

III.1 INTRODUCTION

Dans ce chapitre, nous allons présenter, d'expliquer et de montrer toutes sortes de cartes à puce et bien connu qui à partir de la position dans un lecteur et de lire le code secret est dans, et soient conformes à l'intérieur **pace 1 tefdhhp** et quand sa santé il est transfère à une base de **don_ onées** dans l'ordinateurs via le port parallèle et par conséquent est de montrer une photo et toutes les informations de la personne ou le groupe dans un tableau sur la forme de l'interface.

III.2 STRUCTURE MATERIELLE

Le schéma bloc de des circuits réalisés est représenté par la figure (III.1).

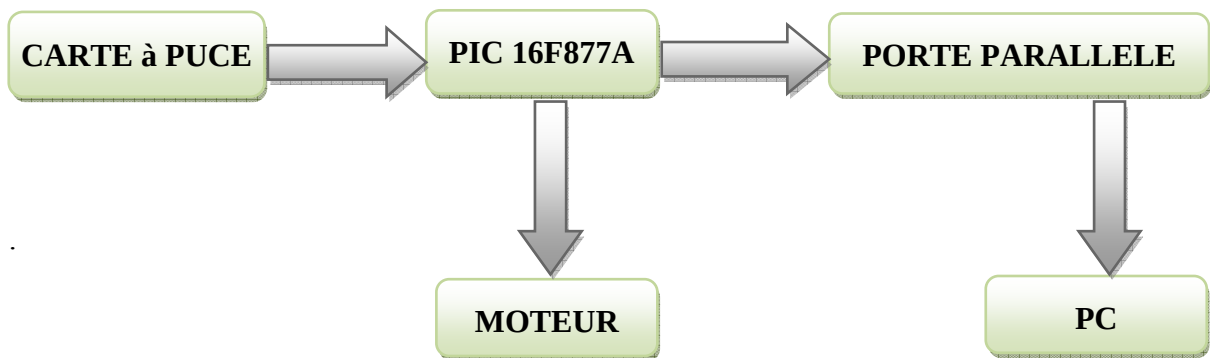


Figure (III.1) : Schéma bloc de structure

La structure matérielle (hardware) est présentée en ce qui suit :

III.3 LES CARTES A PUCE

III.3.1 Historique et Définition

III.3.1.1 Historique

En 1976, la direction générale des télécommunications (DGT, l'actuelle France Télécom) recherchait une nouvelle forme de communication afin de créer des réseaux permettant d'exécuter un transfert électronique de fonds à partir de guichets bancaires automatiques et de terminaux à domicile en toute sécurité, par l'identification de l'utilisateur. À cette époque, la carte à puce était au croisement de trois techniques différentes : la micro-électronique, le traitement de données et la cryptographie.

En 1979, une coopération entre CII-Honeywell Bull et Motorola donnait naissance à la première carte à puce opérationnelle. La carte à puce fut choisie par la DGT pour la toute première expérimentation en matière de télépaiement dans le monde, en utilisant la chaîne de

télévision Antiope pour transmettre des informations aux détenteurs de cartes et le réseau téléphonique pour le paiement électronique.

Entre 1982 et 1984, les banques françaises conduisirent des expérimentations dans trois villes différentes afin de tester la fiabilité technique et économique de la carte dans des conditions réelles d'exploitation et déterminer des spécifications définitives.

Ce projet mettait en oeuvre 125 000 cartes et 650 terminaux. Une carte logique câblée était distribuée par Schlumberger et Philips tandis que CIIHoneywell Bull distribuait une carte à puce.

Seule l'expérience menée avec la carte à puce s'est avérée concluante; sa généralisation sur l'ensemble du territoire français fut réalisée en 1986. Pendant ce temps, la « télécarte » fabriquée par Schlumberger, une carte à mémoire prépayée pour le téléphone, fit son apparition en 1983.

Depuis, d'importants progrès réalisés dans la technologie des semi-conducteurs ont eu des répercussions sur les cartes à puce en matière de taille des mémoires, de performance, de consommation de puissance, de sécurité et de coûts. Plus d'un milliard de cartes à puce sont présentement en service dans différents domaines de l'activité socio-économique.

Selon certaines analyses, la carte à puce devait être aussi populaire au Canada durant les années 1990. Certains spécialistes sont même allés jusqu'à comparer son avènement à l'invasion des micro-ordinateurs. Toutefois, avant 1994, elle n'a suscité que très peu d'intérêt en Amérique du Nord. Par contre, avec le projet de transfert du dossier médical du patient sur une carte à puce et les nouvelles cartes d'appel intelligentes de Bell qui permettent de faire des appels en débitant directement la carte, ces dernières commencent à devenir de plus en plus populaires.

III.3.1.2 Définition

La carte à puce "Smart card" est une carte plastifiée aux dimensions de $85,6 \times 54$ millimètres avec une épaisseur de l'ordre de 70 à 90 centièmes de millimètres. Elle comporte au moins un circuit intégré (la puce). Jusqu'aujourd'hui, il y a encore quelques ambiguës dans la définition et le terme de cette carte.

Outre le nom la carte à puce, elle reconnaît sous le nom ICC (Integrated Circuit Card) qui est utilisé par ISO [1].

III.4 NORMALISATION PARFAITE

III.4.1 Normes

Pour définir une carte à puce, il faut au moins normaliser trois types de paramètres différents:

- **Paramètres physiques** : indiquent la taille de la carte et la position de la puce et de ses contacts.
- **Paramètres électriques** : précisent les tensions d'alimentation et niveaux électriques mis en œuvre ainsi que le brochage de la puce sur la carte.
- **Paramètres logiciels** : définissent le mode de dialogue avec la carte, les commandes qu'elle peut interpréter et son comportement face à ces dernières.

Cela s'est traduit par un certain nombre de normes internationales comme montre dans le Tableau III.1 donne une liste aussi complète que possible dans le cas des cartes en général.

Norme ISO	Titre officiel et sa traduction en français
ISO 7810	IdentificationCards_PhysicalCharacteristics Cartes d'identification – Caractéristiques physiques
ISO 7811-1	IdentificationCards-RecordingTechniqueEmbossing Cartes d'identification – Techniques d'embossage
ISO 7811-2	IdentificationCards-RecordingTechniqueMagneticStripeCartes d'identification – Techniques d'enregistrement magnétique
ISO 7811-3	Identification Cards – Recording Technique Location of Embossed Characters on ID 1CardsCartes d'identification – Emplacement des caractères embossés sur les cartes de type ID 1
ISO 7811-4	Identification Cards – Recording Technique Location of Read-Only Magnetic Tracks –Tracks1and2Cartes d'identification – Position des pistes magnétiques à lecture seule – Pistes 1 et 2
ISO 7811-5	Identification Cards –Recording Technique Location of Read-Xrite Magnetic Tracks-Track3Cartes d'identification – Position des pistes magnétiques à lecture/écriture – Piste 3
ISO 7812-1	Identification Cards – Identification of Issuers Part 1 : Numbering System Cartes d'identification – Identification de l'émetteur, partie 1 : système de numérotation
ISO 7813	IdentificationCards-FinancialTransactionCardsCartes d'identification – Cartes pour transactions financières
ISO 7186-1	IdentificationCardsIntegratedCircuitsCardswithContactsPhysicalCharacteristicsCartes d'identification – Cartes à circuits intégrés avec contacts – Caractéristiques physiques
ISO 7816-2	Identification Cards – Integrated Circuits Cards with Contacts – Dimension and LocationoftheContactsCartes d'identification-Cartes à circuits intégrés avec contacts-Dimension et position des contacts
ISO 7816-3	Identification Cards – Integrated Circuits Cards with Contacts – Electronic Signals andTransmissionProtocolsCartes d'identification – Cartes à circuits intégrés avec contacts – Signaux électroniques et protocoles de transmission
ISO 7816-3 Amendment 1	Protocol type T = 1, Asynchronous Half Duplex Block Transmission Protocol Protocole T = 1, protocoles asynchrone semi-duplex à transmission par blocs
ISO 7816-3 Amendment 2	RevisionofProtocolTypeSelectionRévision du mode de sélection de protocole
ISO 7816-4	Identification Cards-Integrated Circuits Cards with Contacts-Interindustry CommandsforInterchangeCartes d'identification- Cartes à circuits intégrés avec

	contacts – Commandes inter-industries
ISO 7816-5	Identification Cards – Integrated Circuits Cards with Contacts – Number System and Registration Procedure for Application Identifier Cartes d'identification – Cartes à circuits intégrés avec contacts – Système de numération et procédure d'enregistrement pour l'identification des applications
ISO 1177	Information Processing – Character Structure for Start/Stop and Synchronous Character Oriented Transmission Traitement de l'information – Structure des caractères pour les échanges synchrones orientés caractères

Tableau (III.1) : Liste des principales normes relatives aux cartes s'appliquant en totalité ou partie aux cartes à puce

III.4.2 Normes principales

En ce qui concerne la carte à puce, quatre normes principales sont à extraire de ce tableau:

- la norme ISO 7816-1 précisant les caractéristiques physiques de la carte.
- la norme ISO 7816-2 définissant la position et le brochage des contacts de la carte à puce.
- la norme ISO 7816-3 définissant les niveaux électriques et les chronogrammes de bas niveau qui régissent le dialogue avec les cartes à puce.
- la norme ISO 7816-4 définissant les différentes commandes de base des cartes à puce.

La plus intéressante de ces normes est évidemment l'ISO 7816-4 qui permet de comprendre le dialogue entre une carte à puce et son lecteur [2].

III.4.2.1 La norme ISO 7816-1

Cette norme définit comme nous l'avons déjà précisé les caractéristiques physiques des cartes **Fig.(II.1)** ayant une interface physique par des contacts électriques. Elle ne définit pas la nature, le nombre et la position des circuits intégrés dans la carte. Elle fait référence aux normes relatives aux cartes d'identification ISO 7810, ISO 7811 parties 1 à 5, ISO 7812 et ISO 7813.

Les caractéristiques introduites par la norme portent sur :

- la protection contre les ultraviolets et les rayons X
- le profil de surface des contacts
- la résistance mécanique des cartes et des contacts (au pliage, à la torsion)
- la résistance électrique des contacts
- l'interférence électromagnétique entre les pistes magnétiques éventuelles de la carte et de ses circuits intégrés
- l'exposition de la carte à des champs magnétiques
- l'exposition de la carte à une décharge d'électricité statique

- la dissipation thermique du circuit intégré

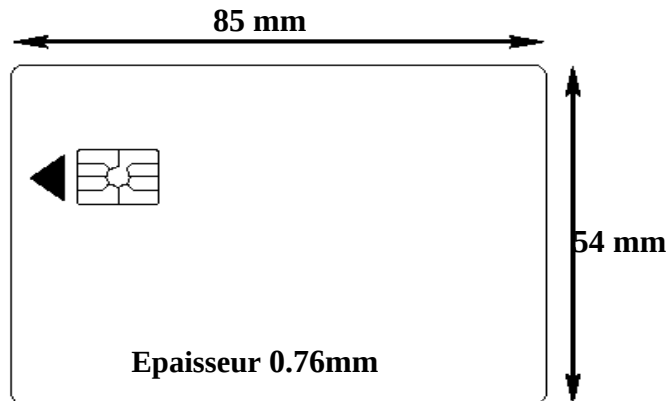


Figure (III.2) : Dimension de la carte à puce

III.4.2.2 la norme ISO 7816-2

La norme ISO 7816-2 définit les aspects électriques et la situation des contacts sur la carte. La carte comporte huit contacts (Fig. 2) dont voici l'utilisation :

- **Le contact 1** correspond à la tension d'alimentation en lecture (V_{cc}).
- **Le contact 2** correspond au signal Reset (RST) : une tension appliquée sur ce contact déclenche l'initialisation physique et logique du composant.
- **Le contact 3** correspond au signal d'horloge (CLK). Il fournit le signal d'horloge externe dont dérivera l'horloge interne.
- **Le contact 4:** est réservé à une utilisation future (RFU : Reserved for Futur Use).
- **Le contact 5** correspond à la masse (GND).
- **Le contact 6** correspond à la tension d'alimentation (V_{pp}) à appliquer, sur demande de la carte, pour programmer la mémoire de données (tension en écriture). il est seulement utilisé sur des cartes anciennes.
- **Le contact 7** est le contact d'entrée/sortie (I/O) par lequel transitent en half-duplex toutes les données échangées entre la carte et le monde extérieur.
- **Le contact 8** est réservé à une utilisation future (RFU).

Une norme utilisant les contacts RFU serait en cours de préparation pour l'utilisation de l'USB avec la carte.

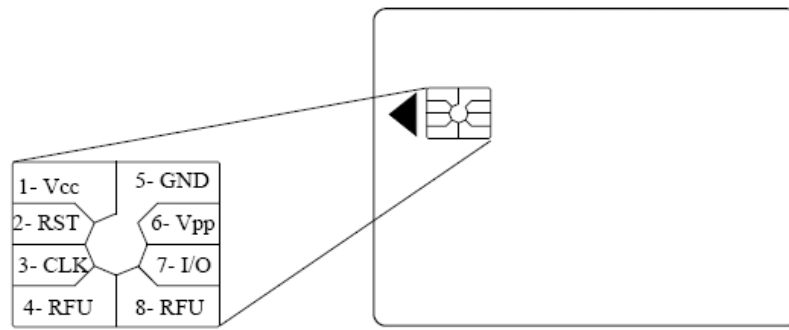


Figure (III.3) : Les contacts de la carte à puce

III.4.2.3 La norme ISO 7816-3

Cette partie normalise :

- les caractéristiques électriques comme la fréquence d'horloge (entre 1 MHz et 5 MHz), la vitesse des communications (jusqu'à 115200 bauds), etc...
- les protocoles de transmission (TPDU : Transmission Protocole Data Unit) :
 - T=0, protocole orienté octet.
 - T=1, protocole orienté paquet.
 - T=14, réservé pour les protocoles propriétaires.
- la sélection du type de protocole (PTS : Protocol Type Selection) si plusieurs protocoles sont supportés.
- la réponse au reset (ATR : Answer To Reset) qui correspond aux données envoyées par la carte immédiatement après la mise sous tension.

III.4.2.4 La norme ISO 7816-4

Visé à assurer une inter-opérabilité. Il spécifie :

- le contenu des messages entre la carte et le lecteur (CAD : Card Acceptance Device) qui utilise le protocole APDU (**A**pplication **P**rotocol **D**ata **U**nits) pour les commandes et les réponses.
- les structures des fichiers et des données :
 - l'accès à ces données.
 - l'architecture de sécurité.
 - la sécurisation des communications.

a. Le protocole APDU

Ce protocole est de niveau application comme son nom l'indique. Il existe deux types de messages APDUs :

- la commande APDU (C-APDU) qui est émise par le CAD en direction de la carte.

- la réponse APDU (R-APDU) qui elle transite de la carte au CAD.

Le modèle de discussion entre le CAD et la carte est un modèle maître-esclave où le CAD est le maître et la carte est l'esclave. La carte est toujours en attente d'une commande APDU. Une réponse APDU aura toujours lieu en retour d'une commande APDU. Les deux types de structures de communication commande et réponse sont toujours couplés.

a.1. La commande et reponse APDU est constituée par

L'instruction APDU					
CLA	INS	P1	P2	Len	Champ du données

- **CLA** (1 octet): Classe d'instruction: indique la structure et formate pour une catégorie d'instruction et réponse APDU.
- **INS** (1 octet): code de l'Instruction: spécifie l'instruction.
- **P1** (1 octet) et **P2** (1 octet): paramètres de l'Instruction--fournissez des qualifications à l'instruction plus loin.
- **Len** (1 octet): Nombre des octets qui présentent dans le champ du données de l'instruction.
- **Champ du données** (les octets égalent à la valeur de Len): Une séquence des octets dans le champ du données de l'instruction.

La response APDU			
INS	Champ du données	SW1	SW2

- **Champ du données**(longueur variable):Une séquence des octets a reçu dans le champ du données de la réponse.
- **SW1** (1 octet) et **SW2** (1 octet): mots de la situation--dénotez l'état du traitement dans la carte.

a.2. Le protocole d'échange d'information

Voici comment se déroule un échange standard d'informations entre le maître et la carte ISO 7816.

- A la mise en service, le maître génère un RESET sur la pin MCLR, la carte répond avec un certain nombres d'octets. Cette réponse s'appelle ATR, pour **réponse au reset**(Answer To Reset).

- Le maître envoie la commande Class INS P1 P2 LEN
- La carte renvoie l'instruction comme accusé de réception INS
- Le maître envoie éventuellement les données complémentaires
- La carte envoie éventuellement la réponse
- La carte envoie le status et repasse en mode d'attente de commande

b. Le système de fichiers des cartes à puce

La norme ISO 7816-4 supporte un système de fichiers hiérarchique qui peut contenir trois types de fichiers :

- **Master File (MF)** : Ce fichier est la racine du système de fichiers. Il peut contenir des Dedicated Files et Elementary Files. Il est unique sur une carte à puce.
- **Dedicated File (DF)**: Ce fichier est en fait un répertoire sur la carte à puce et il peut contenir d'autres Dedicated Files et des Elementary Files. Le MF est un type spécial de DF.
- **Elementary File (EF)** : Ce fichier contient les données et ne peut pas contenir d'autres fichiers. Il en existe quatre types :
 - **transparent file**: est structuré comme des séquences d'octets de données alors que les trois autres types de fichiers sont structurés comme des séquences d'enregistrement individuellement identifiable,
 - **linear fixed**: contient des enregistrements de taille fixée,
 - **linear variable**: contient des enregistrements de taille variable,
 - **cyclic fixed**: contient des enregistrements organisés comme un anneau.

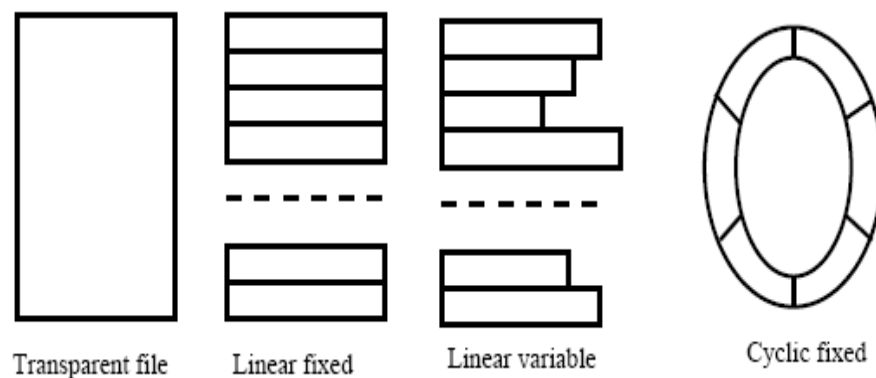


Fig.(III.4): Des types des fichiers.

Un exemple de structure de système fichier:

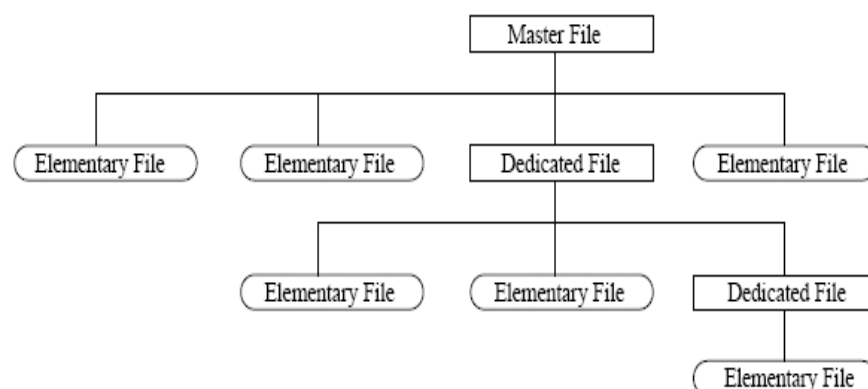


Fig.(III.5): Structure de système fichier [3]

III.5 DIFFERENTS TYPES DE CARTES

Il y a deux principales familles qui cohabitent sur le marché, car leurs modes d'utilisation et leurs possibilités diffèrent de manière très importante; même si elles respectent toutes deux les différents paramètres normalisés que nous venons d'étudier.

III.5.1 carte à mémoire ou carte synchrone

C'est une carte qui sont le plus communes et le moins chères. Elle contient un EEPROM qui stocke de donnée où toutes les données de l'application sont enregistrées. Sa capacité varie de 2ko à 8ko, et un ROM qui stocke des données qui ne changent jamais endant la vie de la carte. Elle tient le numéro, le nom de teneur... de la carte.

La première famille, qui est aussi historiquement la plus ancienne, est la carte à mémoire appelée aussi carte synchrone en raison de son protocole de dialogue. Bien que ce soit une carte à puce, puisqu'une mémoire est bel et bien une puce de circuit intégré ; cette famille de carte est cantonnée à des applications relativement simples. Elle ne contient en effet que de la mémoire. En fait, cette famille peut aujourd'hui être scindée à son tour en deux sous-familles avec les cartes à mémoire "simples" appelées cartes à mémoire "tout court" et les cartes dites à mémoire protégée.

III.5.1.1 Les cartes à mémoire (sous-entendu simples)

Comme le montre le synoptique de "**Fig.(II.5)**", ne contiennent qu'une zone mémoire et le minimum de logique nécessaire pour pouvoir y accéder via les signaux définis à la figure (II.2) "Les contacts de la carte à puce".

Ce sont des cartes de ce type qui étaient utilisées pour réaliser les cartes téléphoniques françaises de première génération appelées aussi T1G. Le niveau de sécurité offert par ces cartes est très faible et ne peut reposer que sur des artifices relativement simples qui ne résistent pas plus de quelques minutes en face d'une personne compétente. France Télécom en a d'ailleurs fait les frais dans ses cabines publiques pendant des années.

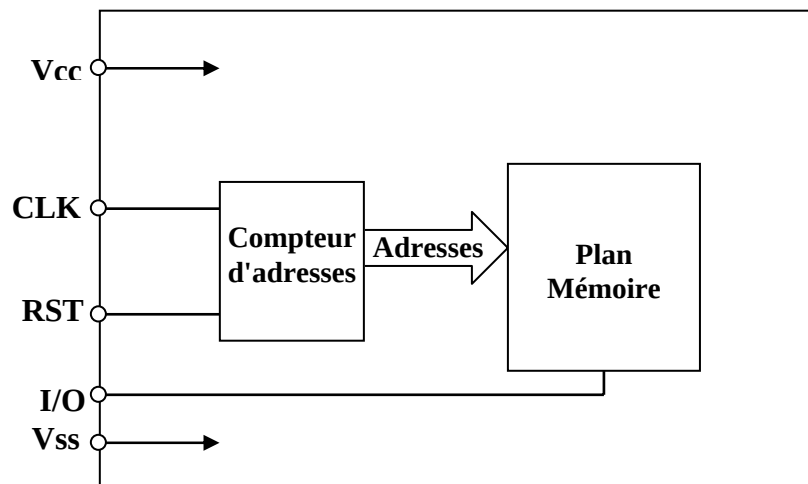


Fig.(III.6):synoptique interne ³⁸général d'une carte à mémoire

III.5.1.2 Les cartes à mémoire protégée

Font un pas en avant en matière de sécurité en associant de la mémoire, dont certaines zones sont accessibles seulement en lecture ou seulement en phase de personnalisation de la carte, et de la logique permettant l'exécution d'automates simples allant jusqu'à la présentation de mots de passe ou autres succédanés de codes PIN.

Ces cartes sont plus sûres que les cartes précédentes mais elles ne permettent pas la mise en place des applications les plus complexes que sont les cartes bancaires, les cartes SIM ou bien encore les cartes de décryptage TV. Il faut en effet pour cela que la carte dispose d'une "intelligence" locale que n'ont pas les cartes à mémoire.

Les cartes à mémoire ne présentant plus aujourd'hui un grand intérêt.

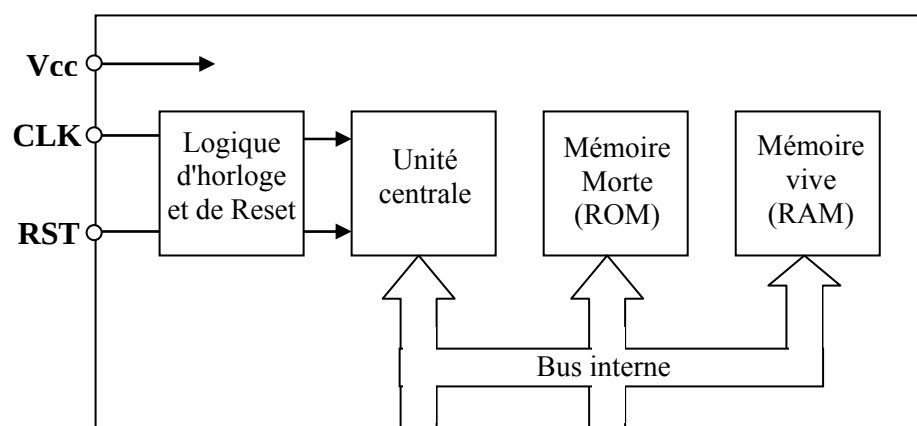
5.2. Les cartes à microcontrôleur ou cartes asynchrones

L'intelligence locale qui fait défaut aux cartes à mémoire existe seulement dans les cartes à puce à microcontrôleur ou smart cards dans la langue de Shakespeare, appelées aussi cartes asynchrones en raison de leur protocole de dialogue.

Les cartes à puce "intelligentes" renferment un microcontrôleur complet ; c'est à dire l'association en un seul circuit d'une unité centrale de microprocesseur CPU qui est le coeur de carte à puce, de mémoire morte ROM qui tient le système d'exploitation de la carte, de mémoire vive RAM qui est utilisée par le processeur pour exécuter des fonctions désirées, de mémoire EEPROM qui tient des programmes et des données de l'application de carte, d'une interface d'entrée/sortie série et de toute la logique nécessaire pour faire fonctionner tout cela comme le montre le synoptique de "la figure 3.2".

C'est donc bien un véritable petit micro-ordinateur complet qui est contenu dans ces cartes; micro-ordinateur dénué de clavier et d'afficheur et dont le seul moyen de communication avec le monde extérieur est sa ligne d'entrée/sortie unique et bidirectionnelle I/O.

L'unité centrale utilisée varie selon la fonction ou la vocation de la carte. On peut trouver de simples microcontrôleurs 8 bits des familles PIC de Microchip ou AVR d'Atmel mais aussi des processeurs beaucoup plus puissants, associés parfois à un cryptoprocèsseur comme dans certaines carte de décryptage télévision récentes.



5.3. Les cartes à puce "vierges" ou "spécifiques"

dont la mémoire de programme du microcontrôleur ne contient rien lorsque vous les achetez. C'est donc à vous d'écrire l'intégralité de leur OS (operating system). Cela demande beaucoup de travail mais permet de disposer de cartes réellement "sur mesure". Les cartes Gold, Silver, Fun ou bien encore Jupiter font partie de cette famille.

5.3.1. La carte Gold ou carte Wafer 1

cette carte est historiquement la première à avoir été réalisée si l'on excepte les Basic Card qui constituent une famille à part. Elle est de ce fait plus simple que certaines cartes actuelles, compte tenu de son âge et donc des composants qui étaient disponibles à l'époque de sa conception. Comme le montre son schéma de " **Fig.(II.6)**", elle ne contient en effet qu'un classique microcontrôleur PIC 16F84 associé à une mémoire EEPROM série externe de type 24LC16. Cette approche, qui a de quoi surprendre puisque le 16F84 contient déjà en interne de la mémoire EEPROM de données, est justifiée par le fait que la taille de cette mémoire est très réduite puisqu'elle n'est que de 64 octets. L'ajout de la 24LC16 externe permet de porter cette capacité de mémorisation à 2 K octets ce qui est déjà plus confortable. Cette carte est appelée à la quasi-unanimité carte Wafer 1 ou bien encore carte Gold en raison de la couleur dorée de certaines versions.

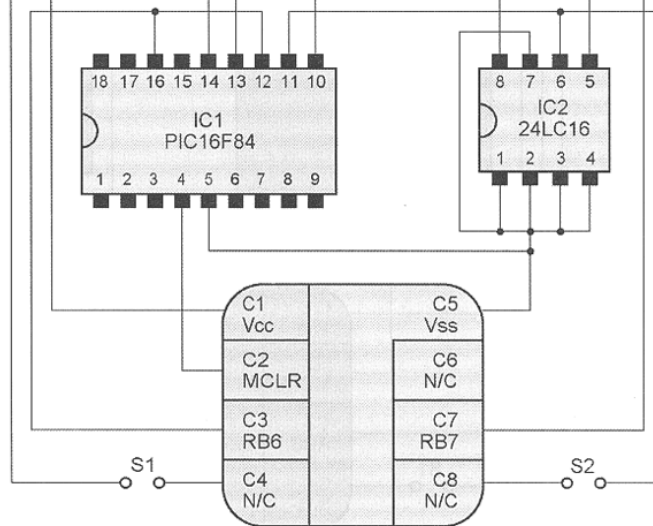


Fig.(III.8):Schéma interne d'une carte à puce Wafer 1 ou Gold Card

La carte Gold existe aussi sous la dénomination Wafer 2 ou bien encore Gold 64. Son schéma est alors identique à celui de la Wafer 1, comme le montre de " **Fig.(II.7)**", mais la mémoire EEPROM devient de la 24LC64 c'est à dire qu'elle offre une capacité de 8 K octets.

Remarquons, sur ces deux schémas, la possibilité d'accéder aux lignes SDA et SCL de la mémoire EEPROM via les contacts C4 et C8, normalement non utilisés du connecteur de carte. Cette possibilité optionnelle, d'où la présence des straps S1 et S2 permet de programmer directement le contenu de l'EEPROM sans passer "au travers" du

microcontrôleur. Il faut bien sûr utiliser pour cela un lecteur de carte modifié mais il en existe sur le marché. Ces deux liaisons n'existent que sur les cartes Gold que l'on réalise soi-même.

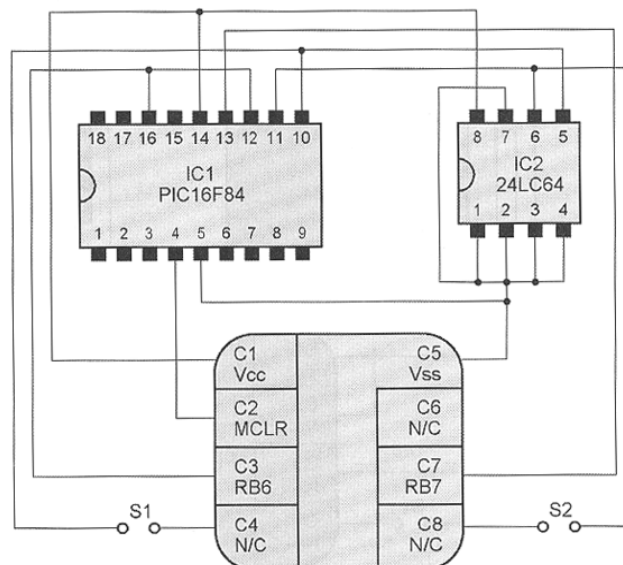
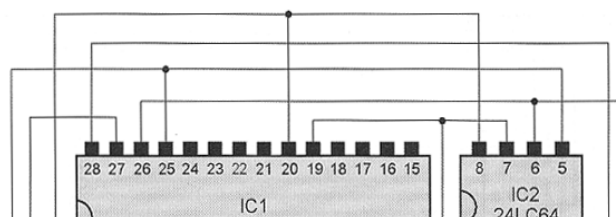


Fig.(III.9) : Schéma interne d'une carte à puce Wafer 2 ou Gold Card

5.3.2. La carte Silver ou carte Wafer 3

Appelée carte Wafer 3, car elle est arrivée après les deux précédentes, ou bien encore carte Silver car de nombreuses versions de cette carte sont disponibles sous une belle livrée argentée ; cette carte reste fidèle à la famille PIC de Microchip mais fait appel à un circuit plus richement doté en ressources internes avec le 16F876, comme le montre son schéma visible de "**Fig(II.7)**"



Comme nous pouvons le constater, le schéma utilisé est identique dans son principe à celui des cartes Gold mais utilise d'office une mémoire EEPROM de 8 Koctets.

Notez cependant que le 16F876 dispose d'une mémoire de programme de 8 K octets contre seulement 1 K octets dans le "petit" 16F84 de la carte Gold. Il propose en outre une RAM interne de 368 octets contre 68 pour le 16F84 et une EEPROM interne de 256 octets contre 64 toujours pour le 16F84.

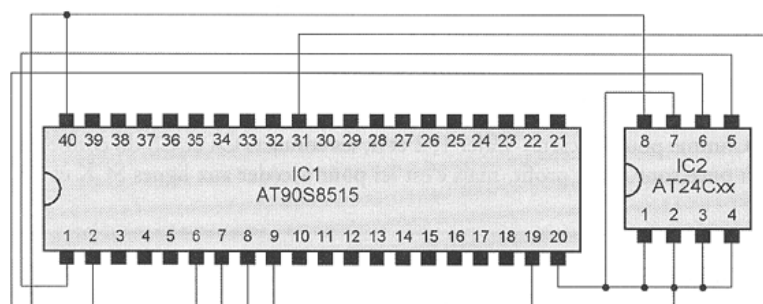
Comme pour la carte Gold, la possibilité d'accéder aux lignes SDA et SCL de la mémoire EEPROM via les contacts C4 et C8 est optionnelle, d'où la présence des straps S1 et S2. Elle permet de programmer directement le contenu de l'EEPROM sans passer "au travers" du microcontrôleur. Il faut bien sûr utiliser pour cela un lecteur de carte modifié mais il en existe sur le marché. Ces deux liaisons n'existent pas dans les vraies cartes Silver.

5.3.3. La carte Fun ou Purple ou encore Wafer 4

Appelée carte Purple (pourpre en anglais), en raison de la couleur des premières versions qui ont été commercialisées; cette carte existe aussi sous le nom de Fun Card.

Comme le montre son schéma de "Fig(II.8)", on abandonne avec cette carte le monde des microcontrôleurs PIC de Microchip pour entrer dans celui tout aussi passionnant des microcontrôleurs AVR de chez Atmel.

nous rappelons que ces circuits sont également des circuits à architecture RISC et qu'ils disposent tous de mémoire de programme de type EEPROM, c'est à dire programmable et effaçable électriquement.



Sur le schéma de la Fun Card on trouve un microcontrôleur Atmel AVR de type AT90S8515 qui est un circuit situé dans la partie haute de la famille AVR. Il renferme 8 K octets de mémoire de programme, 512 octets de RAM interne et de très nombreuses ressources diverses. Comme pour les cartes précédentes, il est associé à de la mémoire EEPROM externe dont la taille varie selon le type de Fun Card choisie.

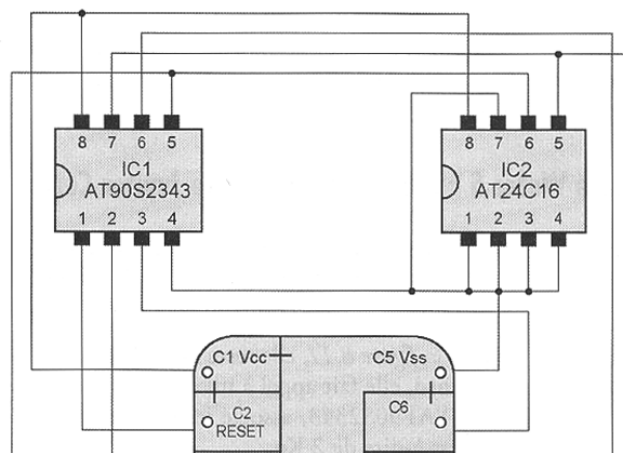
A l'heure actuelle on trouve sur le marché les versions suivantes :

- Wafer 4 avec de la mémoire AT24C64 soit 8 K octets.
- Wafer 4A avec de la mémoire AT24C128 soit 16 K octets.
- Wafer 4B avec de la mémoire AT24C256 soit 32 K octets.

Comme pour les cartes Gold et Silver, les contacts C4 et C8 du connecteur carte à puce sont mis à profit, mais c'est ici pour accéder aux lignes SCK et MOSI du microcontrôleur en phase de programmation de ce dernier.

5.3.4. La carte Jupiter ou Pink ou encore Wafer 5

La couleur rose de certaines versions lui a valu son appellation de Pink Card puisque c'est comme cela que l'on dit rose dans la langue de Shakespeare, mais cette carte s'appelle aussi Wafer 5 ou bien encore Jupiter Card. Comme le montre son schéma de " **Fig(II.9)**", elle reste fidèle à Atmel puisqu'elle utilise encore un microcontrôleur de la famille AVR.



Mais contrairement à la Fun Card, elle fait appel à un des plus petits microcontrôleurs de la famille AVR avec l'AT90S2343, associé ici encore à de l'EEPROM externe de type AT24C16 c'est à dire de 2 K octets.

Elle est donc à réserver aux applications assez peu gourmandes en ressources mémoires, encore que le petit AT90S2343 contienne tout de même 2 K octets de mémoire de programme, soit deux fois plus que le 16F84 des cartes Gold.

Comme pour les cartes Gold et Silver, les contacts C4 et C8 du connecteur carte à puce sont mis à profit, mais c'est ici pour accéder aux lignes SCK et MOSI du microcontrôleur en phase de programmation de ce dernier. [5]

Lorsque l'inclusion d'un lecteur de carte à puce, il est l'envoi d'un code PIN est situé à l'intérieur du

PIC16F877A

6. Les PIC

6.1. Structure d'un PIC

Les PIC, au même titre que les microprocesseurs, sont composés essentiellement de registres ayant chacun une fonction bien définie. Les PIC possèdent également des périphériques intégrés, tels qu'une mémoire EEPROM, un TIMER, des ports d'entrées/ sorties ou bien encore un convertisseur analogique/numérique.

La structure minimale d'un PIC est constituée des éléments ci-dessous :

- Une mémoire de programme contient le code binaire correspondant aux instructions que doit exécuter le microcontrôleur. La capacité de cette mémoire est variable selon les PIC
- Une mémoire RAM sauvegarde temporairement des données. sa capacité est aussi variable selon les PIC.

- Une Unité Arithmétique et Logique (UAL ou ALU en anglais) est chargée d'effectuer toutes les opérations arithmétiques de base (addition, soustraction, etc.) ainsi que les opérations logiques de base (ET, OU logique, etc.).
- Des ports d'entrées/sorties permettent de dialoguer avec l'extérieur du microcontrôleur, par exemple pour prendre en compte l'état d'un interrupteur (entrée logique), ou encore pour commander un relais (sortie logique). Le PIC 16F877A possède 33 lignes d'entrées/sorties (PORTA, PORTB, PORTC, PORTD).
- Un registre compteur de programme (CP ou PC en anglais), est chargé de pointer l'adresse mémoire courante contenant l'instruction à réaliser par le microcontrôleur. Le contenu du registre PC évolue selon le pas de programme.
- Un registre pointeur de pile (PP ou SP en anglais) est essentiellement utilisé lorsque l'on réalise un sous-programme. Le pointeur de pile est chargé de mémoriser l'adresse courante que contient le compteur de programme avant le saut à l'adresse du sous-programme. Lorsque le sous-programme est terminé, le pointeur restitue l'adresse sauvegardée vers le compteur de programme.
- Un registre d'instruction contient tous les codes binaires correspondant aux instructions à réaliser par le microcontrôleur. Le PIC 16F877A comporte 35 instructions.
- Un registre d'état est en relation avec l'UAL et permet de tester le résultat de la dernière opération effectuée par le microcontrôleur. Selon la dernière opération effectuée, des bits sont positionnés dans le registre d'état et ceux-ci peuvent être testés à l'aide d'une instruction de branchement pour effectuer des sauts conditionnels.
- Une horloge système permet de cadencer tous les échanges internes ou externes au microcontrôleur.

La famille des PICs est subdivisée en 3 grandes familles : La famille Base-Line, qui utilise des mots d'instructions de 12 bits, la famille Mid-Range, qui utilise des mots de 14 bits (et dont font partie les 16F84 et 16F876), et la famille High-End, qui utilise des mots de 16 bits.

6.2. Identification d'un PIC

Pour identifier un PIC, nous utilisons simplement son numéro. Les 2 premiers chiffres indiquent la catégorie du PIC, 16 indique un PIC Mid-Range, Vient ensuite parfois une lettre (**L**) Celle-ci indique que le PIC peut fonctionner avec une plage de tension beaucoup plus tolérante. Ensuite, nous trouvons (**C**) indique que la mémoire programme est une EPROM ou

plus rarement une EEPROM, **(CR)** pour indiquer une mémoire de type ROM Ou **(F)** pour indiquer une mémoire de type FLASH.

À ce niveau, on rappelle que seule une mémoire FLASH ou EEPROM est susceptible d'être effacée.

Finalement nous trouvons sur les boîtiers le suffixe **(-XX)** dans lequel XX représente la fréquence d'horloge maximale que le PIC peut recevoir. Par exemple **(-04)** pour un 4MHz.

6.3. Le microcontrôleur PIC 16F877A

Nous allons maintenant s'intéresser à la structure interne du PIC 16F877A, avec lequel nous avons travaillé.

Le 16F877A est un microcontrôleur de MICROCHIP, fait partie intégrante de la famille des Mid Range (16) dont la mémoire programme est de type flash (F) de type 877A et

FLASH	RAM	EEPROM	I/O	A/D	interruptions	TIMER	communication
8K	368	256	33	8	14	3	USART/ MSSP/PSP

capable d'accepter une fréquence d'horloge maximale de 20Mhz.

6.4. Brochage du 16F877A

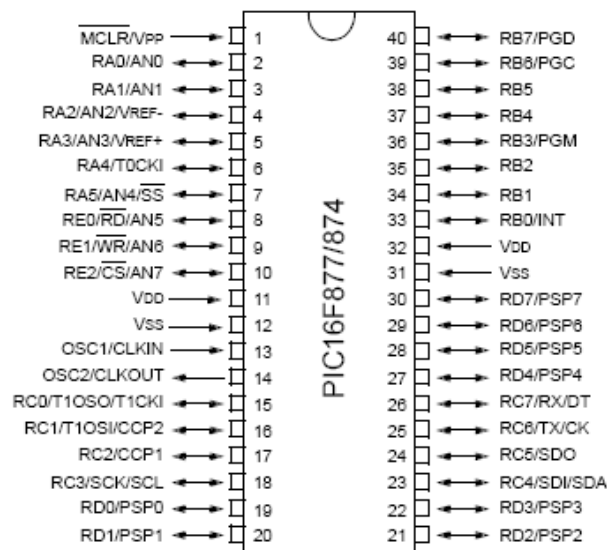


Fig .(III.13) : Brochage du PIC16F877A

6.5. Les particularités électriques

On constate que sur le schéma concernant le 16F877A, on a 2 connexions « VSS » qui sont reliées à la masse. En fait, en interne, ces pins sont interconnectés. La présence de ces 2 pins s'explique pour une raison de dissipation thermique. Les courants véhiculés dans le Pic sont loin d'être négligeables du fait des nombreuses entrées/sorties disponibles.

Le constructeur a donc décidé de répartir les courants en plaçant 2 pins pour l'alimentation VSS, bien évidemment, pour les mêmes raisons (dissipation thermique), ces pins sont situées de part et d'autre du PIC, et en positions relativement centrales. [8]

6.6. Organisation du 16F877A

La mémoire du 16F877A est divisée en 3 parties Pour mieux le connaître regardons de près son Schéma fonctionnel.

- **La mémoire programme :** La mémoire programme est constituée de 8K mots de 14 bits. C'est dans cette zone que nous allons écrire notre programme.
- **La mémoire EEPROM :** La mémoire EEPROM (Electrical Erasable Programmable Read Only Memory), est constituée de 256 octets que nous pouvons lire et écrire depuis notre programme. Ces octets sont conservés après une coupure de courant et sont très utiles pour conserver des paramètres semi-permanents.
- **La mémoire Ram et organisation :** La mémoire RAM est celle que nous allons sans cesse utiliser. Toutes les données qui y sont stockées sont perdues lors d'une coupure de courant.

La mémoire RAM disponible du 16F876 est de 368 octets. Elle est répartie de la manière suivante:

- 80 octets en banque 0, adresses 0x20 à 0x6F
- 80 octets en banque 1, adresses 0xA0 à 0XEF
- 96 octets en banque 2, adresses 0x110 à 0x16F
- 96 octets en banque 3, adresses 0x190 à 0x1EF
- 16 octets communs aux 4 banques, soit 0x70 à 0x7F = 0xF0 à 0xFF
0x17F=0x1F0 à 0x1FF.

6.7. Les ports entrée/sortie

On dispose de 33 broches d'entrées/sorties, chacune configurables soit en entrée soit en sortie (PORTA, PORTB, PORTC, PORTD, PORTE).

Un registre interne au PIC, nommé TRIS, permet de définir le sens de chaque broche d'un port d'entrées/sorties. En règle générale, un bit positionné à « 0 » dans le registre TRIS

donnera une configuration en sortie pour la broche concernée ; si ce bit est positionné à « 1 », ce sera une broche d'entrée.

- **Le port A :** Le 16F877A dispose de 5 canaux d'entrée analogique. Nous pouvons donc échantillonner successivement jusqu'à 5 signaux différents avec ce composant. Les pins utilisés sont les pins AN0 à AN4 (qui sont en fait les dénominations analogiques des pins RA0 à RA3 + RA5).
- **Le port B :** Hors de sa fonction principale autant que ports d'entrées /sorties, on note la pin RB0 qui, en configuration d'entrée, est de type « trigger de Schmitt » quand elle est utilisée en mode interruption « INT » ; La lecture simple de RB0 se fait, elle, de façon tout à fait classique, en entrée de type TTL. Encore il y a (RB3-RB6-RB7) qui peuvent servir dans la programmation (en mode LVP) en cas d'absence de programmeur commercial.
- **Le port C :** C'est un port tout ce qu'il y a de plus classique, Or qu'il a deux pins qu'on utilisera plus tard dans la communication série avec le PC à travers (TX et RX) (pin17 et 18).
- **Le port D :** Ce port fonctionne de façon identique aux autres, dans son mode de fonctionnement général. Le registre TRISD comportera donc les 8bits de direction, pendant que le registre port D correspond aux pins I/O concernés. D'où les 8 pins I/O, en mode entrée, sont du type <trigger de Schmitt>.
- **Le port E :** Ce port n'est présent que sur les PIC 16F877. Il ne comporte que 3 pins RE0à RE2, mais, contrairement aux ports, les bits non concernés de TRISE sont, cette fois, implantés pour d'autres fonctions. Les pins REX peuvent également être utilisés comme pins d'entrées analogiques. D'où le registre ADCON1 qui détermine si ce port est utilisé comme port I/O ou comme port analogique.

6.8. Les interruptions

Le PIC16F877A dispose de plusieurs sources d'interruptions.

- Une interruption externe, action sur la broche INT/RB0.
- Débordement du TIMER0.
- Changement d'état logique sur une des broches du PORTB (RB4 à RB7).
- Une interruption d'un des périphériques (PEIE).
 - Fin de programmation d'une case mémoire de l'EEPROM.
 - Changement d'état sur le PORTD (PSPIE).
 - Fin de conversion analogique numérique (ADIE).

- Réception d'une information sur la liaison série (RCIE).
- Fin d'émission d'une information sur la liaison série (TXIE).
- Interruption SPI ou I2C du module MSSP (SSPIE).
- Interruption du registre de capture et/ou de comparaison 1 (CCPI1E).
- Interruption du registre de capture et/ou de comparaison 2 (CCPI2E).
- Débordement du TIMER1 (TMR1E).
- Débordement du TIMER2 (TMR2E).
- Collision de BUS (BCLIE)

6.9. Le TIMER

Un Timer est un registre interne au micro contrôleur, celui-ci s'incrémente au grès d'un horloge, ce registre peut servir par exemple pour réaliser des temporisations, ou bien encore pour faire du comptage (par l'intermédiaire d'une broche spécifique : RA4/TOKI). Le PIC 16F877A possède trois timers sur 8 bits (il compte jusqu'à 256) configurable par logiciel.

Le TIMER 2.

C'est un TIMER de 8 bits, son horloge ne peut être que l'horloge interne divisée par 4 ($F_{osc}/4$)

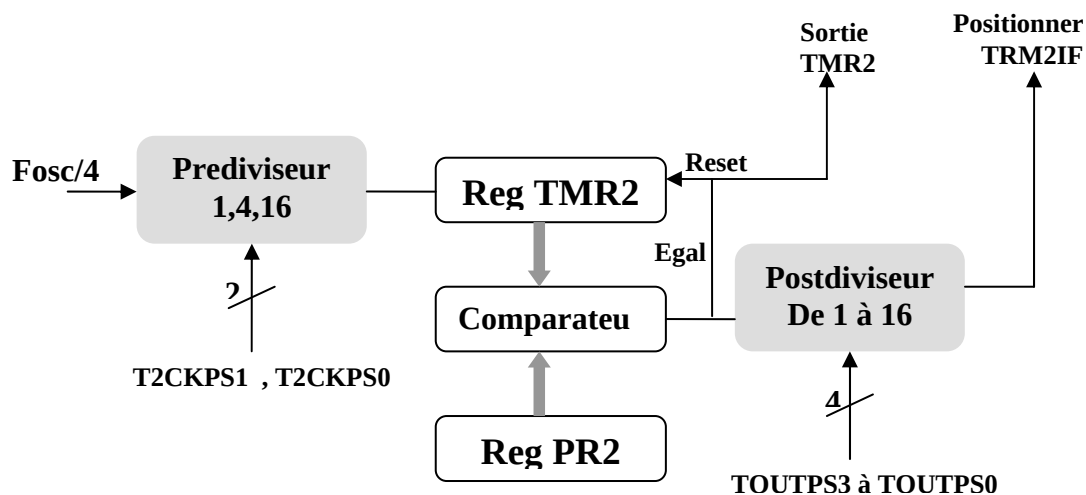


Figure (III.14) : Principe de fonctionnement du Timer2.

Fonctionnement :

Il est incrémenté par l'horloge interne ($F_{osc}/4$) pré divisée ou non. Les bits T2CKPS1 et T2KPS0 permettent de choisir la valeur de la pré division (1,4 ou 16).

Le contenu du registre incrémenté TMR2 et il est comparé au registre PR2, si ces deux registres sont égaux alors une impulsion d'horloge est générée et le contenu de TMR2 est

remis à 00h. Celle-ci peut servir d'horloge pour piloter les liaisons I2C et SPI du module SSP ou encore être divisée par un post diviseur appelé : POSTSCALER.

Son rapport de division peut être de 1 à 16. Les bits T2OUTPS0, T2OUTPS1, T2OUTPS2, et T2OUTPS3, permettent de choisir la valeur de la post division 1, 2, 3, 4, 5 16, Quand la sortie du post diviseur passe à 1 le bit TMR2IF est positionné, celui-ci peut déclencher une interruption si celle-ci a été autorisée (Bit TMR2IE à 1 du registre PIE1). Lorsque le bit TMR2IF passe à 1 lorsqu'une impulsion est générée en sortie du POSTSCALER, il doit être remis à 0 de façon logicielle.

Le principe du timer 2 est différent, dans le sens que l'événement détecté n'est pas le débordement « ordinaire » du timer (c'est-à-dire le passage de 0xFF à 0x00), mais le débordement par rapport à une valeur prédéfinie, Cette valeur étant mémorisée dans le registre PR2 (banque 1).

On peut donc dire que le fonctionnement du timer est le suivant :

- On incrémente le contenu du prédiviseur à chaque cycle d'instruction
- Chaque fois que ce contenu correspond à un multiple de la valeur du prédiviseur, on incrémente TMR2 (contenu du timer 2)
- Chaque fois que le contenu de TMR2 dépasse le contenu de PR2, on remet TMR2 à 0, et on incrémente le contenu du postdiviseur.
- Chaque fois que le contenu du postdiviseur correspond à un multiple de la valeur du postdiviseur, on positionne le flag TMR2IF.

Cet avantage, combiné à la grande flexibilité de l'ensemble prédiviseur/postdiviseur, permet d'obtenir très facilement des durées d'interruption précises sans complications logicielles, donc la durée séparant deux positionnements consécutifs du flag TMR2IF :

Durée totale = temps d'une instruction * prédiviseur * postdiviseur * (PR2 +1)

III.6.10 Le module USART en mode série synchrone

USART signifie « **Universal Synchronous Asynchronous Receiver Transmitter** ». C'est un module qui permet d'envoyer et de recevoir des données en mode série, soit de façon synchrone.

Le module USART de notre PIC gère uniquement 2 pins, à savoir RC6/TX/CK et RC7/RX/DT. Comme nous savons qu'une liaison série synchrone nécessite une ligne dédiée à l'horloge, il ne nous reste donc qu'une seule ligne pour transmettre les données.

On en déduit que le PIC ne pourra émettre et recevoir en même temps en utilisant

l'USART en mode synchrone. On parlera donc de liaison « half-duplex ». Par contre, le mode asynchrone n'a pas besoin de ligne d'horloge, il nous restera alors 2 lignes pour communiquer, chacune étant dédiée à un sens de transfert. Nous pourrions donc envoyer et recevoir des données en même temps. On parlera de liaison « full-duplex ».

Comme, de plus, nous savons maintenant que l'horloge, en mode synchrone, est sous le contrôle du maître, notre USART pourra fonctionner dans les modes suivants :

- **Ssynchrone full duplex** : émission sur TX et réception sur RX
- **Ssynchrone half-duplex** : sur 2 lignes (TX et RX) ou sur une ligne (TX/RX reliées)
- **Synchrone maître** : émission horloge sur CK et émission/réception données sur DT
- **Synchrone esclave** : réception horloge sur CK et émission/réception données sur DT

6.10.A.le registre TXSTA

CSRC	TX9	TXEN	SYNC	-	-	TRMT	TX9D
------	-----	------	------	---	---	------	------

le registre «**TRANSMITT STatus and control register** », comme son nom l'indique, contient des bits de status et de contrôle, la plupart concernant l'émission de données. Voici les bits utilisés en mode synchrone :

- **b7 : CSRC** (Clock SouRCe select bit) : détermine si notre PIC va travailler en tant que maître ou en tant qu'esclave.
- **b6 : TX9** (TRANSMITT 9 bits enable) : indique si les données sont codées sur 8 ou sur 9 bits.
- **b5 : TXEN** (TRANSMITT ENable bit) : permet de lancer l'émission car le module USART étant incapable d'effectuer les 2 opérations en même temps.
- **b4 : SYNC** (SYNChronous mode select bit) permet de choisir entre le mode synchrone (1) et asynchrone (0). Dans notre projet, nous utilisons le mode synchrone, donc ce bit devra être positionné à « 1 ».
- **b3 et b2**: Non Utilisés : lus comme « 0 »
- **b1 : TRMT** (TRansMiT shift register status bit) : indique quand le registre TSR est vide, c'est-à-dire quand l'émission de la dernière valeur présente est terminée.
- **b0 : TX9D** (TRANSMIT 9th bit Data) : contient la valeur du 9^{ème} bit à envoyer, dans le cas d'émissions de mots de 9 bits (via le positionnement de TX9).

6.10.B.Le registre RCSTA

RCSTA, pour « ReCeive STatus and control register », contient d'autres bits de contrôle et de status, axés principalement sur la réception des données. Voici les bits qui le composent.

SPEN	RX9	SREN	CREN	-	-	OERR	RX9D
-------------	------------	-------------	-------------	---	---	-------------	-------------

RCSTA en mode synchrone

Quant au rôle de chacun de ces bits :

- **b7 : SPEN (Serial Port ENable bit) :** met le module USART en service.
- **b6 : RX9 (RECEIVE 9 bits enable bit) :** permet de choisir une réception sur 8 ou sur 9 bits.
- **b5 : SREN (Single Receive ENable bit) :** lance la réception d'un seul octet. La communication se terminera d'elle-même à la fin de la réception de l'octet.
- **b4 : CREN (Continuous Receive ENable bit) :** lance la réception continue. Les octets seront reçus sans interruption jusqu'à ce que ce bit soit effacé.
- **b3 et b2:** Non Utilisés : lus comme « 0 »
- **b1 : OERR (Overflow ERRor bit) :** indique une erreur de type overflow.
- **b0 : RX9D (RECEIVE 9th Data bit) :** contient le 9^{ème} bit de donnée reçue, dans le cas d'une réception des données sur 9 bits (via le positionnement de RX9).

6.10.c.Le registre SPBRG

Ce registre « **Serial Port Baud Rate Generator** » permet de définir la fréquence de l'horloge utilisée pour la transmission, et donc de fixer son débit [9].

comparant le code PIN, et quand sa JUSTE, il est d'envoyer le code PIN sous forme de binaire à l'ordinateur par la PortE PARALLELE et en même temps il est en envoyant un l'alerte à moteur LE élévation LES BARRIERE pour ouvrir et fermer pour un temps limité

III.7 PORT PARALLELE DE L'ORDINATEUR PC

Les ports d'entrée- sortie sont des éléments matériels de l'ordinateur, permettant au système de communiquer avec des éléments extérieurs, c'est à dire d'échanger des données, d'où l'appellation d'interface d'entrée-sortie (notée parfois interface d'E/S).

III.7.1 Description du port parallèle

La transmission des données en parallèle consiste à envoyer des données simultanément sur plusieurs canaux (fils). Les ports parallèles présentent sur les ordinateurs

personnels permettent d'envoyer simultanément 8 bits (un octet) par l'intermédiaire de 8 fils. On trouve deux types de ports:

- o **EPP** (port parallèle amélioré.
- o **ECP** (port parallèle à capacité améliorée)

Une interface parallèle est en principe commandée par des signaux qui peuvent prendre deux niveaux 0V et 5V (**signal TTL**).

Le circuit à réaliser (circuit de commande) sera commandé par le port parallèle, via une interface de programmation, dans notre cas développé en langage Delphi. Les ports parallèles compatibles **TTL**, ne sont pas protégés contre les surcharges, leur utilisation réclame le respect des règles suivantes :

- Les entrées ne supportent que des tensions comprises entre 0V et 5V.
- Ne court circuiter pas les sorties et n'interconnecter pas les sorties.
- Ne court circuité pas les sortie et ne interconnecté les sorties
- N'appliqué pas une tension extérieure sur la sortie. [6]

III.7.2 Géométrie du port parallèle

Le port parallèle se présente sous la forme d'une prise DB25 **Fig.(IV.11)**.

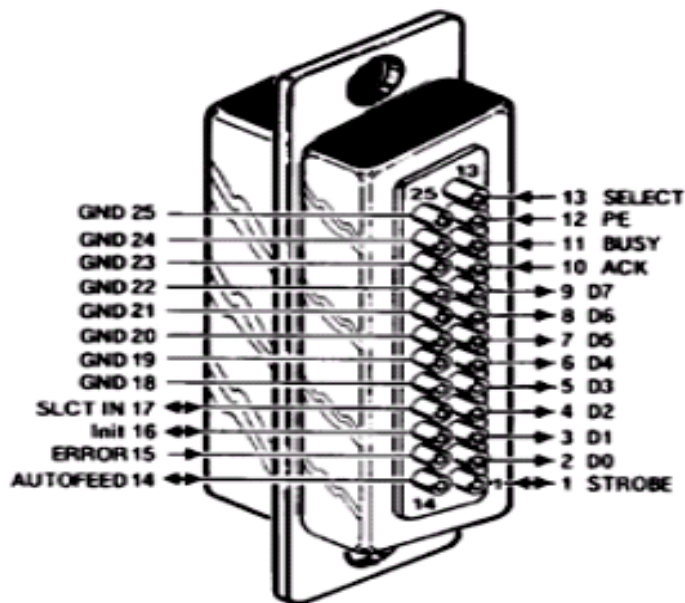


Figure (III.15) : Présentation d'un port parallèle

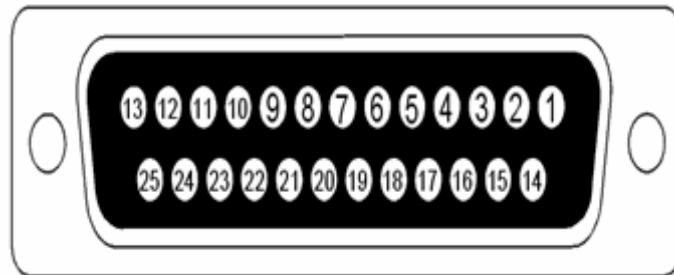


Fig.(III.16): Le port parallèle femelle vu de l'extérieur

7.3.Affectation physique des broches

On peut spécifier le rôle de chaque broche du port parallèle dans le tableau **Tab.(IV.1)** :

Broche	Nom	Sens de transmission
1	STROBE/	Entrée-sortie
2	D ₀	Entrée-sortie*
3	D ₁	Entrée-sortie*
4	D ₂	Entrée-sortie*
5	D ₃	Entrée-sortie*
6	D ₄	Entrée-sortie*
7	D ₅	Entrée-sortie*
8	D ₆	Entrée-sortie*
9	D ₇	Entrée-sortie*
10	/ACK	Entrée
11	BUSY	Entrée
12	PE	Entrée

13	SELECT	Entrée
14	/AUTOFEED	Entrée-sortie
15	/ERROR	Entrée
16	/INIT	Entrée-sortie
17	/SELECT IN	Entrée-sortie
18-25	MASSE	Entrée-sortie

Tab.(III.2) : L'affectation physique des broches

Avec :

- **Les lignes de données (adresse : 378h) :** broches 2 à 9
- **L'état de l'imprimante (adresse : 379h) :** broches 10, 11, 12,13 et 15
- **Commande de l'imprimante (adresse : 37Ah) :** broches 14,16 et 17

7.4.Description du registre de données

STROBE : cette ligne activée avec un niveau bas indique à l'imprimante que des données sont présentées sur les lignes **D₀** à **D₇** et qu'il faut les prendre en compte.

D₀ à D₇ : c'est le bus de données sur lequel véhicule la valeur du caractère à imprimer. On ne peut écrire sur ce port, à moins d'avoir un port parallèle étendu (C'est le cas pour les ports de type ECP/EPP).

7.5.Adresse du port parallèle

L'adresse de base des différents ports parallèles existants :

- LPT1 : 378h ou 888d
- LPT2 : 278h ou 632d

Un port de sortie de 8 bits est associé au registre de données, port par lequel les données sont normalement transmises à l'imprimante. Les sorties sont compatibles **TTL** ce qui veut dire que leur niveau haut correspond à des tensions comprises entre **3.5V** et **5V** et qu'elles ne supportent pas de charges supérieures à 10mA. [7]

le porte parallele ijs porte le chiffre vers le pc qui à son tour montre toutes linformation avec une photo de l'interface.

III.8 REALISATION D'UN SYSTEME DE SECURITE PAR LA CARTE PUCE

IV.2.1 Algorithme de transmission du message de la carte a puce

La carte puce est un composant complexe qui est constitué d'un microcontrôleur et une mémoire, dans notre travail nous avons pas besoin d'utilisé la mémoire externe pour cela nous avons utilisée seulement microcontrôleur à cause du problème de coût. Le microcontrôleur (16F84) envoie en série le code personnel qui contient 100 éléments (caractères), vers le lecteur (Figure III.1). Le temps nécessaire pour envoyée le code est moins de 1 ms

```

Début
    Pour q=1 jusque 8 faire
        Porta[q]=Tab[p][0]
    Fin pour
    Pour q=1 jusque 8
        Portb.f7= Porta[q]
        Pause_us(100)
    Fin pour
Fin
  
```

IV.2.2 Algorithme de Lecture du code de la carte puce :

Le lecteur de la carte puce dans notre travail est le PIC 16F877A qui contient un code global de 95 caractères pour tous les utilisateurs, et 5 caractères pour chaque utilisateur si t'a dire :
 $95 + 5 * 255 = 1370$ caractères.

Le message est Reçu en série par le **Port d pin1**, puis le microcontrôleur compare le message (code) avec les codes enregistré dans la mémoire du microcontrôleur (**PIC 16F877A**) qui va nous donnée l'action (ouverture ou fermeture de la serrure).

On obtient deux cas :

- Si le code est correcte donc le portc envoie un code entre 0 et 255, puis le LPT (port parallèle) lit le code, puis le PC affiche les informations d'utilisateur.
- Si le code est incorrect donc il faut allumée l'alarme.

```

Debut
bd=1
    Pour i=1 jusque 95 faire
        Pour j=1 jusque 7 faire
            portb=Lire caractère série par pord.f1
        fin pour
        si (portb≠bb[i]) alors bd=0
    fin pour
    si (bd) alors
        pour i=0 jusque 5 faire
            Pour j=1 jusque 8 faire
                ba[i]=Lire caractère série par pord.f1
            fin pour
        fin pour
    }
k=0
  
```

```
pour i=1 jusque 255 faire
    bd=1
    pour j=1 jusque 5 faire
        si (ba[j] ≠ bb[j]) bd=0
        fin si
    fin pour
    si (bd) alors
        portc=255-i
        k=1
    fin si
fin pour
si (k=0) alors
    portc=0
    porta=3
fin si
fin
```

III.9.APPLICATION

Les principaux secteurs qui utilisent la carte à puce sont :

- l'industrie des télécommunications avec par exemple les cartes téléphoniques pré-payées (comme par exemple les cartes téléphoniques classiques qui sont des cartes à mémoires avec contact) ou bien avec les cartes SIMs insérées dans les téléphones GSM.
- l'industrie bancaire et monétaire avec les cartes de crédits qui sont des cartes à micro-processeur avec contact (exp: Europay, MasterCard, Visa).
- le secteur de la santé (exp: la carte Vitale)
- l'industrie audiovisuelle avec la télévision à péage, etc.

Mais de nouvelles applications pour la carte à puce sont aujourd'hui à l'étude pour :

- Le porte-monnaie électronique.
- Les transports en commun.
- Le contrôle d'accès physique de personnes à des locaux, etc.
- L'identification : à des sites sur l'Internet, etc.
- Les applications de fidélité (exp: l'accumulation de miles de voyage en avion gratuits à chaque transaction, etc).
- Les jeux comme le pocket-gaming (qui consiste en des terminaux de poche - comme le deviennent les téléphones ou les consoles portables - pour jouer par exemple à des jeux de casino, etc).

Le domaine principal qui permettra à la carte à puce de s'imposer est l'Internet qui au travers des services exigent de plus en plus une identification et une sécurisation des transactions. Ainsi la carte peut être utilisée pour ces services en stockant par exemple les clefs servant à l'authentification et à la sécurisation.

III.10 CONCLUSION

Grâce à son expérience et ce que nous étudions dans ce chapitre conclure que la carte à puce. Est un des moyens le plus de protection et de sécurité a fin de nous permettre rapidement en œuvre

Le processus avec une précision meilleur moment ziaai et sans préavis, l'utilisation accrue de notre

Temps et dans tous les galat et souvent.