وزارة التعليم العالـــي و البحـــث العلمـــــــي

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

جامعـــة – أكلي محند أولحاج- البويرة

UNIVERSITE Akli Mohand Oulhadj —Bouira

(ALGERIE)

Mémoire de Master

Présenté au département de Génie Electrique
Faculté des Sciences et Sciences Appliquées
Pour obtenir le diplôme

De Master

En:

Systèmes électroniques complexes

Par :

Mr. Merakchi Abdenour

Mr. Chouiki Khaled

Thème

Conception et réalisation d'une nouvelle machine de vote électronique

Soutenu le 24/09/2017 devant le jury composé de :

Mr. M.Ch Kihel	Maître Assistant à l'université de Bouira	*Président*
Mme. K. Dichou	Enseignante à l'université de Boumerdes	*Co-Encadreur*
Mr. M. Fekir	Maître Assistant à l'université de Bouira	*Examinateur*
Mme. S. Madi	Maître Assistant à l'université de Bouira	*Examinateur*
Mr. M. Rezki	Maître de conférences à l'université de Bouira	*Encadreur*

Dédicace :

Ce modeste travail est dédié :

A mes très chers parents.

A mes chers frères et sœurs

A mes chères amis.

A tous les nombres de ma famille grands et petits.

A tous les étudiants de la promotion 2016/2017

Option : Système Electronique Complexe

Chouiki khaled

Dédicace

Ce modeste travail est dédié :

À mon père

À ma mère

À tous mes frères et sœurs, ainsi que leurs enfants

À mes beaux parents et à toute ma famille

À tous mes amis et collègues

À tous les étudiants de la promotion 2016/2017

Option : Système Electronique Complexe

MERAKCHI ABDENOUR

Remerciements

En premier lieu, nous tenons à remercier Dieu, notre créateur pour nous avoir donné la force pour accomplir ce travail.

Nous tenons à remercier Dr. Karima DICHOU, pour sa patience, et surtout pour sa confiance, ses remarques et ses conseils, sa disponibilité et sa bienveillance.

Nous remercions également de l'encadreur Dr Med REZKI et les membres du jury pour avoir accepté d'évaluer ce travail et pour toutes leurs remarques et critiques, ainsi que le personnel et les enseignants de L'université de Bourra sans oublier les enseignants qui ont contribué à notre formation.

Nous remercions monsieur le chef du Département de Génie Electrique ainsi que tout le personnel et les enseignants du département pour leur soutien inestimable.

Nous tenons à exprimer nos sincères remerciements à tous les professeurs qui nous ont enseigné et qui par leurs compétences nous ont soutenu dans la poursuit de nos études.

Enfin, on remercie tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail.

Merci à vous tous

Résumé :

Dans le monde ou la technologie est de plus on plus avancée dans tous les domaines, celui du vote électronique représente le moyen d'expression principal d'une démocratie qui existe dans toutes les sphères de la société. Il existe deux catégories de systèmes de vote électronique ; on trouve le vote qui se déroule dans un bureau et un autre qui se fait à distance (par internet ou réseau informatique).

Le système le plus connu et plus utilisé dans le monde est celui des bureaux. Pour cette raison, nous avons décidé d'étudier ce dernier afin de l'amélioré. Parmi leurs problèmes majors on trouve celui de l'authentification de l'administrateur qui doit être bien assurée. Notre projet consiste à concevoir et réaliser une nouvelle machine de vote électronique autonome, à base celles existantes, en utilisant un microcontrôleur et une carte à puce permettant d'assurer une authentification dynamique de l'administrateur du bureau pour plus de sécurité. Cela toute en choisissant judicieusement les composants de la machine ainsi que la carte adéquate.

La machine conçue est à base d'un microcontrôleur de la famille PIC qui communique avec l'électeur avec un clavier et un afficheur LCD, et qui communique avec la carte à puce afin d'authentifier l'administrateur. Les résultats du vote seront enregistrés dans la mémoire EEPROM interne du PIC afin de les protéger contre les attaques. Après la vérification du fonctionnement de notre système en le simulant sous le logiciel Proteus, nous l'avons réalisé sur plaque d'essai. notre système fonctionnement prévu.

Mots clés: Vote électronique, Microcontrôleur, Carte à puce.

ملخص :

في عالم تكون فيه التكنولوجيا أكثر تقدما في جميع المجالات، فإن التصويت الإلكتروني هو الوسيلة الرئيسية للتعبير عن الديمقراطية القائمة في جميع مجالات المجتمع.

يتمثل مشروعنا في انجاز آلة لتصويت إلكتروني مستقلة جديدة باستخدام متحكم وبطاقة ذكية لتسهيل عمل الإدارة وعملية التصويت .لايتم التعامل أو لمس هذه الآلة إلا عن طريق مدير مسؤول عن المكتب . ولهذا قمنا باستخدام كلمة مرور بالبطاقة الذكية لمزيد من الأمن.

في سياق هذا العمل، في الجزء الأول قمنا بعرض لأنواع من آلات التصويت الإلكترونية الموجودة، في الجزء الثاني تحدثنا عن البطاقة الذكية وتكييفها للنظام الذي نود تصميمه، وسيتم تخصيص الجزء الثالث لتصميم ووصف وبرمجة النظام ، وأخيرا في الجزء الأخير سوف نقوم بمحاكاة وانجاز النظام وتركيبه .

كلمات سرية: الانتخاب الإلكتروني، الميكروكنترولر، بطاقة ذات رقاقة الكترونية.

Sommaire

Chapitre 2 : Cartes à puce

Chapitre 3 : Description du système à concevoir

Chapitre 4 : Simulation et réalisation

Liste des figures

Chapitre 1

Chapitre 2

Chapitre 3

Chapitre 4

Liste des tableaux

Chapitre 2

Chapitre 3

Chapitre 4

LISTE DES ABREVIATIONS

ATR : Answer To Reset.

APDU : Application Protocol Data Unit.

CLK : Clock.

CPU : Central Processing Unit.

EEPROM : Electrically-Erasable Programmable Read-Only Memory.

EMV : Europay, Mastercard, Visa.

EPROM : Electrically Programmable Read-Only Memory.

GND : Ground (supply voltage).

GSM : Global System for Mobile communications.

I/O : Input/Output.

I^2C : Inter Integrated Circuit bus.

IC : Integrated Circuit.

ID : Identity.

ISO : International Standardisation Organisation.

MSB : Most Significant Bit.

LSB : Less Significant Bit.

PIN : Personal Identification Number.

RAM : Random Access Memory.

RF : Radio Frenquency.

RFID : Radio Frenquency Identification.

RFU : Reserved for Future Use.

ROM : Read-Only-Memory.

RSA : Rivest-Shamir-Adleman.

RST : Reset.

SIM : Subscriber Identification Module.

SW1 : Status Word 1.

SW2 : Status Word 2.

TPDU : Transport Protocol Data Unit.

Vcc : Supply voltage(+).

Vpp : Programming voltage.

RS232 : Recommended Standard 232.

Introduction générale

Depuis de nombreuses années le vote électronique existe sous la forme de machine à voter ou sous le nom d'urnes électroniques. Les machines de vote électroniques sont des dispositifs électroniques utilisés pour enregistrer des voies à la place des bulletins de vote et des boîtes qui ont été utilisés précédemment dans les systèmes de vote classiques. Cela afin d'accélérer le processus.

Il existe deux grandes catégories de systèmes de vote électronique. Le premier est basé sur la visite du bureau de vote en remplaçant le papier par des machines ou des urnes électroniques. Le deuxième système est basé sur la technologie à distance: réseaux informatiques et Internet. Les systèmes les plus populaires sont ceux installés dans les bureaux de vote. Malgré ces avantages, il n'est pas encore utilisé dans certain pays à cause du manque de confiance en sa technologie.

L'objectif de ce travail est d'amélioré le système de vote électronique qui se fait dans les bureaux de vote vu qu'il est le plus répondu dans le monde, dans le but de trouver des solutions à ses inconvénients. Le problème que nous voulons résoudre est au niveau de la manipulation de la machine de vote qui doit être assuré par un administrateur déclaré responsable du bureau.

Comme les cartes à puce sont de plus en plus utilisées dans la vie quotidienne, nous avons pensé à l'utilisation d'une authentification dynamique basée sur une carte à puce associé à un mot de passe avec un nombre d'essais limité pour l'administrateur. Plusieurs travaux ont été fait dans le but d'améliorer ces machines mais aucun d'entre eux n'a proposé l'utilisation d'une carte à puce pour l'authentification de l'administrateur. Le choix des composants de la machine ainsi que ceux de la carte qui convient le mieux à notre application doit être fait soigneusement afin de réduire les couts tout en assurant le bon fonctionnement. [8]
Ce mémoire est organisé comme suit :

La première partie de notre travail est consacré à la présentation des déférents types de machines de vote électroniques existantes. Dans la deuxième partie nous présenterons une étude sur les cartes à puce et essentiellement sur la carte qui convient à notre système. La troisième partie sera dédiée à la conception, description et programmation de notre système, et finalement, dans la dernière partie nous passerons à sa simulation et réalisation.

CHAPITRE 1

Le vote électronique

1.1. Introduction

Le terme de "vote" désigne une multitude de réalités : On vote pour désigner le président de la république de son pays, éliminer un candidat, choisir les représentants du personnel de son entreprise … etc.

Le but de ce premier chapitre est de présenter les différents systèmes de vote électronique, les systèmes en usage dans différents pays, ainsi que les avantages et les inconvénients de ces systèmes.

1.2. Le vote classique

Dans le système de vote classique, les choix et les intentions des électeurs sont représentés sous forme de papier, les électeurs qui entrent dans le bureau de vote doivent être identifiés. Si l'identification est adoptée, ils sont capables de voter.

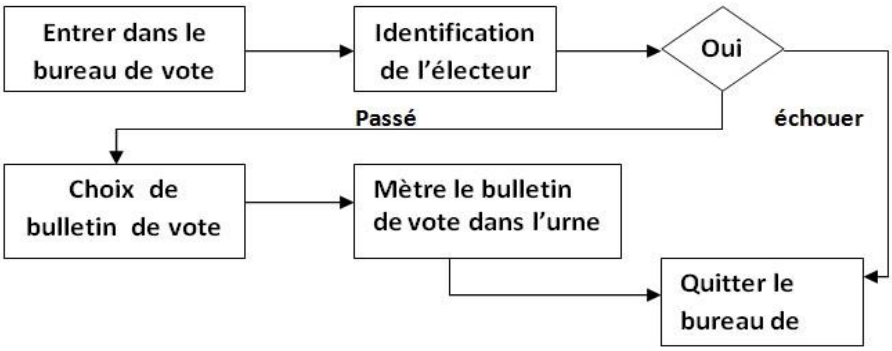


Figure1.1: Chronogramme de Système de vote classique [2]**.**

1.3. Le vote électronique

Le vote papier traditionnel peut être fastidieux et incommode. Le vote électronique non seulement accélère l'ensemble du processus, mais il est moins cher et plus confortable pour les électeurs et les autorités. Il réduit également les chances d'erreurs. Un système de vote électronique doit fournir toutes les fonctionnalités de base que le vote classique. En outre, il devrait fournir plus de services afin de rendre le processus plus fiable et sécurisé. Les données de l'élection sont enregistrées, stockées et traitées principalement comme information numérique [8].

1.4. Historique du vote électronique

Il a fallu attendre les années 1970 pour l'apparaissions des dispositifs de vote appelés "machines à voter". Ils se présentent comme une boîte munie de 16 touches et fonctionnent sans électricité (Alcatel). Un électeur exprime son choix en appuyant sur un bouton, entraînant une roue d'entée qui ajoute une unité sur un compteur de type odomètre mécanique [3].

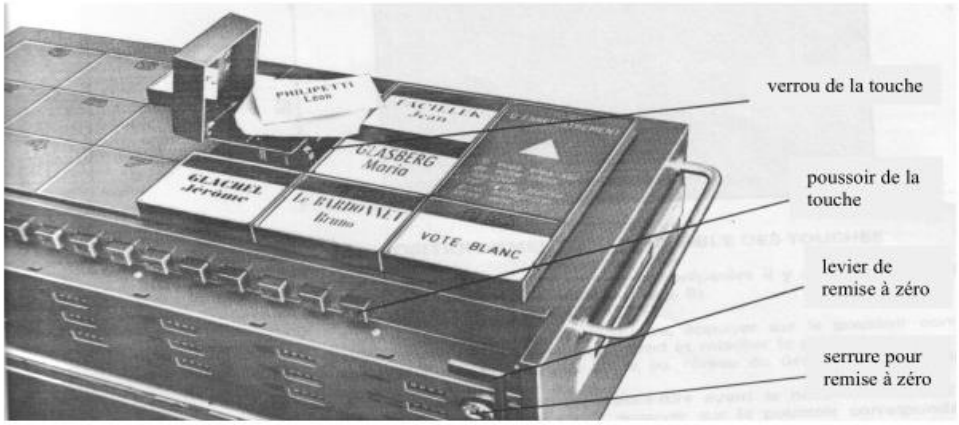


Figure1.2: Machine à voter Alcatel 1970 [3].

À partir du milieu des années 1990, les modalités de vote subissent la grande créativité. Dès lors, sont promues deux tendances qui proposent une informatisation différentes des modes de scrutin. Pour la première de ces tendances, le vote électronique signifie l'intégration via l'urne électronique « machine à voter », de procédés permettant de faire intervenir des entreprises privées dans le système de vote. L'argument commercial principal utilisé pour promouvoir ces produits repose sur l'idée d'accélérer le processus de traitement des suffrages exprimés. Pour la deuxième de ces tendances, l'informatisation du processus de vote permettra de voter à distance. C'est-à-dire de voter de chez soi, ou de n'importe où dans le monde, et ainsi éviter de se déplacer dans des bureaux de vote, mais sans aucune garantie de l'individu qui vote.[4]

Depuis les années 2000, le vote électronique se répand dans de nombreuses démocraties mais en se présentant sous des formes diverses. Certains pays tels que le (les États-Unis, l'Inde, la Suisse, France, canada, Brésil le Venezuela …etc.), utilisent déjà des systèmes de vote électronique dans le cadre d'élections majeures. Pour d'autres pays,

notamment en Europe, ces systèmes sont encore à l'état de projet ou bien font l'objet d'utilisations ponctuelles [4].

1.5. Le système de vote électronique

Un système de vote électronique doit fournir toutes les fonctionnalités de base que le vote classique. En outre, il devrait fournir plus de services afin de rendre le processus plus fiable et sécurisé. Les données de l'élection sont enregistrées, stockées et traitées principalement comme information numérique.

La recherche sur le vote électronique est un sujet très important pour le progrès de la démocratie. De ce fait, il connaisse aujourd'hui des développements importants, un bon système de vote électronique doit assurer quelques propriétés qui définissent des exigences concernant sa sécurité et son implémentation. Dans ce qui suit, nous allons définir les exigences de sécurité dont nous tiendrons compte lors de la conception du système. Les principales exigences de sécurité à respecter sont : **[21, 22, 23]**

1. Précision : un vote valide doit être compté, tandis qu'un vote invalide (i.e. ne vérifie pas la loi électoral) n'est pas pris en considération dans la phase de décompte.
2. Démocratie : seules les votants éligibles peuvent voter une seule fois.
3. Vérifiabilité : les votent peuvent être vérifié par les officiels de votes et par les votants.
4. Neutralité : le matériel et les processus électoraux ne favorisent aucun candidat ou parti en regard d'un autre.
5. Confidentialité : aucune personne ne doit pouvoir faire le lien entre le votant et son bulletin électronique (anonymat).
6. Intégrité : une vote ne peut être modifiée du moment où elle est déposée dans l'urne électronique.
7. Disponibilité : le système doit être utilisable et fiable le long des trois phases citées précédemment.
8. Authentification: s'assurer de l'identité du votant.
Le respect de ces propriétés, permet d'obtenir un système de vote électronique efficace contre les différentes attaques existantes aujourd'hui

1.6. Type de vote électronique

Il existe deux grandes catégories de systèmes de vote électronique, selon l'endroit où l'on vote : local ou à distance.

1.6.1. Systèmes de vote électronique basé sur la visite du bureau de vote

Pour ce type, le vote électronique s'effectue dans les bureaux traditionnels. Ou lieu d'utiliser le papier, des machines sont placées permettant à l'utilisateur de sélectionner leurs choix.

Dans ce cas, les électeurs sont toujours identifiés par l'utilisation de cartes d'identification. Les électeurs ne remplissent pas les cartes de vote sous la forme de papier, mais ils utilisent des boutons poussoirs sur différents appareils électroniques disponibles dans les bureaux de vote.

Les avantages de ce système sont les résultats qui sont rapidement calculés. En utilisant une machine autonome sans ou avec connectivité réseau, personne ne peut interfacer avec sa programmation et manipuler le résultat. Cependant, ce n'est pas sûr de prévenir le vote multiple par une seule personne.
La majorité de ces machines sont à base de microcontrôleur [5].

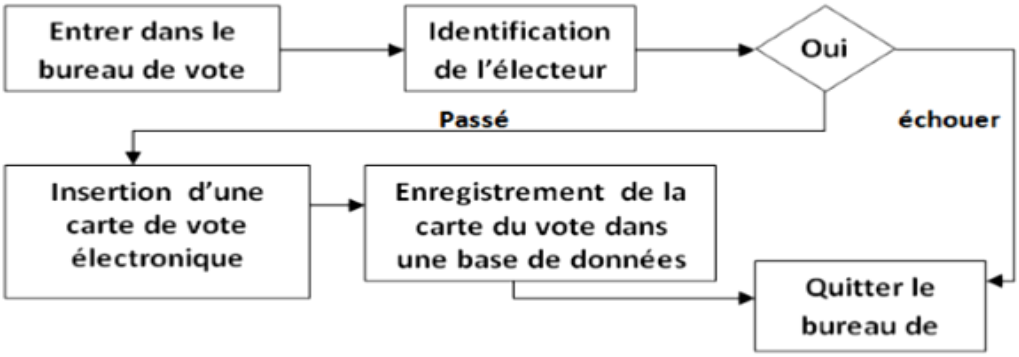


Figure1.3: Systèmes de vote électronique basé sur la visite du bureau de vote [2].

1.6.2. Systèmes de vote électronique à distance

Ce système est l'équivalent électronique du vote postal dans la mesure ou les électeurs peuvent exprimer leur vote depuis leur domicile, c'est-à-dire dans un environnement non supervise un tel système pourrait être utilisé pour les citoyennes voter dans leurs résidence. Ce système est basé sur la technologie à distance. Habituellement. Ils utilisent des réseaux informatiques et Internet pour le vote. Ils peuvent également, voter depuis l'étranger. Ceux-ci constituent les avantages les plus importants du système à distance [8].

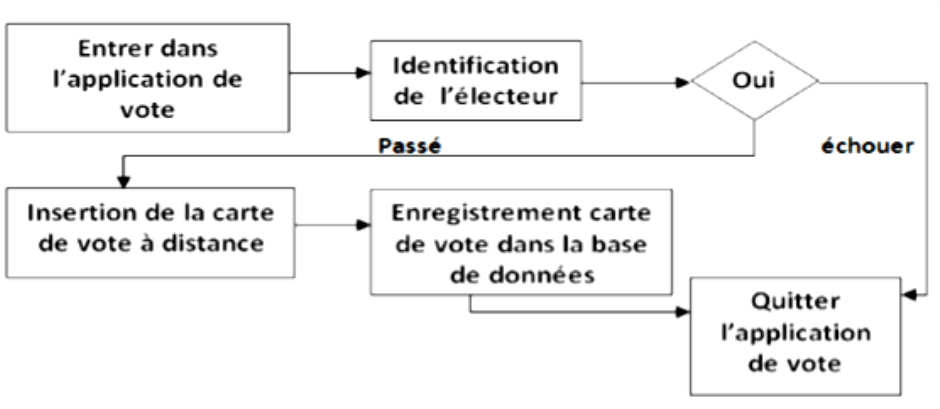


Figure1.4. Systèmes de vote électronique à distance [2].

1.7. Exemple des systèmes de vote existants

Plusieurs pays ont adopté le vote électronique, le but de cette partie est d'étudier les systèmes de vote électronique qui existent aujourd'hui dans le différents pays.

Les systèmes de vote électronique sont connaît une plus grande diversité, les systèmes les plus utilisés :

- la machine de vote AccuVote-TS (Diebold). [4]
- La machine de vote eSlate. [4]
- machine de vote Nedap/Groenendaal ES3B. [4]
- ordinateurs de vote avec bulletin dématérialisé (OdV-BD). [4]
- ordinateurs de vote avec bulletin matérialisé (OdV-BM). [4]
- kiosques. [19]
- boîtiers électroniques de vote. [19]
- Short Message Service (SMS). [20]
- vote par internet.
- vote par correspondance hybride [3].

1.7.1. Au Etats-Unis

1.7.1.1. Machine de vote AccuVote-TSX

Parmi les systèmes les plus utilisés actuellement aux États-Unis signalons l'urne (AccuVote-TSX) de la société (Diebold). L'interface de cette dernière est un écran tactile pourvu de cases à cocher : elle est présentée dans la figure 1.5. L'interface utilisateur présente en particulier un lecteur de cartes magnétiques d'identification de l'électeur. Outre la sélection des principaux candidats associés aux cases à cocher, l'électeur peut aussi choisir d'écrire le nom d'un autre candidat grâce à un clavier virtuel tactile. Ce système est également pourvu d'une interface audio-vocale.

Figure1.5: Machine de vote Diebold AccuVote-TS [6].

1.7.1.2. Machine de vote eSlate

Un autre exemple d'interface est celle du système eSlate (eSlate Electronic Voting System) muni d'une roulette qui permet de déplacer un curseur sur un écran où figurent les noms des différents candidats.

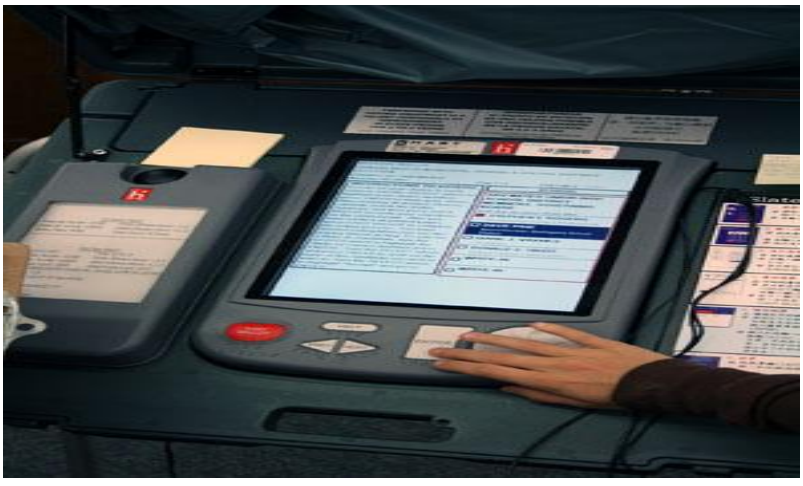


Figure1.6: Le système eSlate produit par Hart Intercivic [7].

Remarque :

Ce deux machines de vote (eSlate hart Intercic et Diebold AccuVote-TS) sont utilisés aussi au canada.

1.7.2. En France

1.7.2.1. Ordinateurs de vote avec bulletin dématérialisé (OdV-BD) [3]

Les ordinateurs de vote avec bulletin dématérialisé sont placés dans des bureaux de vote et remplacent l'usage du matériel habituel : urne, bulletins, isoloir.

L'ordinateur présente les candidats, les listes, ou les différentes réponses en cas de référendum, l'électeur fait son choix à l'aide de l'ordinateur de vote, en appuyant sur un bouton, ou en utilisant une souris, ou bien encore en mettant ses doigts directement sur l'écran tactile ; l'électeur peut confirmer son choix, ou bien revenir en arrière pour éventuellement en changer. Une fois son choix confirmé, il quitte l'isoloir et émarge.

Le dispositif enregistre les votes mais ne délivre pas de résultats avant son passage en mode dépouillement. Les ordinateurs de vote ne contrôlent pas les identités des électeurs. Ce sont les membres du bureau de vote qui continuent d'effectuer ce contrôle en faisant signer le registre des émargements, comme dans la procédure de vote traditionnelle.

Certains modèles utilisés à l'étranger transmettent directement leurs résultats sous une forme électronique en utilisant internet ou en stockant les résultats sur un support informatique (comme un disque dure ou une clef USB) qui est physiquement transmis au bureau centralisateur. Un ordinateur centralisateur recueille les contenus des supports informatiques et effectue la totalisation. Il existe aussi des dispositifs qui nécessitent l'usage d'une carte à puce.

1.7.2.2. Ordinateurs de vote avec bulletin matérialisé (OdV-BM)

Il s'agit d'un aménagement des OdV-BD destiné à munir ces dispositifs d'une possibilité de vérification des résultats. Lorsqu'un électeur exprime un choix, un bulletin est imprimé. L'électeur peut le voir à travers une vitre et vérifier sa correction. En cas de validation, le bulletin est acheminé vers une urne en vue d'un éventuel dépouillement ultérieur, sinon il est détruit et l'électeur peut de nouveau faire un choix [3].

1.7.2.3. Les Kiosques de vote

Les kiosques sont des terminaux connectés à un ordinateur central (le serveur). Ils sont situés dans des bureaux de vote. Le dispositif recueille les votes et se charge de la gestion de l'émargement des électeurs. Tout électeur peut voter en utilisant n'importe quel kiosque, dans n'importe quel bureau de vote. Après la phase d'identification et d'authentification, le vote se déroule comme avec un ordinateur de vote avec bulletin dématérialisé. [3]

Figure1.7: Kiosques de vote avec isoloirs [3].

1.7.2.4. Boîtiers électroniques de vote

Ces systèmes de vote sont utilisés lors des assemblées où l'ensemble des électeurs est réuni dans une même salle. Chaque électeur dispose d'un boîtier muni d'un pavé numérique. Il exprime son choix en saisissant un code numérique qui est transmis à l'ordinateur centralisateur. À la clôture du vote, ce dernier présente les résultats. [3]

Figure1.8: Boîtiers électroniques de vote [3].

1.7.3. En Allemagne

1.7.3.1. La machine de vote Nedap/Groenendaal ES3B

Cet machine est vendu par les compagnies Nedap (qui font principalement le matériel) et Groenendaal (qui font principalement le logiciel) , Le logiciel IIS (Integral Stem System) est téléchargé sur un PC sous le Microsoft Windows et employé pour administrer la machine Nedap/Groenendaal.

Les principaux équipements de la machine, sont :
(Une unité de lecteur, Les clefs mécaniques pour le système, Modules de mémoire de vote).

Ce modèle est largement utilisée dans Allemagne. Elle est aussi utilisée en pays bas et en France mais avec des petites modifications.

Figure1.9: machine de vote Nedap/Groenendaal ES3B [1].

1.7.4. En Suisse

Le vote électronique employé à Genève en Suisse utilise le Web depuis 2003. L'électeur reçoit une carte par courrier contenant deux données d'accès à ce système ; un des codes est secret (pour en prendre connaissance, il faut procéder à un « grattage » de cette carte). Pour voter, l'électeur peut utiliser n'importe quel ordinateur connecté à Internet. Avant de voter, l'électeur doit procéder à l'examen du certificat de sécurité fourni par le serveur Web et remplir un formulaire avec ses données d'identification, ainsi que le premier code d'accès reçu par courrier. Après avoir voté pour les différentes élections, l'électeur valide le vote en saisissant son code secret.

Figure1.10: Bulletin de vote du système de vote par Internet utilisé en Suisse [7].

1.7.5. En Brésil

Figure1.11: L'urne électronique brésilienne [7].

L'objectif initial de l'urne électronique brésilienne était d'accélérer le comptage des votes. Elle a pris la forme d'un poste informatique incliné à 45°. Celui-ci est constitué d'un écran monochrome de 9.4" et d'un clavier mécanique comporte 10 touches numériques disposées de la même manière que sur un appareil téléphonique : en dessous ce trouvent trois touches de commandes rectangulaires Ces touches correspondent aux fonctions suivantes : Voter blanc (couleur blanche), corriger son vote (couleur rouge), confirmer le vote (couleur verte).

1.8. Avantages de vote électronique

Les systèmes de vote électronique pourraient simplifier le processus électoral et le rendre plus accessible aux électeurs, de sorte que :

- Rapide de manipulation.
- Préviennent le vote multiple.
- Sécurisé avec l'authentification des utilisateurs.
- améliorant et facilitant le scrutin de tout, y compris des personnes handicapées, par la mise en place d'interfaces humain-machine adaptées. [4]

1.9. Inconvénients de vote électronique

- Les systèmes de vote électronique ont des problèmes de sécurité (L'atteinte à la sincérité du suffrage et au secret du vote, les logiciels peuvent présenter des erreurs il est facile de frauder en changeant le logiciel de vote avant les élections).
- Les systèmes de vote électroniques sont opaques (leur fonctionnement est hors de portée des électeurs). [4]
- L'absence de preuves de vote matérielles à recompter exclut toute vérification indépendante des résultats énoncés par la machine à l'issue du scrutin. Il est donc impossible de prouver si ces résultats sont justes ou erronés. Paradoxalement, cette absence de preuves rend juridiquement incontestables les résultats pourtant invérifiables. [4]

1.10. Conclusion

Les systèmes de vote électroniques existant aujourd'hui permettent l'election dans une station ou à distance (par internet). Malgré ces avantages, certain pays ne l'adoptent pas car ils ne font pas confiance à sa technologie ou à la sécurité informatique. Des chercheurs travaillent toujours pour l'élaboration d'un nouveau système plus rapide, sécurisé et digne de confiance que ceux qu'on trouve actuellement.

CHAPITRE 2

Les cartes à puce

2.1. Introduction[9]

Depuis son apparition, la carte à puce ne cesse de conquérir de nouveaux marchés tout en gagnant la confiance du consommateur. Elle est aujourd'hui omniprésente dans notre environnement : télécartes, cartes bancaires, cartes Vitale, cartes de décryptage de télévision par satellite ainsi que toutes les versions de cartes privatives de diverses enseignes commerciales sont autant de cartes à puce issues d'une même technologie. [1]

Dans ce chapitre nous faisons le point sur le concept de carte à puce, son évolution et ses différents types. Cela inclus les standards internationaux et leurs caractéristiques physiques et électriques. Nous décrivons également sa communication avec lecture.

2.2. Définition de la carte à puce

Une carte à puce est une carte plastique qui intègre un circuit électronique capable de manipuler (stocker, calculer, etc.) des informations de façon sécurisée. C'est en quelque sorte un ordinateur portable et résistant aux altérations. Il s'agit donc bien d'une carte intelligente (smart card comme on l'appelle dans les pays anglo-saxons) et non d'une simple carte à piste magnétique puisqu'elle embarque un circuit logique. D'ailleurs contrairement à cette dernière, la carte à puce n'a pas besoin d'accéder à une base de données distante lors d'une transaction puisqu'elle possède sa propre puissance de calcul et sa propre capacité de stockage d'informations dans un mode sécurisé. La plupart des normalisations applicables à ces cartes sont décrites dans l'ISO 7816. [9]

2.3. Historique

La première carte à puce à base de microcontrôleur a vu le jour en 1979. Elle était fabriquée par Motorola pour Bull CP8 et disposait d'une unité centrale de type 6805 (un petit microcontrôleur 8 bits de Motorola) et d'une mémoire programmable de seulement 1 Ko, ce qui était déjà beaucoup pour l'époque.

Entre 1980 et 1981, les premières expérimentations de télévision payante-on ne disait pas encore crypté- ont eu lieu grâce aux efforts conjoints de Bull et de Philips, mais n'ont été suivies d'aucun débouché commercial de grande ampleur. La véritable « sortie publique » de la carte à puce eut lieu en réalité en 1983 avec les premières cartes de paiement téléphonique de France Télécom. [10]

Fig. 2. 1: La première véritable application de la carte à puce a été la télécarte [10].

Année	Evénement
1979	Première carte à puce fabriquée par Motorola pour Bull CP8.
1980-1981	Premières expérimentations de télévision payante.
1983	Première cartes à puce téléphonique France Télécom.
1984	Première version de la carte bleue à puce à base de carte Bull CP8.
1987	Publication des normes ISO 7816.
1989	Premières cartes GSM pour téléphones mobiles (Gemplus).
1998	Premières cartes à puce programmables en Java ou « Java Cards ».

Tab.2. 1: Quelques années significatives de l'évolution de la carte à puce. [10]

2.4. Domaines d'utilisation

La carte à puce représente un moyen sécurisé dépassant de loin les cartes à piste magnétiques. Depuis sa création, la carte à puce s'est fait un grand chemin dans les domaines suivants : le bancaire, les télécommunications, le multi-applicatif (carte d'université, carte d'accès, carte de fidélisation) et autres. Aujourd'hui, le premier client de ces cartes est l'industrie des télécommunications grâce aux cartes SIM et USIM. Selon l'étude de l'association EuroSmart, 5200 millions de pièces ont été distribuées l'an 2012. [11]

Aujourd'hui on trouve plus de 5 milliards de cartes. Les domaines d'utilisation des cartes à puce sont de plus en plus vastes, on les trouve dans :

- **Monétique** : Carte bancaire, Porte-monnaie.
- **Identification** : cartes d'identité nationales, passeport, passeport biométrique.
- **Enseignement** : carte d'étudiant et/ou de restauration.
- **Téléphonie mobile** : carte SIM.
- **Secteur médical** : carte Vitale en France, chifa en Algérie.
- **Titre de transport.**
- **Sécurité informatique** (authentification forte et signature électronique). [12]

2.5. Standardisation

Les cartes sont très standardisées car elles doivent être utilisables avec la gamme la plus large possible de lecteurs dans le monde entier. la normalisation concerne au moins 3 points

Des paramètres physiques : Taille de la carte, position de la puce et ses contacts.

Des paramètres électriques : tension d'alimentation et niveaux électriques mise en œuvre ainsi que le brochage de la puce sur la carte.

Des paramètres logiciels : définissent le mode de dialogue avec la carte (commandes).

Les normes ISO 7816 :

ISO 7816-1:

-définit les caractéristiques physiques

ISO 7816-2:

-définit le dimensionnement physique

ISO 7816-3:

-définit l'interface électrique

-Les protocoles de transmission TPDU

T=0:

Protocole orienté Octet

T=1:

Protocole orienté paquet

T=14:

Réservé pour les protocoles propriétaires

-la sélection d'un type de protocole

-la réponse à un RESET (ATR)

-la fréquence d'horloge

-la vitesse de communication

ISO 7816-4:-définit les messages APDU. [13]

2.6. Différents types de cartes a puce

La différence entre les cartes à puce est décrite par deux critères. Le premier est la méthode d'écriture et de lecture des données et le deuxième est le type et le contenu de la puce intégré dans la carte (voir Table 2.2). [11]

Type de puce	**Mémoire**	**Sans sécurité** **Avec sécurité**
	Microprocesseur	**Avec coprocesseur de cryptographie** **Sans coprocesseur**
Méthode de transmission	**Avec contact**	
	Sans contact	
	A interface dual	

Tab.2. 2: Classification des cartes à puce selon le contenu de la puce et la méthode de transmission [11]

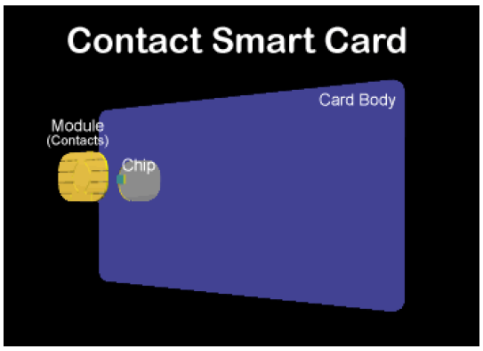


Fig. 2. 3: Carte à puce avec contact

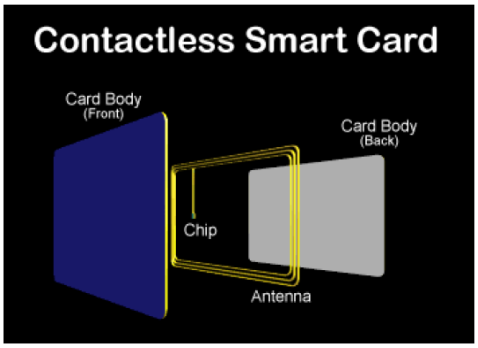


Fig. 2. 4: Carte à puce sans contact

2.6.1. Cartes à contacts

On peut distinguer trois grandes familles de cartes à puce à contact

1. Les cartes à mémoire simple.
2. Les cartes à mémoire personnalisée.
3. Les cartes à microprocesseur.

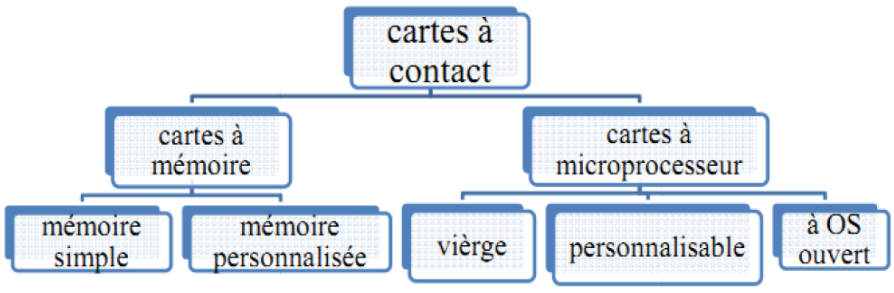


Fig. 2. 5: Différents type de carte à puce [13].

a. Les cartes à mémoire :

La carte à mémoire appelée aussi carte synchrone en raison de son protocole de dialogue. On y trouve :

a.1. les cartes à mémoire simple :

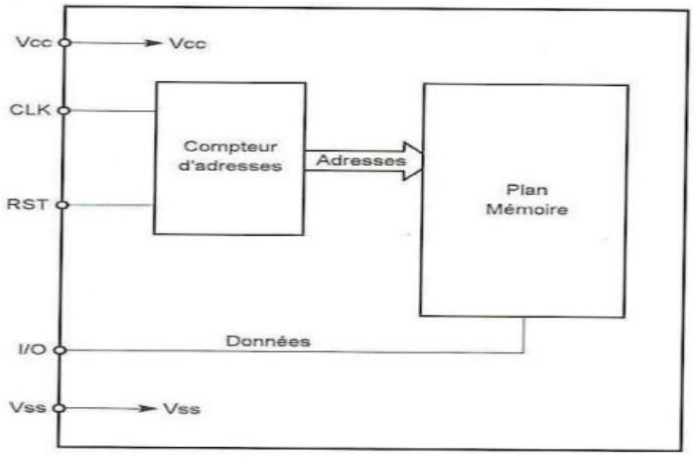


Fig. 2. 6: Synoptique interne d'une mémoire simple [10] .

Comme son nom l'indique, une carte à mémoire simple renferme une certaine quantité de mémoire, sans aucune protection particulière. Cela signifie que n'importe qui peut y lire ou y écrire des informations. La plupart des cartes à mémoire simple sont réalisées en technologie EEPROM, et son donc recyclables (effaçables et réinscriptibles). Leur capacité est généralement de l'ordre de quelques kilobits, voire quelques dizaines. Ces cartes sont en principe destinées à des applications n'ayant pas à être sécurisées : fichier portable non secret, cartes de suivi de production.[10]

a.2. Les cartes à mémoire personnalisée :

En revanche, pour mériter l'appellation de « carte à mémoire personnalisée », une carte à puce doit contenir au moins l'un des quatre systèmes de protection suivants, réalisés en logique câblée sans le secours d'un microprocesseur :

- Zone protégée en écriture après destruction d'un fusible.
- Zone protégée en lecture et écriture par un « code porteur » (code confidentiel dit « PIN » .
- Blocage de la cane au bout d'un nombre donné de présentations d'un PIN erroné ;
- Protection par un «code émetteur» (l'émetteur étant l'organisme qui délivre les cartes et décide de leur contenu).

Deux technologies coexistent dans cette famille : EPROM ineffaçable puisque la puce est enfermée dans un enrobage opaque aux ultraviolets, et EEPROM effaçable puis reprogrammable.

Les capacités disponibles sont généralement inférieures à 1 kilobit. Ces cartes sont un peu plus perfectionnées d'envisager des applications exigeant à la fois une meilleure sécurité et de fréquentes mises à jour d'informations : cartes prépayées rechargeable, porte-monnaie électronique simple, contrôle d'accès, dossier portable sécurisé, carte d'abonnement.

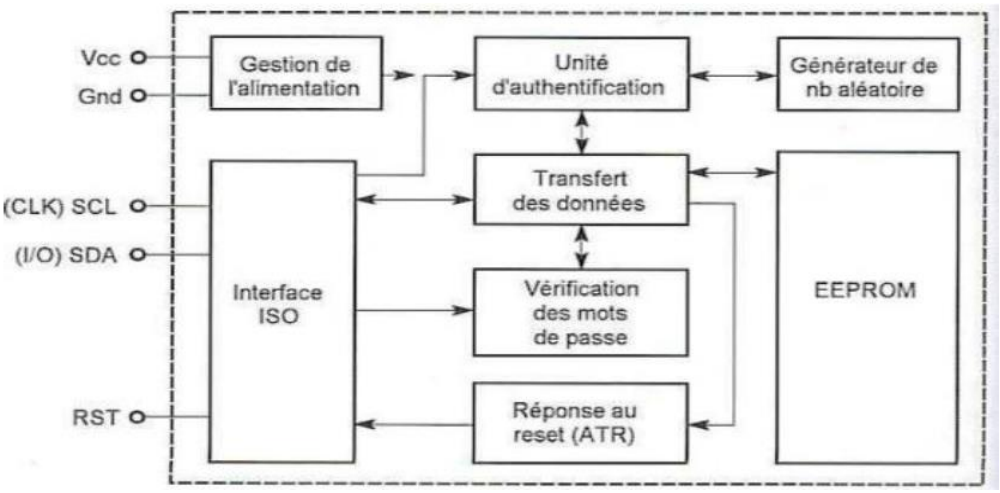


Fig. 2. 7: Synoptique interne d'une carte mémoire personnalisée [10].

b. Les cartes à microcontrôleur :

La carte à microcontrôleur (dite aussi à microprocesseur ou bien à microcalculateur), 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, de mémoire morte, de mémoire vive, de mémoire EEPROM, d'une interface d'entrée/sortie série et de toute la logique nécessaire pour faire fonctionner tout cela.

Produit informatique à part entière, elle contient plusieurs systèmes de protection :

- Zone protégée en écriture ou en écriture et lecture par un code secret émetteur, ceci après personnalisation par l'émetteur ;

-Zone protégée en lecture et écriture par un code secret porteur (PIN)

- Blocage de la carte après présentation d'un nombre donné de codes secrets erronés, mais avec possibilité de réhabilitation par l'organisme émetteur ;

- Mise en ouvre d'algorithmes cryptographiques (par exemple DES ou RSA) pour assurer la sécurité des transferts de données.

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 crypto processeur comme dans certaines carte de décryptage télévision récentes.

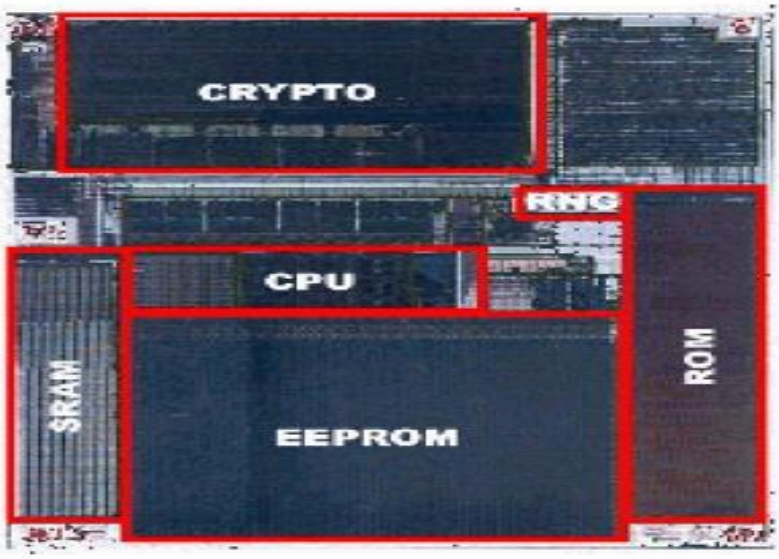


Fig. 2. 8: Un microprocesseur de carte à puce

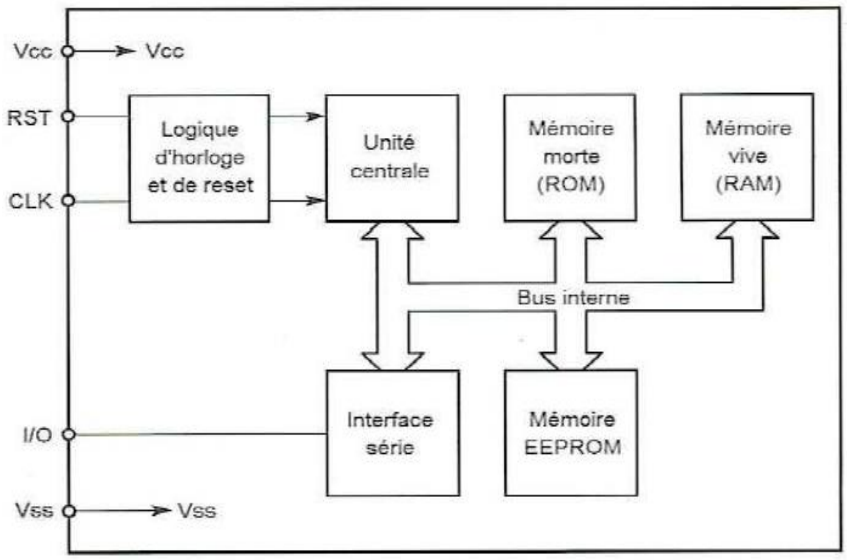


Fig. 2. 9: Synoptique interne d'une carte à microcontrôleur [10].

Ces cartes conviennent aux applications les plus « sensibles en sécurité» : carte bancaire, carte santé, carte de TV à péage, contrôle d'accès sécurisé, mono-service ou multiservices. On distinguer trois catégories différentes de cartes asynchrone :

b.1. Les cartes à puce "vierges" ou "spécifiques"

La mémoire de programme du microcontrôleur ne contient rien lorsqu'on les achète. C'est donc à nous 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. [10] [13]

b.2. Les cartes à puce dites "personnalisables"

La plupart des cartes à microprocesseur sont supportées par un puissant système d'exploitation appelé, par exemple, « COS » *(Chip Operating System* ou *Card Operating System).* [18] [19]

Il s'agit d'un logiciel « masqué » en ROM et dont les caractéristiques sont les suivantes :

1 Organisation de la mémoire en zone banalisées, protégées ou non

2 Gestion dynamique de l'espace mémoire

3 Gestion des codes confidentiels

4 Changement éventuel de sous-programmes spécifiques à l'application, lors de la personnalisation

Ces cartes permettent de développer très rapidement une application quasiment sans programmer.

b.3. Les cartes à puce dites "à OS ouvert"

C'est en quelque sorte une carte à puce "non terminée", c'est à dire une carte dans laquelle vous allez pouvoir programmer votre propre interpréteur de commande, et donc votre propre jeu d'instructions et vos propres mécanismes de gestion de fichiers. Dans ces cartes, le fabricant à programmé un interpréteur de P-code ; ce P- code provenant lui-même de la compilation de langage évolué tel que Java dans la Java Card ou Basic dans la Basic Card®. Comme pour les cartes à puce vierges, on peut ainsi réaliser une application "sur mesure" mais avec une programmation plus facile car elle est réalisée en langage évolué et non en langage machine. Hn outre, l'interpréteur de P-code prend en charge tous les protocoles de

dialogue de bas niveau que l'on n'a pas ainsi à programmer. Tout ceci est synthétisé visuellement sur la figure ci-dessous dans le cas d'une carte à puce de ce type. [10]

Fig. 2. 10: Principe des cartes à puce dites "à OS ouvert" (exp. Protiva smart Card) [10].

Une telle approche offre de multiples avantages que l'on peut résumer de la manière suivante :

- La carte est réellement programmable en fonction des besoins de l'application, comme lors d'une programmation par masque.
- La carte se programme en langage évolué, Java, C, Basic, accessible au commun des mortels et surtout indépendant du processeur contenu dans la carte.
- Les contraintes de programmation en grande série, imposées par la programmation par masque, n'existent plus car les cartes de ce type se programment unitairement si nécessaire.
- L'application ainsi développée est portable puisqu'elle peut être exécutée, du moins en théorie, dans toute carte compatible du langage utilisé.

2.6.2. Cartes sans contacts

Le principe « à contact » de la carte à puce traditionnelle constitue un handicap, et parfois même un vice rédhibitoire, pour bon nombre d'applications potentielles. Outre les problèmes de fiabilité que cela peut poser (pollution et usure des contacts, vulnérabilité des lecteurs), ce principe exclut d'emblée de nombreuses familles d'applications. Débarrassée de ses contactes, la carte à puce peut gagner énormément en robustesse, et donc affronter les pires environnements^ intempéries, eau de mer, hautes températures, produits chimiques... C'est particulièrement appréciable en milieu industriel et en extérieur (applications automobiles, marines ou sportives, par exemple). Ce lecteur pouvant fort bien télé-alimenter la « puce » par induction, l'absence de pile dans le badge constitue un avantage déterminant sur le plan de la fiabilité et de la compacité. Mais la technique à mettre en œuvre est délicate

malgré la simplicité de son principe. Des procédés de modulation et de supervision forte élaborés sont nécessaires pour garantir le très haut degré de sûreté que l'on exige dans les applications visées. [14]

2.7. Caractéristiques physiques et électriques des cartes à puce

2.7.1. Format de la carte

La partie 1 du standard ISO7816 définit les formats universels et les caractéristiques mécaniques du support en plastique de la carte à puce. La norme présente un format principal (id 1) de la carte à puce comme décrit dans la Figure 2.11 :

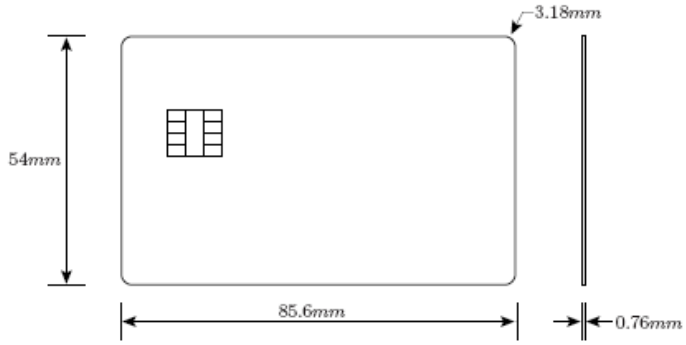


Fig.2. 11: Format de la carte à puce id 1 [11].

Le format (id 1) est celui utilisé pour les cartes bancaires, carte d'identité et autres. D'autres formats sont définis dans la norme et qui peuvent être déduites à partir du format principale (id 1) comme celle de la carte SIM ou la Mini SIM (id000), la carte Micro SIM et puis la Nano SIM (voir la Figure 2.12). [3]

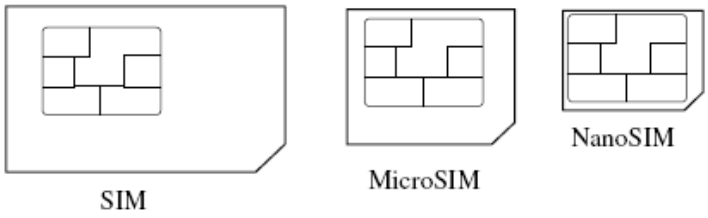


Fig.2. 12: Différents types de carte SIM.

D'autres caractéristiques physiques de la carte à puce sont imposées par la norme comme :

_ Elle doit être opaque aux rayons UV pour protéger les données écrites dans l'EEPROM.

_ Elle doit résister aux rayons X et donc l'architecture matérielle ne peut être dévoilée.

_ Elle doit résister aux détériorations de sa surface.

_ Elle doit protéger la puce lors de la manipulation de stockage et lors d'une utilisation normale.

_ La zone des contacts de la carte à puce à contact doit résister à la pression causée par une bille d'acier de 1.55 mm de diamètre appliquée avec une force intérieure inférieur à 1.5 N.
_ La carte ne doit pas être endommagée par un champ magnétique statique de 79500 $A.tr.m^{-1}$. [3]

2.7.2. La puce

La puce est considérée comme la partie la plus importante dans une carte à puce.
Elle est aussi le seul moyen de communication pour la carte à puce avec contact. La partie 2 du standard ISO7816 spécifie le dimensionnement physique (partie extérieur) et l'emplacement des contacts de la puce. La Figure 2.11 décrit la position des huit contacts (C1 _ C8).
Chaque producteur de carte crée un dessin propre à lui pour le module de contact, l'important est de respecter la spécification. La Figure 2.13 montre quelques exemples de module de carte à puce.

Le standard a défini un numéro et une fonction pour chacun des huit contacts existants dans le module. La Figure 2.8 présente l'ordre et la fonction de chaque contact. Le C1 est utilisé pour l'alimentation (Vcc). Le C2 est utilisé pour l'entré d'initialisation (RST). Le C3 est utilisé pour l'entrée d'horloge (CLK). Le C5 est utilisé pour la référence électrique (GND). Le C6 a été utilisé dans les années 80 pour l'alimentation utile pour la programmation de l'EEPROM, ceci n'est plus nécessaire
avec la technologie CMOS actuelle. Le C7 est utilisé pour la communication série

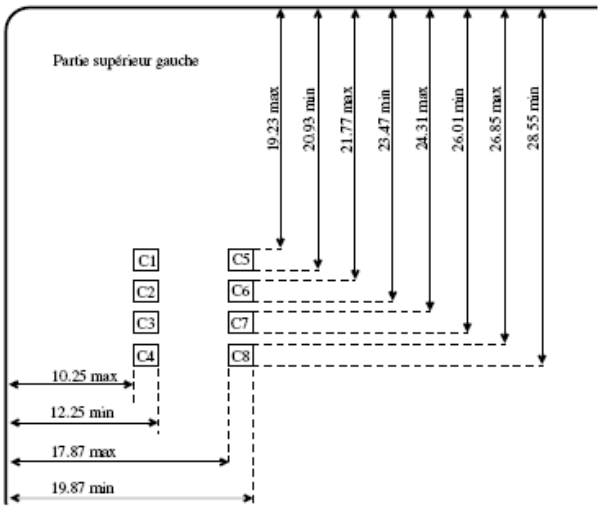


Fig.2. 13: Emplacements des huit contacts (valeurs en mm) [11].

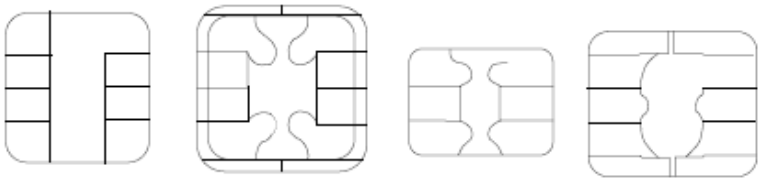


Fig.2. 14: Types de module de cartes à puce [11].

en entrée et en sortie (I/O). Les deux contacts C4 et C8 ont été laissés pour une fonction future (FF). L'une des utilisations actuellement proposées est la fonction USB, sachant que cette fonction a besoin de quatre contacts (Vcc, GND, D+, D-), le Vcc et le GND sont déjà affectés à C1 et C5 et le D+, D- seront affectés à C4 et C8. Une carte à puce avec connexion USB peut avoir un débit de transfert plus élevé atteignant le 12 Mbits/seconde [11].

2.7.3. Architecture de la carte à puce

Dans la figure 2.15 les cartes sont divisées en deux familles : les cartes à mémoire et les cartes à microprocesseur. Dans cette partie, l'architecture de chacune de ces familles est décrite selon un esprit de fonctionnement mais ne représente pas une architecture de carte bien précise. [11]

C1	C5
C2	C6
C3	C7
C4	C8

C1	C5
C2	C6
C3	C7

(a) Numéros des contacts (cartes à 8 et 6 contacts)

Vcc	GND
Rst	Vpp
CLK	I/O
FF	FF

Vcc	GND
Rst	Vpp
CLK	I/O

(b) Fonctions des contacts (carte à 8 et 6 contacts)

(c) **Fig.2. 15:** Numéros et fonctions des contacts du module de la carte à puce [11].

2.7.4. Brochage des cartes à puce

La carte possède 8 contacts et parmi eux on trouve 2 réservés à une future utilisation. Les différentes connections sont présentées dans la figure 2.16. Les fonctionnalités de ces connections sont données dans le tableau 2.3.

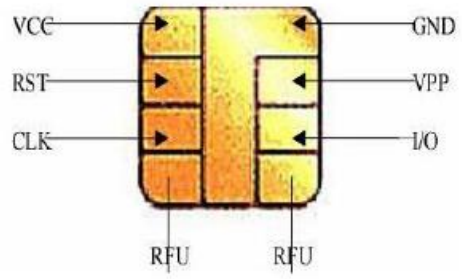


Fig.2. 16: Brochage des contacts d'une carte à puce selon la norme ISO 7816-3.

Position	Abréviation technique	Opération
C1	Vcc	Tension d'alimentation
C2	MCLR	Remise à zéro
C3	CLK	Fréquence d'horloge (Entrée)
C4, C8	RFU	Réservé à une Future Utilisation
C5	Vss	Masse
C6	Vpp	Tension de programmation de l'EPROM
C7	I/O	Données série (Entrée / Sortie)

Tab.2. 3: Fonctionnalités des contacts de la carte à puce Wafer 2 [15] [14].

2.7.5. Caractéristiques électriques

Vcc : 4.75 ≤Vcc≤5.25v

RST : valeur min = 4V ou Vcc-0.7V

CLK : Min=2.4 ou 0.7Vcc ou encore Vcc-0.7V

Max= Vcc

E/S : entrée : Min= 2V ou 0.7 Vcc

Max= Vcc

Sortie : Min=2.4V ou 3.8V

Max = Vcc. [2,10]

2.8. SECURITE DES CARTES A PUCE

Tout système ou produit issu des Technologies de l'Information peut être le sujet d'une procédure de certification, menée par un organisme habilité, à l'issue de la quelle un certificat (selon une norme bien définie) est délivrée. On dit alors que le produit est conforme à la

norme considérée. Ce processus de certification peut être appliqué pour une carte à puce. L'émetteur de la carte ou son utilisateur final peuvent alors s'appuyer sur un tel certificat afin de renforcer le degré de confiance qu'ils auront dans la carte. Le processus de certification consiste en une procédure d'évaluation sécuritaire menée par des organismes reconnus et réalisée dans un cadre officiel. Il s'agit d'une procédure relativement complexe dont l'objectif est d'établir le niveau de confiance que pourra avoir un utilisateur final en la sécurité d'un produit. Le produit doit donc être conforme à une norme donnée spécifiée et satisfait l'ensemble des règles de sécurité définies par cette norme.

2.9. Lecteurs de cartes à puce

On appelle « lecteurs » de cartes à puce des appareils souvent capable aussi bien de lire que d'écrire dans celles-ci.

Le terme le plus approprié est par conséquent « lecteur-encodeur ».

2. 9.1. Les interfaces matérielles des lecteurs

Un lecteur de cartes à puce est la réunion dans un boitier ou sur un simple circuit imprimé d'un : - Coupleur ;

- Connecteur de cartes ;
- Système d'alimentation ;

- Circuit assurant la liaison avec un système informatique « hôte ».

Certains lecteurs pourront être autonomes, et donc dotés d'un afficheur et, éventuellement, d'un clavier complet ou simplement de quelques boutons. D'autres lecteurs se connectent à un PC, on y trouve deux types :

- Lecteurs à interface série.
- Lecteurs à interface USB.

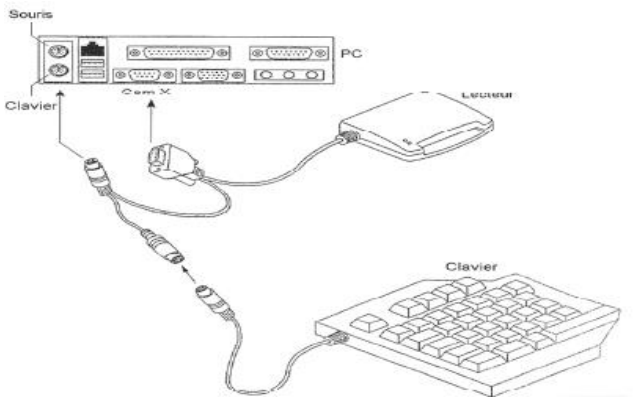


Fig.2. 17: La connexion d'un lecteur à interface avec un ordinateur.

Les lecteurs USB sont alimentés par l'intermédiaire du bus USB. Par contre les lecteurs à interface série utilisent une alimentation externe.
Des versions équipées d'un émetteur-récepteur radio permettent même de communiquer avec les cartes à puce sans contacts». [10], [14]

2.9.2. Le choix d'un lecteur

Lorsque l'on se préoccupe de développer une application carte à puce, le choix d'un lecteur est presque aussi important que celui de la carte elle-même. Le choix d'un lecteur de carte dépend de ce qu'on veut faire mais aussi et surtout des types de cartes qu'on souhaite lire ou programmer. [10], [16]
Si on utilise des cartes vierges telles que les cartes Gold, Silver, Fun ou Jupiter, il nous faut un programmateur, au moins pendant la phase de développement de notre application. C'est en effet ce programmateur qui va programmer le microcontrôleur de la carte et/ou sa mémoire EEPROM. Une fois cette carte programmée, et si nous avons écrit un programme compatible des normes ISO 7816 3 et 4, un lecteur classique pourra ensuite être utilisé pour lire et écrire dans notre carte.

Si on utilise des cartes personnalisables ou des cartes à puce à OS ouvert comme la Basic Card, un lecteur classique suffit.
Si nous ne voulons que lire dans une carte, que ce soit une carte bancaire, Vitale ou tout autre carte conforme aux normes ISO 7816 3 et 4, le lecteur classique convient aussi. [10], [16]

Fig.2. 18: Un programmateur classique en boîtier Multipro 2000.

Fig.2. 19: Le lecteur Cyber Mouse Le d'ACS

2.10. Insertion de la carte dans un lecteur

2. 10.1. Etapes d'activation de l'interface avec la carte

Une fois la carte placée dans le lecteur, les opérations suivantes sont déclenchées : [10], [17]

- Mise au niveau bas de l'entrée Reset ;
- Alimentation de la carte via son entrée Vcc ;
- Mise en mode réception de la ligne I/O du circuit d'interface du lecteur ;
- Mise au niveau repos de Vpp de la carte ;
- Génération d'une horloge stable sur l'entrée CLK de la carte ;
- Un reset est alors provoqué par le circuit d'interface.

Après la connexion de la carte et l'activation de ses contacts, un reset est alors provoqué par le circuit d'interface, la carte répond par une réponse au reset ou ATR. Ensuite, un dialogue entre la carte et l'application à lieu via le circuit d'interface. Et à la fin les contacts sont désactivés par le circuit d'interface, et à ce moment-là on peut retirer la carte.

2. 10.2. Désactivation de l'interface avec la carte

La désactivation normale a lieu lorsque la transaction en cours se termine et le terminal nous invite à retirer la carte. Avant l'affichage du message de retrait, il y a : [10], [17]

- Mise au niveau bas de l'entrée RST ;
- Mise au niveau bas de CLK ;
- Mise au niveau inactif de Vpp ;
- Mise au niveau inactif d'I/O ;
- Coupure de l'alimentation Vcc.

Retrait de la carte (retrait de la carte avant la fin de la transaction doit être prise en charge par L'application).

2.11. ATR (Answer To Reset)

Dès que la carte est mise sous tension, elle envoie un message de réponse d'initialisatic

Fig.2. 20: La réponde au RESET (ATR).

2.11.1. Caractéristiques de l'ATR

- L'ATR est la réponse de la carte au Reset du terminal ;
- L'ATR au minimum = 2octets, au maximum = 33 octets ;
- Transmission en mode asynchrone semi-duplex ;
- La fréquence d'horloge comprise entre 1 et 5 MHz pour permettre à n'importe quel lecteur de lire le 1er caractère ;
- Communication entre le lecteur et la carte via la ligne bidirectionnelle I/O. [10]

2.11.2. Chronogramme de la réponse au Reset

- Commande Asynchrone (protocole RS232) : bit start + 8bits de données+ bit de parité paire + temps de garde (un ou plus bits Stop) (voir figure 2. 21) ;
- L : niveau bas et H : niveau haut ;
- Le délai entre 2 caractères est au moins de 12 etu et TG = 2 etu. [13]

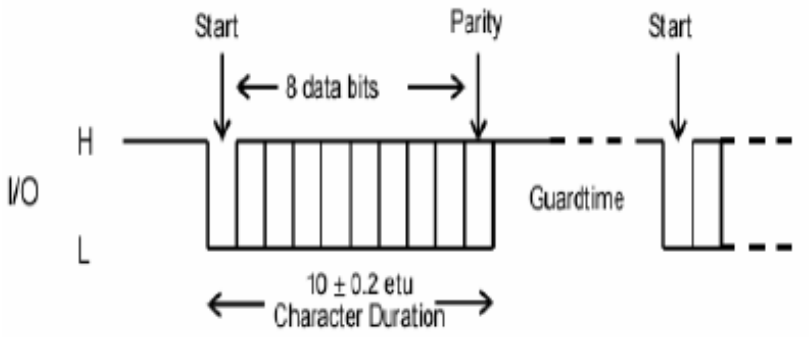


Fig.2. 21: Chronogramme d'échange d'un octet entre une carte et un lecteur

2.11.2.1. Caractère initial de l'ATR

- Premier caractère de l'ATR = TS (caractère initial)
- TS peut prendre 2 valeurs : (HHLLLLLL)2 ou (LLHHHHHH)2

1 : Convention inverse : TS=3F (en héxa)

Niveau bas L = « un » logique ;

Niveau haut H = « zéro » logique ;

Bit transmis en premier = bit 7 de poids fort ;

Bit transmis en dernier = bit 0 de poids faible ;

2 : convention directe : TS=3B (en héxa)

Niveau bas L = « 0 » logique ;

Niveau haut H = « 1 » logique ;

Bit transmis en premier = bit 0 de poids faible ;

Bit transmis en dernier = bit 7 de poids fort ;

2.11.2.2. Caractère T0

Appelé aussi caractère de format, c'est le 2ème caractère de l'ATR dont la présence est également obligatoire. [10]

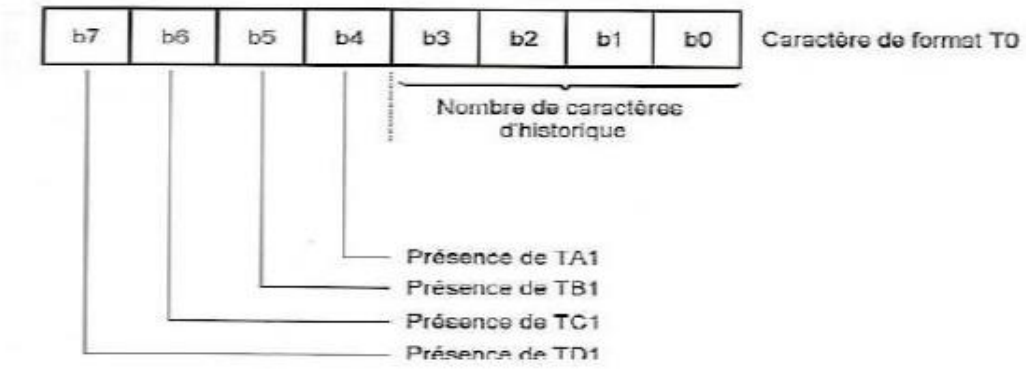


Fig.2. 22: Contenu du caractère de format de l'ATR [10].

La figure 2.22 présente le contenu du caractère T0. La partie basse, appelée K dans la norme, (b0 à b3) code le nombre de caractères d'historique (il ne peut y avoir plus que 15 caractères d'historique).

La partie hausse, appelée Y1 dans la norme, (b4 à b7) dont chaque bit indique la présence ou l'absence d'un des caractères TA1, TB1, TC1 et TD1.

La présence de TA1 et TD1 est codée dans T0 tandis que la présence des caractères TAi à TDi avec (i>1) est codée dans le caractère TD précédent.

La figure 2.23 donne l'allure générale de l'ATR. [10]

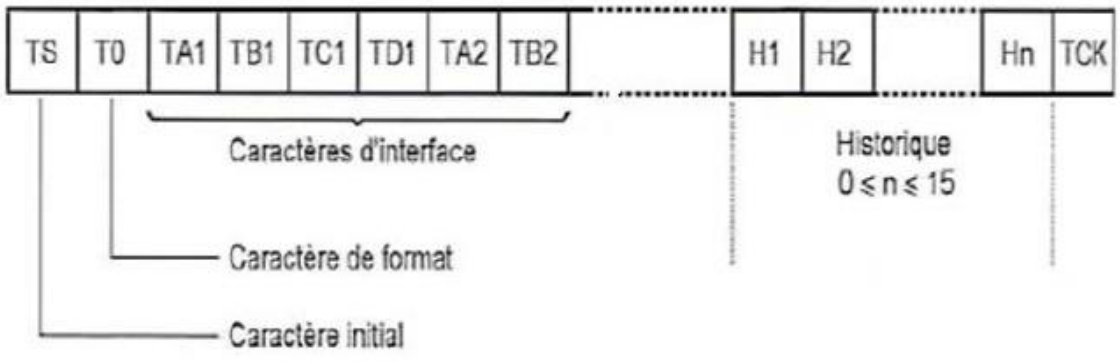


Fig.2. 23: Allure générale de l'ATR [10].

2.11.2.3. Caractère TA1

Permet de définir la vitesse de transmission utilisée après la phase de réponse au RESET.

2.11.2.4. Caractère TB1

L'utilisation de ce caractère est de plus en plus rare, il sert à coder la valeur de la haute tension de programmation Vpp ainsi que le courant nécessaire.

2.11.2.5. Caractère TC1

Code un paramètre appelé N qui est un temps de garde supplémentaire (le temps de garde est le temps qui s'écoule entre la fin d'un caractère et le début du suivant).

Ce temps de garde supplémentaire ne doit exister que lors des dialogues qui ont lieu dans le sens lecteur vers carte. Dans le sens carte vars lecteur ce temps supplémentaire n'existe pas.

2.11.2.6. Caractère TD1

Outre le fait qu'il code la présence éventuelle (et peu fréquente) des autres caractères TAi, TBi, TCi et TDi, code également sur ses 4bits de poids faible le numéro du protocole utilisé (voir la figure 2.24).

Présence de TA_2, TB_2, TC_2, TD_2 — protocole

	b8	b7	b6	b5	b4	b3	b2	b1	
T=1	1	0	0	0	0	0	0	1	=(81)

Fig.2. 24: Contenu du caractère TD1 de l'ATR

2.11.2.7. Caractères d'historique

Leurs présence est facultative, cas caractères ne sont pas définis dans une norme et leurs signification est laissée à l'appréciation du fabricant de la carte.

2.11.2.8. Caractère TCK

Le caractère TCK ne doit être présent que si un protocole différent de zéro a été spécifié au moyen de l'octet TD1.C'est un caractère de contrôle qui est calculé de telle façon que le OU exclusif entre tous les octets compris entre T0 (inclus) et TCK lui-même (également inclus) soit nul.

2.12. Les protocoles TPDU/APDU [10],[17],[18]

- TPDU: Transmission Protocol Data Unit.

- APDU : Application Protocol Data Unit**.** [10], [18], [17]

2.12.1. Protocoles TPDU

Il existe deux protocoles T=0 et T=1, le T=0 est le plus utilisé.

- **Protocole T=0 :** est de type caractère, son mode de fonctionnement est de type commande/réponse. Le terminal est l'initiateur des échanges.

- **Protocole T=1** : est un protocole ambitieux très peu utilisé, il est plus proche du modèle OSI car les échanges s'effectue en blocs structurés. [15]

2.12.2. Protocoles APDU

La communication entre l'hôte et la carte est half-duplex. Elle se fait à l'aide de paquets appelés APDU.

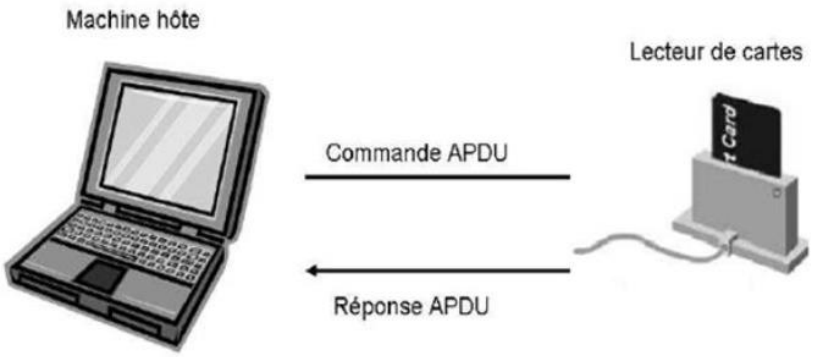


Fig.2. 25: Le modèle de communication de la carte à puce.

Un APDU contient une commande ou une réponse (voir la figure 2.25).

Le mode Maître/Esclave est utilisé. Ainsi la carte joue un rôle passif et attend une commande APDU à partir de l'hôte. Elle exécute l'instruction spécifiée dans la commande et retourne une réponse APDU. [10], [18]

2.12.3. Format des commandes/réponses APDU

Entête obligatoire				**Corps optionnel**		
CLA	INS	P1	P1	Lc	données	Le
1 octet	1 octet	1 octet	1 octet	1 octet	Non définie	Non définie

Tab.2. 4 : Format des commandes APDU [15].

CLA : classe ;

INS : code de l'instruction ;

P1, P2 : paramètres de l'instruction ;

Lc : nombre d'octet présents dans le champ de données ;

Données : données à envoyer vers la carte ;

Le : nombre d'octet à recevoir de la carte ; [10], [18]

Corps optionnel	partie obligatoire	
données	SW1	SW2
varie	1 octet	1 octet

Tab.2. 5 : Format des réponses APDU [15].

SW1, SW2 : Status Words (mots d'état) donnent l'état de traitement par la carte ;

Données : données reçues de la carte.

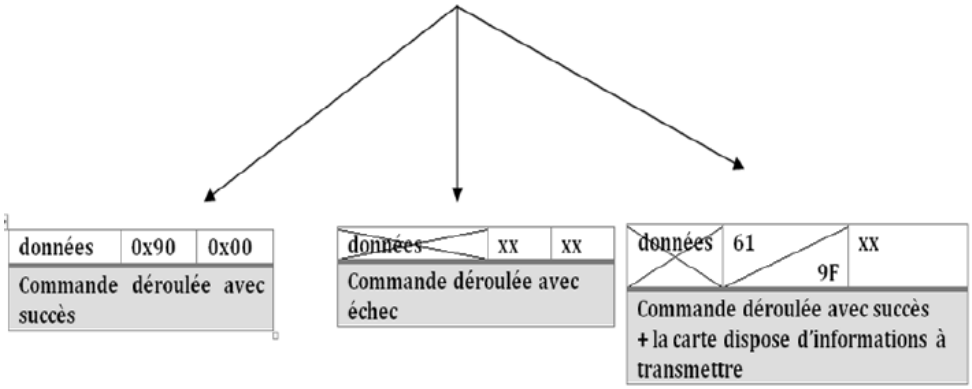


Fig.2. 26: Le Les différentes réponses APDU de la carte.

2.12.4. Commandes APDU

Il existe 5 types de commandes APDU selon qu'il y'a ou non échange de données utiles.

2.12.4.1. Envoi d'une commande sans données utiles

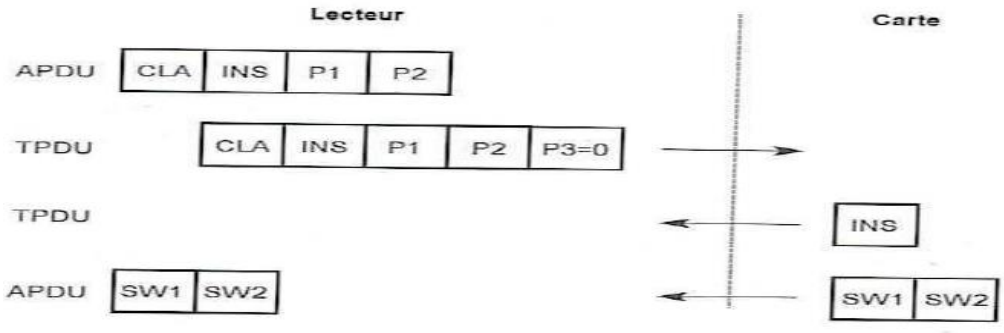


Fig.2. 27: Envoi d'une commande sans données utiles [10].

2.12.4.2. Commande avec réception de données utiles depuis la carte

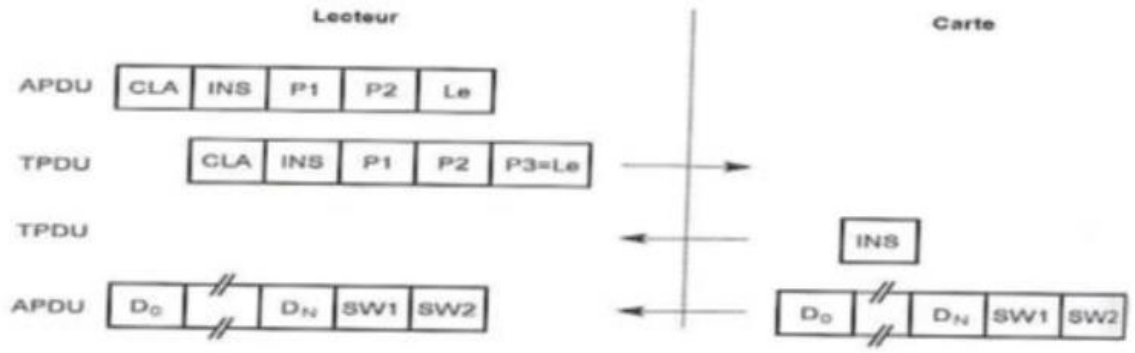


Fig.2. 28: Commande avec réception [10].

2.12.4.3. Commande avec invitation à lire des données depuis la carte

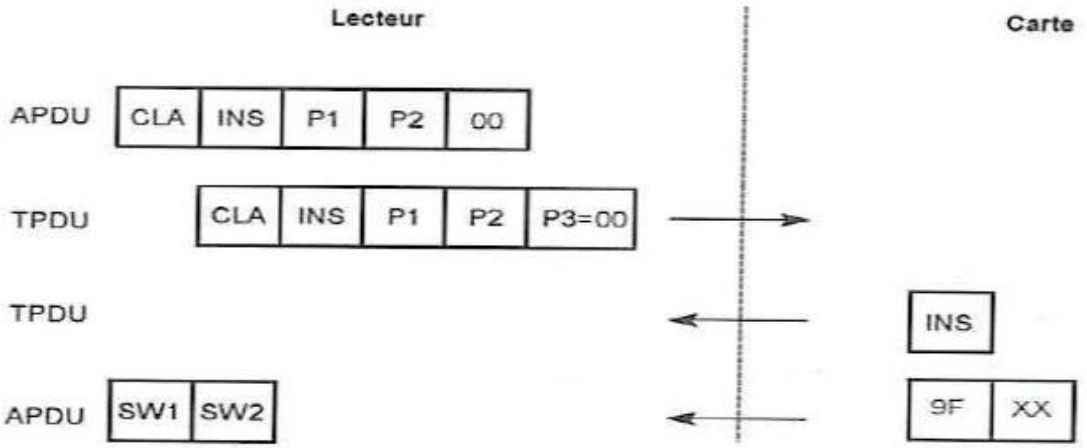


Fig.2. 29: Commande avec invitation à lire des données depuis la carte [10].

2.12.4.4. Commande avec envoi de données utiles vers la carte

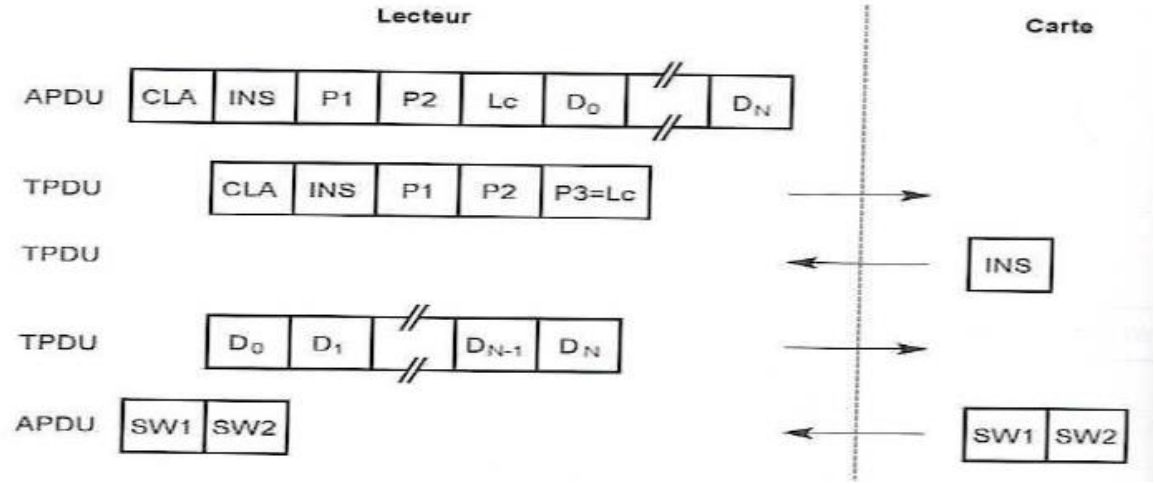


Fig.2. 30: Commande avec envoi de donnée utiles vers la carte [10].

2.12.4.5.Commande avec envoi et réception de données utiles

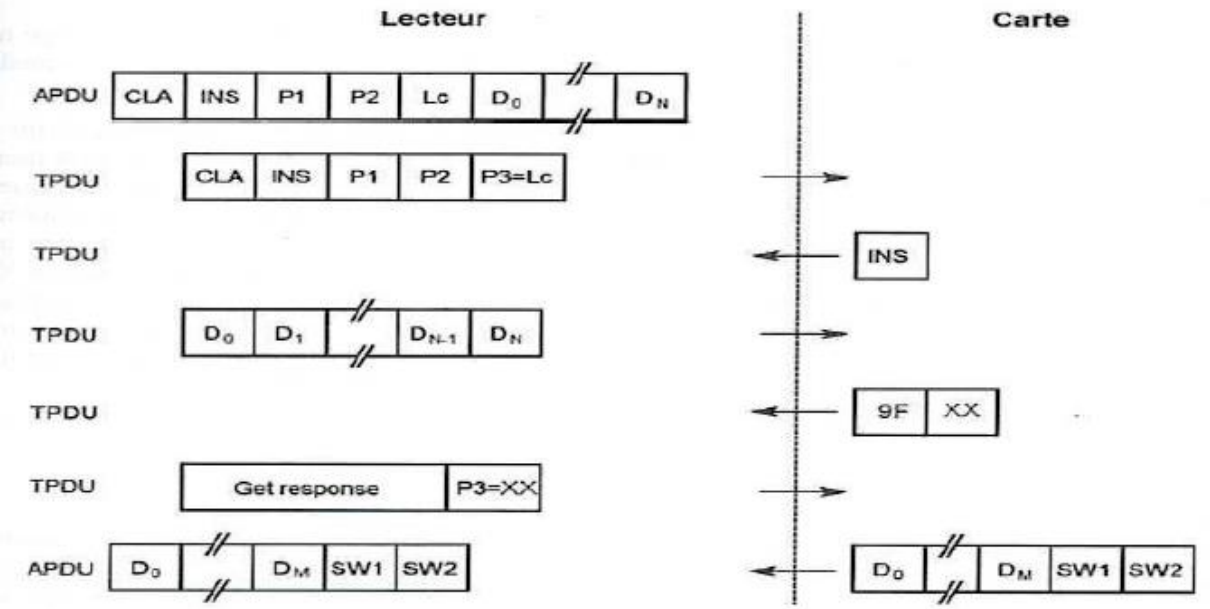


Fig.2. 31: Commande avec envoi et réception de données utiles [10].

2.13. Conclusion

L'évolution de la technologie a permis la conception de cartes à puce de plus en plus performantes et sécurisées ainsi, l'implémentation d'applications est devenue plus rapide mais leurs coûts augmentent aussi.

Le choix de la carte à puce qui convient le mieux à une application donnée est devenu une tâche difficile, en particulier pour celles destinées à un grand nombre d'utilisateurs. Il faut réduire les coûts tout en assurant le bon fonctionnement. Il serait donc crucial pour les développeurs d'applications de s'interroger sur les performances des cartes à puce utilisées tout en réduisant les couts.

Dans le chapitre suivant, nous introduisons notre contribution par la description d'une nouvelle machine de vote électronique embarquée à base d'un microcontrôleur et utilisant une carte à puce.

CHAPITRE 3

Description du système à concevoir

3.1. Introduction

Après l'étude des différents types de systèmes de vote électronique, nous avons amélioré celui qui se fait dans les bureaux car c'est le type le plus répondu dans le monde.

Nous avons ajouté une authentification à base de carte à puce pour mieux sécurisé l'identification de l'administrateur de la machine.

Nous commencerons par la description du nouveau système toute en justifiant le choix des composants et en fin nous expliquerons déroulement du vote.

3.2. Description du système

La conception de circuit est divisée en deux parties, la carte à puce et la machine illustré dans la figure 3. 1.

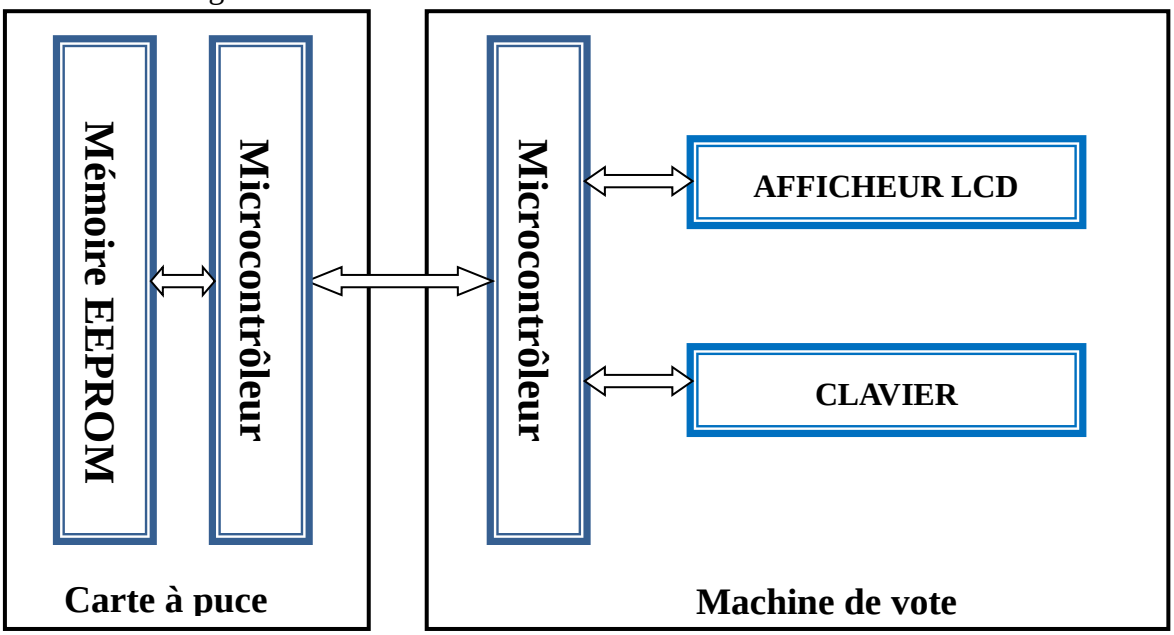


Fig. 3. 1: Le bloc diagramme du système.

La carte a puce se compose de microcontrôleur 16F877A qui assure le dialogue avec la machine et une mémoire EEPROM pour stocker les données de la machine comprend également un microcontrôleur qui représente la composante principale permettant la communication avec la carte à puce, la machine communique avec l'utilisateur en utilisant le clavier 12 touches et l'écran LCD 16*2(deux lignes de 16 caractères).

3.3. Partie carte à puce

3.3.1. Choix de la carte

Une carte à puce wafer 3 est plus sûre et contient beaucoup plus de mémoire qu'une carte à bande magnétique, comprenant une puce à circuit intégré. Sa taille physique est décrite dans la norme ISO 7816, le placement puce de la carte a été défini dans norme ISO 7816-2.

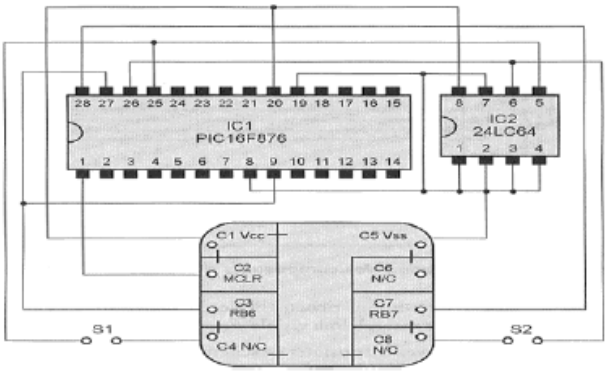


Fig. 3. 2: Architecteur de carte a puce Wafer 3.

3.3.2. Principe du fonctionnement

Dans ce système, on a organigramme de la carte à puce

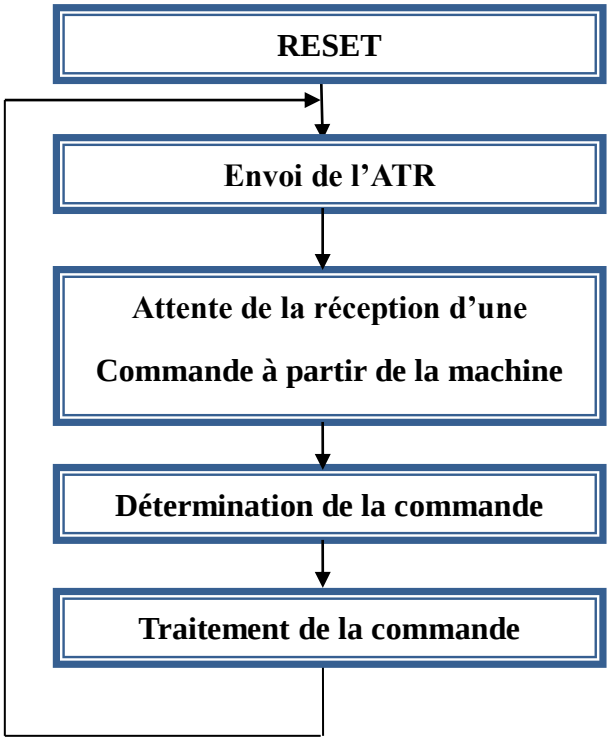


Fig. 3. 3: Organigramme de la carte à puce.

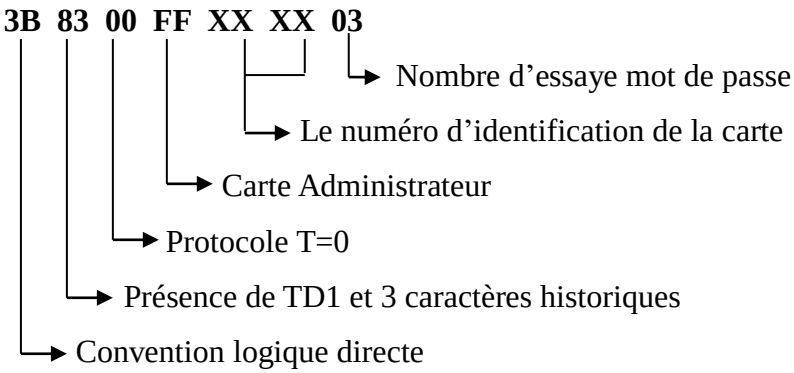
Une fois que la carte à puce est insérée dans le lecteur (machine de vote électronique) elle est remise à zéro ensuite elle envoie l'ATR (Answer To Reset) vers son lecteur pour échanger les

informations nécessaires afin d'assurer la communication. Par la suite, la carte passe à l'attente de la réception d'une commande à partir du lecteur et dès qu'elle reçoit une, elle passe à la détermination de la commande ensuite son traitement.

3.3.3. ATRs des cartes a puce utilisées [21][22]

L'ATR (réponse à réinitialiser) est la première communication qu'une carte à puce envoie après la détection d'une réinitialisation. Entre autres, l'ATR fournit le terminal des informations sur les protocoles de transmission et les taux de transmission de données pris en charge par la carte à puce. L'ATR est toujours transmis avec une valeur de diviseur de 372, ce qui donne un débit de données de transmission de 9 600 bps avec une fréquence d'horloge de 3,5712 MHz.

- **Carte administrateur**



3..4. Commandes utilisées [21]. [22]

La carte à puce communique avec la machine en utilisant commandes APDU.

La commande VERIFY : utilise 4 caractères pour la vérification d'un mot de passe. Le format de la commande VERIFY est présenté dans le tableau 3.1.

Commande APDU					
CLA	**INS**	**P1**	**P2**	**LC**	**Données**
00	**20**	**00**	**00**	**04**	**XX XX XX XX**

Tab.3. 1: Format de la commande VERIFY.

XX : 1 octet du mot de passe.

	Réponse APDU	
Etat	**SW1**	**SW2**
Succès	**90**	**00**
Erreur	**98**	**40**

Tab.3. 2: Format de réponse de la commande VERIFY.

3.4. Partie machine de vote électronique

Quand la machine est allumée, elle va attendre l'insertion d'une carte à puce, une fois que la carte est insérée elle va envoyer l'ART à la machine qui va vérifier si c'est la bonne carte a puce et si elle n'est pas bloquée et elle détermine si il s'agit d'une carte administrateur.
La machine va authentifier l'utilisateur qui va utiliser un mot de passe avec 4 lettres ou caractères, la vérification se fera avec la commande VERIFY, le mot de passe va être inséré avec un clavier à 12 touches, si le mot de passe est correcte l'utilisateur va avoir accès au système si non le nombre de fois où il aura le droit à l'essai de mot de passe va être décrémenté, sachant que le nombre de fois autorisé est 3 après ça la carte va être bloquée.

Pour une carte administrateur la machine va afficher sur le LCD les résultats de vote et elle va procéder à une authentification par un mot de passe pour permettre la récupération des résultats. La machine va inviter l'électeur à entrer le numéro de candidas sur qui il souhaite voté, ce système offre un chois entre deux candidas, avant de validé la machine va demander a l'électeur de confirmer son chois en utilisant la touche «# »ou soutire en utilisant la touche «*».

Une fois que le vote est confirmé la machine va incrémenter le nombre de votant pour le candidat voté et c'est pareille pour tous les l'autre candida on enregistrant les résultats dans la mémoire EEPROM interne de microcontrôleur pour plus de sécurité.
Lorsque la machine est en marche, elle va atendre la réception de la carte à puce. Une fois que la carte est incéré.

3.4.1. Programmation de la machine de vote électronique

3. 4. 1.1. Organigramme

La machine authentifie les utilisateurs en utilisant un mot de passe de 4 caractères. Cela en utilisant un clavier de 12 touches. Si le mot de passe est correcte nous allons avoir accès au système, sinon le nombre d'essai du mot de passe sera décrémenté ; sachant que le nombre d'essai maximum est 3 et après la carte sera bloquée.

Pour une carte administrateur, la machine affiche les résultats de l'élection. La machine invite l'électeur à voter en insérant le numéro qui correspond à son candidat. (Voir l'organigramme de la figure 3.4).

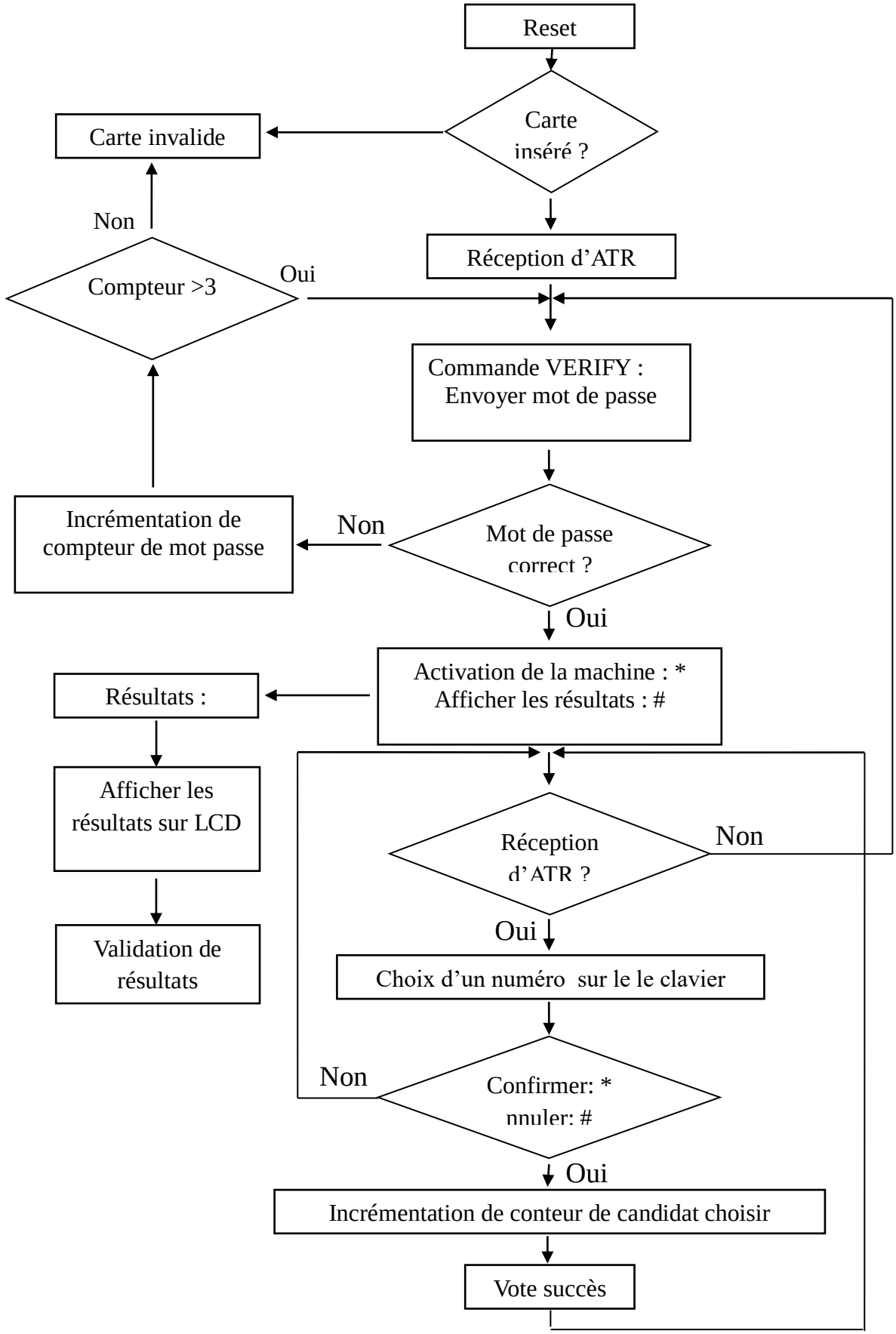


Fig. 3. 4: Organigramme de la machine.

3.5. Circuit du système

Afin de vérifier le fonctionnement de la machine de vote électronique, on a réalisé son schéma sous PROTEUS comme suit :

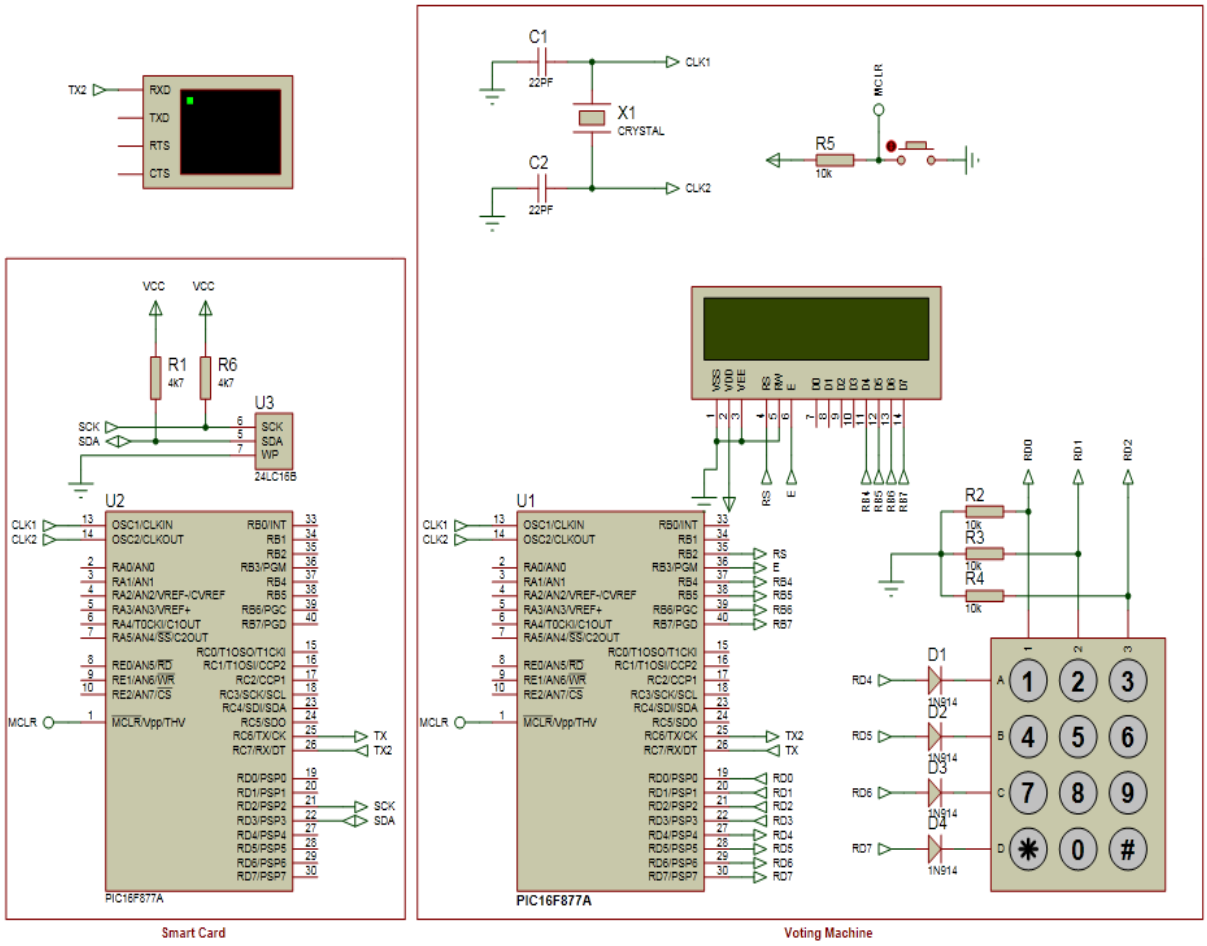


Fig. 3. 5: Circuit du système de vote électronique conçu.

3.5.1. Carte à puce

La carte à puce contient un microcontrôleur PIC 16F877A et la mémoire EEPROM 24LC16.

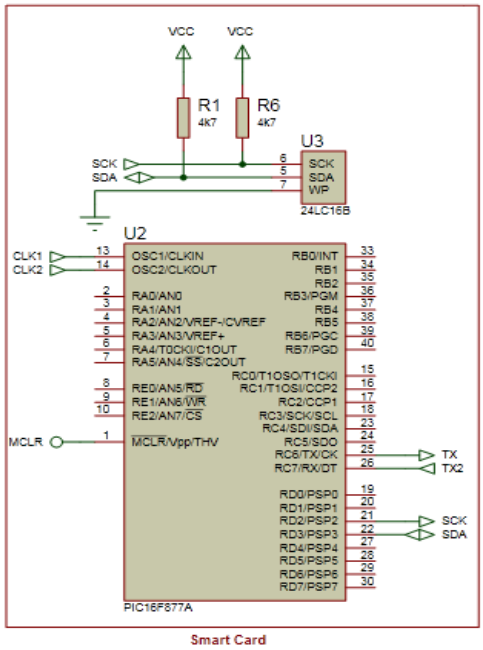


Fig. 3. 6: La carte à puce utilisée.

3.5.2. Machine de vote

La machine contient un microcontrôleur 16F877A, et un clavier de 12 touches et d'un écran LCD avec deux lignes de seize caractères.

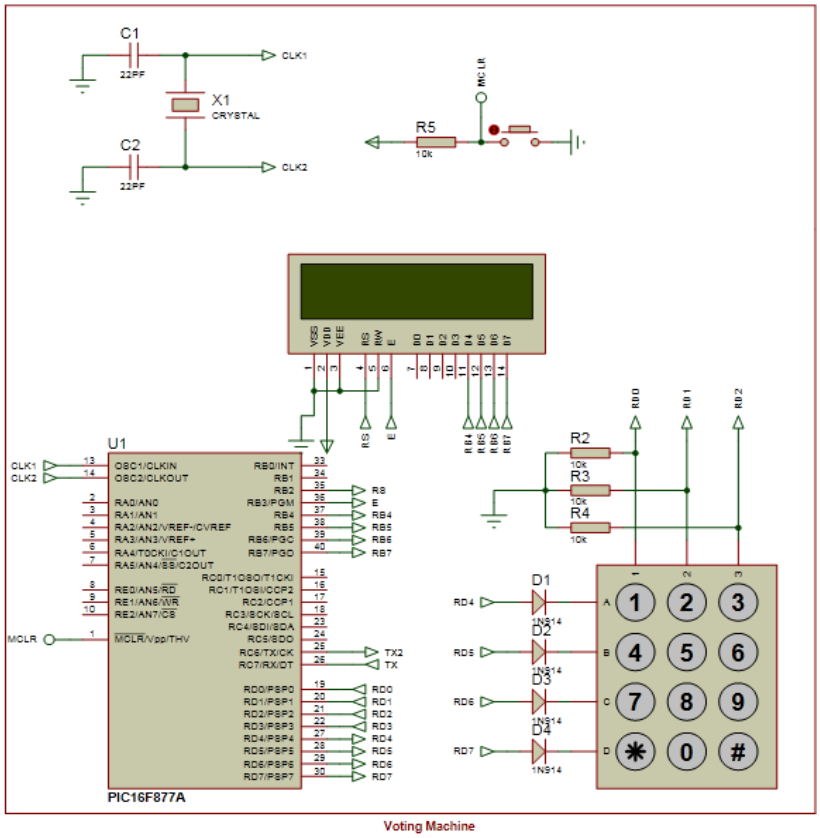


Fig. 3. 7: La machine de vote électronique.

3.6. Outils utilisé

3.6.1. Source de courant

Microcontrôleur fonctionne sur 5V volts alimentation WRT sol.

Le transformateur abaisseur quitte l'alimentation en courant alternatif du secteur et le pont redresseur convertit en courant continu. Ce courant continu est adopté à travers le circuit de filtre pour obtenir une forme d'onde plus lissée. Régulateur de tension produit une tension de sortie qui reste fixe constante indépendamment des variations de la tension d'entrée et de la charge. Régulateur de tension 7805 est utilisé ici, ce qui donne +5 V sortie qui est alors donnée au microcontrôleur pour son opération.

3.6.2. Microcontrôleur PIC16F877A

Le microcontrôleur est un dérivé du microprocesseur. Sa structure est celle des systèmes à base de microprocesseurs. Il est donc composé en plus de l'unité centrale de traitement, d'une mémoire (mémoire vive RAM et mémoire morte ROM), une (ou plusieurs) interface de communication avec l'extérieur matérialisé par les ports d'entrée/sortie.

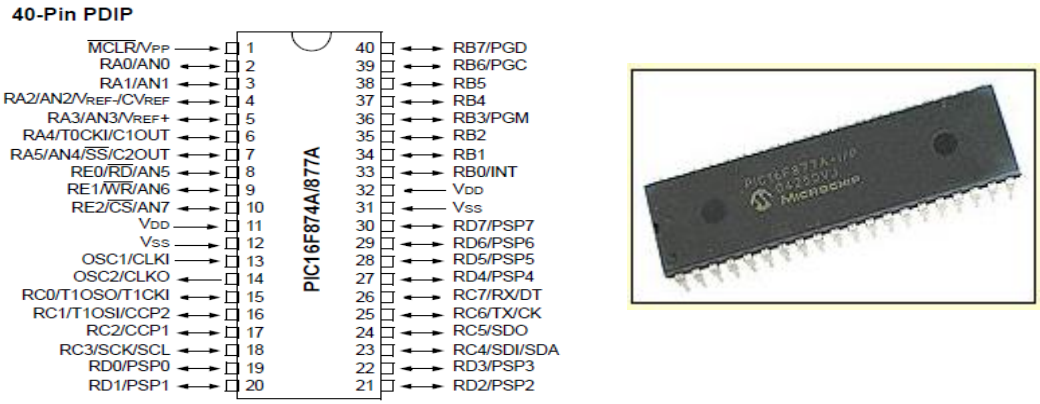


Fig. 3. 8: Microcontrôleur PIC16F877A.[7]

3.6.3. Afficheur LCD (Liquid Crystal Display)

C'est un affichage matriciel, il affiche des caractères alphanumériques et des symboles. Les écrans à cristaux liquides sont utilisés dans des dispositifs de Batterie alimentés, tels que les montres numériques, calculatrices, thermomètres numériques etc. Un LCD 16X2 a été utilisé dans le système modélisé pour afficher les informations du candidat et les résultats.

Fig. 3. 9 : Afficheur LCD (16X2).

3.6.4. Le clavier matriciel

Ce clavier comprend 12 touches disposées en 4 lignes et 4 colonnes. L'appui sur une touche fait communiquer une ligne avec une colonne.

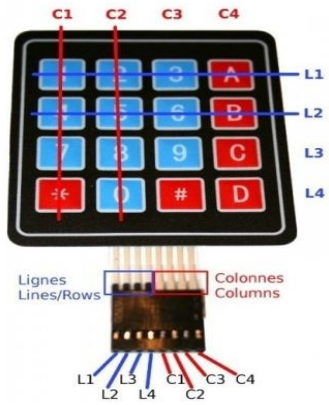


Fig. 3. 10: clavier matriciel (4X3).

Les lignes sont des sorties. Les colonnes sont des entrées maintenues au niveau haut par une résistance interne à microcontrôleur. Le système envoie par balayage un niveau bas sur chaque ligne (1 seule à la fois) et balaye les colonnes en lecture.
Quand il lit un niveau bas, c'est que la colonne est reliée par une touche appuyée à la ligne qui est basse à ce moment.

3.6.5. Cristal Oscillateur

Il est un circuit oscillateur qui le déploie propriété de résonance mécanique du piézo-électrique cristal de création d'un signal électrique précis. La fréquence du cristal en gardant une trace du temps, fournit le signal d'horloge pour le microcontrôleur. Quartz fréquence de 4 MHz est utilisé dans le système proposé. Quartz cristaux sont utilisés dans les montres, calculatrices, compteurs, générateurs de signaux, et oscilloscopes.

Fig. 3. 11: Cristal Oscillateur.

3.6.6. Un virtuel terminal

Un terminal virtuel est utilisé pour afficher les données échangées entre la carte et la machine comme indiqué dans la figure 12.

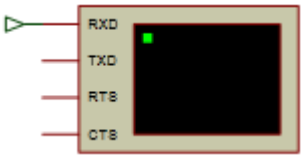


Fig. 3. 12: Un virtuel terminal

3.7. Déroulement de vote

La machine est installée dans un bureau de vote initialisée avec la liste des électeurs ayant droit.

La machine de vote ne doit être touchée ou manipulée que par un administrateur déclaré responsable du bureau. Pour assurer cela, nous proposons l'utilisation d'une authentification basée sur une carte à puce et un mot de passe pour plus de sécurité. Une fois l'administrateur présent au bureau, il insert sa carte à puce de vote dans la machine. La machine l'invite à entrer son mot de passe. Si le mot de passe est faux, la machine l'invite à le vérifier une autre fois (3 fois au maximum). Si le mot de passe est juste la machine est prête pour le vote.

Lorsqu'un électeur se présente et que son identité est vérifiée par un agent, il passe à la machine afin de faire son choix à l'aide d'un clavier. L'électeur est invité à entrer le numéro correspondant à son candidat. Afin d'éviter des erreurs de frappe, la machine lui demande de confirmer son choix. Si ce n'est pas le bon choix, l'électeur saisi à nouveau le numéro de son candidat et le confirme. Une fois le vote confirmé, la machine est remise à l'attente d'un nouvel électeur.

3.8. Conclusion

Dans ce chapitre nous avons présenté le système que nous avons amélioré en lui ajoutant une authentification à base de carte à puce pour l'administrateur ainsi que l'explication du de son déroulement dans le bureau.

Dans le chapitre suivant, nous aborderons la simulation et la réalisation de notre nouveau système.

CHAPITRE 4

Simulation et réalisation de la machine de vote électronique

4.1. Introduction

Notre machine de vote électronique utilise un Microcontrôleur et une carte à puce. La simulation se fait avec PROTEUS Professional V7.6. Les microcontrôleurs sont programmés en langage C sous micro PRO for PIC V 6.6.1 et utilisation de pickit 3 et pickit programmer 3 pour implantation du programme et vérification de son fonctionnement.

Ce chapitre va présenter les déférents scénarios qui peuvent ce produire lors d'utilisation de la machine en simulant son fonctionnement ainsi que la vérification pratique en réalisant le circuit sur une plaque d'essai.

4.2. Simulation

Quand l'administrateur inséré la carte à puce dans le lecteur, la machine vérifiée si la carte n'est pas bloquée. L'écran LCD affiche un message « Enter your password » Comme indiqué dans la figure suivante.

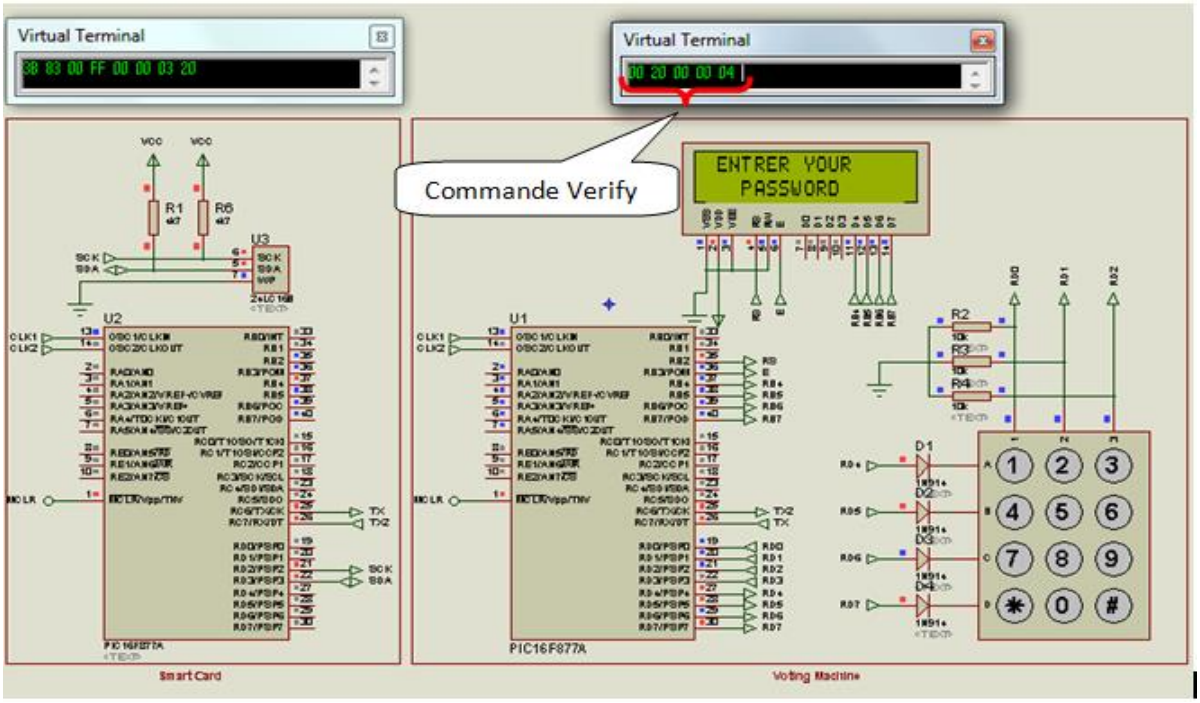


Fig.4.1 : la machine invite l'utilisateur à taper son mot de passe.

L'administrateur saisi son mot de passe.

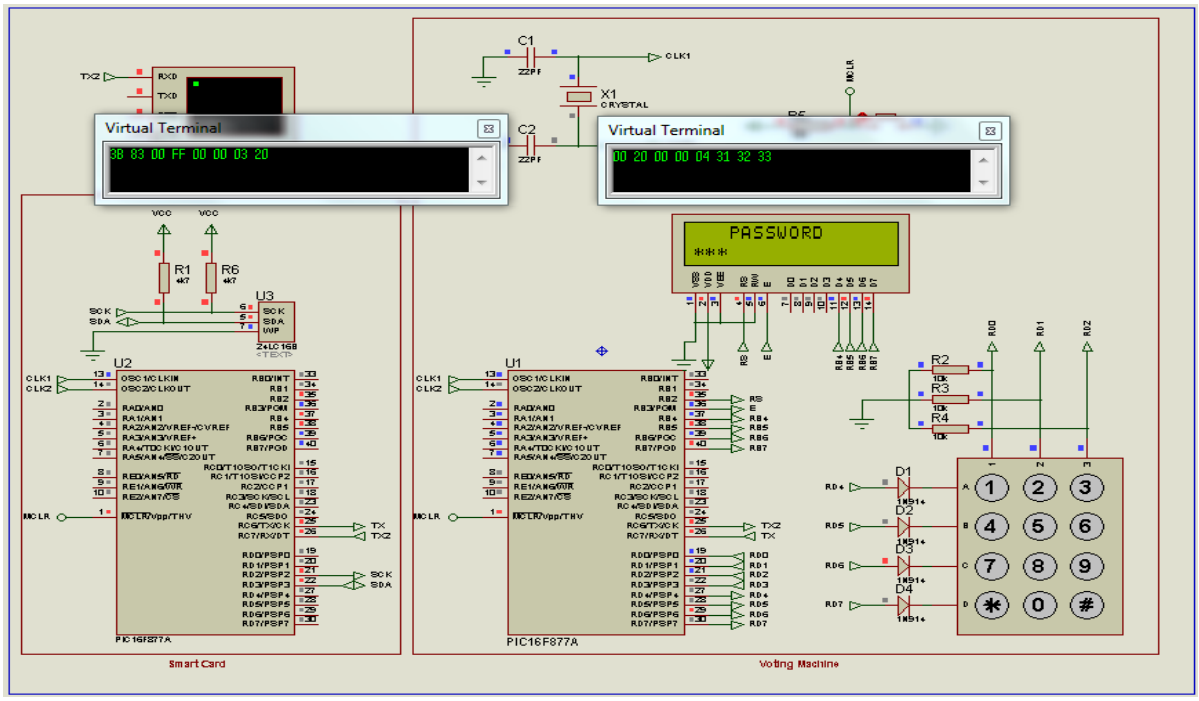


Fig.4.2: L'administrateur commence à saisir son mot de passe.

Le teste est effectué avec deux mots de passe, le premier est incorrecte et le second est correcte.

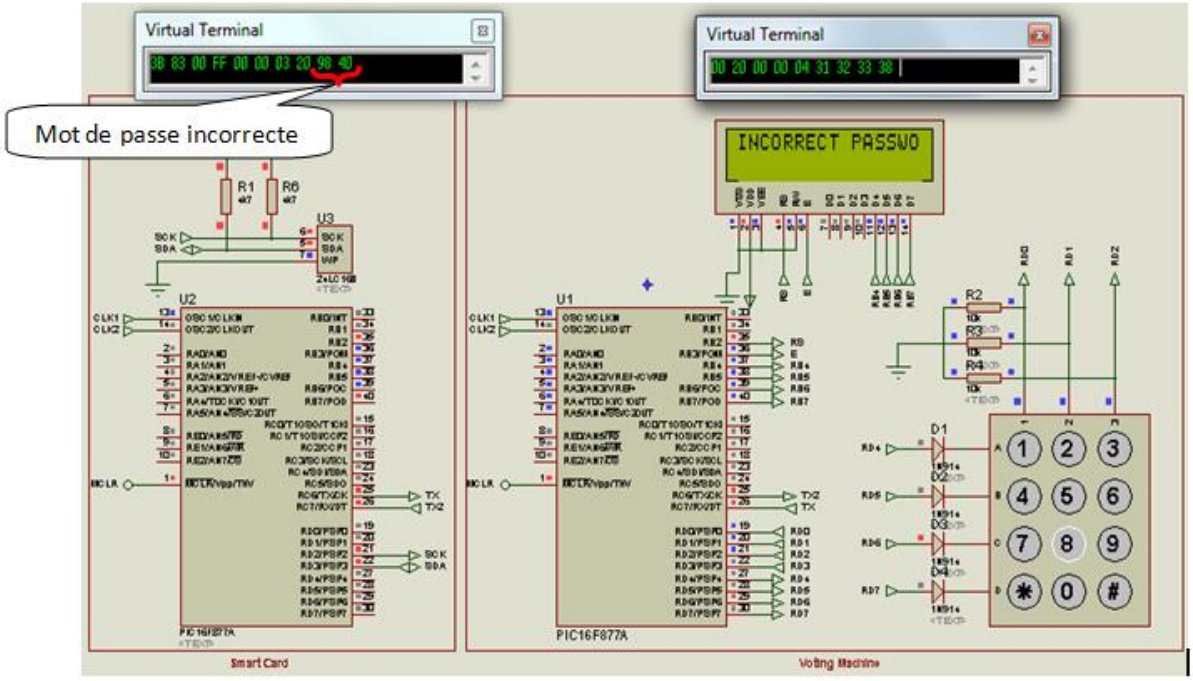


Fig.4.3: Le mot de passe est incorrect.

Quand le mot de passe est incorrect la machine demande à nouveaux à l'administrateur de saisir son mot de passé, la machine lui offre un nombre d'essai limité à 3.

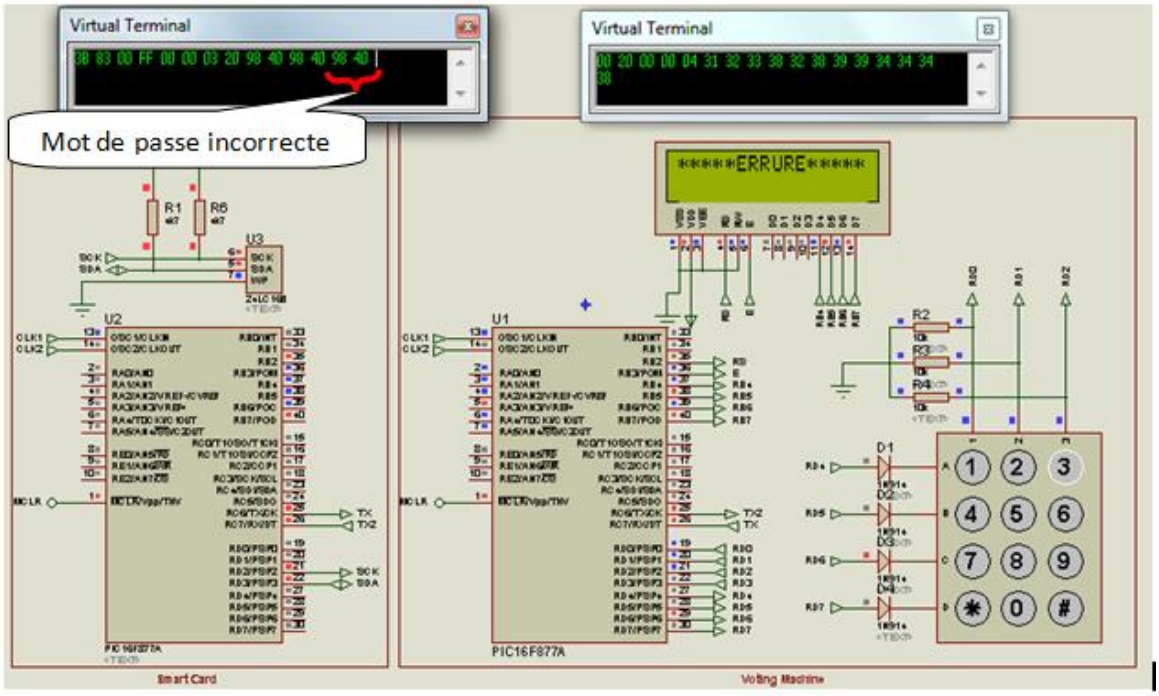


Fig.4.4 : Le mot de passe est incorrect.

Quand le mot de passé est correct. L'écran LCD affiche un message « VOTE INITIATED :# ,SHOW RESULTS :* » Comme indiqué dans la figure suivante.

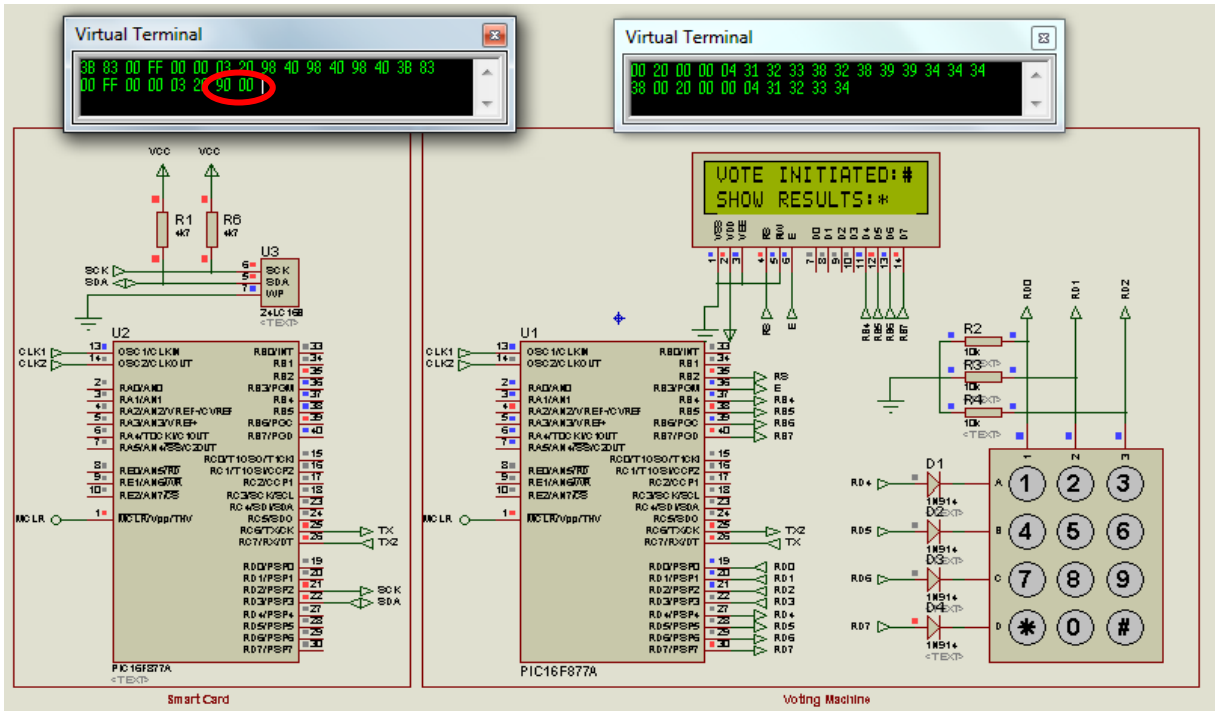


Fig.4.5 : Le mot de passe est correct.

Pour initialise le vote, l'administrateur tape sur'#'. L'écran LCD affiche un message « INSERT CANDIDAT NUMBER : » Comme indiqué dans la figure suivante.

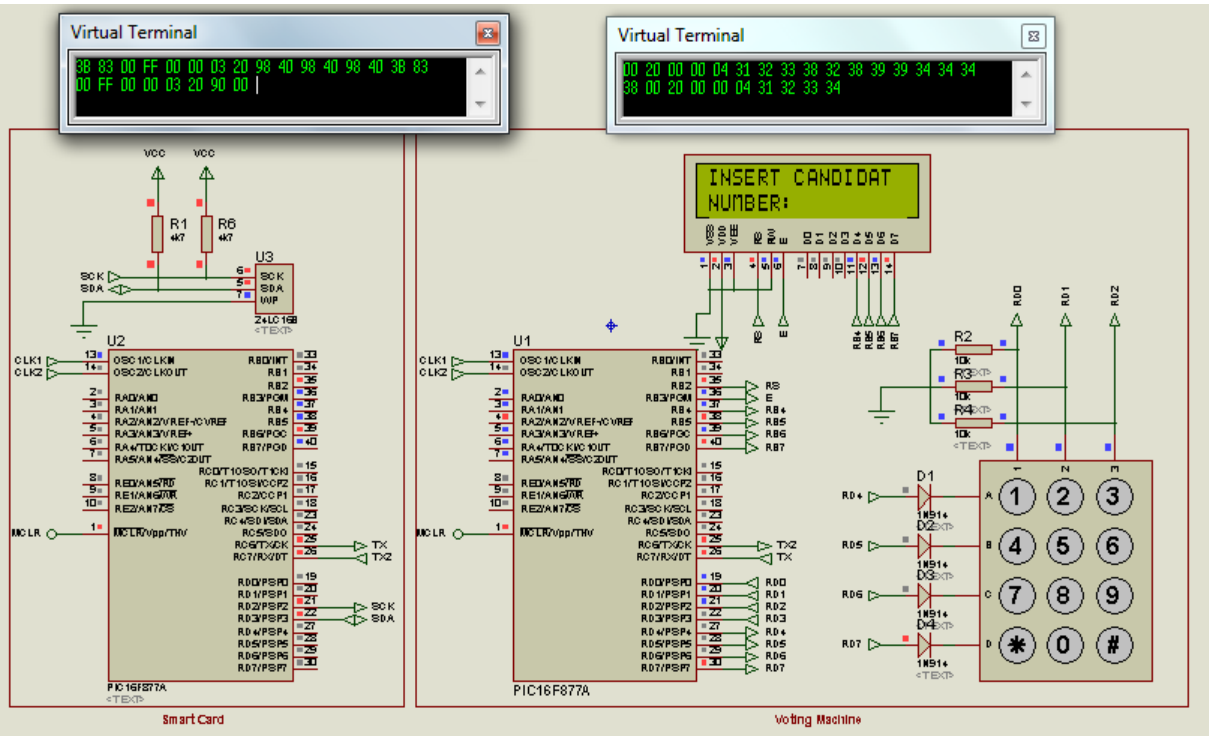


Fig.4.6 : Initialisation de vote.

Si l'électeur choisit un numéro de candidat non existant, la machine va l'inviter à nouveau de saisir un numéro de candidat.

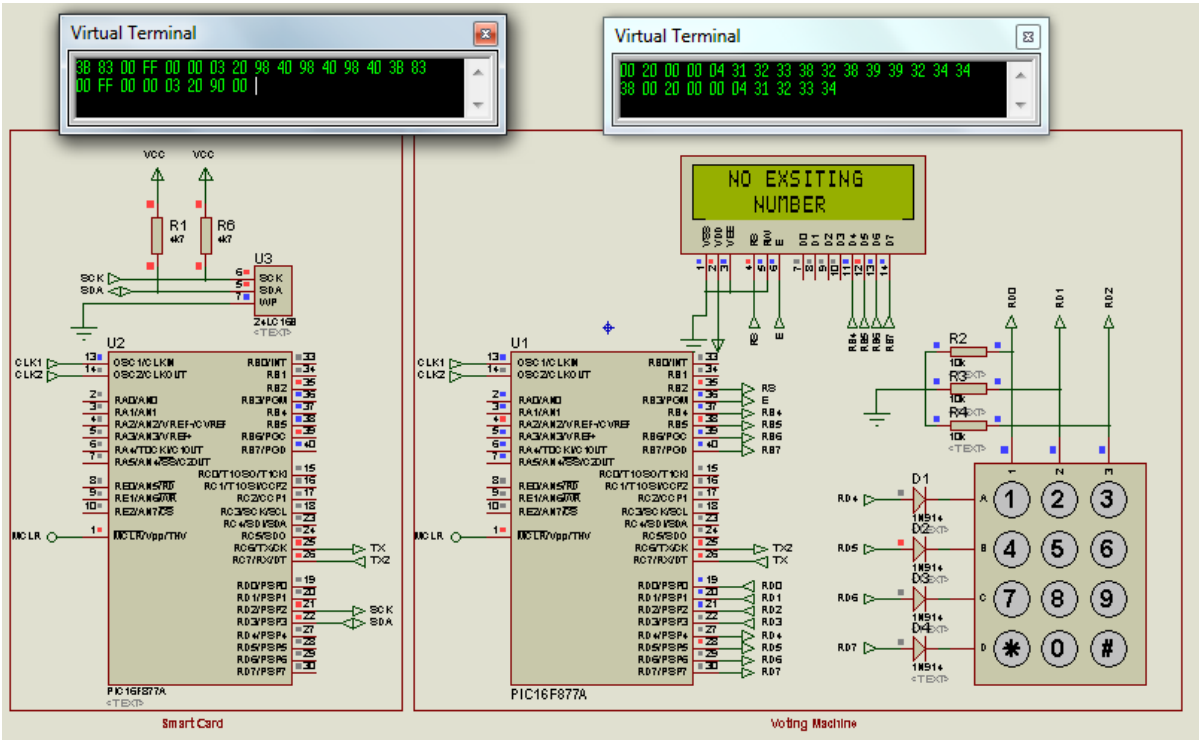


Fig.4.7 : La sélection d'un numéro de candidat non existant.

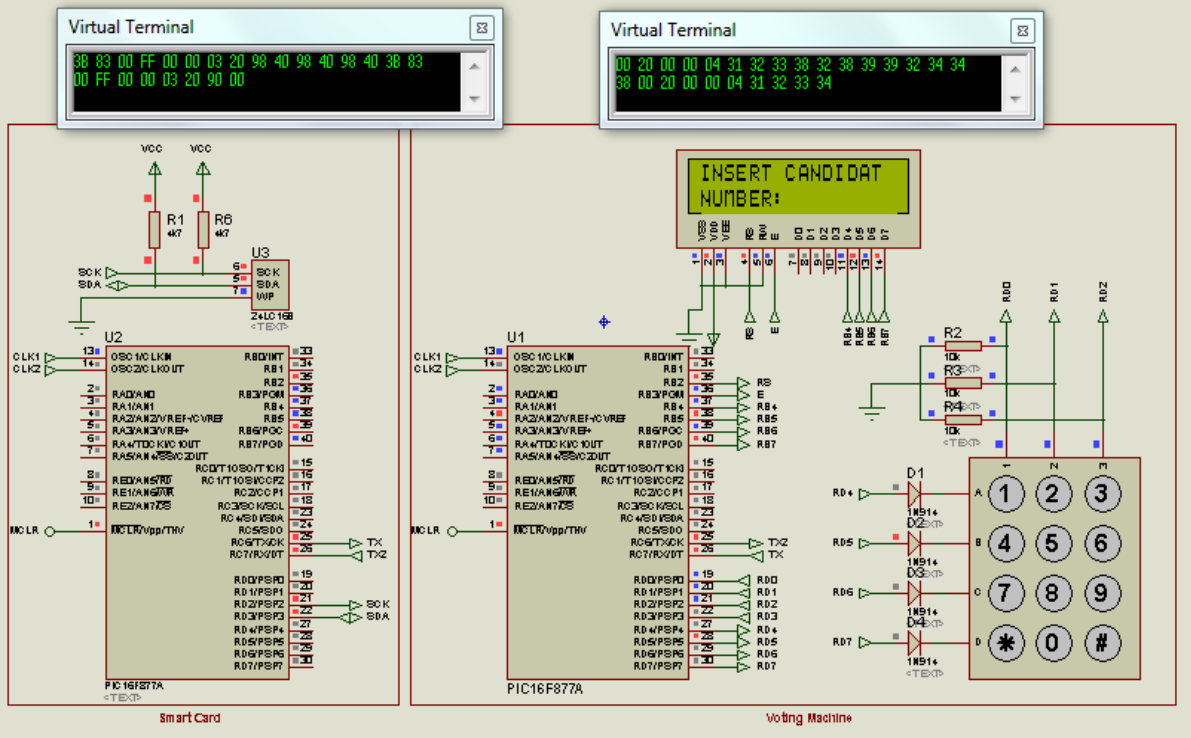


Fig.4.8 : La machine invite l'électeur à introduire le numéro du candidat choisi a nouveau.

Quand l'électeur choisi un bon numéro de candidat, la machine va afficher à travers le LCD le numéro de candidat choisi.

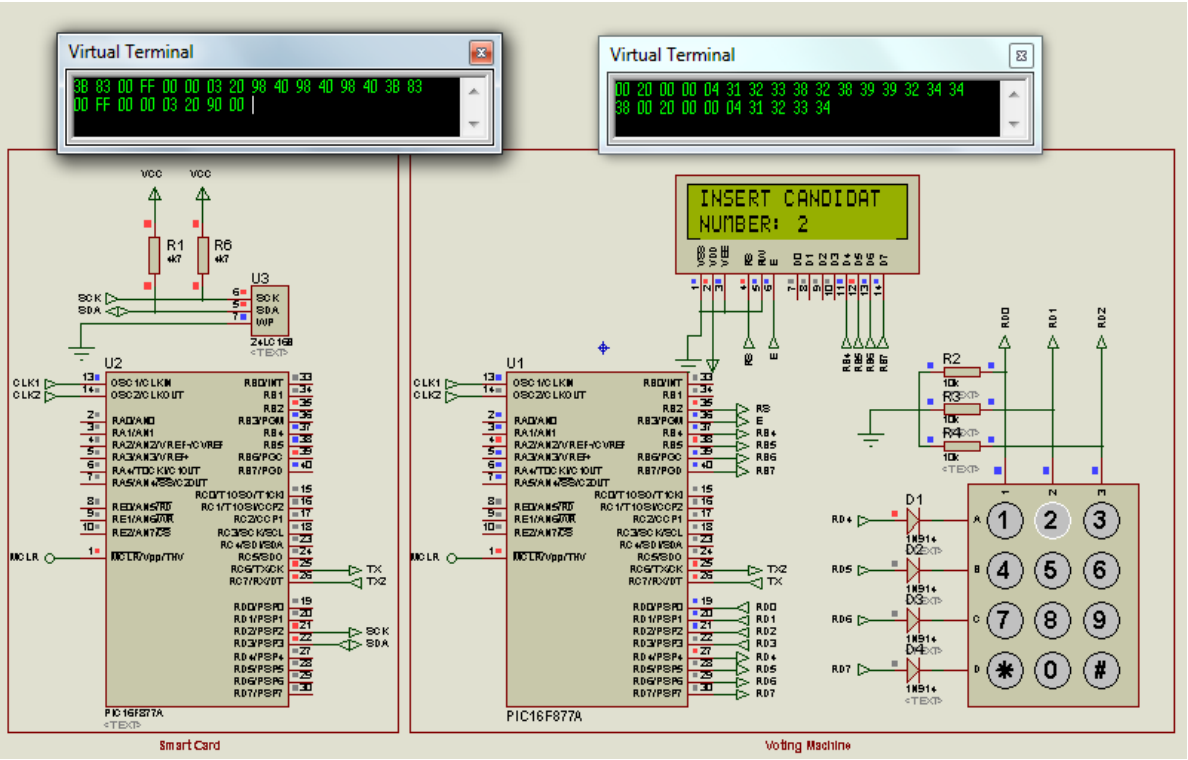


Fig.4.9 : La machine affiche le numéro du candidat sélectionné.

Pour ne pas avoir d'erreur ou un appui accidentel sur une touche du clavier par exemple, la machine invite l'électeur à valider par la touche «#», ou annuler par «* ».

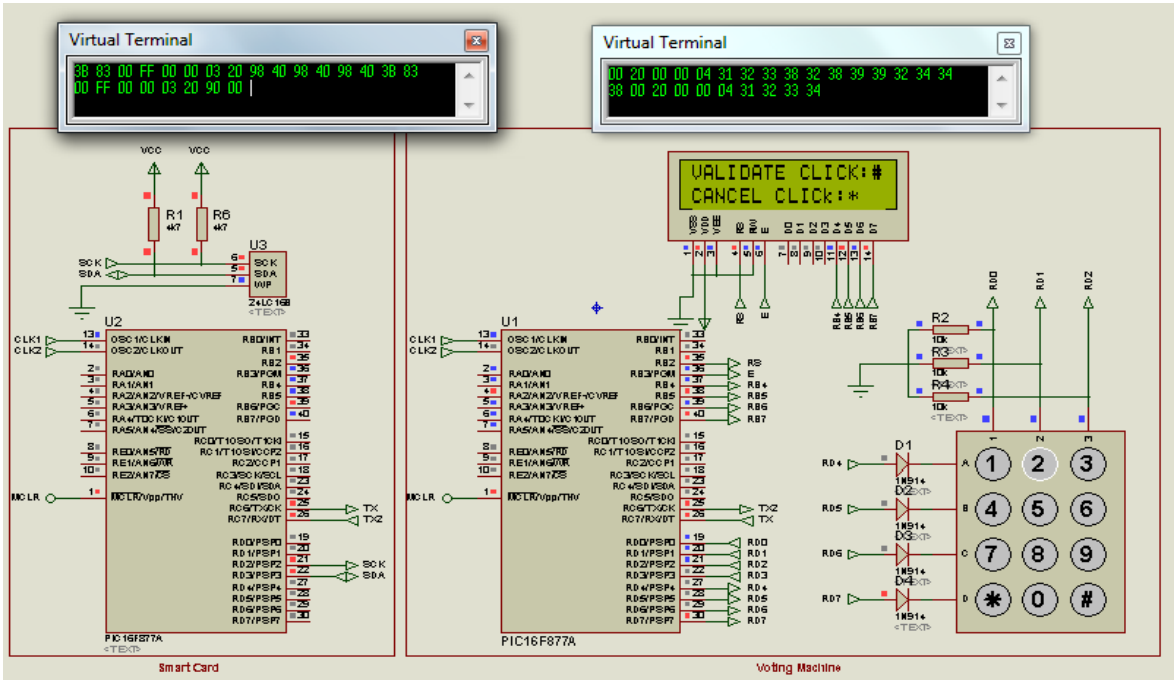


Fig.4.10: Validation ou annulation du vote.

Si l'électeur appui sur « * » pour annuler, il sera invité a sélectionné à nouveau un numéro de candidat.

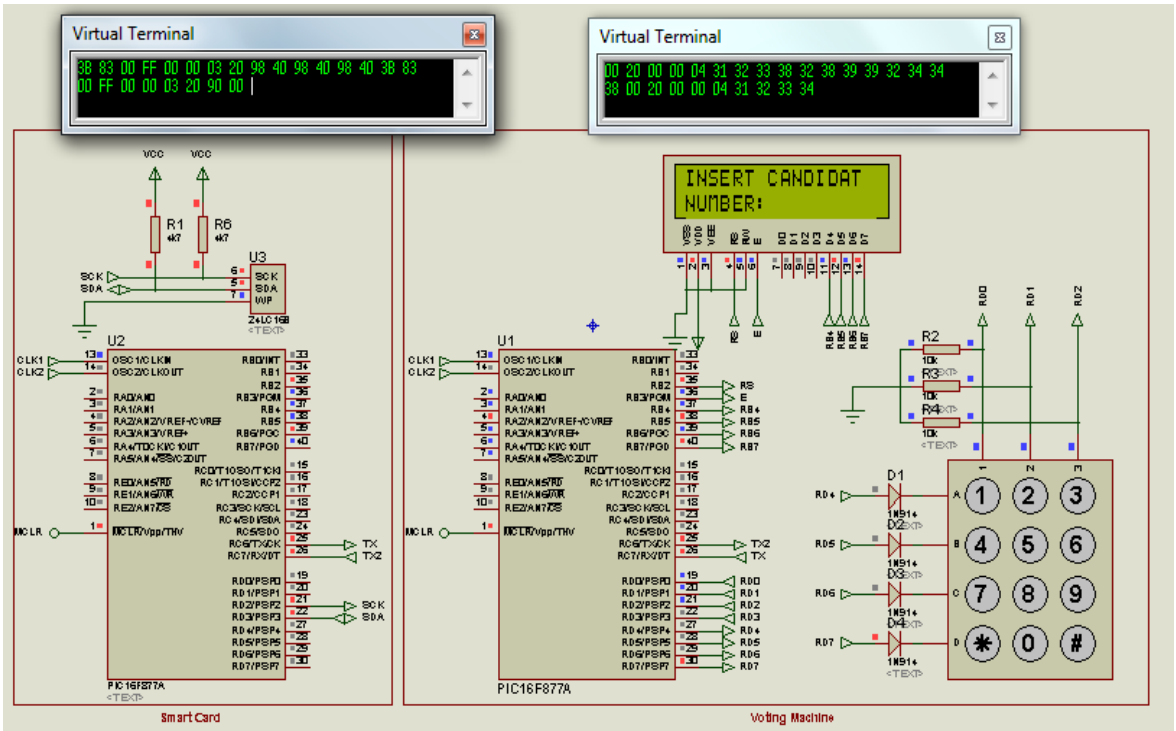


Fig.4.11: La machine invite l'électeur à choisir un autre candidat.

Après la confirmation du choix d'un candidat, deux compteurs internes sont incrémenté dans l'EEPROM interne du microcontrôleur (U1) : le premier incrémente le nombre de voix totale dans le bureau et le 2ème incrémente le nombre de voix correspondant au candidat choisi.

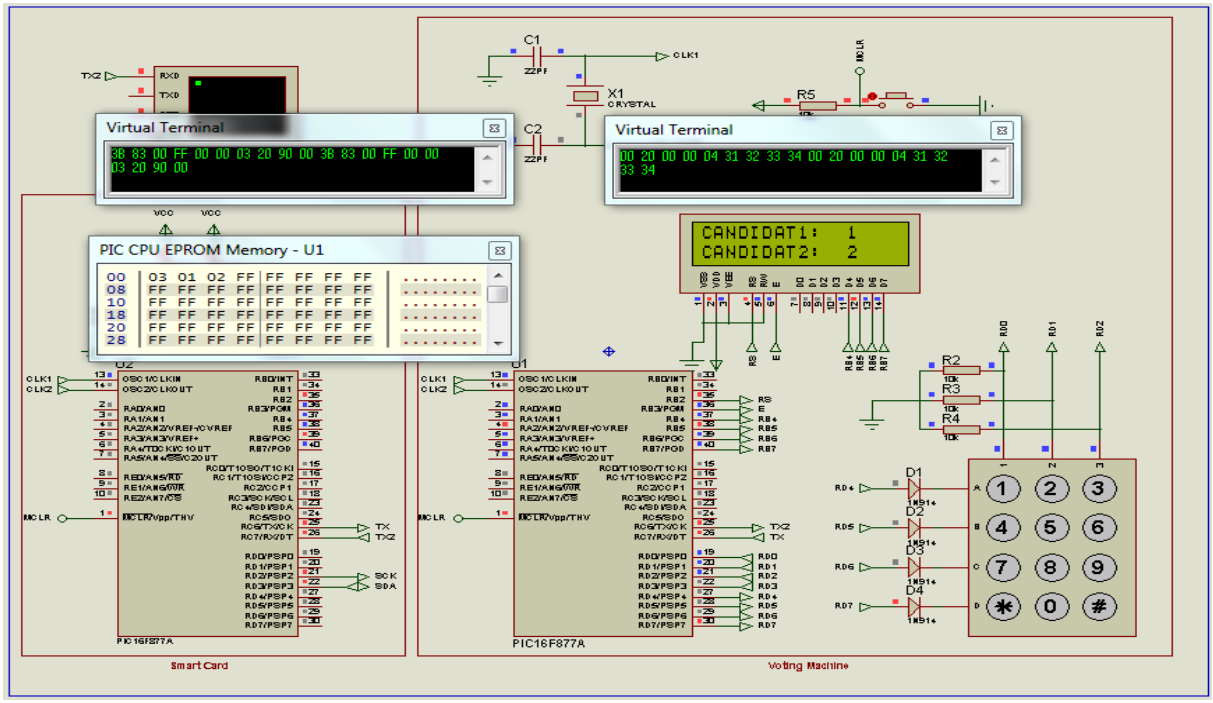


Fig.4.12: La machine valide le vote.

4.3. Réalisation

4.3.1. Programmeurs

PICkit est une famille de programmeurs pour microcontrôleur PIC de Microchip Technology. Ils permettent de programmer les microcontrôleurs et de déboguer les programmes in situ, ainsi que de programmer certaines mémoires EEPROM. Certains modèles proposent également des fonctions d'analyseur logique et de terminal série.

4.3..1. 1. PICkit 3

Microchip continue sa gamme de programmateur avec le PICkit 3, une version ameliorée du PICkit 2 avec les mêmes dimensions et un nouveau boitier translucide. Ce PICkit contient un processeur 16 bits PIC24F plus rapide et supporte une plage de tensions de programmation étendue. Le PICkit 3 tout comme le PICkit 2 possède des régulateurs de tension à découpage. Cela leur permet, dans le cas du PICkit 2, de générer des tensions de 2,5 a 5V, ou dans le cas du PICkit 3, de 2,5 a 5,5V à partir de l'entrée 5V USB (courant d'environ

100mA). Ces 2 programmateurs possèdent des options pour étalonner la sortie avec un multimètre pour une meilleure précision. De plus, pour certains PICs, une tension de programmation MCLR de 13-14V peut être générée. Cette tension est nécessaire pour reprogrammer la mémoire flash.

Fig.4.13: PICkit 3.

4.3. 1. 2. PROGRAMMATION

Les PIC disposent de plusieurs technologies de mémoire de programme : flash, ROM, EPROM, EEPROM, UVPROM. Certains PIC18 permettent l'accès externe à la FLASH et à la RAM.

La programmation du PIC peut se faire:

- par programmation in-situ en utilisant l'interface de programmation / debugge universel ICSP de Microchip. Il suffit alors d'ajouter simplement un connecteur ICSP au microcontrôleur sur la carte fille pour permettre sa programmation une fois soudé ou sur son support (sans avoir besoin de le retirer). Il existe pour cela plusieurs solutions libres (logiciel + interface à faire soi-même) ou commerciales (par exemple : PICkit 3, ICD3 et Real-Ice de Microchip).

4.3. 1. 3. Déboguage

Le déboguage logiciel peut être réalisé de façon logicielle (simulateur) ou hardware (débogueur externe).

Dans les 2 cas, un environnement tel que MPLAB X ou mikroC PRO for PIC peut être utilisé ; Plusieurs solutions existent pour déboguer un programme écrit pour un microcontrôleur PIC : simulateur (dans MPLAB X ou mikroC PRO for PIC) ;

débogueur hardware *in-situ via l'ICSP* ; simulateur Proteus.

4. 3.2. Programmation d'un PIC 16F877A

- Choix du PIC 16F877A.

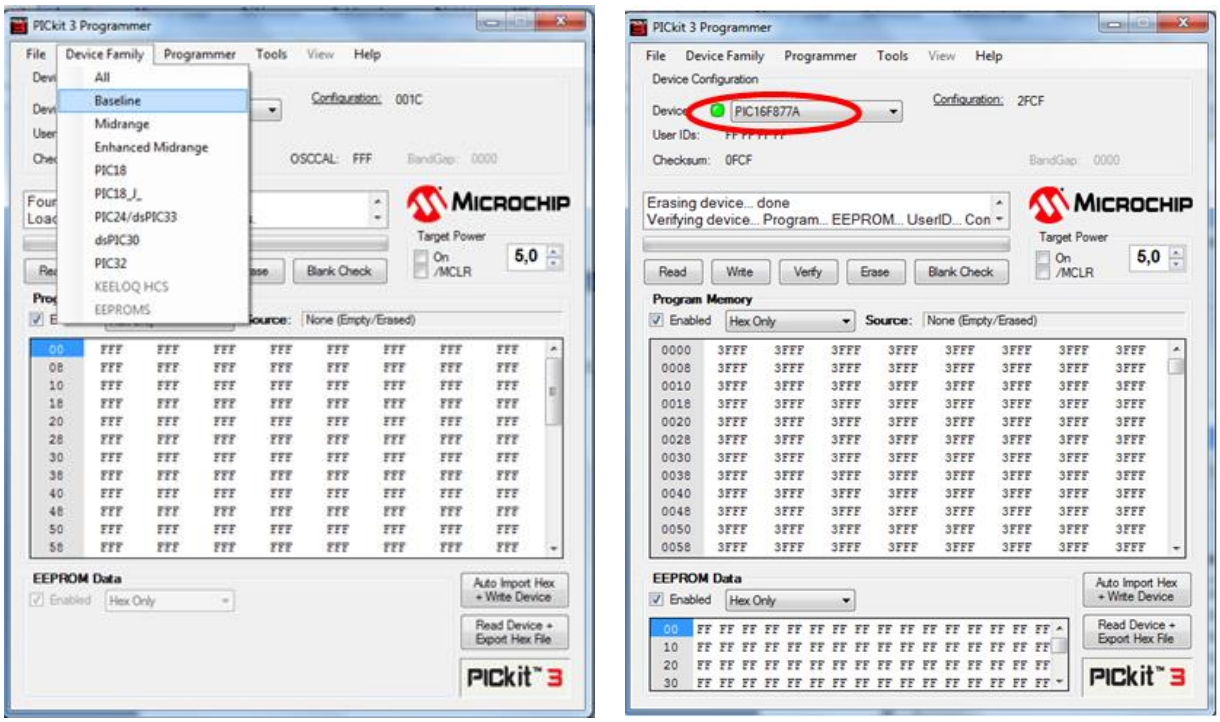


Fig.4.14: Choix du PIC 16F877A.

- Importation de programme réalisé pour carte a puce.

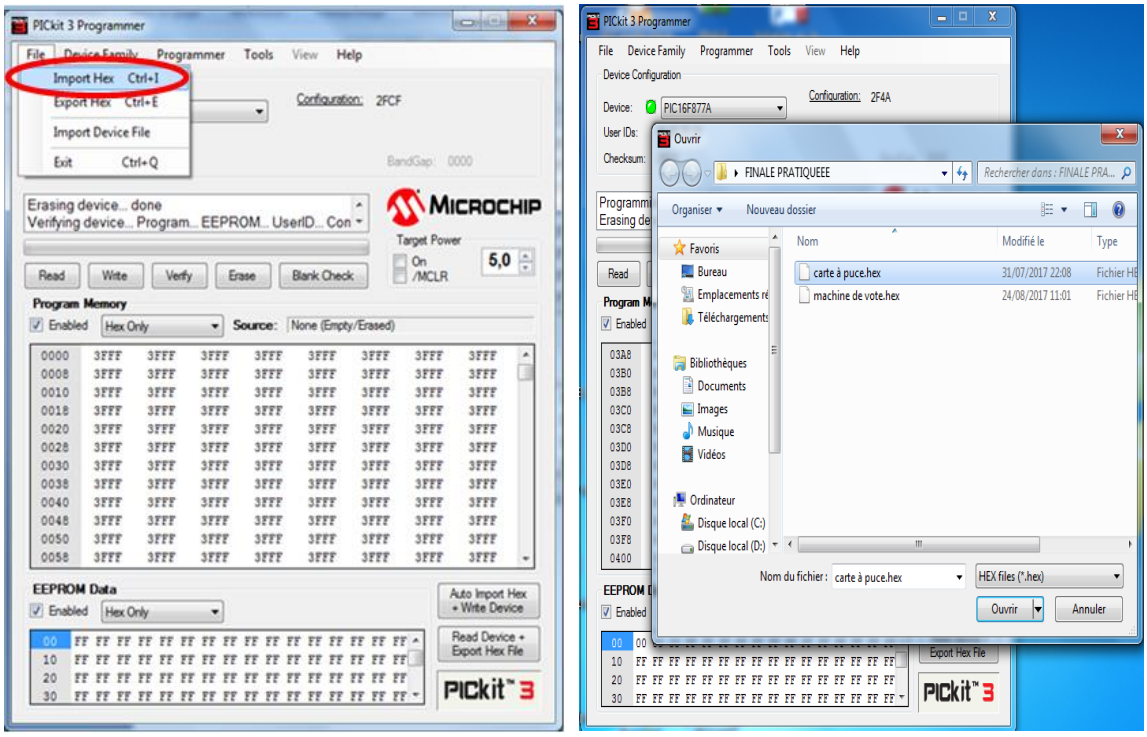


Fig.4.15: Importation de programme réalisé.

- Ecriture du programme.

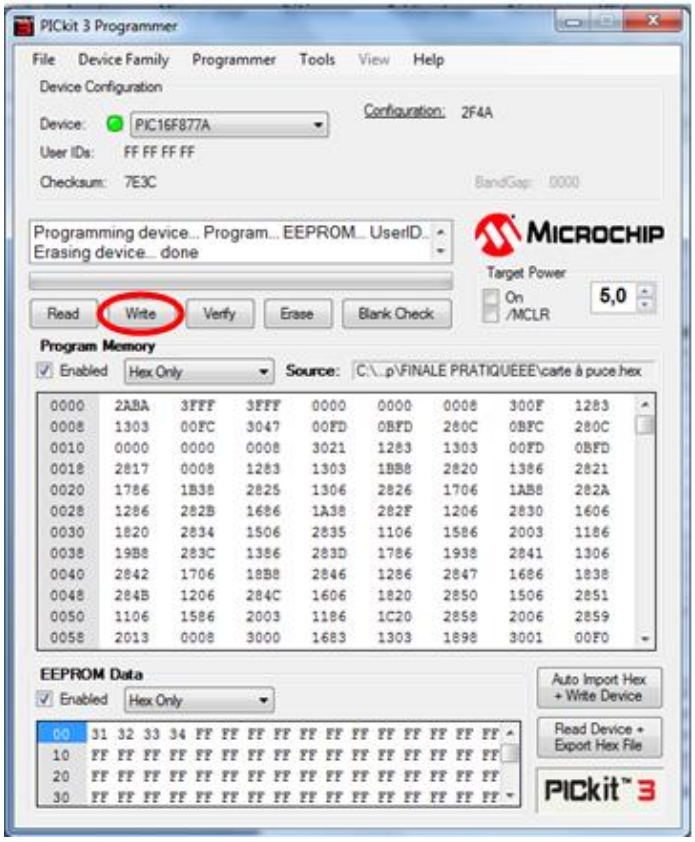


Fig.4.16: Ecriture du programme.

- Importation et écriture du programme réalisé pour machine de vote électronique.

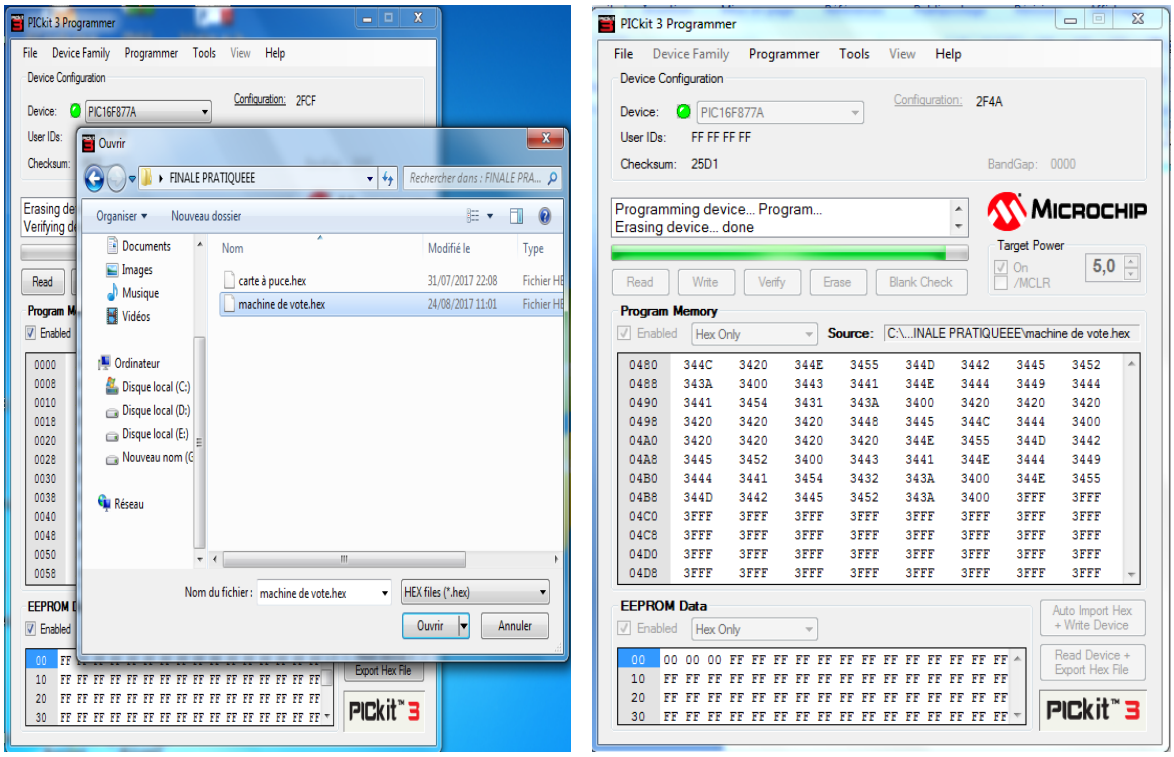


Fig.4.17: Importation et écriture du programme réalisé.

4.3.3. Carte a puce utilisée

Il existe différents types de carte à puce et le choix d'une carte doit être fait très soigneusement. Les cartes Java ou Basic coutent trop chère et pour cela nous avons choisi une carte Wafer 3 à base d'un microcontrôleur PIC (ca donnera les même performances de fonctionnement et de sécurité) car ce sont les moins chères du marché (Wafer 1, 2 ou 3).

4.3.3.1. Carte à puce Wafer 3

Appelée carte Wafer 3, car elle est arrivée après les deux précédentes (Wafer 1,2), ou bien encore carte Silver car de nombreuses versions de cette carte sont disponibles sous une belle livrée argentée (voir les figures 4.18).

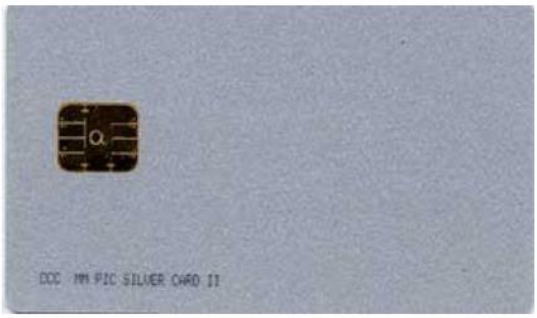


Fig.4.18: La carte Silver.

4.3.3.2. Schéma de la Wafer 3

Cette carte reste fidèle à la famille PIC de Microchip mais fait appel à un circuit doté en ressources internes avec le 16F877, comme le montre son schéma visible ci-dessous.

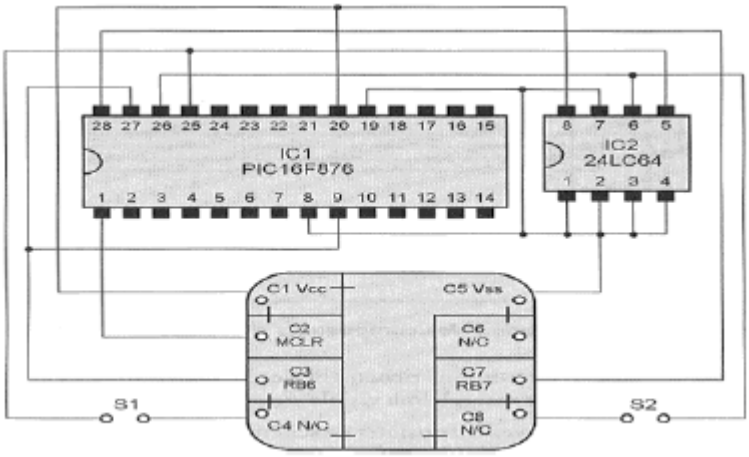


Fig.4.19: Schéma de la Wafer 3.

4.3.3.3. Réalisation de la Wafer 3

- **Liste des composants**

PIC 16F877A

24LC64.

4.3.3.4. Circuit de la Wafer 3

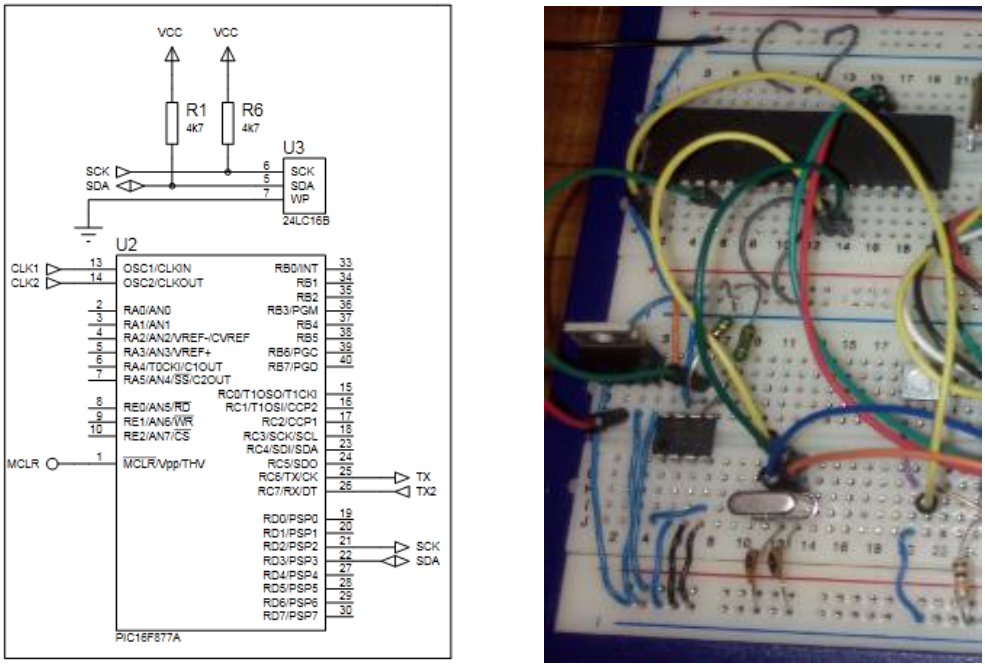


Fig.4.20: Circuit de la Wafer 3.

4.3.3.5. Circuit de machine de vote électronique

- **Source de courant**

Le microcontrôleur fonctionne sur 5V volts. La figure 3.18 représente le schéma de l'alimentation électrique.

Le régulateur de tension produit une tension de sortie qui reste fixe constante indépendamment des variations de la tension d'entrée et de la charge conditions. Régulateur de tension 7805 est utilisé ici, ce qui donne +5 V sortie qui est alors donnée au microcontrôleur pour son opération.

Fig.4.21: Source de Courant.

- **Horloge**

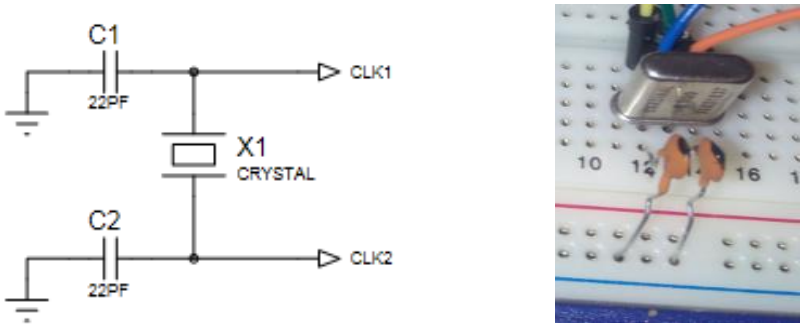


Fig.4.22: Horloge.

4.3.3.6. REALISATION DE MACHINE DE VOTE ELECTRONIQUE

- **2.2. 1.Liste des composants**

1- PIC 16F877A.	**7- Condensateurs:**
2- Quartz 8 MHz.	**C1, C2: 22 pF céramique.**
3- Régulateur de tension L7805C.	**8- Résistances ¼ de watt 5 % :**
4- clavier matriciel (4X4).	**R2, R3, R4, R5=10KΩ**
5- Afficheur LCD (16X2).	**8-Potentiomètre (résistance variable 0-100KΩ).**
6- Mémoire EEPROM 24LC64.	**9- Bouton poussoir.**

Tab.4. 1: Liste des composants nécessaires.

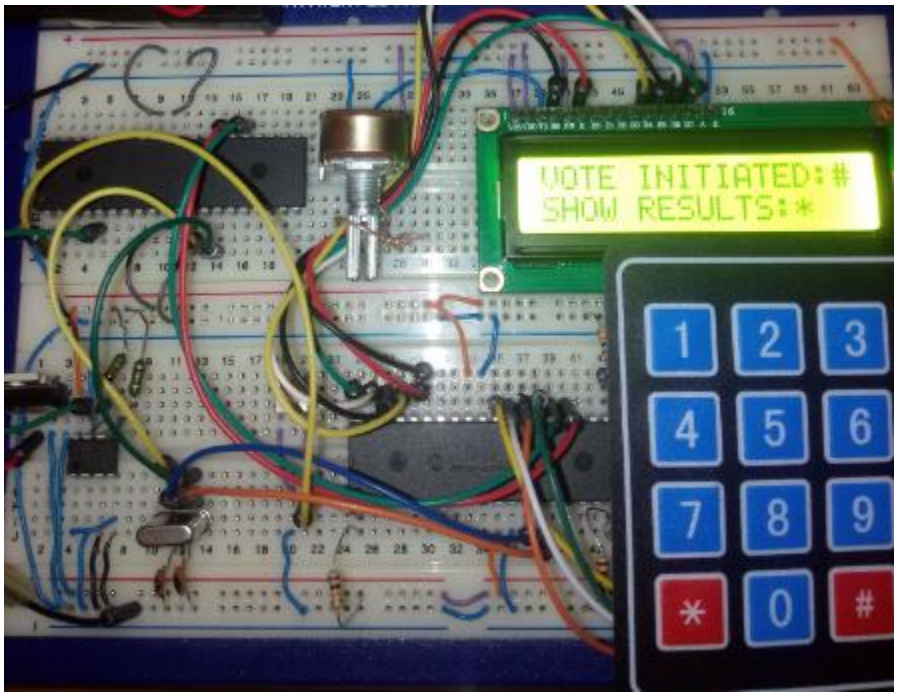


Fig.4.23: Machine de Vote Electronique.

4.4. Conclusion

La simulation et la réalisation de la machine fonctionne normalement avec plusieurs possibilités. Le choix d'un candidat se fait en utilisant un clavier. La machine compte le nombre d‘électeurs et de voies pour chaque candidat.

L’ajout d’une carte à puce a ajouté plus de sécurité à l’authentification de l’administrateur ce qui fait qu’il le seul à avoir accès à la machine que se soit pour l’initialiser ou pour afficher les résultats.

Conclusion générale

Dans ce projet, nous avons conçu, simulé et réalisé une machine de vote électronique améliorée basée sur l'utilisation d'un microcontrôleur et d'une carte à puce permettant de mieux sécuriser l'accès de l'administration à la machine pour donner plus de confiance au résultat de vote.

A l'entrée du bureau, un agent vérifie l'identité de l'électeur. Ensuite, il sélectionne le candidat de son choix au niveau de la machine à l'aide d'un clavier. Après validation du vote, cette dernière est remise à l'attente d'un nouvel électeur. A la fin, un administrateur affiche les résultats de l'élection en s'authentifiant à l'aide de sa carte à puce administrateur et un mot de passe.

Notre système satisfait les propriétés suivantes :

- Facilité et rapidité de manipulation ;
- Rapidité du calcul des résultats ;
- Une authentification de l'administrateur basée sur une carte à puce et un mot de passe afin d'éviter la manipulation des résultats.
- Une machine autonome sans connectivité réseau pour prévenir l'interfaçage avec sa programmation.
- Le système est flexible en utilisant un clavier permettant d'avoir un nombre important de candidats (non limité par quelques boutons poussoirs).

Ce projet peut être amélioré en ajoutant un algorithme cryptographique afin de sécuriser les échanges entre la carte et la machine. Ainsi qu'une authentification biométrique avec l'empreinte digital par exemple afin d'ajouter plus de sécurité.

Ce travail nous a permet d'enrichir nos connaissances dans plusieurs domaines : celui de la carte à puce, la manipulation à base de microcontrôleurs, et de la programmation des protocoles communication et de commandes en langage MicroC.

REFERENCES BIBLIOGRAPHIQUES

[1] M. El Abed, " Le scrutin électronique", rapport de bibliographie IFSIC, N°27006119.

[2] A. Al-Ameen et T. Samani ." The technical feasibility and security of e-voting ", Int. Arab J. Inf. Technologie, Vol. 10, N° 4, P : 397-404.2013

[3] C Enguehard, " La controverse des machines à voter en France", mémoire de master 2, Ecole des hautes études en sciences sociales, Centre Alexandre Koyré, septembre 2011.

[4] C Enguehard. "Vote électronique et preuve papier", $14^{éme}$ Colloque international "De l'insécurité numérique à la vulnérabilité de la société", Paris, France, Jun 2007.

[5] N. Coodman, J. pamett, et J. Debardlen, " une analyse comparative du vote électronique", rapport préparé pour élections canada par le dialogue transatlantique canda_euroupe, Carleton université, Canada, Février 2010.

[6] A. Villafiorita, K. Weldemariam, et R. Tiella ,"Development, formal verification, and evaluation of an E-voting system with VVPAT ", IEEE Transactions on Information Forensics and Security, Vol. 4, N 4, P: 651-661. 2009.

[7] V. Bhatia et R. Gupta, "A novel electronic voting machine design with voter information facility using microcontroller", IEEE International Conference on Computing for Sustainable Global Development (INDIACom), P: 274-276. 2014.

[8] K, Dichou, "contribution à l'étude de cartes a puce avancés", Thèse de Doctorat LMD en université Mhamed BOUGARA Boumerdes, Boumerdes 2016.

[9] International Organization for Standardization. ISO7816. ISO.
www.ISO.com/7816

[10] C Tavernier. "LES Cartes à Puce : Théorie et mise en œuvre ", $2^{éme}$ édition DUNOD. Paris, 2002.

[11] M CHAMI," La Carte à Puce Principes, Applications et Exercices corrigés ", ISBN 978-9954-33-540-6. 2014.

[12] H Nicklous, S Martin ,F Seliger, . "Smart card application development using Java", Springer-Verlag New York, 2002.

[13] P GUEULE " Cartes à puce" ,2ème édition ,DUNOD. Paris. 2001, 160p.

[14] J Patil ,"Implementation of Smart Card Reader Using CPLD", 4^{th}International Conference on Computers and Devices for Communication, CODEC, 2009.

[15] D. Sauveron, "Étude et réalisation d'un environnement d'expérimentation et de modélisation pour la technologie Java CardTM. Application à la sécurité". Thèse de doctorat en Université Bordeaux I, Décembre 2004.

[16] P. Gueule," PC et cartes à puce," DUNOD, Paris. 2000.

[17] K. Markantonakis, K. Mayes, "Secure smart embedded devices, platforms and applications" , Springer, University of London, 2014.

[18] W. Rankl, W. Effing," Smart Card Handbook", 4^{th}Ed, John Wiley & Sons, New York, 2010.

[19] A Mehmoud, S Muhammad , C Arshad, A Khawja , "Micro-controller based smart electronic voting machine system", IEEE International Conference on Electro/Information Technology (EIT), P: 438-442, 2014.

[20] A Review , "Electronic Voting Machine", the International Conference on Pattern Recognition (Informatics and Medical Engineering), March 21-23,2012

[21] A. Al-Ameen, S.A. Talab," The technical feasibility and security of e-voting", International Arabic Journal of Information Technology, vol. 10, Issue 4,P : 397-404, 2013.

[22] D.A. Kumar, T.U.S. Begum, "Electronic voting machine A review", International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME), Salem, Tamilnadu, P : 41-48, Mar 2012.

[23] G. Z. Qadah ,T. Rani, "Electronic voting systems: Requirements, design, and implementation". Computer Standards & Interfaces, Vol. 29, N° 3, P: 376-386, 2007.

Annexes

```
 1: const char txt1[] = "INSERT CANDIDAT";
 2: const char txt2[] = "NUMBER:";
 3: const char txt3[] = "  NO EXSITING";
 4: const char txt4[] = "    NUMBER";
 5: const char txt5[] = "ENTER NEW CODE";
 6: const char txt6[] = "VALIDATE CLICK:#" ;
 7: const char txt7[]  = "CANCEL CLICk:*";
 8: const char txt8[]  = "SUCCESSFUL VOTE";
 9: const char txt9[]  = "      HELD";
10: const char txt10[] = "   ENTRER YOUR  ";
11: const char txt11[] = "    PASSWORD    ";
12: const char txt12[] = "    PASSWORD   ";
13: const char txt13[] = "VOTE INITIATED:#";
14: const char txt14[] = "  SHOW RESULTS:*";
15: const char txt15[] = " VOTE INITIATED" ;
16: const char txt16[] = "  VOTE RESULTS ";
17: const char txt17[] = "INSERT YOUR CARD";
18: const char txt18[] = "     CORRECT  ";
19: const char txt19[] = "INCORRECT PASSWORD  ";
20: const char txt20[] = "*****ERRURE*****";
21: const char txt21[] =  "CANDIDAT1:";
22: const char txt22[] = "CANDIDAT2:";
23: const char txt23[] = "TOTAL NUMBER:";
24:
25: char msg[17];
26:
27: sbit LCD_RS at RB2_bit;
28: sbit LCD_EN at RB3_bit;
29: sbit LCD_D7 at RB7_bit;
30: sbit LCD_D6 at RB6_bit;
31: sbit LCD_D5 at RB5_bit;
32: sbit LCD_D4 at RB4_bit;
33:
34: sbit LCD_RS_Direction at TRISB2_bit;
35: sbit LCD_EN_Direction at TRISB3_bit;
36: sbit LCD_D7_Direction at TRISB7_bit;
37: sbit LCD_D6_Direction at TRISB6_bit;
38: sbit LCD_D5_Direction at TRISB5_bit;
39: sbit LCD_D4_Direction at TRISB4_bit;
40:
41: char keypadPort at PORTD;
42:
43:   int amine()
44:   { char  kp;
45:
46:   while(1){
47:   kp=0;
48:   while(!kp){kp = Keypad_Key_Click(); }
49:   switch(kp) {
50:   case 1: kp = '1'; break;// 1
51:   case 2: kp = '2'; break;// 2
52:   case 3: kp = '3'; break;// 3
53:   case 4: kp = 'A'; break;// A
54:   case 5: kp = '4'; break;//4
55:   case 6: kp = '5'; break;// 5
56:   case 7: kp = '6'; break;// 6
57:   case 8: kp = 'B'; break;//B
58:   case 9: kp = '7'; break;//  7
59:   case 10: kp = '8'; break;// 8
60:   case 11: kp = '9'; break;//9
61:   case 12: kp = 'C'; break;//C
62:   case 13: kp = '*'; break;// *
```

```
 63:    case 14: kp = '0'; break;// 0
 64:    case 15: kp = '#'; break;// #
 65:    case 16: kp = 'D'; break;// D
 66:    }
 67:    return kp;
 68:
 69:    }
 70:    }
 71:
 72:    void delay_20ms()
 73:    {
 74:    Delay_ms(20);
 75:
 76:    }
 77:
 78:    void APDU()
 79:    {
 80:     char a;
 81:     int uart_r;
 82:     UART1_Init(9615);
 83:     Delay_ms(200);
 84:     a=0;
 85:     while(a<7){
 86:                 if (UART1_Data_Ready()) {uart_r = UART1_Read();
 87:                                                       if(uart_r ==0x03){
 88:                                                                          UART1_Write
    e(0X00);Delay_ms(200);
 89:                                                                          UART1_Write
    e(0X20);Delay_ms(200);
 90:                                                                          UART1_Write
    e(0X00);Delay_ms(200);
 91:                                                                          UART1_Write
    e(0X00);Delay_ms(200);
 92:                                                                          UART1_Write
    e(0X04);Delay_ms(200);
 93:                                                                         }
 94:                                                                         a++;
 95:
 96:                                          }
 97:              }
 98:    }
 99:
100:    void mot_pass()
101:    { char sofi, matr [6];
102:      int i=1,j=2;
103:
104:      for(i=1;i<5;i++)
105:      {
106:       sofi= amine();
107:       lcd_chr(j, i, '*');
108:       matr [i]=sofi;
109:       UART1_write(matr [i]);
110:       if(i==5){ Lcd_Cmd(_LCD_CLEAR);i=0;};
111:
112:      }
113:    }
114:
115:   char * CopyConst2Ram(char * dest, const char *src)
116:   {
117:     char * d ;
118:     d = dest;
119:     for(;*dest++ = *src++;);
```

```
120:    return d;
121:  }
122:
123: char  kp,abdou;
124: char uart_r[3] ;
125: char n,cnt=0;
126: char txt[4];
127: unsigned short CV=0,CC1=0,CC2=0;
128:
129: void main()
130: {
131:     trisd=0b00001111;
132:     trisa=0x00;
133:     CC1 = EEPROM_Read(0x01);
134:     delay_ms(20);
135:     CC2 = EEPROM_Read(0x02);
136:     delay_ms(20);
137:     CV = EEPROM_Read(0x00);
138:     delay_ms(20);
139:     Lcd_Init();
140:     Lcd_Cmd(_LCD_CURSOR_OFF);
141:     Lcd_Cmd(_LCD_CLEAR);
142:     lcd_out(1,1,CopyConst2Ram(msg, txt17));
143:     APDU();
144: line1:    Lcd_Cmd(_LCD_CLEAR);
145:     lcd_out(1,1,CopyConst2Ram(msg, txt10));
146:     lcd_out(2,1,CopyConst2Ram(msg, txt11));
147:     delay_ms(2000);
148:     Lcd_Cmd(_LCD_CLEAR);
149:     lcd_out(1,1,CopyConst2Ram(msg, txt12));
150:     mot_pass();
151:     n=0;
152:     while(1){ UART1_Init(9615);
153:     Delay_ms(200);
154:                 if (UART1_Data_Ready()) {uart_r[n] = UART1_Read();
155:
156:                                           if(n==1||n==3||n==5)
157:                                          {
158:                                                 if(uart_r[0] ==0x90&&uart_r[1]
 ==0x00){Lcd_Cmd(_LCD_CLEAR);porta.b2=1; lcd_out(1,1,CopyConst2Ram(msg, txt18));
delay_ms(2000);break;}
159:                                                 else
160:                                                 {
161:                                                  if(cnt<3){Lcd_Cmd(_LCD_CLEAR)
;lcd_out(1,1,CopyConst2Ram(msg, txt19));delay_ms(2000);porta.b0=1;goto line1;}
162:                                                  if(cnt=3){Lcd_Cmd(_LCD_CLEAR)
;porta.b3=1; lcd_out(1,1,CopyConst2Ram(msg, txt20));}
163:                                                 }
164:
165:                                          }
166:                                          n++; cnt++;
167:                                      }
168:                                }
169:     delay_ms(500);
170:     Lcd_Cmd(_LCD_CLEAR);
171:     lcd_out(1,1,CopyConst2Ram(msg, txt13));
172:     lcd_out(2,1,CopyConst2Ram(msg, txt14));
173:     abdou = amine();
174:     if(abdou =='#'){Lcd_Cmd(_LCD_CLEAR);lcd_out(1,1,CopyConst2Ram(msg, txt15));del
lay_ms(1000);goto line2;}
175:     if(abdou =='*'){Lcd_Cmd(_LCD_CLEAR);lcd_out(1,1,CopyConst2Ram(msg, txt16));del
lay_ms(1000);Lcd_Cmd(_LCD_CLEAR); lcd_out(1,1,CopyConst2Ram(msg, txt21));
```

```
176:
177:       ByteToStr(CC1, txt);
178:       Lcd_Out_Cp(txt);
179:       ByteToStr(CC2, txt);
180:       lcd_out(2,1,CopyConst2Ram(msg, txt22));
181:       Lcd_Out_Cp(txt);
182:       delay_ms(4000);
183:       CV=CC1+CC2;
184:       EEPROM_Write(0x00,CV);delay_ms(20);
185:       Lcd_Cmd(_LCD_CLEAR);
186:       ByteToStr(CV, txt);
187:       lcd_out(1,1,CopyConst2Ram(msg, txt23));
188:       Lcd_Out_Cp(txt);
189:       delay_ms(4000);
190:       goto line1;
191:
192:
193:
194: line2:
195:   do
196: {
197:
198:       Keypad_Init();    // Initialize Keypad
199:       Lcd_Init();
200:
201: line: Lcd_Cmd(_LCD_CURSOR_OFF);
202:       Lcd_Cmd(_LCD_CLEAR);
203:       lcd_out(1,1,CopyConst2Ram(msg, txt1));
204:       lcd_out(2,1,CopyConst2Ram(msg, txt2));
205:
206:       abdou = amine();
207:       lcd_chr(2,9, abdou);delay_ms(1000);
208:       Lcd_Cmd(_LCD_CLEAR);
209:       if(abdou==0X30||abdou==0X34||abdou==0X33||abdou==0X35||abdou==0X36||abdou==
    0X37||abdou==0X38||abdou==0X39||abdou==0X41||abdou==0X42||abdou==0X43||abdou==
    4||abdou==0X23||abdou==0X2A)
210:       {Lcd_Cmd(_LCD_CLEAR);lcd_out(1,1,CopyConst2Ram(msg, txt3));lcd_out(2,1,CopyC
    Const2Ram(msg, txt4));delay_ms(1000);Lcd_Cmd(_LCD_CLEAR);goto line;}
211:
212:       if(abdou==0X31)  {Lcd_Cmd(_LCD_CLEAR);
213:                         lcd_out(1,1,CopyConst2Ram(msg, txt6));
214:                         lcd_out(2,1,CopyConst2Ram(msg, txt7)); delay_ms(500);
215:
216:        abdou = amine();
217:        if(abdou==0X23)  {Lcd_Cmd(_LCD_CLEAR);
218:                         CC1++;CV++;
219:                         EEPROM_Write(0x01,CC1);delay_ms(20);lcd_out(1,1,CopyConst2R
    Ram(msg, txt8));
220:                         lcd_out(2,1,CopyConst2Ram(msg, txt9));delay_ms(3000);Lcd_Cm
    md(_LCD_CLEAR);goto line; }
221:       abdou = amine();
222:       if(abdou==0X2A){goto line;};
223:                         }
224:
225:        if(abdou==0X32) {Lcd_Cmd(_LCD_CLEAR);
226:                         lcd_out(1,1,CopyConst2Ram(msg, txt6));
227:                         lcd_out(2,1,CopyConst2Ram(msg, txt7)); delay_ms(500);
228:
229:        abdou = amine();
230:        if(abdou==0X23)  {Lcd_Cmd(_LCD_CLEAR);
231:                         CC2++;
232:                         EEPROM_Write(0x02,CC2);delay_ms(20);lcd_out(1,1,CopyConst2R
    Ram(msg, txt8));
```

```
233:                         lcd_out(2,1,CopyConst2Ram(msg, txt9));delay_ms(3000);Lcd_Cm
    md(_LCD_CLEAR);
234:                         goto line; }
235:       abdou = amine();
236:       if(abdou==0X2A){goto line;};
237:                         }
238:
239:
240:
241:
242:
243:  }
244: while(1);
245:
246:  }
247: }
```

```
 1: const char txt1[] = "INSERT CANDIDAT";
 2: const char txt2[] = "NUMBER:";
 3: const char txt3[] = "  NO EXSITING";
 4: const char txt4[] = "    NUMBER";
 5: const char txt5[] = "ENTER NEW CODE";
 6: const char txt6[] = "VALIDATE CLICK:#" ;
 7: const char txt7[]  = "CANCEL CLICk:*";
 8: const char txt8[]  = "SUCCESSFUL VOTE";
 9: const char txt9[]  = "      HELD";
10: const char txt10[] = "   ENTRER YOUR  ";
11: const char txt11[] = "    PASSWORD    ";
12: const char txt12[] = "    PASSWORD   ";
13: const char txt13[] = "VOTE INITIATED:#";
14: const char txt14[] = "  SHOW RESULTS:*";
15: const char txt15[] = " VOTE INITIATED" ;
16: const char txt16[] = "  VOTE RESULTS ";
17: const char txt17[] = "INSERT YOUR CARD";
18: const char txt18[] = "     CORRECT  ";
19: const char txt19[] = "INCORRECT PASSWORD  ";
20: const char txt20[] = "*****ERRURE*****";
21: const char txt21[] =  "CANDIDAT1:";
22: const char txt22[] = "CANDIDAT2:";
23: const char txt23[] = "TOTAL NUMBER:";
24:
25: char msg[17];
26:
27: sbit LCD_RS at RB2_bit;
28: sbit LCD_EN at RB3_bit;
29: sbit LCD_D7 at RB7_bit;
30: sbit LCD_D6 at RB6_bit;
31: sbit LCD_D5 at RB5_bit;
32: sbit LCD_D4 at RB4_bit;
33:
34: sbit LCD_RS_Direction at TRISB2_bit;
35: sbit LCD_EN_Direction at TRISB3_bit;
36: sbit LCD_D7_Direction at TRISB7_bit;
37: sbit LCD_D6_Direction at TRISB6_bit;
38: sbit LCD_D5_Direction at TRISB5_bit;
39: sbit LCD_D4_Direction at TRISB4_bit;
40:
41: char keypadPort at PORTD;
42:
43:   int amine()
44:   { char  kp;
45:
46:   while(1){
47:   kp=0;
48:   while(!kp){kp = Keypad_Key_Click(); }
49:   switch(kp) {
50:   case 1: kp = '1'; break;// 1
51:   case 2: kp = '2'; break;// 2
52:   case 3: kp = '3'; break;// 3
53:   case 4: kp = 'A'; break;// A
54:   case 5: kp = '4'; break;//4
55:   case 6: kp = '5'; break;// 5
56:   case 7: kp = '6'; break;// 6
57:   case 8: kp = 'B'; break;//B
58:   case 9: kp = '7'; break;//  7
59:   case 10: kp = '8'; break;// 8
60:   case 11: kp = '9'; break;//9
61:   case 12: kp = 'C'; break;//C
62:   case 13: kp = '*'; break;// *
```

```
 63:   case 14: kp = '0'; break;// 0
 64:   case 15: kp = '#'; break;// #
 65:   case 16: kp = 'D'; break;// D
 66:   }
 67:   return kp;
 68:
 69:   }
 70:   }
 71:
 72:   void delay_20ms()
 73:   {
 74:   Delay_ms(20);
 75:
 76:   }
 77:
 78:   void APDU()
 79:   {
 80:    char a;
 81:    int uart_r;
 82:    UART1_Init(9615);
 83:    Delay_ms(200);
 84:    a=0;
 85:    while(a<7){
 86:                if (UART1_Data_Ready()) {uart_r = UART1_Read();
 87:                                                    if(uart_r ==0x03){
 88:                                                                       UART1_Write
    e(0X00);Delay_ms(200);
 89:                                                                       UART1_Write
    e(0X20);Delay_ms(200);
 90:                                                                       UART1_Write
    e(0X00);Delay_ms(200);
 91:                                                                       UART1_Write
    e(0X00);Delay_ms(200);
 92:                                                                       UART1_Write
    e(0X04);Delay_ms(200);
 93:                                                                  }
 94:                                                                    a++;
 95:
 96:                                          }
 97:             }
 98:   }
 99:
100:   void mot_pass()
101:   { char sofi, matr [6];
102:     int i=1,j=2;
103:
104:     for(i=1;i<5;i++)
105:     {
106:      sofi= amine();
107:      lcd_chr(j, i, '*');
108:      matr [i]=sofi;
109:      UART1_write(matr [i]);
110:      if(i==5){ Lcd_Cmd(_LCD_CLEAR);i=0;};
111:
112:     }
113:   }
114:
115:  char * CopyConst2Ram(char * dest, const char *src)
116:  {
117:    char * d ;
118:    d = dest;
119:    for(;*dest++ = *src++;);
```

```
120:    return d;
121:  }
122:
123: char  kp,abdou;
124: char uart_r[3] ;
125: char n,cnt=0;
126: char txt[4];
127: unsigned short CV=0,CC1=0,CC2=0;
128:
129: void main()
130: {
131:     trisd=0b00001111;
132:     trisa=0x00;
133:     CC1 = EEPROM_Read(0x01);
134:     delay_ms(20);
135:     CC2 = EEPROM_Read(0x02);
136:     delay_ms(20);
137:     CV = EEPROM_Read(0x00);
138:     delay_ms(20);
139:     Lcd_Init();
140:     Lcd_Cmd(_LCD_CURSOR_OFF);
141:     Lcd_Cmd(_LCD_CLEAR);
142:     lcd_out(1,1,CopyConst2Ram(msg, txt17));
143:     APDU();
144: line1:    Lcd_Cmd(_LCD_CLEAR);
145:     lcd_out(1,1,CopyConst2Ram(msg, txt10));
146:     lcd_out(2,1,CopyConst2Ram(msg, txt11));
147:     delay_ms(2000);
148:     Lcd_Cmd(_LCD_CLEAR);
149:     lcd_out(1,1,CopyConst2Ram(msg, txt12));
150:     mot_pass();
151:     n=0;
152:     while(1){ UART1_Init(9615);
153:     Delay_ms(200);
154:                 if (UART1_Data_Ready()) {uart_r[n] = UART1_Read();
155:
156:                                           if(n==1||n==3||n==5)
157:                                          {
158:                                                 if(uart_r[0] ==0x90&&uart_r[1]
 ==0x00){Lcd_Cmd(_LCD_CLEAR);porta.b2=1; lcd_out(1,1,CopyConst2Ram(msg, txt18));
delay_ms(2000);break;}
159:                                                 else
160:                                                 {
161:                                                  if(cnt<3){Lcd_Cmd(_LCD_CLEAR)
;lcd_out(1,1,CopyConst2Ram(msg, txt19));delay_ms(2000);porta.b0=1;goto line1;}
162:                                                  if(cnt=3){Lcd_Cmd(_LCD_CLEAR)
;porta.b3=1; lcd_out(1,1,CopyConst2Ram(msg, txt20));}
163:                                                 }
164:
165:                                          }
166:                                          n++; cnt++;
167:                                       }
168:                                 }
169:     delay_ms(500);
170:     Lcd_Cmd(_LCD_CLEAR);
171:     lcd_out(1,1,CopyConst2Ram(msg, txt13));
172:     lcd_out(2,1,CopyConst2Ram(msg, txt14));
173:     abdou = amine();
174:     if(abdou =='#'){Lcd_Cmd(_LCD_CLEAR);lcd_out(1,1,CopyConst2Ram(msg, txt15));del
lay_ms(1000);goto line2;}
175:     if(abdou =='*'){Lcd_Cmd(_LCD_CLEAR);lcd_out(1,1,CopyConst2Ram(msg, txt16));del
lay_ms(1000);Lcd_Cmd(_LCD_CLEAR); lcd_out(1,1,CopyConst2Ram(msg, txt21));
```

```
176:
177:       ByteToStr(CC1, txt);
178:       Lcd_Out_Cp(txt);
179:       ByteToStr(CC2, txt);
180:       lcd_out(2,1,CopyConst2Ram(msg, txt22));
181:       Lcd_Out_Cp(txt);
182:       delay_ms(4000);
183:       CV=CC1+CC2;
184:       EEPROM_Write(0x00,CV);delay_ms(20);
185:       Lcd_Cmd(_LCD_CLEAR);
186:       ByteToStr(CV, txt);
187:       lcd_out(1,1,CopyConst2Ram(msg, txt23));
188:       Lcd_Out_Cp(txt);
189:       delay_ms(4000);
190:       goto line1;
191:
192: line2:
193:   do
194: {
195:       Keypad_Init();    // Initialize Keypad
196:       Lcd_Init();
197:
198: line: Lcd_Cmd(_LCD_CURSOR_OFF);
199:       Lcd_Cmd(_LCD_CLEAR);
200:       lcd_out(1,1,CopyConst2Ram(msg, txt1));
201:       lcd_out(2,1,CopyConst2Ram(msg, txt2));
202:
203:       abdou = amine();
204:       lcd_chr(2,9, abdou);delay_ms(1000);
205:       Lcd_Cmd(_LCD_CLEAR);
206:       if(abdou==0X30||abdou==0X34||abdou==0X33||abdou==0X35||abdou==0X36||abdou==
    0X37||abdou==0X38||abdou==0X39||abdou==0X41||abdou==0X42||abdou==0X43||abdou==
    4||abdou==0X23||abdou==0X2A)
207:       {Lcd_Cmd(_LCD_CLEAR);lcd_out(1,1,CopyConst2Ram(msg, txt3));lcd_out(2,1,CopyC
    Const2Ram(msg, txt4));delay_ms(1000);Lcd_Cmd(_LCD_CLEAR);goto line;}
208:
209:       if(abdou==0X31)  {Lcd_Cmd(_LCD_CLEAR);
210:                         lcd_out(1,1,CopyConst2Ram(msg, txt6));
211:                         lcd_out(2,1,CopyConst2Ram(msg, txt7)); delay_ms(500);
212:
213:        abdou = amine();
214:        if(abdou==0X23)  {Lcd_Cmd(_LCD_CLEAR);
215:                         CC1++;CV++;
216:                         EEPROM_Write(0x01,CC1);delay_ms(20);lcd_out(1,1,CopyConst2R
    Ram(msg, txt8));
217:                         lcd_out(2,1,CopyConst2Ram(msg, txt9));delay_ms(3000);Lcd_Cm
    md(_LCD_CLEAR);goto line; }
218:       abdou = amine();
219:       if(abdou==0X2A){goto line;};
220:                         }
221:
222:        if(abdou==0X32) {Lcd_Cmd(_LCD_CLEAR);
223:                         lcd_out(1,1,CopyConst2Ram(msg, txt6));
224:                         lcd_out(2,1,CopyConst2Ram(msg, txt7)); delay_ms(500);
225:
226:        abdou = amine();
227:        if(abdou==0X23)  {Lcd_Cmd(_LCD_CLEAR);
228:                         CC2++;
229:                         EEPROM_Write(0x02,CC2);delay_ms(20);lcd_out(1,1,CopyConst2R
    Ram(msg, txt8));
230:                         lcd_out(2,1,CopyConst2Ram(msg, txt9));delay_ms(3000);Lcd_Cm
    md(_LCD_CLEAR);
```

```
231:                          goto line; }
232:       abdou = amine();
233:       if(abdou==0X2A){goto line;};
234:                          }
235:  }
236: while(1);
237:
238:  }
239: }
```