



جامعة بجاية
Tasdawit n Bgayet
Université de Béjaïa

Ministère de l'enseignement supérieur et de la recherche scientifique

Université Abderrahmane Mira de Bejaia

Faculté de Technologie

Département ATE

Mémoire de fin de cycle

En vue de l'obtention du diplôme de Master 2

Option : Instrumentation

Thème

Mise en œuvre d'un capteur température par protocole de communication spi dans l'environnement microcontrôleur PIC16F877

Présenté par :

BOUGUELA SOFIANE

BENMOUHOU B YOUCEF

Encadré par :

Dr. HANFOUG SALAH

Session : 2021/2022

Table des matières

Introduction générale :.....	1
Chapitre I : Description et présentation des outils du projet.	
1 Introduction :.....	3
2 Description matériel :.....	3
2.1 Le microcontrôleur PIC 16F877 :.....	3
2.1.1 Structure du PIC 16F877 :.....	3
2.1.2 Le cœur du PIC : le microprocesseur	4
2.1.3 La mémoire :	4
2.1.3.1 La Mémoire Programme :.....	4
2.1.3.2 La Mémoire RAM :	4
2.1.3.3 La Mémoire EEPROM :.....	4
2.1.3.4 La Pile et le Compteur Programme :	5
2.2 Sondes de température numériques TC72 :.....	5
2.2.1 mode de conversion de température en continu ou un mode de conversion monostable :.....	5
2.2.1.1 Le mode de conversion de température en continu :.....	5
2.2.1.2 Le mode monostable :	5
2.2.2 Caractéristiques - Interface série SPI - Economie d'énergie - Mode de conversion de température en continu et mode monostable (4).....	5
2.2.3 Caractéristiques techniques :.....	6
2.3 Les afficheurs LCD:.....	6
2.3.1 Brochage d'un LCD:.....	7
2.3.2 La mémoire du générateur de caractères (CG RAM):.....	8
2.3.3 Schéma fonctionnel d'un LCD	9
2.3.4 Commande d'un afficheur LCD :	9
2.3.4.1 Mode 8 bits :	9

2.3.4.2	Mode 4 bits :	9
3	Description logiciel :	10
3.1	Logiciel ISIS PROTEUS :	10
3.1.1	Définition :	10
3.1.2	Description du fonctionnement du logiciel ISIS :	11
3.1.3	L'interface utilisateur :	12
3.1.4	Fenêtre d'ensemble (vue d'ensemble) :	12
3.1.5	Fenêtre d'édition :	12
3.1.6	La boîte à outils :	13
3.2	Définition du logiciel ARES :	13
3.2.1	Définition :	14
4	Conclusion :	15

Chapitre II : Etude du protocole de communication SPI.

1	Introduction :	17
1.1	Origine	17
1.2	Présentation	17
1.3	Philosophie du bus SPI.....	18
2	Schéma de principe	18
2.1	Format de transfert de données.....	19
2.2	Synoptique d'une liaison SPI Maître-Multi-Esclaves.....	19
3	Bus SPI :	20
3.1	Définition :	20
3.2	Fonctionnement :	21
3.3	Configuration du bus SPI :	24
3.3.1	Configuration en bus :	25
3.4	Protocole de gestion d'un réseau SPI :	26
3.4.1	La Scrutation :	26

3.4.2	L'interrogation :	26
3.5	Flux de données :	26
3.6	Le 16F877 et le mode SPI :	26
3.6.1	Le mode slave :	27
3.7	Comparaison entre le I2C et le SPI :	27
4	Conclusion :	28

Chapitre III : Conception et réalisation pratique.

1.	Introduction :	30
2.	Description du système :	30
2.1	Schéma synoptique :	30
3.	Simulation du circuit :	31
3.1	Schéma électronique :	31
3.2	Programmation du pic 16F877 :	32
3.3	Organigrammes :	37
3.4	Routage et création du circuit imprimé :	41
4.	Réalisation pratique :	42
4.1	4.1 Listes des composants :	42
4.2	Gravure du programme sur le pic :	45
4.3	Réalisation du circuit imprimé et implantation des composants :	46
5	Conclusion :	47
	Conclusion générale	48
	Bibliographie	49
	Annexe :	50
	Résumé :	52
	Abstract:	52

Liste des figures :

Chapitre I : Description et présentation des outils du projet.

FIGURE I 1: LE SCHEMA CI-DESSUS REPRESENTE LES DIFFERENTS MODULES DU PIC 16F877	3
FIGURE I 2: CARACTERISTIQUE TECHNIQUE TC 72	6
FIGURE I 3: AFFICHEUR LCD	7
FIGURE I 4: SCHEMA FONCTIONNEL DE L'LCD.....	9
FIGURE I 5: LOGICIEL PROTEUS.	12
FIGURE I 6: PRESENTATION DE L'ECRAN ISIS.	14
FIGURE I 7: PRESENTATION DE L'ECRAN ARES.	15
FIGURE I 8: LOGICIEL MIKROC PRO FOR PIC.	16

Chapitre II : Etude du protocole de communication SPI.

FIGURE II 1: ILLUSTRE LES CONNEXIONS SPI.....	17
FIGURE II 2: PHILOSOPHIE DU BUS SPI	18
FIGURE II 3: SCHEMA DE PRINCIPE SIMPLIFIE D'UNE LIAISON SPI.....	18
FIGURE II 4: FORMAT DE TRANSFERT DE DONNEES	19
FIGURE II 5: MODULE MSSP EN MODE SPI INTEGRE AU PIC 16F877.	23
FIGURE II 8: CONFIGURATION EN MODE BUS D'UNE COMMUNICATION SERIE SPI.	25

Chapitre III : Conception et réalisation pratique.

FIGURE III 1: SCHEMA SYNOPTIQUE DU CIRCUIT.	30
FIGURE III 2: SCHEMA ELECTRONIQUE GLOBALE DU CIRCUIT.....	32
FIGURE III 3: CREATION D'UN NOUVEAU PROJET.....	33
FIGURE III 4: CHOIX DES PARAMETRES DU PROJET.	34
FIGURE III 5: FENETRE D'EDITION DU PROGRAMME.....	36
FIGURE III 6: CHARGEMENT DU PROGRAMME DANS LE PIC SUR ISIS.....	37
FIGURE III 7: SCHEMA DE ROUTAGE.	41
FIGURE III 8 : GRAVURE DU PROGRAMME SUR LE PIC.....	45
FIGURE III 9: LE TYPON.	46

Liste des tableaux :

Chapitre II : Etude du protocole de communication SPI.

TABLEAU 1.II : : COMPARAISON ENTRE LE PROTOCOLE I2C ET SPI. 27

Liste des abréviations

PIC : Programmable Interface Controller.
CMOS: Complementary Metal Oxide Semiconductor.
RISC : Reduced Instructions Set Computer.
DIN : Data IN.
DOUT : Data OUT.
GND : Ground.
CS : Chip Select.
CLK : Clock.
BCD : Binary Coded Decimal MOSI
PCB : Printed Circuit Board.
IDE : Integrated Développement Environnement.
CPU : Central Processor Unit.
RAM : Random Access Memory ou mémoire à accès direct.ROM : Read Only Memory.
MIPS : Million d'Instructions Par Seconde.
EPROM: Erasable Programmable Read Only Memory.
EEPROM: Electrical Erasable Programmable Read Only Memory.
VSS : Neutre.
VPP : Tension de programmation.
INT: Interruptions.
LCD : Cristaux Liquide Display.
I2C: (IIC) Inter Integrated Circuit.
SDA : Signal de donnée
SCL : Signal d'horloge
ACK : Acknowledge (reconnaître).
VDD : Voltage Drain Drain.
VSS : Voltage Source Source.
MCLR : Master Clear.
TTL : Transistor Transistor Logique. CCP.
SSPSR : MSSP Shift Register.
DAC: Digital to Analog Converter.ADC: Analog to Digital Converter

Introduction générale

Introduction générale :

Actuellement, l'électronique est au centre de toutes les réalisations et au cœur de plusieurs développements et perfectionnements. Les dispositifs électroniques sont aussi bien retrouvés dans le domaine médical, industriel mais aussi au niveau des systèmes de contrôle et de régulation automatique.

La protection thermique du nouveau-né comporte une série de mesures prises à la naissance et pendant les premiers jours de la vie afin d'assurer que la température corporelle de l'enfant ne s'abaisse pas trop ou ne s'élève pas trop, mais se maintient dans la normale entre 36.5 et 37.5 °C. la couveuse sert donc à maintenir cette température.

Deux objectifs sont assignés à ce travail. Le premier objectif c'est d'affûter nos connaissances théoriques et pratiques de par les circuits qui vont être manipulés au cours de ce travail. Le deuxième objectif, c'est de la familiarisation avec les constituants d'une couveuse ainsi que son principe de fonctionnement.

Notre travail est basé sur l'exploitation d'un capteur de température TC 72, un microcontrôleur, un bloc d'alimentation, un bloc de chauffage et un système de ventilation .

Le présent rapport est subdivisé en trois parties principales :

Dans un premier chapitre, nous étudierons les outils matériels et logiciel nécessaires à notre réalisation.

Le deuxième chapitre sera consacré à la description du microcontrôleur plus spécialement le protocole de communication SPI .

Et le troisième chapitre se portera sur les différentes étapes de la réalisation pratique. Finalement le travail sera fini par une conclusion générale.

Chapitre I

Description et présentation des outils du projet

1 Introduction :

Pour la mise en œuvre de cette réalisation électronique nous avons choisis les éléments les plus performant afin d'arriver à des résultats satisfaisants.

Dans ce chapitre nous avons projeté la lumière sur tous les outils nécessaires à notre réalisation où nous avons présenté de manière générale les différents composants ainsi que les logiciels de conception et de simulation avec lesquels nous avons travaillé.

2 Description matérielle :

2.1 Le microcontrôleur PIC 16F877 :

2.1.1 Structure du PIC 16F877 :

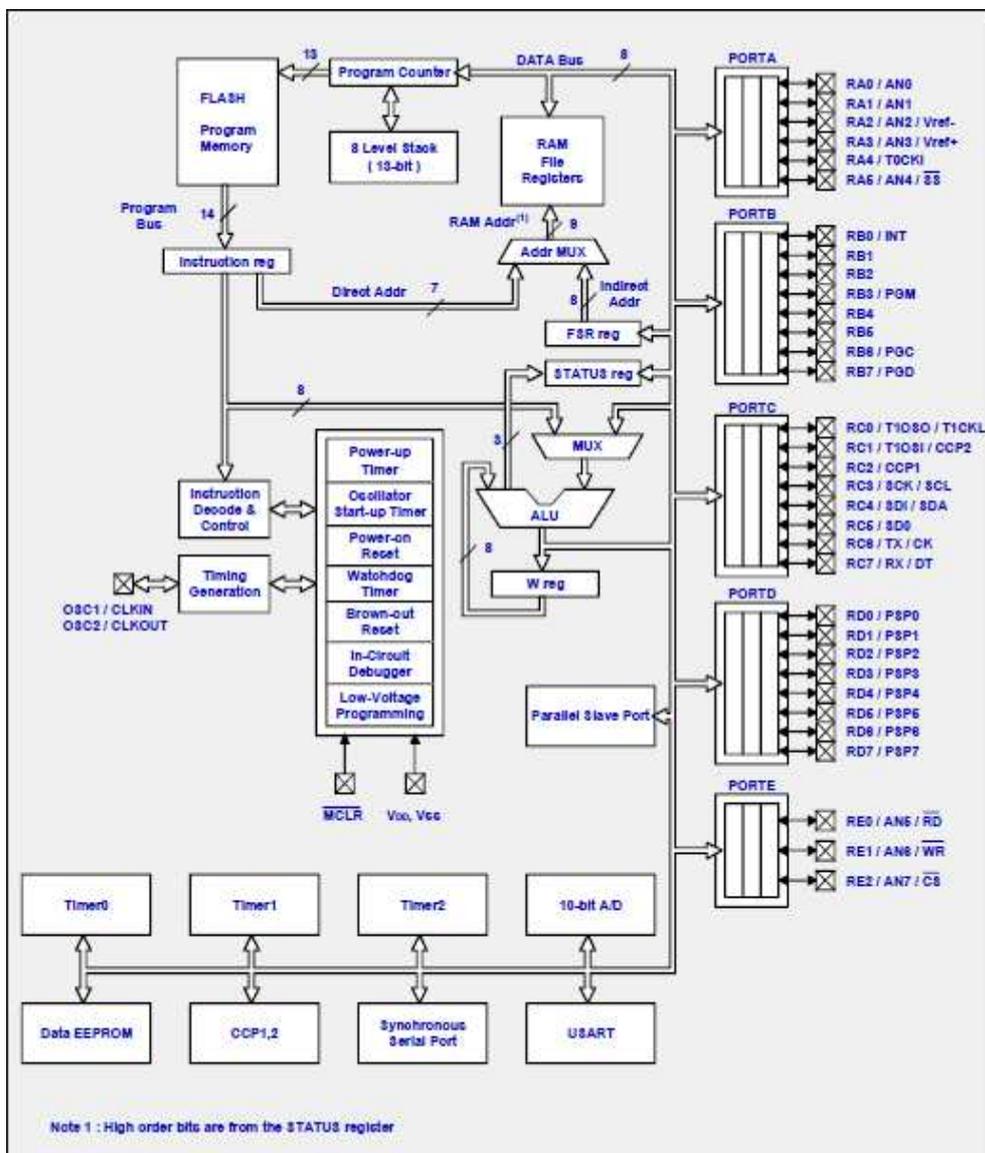


Figure I 1: Le schéma ci-dessus représente les différents modules du PIC 16F877

Chapitre I : Description et présentation des outils du projet.

2.1.2 Le cœur du PIC : le microprocesseur

Un microcontrôleur, c'est avant tout un microprocesseur, une unité de traitement logique qui effectue l'une après l'autre les opérations contenues dans un microprogramme stocké en mémoire (la mémoire FLASH). On peut le voir sur le schéma (en jaune ci-dessous), il est essentiellement composé 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 « statuts 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 les différents bus qui relient tous ces éléments entre eux. (1)

2.1.3 La mémoire :

Sur le PIC, il n'y a pas une mais trois mémoires :

2.1.3.1 La Mémoire Programme :

De type FLASH sur le 16F877. Capacité : 8K. C'est dans celle-ci qu'est stocké le programme du PIC. Après compilation de votre code, le compilateur génère un fichier « .hex », 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 (DATA Bus), 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 programmée qu'une seule fois.

2.1.3.2 La Mémoire RAM :

Qui 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.

L'espace mémoire RAM adressable est de **512** positions de 1 octet chacune :

- 96 positions sont réservées au SFR (Spécial Fonction Registrés) qui sont les registres de configuration du PIC.
- Les 416 positions restantes constituent les registres GPR (General Purpose Registrés) ou RAM utilisateur. Sur le 16F876 et 16F877, 3 blocs de 16 octets chacun ne sont pas implantés physiquement d'où une capacité de RAM utilisateur de 368 GPR.

2.1.3.3 La Mémoire EEPROM :

L'EEPROM est plutôt une mémoire de stockage de données à long terme, alors que la RAM

Chapitre I : Description et présentation des outils du projet.

est utilisée pour les variables du programme. Sur le PIC 16F877, on a 256 octets d'EEPROM disponible. Les mémoires de type EEPROM sont limitées en nombre de cycles 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 croit. (2)

2.1.3.4 La Pile et le Compteur Programme :

Le « Program Counter » ou PC est le compteur qui pointe dans la mémoire programme la prochaine instruction à exécuter. Il est lié à la pile système (Pile =Stack en anglais) qui est sur le PIC 16F877 une pile 8 niveau. (3)

2.2 Sondes de température numériques TC72 :

La série de sondes de température numériques TC72 a été développée pour la gestion de la température dans les PC, téléphones mobiles, disques durs, systèmes de divertissement et Equipements bureautiques, entre autres.

Vous pouvez utiliser les sondes en 2 modes :

2.2.1 Mode de conversion de température en continu ou un mode de conversion monostable :

2.2.1.1 Le mode de conversion de température en continu :

mesure la température toutes les 150 millisecondes, puis stocke les données dans les registres de température.

2.2.1.2 Le mode monostable :

effectue 1 mesure de température avant de revenir en mode Sleep.

2.2.2 Caractéristiques

- Interface série SPI
- Economie d'énergie
- Mode de conversion de température en continu et mode monostable (4)

Chapitre I : Description et présentation des outils du projet.

2.2.3 Caractéristiques techniques :

Attribut	Valeur
Fonction du capteur	Température
Type de sortie	Numérique
Type d'interface	SPI
Précision	±5°C
Type de montage	CMS
Nombre de broches	8
Plage de température de fonctionnement	-55 à +125 °C.
Température de fonctionnement minimum	-55 °C
Température d'utilisation maximum	+125 °C
Tension d'alimentation de fonctionnement minimum	2,7 V
Tension d'alimentation fonctionnement maximum	5,5 V
Type de boîtier	MSOP
Longueur	3mm
Hauteur	0.95mm
Largeur	3mm
Dimensions	3 x 3 x 0.95mm
Résolution	10 bits

Figure I 2: caractéristique technique TC 72

2.3 Les afficheurs LCD :

Les afficheurs à cristaux liquides, autrement appelés afficheurs LCD (Liquide Crystal Display), sont des modules compacts, intelligents qui nécessitent peu de composants externes pour un bon fonctionnement. Ils consomment relativement peu (de 1 à 5 mA), sont relativement bon marché et s'utilisent avec beaucoup de facilité.

Plusieurs afficheurs sont disponibles sur le marché et diffèrent les uns des autres, non

Chapitre I : Description et présentation des outils du projet.

seulement par leurs dimensions, (de 1 à 4 lignes de 6 à 80 caractères), mais aussi par leurs caractéristiques techniques et leur tension de service. Certains sont dotés d'un rétro éclairage de l'affichage. Cette fonction fait appel à des LED montées derrière l'écran du module. Un exemple de module LCD est donné à la Figure.

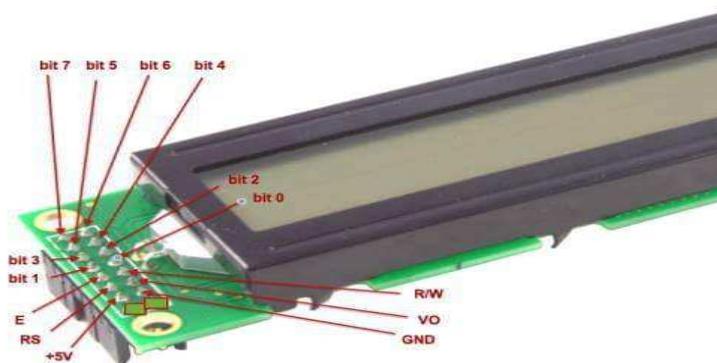


Figure I 3: afficheur LCD

2.3.1 Brochage d'un LCD :

Un circuit intégré spécialisé est chargé de la gestion du module. Il remplit une double fonction : d'une part il commande l'affichage et de l'autre il se charge de la communication avec l'extérieur.

- Deux broches pour l'alimentation du module
- Une broche pour le réglage du contraste de l'afficheur
- Une broche (RS) : TTL pour la Sélection du registre (Registre Select). Grâce à cette broche, l'afficheur est capable de faire la différence entre une commande et une donnée : Un niveau bas indique une commande et un niveau haut indique une donnée.
- Une broche (R/W) : TTL qui permet la sélection du mode lecture ou écriture
- Une broche E : TTL pour la commande des opérations d'écriture ou de lecture.
- Un ensemble de huit Broches utilisées pour le transfert des données ou des instructions. Le transfert peut se faire sur 8 bits, toutes les broches sont alors utilisées, ou sur 4 bits, dans ce cas, seules quatre broches sont utilisées.
- Deux broches supplémentaires sont disponibles sur les afficheurs LCD avec retro éclairage

L'afficheur possède deux types de mémoire, la DD RAM et la CG RAM, respectivement mémoire d'affichage et mémoire du générateur de caractères.

La DD RAM est la mémoire où sont stockés les caractères actuellement affichés à l'écran. Pour un afficheur de 2 lignes de 16 caractères, les adresses sont définies de la façon suivante :

L'adresse 00H correspond à la ligne du haut à gauche, 0FH à droite. L'adresse 40H correspond à la ligne du bas à gauche, 4FH à droite. La zone invisible correspond à la mémoire de l'afficheur (48 caractères). Lorsqu'un caractère est inscrit à l'adresse 27H, le caractère suivant apparaît à la ligne suivante

Chapitre I : Description et présentation des outils du projet.

2.3.2 La mémoire du générateur de caractères (CG RAM) :

Le générateur de caractère permet la création d'un maximum de 8 caractères ou symboles 5x7. Une fois les nouveaux caractères chargés en mémoire, il est possible d'y accéder comme s'il s'agissait de caractères classiques stockés en ROM.

La CG RAM utilise des mots de 8 bits, mais seuls les 5 bits de poids faible apparaissent sur le LCD. Ainsi D4 représente le point le plus à gauche et D0 le point le plus à droite. Par exemple, charger un octet de la CG RAM à 1Fh fait apparaître tous les points de cette rangée ; charger un octet à 00h éteint tous ces points. Les 8 lignes d'un caractère doivent être chargées dans la CG RAM.

La CG RAM peut être utilisée pour créer des caractères en vidéo inversée, des caractères avec des accents, etc. La limitation d'un total de 8 caractères peut être contournée en utilisant une bibliothèque de 8 symboles résidant dans le système hôte. Un maximum de 8 caractères peut être affichés à la fois. La CG RAM peut être rechargée périodiquement en fonction des besoins. Si un caractère actuellement sur l'afficheur est changé, alors le changement est immédiatement apparent sur l'afficheur.

2.3.3 Schéma fonctionnel d'un LCD

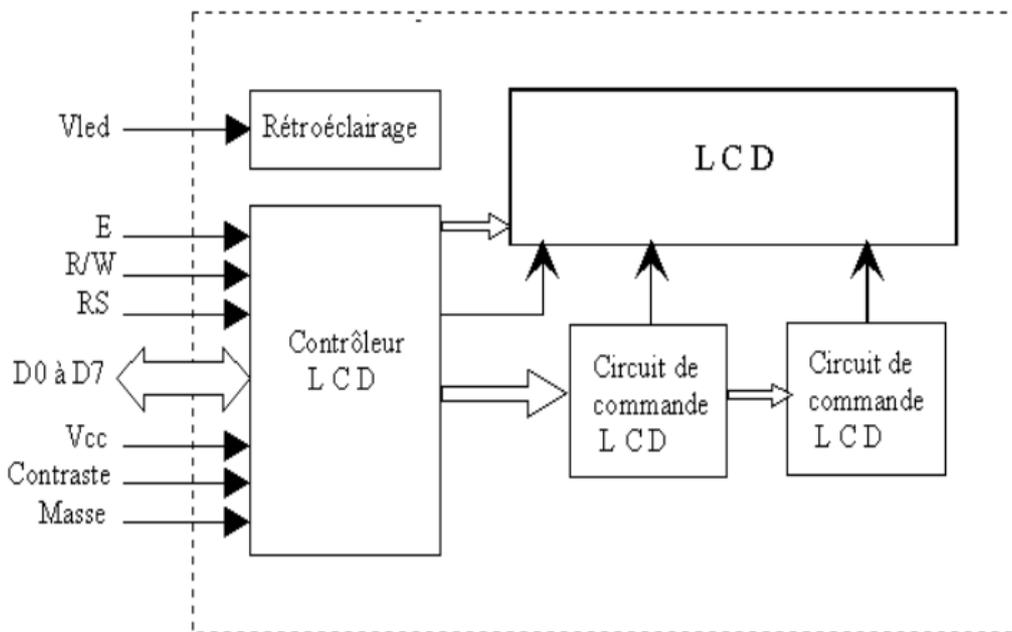


Figure I 4:Schéma fonctionnel de l'LCD.

Chapitre I : Description et présentation des outils du projet.

2.3.4 Commande d'un afficheur LCD :

Deux modes de fonctionnement de l'afficheur sont disponibles, le mode 4 bits et le mode 8 bits, modes que l'on choisira à l'initialisation de l'afficheur.

2.3.4.1 Mode 8 bits :

Dans ce mode 8 bits, les données sont envoyées à l'afficheur sur les broches D0 à D7. On place la ligne RS à 0 ou à 1 selon que l'on désire transmettre une commande ou une donnée. Il faut aussi placer la ligne R/W à 0 pour indiquer à l'afficheur que l'on désire effectuer une écriture. Il reste à envoyer une impulsion d'au moins 450 ns sur l'entrée E, pour indiquer que des données valides sont présentes sur les broches D0 à D7. L'afficheur lira la donnée sur le front descendant de cette entrée. Si on désire au contraire effectuer une lecture, la procédure est identique, mais on place cette fois la ligne R/W à 1 pour demander une lecture. Les données seront valides sur les lignes D0 à D7 lors de l'état haut de la ligne E.

2.3.4.2 Mode 4 bits :

Il peut, dans certains cas, être nécessaire de diminuer le nombre de fils utilisés pour commander l'afficheur, comme, par exemple lorsqu'on dispose de très peu de broches d'entrées sorties disponibles sur un microcontrôleur. Dans ce cas, on peut utiliser le mode quatre bits de l'afficheur. Les données sont alors écrites ou lues en envoyant séquentiellement les quatre bits de poids fort suivi des quatre bits de poids faible. Une impulsion positive d'au moins 450 ns doit être envoyée sur la ligne E pour valider chaque demi-octet ou nibble. Dans les deux modes, on peut, après chaque action sur l'afficheur, vérifier que celui-ci est en mesure de traiter l'information suivante. Pour cela, il faut demander une lecture en mode commande, et tester le flag Busy BF.

Lorsque BF=0, l'afficheur est prêt à recevoir une nouvelle commande ou donnée. Il se peut qu'on dispose encore de moins de broches disponibles dans l'application envisagée. Dans ce cas, on peut alors relier la ligne R/W à la masse de façon à forcer l'afficheur en écriture. On a alors besoin, hors alimentation de seulement six fils en mode 4 bits, et dix fils en mode 8 bits, pour commander l'afficheur, mais on ne peut alors plus relire l'afficheur. Ceci n'est pas gênant dans la mesure où on sait ce qu'on a écrit sur l'afficheur, mais on ne peut alors plus relire le flag Busy. Il faut alors utiliser des temporisations après chaque écriture sur l'afficheur. On perd alors un peu en temps d'affichage, mais on gagne une broche d'entrée/sortie.

3 Description logiciel :

3.1 Logiciel ISIS PROTEUS :

3.1.1 Définition :

ISIS « Intelligent Schematic Input System » est au cœur de Proteus, et il est bien plus que juste un autre paquet schémas. Il combine un puissant environnement de conception avec la possibilité de définir la plupart des aspects du dessin apparence.

Nous avons besoin d'une entrée rapide de la complexité et de la simulation pour la conception des circuits imprimés, ou la création de schémas attrayants pour publication, ISIS est l'outil pour l'emploi. Grâce à des modules additionnels dite bibliothèques de composants, ISIS est capable de simuler le comportement d'un microcontrôleur « PIC, Atmel, 8051, ARM, HC11... » et son interaction avec les composants qui l'entourent.

Également il permet aussi de simuler des montages. Les résultats de la simulation sont très proches de la réalité. Il permet la simulation d'éléments analogique, numérique, logique, de microcontrôleurs, d'afficheur LCD...

De plus, il dispose d'éléments de mesure virtuels : un oscilloscope, un voltmètre, un analyseur logique... C'est un logiciel très facile à manipuler et à comprendre, c'est pour cela que nous l'avons choisi, d'ailleurs, nous avons réservé quelques pages pour décrire son fonctionnement.

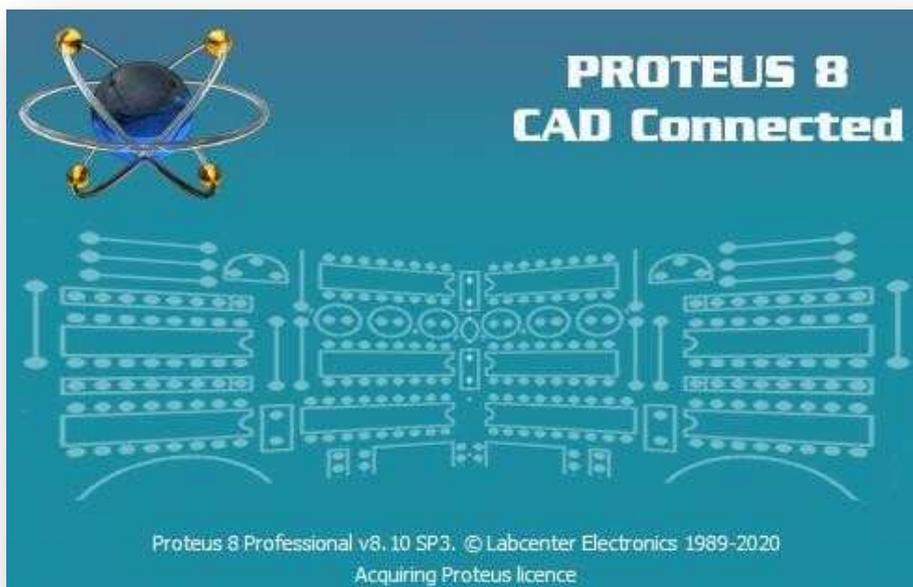


Figure I 5: Logiciel PROTEUS.

Chapitre I : Description et présentation des outils du projet.

3.1.2 Description du fonctionnement du logiciel ISIS :

Après l'installation de notre logiciel, on lance l'éditeur ISIS. Au sommet de l'écran se trouve la barre de menu dans laquelle il y'a certaines fonctions et icônes qu'on va choisir parmi eux qu'il nous faut.

Ce logiciel nous permet de dessiner des schémas de haute qualité, notamment en donnant la possibilité de contrôler parfaitement l'apparence du dessin : largeurs de lignes, styles de remplissage, couleurs et polices, etc.....

Nous pouvons ainsi produire des schémas attrayants tels que ceux publiés dans les magazines. Une fois notre schéma est terminé. Nous pourrons l'exporter dans un fichier graphique ou le copier dans le presse-papiers pour l'incorporer dans d'autres documents.

L'interface utilisateur de l'éditeur de schémas ISIS est facile et très pratique. Les actions courantes sont simples et rapides à exécuter. Il nous permet de placer un lien à tout instant par un simple clic sur une broche de composant ou sur un lien déjà placé. De plus, les opérations déplacement, édition, déplacement et suppression se font directement à l'aide de la souris, sans devoir passer par des menus ou des icônes.

Ce simulateur contient des aides au concepteur, comme des indicateurs visuels qui indiquent l'action réalisée sur clic gauche et une mise en surbrillance des objets qui indique quel est l'objet ou le groupe d'objets concernés par l'action

Chapitre I : Description et présentation des outils du projet.

3.1.3 L'interface utilisateur :

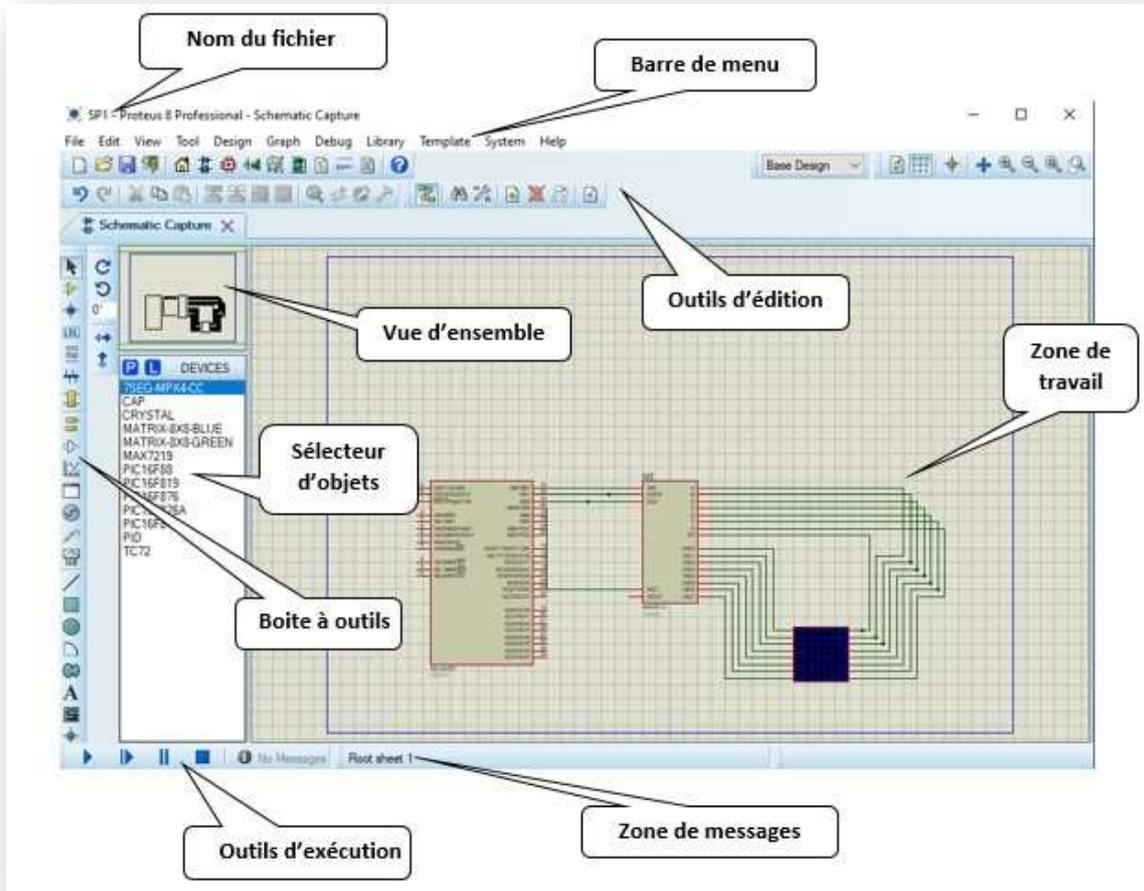


Figure I 6: Présentation de l'écran ISIS.

L'écran ISIS :

3.1.4 Fenêtre d'ensemble (vue d'ensemble) :

Le cadre en bleu délimite l'espace de travail tel qu'il a été défini par la commande « définir taille des feuilles » du menu « système ».

Le cadre en vert délimite la zone de travail, c'est à dire la partie du schéma visible dans la fenêtre principale. Vous pouvez déplacer cette zone de travail en pointant la souris sur la zone désirée de la fenêtre d'ensemble et en effectuant un clic gauche.

3.1.5 Fenêtre d'édition :

C'est dans cette fenêtre que vous éditez votre circuit. C'est là que vous placez et câblez les composants.

Vous pouvez redéfinir la zone de travail à l'aide de la fenêtre d'ensemble ou en utilisant la

Chapitre I : Description et présentation des outils du projet.

commande « Zoom » du menu « Affichage ».

3.1.6 La boîte à outils :

Elle est composée d'un ensemble d'icônes dont les fonctions seront détaillées ultérieurement et d'un sélecteur d'objet utilisé pour choisir les boîtiers, le style des pastilles, des traces, des traversées, etc...

3.2 Définition du logiciel ARES :

Le logiciel ARES est un outil d'édition et de routage qui complète parfaitement ISIS. Un schéma électrique réalisé sur ISIS peut alors être importé facilement sur ARES pour réaliser le PCB de la carte électronique. Bien que l'édition d'un circuit imprimé soit plus excitante lorsqu'elle est réalisée manuellement, ce logiciel permet de placer automatiquement les composants et de réaliser le routage automatiquement. (6)

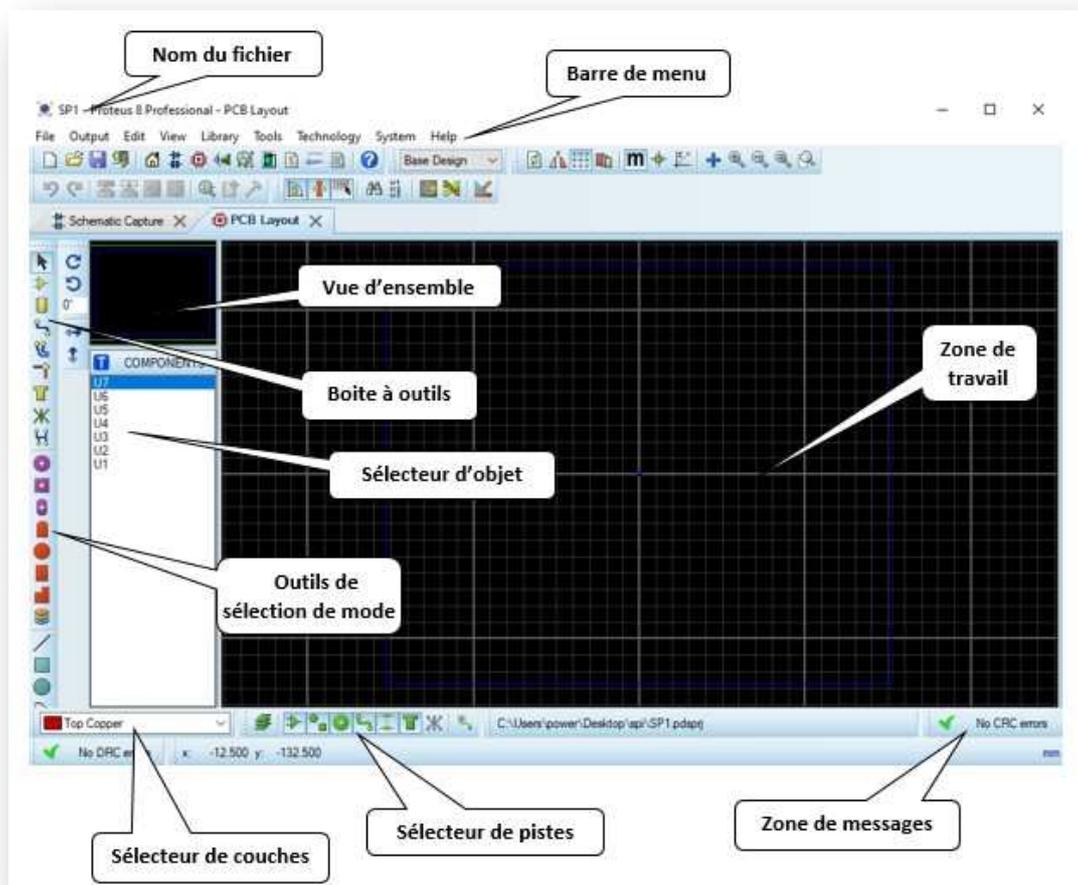


Figure I 7: Présentation de l'écran ARES.

Chapitre I : Description et présentation des outils du projet.

Le compilateur MikroC PRO :

3.2.1 Définition :

Le langage mikroC pour PIC a trouvé une large application pour le développement de systèmes embarqués sur la base de microcontrôleur. Il assure une combinaison de l'environnement de programmation avancée IDE, et d'un vaste ensemble de bibliothèques pour le matériel, de la documentation complète et d'un grand nombre des exemples.

La nouvelle version appelée mikroC PRO dispose de très nombreuses améliorations du compilateur mikroC : nouvelles variables utilisables, nouvelle interface IDE, amélioration des performances du linker et de l'optimisateur, cycle de compilation plus rapide, code machine généré plus compact (jusqu'à 40 % suivant les cas), nouveaux PIC supportés, Environnement de développement encore plus ergonomique, nouveaux exemples d'applications



Figure I 8: Logiciel mikroC PRO for PIC.

4 Conclusion :

Nous avons présenté dans ce chapitre les éléments matérielles et logiciels dont nous avons besoin pour concrétiser notre projet, cette étude théorique nous a permis de mieux comprendre chaque outil et c'est ce qui facilitera la partie pratique de cette réalisation.

Chapitre II

Etude du protocole de communication SPI

Chapitre II : Etude du protocole de communication SPI.

1 Introduction :

1.1 Origine

La sophistication et la complexité des appareils grand public ne permet plus l'échange entre composants électroniques par liaisons parallèles, ces liaisons devenaient de plus en plus nombreuses et encombrantes. Dans ces systèmes à base de microcontrôleurs (téléphones mobiles ...), les échanges de données entre les composants se font maintenant avec des bus série.

Dans le but de minimiser ces liaisons, et par conséquent d'en augmenter la fiabilité, a été créé le bus série SPI (Serial Peripheral Interface), initialement développé par Motorola.

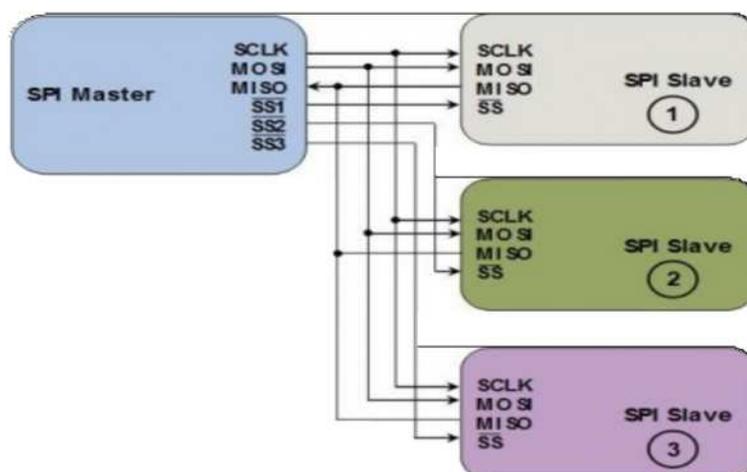
D'autres fabricants (Microchip, Atmel, Texas Instrument...) ont adopté pour ce type de liaison et de nombreux composants sont apparus (mémoires, capteurs, micro contrôleurs...). (2)

1.2 Présentation

Dans les circuits d'interface le bus SPI n'est pas seul est se trouve talonné depuis de nombreuses années par les circuits dotés d'une liaison série synchrone de type SPI. Cette appellation rencontre cependant des signaux et des chronogrammes qui sont beaucoup moins bien normalisés que ceux du bus I2C, tout simplement parce que la liaison du bus SPI ne fait l'objet d'aucune norme officielle.

Certains circuits indiquent qu'ils sont munis d'une interface SPI, mais vous pourrez aussi trouver comme appellation « Microwire » (qui est une marque déposée de National Semiconducteur) ou bien encore « bus série trois fils ». Un bus de ce type permet la connexion, sous forme série maître- esclave, de plusieurs circuits disposant d'interfaces compatibles, avec seulement trois fils de liaisons.

Les dernières versions d'Arduino incluent une bibliothèque qui permet la communication avec les périphériques SPI. (4)



1.3 Philosophie du bus SPI

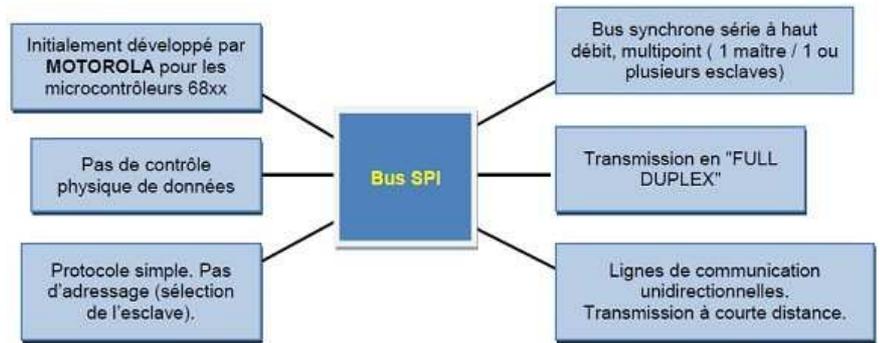


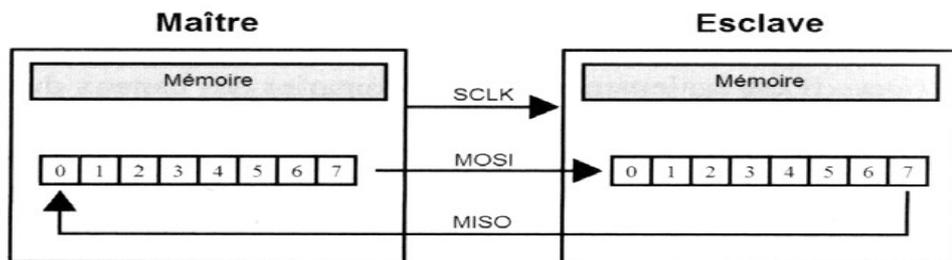
Figure II 2:philosophie du bus SPI.

2 Schéma de principe

La figure ci-dessous correspond au schéma de principe simplifié d'une liaison de données SPI, avec ses principaux composants.

Deux registres à décalages sont généralement mis en œuvre. Ils peuvent être de type matériel ou logiciel, selon les dispositifs employés. Par exemple, le RasPi implémente son registre à décalage De façon logiciel alors que le MCP3008 (ADC) est équipé d'un registre à décalage matériel. Que que soit leur mise en œuvre, ces deux registres à décalage forment un buffer circulaire inter puce, qui se trouve au cœur de la liaison SPI.

Les communications sont initiées par le maître, qui commence par sélectionner l'esclave. Au cours de chaque cycle d'horloge, le maître envoie un bit à l'esclave, qui le lit sur la ligne MOSI. En parallèle, l'esclave envoie un bit au maître, qui le lit sur la ligne MISO. Ces lectures et



Chapitre II : Etude du protocole de communication SPI.

écritures simultanées entre le maître et l'esclave forment une communication Full-Duplex. La fréquence de l'horloge dépend principalement des possibilités de réponse de l'esclave. Par exemple, le MCP3008 alimenté en 5 V est capable de gérer des fréquences de données allant jusqu'à 3,6 MHz. (9)

2.1 Format de transfert de données

Lorsque $CPHA = 0$, les données sont valides au premier front du signal d'horloge. La polarité $CPOL$ détermine s'il s'agit d'un front montant ou descendant.

En effet, pour $CPOL=0$, au repos, l'horloge est au niveau BAS ; le premier front est donc un front montant.

Pour $CPOL=1$, au repos, l'horloge est sur le niveau HAUT ; le premier front est donc un front descendant.

La polarité de l'horloge n'ayant pas d'influence sur le moment où le premier bit de données est valide elle n'a pas d'effet sur le format du transfert de données (voir figure ci-dessous). (4)

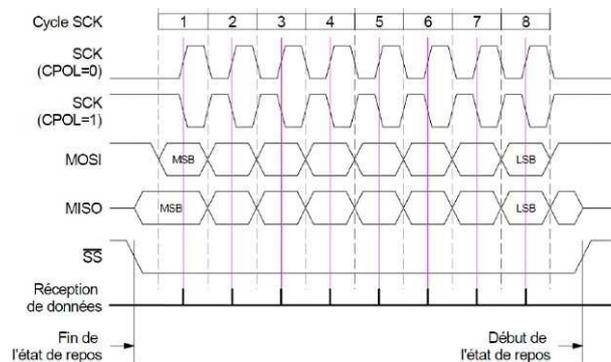


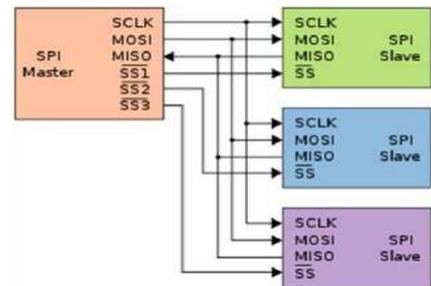
Figure II 4:format de transfert de données

2.2 Synoptique d'une liaison SPI Maître-Multi-Esclaves

Le maître sélectionne un seul et unique esclave avec lequel il veut rentrer en communication par la mise à niveau logique zéro de /SS 1 2 3, puis, après 8 fronts d'horloge, l'octet de donnée est transféré.

La patte MISO de l'esclave non sélectionné est à l'état haute impédance.

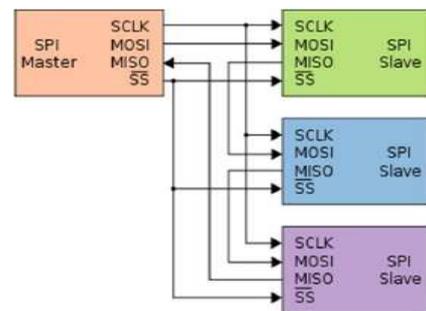
La seule limite aux nombres d'esclaves est en fait la possibilité de broches SS du maître.



Cas de la Daisy Chain (esclave en guirlande, en cascade...)

Le maître sélectionne tous les esclaves par la mise à niveau logique zéro de /SS, puis après 3*8 fronts d'horloge, les 3 octets de données sont transférés (dans le cas d'un octet par esclave).

Cette disposition permet de réduire le nombre de lignes /SS, mais en contrepartie il faudra un "buffer" plus grand dans le maître (ou une gestion du soft plus élaborée). (10)



3 Bus SPI :

3.1 Définition :

Le terme SPI signifie Serial Périphérie Interface (Périphérique d'Interface Série). C'est un protocole de communication courant, utilisé pour échanger des données, en lecture/écriture, entre plusieurs périphériques, dont un seul doit être configuré comme Master, généralement un Microcontrôleur, et tous les autres comme Slaves. Ces derniers peuvent être des cartes SD, des DAC, des ADC, des DS1306 Real Time Clocks, des contrôleurs d'affichage (MAX2719 serial display driver), des 25LC256 Serial EEPROM, des capteurs intelligents, des registres à décalage, et bien plus encore. (10)

3.2 Fonctionnement :

Tout échange d'informations devrait passer par le Master : Entre le Master et les Slave le transfert d'informations est direct, par contre si deux Slaves envisagent de communiquer entre eux, ils devront utiliser le Master comme intermédiaire. La communication série SPI est SYNCHRONE et FULL DUPLEX (duplex intégral). Synchrone parce qu'elle fonctionne à l'aide d'un signal d'horloge partagé entre les membres de cette configuration, et cette horloge est fourni par le Master. FULL DUPLEX parce que l'échange d'information se fait dans les deux sens au rythme de l'horloge. Outre sa vitesse de transfert de données supérieure ; SPI utilise également un protocole de transfert de données très simple par rapport aux autres méthodes de transfert de données en série. Le Master utilise le même registre (8 bits) pour envoyer/recevoir les données depuis les Slaves, et chacun de ceux-ci utilise son propre registre (8 bits) pour envoyer/recevoir les données depuis/vers le Master. Le transfert complet d'un caractère s'effectue en seulement 8 cycles d'horloge, ce qui consommerait 0,8 us si on retient l'horloge de 10 MHz.

Le mode SPI permet la transmission et la réception synchrone de 8 bits de données simultanément. Trois pines sont utilisées à cette fin.

Serial Data Out (SDO) RC5/SDO : Utilisée par le Master, pour envoyer les données aux esclaves et par les esclaves pour envoyer des données au Master. Le bit MSB est transféré en premier.

Chapitre II : Etude du protocole de communication SPI.

- Serial Data In (SDI) RC4/SDI/SDA : Utilisée par les Slaves pour recevoir les données depuis le Master et par le Master pour recevoir des données depuis les Slaves. Depuis cette entrée, les données sont reçues avec le bit MSB en premier. Le registre de décalage (SSPSR) du module SPI assure le bon positionnement des bits dans le buffer de réception (SSPBUF) de sorte que les bits de la donnée reçue soient correctement positionnés.
 - Serial Clock (SCK) RC3/SCK/SCL : C'est l'horloge qui cadence l'échange sériel d'informations. Une quatrième pince peut également faire partie de cet ensemble.
- Slave Select (SS) RA5/AN4/SS : Utilisée par le Master pour sélectionner lequel de ses Slaves est concerné par l'échange. Généralement cette ligne est active à l'état bas, parce que le CS de la plupart des boîtiers est actif à l'état bas. N'importe quelle pince, configurée comme sortie logique, peut être utilisée.

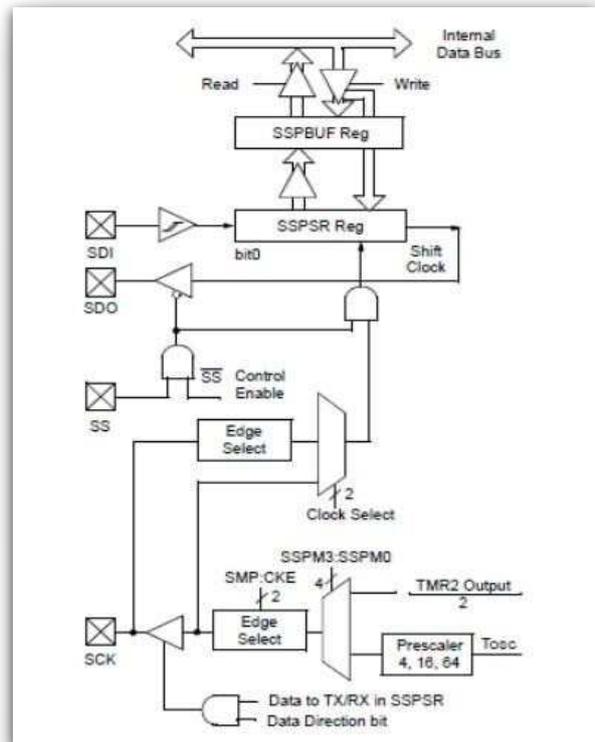


Figure II 5: Module MSSP en mode SPI intégré au PIC 16F877.

Chapitre II : Etude du protocole de communication SPI.

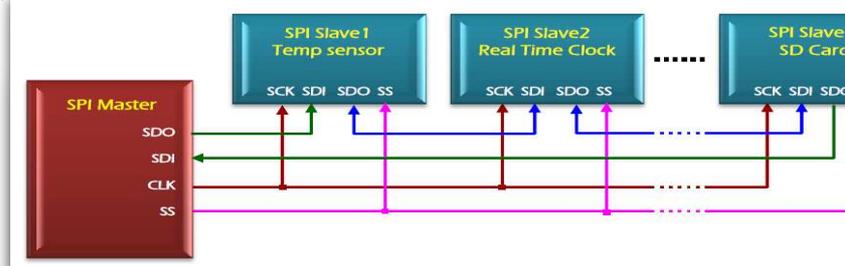
Pour la sélection des Slaves. Le mode SPI propose une seule ligne (RA5/AN4/SS) comme sortie de sélection du Slave, comme cette ligne est multiplexée avec l'entrée analogique AN4, il faut initialiser le registre ADCON1 de sorte à la configurer comme une entrée/sortie digital.

Les deux registres (SSPCON et SSPSTAT) permettent de spécifier les options de fonctionnement en mode SPI à savoir :

- Master mode, SCK représente la sortie d'horloge générée par le Master
- Slave mode, SCK représente l'entrée d'horloge vers le Slave
- Clock Polarity: L'état de repos de l'horloge (Idle state of SCK)
- Phase d'échantillonnage des données en entrée (data input sample phase) : Spécifie le temps d'échantillonnage des données en milieu ou en fin (middle or end of data output time)
- Front d'horloge (Clock edge): spécifie le front d'échantillonnage des données, front montant ou front descendant (output data on rising/falling edge of SCK)
- Fréquence d'horloge (Clock Rate): Uniquement pour le Master
- Mode de fonctionnement du Slave (Slave Select mode) : uniquement pour les Slaves Tous les échanges ne peuvent être initiés que par le Master, et la broche SCK partage le signal d'horloge généré par celui-ci avec tous les Slaves. Cette horloge peut être dérivée de l'horloge du système ou fournie par le timer2. Elle n'est active que pendant les transmissions. Si aucune transmission n'a lieu, la sortie horloge du Master est au niveau du repos (fixé pendant la configuration du PIC Master). (10)

3.3 Configuration du bus SPI :

Le Bus SPI peut avoir plusieurs Slaves dont chacun dispose de son propre SS (Chip Select), et le Master devrait disposer de suffisamment de lignes pour sélectionner indépendamment chacun d'entre eux. Le nombre de ces lignes est fonction du nombre de Slaves. Deux configurations sont possibles :



Chapitre II : Etude du protocole de communication SPI.

Ce type de structure est généralement utilisé dans des situations de sortie uniquement, telles que le contrôle de voyants (LED) pour lesquels il n'y a pas de signal de retour vers le Master. Dans ces cas, on peut laisser la ligne SDI (MISO) du Master déconnectée. Toutefois, si les données doivent être renvoyées au Master, on peut le faire en fermant la boucle (Daisy Chain) (fil vert dans la figure ci-dessus). Notez que si on procède ainsi, les données renvoyées par le Slave 1 devront propager le long de tous les autres Slaves avant de revenir au maître. Il faut donc veiller à envoyer suffisamment de commandes de réception pour obtenir les données dans ton besoin.

3.3.1 Configuration en bus :

C'est la configuration dont SPI est normalement utilisé. Les membres de cette configuration sont disposés en Bus, et le nombre de lignes de sélection dont doit disposer le Master, pour gérer la communication, est fonction du nombre de Slave mis en joues. Tous les Slaves sont à l'écoute, mais seul le Slave qui voit son SS (CS) à l'état bas est concerné par le transfert.

Étant donné que les broches SDO/SDI des Slaves sont connectées ensemble, elles doivent être des broches logiques à trois états.

Les périphériques Slaves dont ces broches ne prennent pas en charge les trois états peuvent être utilisés en ajoutant des tampons à trois états contrôlés par le signal Chip Select.

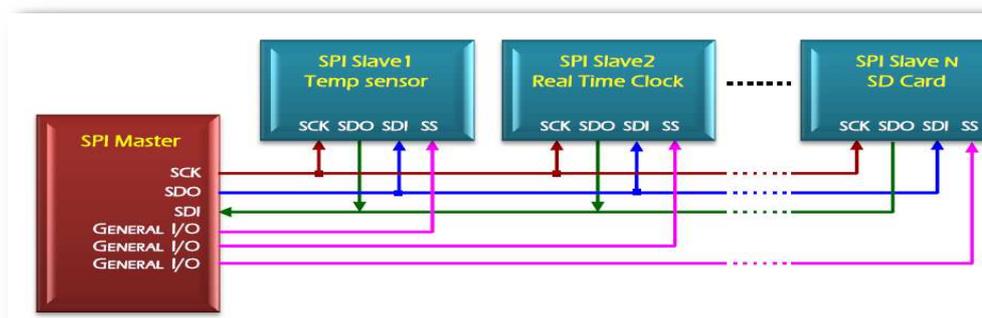


Figure II 6: Configuration en mode BUS d'une communication série SPI.

3.4 Protocole de gestion d'un réseau SPI :

Un réseau SPI est la composition d'un Master SPI et un certain nombre de périphériques Slave SPI. Pour que les membres de ce réseau se comprennent, une politique de dialogue est plus que nécessaire. Deux possibilités sont alors offertes :

3.4.1 La Scrutation :

Dans laquelle le Master SPI interroge, à tour de rôle, chacun des Slaves SPI pour savoir s'il envisage ou non d'opérer un transfert. Cette politique souffre de deux inconvénients majeurs :

- Perte de temps non justifiée quant à interroger des périphériques qui, peut-être, n'ont jamais besoin de service (échanger de données).
- Risque de s'occuper tardivement de périphériques qui ont besoin de service urgent.

3.4.2 L'interrogation :

Pour contourner les deux inconvénients de la première technique, cette technique adopte plutôt une vision inverse dans laquelle c'est le Slave, qui envisage d'opérer un transfert, qui informe le Master, à travers sa (ses) ligne(s) d'interruption(s), de cette intention. En réponse, le Master lui accorde (ou non) cette requête et le transfert aura lieu.

3.5 Flux de données :

Le transfert série en mode SPI est en FULL DUPLEX (duplexe intégral), c'est-à-dire que chaque membre de ce transfert peut émettre et recevoir en même temps. Ceci est rendu possible grâce à une organisation spéciale du registre de décalage du module SPI. La figure ci-dessous explicite cette idée. (10)

3.6 Le 16F877 et le mode SPI :

Le mode Master :

- À tout moment, le Master (maître) peut initier le transfert de données parce qu'il contrôle l'horloge SCK ;
- Le Master détermine quand l'esclave (Slave) (Processeur 2) peut émettre les données par protocole software ;
- Si le module SPI est à utiliser en réception uniquement, la sortie SDO doit être désactivée (programmée comme entrée) ;
- La polarité de l'horloge est sélectionnée par la programmation appropriée du bit CKP (SSPCON<4>) ;

Chapitre II : Etude du protocole de communication SPI.

- En mode Master, le débit de l'horloge SPI (bit rate) est fixé par programme par l'utilisateur à l'un des débits suivants :
 - FOSC/4 (or TCY)
 - FOSC/16 (or 4 * TCY)
 - FOSC/64 (or 16 * TCY)
 - Timer2 output/2

3.6.1 Le mode slave :

- En mode Slave, les données sont transmises où reçues quand les impulsions d'horloge apparaissent sur la pîne SCK. Quand le dernier bit est Latché, le bit d'interruption SSPIF : PIR1<3> ainsi que le bit BF:SSPSTAT<0> sont mis à un (set)
- Dans ce mode l'horloge SCK provient du Master et est appliquée à la pîne SCK ;
- En mode sleep (sommeil), le Slave peut emmètre/recevoir les données. A la réception d'un octet le microcontrôleur est réveillé (wake-up). (11)

3.7 Comparaison entre le I2C et le SPI :

Tableau 1.II : : Comparaison entre le protocole I2C et SPI.

I2C	SPI
S'il existe des variations de l'I2C qui montent au-dessus de 1MHz, la grande majorité des implémentations que l'on trouve utilisent Généralement Même 100 ou 400 kHz.	Pour le SPI il est possible de trouver certains composants au-delà de 20 Mbits.
C'est un véritable protocole qui permet l'interconnexion de multiples boitiers dans différentes configurations : Maitre / Esclave, Maitre / Multiple esclaves, Multiple Maitres / Multiples esclaves.	En général point à point, bien que l'on puisse connecter plusieurs esclaves mais il faut ajouter des lignes supplémentaires. Un seul maitre génère l'horloge.
Dû à la configuration collecteur/ drain ouvert Sur les 2 lignes de transmission (SDA + SCL), consommation relativement élevée.	Signaux de type TTL/CMOS Consommation faible.

4 Conclusion :

Après l'étude du protocole de communication SPI, nous avons pris connaissance de ces diverses propriétés ce qui nous a permis de mieux l'exploiter en tant qu'unité de contrôle pour notre système électronique.

Chapitre III

Conception et Réalisation pratique



1. Introduction :

Dans les chapitres précédents nous avons fait une étude générale de tous les éléments constituant notre projet et c'est ce qui nous a facilité la réalisation pratique.

Le système électronique proposé dans ce projet consiste à afficher des caractères sur des matrices LED 8x8 commandés par un microcontrôleur PIC16F877 en utilisant le protocole de communication SPI.

Ce chapitre sera consacré à la description des différentes étapes de conception et réalisation physique de notre projet.

2. Description du système :

2.1 Schéma synoptique :

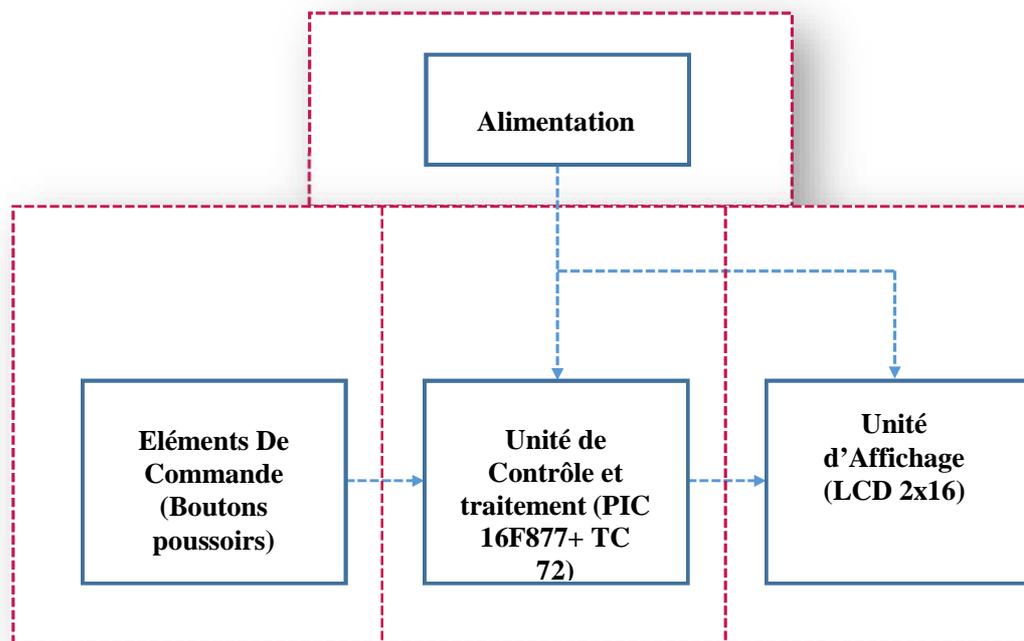


Figure III 1:Schéma synoptique du circuit.

Présentation des blocs :

Le schéma de la figure ci-dessus donne une vue générale sur le principe de fonctionnement de notre montage, il est articulé par quatre principales parties fonctionnelles :

- **Alimentation** : une batterie de 5v et 12v.
- **Éléments de commande** : boutons poussoirs.
- **Unité de contrôle de traitement** : PIC16F877 et capteur température TC 72.
- **Unité d'affichage** : afficheur LCD 2x8.

3. Simulation du circuit :

Avant de commencer à implémenter réellement notre système, nous nous sommes tournés vers des simulations de différentes parties du système. Pour cela, nous utilisons le logiciel ISIS Proteus, qui est un très bon logiciel de simulation électronique.

3.1 Schéma électronique :

Afin de créer un schéma électronique de notre carte, nous avons utilisé le logiciel ISIS Proteus. Pour le saisir, nous avons créé un nouveau projet à l'écran, puis placé les composants que nous avons sélectionné dans la bibliothèque de composants dans l'espace de travail.

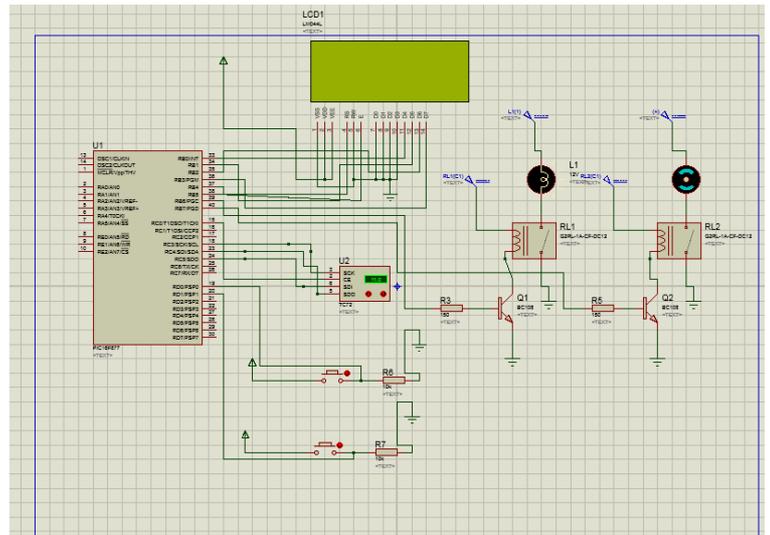


Figure III 2: Schéma électronique globale du circuit.

3.2 Programmation du pic 16F877 :

Pour la programmation de notre microcontrôleur nous avons choisi la façon la plus simple et efficace et cela en utilisant le compilateur Mikro C pro for pic.

Pour la création d'un nouveau projet, le saisi et la compilation du programme nous avons suivi les étapes suivantes :

- Double cliquer sur l'icône du logiciel Mikro C PRO FOR PIC pour le lancer.
- Ensuite on clique sur « new project » pour créer un nouveau projet.

Chapitre III : Conception et réalisation pratique.

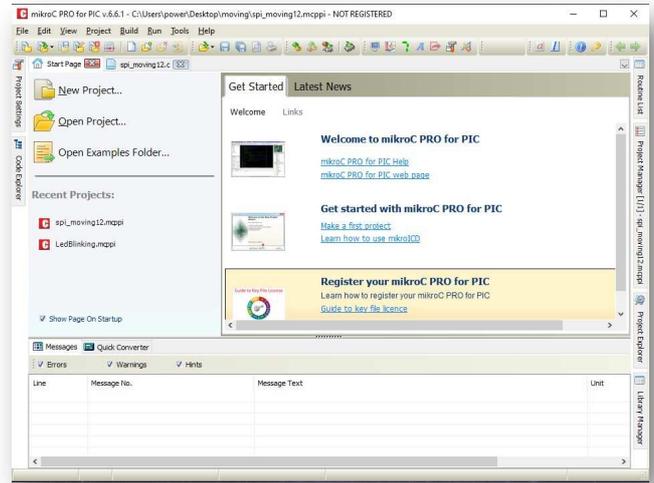


Figure III 3: Création d'un nouveau projet.

- Après l'action précédente une fenêtre apparaîtra, celle-ci va nous permettre de :
 - Nommer notre projet.
 - Choisir son emplacement.
 - Sélectionner la référence du pic avec lequel on souhaite travailler donc on sélectionne le P16F877.
 - Définir la fréquence d'oscillation avec laquelle travaille notre pic, dans ce cas on sélectionne 8.000000 MHz.

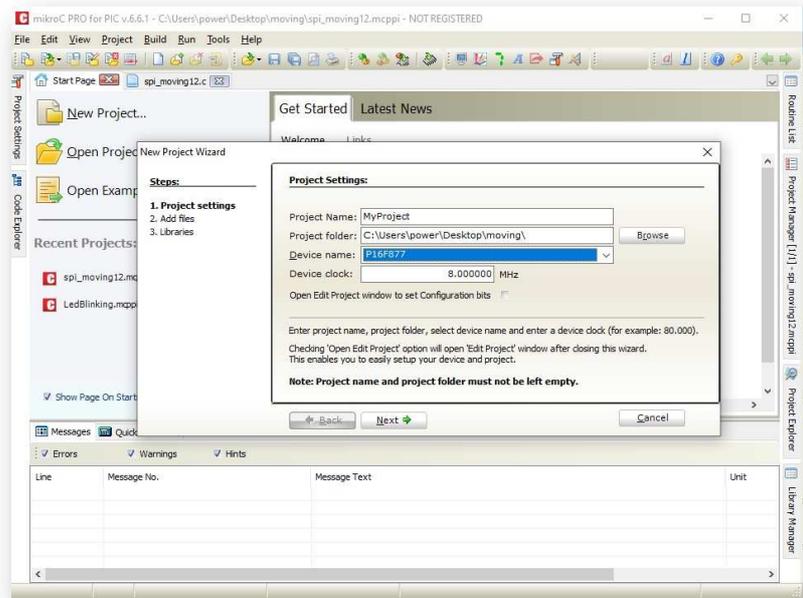


Figure III 4: Choix des paramètres du projet.

On appui sur « Next » et enfin notre projet est créé et la configuration est terminée, maintenant cette fenêtre s'affiche pour qu'on puisse écrire notre programme dedans.

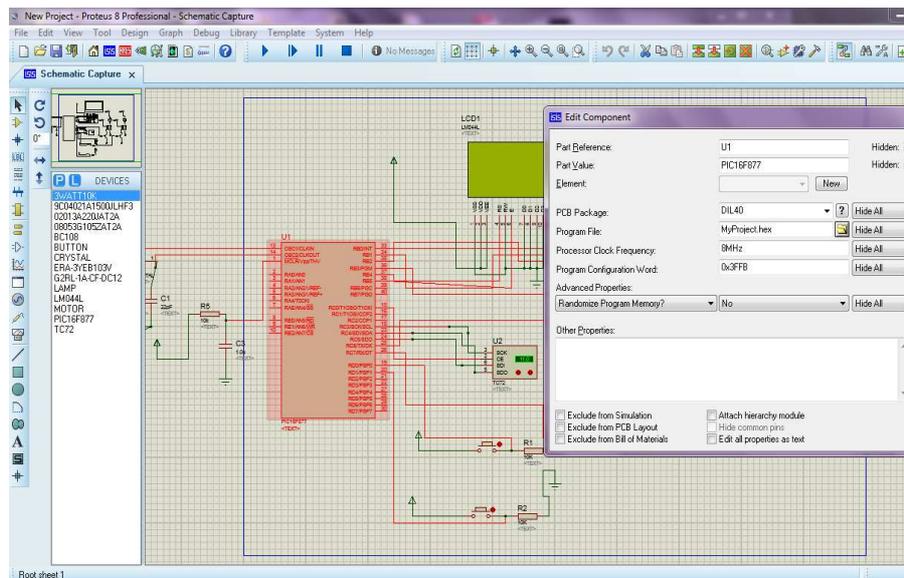
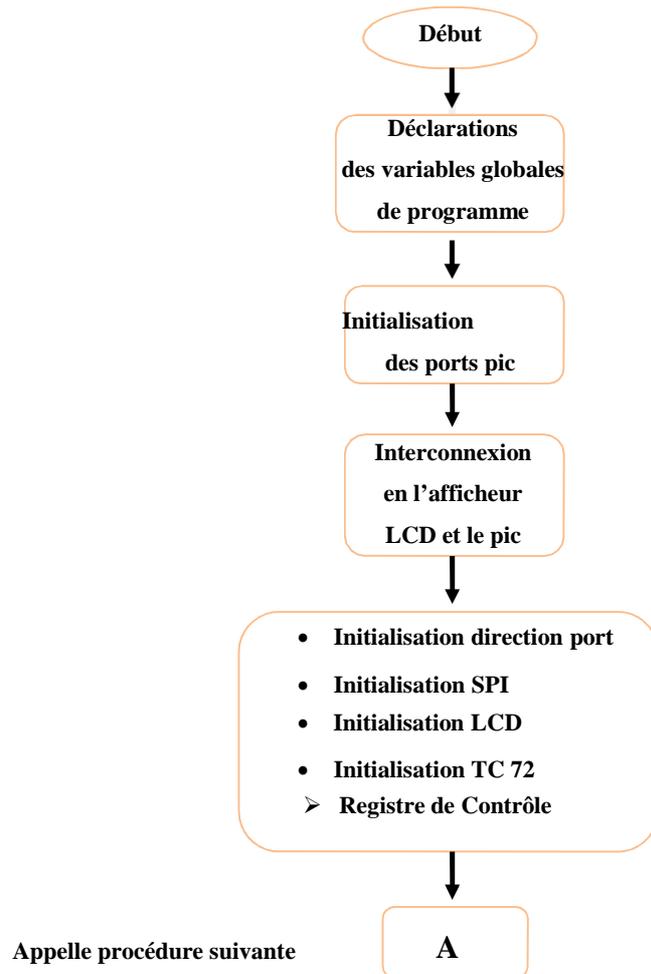


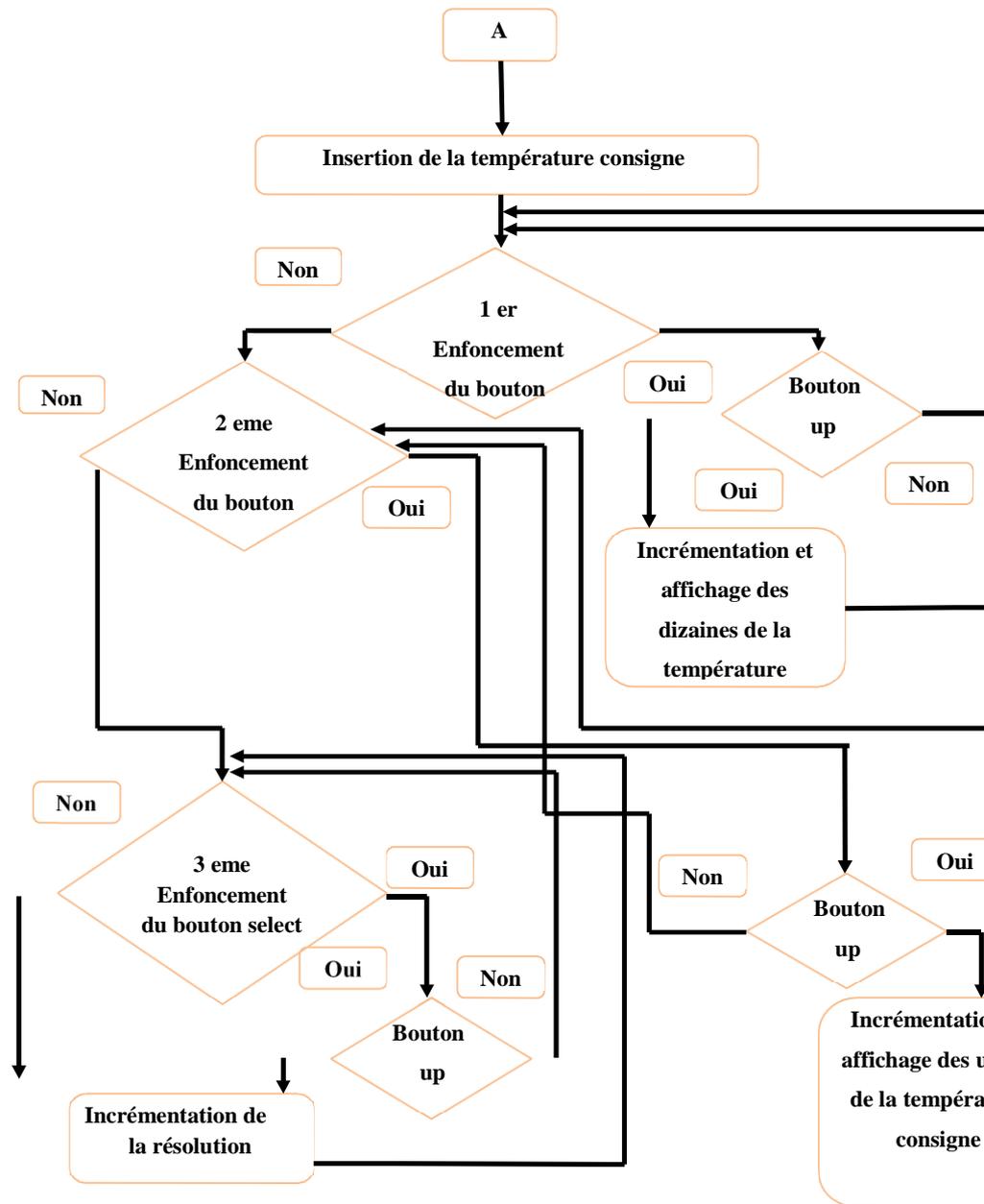
Figure III 6: Chargement du programme dans le PIC sur ISIS.

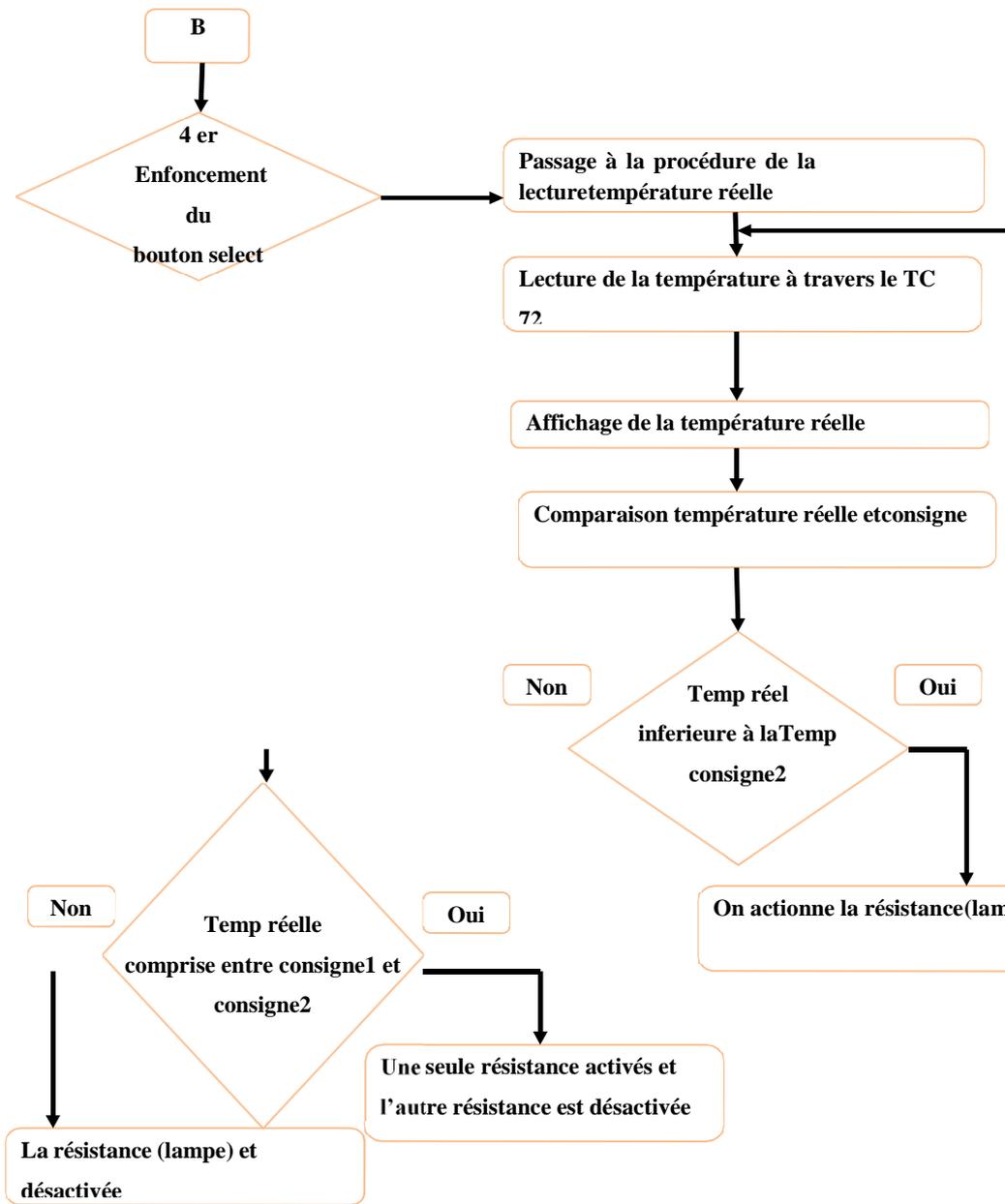
- On entre sur (program file) et on apporte le fichier (file.HEX) puis on appui sur ok.
- Le programme est chargé dans le PIC16F877.
- On clique sur le bouton (play) pour commencer la simulation.

3.3 Organigrammes :

L'organigramme ci-dessous est une représentation générale du programme d'affichage







3.4 Routage et création du circuit imprimé :

Avant de passer à la réalisation matérielle de notre carte électronique nous allons d'abord réaliser le schéma du circuit imprimé et cela en utilisant le logiciel ARES pour simuler les connexions de tous les composants implantés sur notre carte.

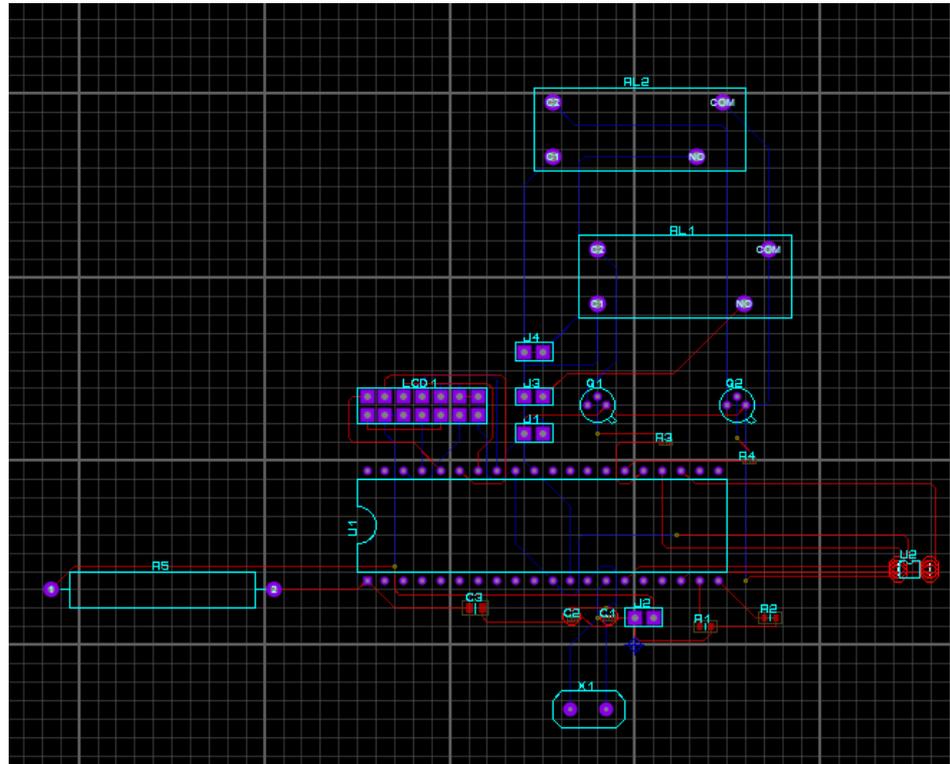


Figure III 7:Schéma de routage.

4. Réalisation pratique :

4.1 4.1 Listes des composants :

- **Un microcontrôleur PIC16F877 :**



- Ecran LCD 2x16:



- Condensateur 22 pF :



- Capteur température TC 72 :



Chapitre III : Conception et réalisation pratique.

- Résistances 1K Ω :



- Boutons poussoirs :



- Un Quartz 8 MHz :



Relay12v :



Un Connecteur 40 pins :



4.2 Gravure du programme sur le pic :

Après la simulation du programme avec le logiciel mikroC for pic pro, il est prêt à être chargé dans le pic et pour le faire nous avons utilisé le logiciel PICkit2 qui nous a permis de transférer le programme en hexadécimale vers le pic. Ensuite nous avons vérifié en cliquant sur « vérifie » pour s'assurer du transfert complet du programme.

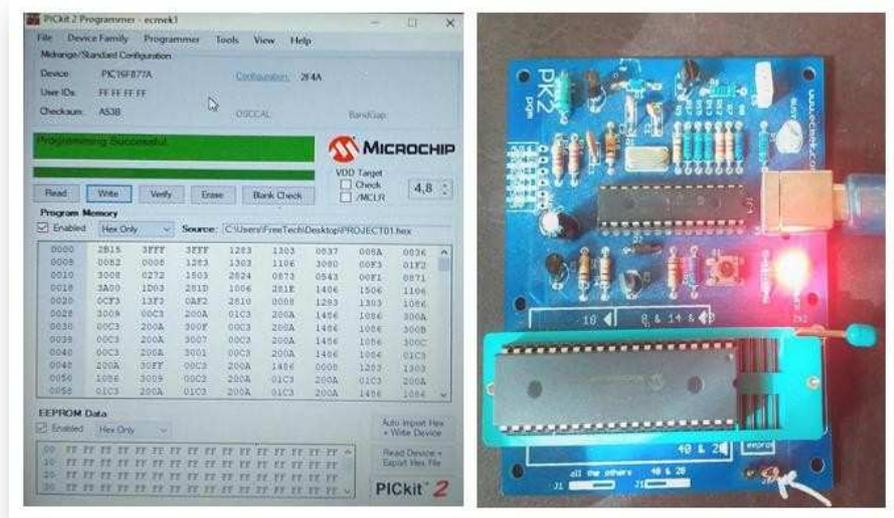


Figure III 8 : Gravure du programme sur le Pic.

4.3 Réalisation du circuit imprimé et implantation des composants :

Après la conception logiciel du circuit et le routage nous avons obtenu le schéma du circuit imprimé, qui nous a ensuite permis d'imprimer le typon sur un transparent spécial et grâce à une imprimante laser.

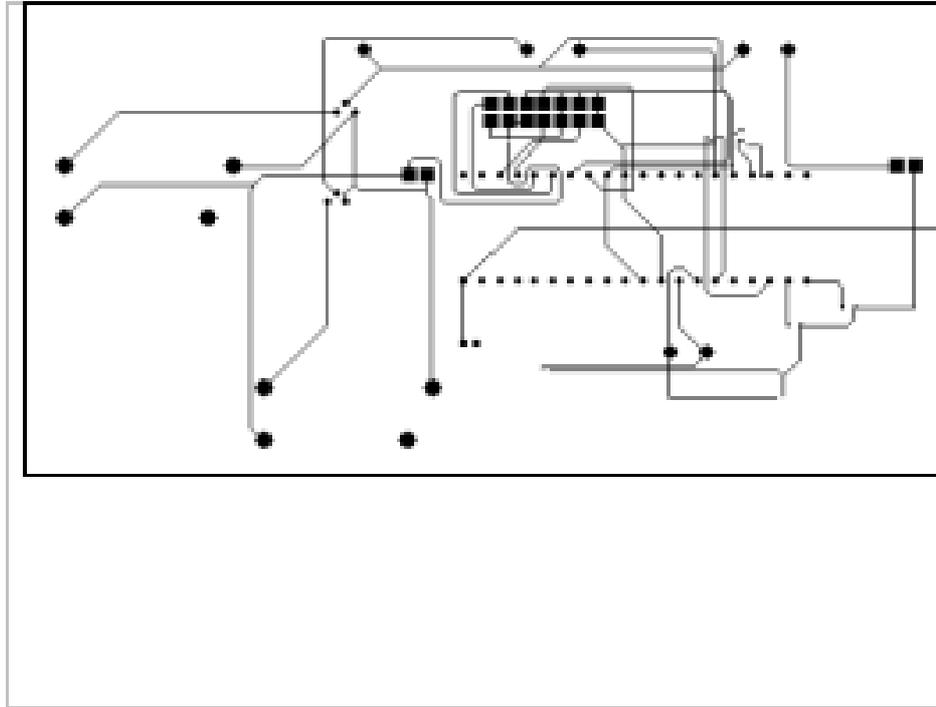


Figure III 9: Le typon.

Chapitre III : Conception et réalisation pratique.

On va donc déposer le typon sur la face cuivrée de la carte époxy et démarrer le transfert du circuit en posant un fer à repasser bien chauffé sur le typon. Après cette opération il faut la refroidir, élever le typon et sécher la carte.

On passe ensuite à la gravure, qui consiste en l'élimination du cuivre inutile afin de n'avoir que les pistes de connexion qui nous seront utiles. La gravure pourra se faire avec de l'acide ou au moyen d'une insoleuse UV. Puis on doit percer les trous du montage pour pouvoir placer les composants.

Maintenant que notre circuit imprimé est terminé, on dépose les composants sur la plaque tout en vérifiant le positionnement et le bon sens de chacun d'entre eux, on procède ensuite à la soudure des composants, et enfin notre carte électronique est prête.

5 Conclusion :

Au niveau de ce dernier chapitre nous avons présenté les différentes étapes de réalisation pratique de notre projet, cela nous fait une expérience très enrichissante, à savoir la programmation du PIC16F877 avec le logiciel mikroC, la conception grâce à PROTEUS qui permet de réaliser facilement des schémas fonctionnels et électriques puis de gérer le routage du circuit. Ensuite la fabrication de la carte électronique en passant par plusieurs étapes.

Conclusion générale

Le but de ce projet est de nous familiariser avec le fonctionnement et l'utilisation des microcontrôleurs, qui sont devenus aujourd'hui des composants électroniques clé et incontournables pour tous systèmes automatisés.

À l'issue de ce projet nous avons été capable de :

- Concevoir et utiliser une diversité de matériels et logiciels qui nous ont été utile.
- D'approfondir nos connaissances dans la conception des cartes et circuits imprimés.
- La programmation des microcontrôleurs PIC.
- La mise en pratique des connaissances théoriques en électronique et en programmation mikroC PRO FOR PIC.

L'objectif de ce projet est d'étudier les microcontrôleurs PIC et leurs applications en général avec un PIC de la famille 16F877 et de réaliser un objet physique qui peut nous aider à concevoir ces applications.

Donc pour atteindre tous ces objectifs on a procédé comme il suit :

- À travers du premier chapitre, nous avons présenté les différents outils et logiciels utilisés.
- Le deuxième chapitre, a été consacré à l'étude du PIC 16F877 et ses protocoles de communication.

Le troisième et le dernier chapitre évoque les différentes étapes de la conception et de la réalisation du projet tout en illustrant ça avec des schémas synoptiques.

Ce projet a été très bénéfique sur plusieurs plans. D'une part, il nous a permis d'appliquer nos connaissances acquises pendant cinq ans d'étude et d'analyser le comportement interne d'un microcontrôleur PIC 16F877. D'une autre part, il nous a permis de toucher à la réalisation et la conception et savoir les différentes étapes à suivre pour avoir un circuit imprimé.

En fin ce projet, est le début d'un long voyage dans une science qui penche à trouver des solutions et à résoudre les différentes problématiques rencontrées au cours de notre vie quotidienne.

Bibliographie

1. **Toujeni, Nizart.** Introduction sur les microcontrôleurs. Ecole Iset Nabeul. : s.n.
2. **Weiss, Jaques.** Microcontrôleurs PIC. février 2002.
3. **Oumnad, Abdelmajid.** Les microcontrôleurs 'Etude détaillée du PIC16F877.
4. **<https://www.digikey.fr>.** digikey. [En ligne] **<https://www.digikey.fr/fr/articles/why-how-to-use-serial-peripheral-interface-simplify-connections-between-multiple-devices>**.
5. **Sarhene, Hamid.** Cours Electronique. 2020.
6. **www.elektronique.fr.** elektronique. [En ligne] **<http://www.elektronique.fr/logiciels/proteus.pdf>**.
7. **<https://www.electronique-mixte.fr>.** electronique-mixte. [En ligne] **[https://www.electronique-mixte.fr/mikroc/..](https://www.electronique-mixte.fr/mikroc/)**
8. **Tavernier, Christian.** Programmation en C des PICs. Paris : s.n., 2005.
9. **Kammoun, Y.Rkhissi.** Cours microcontrôleurs. Ecole Iset Nabeul : s.n.
10. **T.Berenguer.** Le bus SPI.
11. **Dam, Bert Van.** 50 nouvelles applications des microcontrôleurs PIC. 2010.
12. **Tavernier, Christian.** Les microcontrôleurs PIC 'recueil d'application. PARIS : 3^{ème} édition, 2005.
13. **Tiana, Razafiarisera Ralay.** Mémoire de fin d'étude "Résistivimetre à base de PIC16F877". université d'Antananarivo : s.n., 2007.
14. **Turner, Thomas Grenier & Dominique.** Microcontrôleurs 'Famille PIC 16.
15. **www.les-electroniciens.com.** les-electroniciens. [En ligne] **<https://www.les-electroniciens.com/sites/default/files/cours/hcs12.pdf>** .
16. **(ICCI), 17th international conference on computer and information technology.** 2014.
17. **Hammanur, Karthik.** Inter Integrated Circuit. 2005.
18. **<http://projet.eu.org>.** projet.eu. [En ligne] **http://projet.eu.org/pedago/sin/term/8-bus_SPI.pdf**.

Annexe :

PIC 16F877 :



MICROCHIP

PIC16F87X

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

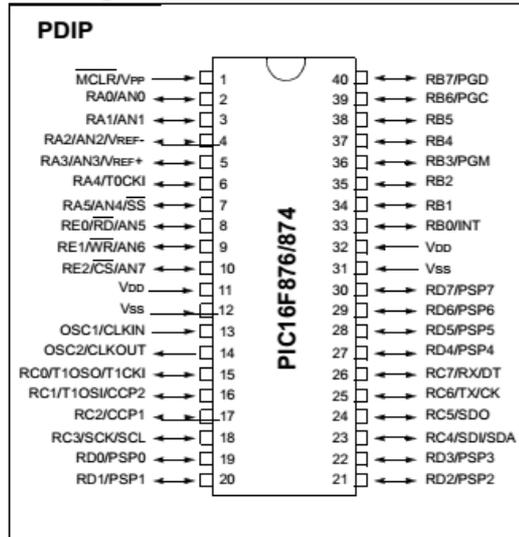
Devices Included in this Data Sheet:

- PIC16F873 • PIC16F876
- PIC16F874 • PIC16F877

Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,
Up to 368 x 8 bytes of Data Memory (RAM)
Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and
Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC
oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM
technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two
pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial, Industrial and Extended temperature
ranges
- Low-power consumption:
 - < 0.6 mA typical @ 3V, 4 MHz
 - 20 µA typical @ 3V, 32 kHz
 - < 1 µA typical standby current

Pin Diagram



Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler,
can be incremented during SLEEP via external
crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period
register, prescaler and postscaler
- Two Capture, Compare, PWM modules
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master
mode) and I²C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver
Transmitter (USART/SCI) with 9-bit address
detection
- Parallel Slave Port (PSP) 8-bits wide, with
external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for
Brown-out Reset (BOR)

TC 72 :

Electrical Specifications: Unless otherwise noted, all parameters apply at $V_{DD} = 2.65V$ to $5.5V$, $T_A = -55^{\circ}C$ to $+125^{\circ}C$.						
Parameters	Sym	Min	Typ	Max	Units	Conditions
Power Supply						
Operating Voltage Range	V_{DD}	2.65	—	5.5	V	Note 1
Operating Current: Normal Mode, ADC Active	I_{DD-CON}	—	250	400	μA	Continuous temp. conversion mode (Shutdown Bit = '0')
Shut-Down Supply Current	I_{SHD}	—	0.1	1.0	μA	Shutdown Mode (Shutdown Bit = '1')
Temperature Sensor and Analog-to-Digital Converter						
Temperature Accuracy (Note 1)	T_{ACY}	-2.0	—	+2.0	$^{\circ}C$	$-40^{\circ}C < T_A < +85^{\circ}C$
		-3.0	—	+3.0		$-55^{\circ}C < T_A < +125^{\circ}C$
Resolution		—	10	—	Bits	Note 4
ADC Conversion Time	t_{CONV}	—	150	200	ms	
Digital Input / Output						
High Level Input Voltage	V_{IH}	$0.7 V_{DD}$	—	—	V	
Low Level Input Voltage	V_{IL}	—	—	$0.2 V_{DD}$	V	
High Level Output Voltage	V_{OH}	$0.7 V_{DD}$	—	—	V	$I_{OH} = 1\text{ mA}$
Low Level Output Voltage	V_{OL}	—	—	$0.2 V_{DD}$	V	$I_{OL} = 4\text{ mA}$
Input Resistance	R_{IN}	1.0	—	—	$M\Omega$	
Pin Capacitance	C_{IN}	—	15	—	pF	
	C_{OUT}	—	50	—		

Résumé :

Les systèmes de régulation de température sont divers et varient selon les appareils électroniques et dépendent surtout de leurs domaines d'application.

Dans notre projet, nous avons utilisé un microcontrôleur Pic 16F877, un bus de communication SPI. Pour mesurer la température un capteur TC 72 sera mis en œuvre et les résultats seront afficher par la suite dans un écran LCD.

Tous cet élément cité constitue la carte électronique d'une couveuse d'un prématuré.

Abstract:

Systems of temprature regulation are differents and vary according to electroniquecartes and their domain of application.

In our project ,we have used a microcontroler Pic 16f877 , a bus of communication SPI , a capter TC 72. In order to show results we have also used an LCD screen.

All these elements constitute an electrique board of an incubator.