

Remerciements

*A l'issue de cette fin d'étude, j'adresse mes sincères remerciements
premièrement à allah le tout puissant qui nous a donné la santé, la
volonté et la patience, pour achever ce travail.*

*Je tiens à remercier en premier lieu Monsieur LIMANE DJAMEL
pour nous avoir aidés dans la réalisation pratique de ce projet.*

*Je tiens aussi à remercier tous ceux qui ont contribué de loin ou de
près à la réalisation de ce travail*

*Je remercie également tous les membres du jury pour l'intérêt
qu'ils ont porté à mon travail.*

Au nom de dieu clément et miséricordieux

Dédicace

Avant tout, je tiens à remercier le bon dieu, et l'unique qui m'offre le courage et la volonté nécessaire pour affronter les différentes difficultés de la vie, Je dédie ce modeste travail

A ma mère.

A mon père.

A mes frères et mes sœurs.

A toute ma famille.

A mes amis et mes collègues de la promotion 2018

« informatique ».

A mes amis (chaque un à son nom).

A tous les musulmans dans le monde entier.

S.SAIDAT

S.FARSI

Liste des figures	v
RESUME	1
1 Généralités.....	2
2 Motivation	2
3 Objectif	2
4 Présentation du mémoire	3
 CHAPITRE I Les systèmes embarqués et les microcontrôleurs dans les systèmes embarqués 	
I .Les systèmes embarqués	4
I .1Introduction	4
I.2 Système embarqué typique	5
I.3 Circuit logique programmable « FPGA »	6
I.4 Caractéristiques principales d'un système embarqué	7
II.Les microcontrôleurs dans les systèmes embarqués	7
II.1 Organisation du plan mémoire :	8
II.2 Architecture de Von Neumann et architecture de Harvard	8
II.3 Architecture de Harvard modifiée DSP (Digital Signal Processor)	10
II.4 Définition d'un microcontrôleur	11
II.5 Composants intégrés d'un microcontrôleur	11
III .Le microcontrôleur ATMEL AVR (ATMEGA)	12
III .1 Définition	12
III .2 La Famille AVR	12
III .3 Les caractéristique du µc ATMEGA32.....	13
III .4 L'Architecture interne de l'ATMEGA32	14
III .5Présentation Physique	14
III .6Descriptions des broches	15
IV.Conclusion	16
 CHAPITRE II Les logiciels et matériels utilisé 	
I.Introduction	17
II Les logiciels	17

II.1Le compilateur BASCOM-AVR pour AVR" d'Atmel.....	17
II.1.1 Définition	17
II.1.2 Principales caractéristiques	18
II.1.3 Fonctions spéciales	18
II.1.4 Préparation de Bascom.....	19
II.1.5 ENREGISTREMENT, COMPILATION, CONTROLE.....	20
II.1.6 Le simulateur du "BASCOM-AVR"	20
II.1.7 Le transfère vers le microcontrôleur	21
III .Le système Android _____	21
III .1L'outil App Inventor	22
III .2Historique de logiciel App Inventor	22
III .3Langage JAVA	23
III .4Structure d'une application App Inventor	23
IV. Les matérielles _____	25
IV. 1Plaque d'essai	25
IV. 2Les straps	26
IV. .3Afficheur LCD	26
IV. .4Module de Relais à 4 canaux.....	27
V. Conclusion _____	28

CHAPITRE III Conception de la carte et Réalisation

I.Introduction _____	29
II.La programmation du microcontrôleur l'Atemega AVR _____	30
II. 1Lancement de Compilateur :	30
II. 2.Saisie du programme :	31
II. 3Compilation : compile F7.....	31
II. 4Le transfert vers le microcontrôleur.	32
III.Création application android _____	33
Conclusion générale et perspectives _____	38

LISTE DES FIGURES

<i>Figure 1: Architecture matérielle d'un système embarqué. [10].</i>	5
<i>Figure 2: Architecture de Von Neuman. [2].</i>	9
<i>Figure 3: Architecture de Harvard [2].</i>	9
<i>Figure 4: Structure de Harvard modifiée [2].</i>	10
<i>Figure 5: Synoptique générale d'un ATMEGA 32 [3]</i>	14
<i>Figure 6: Brochage typique d'un ATMEGA32 en boîtier PDIP et PLCC [3]</i>	15
<i>Figure 7: Détermination de microcontrôleur utilisé</i>	19
<i>Figure 8 : Configuration de LCD.</i>	19
<i>Figure 9: Le simulateur du "BASCOM-AVR".</i>	20
<i>Figure 10: Le transfère vers le microcontrôleur.</i>	21
<i>Figure 11: Les composants graphiques.</i>	23
<i>Figure 12 Les propriétés et les classes sous App Inventor.</i>	24
<i>Figure 13: En-tête d'éditeur de blocks App Inventor.</i>	25
<i>Figure 14: Plaque d'essai [14].</i>	26
<i>Figure 15: les straps. [15].</i>	26
<i>Figure 16: Afficheur LCD. [15].</i>	27
<i>Figure 17: Module de Relais à 4 canaux[15].</i>	27
<i>Figure 18: Installation des composants électrique sur la carte d'essai</i>	29
<i>Figure 19: Fenêtre principale de compilateur bascom-AVR.</i>	30
<i>Figure 20: Exemple de programme.</i>	31
<i>Figure 21: Compilation de programme.</i>	31
<i>Figure 22: Création de fichier (.hex).</i>	32
<i>Figure 23: Création de nouveau projet sur App Inventor.</i>	33
<i>Figure 24: Première interface de la création App Inventor.</i>	34
<i>Figure 25: l'ajout d'un bouton sous App Inventor.</i>	34
<i>Figure 26: Schéma global du Screen 1</i>	35
<i>Figure 27: Un bloc pour la gestion du bouton ON et un autre pour le bouton OFF.</i>	35
<i>Figure 28: Fenêtre d'enregistrement de l'application.</i>	36
<i>Figure 29: Photo réelle de notre application CCN 4 MOTEUR.</i>	36
<i>Figure 30: Interface PC</i>	37

Résumé

L'objectif de ce travail est de réaliser une carte d'interface à base de microcontrôleur dans le but de commander des moteurs via un PC et un mobile. Ainsi, cette mémoire se compose à deux parties : La première partie concerne une étude théorique sur les systèmes embarqués et les microcontrôleurs AVR, et la deuxième partie est consacrée pour l'étude conceptuelle d'un système de contrôle commande du moteur via un PC et le microcontrôleur.

*Cette étude est faite dans l'optique de réaliser un support comportant : **Sur le plan Hardware:** une carte de commande comportant le microcontrôleur Atmega AVR avec quatre moteurs est réalisé.*

***Sur le plan Software:** une interface graphique est développée en employant le visuel basic, et une application android en employant l'utile App Inventor.*

Mots clés : système embarqué, microcontrôleurs Atmega, bascom AVR, App inventor, breadboard.

ABSTRACT

The objective of this work is to create a microcontroller-based interface board for the purpose of controlling motors via a PC and a mobile, thus, this memoir is composed of two parts:

The first part concerns a theoretical study on AVR board systems and microcontrollers, and the second part is dedicated to the conceptual study of a control system of the motor via a PC and the microcontroller. This study is done with the aim of producing a support comprising On the hardware side: a control card with the Atmega AVR microcontroller with four motors is realized. On the Software level: a graphical interface is developed by using the basic visual, and an android application by using the useful App Inventor.

Key words: embedded system, microcontrollers, bascom AVR, invantor App, breadboard

الملخص

الهدف من هذا العمل هو إنشاء لوحة واجهة قائمة على متحكم دقيق للتحكم في المحركات عبر الكمبيوتر الشخصي والهاتف المحمول، تتكون هذه المذكرة من جزأين: الجزء الأول يتعلق بدراسة نظرية عن الأنظمة المدمجة و المتحكمات الدقيقة الجزء

الثاني مخصص للدراسة المفاهيمية لنظام التحكم في المحرك عبر جهاز الكمبيوتر والمتحكم الدقيق.

تم إجراء هذه الدراسة لتحقيق مايلي: على مستوى الأجهزة: إنشاء بطاقة تحكم مكونة من متحكم وأربعة محركات.

على مستوى البرامج: تطوير واجهة رسومية باستخدام Visual Basic ، وتطبيق Android باستخدام الأداة App Inventor.

الكلمات المفتاحية: نظام مدمج، متحكم Atmega، لغة البرمجة AVR bascom ،App inventor ،لوحة التجارب.

INTRODUCTION GENERALE

1 GÉNÉRALITÉS

La forte augmentation des ventes de smart phone et de tablettes électronique se fait en même temps qu'une adoption rapide par le grand public des technologies de la domotique ainsi que l'autopilotage. Au fond, le smart phone, avec sa connectivité intégrée (Bluetooth, WIFI), devient une télécommande universelle pour toute la maison et les équipements électriques. Les utilisateurs pourront à terme contrôler à distance un très grand nombre de fonctions sans avoir à tenir compte de la marque ou de l'origine du produit qu'ils pilotent.

D' autre part La technologie de pointe utilise pour la commande des processus industriels surtout les équipements électriques jusqu'a les navettes spéciales les microcontrôleurs qui sont le cerveau de guidage et contrôle majeur dans l'époque du numérique.

Pour répondre à cette évolution majeure, nous avons créé une carte qui permet de contrôler n'importe quel appareil, véhicule, machine à travers un smart phone ou à une tablette. Cette carte électronique réalisée rend la commande et le contrôle facile et souple lors du pilotage à distance.

2 MOTIVATION

Dans la vie moderne, on utilise pas mal d'outils et d'accessoires de commande à distance afin de simplifier notre contrôle, donc nous chercherons toujours à se concentrer sur la souplesse de la commande et de contrôler sur une zone bien définie (notre contour) le plus grand nombre possible d'accessoires. Le smart phone occupe la premier place d'objets qui ne nous quittent pas donc notre travail se concentre sur l'utilisation de ce dernier avec bien sur sa liaison avec un système ou une carte de commande (carte d'interface).

3 OBJECTIF

Dans ce projet trois objectifs ont été visés :

- Le premier est de regrouper suffisamment d'informations sur une grande catégorie de microcontrôleurs ATmega AVR : son langage de programmation, et son principe de fonctionnement.

- Le deuxième consiste à réaliser une carte électrique capable d'exécuter une action entre un smart phone et une carte d'interfaçage en expliquant les différents blocs de sa construction.
- Le troisième est de réaliser une application sous smart phone et de le programmer sous l'environnement JAVA afin de simplifier la commande et de montrer l'intérêt de la programmation orientée objet.

4 PRÉSENTATION DU MÉMOIRE

Le premier chapitre sera consacré à une étude approfondie sur les systèmes embarqués et les microcontrôleurs, puis, on mettra la lumière sur un modèle de base qui est (ATmega AVR) sa construction son environnement de programmation et son principe de fonctionnement afin de simplifier son utilisation.

Le deuxième chapitre abordera le sujet sur les logiciels utilisés dans la réalisation de nos projets.

Dans le troisième chapitre, on présentera un algorithme; on expliquera le fonctionnement de notre programme, son déclenchement à l'action à distance et cela sur deux modes de commande (domotique et l'autoguidage) après le chargement du code dans le microcontrôleur et de réaliser une application capable de gérer une telle commande sous smart phone.

Enfin, on terminera avec une conclusion générale qui résumera l'intérêt de notre étude : les différents résultats obtenus expérimentalement seront donnés sur annexe (A) et (B) sous forme de photos, donnant ainsi un aperçu sur les performances de la carte d'interfaçage, et une idée sur les problèmes à résoudre ultérieurement.

Chapitre

1

Les systèmes embarqués et les microcontrôleurs dans les systèmes embarqués

I .LES SYSTÈMES EMBARQUÉS

I .1 INTRODUCTION

Un système embarqué est un système électronique et informatique autonome ne possédant pas des entrées/sorties standards comme un clavier ou un écran d'ordinateur. Il est piloté dans la majorité des cas par un logiciel, qui est complètement intégré au système qu'il contrôle. On peut aussi définir un système embarqué comme un système électronique soumis à diverses contraintes. [09].

Les concepteurs des systèmes électroniques sont aujourd'hui confrontés à la complexité croissante des algorithmes mis en œuvre et à la variété des cibles potentielles FPGAs et/ou DSPs. Actuellement, il n'est pas rare que ces systèmes intègrent plusieurs douzaines voire des centaines de processeurs. A l'origine, ce sont des systèmes matériels et logiciels intégrés dans des avions militaires ou des missiles. Ensuite dans le civil: avions, voitures, machine à laver...! La machine et le logiciel sont intimement liés et noyés dans le matériel et ne sont pas aussi facilement discernables comme dans un environnement de travail classique de type PC. On mentionne les contraintes physiques fortes: dimensions, poids, taille, autonomie, consommation, fiabilité, contraintes temporelles (temps réels). [09].

I.2 SYSTÈME EMBARQUÉ TYPIQUE

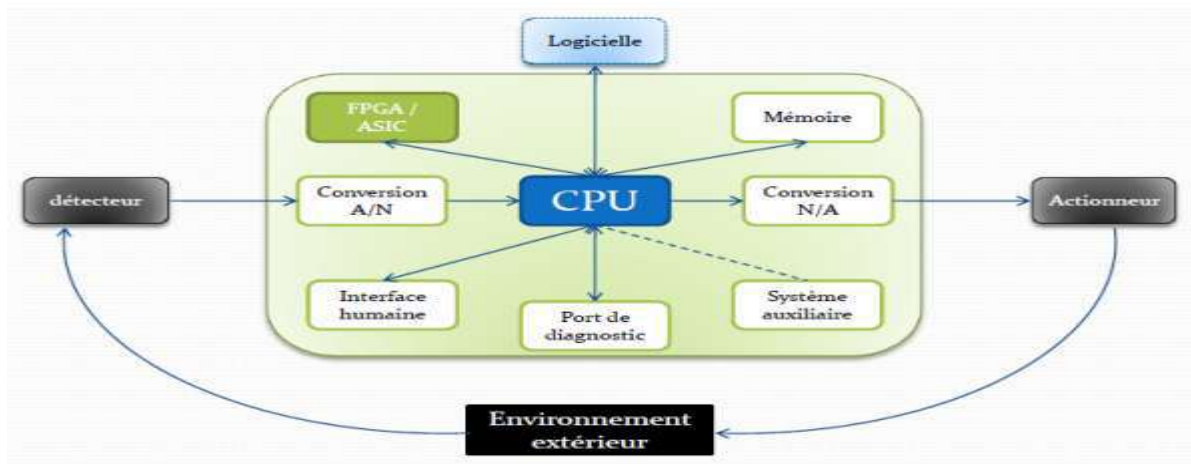


Figure 1: Architecture matérielle d'un système embarqué. [10].

On retrouve en entrée des capteurs généralement analogiques couplés à des convertisseurs A/N. On retrouve en sortie des actionneurs généralement analogiques couplés à des convertisseurs N/A. Au milieu, on trouve le calculateur mettant en œuvre un processeur embarqué et ses périphériques d'E/S. Il est à noter qu'il est complété généralement d'un circuit FPGA (Field-Programmable Gate Array) jouant le rôle de coprocesseur afin de proposer des accélérations matérielles au processeur.

On retrouve en fait un système d'asservissement entre les entrées et les sorties, Il est à noter que l'expression la plus simple de cette figure (1) est de considérer comme capteurs des interrupteurs et comme actionneurs des LED Sur ce schéma théorique se greffe un paramètre important : le rôle de l'environnement extérieur. Un système embarqué doit faire face à des environnements plus hostiles. Il doit faire face à un ensemble de paramètres agressifs :

- Variations de la température.
- Vibrations, chocs.
- Variations des alimentations.
- Corrosion.
- Eau, feu, radiations.

L'environnement dans lequel opère le système embarqué n'est pas contrôlé ou contrôlable. Cela suppose donc de prendre en compte ce paramètre lors de sa conception. On doit par exemple

prendre en compte les évolutions des caractéristiques électriques des composants en fonction de la température, des radiations..., les systèmes embarqués sont aujourd'hui fortement communicants. Cela est possible grâce aux puissances de calcul offertes par les processeurs pour l'embarqué (32 bits en particulier) et grâce aussi à l'explosion de l'usage de la connectivité Internet. La connectivité internet permet fondamentalement de contrôler à distance un système embarqué par Internet. Ce n'est en fait que l'aboutissement du contrôle à distance d'un système électronique par des liaisons de tout type : liaisons RS.232, RS.485, bus de terrain...

Cela permet l'emploi des technologies modernes d'Internet pour ce contrôle à distance par l'utilisateur : il suffit d'embarquer un serveur internet dans son équipement électronique pour pouvoir le contrôler ensuite à distance, de n'importe où, à l'aide d'un simple navigateur. Il n'y a plus d'interface humaine spécifique à concevoir pour cela, ce rôle étant rempli par le navigateur Internet. Cela est une réalité : les chauffagistes proposent maintenant des chaudières pouvant être pilotées par le web ! Il faut aussi noter la montée en puissance des communications sans fil dans l'embarqué au détriment des communications filaires pour limiter le câblage et faciliter la mise en place du système embarqué. Le wifi et toutes les normes de réseaux sans fil IEEE 802.15 comme Zigbee dans l'embarqué et surtout en domotique (réseaux de capteurs sans fil par exemple). Mais ne nous méprenons pas sur ces facilités et commodités, cela a bien sûr un revers : la sécurité du système embarqué, puisque connecté à Internet. [1]

Un système embarqué est composé de :

- Unité centrale de traitement (Central processing unit « CPU »)
- Les Convertisseurs analogiques – numériques (CAN)
- Les Convertisseurs Numériques – Analogiques (CNA)
- Les mémoires
- Les Circuits logiques programmables « FPGA

I.3 CIRCUIT LOGIQUE PROGRAMMABLE « FPGA »

Un circuit logique programmable, ou réseau logique programmable, est un circuit intégré logique qui peut être reprogrammé après sa fabrication. Il est composé de nombreuses cellules logiques élémentaires librement assemblables.

La plupart des grands FPGA modernes sont basés sur des cellules SRAM aussi bien pour le routage du circuit que pour les blocs logiques à interconnecter. FPGA jouant le rôle de coprocesseur afin de proposer des accélérations matérielles au processeur [1]

I.4 CARACTÉRISTIQUES PRINCIPALES D'UN SYSTÈME EMBARQUÉ

Les principales caractéristiques d'un système embarqué sont les suivantes :

- C'est un système principalement numérique.
- Il met en œuvre généralement un processeur.
- Il exécute une application logicielle dédiée pour réaliser une fonctionnalité précise et n'exécute donc pas une application scientifique ou grand public traditionnelle.
- Il n'a pas réellement de clavier standard (Bouton Poussoir, clavier matriciel...). L'affichage est limité (écran LCD...) ou n'existe pas du tout.
- Ce n'est pas un PC en général mais des architectures similaires (x86) basse consommation sont de plus en plus utilisées pour certaines applications embarquées. De ce constat, on peut voir.
- Qu'un PC standard peut exécuter tout type d'applications car il est généraliste alors qu'un système embarqué n'exécute qu'une application dédiée.
- Que des circuits numériques ou des circuits analogiques sont utilisés en plus pour augmenter les Performances du système embarqué ou sa fiabilité. [11].

II. LES MICROCONTRÔLEURS DANS LES SYSTÈMES EMBARQUÉS

Le processeur constitue le noyau principal d'un système embarqué. Il doit avoir un certain nombre de caractéristiques lui permettant d'accomplir sa tâche convenablement : Rapidité, Consommation, puissance....

La notion de puissance est la capacité de traiter un grand nombre d'opérations par seconde sur de grands nombres et en grande quantité. Intrinsèquement la puissance se joue donc sur les trois critères suivants:

- La longueur des mots : données et instructions (on parle de largeur du bus des données).
- Le nombre d'octets que le microprocesseur peut adressé (on parle de largeur du bus des adresses).
- La vitesse d'exécution des instructions liée à la fréquence de fonctionnement de l'horloge de synchronisation exprimée en MHZ. Sur ce il ya différent architecture et technologie d'un système de processeur dédié.

Les classifications des microcontrôleurs à deux niveaux :

- Au niveau du processeur:
 - RISC : Reduced Instruction Set Computer.
 - CISC : Complex Instruction Set Computer.
- Au niveau de l'organisation de la mémoire
 - Architecture Von Neumann : une mémoire unique pour le programme et pour les données.
 - Architecture Harvard : le programme et les données sont stockés dans des mémoires. **[16]**.

II.1 ORGANISATION DU PLAN MÉMOIRE :

Il existe 2 types de plan mémoire :

- architecture de type Von Neuman.
- architecture de type Harvard.

II.2 ARCHITECTURE DE VON NEUMANN ET ARCHITECTURE DE HARVARD

À chaque cycle d'horloge, le processeur sait par le compteur de programme l'instruction qu'il doit faire exécuter. Il va chercher chaque instruction en mémoire, l'exécute avec les données correspondantes et retourne les données résultantes en mémoire.

Dans l'architecture de la machine de Von Neumann, le programme et les données sont enregistrés sur la même mémoire. Chaque instruction contient la commande de l'opération à effectuer et l'adresse de la donnée à utiliser, il faut donc souvent plusieurs cycles d'horloge pour exécuter une instruction. La figure 2 indique une architecture simple de Von Neumann, constituée d'un bus de données et de programme et d'un bus d'adresses. [10].

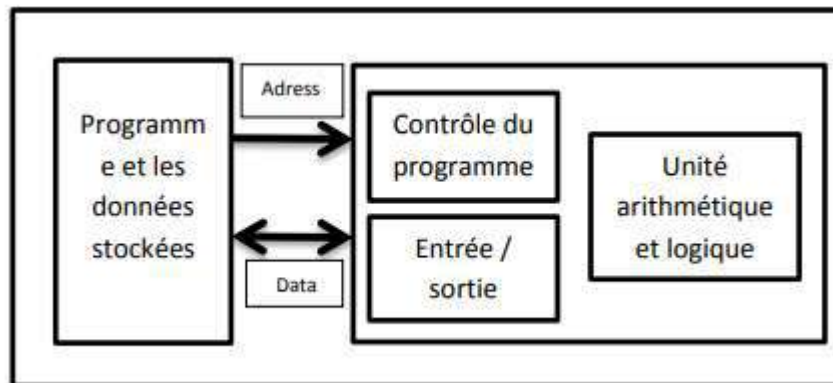


Figure 2: Architecture de Von Neuman. [2].

On voit que les échanges s'effectuent de manière simple entre l'unité arithmétique et logique (ALU), c'est-à-dire l'unité centrale et la mémoire unique, par un bus transitant les codes de programme et les données. On a ainsi des données « collées » aux instructions. Les microprocesseurs et beaucoup de microcontrôleurs utilisent cette architecture car elle est très souple pour la programmation, dans l'architecture dite de Harvard (car mise au point dans cette université américaine en 1930), on sépare systématiquement la mémoire de programme de la mémoire des données : l'adressage de ces mémoires est indépendant.

La figure 3 indique une architecture simple de Harvard constituée d'un bus de données, d'un bus de programme et de deux bus d'adresse. [10].

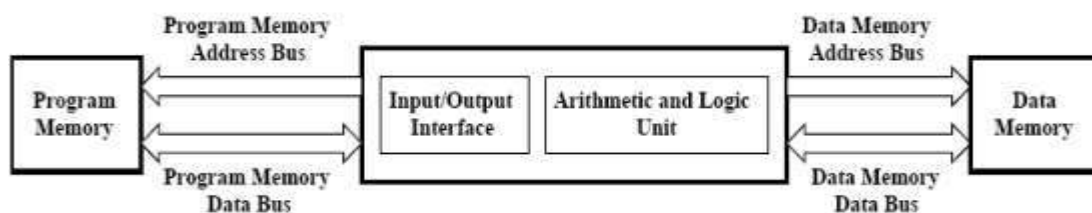


Figure 3: Architecture de Harvard [2].

On voit que les échanges s'effectuent de manière double entre l'unité centrale et les deux mémoires, ce qui permet une grande souplesse pour l'enregistrement et l'utilisation des données. D'ailleurs, la mémoire de programme est également utilisée en partie comme mémoire de données pour obtenir encore plus de possibilités de traitement avec des algorithmes complexes. [2]

L'architecture Harvard est souvent utilisée dans :

- les processeurs numériques de signal (DSP) ;
- les microcontrôleurs, notamment les PIC de Micro chip et les AVR d'ATmel.

II.3 ARCHITECTURE DE HARVARD MODIFIÉE DSP (DIGITAL SIGNAL PROCESSOR)

La structure de HARVARD est cependant pénalisante par le nombre de broches utilisées et le nombre de boîtiers mémoires utilisés comme de nombreux autres constructeurs (INTEL avec le MCS 51, MICROCHIP avec les microcontrôleurs PIC) utilise une structure de HARVARD modifiée, qui consiste à n'utiliser qu'un bus commun pour les accès en mémoire externe (architecture de VON NEUMANN) et à utiliser une structure de HARVARD pour les accès en mémoire interne. Charge ensuite au programmeur, si la taille de la mémoire programme ou de la mémoire données est trop importante, de placer en mémoire interne le bout de programme et de données qui demande une rapidité optimale d'exécution. [10]

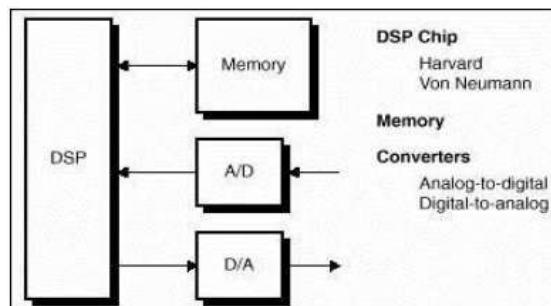


Figure 4: Structure de Harvard modifiée [2].

II.4 DÉFINITION D'UN MICROCONTRÔLEUR

(En notation abrégée μc , ou uc ou encore MCU en Anglais) est un circuit intégré qui rassemble les éléments essentiels d'un ordinateur : processeur, mémoires (mémoire morte pour le programme, mémoire vive pour les données), unités périphériques et interfaces d'entrées-sorties. Les microcontrôleurs se caractérisent par un plus haut degré d'intégration, une plus faible consommation électrique et un coût réduit par rapport aux microprocesseurs polyvalents utilisés dans les ordinateurs personnels.

Par rapport à des systèmes électroniques à base de microprocesseurs et autres composants séparés, les microcontrôleurs permettent de diminuer la taille, la consommation électrique et le coût des produits.

Les microcontrôleurs sont fréquemment utilisés dans les systèmes embarqués, comme les contrôleurs des moteurs automobiles, les télécommandes, les appareils de bureau, l'électroménager, les jouets, la téléphonie mobile, etc. [12].

II.5 COMPOSANTS INTÉGRÉS D'UN MICROCONTRÔLEUR

Un microcontrôleur intègre sur un unique circuit :

- Un processeur (CPU), avec une largeur du chemin de données allant de 4 bits pour les modèles les plus basiques à 32 ou 64 bits pour les modèles les plus évolués ;
- De la mémoire vive (RAM) pour stocker les données et variables ;
- De la mémoire morte (ROM) pour stocker le programme. Différentes technologies peuvent être employées : EPROM, EEPROM, mémoire flash (la plus récente) ;
- Souvent un oscillateur pour le cadencement, Il peut être réalisé avec un quartz, un circuit RC ou encore une PLL ;
- Des périphériques, capables d'effectuer des tâches spécifiques, On peut mentionner entre autres :
 - ✓ Les convertisseurs analogiques-numériques (CAN) (donnent un nombre binaire à partir d'une tension électrique).
 - ✓ Les convertisseurs numériques-analogiques (CNA) (effectuent l'opération inverse).

- ✓ Les générateurs de signaux à modulation de largeur d'impulsion (MLI, ou en anglais, PWM pour Pulse Width Modulation).
- ✓ Les timers/compteurs (compteurs d'impulsions d'horloge interne ou d'événements externes).
- ✓ les chiens de garde (watchdog).
- ✓ Les comparateurs (comparent deux tensions électriques).
- ✓ Les contrôleurs de bus de communication (UART, I²C, SSP, CAN, FlexRay, USB, Ethernet, etc.).

Certains microcontrôleurs ont un nombre très restreint de broches, si bien qu'une broche donnée peut correspondre à plusieurs périphériques internes. La fonction choisie doit alors être sélectionnée par logiciel.

Le choix des périphériques à intégrer dans un microcontrôleur est délicat. Les fabricants doivent réaliser un compromis entre des besoins contradictoires : utiliser des fréquences élevées, réduire la taille du circuit, apporter des fonctionnalités nombreuses, élaborer une architecture flexible, assurer des coûts modérés, etc.[14]

III .LE MICROCONTRÔLEUR ATMEL AVR (ATMEGA)

III .1 DÉFINITION

Les microcontrôleurs de la famille ATMEGA en technologie CMOS sont des modèles à 8 bits AVR (Architecteur Harvard Modifié) basés sur l'architecture RISC. En exécutant des instructions dans un seul cycle d'horloge simple, l'ATMEGA réalise des opérations s'approchant de 1 MIPS par MHz permettant de réaliser des systèmes à faible consommation électrique et simple au niveau électronique.[3]

III .2 LA FAMILLE AVR

La famille AVR 8bits (ATMEL) regroupe une centaine de composants, chacun disponible dans plusieurs types de boîtiers physique (package).

La famille AVR peut être découpée en 3 sous familles principales :

- Attiny AVR (8 à 20 broches).

- Méga AVR (32 à 100 broches).

- xMega AVR (44 à 100 broches).

Dans ce projet on va utiliser le **ATMEGA** qu'est considéré comme l'entrée de gamme de la famille Méga AVR.

III .3 LES CARACTÉRISTIQUES DU μ C ATMEGA32

La série ATMEGA fournit les particularités suivantes :

- Une FLASH RAM de 32 Ko (Flash Programmable).
- Une EEPROM de 1 Ko (Electrique Ecriture Programmable ROM),
- Une RAM de 2 Ko octets (Random Access Memory).
- 40 lignes d'entrée-sortie universelles (Port A, B, C, D,).
- 32 registres de travaux universels qui dialoguent directement avec l'unité centrale (ALU).
- Une horloge en temps réel (RTC).
- Trois Timer/Compteurs flexibles avec comparaison des modes, interruption internes et externes.
- Un comparateur analogique (entrée sur PB2, PB3).
- Un convertisseur Analogique/Numérique ADC de 8 canaux à 10 bits.
- Un Chien de Garde programmable avec oscillateur interne.
- Une ou deux interfaces série USART programmable & périodique (série asynchrone et synchrone).
- Une interface la série SP trois modes sélectionnables (série synchrone).
- Une interface I2C pour la gestion d'un bus à 2 fils, Une interface LCD pour piloter directement un écran LCD (modèles spéciaux).
- Tension d'alimentation de 2,7 V à 5,5 V. [3]

III .4 L'ARCHITECTURE INTERNE DE L'ATMEGA32

Le cœur AVR (Architecteur Harvard Modifié) combine un jeu de 131 instructions riches avec 32 registres spéciaux travaillant directement avec l'Unité Arithmétique et Logique ALU, qui représente le registre d'accumulateur A(B ou D) dans les microcontrôleurs classiques.[3]

Nous abordons le vif du sujet avec le synoptique qui présente le fonctionnement général du microcontrôleur ATMEGA :

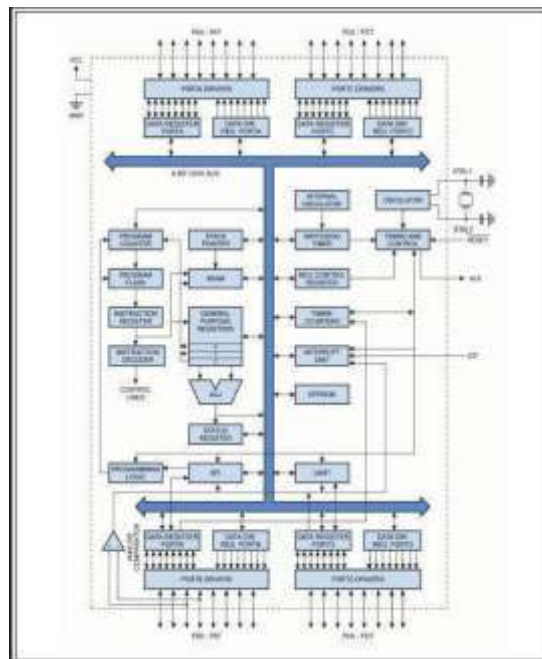


Figure 5: Synoptique générale d'un ATMEGA 32 [3]

III .5 PRÉSENTATION PHYSIQUE

L'ATMEGA se présente sous la forme d'un circuit intégré à 40 broches pour le modèle ATMEGA32 en boîtier PDIP ou le boîtier TQFP/MLF à 44 broches

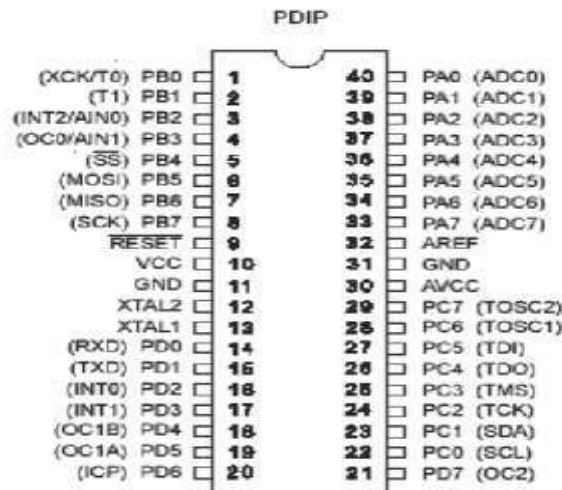


Figure 6: Brochage typique d'un ATMEGA32 en boîtier PDIP et PLCC [3]

III .6DESCRIPTIONS DES BROCHES

Port A (PA7.. PA0) le Port A est un port d'entrée-sortie à 8 bits bidirectionnel avec des résistances internes de tirage (choisi pour chaque bit). Il sert aussi pour les entrées analogiques du convertisseur A/D. Le Port A (comme le B, C et D) est en position trois états quand une condition De reset devient active, même si l'horloge ne court pas.

Port B (PB7.. PB0) le Port B est un port d'entrée-sortie à 8 bits bidirectionnel avec des résistances internes de tirage (choisi pour chaque bit). Il sert aussi de comparateur analogique (sortie sur PB2, PB3), ou de SPI.

Port C (PC7.. PC0) le Port C est un port d'entrée-sortie à 8 bits bidirectionnel avec des résistances internes de tirage (choisi pour chaque bit). Il sert aussi comme oscillateur pour le Timer/Compteur2 et d'interface I2C.

Port D (PD7.. PD0) le Port D est un port d'entrée-sortie à 8 bits bidirectionnel avec des résistances internes de tirage (choisi pour chaque bit). Il sert aussi d'USART et d'entrées pour les interruptions externes.

RESET déclenché par un front descendant maintenue plus de 50 ns il produira le Reset du microcontrôleur, même si l'horloge ne court pas.

XTAL1 Entrée de l'oscillateur externe ou libre pour l'horloge interne.

XTAL2 Production de l'amplificateur d'oscillateur.

AVCC est une broche de tension d'alimentation pour le Convertisseur A/D qui doit être connectée à VCC via un filtre passe-bas pour éviter les parasites.

AREF est l'entrée de référence analogue pour le Convertisseur A/D avec une tension dans la gamme de 2 V à AVCC avec filtre passe bas.

AGND masse Analogique. Si la masse analogique est séparée de la masse générale, brancher cette broche sur la masse analogique, sinon, connecter cette broche à la masse générale GND.

VCC broches d'alimentation du microcontrôleur (+3 à +5V).

GND masse de l'alimentation.[4]

IV.CONCLUSION

Dans ce chapitre, nous avons présenté d'une manière les systèmes embarqué, leur principes de fonctionnement et leurs architecture, ainsi nous avons vu une étude détaillée sur les microcontrôleurs et leur architecture et organisation de leur mémoire, après on s'est intéressé au microcontrôleur ATMEL AVR (ATMEGA 32) conçu en technologie la plus récente, avec des composants RISC et une Architecture Harvard Modifiée.

Chapitre

2

Les logiciels et matériels utilisé

I.INTRODUCTION

Les outils et environnements de travail Dans ce chapitre, nous allons détailler l'ensemble des outils matériels et logiciels ayant permis la mise en place du système. Nous parlerons des langages de programmation, de logiciels, de carte électronique.

Pour la simplicité et la facilité de la programmation, plusieurs langages ont été créés. En cherchant le compilateur le plus adapté aux microcontrôleurs "AVR" d'Atmel, on trouve BASCOM-AVR.

II LES LOGICIELS

II.1LE COMPILATEUR BASCOM-AVR POUR AVR" D'ATMEL

II.1.1 DÉFINITION

Spécialement développés pour les microcontrôleurs "AVR" d'Atmel™ dotés d'une mémoire interne, le "BASCOM-AVR" est un super compilateur BASIC qui vous permettra de travailler très efficacement et rapidement.

Ces nombreuses instructions "spécialisées" et son mode "simulation" en font un formidable outil de développement. [5]

II.1.2 PRINCIPALES CARACTÉRISTIQUES

- ✓ Langage "BASIC" structuré (gestion d'étiquettes)
Gestion des boucles conditionnelles: IF - THEN - ELSE - ENDIF SELECT – CASE.
- ✓ Gestion inconditionnelles: DO-LOOP, WHILE-WEND, FOR ... NEXT (avec sortie possible).
- ✓ Possibilité d'utiliser des noms "longs" de variables et d'étiquettes (jusqu'à 32caractères).
- ✓ Gestion de variables de type divers:

Bit (1 ou 0), Byte (8 bits),Integer (16 bits),Word (16 bits signés), Long(32bits),Single (32 bits signé),String,Tableau.
- ✓ Accès à des fonctions mathématiques à virgule flottante.
- ✓ Variables locales dans les sous-routines ou fonctions-utilisateur.
- ✓ Editeur avec indicateur d'erreurs (par surlignement).
- ✓ Aide contextuelle en ligne. [5]

II.1.3 FONCTIONS SPÉCIALES

- Commandes d'afficheurs LCD alphanumériques (1 x 16 à 4 x 40 car. - mode 4 / 8 bits).
 - Commandes d'afficheurs LCD graphiques (128 x 64 - 240 x 64 - 240 x 128 - 128 x 128).
 - Dialogue possible avec composants externes via communication I2C™ - 1 Wire™.
 - Gestion de claviers matricés, de clavier de PC.
 - Gestion de communication infrarouge au standard RC5.
 - Gestion de communication RS-232 et SPI.
 - Interfaçage TCP/IP (en option avec composant W3100A).
 - Terminal d'émulation intégré avec option de téléchargement.
 - Simulateur intégré.
 - Logiciel de programmation ISP intégré (nécessite programmeur externe).
 - Gestion de librairie externe vous permettant d'ajouter de nombreuses fonctions par Add-on.
- [5]

II.1.4 PRÉPARATION DE BASCOM

Il faut préparer Bascom pour utiliser le μ -contrôleur choisi et la configuration LCD proposée : Pour ce faire, on va ouvrir le menu OPTIONS puis Compiler, puis on va choisir le μ - contrôleur dans le sous-menu Chip, on en profitera pour configurer les ports du LCD. [15].

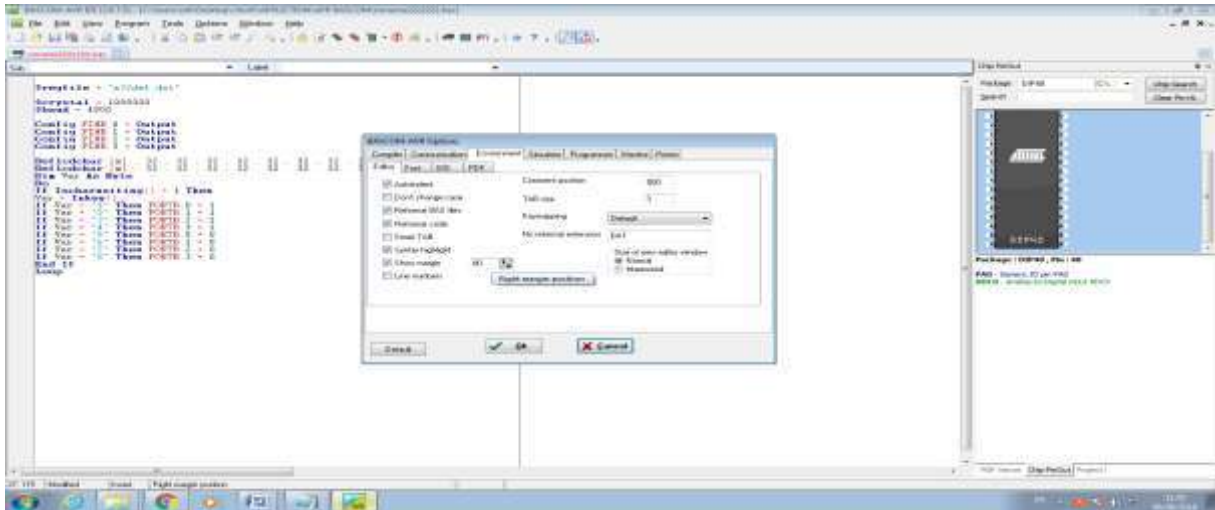


Figure 7: Détermination de microcontrôleur utilisé

Le LCD

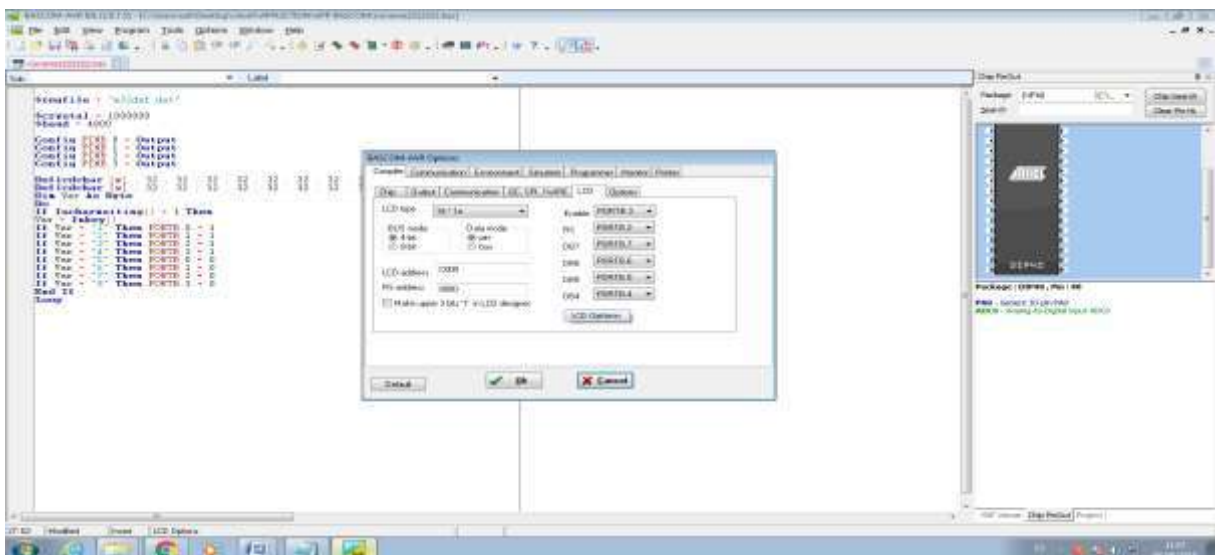


Figure 8 : Configuration de LCD

II.1.5 ENREGISTREMENT, COMPILATION, CONTROLE

On commence par une sauvegarde, en tapant Ctrl+S ou en choisissant l'icône représentant une disquette.

Compilation F7

Si le compilateur ne trouve pas d'erreur on passe au choix suivant, sinon en tapant sur l'erreur, la (es) ligne(s) en faute est surlignée. Le compilateur génère les fichiers demandés dans le menu Options\Compiler\ Output, parmi ces fichiers nous trouverons celui ou ceux qui seront transmis au μ -contrôleur.

Contrôle

CTRL+F7 Syntax Check Permet de contrôler l'ensemble du programme sans le compiler.

II.1.6 LE SIMULATEUR DU "BASCOM-AVR"

Une des principales caractéristiques du compilateur "BASCOM-AVR" réside dans la possibilité qu'il vous offre de pouvoir tester votre programme à l'aide de son simulateur intégré, ce dernier vous permettra d'exécuter votre programme en mode "pas-à-pas" ou jusqu'à une ligne particulière.

Le programme est compilé, sans erreur ? On peut simuler le fonctionnement avec le menu PROGRAM\SIMULATE ou avec la touche F2 ou encore avec la petite icône.

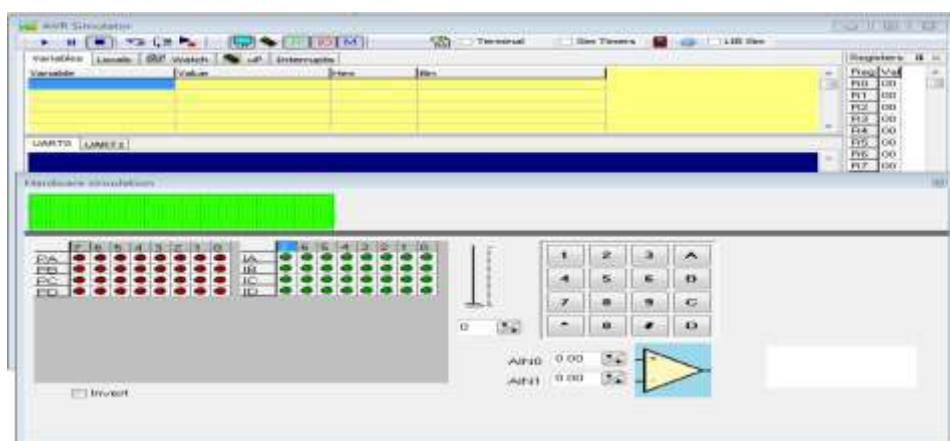


Figure 9: Le simulateur du "BASCOM-AVR".

III .1L'OUTIL APP INVENTOR

App Inventor est un outil de développement des applications en ligne pour les smart phones sous androïde et permet à chacun de créer son application personnelle pour le système d'exploitation Androïde qui est développée par Google. La plateforme de développement est offerte à tous les utilisateurs possédant un compte Gmail. Elle rappelle certains langages de programmation simplifiés basés sur une interface graphique similaire à Scratch. Les informations des applications sont stockées sur des serveurs distants. [7]

Elles sont actuellement entretenues par le Massachusetts Institute of Technology (MIT).

L'environnement de App Inventor contient trois fenêtres qui sont proposées pendant le développement :

- Une pour la création de l'interface homme machine : permet de créer l'allure de l'application (App Inventor Designer) ;
- Une pour la programmation par elle-même : elle permet, par l'assemblage des blocs de créer le comportement de l'application (App Inventor Block Editor) ;
- Une pour l'émulateur : qui permet de remplacer un terminal réel pour vérifier le bon fonctionnement du programme.

La connexion d'un terminal réel sous Androïde permettra ensuite d'y télécharger le programme pour un test réel. Ce terminal pourra aussi bien être un téléphone qu'une tablette ; le comportement du programme sera identique. [13]

III .2HISTORIQUE DE LOGICIEL APP INVENTOR

2009 : Début du développement du logiciel App Inventor par Google à partir de recherches dans l'enseignement de l'informatique faites par le MIT, Boston près de New-York. L'Objectif de l'enseignement permet à des étudiants qui débutent en informatique d'apprendre à programmer sans se noyer sous le code Java.

2011: Google rend App Inventor open source. Le MIT poursuit le développement.

2012: Version bêta d'App Inventor diffusé par le MIT encore en version bêta aujourd'hui.

III .3LANGAGE JAVA

JAVA est un langage de programmation orienté objet, développé par Sun Microsystems, sorti en 1995. Sun Microsystems est racheté en 2009 par Oracle, une application écrite en JAVA et facilement portable sur plusieurs systèmes d'exploitation. Une application exécutable sous Androïde (interprétable par une interface en JAVA) est un fichier avec l'extension « APK » [8]

III .4STRUCTURE D'UNE APPLICATION APP INVENTOR

Une application développée sous App Inventor est constituée de deux parties distinctes mais étroitement liées.

A. L'interface graphique : Pour créer l'application sous App Inventor l'interface graphique contient nos propriétés (taille, couleurs, position, textes Ets). Cette partie contient des composants (visibles ou invisibles) [7].

Cette interface graphique contient quatre parties :

Partie 1 : Une palette sous App Inventor contenant tous les éléments qui peuvent être positionnés sur l'écran du téléphone.



Figure 11: Les composants graphiques

Partie2 : C'est la surface du téléphone ajusté automatiquement par app inventor ou manuellement par nous même en utilisant le composant « Screen arrangement ».

Partie 3 : La liste des éléments et des médias utilisés sur l'écran.

Partie 4 : Les propriétés des différents éléments utilisés par exemple la couleur et la taille du bouton ou texture.

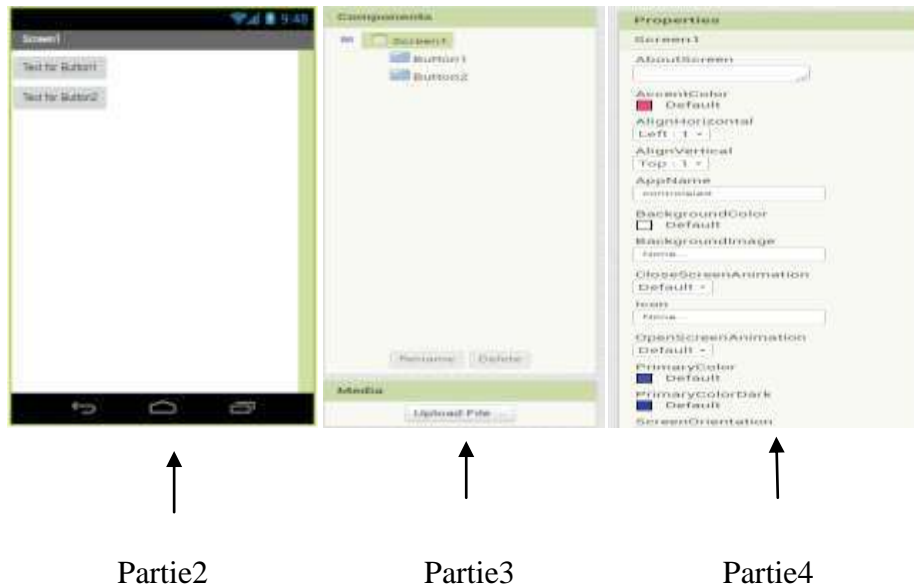


Figure 12 Les propriétés et les classes sous App Inventor

Pour ajouter un bouton sur l'écran d'émulateur virtuel, on clique sans lâcher sur le mot « Bouton » dans la palette en déplaçant la souris sur Viewer et relâcher le bouton et là un nouveau bouton va apparaître sur le Viewer.

Pour ajuster la place du bouton, on utilise la propriété Width de notre exemple ; on utilise le paramètre Fill parent. Pour utiliser le bouton sur la largeur de l'écran.

B. Editeur de blocs (Fenêtre Scratch) Une fois les composants de l'écran de téléphone mis en place et désigné, nous passons à la deuxième phase de développement d'une application via App Inventor : l'interface Scratch, pour cela, il faut cliquer sur «Open the Blocks Editor» en haut à droite de la page.

L'interface Scratch permet d'imbriquer des éléments graphiques entre eux pour effectuer la partie programmation de l'application à développer. De cette partie, on peut assembler les différents blocs de l'application et indiquer comment les composants doivent se comporter et qui s'affichent dans l'émulateur virtuel (par exemple ; ce qui se produit quand un

utilisateur clique un bouton) déterminant le fonctionnement même de l'application, en réaction à des événements (internes ou externes) ou à des réponses. [8]



Figure 13: En-tête d'éditeur de blocs App Inventor

La fenêtre à gauche contient les blocs qui assemblent la partie droite de la fenêtre pour décrire le comportement de notre application. Les blocs peuvent être standards (dans l'onglet « Built-in ») ou définis spécifiquement pour l'application. [8]

IV. LES MATÉRIELLES

Étude de la partie matérielle On va aborder dans cette partie les différents organes utilisés dans notre système. Nous allons décrire les éléments qui ont composé notre carte.

IV. 1 PLAQUE D'ESSAI

Une plaque d'essai permet de réaliser des montages électroniques sans soudure. En règle générale les plaques d'essais sont de forme rectangle. Il y a plusieurs rangés de trous: certaines rangés sont verticale tandis que d'autres sont horizontal.

Plusieurs modèles existent, nous utiliserons des plaques d'essai comme celle représentée ci-dessous. La plaque d'essai comporte des connexions cachées, chaque bande de cuivre met en contact 5 trous. Les trous sont espacés exactement de 2,54 mm (un dixième de pouce).

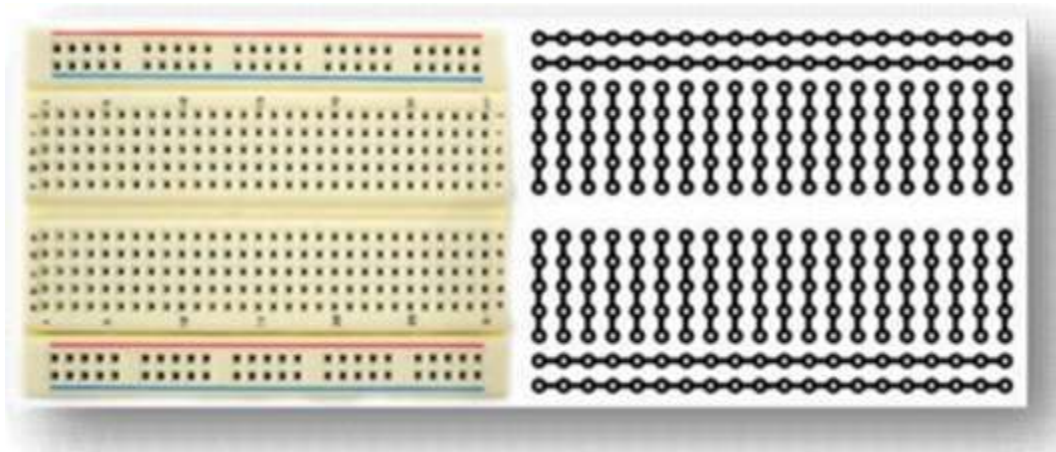


Figure 14: Plaque d'essai [14].

IV. 2LES STRAPS

Qui sont des fils de cuivre isolent, de longueur et couleur variables.



Figure 15: les straps. [15].

IV .3AFFICHEUR LCD

Les afficheurs à cristaux liquides, autrement appelés afficheurs LCD (Liquid Crystal Display), sont des modules compacts intelligents et nécessitent peu de composants externes pour un bon fonctionnement. Ils consomment relativement peu (de 1 à 5 mA), sont relativement bons marchés et s'utilisent avec beaucoup de facilité. [15].

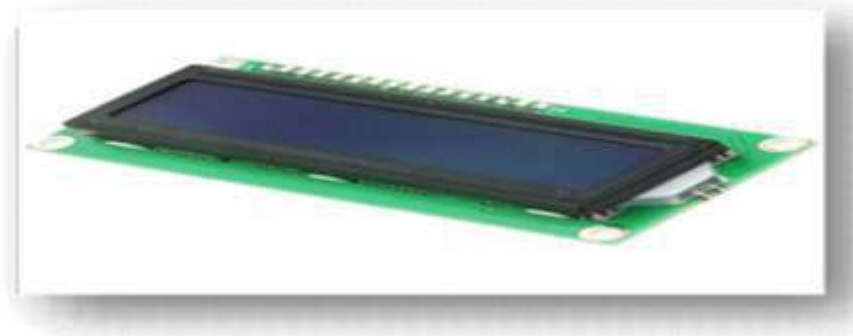


Figure 16: Afficheur LCD. [15].

IV .4MODULE DE RELAIS À 4 CANAUX

Il s'agit d'une carte d'interface de relais, qui peuvent être contrôlé directement par un large éventail de microcontrôleurs comme Arduino, AVR, PIC, ARM, API, etc. Ce module de relais est bas actif 5V. Il est également capable de contrôler appareils divers et autres équipements avec le grand courant. Cette interface standard peut être connectée directement avec les microcontrôleurs. Le voyant rouge qui indique l'état de travail est propice à l'utilisation de sécurité. Le module de relais est largement utilisé pour tout contrôle MCU, le secteur industriel, contrôle PLC, contrôle de la maison intelligente. Dans notre cas, nous avons choisi un module de relais à 8 canaux car nous avons besoin de contrôler de l'éclairage de 4 pièces ainsi de la ventilation. [15].



Figure 17: Module de Relais à 4 canaux[15].

V. CONCLUSION

Après identification du langage bascom et logiciel App inventor. Nous considérons l'un des meilleurs logiciels pour programmation et communication des microcontrôleurs. Par conséquent, nous allons utiliser ces deux logiciels Après l'installation des composants de la carte pour réaliser notre projet d'étude.

Chapitre

3

Conception de la carte et Réalisation

I.INTRODUCTION

Dans ce chapitre, nous allons décrire le processus de réalisation de notre système qui n'est rien d'autre que la mise en application de ce qui a été énoncé dans le chapitre précédent. Ceci en mettant en évidence l'ensemble des environnements (logiciels et matériels) de développement, le fonctionnement du système, ainsi qu'un aperçu sur les interfaces de l'application.

Nous avons acheté tous les composants électroniques, et On commence par la construction de la carte et relier les différents composants.

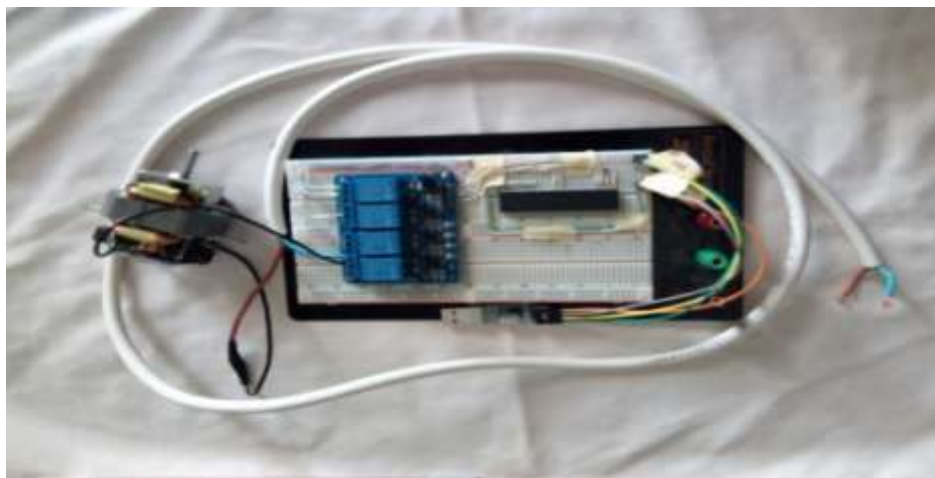


Figure 18: Installation des composants électrique sur la carte d'essai

II. 2.SAISIE DU PROGRAMME :

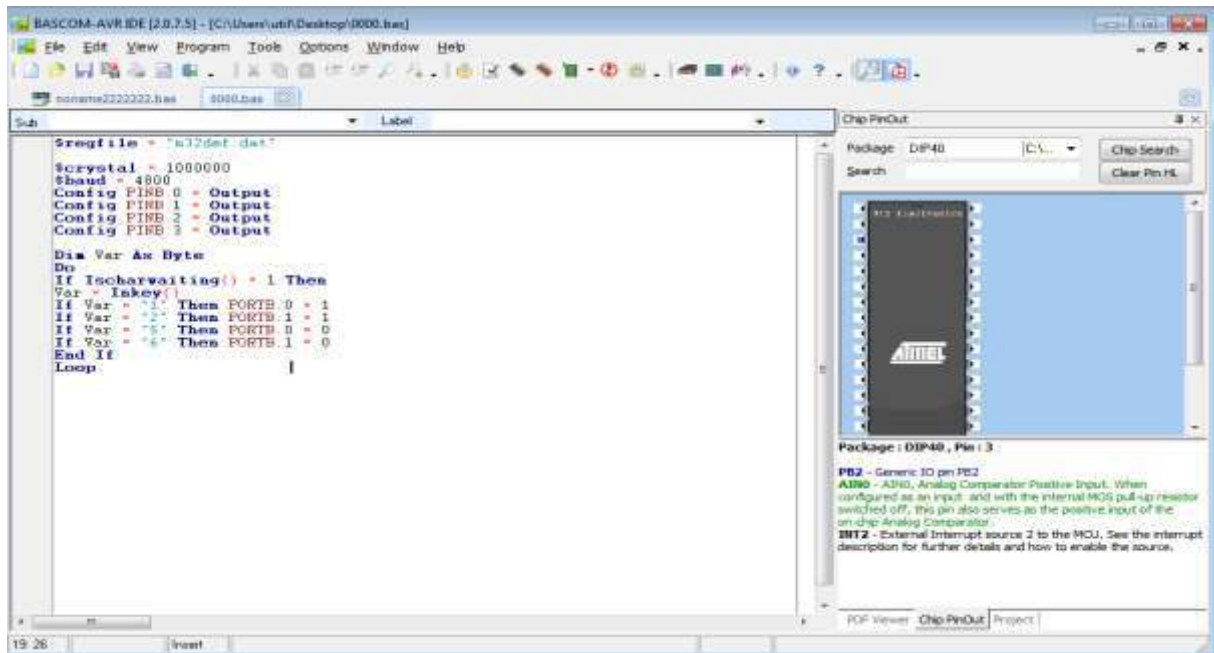


Figure 20: Exemple de programme.

II. 3COMPILATION : COMPILER F7

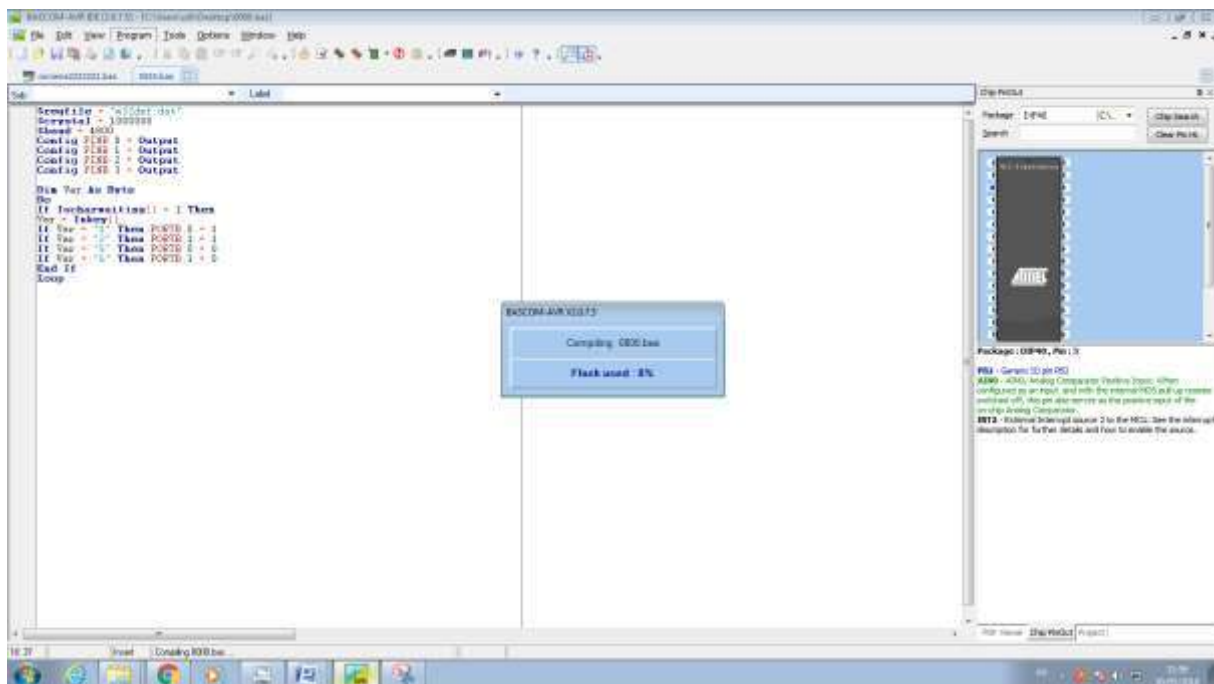


Figure 21: Compilation de programme.

1. Si tout est correct, un fichier en format hexadécimal (.hex) sera créé. Cette étape est indispensable pour que le programme soit reconnu et exécuté par le microcontrôleur

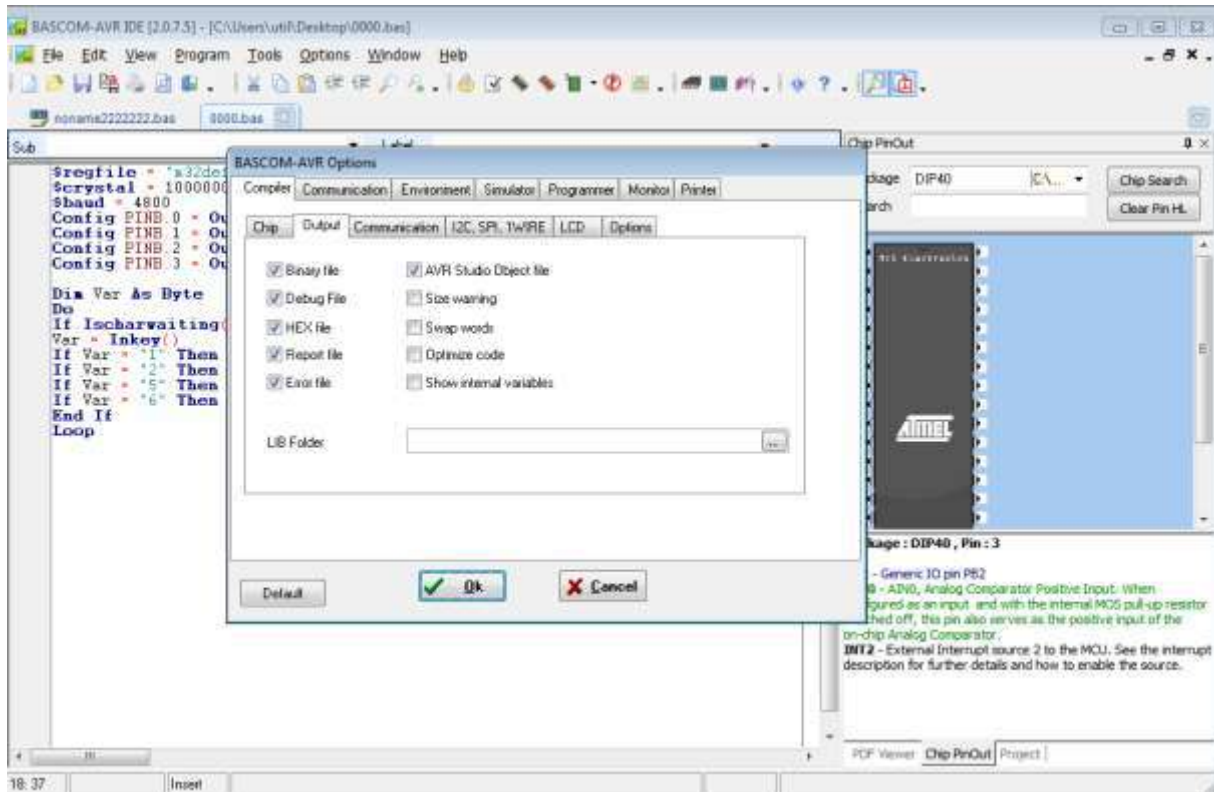


Figure 22: Création de fichier (.hex).

II. 4LE TRANSFÈRT VERS LE MICROCONTRÔLEUR.

Avant de transférer le programme vers le μ -contrôleur, il faut configurer le programmeur avec l'Option Programmer.

On transfère le programme compilé par l'intermédiaire du sous menu Program \ send to chip avec le raccourci clavier F4 ou par l'icône de transfert.

III. CRÉATION APPLICATION ANDROÏD

Ensuite on va créer une application androïde, on utilise l'outil de développement des applications en ligne pour les smart phones sous androïde **App Inventor**.

Maintenant, nous allons réaliser l'application Android :

1. Se connecter à Internet.
2. Ouvrir notre navigateur et se connecter au compte Google.
3. Rendez-vous ici : <http://appinventor.mit.edu/explore/>. Cela peut-être assez long parfois car App Inventor est lancée depuis le Web en mode Cloud.
4. Créer un nouveau projet : Cliquer sur My Project (en haut à gauche) / New / Project Name (sans espace) / OK. Notre exemple (Led_control).

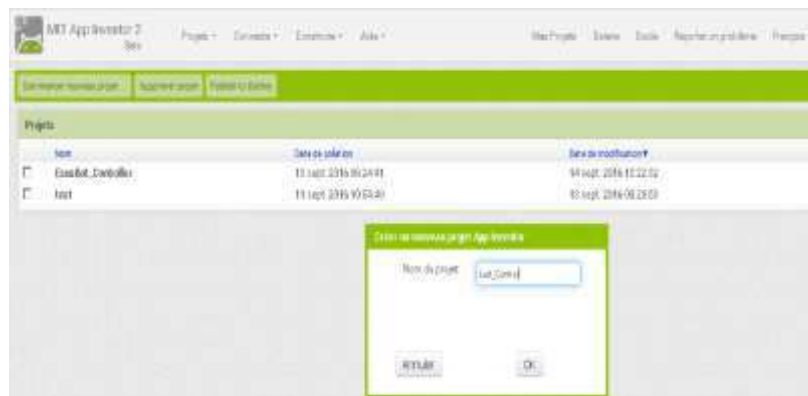


Figure 23: Création de nouveau projet sur App Inventor.

Vous devez arriver sur l'interface de votre application :



Figure 24: Première interface de la création App Inventor.

- Ajouter les boutons :

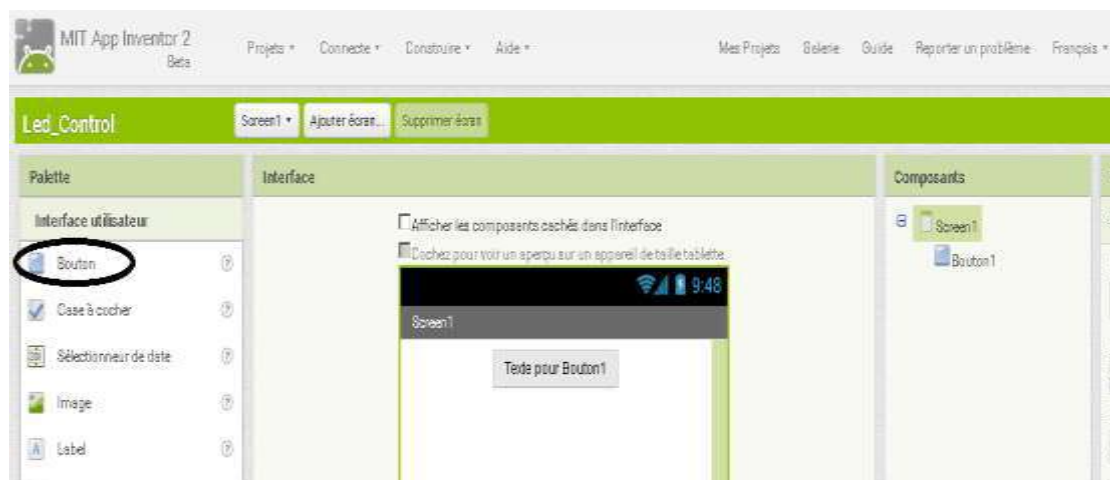


Figure 25: l'ajout d'un bouton sous App Inventor.

- Modifier le texte du bouton : **LED ON**
- Ajouter un deuxième bouton avec : **LED OFF**

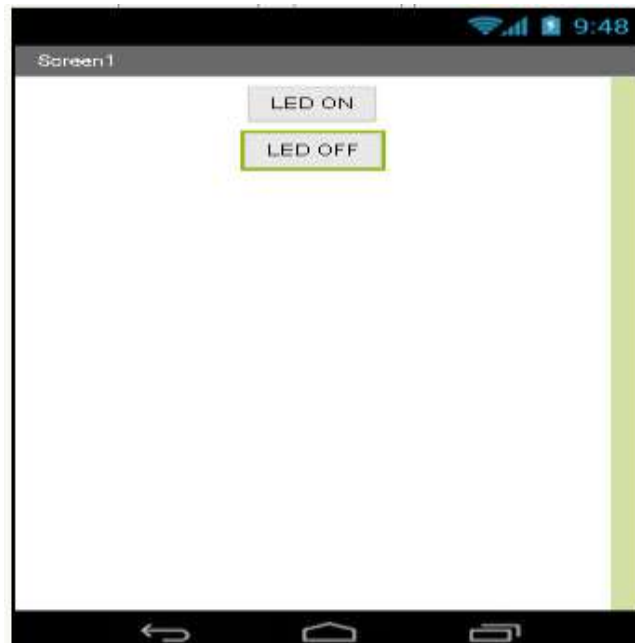


Figure 26: Schéma global du Screen 1.

- **Réalisation des blocs :**

Voici le programme en bloc pour la gestion des deux boutons :



Figure 27: Un bloc pour la gestion du bouton ON et un autre pour le bouton OFF.

- Construction de l'application sur votre ordinateur :



Figure 28: Fenêtre d'enregistrement de l'application.

- **Transférer l'application vers le smartphone.**

Autoriser l'installation de l'application sur votre téléphone.

Cette figure montre que notre application nommée « CCN 4 MOTEUR » peut être installée sur smart phone, sa taille est de deux mégas comme première version.



Figure 29: Photo réelle de notre application CCN 4 MOTEUR.



Figure 30 :Interface PC

En outre, nous avons développé une interface pc on utilise le langage visuel basic. sur cette figure nous trouvons l'interface d'accueil de notre application invitant l'utilisateur à faire un choix entre deux méthodes de connexion, une via WIFI, GSM, et une autre en local.

CONCLUSION GÉNÉRALE ET PERSPECTIVES

Notre mémoire a bien montré les techniques essentielles qu'il faut avoir pour maîtriser la commande par les microcontrôleurs, ce dernier joue un rôle important dans les systèmes embarqués.

Un système embarqué est défini comme un système électronique et informatique autonome, souvent temps réel, spécialisé dans une tâche bien précise. Le terme désigne aussi bien le matériel informatique que le logiciel utilisé, Ses ressources sont généralement limitées, le système à réaliser est considéré comme un system embarqué.

Dans ce travail, un assemblage de plusieurs logiciels est fait afin de réaliser un tel résultat.

En effet, ce projet a été bénéfique pour notre formation il nous a permis de :

- ✓ Comprendre la philosophie des systèmes embarqués, leur architecture et leur position Risc vis à vis des systèmes électroniques standards en termes d'utilisation de processeur.
- ✓ Programmer modestement en langage BASCOM la famille des microcontrôleurs AVR d'Atmel.
- ✓ Connaitre la famille des microcontrôleurs AVR, connue par leur richesse en terme d'élément d' E / S et possibilités de programmabilité et souplesse de développement et de mise en over pratique.

Une telle réalisation n'est pas dénuée de difficultés. Il est à noter que nous nous sommes confrontés à plusieurs problèmes surtout dans la partie réalisation de la carte. Cependant, on peut dire que malgré ces difficultés, les résultats obtenus à travers cette étude qu'ils soient pratiques ou théoriques, permettent d'ouvrir la porte à d'autres études.

Comme **perspectives** nous proposons :

- La commande du microcontrôleur à distance via Internet.
- Utilisation la cartes réalisée pour commander d'autres systèmes.
- Nous souhaitons vivement que ce projet puisse servir comme élément de base pour d'autres études plus approfondies pour le faire intégrer sous des systèmes plus complexes.

Bibliographie

- [1] : L- TAYARI. 2014-2015. Support de cours systèmes embarqués. [En Ligne]. Disponible sur : <http://systemesembarques.e-monsite.com/medias/files/cours-systemes-embarques2015.pdf> (Consulté le15/12/2017).
- [2]: Architecture de Von Neumann et architecture de Harvard technique - (<http://www.scrigroup.com/limba/franceza/159/Architecture-de-Von-Neuman-et-41539.php>) (Consulté le20/01/2018).
- [3]: JUILLOT Guillaume. 2 septembre 2003. La programmation des ATMEL AVR. [En Ligne]. Disponible sur : <https://les-electroniciens.com/sites/default/files/cours/programmationavr.pdf>(Consulté le10/03/2018).
- [4]: Walid Balid 2011-2012.Microprocessors & Microcontrollers Lab Sessions. [En Ligne]. Disponible sur : <https://download-engineering-pdf-ebooks.com/14438-free-book> (Consulté le03/03/2018).
- [5] : Jean-Pierre Duval. Basic des microcontrôleurs. [En Ligne]. Disponible sur : <file:///C:/Users/util/Desktop/memoire/bascomfr-livre2.pdf> (Consulté le15/03/2018).
- [6] : <https://fr.wikipedia.org/wiki/Andro%C3%AFde>(Consulté le20/03/2018).
- [7] : Terminale STMG SIG.2013-2014. DÉBUTER AVEC APP INVENTOR. [En Ligne]. Disponible sur : http://sig.fgranotier.info/IMG/pdf/debuter_app_inventor.pdf(Consulté le03/05/2018).
- [8] : https://fr.wikipedia.org/wiki/Java_%28langage%29(Consulté le25/03/2018).
- [9] Cours Systèmes Embarqués : Introduction. [En Ligne]. Disponible sur : <http://www.technologuepro.com/cours-systemes-embarques/cours-systemes-embarquesintroduction.htm> (Consulté le 25/03/2018).
- [10] Architecture d'un système embarqué [En Ligne]. Disponible sur : <http://www-igm.univmlv.fr/~dr/XPOSE2002/SE/architecture.html> (Consulté le 27/03/2018).
- [11] R- Grisel, N-Abouchi. Les systèmes embarqués Introduction. [En Ligne]. Disponible sur : http://richard.grisel.free.fr/Master_OSM/2_Introduction_Embedded_systems.pdf (Consulté le 18/04/2018).

- [12] A- Doncescu. Introduction aux Systèmes Embarqués et Microcontrôleurs. [En Ligne]. Disponible sur : <https://homepages.laas.fr/adoncesc/SystemEmbed/Cours1-introd.pdf> (Consulté le 18/04/2018).
- [13] A-Krama, A-Gougui. Etude et réalisation d'une carte de contrôle par Arduino via le système Androïde.2014/2015. Université Kasdi Merbah Ouargla.
- [14] A-Brahimi et H- Guezouli. Etude et réalisation d'une carte de commande à base d'un microcontrôleur PIC 16f877 pour ponts redresseurs triphasés à thyristors. 2014. Université Abou-Bekr Belkaïd. Tlemcen.
- [15]G-Berthome. L'afficheur LCD. [En Ligne]. Disponible sur :http://gilles.berthome.free.fr/02-Syntheses/EConversion_electriques_physiques/Synthese_afficheur_LCD.pdf (Consulté le 20/04/2018).
- [16] Architecture d'un système embarqué [En Ligne]. Disponible sur : <http://www-igm.univmlv.fr/~dr/XPOSE2002/SE/architecture.html> (Consulté le 27/05/2018).