

Dans le cas d'une erreur de programmation le bouton d'exécution s'affiche comme suite : 

La fenêtre suivante représente la face avant de l'interface réalisé **:**

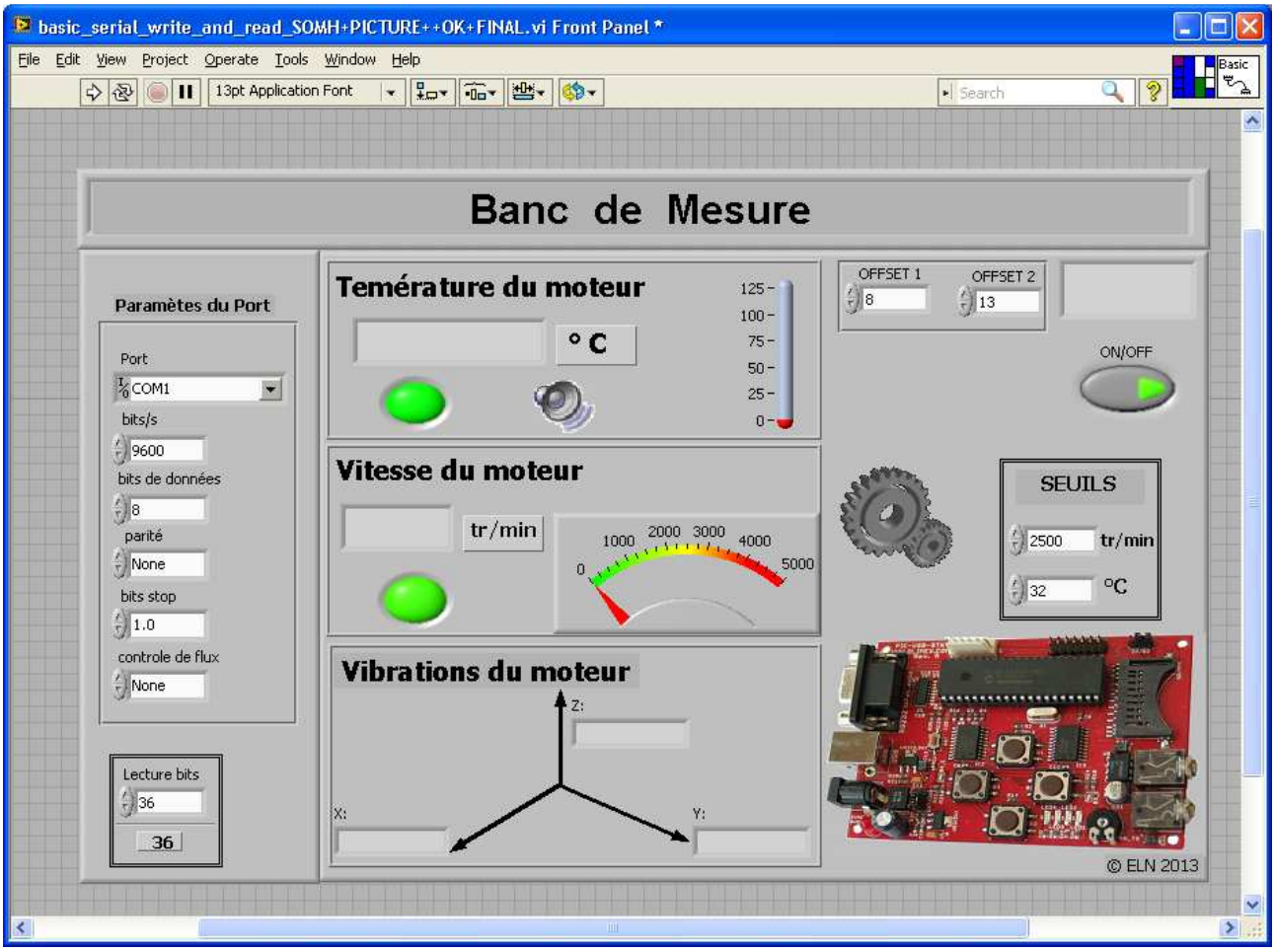


Figure IV.13 Diagramme de l'interface réalisé

A fin de manipuler l'interface réalisé la description des boutons de commande sont représentés ci-dessus :

 **Exécution du programme**

 **Exécution continue**

 **Arrêt de l'exécution**

 **Pause / rependre**

Configuration du format du texte

 **Aligner les objets**

 **Egalisation de l'espacement entre les objets**

 **Plant de l'objet**

 **Redimensionner les objets de la face avant**

IV.3 Réalisation pratique :

IV.3.1.1 Programmation du DS18B20 :

Il est très important de suivre en ordre les étapes mentionnées ci-dessous à chaque fois que le DS18B20 est sollicité, car il ne répondra pas dans le cas contraire. Les étapes sont :

- Initialisation (un Reset au démarrage), en utilisant la commande *Ow_Reset (PORT, pin).*
- Les commandes ROM (ex : ‘0x33’lecture de ROM commande…).
- Fonctions de commande du DS18B20 (ex :’0xCC’ adresser tous les esclaves, ’0xBE’ lecture du bloc note …).

IV.3.1.2 Circuit électrique du branchement :

La figure ci-dessous représente le branchement du DS18B20 au PIC18F4550.

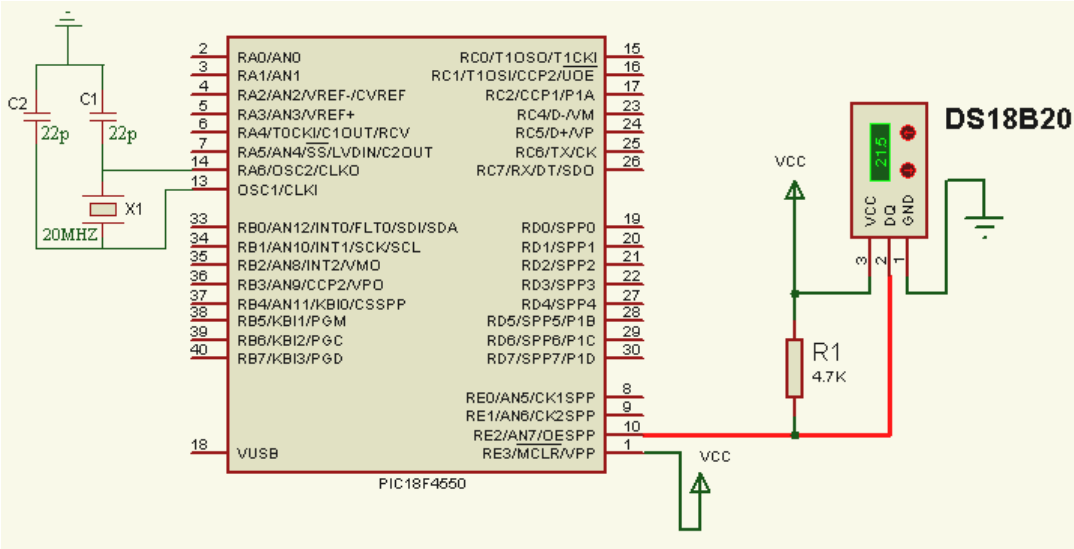


Figure IV.14 Circuit de branchement du DS18B20 au PIC 18F4550

IV.3.1.3 Organigramme du fonctionnement du DS18B20 :

La figure IV.15 montre l'organigramme du fonctionnement du DS18B20.

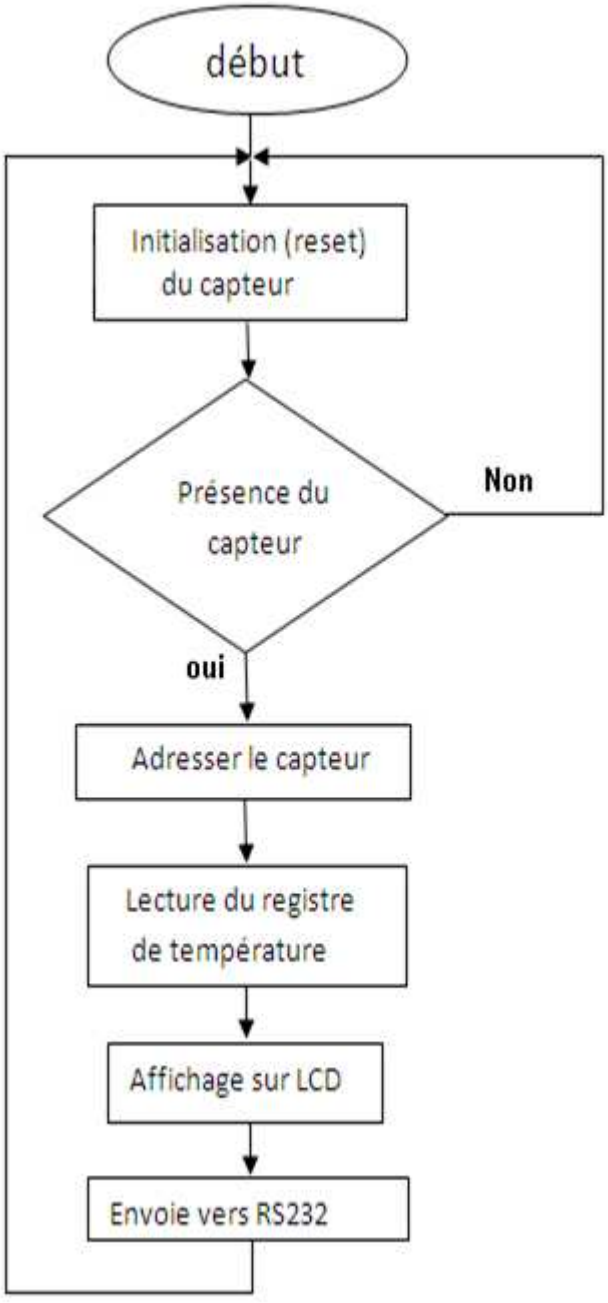


Figure IV.15 Organigramme de fonctionnement du DS18B20

IV.3.2.1 Programmation du BMA 180 :

Pour le bon fonctionnement du capteur d'accélération il est indispensable de suivre les étapes déterminé par le constructeur :

- Initialisation du bus I2C.
- Vérifier la condition de démarrage du BMA180.
- Adressage + lecture des données.
- Affichage des résultats.

IV.3.2.2 Circuit électrique du branchement :

La figure suivante représente le bronchement du BMA180 au PIC 18F4550.

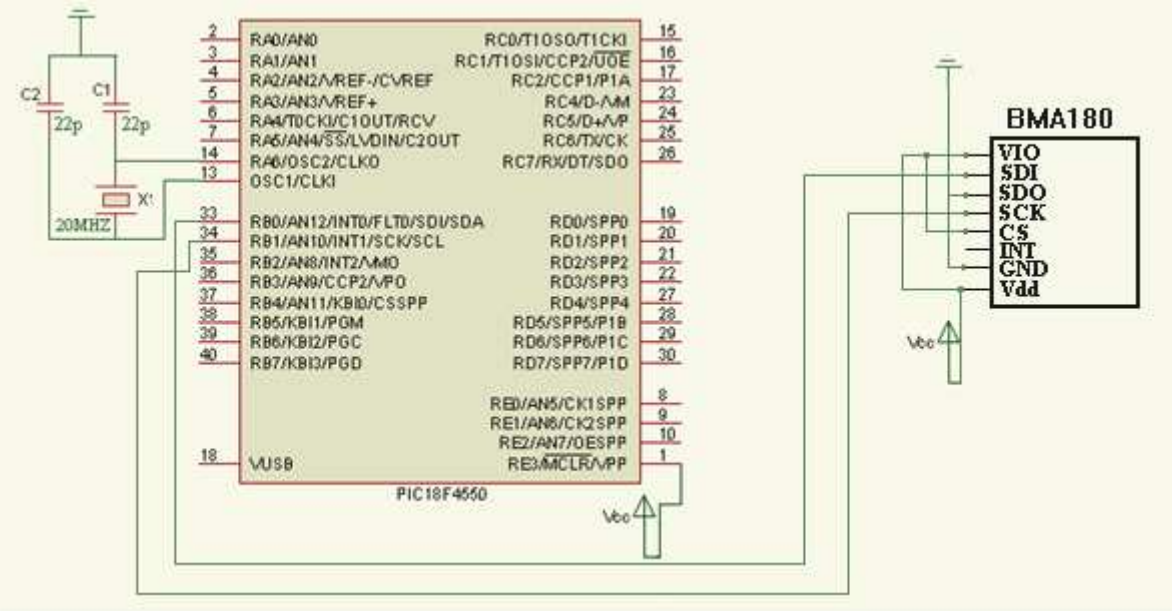


Figure IV.16 Circuit de branchement du BMA180 au PIC 18F4550

IV.3.2.3 Organigramme du BMA180 :

La figure IV.18 montre l'organigramme du fonctionnement du BMA180.

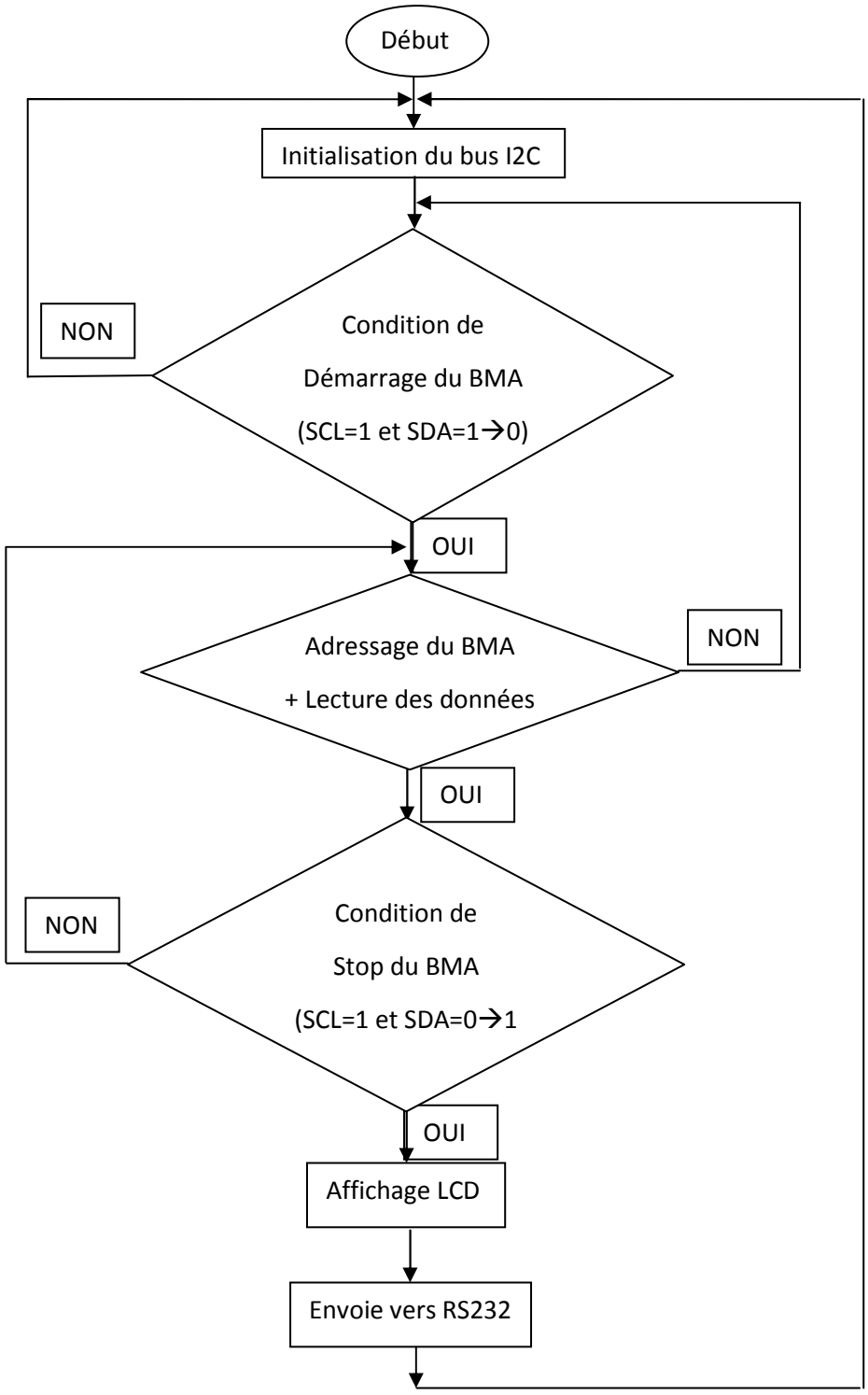


Figure IV.17 Organigramme de fonctionnement du BMA180

IV.3.3.1 Programmation du capteur de vitesse :

Le train d'impulsions reçu par le pic étant de tension 0.17V et 4.6V (compatible TTL), ces impulsions seront interpréter par le pic comme étant des niveaux logique respectivement 0 et 1.

Le Timer0 du PIC18F4550 compte les impulsions délivrées par le dispositif pendant une temporisation de 1s. En laissant le timer0 déborder 60 fois en aura le nombre d'impulsions pour une durée d'une minute, ce qui correspond à la vitesse du moteur.

IV.3.3.2 Calcul de la temporisation du Timer0 :

Le but étant de calculer les impulsions issues de la pin E2 grâce à l'utilisation du Timer0, en mode compteur et avec une temporisation de 1s en utilisant la formule suivante :

Temps = (2^{16} - valeur Décimale chargée TMR0) * 4 * prescaler * 1/Oscillateur en Hz.

Temps = (65536 - 26473)*4*128*1/20000000 = 1 ,0000128 s.

IV.3.3.3 Circuit électrique du branchement :

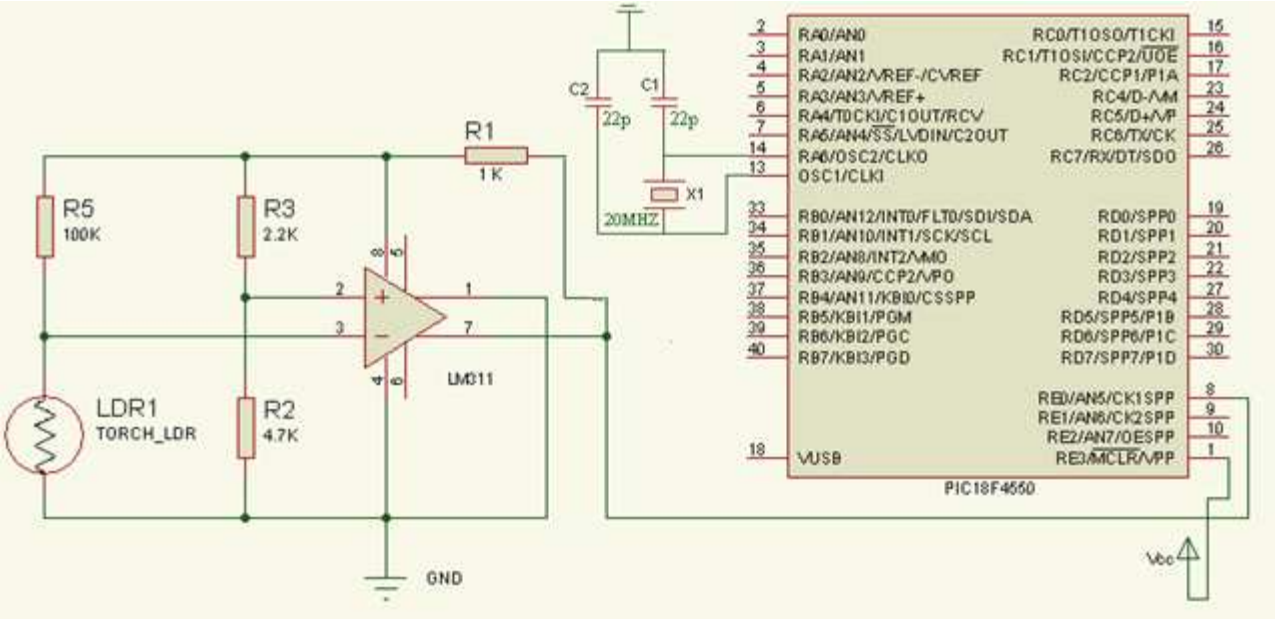


Figure IV.18 Circuit de branchement du capteur de vitesse au PIC 18F4550

IV.3.3.4 Organigramme du capteur de vitesse :

Les figures IV.19 et IV.20 illustrent les organigrammes de fonctionnement du capteur de vitesse.

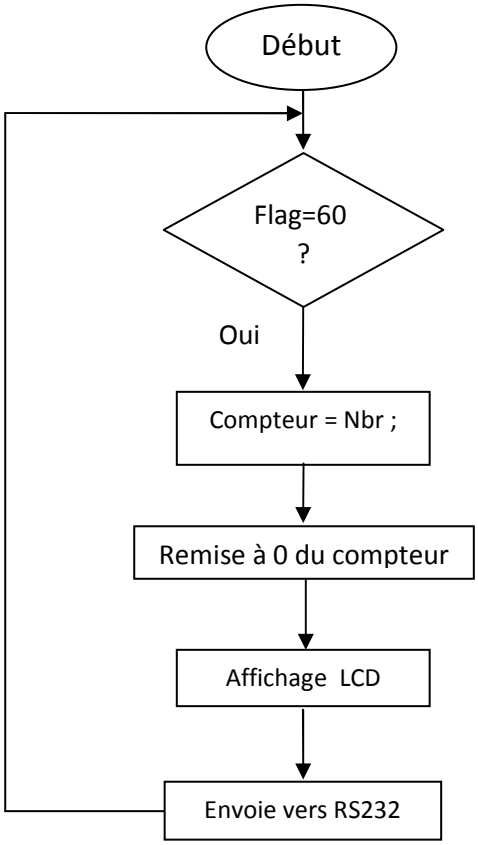


Figure IV.19 Organigramme Timer0

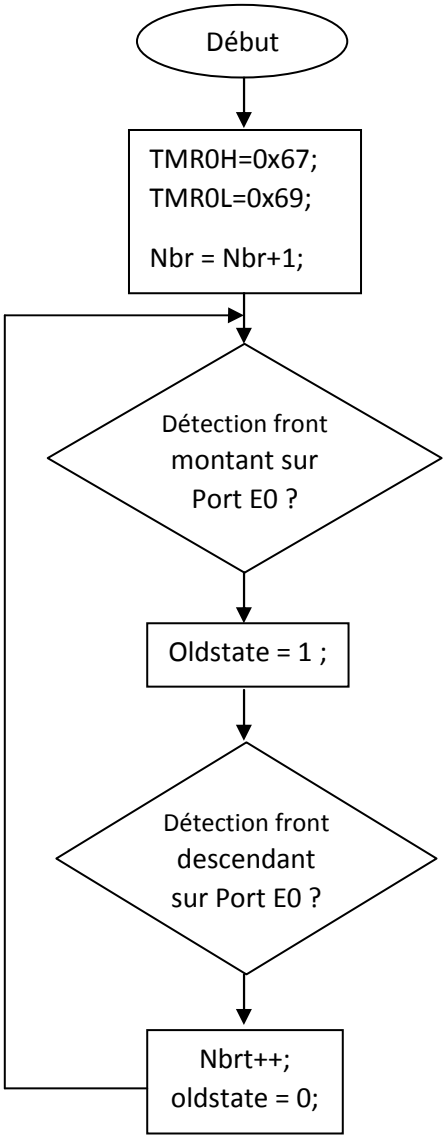


Figure IV.20 Organigramme D'interruption du Timer0

- **Circuit global :**

La figure suivante représente le circuit électrique global du système réalisé dans notre projet.

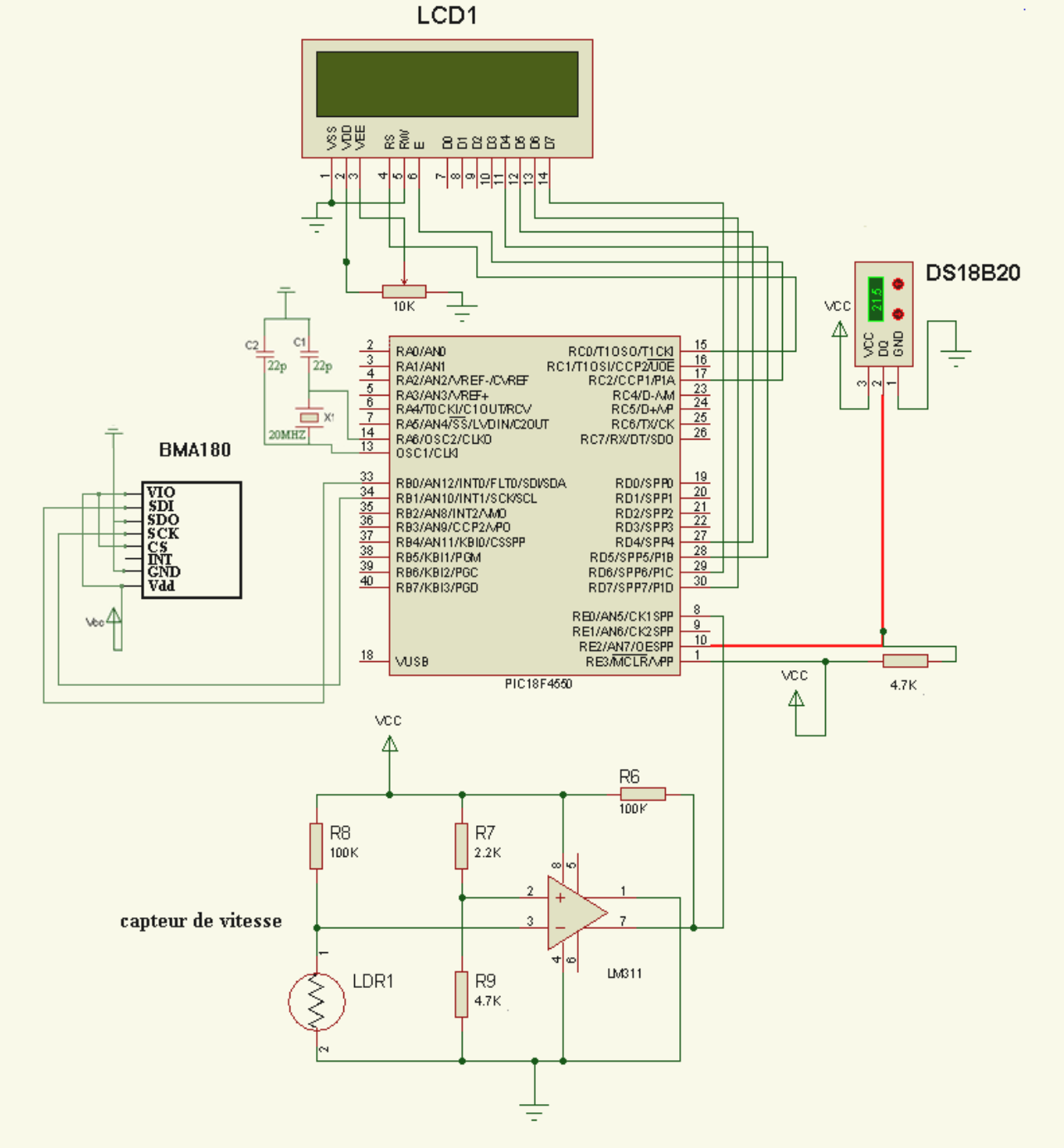


Figure IV.21 Circuit global du système

IV.4 Mise en œuvre et test :

La figure suivante représente la mise en œuvre et le test du système réalisé au niveau du laboratoire.

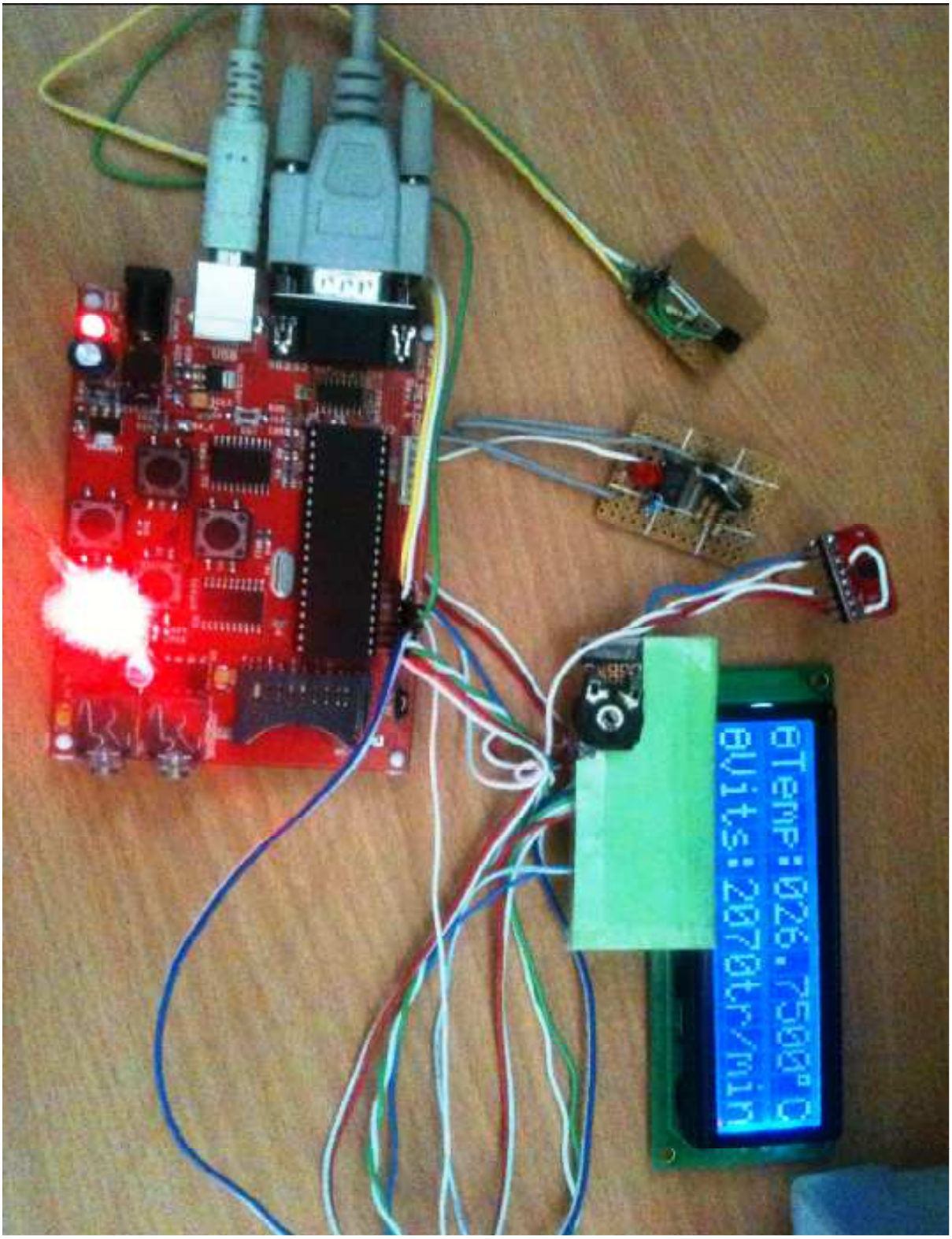


Figure IV.21 Mise en œuvre et test du système

La figure suivante représente les résultats obtenus avec l'interface.

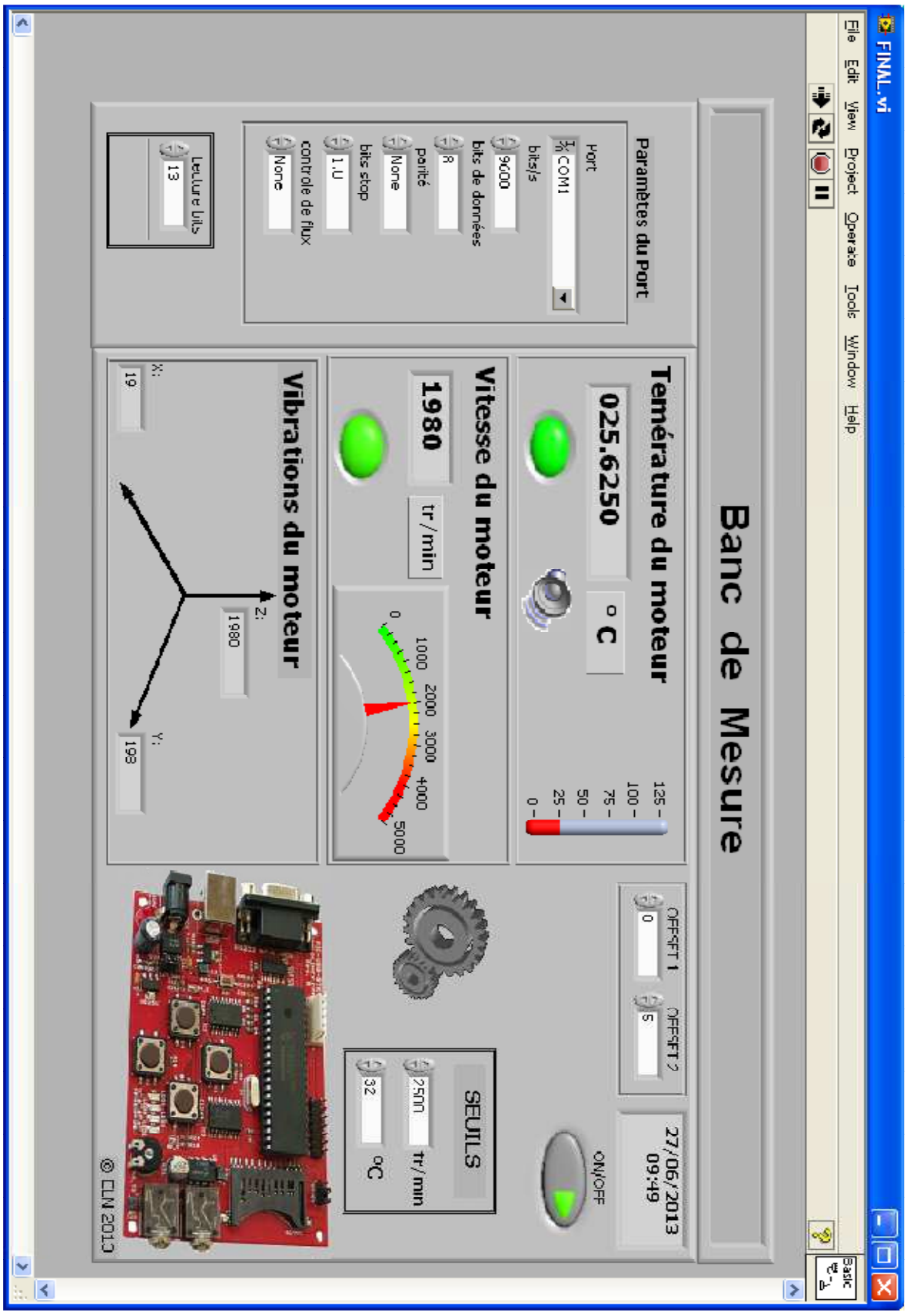


Figure IV.22 Les résultats obtenus avec l'interface

Conclusion

Le travail qui nous a été confié nous a permis d'approfondir nos connaissances dans le domaine d'acquisition numérique de données en utilisant les microcontrôleurs PICs et aussi la programmation dans l'environnement de MikroC ainsi que la programmation graphique sous LabVIEW.

L'utilisation des capteurs numériques dans notre projet n'est guère un choix arbitraire, leurs caractéristiques en matière de consommation, temps de réponse, facilité d'implantation les rendent plus utilisables.

Toute fois, notre travail peut faire l'objet d'améliorations en augmentant le nombre de capteurs utilisés ou bien en ajoutant de nouveaux capteurs pour l'acquisition d'autres grandeurs comme l'humidité ou la pression, ainsi qu'un système de lecture à distance en utilisant le Bluetooth ou RFID.

Pour conclure, nous espérons qu'on a été à la hauteur de la tâche qui nous a été confiée et que notre travail puisse servir de support pour les promotions à venir afin de les inciter à s'intéresser davantage au côté pratique de l'électronique.

Bibliographie

Bibliographie

[1] N. Tandon, G.S. Yadava, K.M. Ramakrishna, A comparison of some condition monitoring techniques for the detection of defect in induction motor ball bearings, Mechanical Systems and Signal Processing 21 (1) (2007) .

[2] J. Faiz, B.M. Ebrahimi, B. Akin, H.A. Toliyat, Dynamic analysis of mixed eccentricity signatures at various operating points and scrutiny of related indices for induction motors, IET Electric Power Applications 4 (1) (2010) .

[3] P.J. Tavner, Review of condition monitoring of rotating electrical machines, IET Electric Power Applications 2 (4) (2008).

[4] A.S. Erbay, Multisensor fusion for induction motor aging analysis and fault diagnosis, PhD thesis, The University of Tennessee, 1999.

[5] JACOB FRADEN ,Handbook of modern sensors. Phisics, designe and application. Third edition.

[6] T.Toumi,N.Guerbas, Mémoire Ingénieur , Conception et réalisation d'un système d'acquisition et de command d'un four de chimie avec une interface LabVIEW,département d'électronique, (2012).

Sites :

http://www.datasheetcatalog.net

http://www.Sparkfun.com