

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البلدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Projet de Fin d'Études

présenté par

SMAIN Mohamed Abdelilah

&

GHIDA Said

pour l'obtention du diplôme de Master en Électronique spécialité Automatique

Thème

Etude et réalisation d'un automate programmable didactique

Proposé par : Dr CHENTIR Amina

Année Universitaire 2012-2013



REMERCIEMENTS

*Nous remercions **ALLAH** tout puissant qui nous a donné le courage,
la force et la volonté pour réaliser ce modeste travail.*

*Nous remercions tout particulièrement Melle **CHENTIR AMINA**, pour la
proposition de ce sujet, son encadrement, sa disponibilité et son soutien dans notre
recherche d'information et ses conseils lors de l'élaboration du projet.*

*Nous remercions aussi l'ensemble du personnel du laboratoire **LATSI** et
l'association scientifique **Science et Horizon** pour nous avoir permis de réaliser ce
projet dans de bonnes conditions.*

*Enfin, nous remercions tous ceux qui ont contribué de près ou de loin, par leurs aides
et conseils, à finaliser ce projet.*



Dédicace

*Je dédie ce modeste travail à mes chers parents qui ont fait de moi
ce que je suis.*

A mon frère et ma sœur.

A mes Oncles, Tantes et cousins et alliés de la famille.

*ET à ma promotrice **CHENTIR AMINA.***

*En ce moment, je ne peux oublier
l'ensemble des amis que j'ai connu pendant mes études et à
ceux qui ont prodigué leurs vifs conseils, encouragements et
témoigné de leur amitié.*

Dédicaces

A la mémoire de mon père, sans toi papa mes joies ne seront jamais complètes, mais j'essaye toujours d'avancer dans mes études comme tu as toujours souhaité, j'espère que je suis devenu le fils que tu voulais avoir, et que tu seras fière de moi là ou tu es.

A celle qui m'a toujours poussé et motivé dans mes études. Sans elle, je n'aurais certainement pas fait d'études, Ce premier fruit de ma réussite représente donc l'aboutissement du soutien et des encouragements qu'elle m'a prodigués tout au long de ma scolarité, et de l'équilibre qu'elle a su maintenir durant toutes ces années.

A ma chère maman

A l'adorable sœur Kheira qui m'encourage tout le temps, et mon frère Rafik qui me fait rire, pendant tout les moments de fatigue et de stress.

Une dédicace spéciale à nesrine, celle qui été toujours avec moi par ces encouragements, ces soutiens et motivations.

A ma grand-mère et toutes mes tentes et mes oncles.

A mes amis Amine, Walid, Amine et tous les amis de la promo.

Et enfin je n'ai pas clôturé cette page sans te dire Merci mon chère binôme Said, pour le partage, la patience et la compréhension.

Abdelilah.

Sommaire

Remerciements

Dédicaces

Résumés

Liste Des Figures Et Tableaux

Liste Des Abréviations

Sommaire

Introduction Générale.....1

Chapitre 1 : Généralités

1.1	Introduction.....	4
1.2	Définition.....	4
1.3	Application des automates programmables.....	5
1.4	Architecture des automates programmables.....	5
1.4.1	Aspect extérieur.....	5
1.4.2	structure interne.....	6
1.4.2.1	Module d'alimentation.....	6
1.4.2.2	Le processeur.....	7
1.4.2.3	Les mémoires.....	7
1.4.2.4	Les interfaces d'entrées/sorties.....	7
1.5	Fonctionnement de l'A.P.I.....	8
1.6	Place de l'API dans le système automatisé de production (S.A.P).....	10
1.6.1	Les systèmes automatisés de production.....	10
1.6.2	Structure des systèmes automatisés.....	10
1.6.3	Nature des informations traitées par l'automate.....	11
1.7	Le microcontrôleur PICAXE.....	11

1.7.1	Définition du PICAXE.....	12
1.7.2	Le microcontrôleur PICAXE 40X2	12
1.7.2.1	Les entrées/sorties.....	13
1.7.2.2	Entrées analogiques.....	13
1.7.2.3	Fréquence d'horloge	14
1.7.2.4	La RAM.....	14
1.7.2.4.1	General RAM	14
1.7.2.4.2	Scratchpad RAM	14
1.7.2.5	Programme	15
1.7.2.6	Broches d'interruption matérielle	15
1.7.2.7	Boot I2C	15
1.7.3	Quelques commandes disponibles sur le PICAXE :.....	15
1.7.4	Le microcontrôleur 20X2 :.....	16
1.7.5	Programmation du PICAXE	16
1.8	Conclusion :	17

Chapitre 2 : Etude Théorique

2.1	Introduction.....	18
2.2	Caractéristiques de l'automate programmable.....	18
2.3	Présentation des différents blocs.....	19
2.3.1	Bloc d'alimentation.....	19
2.3.2	Les circuits de servitude.....	20
2.3.3	L'interface de programmation	21
2.3.4	Les sept entrées du convertisseur analogique/numérique.....	21
2.3.5	Les huit entrées numériques.....	22
2.3.6	Les quatorze sorties numériques.....	23
2.3.7	L'afficheur LCD et le buzzer	24
2.4	Les périphériques	25
2.4.1	La commande de deux moteurs à courant continu (DC)	25
2.4.1.1	Présentation du moteur à courant continue	25
2.4.1.2	Constitution d'un moteur à courant continue	26

2.4.2	La commande d'un moteur pas à pas unipolaire	28
2.4.2.1	Présentation du moteur pas à pas	28
2.4.2.2	constitutions d'un moteur pas à pas.....	28
2.4.3	La commande d'un servomoteur	30
2.4.4	La commande d'un relais	31
2.4.5	La lecture de la valeur d'une résistance ajustable.....	32
2.4.6	La lecture de la valeur d'une photorésistance.....	32
2.4.7	La lecture de l'état logique d'une touche.....	33
2.5	Conclusion	34

Chapitre 3 : Réalisation et tests

3.1	Introduction.....	35
3.2	Simulation sur PROTEUS et réalisation.....	35
3.2.1	Simulation du bloc d'alimentation.....	35
3.2.2	La carte principale.....	36
3.2.3	Les périphériques	39
3.2.3.1	Les entrées	39
3.2.3.2	Les sorties.....	41
3.2.4	Connexion entre l'automate et les périphériques.....	45
3.3	Programmation.....	47
3.4	Les tests.....	49
3.4.1	En mode Flowchart	49
3.4.1.1	Test d'une touche	50
3.4.1.2	Lecture de la valeur d'une résistance ajustable	51
3.4.1.3	Evaluation d'un niveau d'éclairage.....	52
3.4.1.4	Rotation d'un servomoteur	54
3.4.1.5	Commande de deux moteurs à courant continu	55
3.4.2	En mode Basic	58
3.4.2.1	Commande d'un moteur pas à pas unipolaire	58
3.4.2.2	Commande de deux moteurs à courant continu	60

3.5 Conclusion	60
Conclusion Générale	61
Bibliographie.....	62

Annexes

INTRODUCTION GENERALE

Le but d'un système automatique est de réaliser des tâches trop complexes ou dangereuses pour l'homme, faire des tâches répétitives et pénibles, et d'accroître la précision.

L'automatique est donc la discipline scientifique traitant, d'une part, de la caractérisation des systèmes automatisés et d'autre part, du choix, de la conception, et de la réalisation du système de commande.

Un automate programmable industriel, est l'ensemble électronique qui gère et assure la commande d'un système automatisé. Il se compose de plusieurs parties et notamment d'une mémoire programmable dans laquelle l'opérateur écrit, dans un langage propre à l'automate, des directives concernant le déroulement du processus à automatiser.

Au début des années 50, les ingénieurs étaient déjà confrontés à des problèmes d'automatismes. Les composants de base de l'époque étaient les relais électromagnétiques à un ou plusieurs contacts. Les circuits conçus comportaient des centaines voir des milliers de relais. Le transistor n'était connu que comme un composant d'avenir et les circuits intégrés étaient inconnus.

Vers 1960, les semi-conducteurs (diodes, transistors) sont apparus dans les automatismes sous forme de circuit digitaux. Ce n'est que quelques années plus tard, que l'apparition des circuits intégrés a amorcé une révolution dans la façon de concevoir les automatismes

Ceux-ci étaient très peu encombrants et leur consommation était des plus réduite, on pouvait alors concevoir des fonctions de plus en plus complexes à des coûts toujours décroissants.

C'est en 1969, que les constructeurs américains d'automobiles (General Motors en particulier) ont demandé aux firmes fournissant les matériels d'automatismes, des systèmes plus évolués et plus souples pouvant être modifiés simplement sans coût exorbitant.

Les ingénieurs américains ont résolu le problème en créant un nouveau type de produit nommé « automates programmables ». Les pionniers, à l'époque, étaient ALLEN-BRADLEY, MODICON ET DIGITAL EQUIPEMENTS qui construisirent les 3

premiers prototypes en 1969. TEXAS INSTRUMENTS regagne le groupe en 1970. En France, il fallait attendre 1973 avec le PB6 de MERLIN-GERIN qui instaure la dénomination sans doute un peu malheureuse d'automate programmable. Les automates programmables de cette époque n'étaient rentables que pour des installations d'une certaine complexité, mais la situation a très vite changée, ce qui a rendu les systèmes câblés obsolètes.

De nombreux modèles d'automates sont aujourd'hui disponibles ; depuis les nano automates bien adaptés aux machines et aux installations simples avec un petit nombre d'entrées/sorties, jusqu'aux automates multifonctions capables de gérer plusieurs entrées/sorties et destinés aux pilotages de processus complexes.

Dans ce même contexte, notre projet de fin d'études consiste à la réalisation d'un automate programmable, qui renferme deux microcontrôleurs PICAXE : un 40X2, le plus puissant de la gamme, ayant en charge toutes les opérations, même les plus complexes et un 20X2 servant de coprocesseur d'affichage « sériel » pour un écran LCD.

Notre automate comporte sept entrées analogiques, huit entrées numériques, quatorze sorties numériques, un port I2C, un afficheur LCD de quatre lignes de vingt caractères et un buzzer.

Nous avons ensuite développé autour, des périphériques qui servent à envoyer ou recevoir des informations de l'automate afin de piloter un processus automatisé. A noter, que ces deniers sont conçus de manière modulaire afin d'en faciliter l'utilisation.

Les périphériques développés sont :

- Commande de deux moteurs à courant continu.
- Commande d'un moteur pas à pas unipolaire.
- Commande d'un servomoteur.
- Commande d'un relais.
- La lecture de la valeur d'une résistance ajustable.
- La lecture de la valeur d'une photorésistance.
- La lecture de l'état logique d'une touche.

Après une introduction générale, notre mémoire est organisé en trois chapitres :

- Le premier chapitre traite des généralités sur les automates programmables et les microcontrôleurs.
- le deuxième chapitre est consacré à l'étude théorique des différentes parties constituant notre automate.
- Le troisième chapitre sera réservé à la réalisation des différents modules et aux tests.

Et à la fin, une conclusion générale vient conclure ce modeste travail.

Liste des abréviations

ADC	Analog to Digital Converter
AJ	Résistance Ajustable
API	Automate Programmable Industriel
BP	Bouton Poussoir
CI	Circuit Intégré
CMOS	Complementary Metal Oxide Semiconductor
EANA	Entré Analogique
EEPROM	Electrically Erasable Programmable Read Only Memory
EN	Entré Numérique
E/S	Entré/Sortie
GND	Ground
LCD	Liquid Crystal Display
LDR	Light Dependent Resistor
MJ / 4P4C	Modular Jack 4 Port 4 Connector
MLI	Modulation de Largeur d'Impulsion (En anglais: PWM)
NOB	Number Of Bits
PC	Personnel Computer
PHR	Photorésistance
PIC	Peripheral Interface Controller
PLC	Programmable Logic Controller
PWM	Pulse Width Modulation
RAM	Random-Access-Memory
ROM	Read-Only-Memory

RST	ReSeT (réinitialiser)
SAP	Système A utomatisé de P roduction
SCL	Signal C lock
SDA	Signal D ata
SN	Sortie N umérique
TR	T ransformateur
USB	U niversal S erial B us

1.1 Introduction

L'Automate Programmable Industriel API (Programmable Logic Controller, PLC) est aujourd'hui le constituant le plus répandu dans les automatismes. On le trouve pratiquement dans tous les domaines industriels vue sa grande flexibilité, son aptitude à s'adapter, sa robustesse et sa réactivité, ayant des entrées et des sorties physiques, utilisé pour automatiser des processus comme la commande des machines sur une ligne de montage dans une usine. Là où les systèmes automatisés plus anciens employaient des centaines ou des milliers de relais, un simple automate suffit.

1.2 Définition

Un automate programmable industriel est une machine électronique spécialisée dans la conduite et la surveillance en temps réel de processus industriels.

Il suit des instructions introduites dans ses mémoires sous formes de programme, et s'apparente par conséquent aux machines de traitement d'information.

Trois caractéristiques fondamentales le distinguent des outils informatiques tels que les ordinateurs utilisés dans les entreprises:

- 1) Il peut être directement connecté aux capteurs et préactionneurs grâce à ses entrées/sorties industrielles
- 2) Il est conçu pour fonctionner dans des ambiances industrielles sévères (température, vibrations, microcoupures de la tension d'alimentation, parasites, etc..).
- 3) En fin, sa programmation à partir des langages spécialement développés pour le traitement des fonctions d'automatismes des fonctions d'automatismes facilitent son exploitation et sa mise en œuvre.

Selon la norme française EN 61131-1, un automate programmable est un:

Système électronique fonctionnant de manière numérique, destiné à être utilisé dans un environnement industriel, qui utilise une mémoire programmable pour le stockage interne des instructions orientées aux fins de mise en œuvre des fonctions spécifiques, telles que des fonctions de logique, de mise en séquence, de temporisation, de comptage et de calcul arithmétique, pour commander au moyen d'entrées et de sorties Tout ou Rien ou analogiques divers types de machines ou de

processus. L'automate programmable et ses périphériques associés sont conçus pour pouvoir facilement s'intégrer à un système d'automatisme industriel et être facilement utilisé dans leurs fonctions prévues [1].

Si on veut résumer, on peut dire qu'un Automate Programmable Industriel (API) est une machine électronique programmable par un personnel non informaticien et destiné à piloter en ambiance industrielle et en temps réel des procédés industriels [2].

1.3 Application des automates programmables

Les automates programmables sont présents dans beaucoup de domaines soit par exemple les domaines suivants [5] :

- 1) Commande de machines (convoyage, stockage, emballage).
- 2) Automatisme du bâtiment (chauffage, climatisation, alarme, éclairage).
- 3) Régulation du processus (chimie, pétrochimie, traitement des eaux).
- 4) Contrôle de systèmes (production et distribution d'énergie).
- 5) Transport.

1.4 Architecture des automates programmables

1.4.1 Aspect extérieur

Les automates peuvent être de type **compact** ou **modulaire** :

- Le type *compact*

On distinguera les *modules de programmation* (LOGO de Siemens, ZELIO de Schneider, MILLENIUM de Crouzet ...) des *microautomates*. Il intègre le processeur, l'alimentation, les entrées et les sorties. Selon les modèles et les fabricants, il pourra réaliser certaines fonctions supplémentaires (comptage rapide, E/S analogiques ...) et recevoir des extensions en nombre limité [6].

Ces automates, de fonctionnement simple, sont généralement destinés à la commande de petits automatismes (voir figure 1.1) [6].

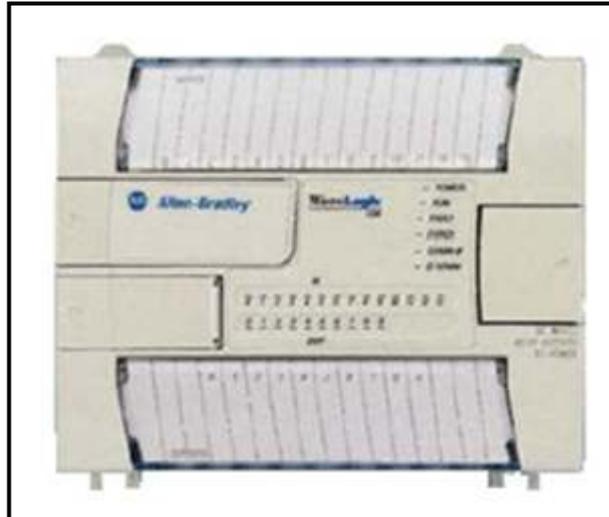


Figure 1.1 : Automate Compact [6]

- Le type *modulaire* :

Le processeur, l'alimentation et les interfaces d'entrées / sorties résident dans des unités séparées (**modules**) et sont fixées sur un ou plusieurs **racks** (tiroirs) contenant des bus et des connecteurs [6].

Ces automates sont intégrés dans les automatismes complexes où puissance, capacité de traitement et flexibilité sont nécessaires (voir figure 1.2) [6].



Figure 1.2 : Automate Modulaire [6]

1.4.2 structure interne

1.4.2.1 Module d'alimentation

Il assure la distribution d'énergie aux différents modules [6].

1.4.2.2 Le processeur

Le processeur a pour rôle principal le traitement des instructions qui constituent le programme de fonctionnement de l'application. Mais en dehors de cette tâche de base, il réalise également d'autres fonctions :

- 1- Gestion des entrées/sorties.
- 2- Surveillance et diagnostic de l'automate par une série de tests lancés à la mise sous tension ou cycliquement en cours de fonctionnement.
- 3- Dialogue avec le terminal de programmation, aussi bien pour l'écriture et la mise au point du programme en cours d'exploitation, pour des réglages ou des vérifications de données [1].

1.4.2.3 Les mémoires

Un système à processeur est toujours accompagné d'un ou plusieurs types de mémoires. Les automates programmables industriels possèdent pour la plupart les mémoires suivantes :

- Mémoire de travail.
- Mémoire système.
- Mémoire de chargement.
- Mémoire RAM non volatile.
- Mémoire ROM.

1.4.2.4 Les interfaces d'entrées/sorties

- **Interface d'entrée** : elle permet de recevoir les informations du Système Automatisé de Production (S.A.P) ou du pupitre et de mettre en forme (filtrage, ...) ce signal tout en l'isolant électriquement (optocouplage) [6].
- **Interface de sortie** : elle permet de commander les divers préactionneurs et éléments de signalisation du S.A.P. tout en assurant l'isolement électrique [6].

La figure 1.3 montre la structure interne de l'automate programmable.

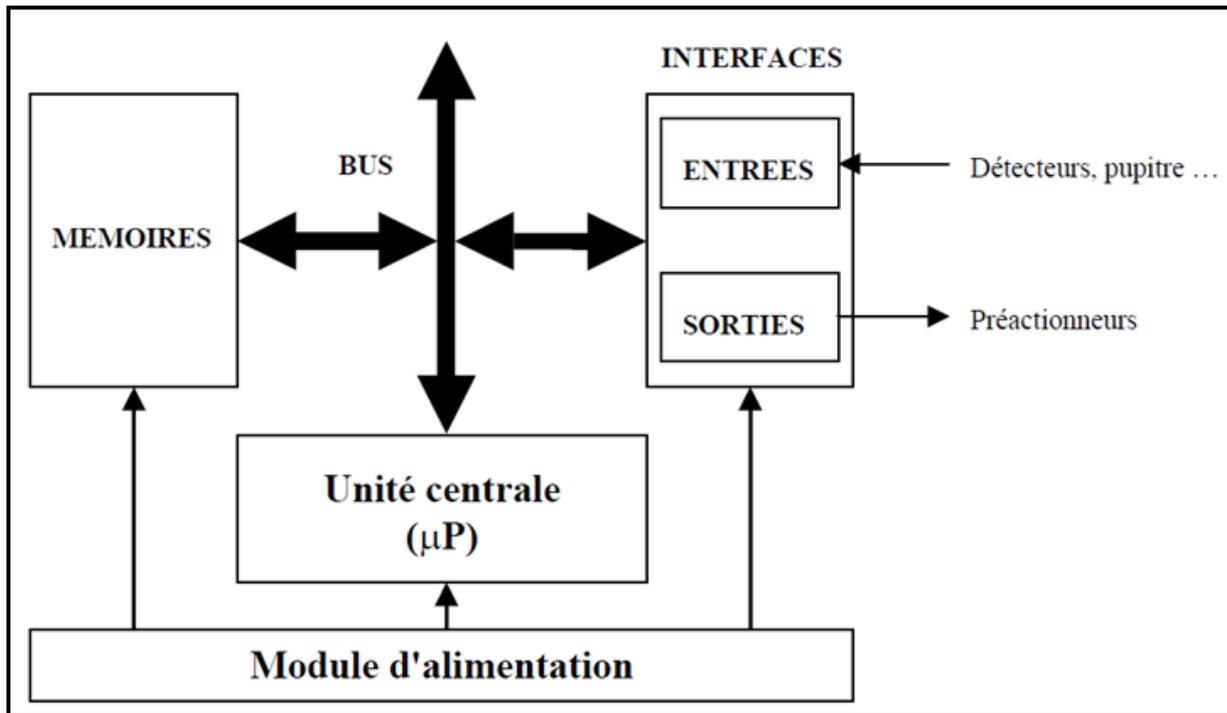


Figure 1. 3 : structure interne de l'automate programmable [6]

1.5 Fonctionnement de l'A.P.I

Lorsque l'API est en fonctionnement, trois phases se succèdent [7] :

- **PHASE 1 : PHOTOGRAPHIE DES ENTREES**

Durant cette phase qui dure quelques microsecondes :

- les entrées sont « photographiées » et leurs états logiques sont stockés dans une zone spécifique de la mémoire de donnée.
- Le programme n'est pas scruté.
- Les sorties ne sont pas mises à jour.

- **PHASE 2 : EXECUTION DU PROGRAMME**

Durant cette phase qui dure quelques millisecondes :

- Les instructions de programme sont exécutées une à une. Si l'état d'une entrée doit être lu par le programme, c'est la valeur stockée dans la mémoire de données qui est utilisée.
- Le programme détermine l'état des sorties et stocke ces valeurs dans une zone de la mémoire de données réservée aux sorties.
- Les entrées ne sont pas scrutées.
- Les sorties ne sont pas mises à jour.

Notez que pendant cette phase, seules la mémoire de données et la mémoire programme sont mises à contribution. Si une entrée change d'état sur le module d'entrées, l'API ne « voit » pas ce changement.

- **PHASE 3 : MISE À JOUR DES SORTIES**

Durant cette phase qui dure quelques microsecondes :

- Les états des sorties mémorisés précédemment dans la mémoire de données sont reportés sur le module de sorties.
- Les entrées ne sont pas scrutées.
- Le programme n'est pas exécuté.

Les trois phases exécutées par l'automate programmable sont montrés dans la figure 1.4.

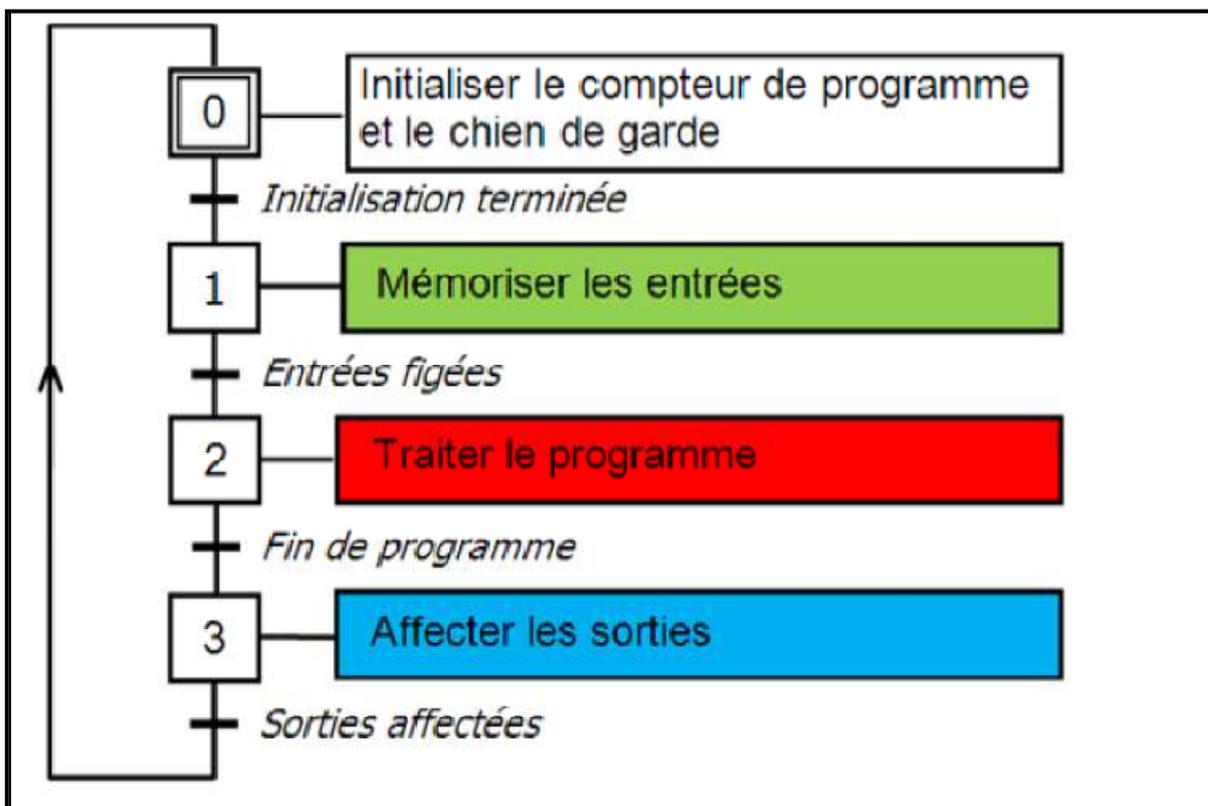


Figure 1. 4 : fonctionnement de l'A.P.I [2]

1.6 Place de l'API dans le système automatisé de production (S.A.P)

1.6.1 Les systèmes automatisés de production

L'objectif de l'automatisation des systèmes est de **produire**, en ayant recours le moins possible à l'homme, des produits de **qualité** et ce pour un **coût** le plus faible possible.

Un système automatisé est un ensemble d'éléments en interaction, et organisés dans un but précis : agir sur une **matière d'œuvre** afin de lui donner une **valeur ajoutée**.

Il est soumis à des **contraintes** : énergétiques, de configuration, de réglage et d'exploitation qui interviennent dans tous les modes de marche et d'arrêt du système [6].

1.6.2 Structure des systèmes automatisés

Tout système automatisé peut se décomposer selon le schéma ci-dessous (figure 1.5).

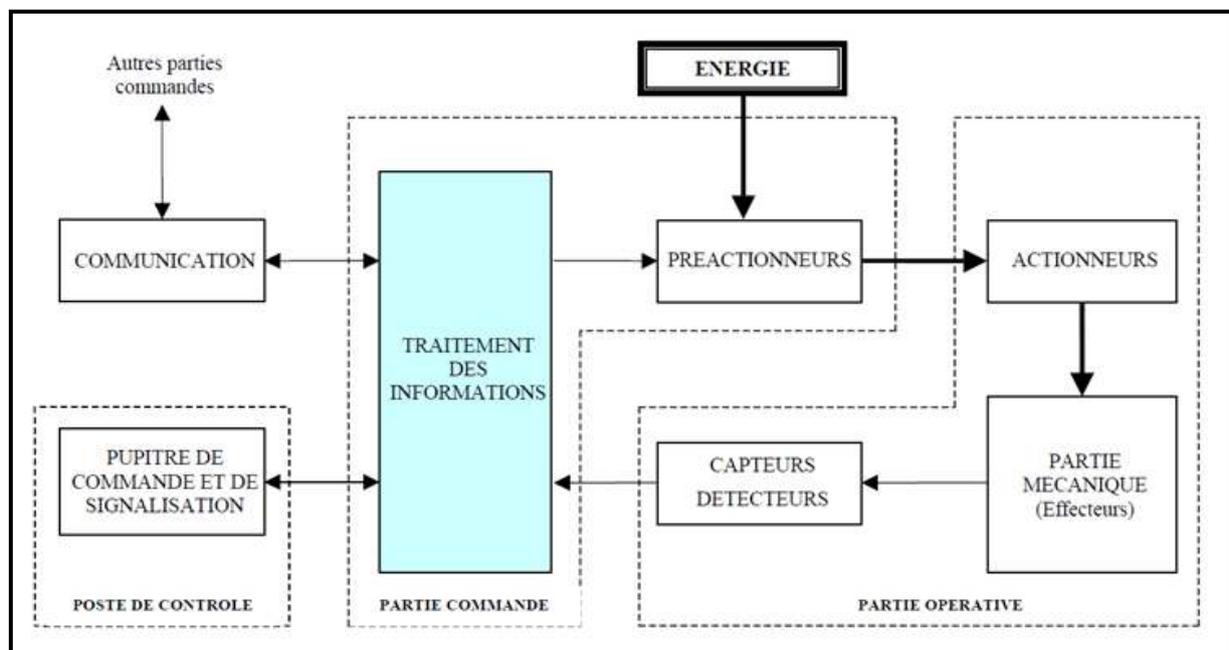


Figure 1. 5 : structure des systèmes automatisés [6]

➤ Partie opérative

Elle agit sur la matière d'œuvre afin de lui donner sa valeur ajoutée.

Les **actionneurs** (moteurs, vérins) agissent sur la partie mécanique du système qui agit à son tour sur la matière d'œuvre [6].

Les **capteurs / détecteurs** permettent d'acquérir les divers états du système. [6]

➤ Partie commande

Elle donne les ordres de fonctionnement à la partie opérative.

Les **préactionneurs** permettent de commander les actionneurs ; ils assurent le **transfert d'énergie** entre la source de puissance (réseau électrique, pneumatique ...) et les actionneurs.

Exemple : contacteur, distributeur ...

Ces préactionneurs sont commandés à leur tour par le bloc **traitement des informations**.

Celui-ci reçoit les consignes du **pupitre de commande** (opérateur) et les informations de la partie opérative transmises par les capteurs / détecteurs.

En fonction de ces consignes et de son programme de gestion des tâches (implanté dans un automate programmable ou réalisé par des relais (on parle de logique câblée)), La partie commande va commander les préactionneurs et renvoyer des informations au **pupitre de signalisation** ou à d'autres systèmes de commande et/ou de supervision en utilisant un **réseau** et un **protocole de communication [6]**.

➤ **Poste de contrôle**

Composé des **pupitres de commande et de signalisation**, il permet à l'opérateur de commander le système (marche, arrêt, départ cycle ...).

Il permet également de visualiser les différents états du système à l'aide de voyants, de terminal de dialogue ou d'interface homme-machine [6].

1.6.3 Nature des informations traitées par l'automate

Les informations peuvent être de type [6] :

- **Tout Ou Rien (T.O.R.)** : l'information ne peut prendre que deux états (vrai/faux, 0 ou 1 ...). C'est le type d'information délivrée par un détecteur, un bouton poussoir ...
- **Analogique** : l'information est continue et peut prendre une valeur comprise dans une plage bien déterminée. C'est le type d'information délivrée par un capteur (pression, température ...)
- **Numérique** : l'information est contenue dans des mots codés sous forme binaire ou bien hexadécimale. C'est le type d'information délivrée par un ordinateur ou un module intelligent.

1.7 Le microcontrôleur PICAXE

Un microcontrôleur est souvent décrit comme un «ordinateur-sur-une-puce». Il s'agit d'un circuit intégré de faible coût qui contient la mémoire, des unités de traitement, et

l'entrée / sortie d'un circuit à une seule unité. Les microcontrôleurs sont délivrés «blancs» et ensuite programmés avec un programme spécifique de contrôle.

Une fois programmé, le microcontrôleur est introduit dans un système pour le rendre plus intelligent et facile à utiliser [3].

Les puces PICAXE sont populaires parce qu'ils sont très peu coûteux et simples à programmer à l'aide d'un logiciel facile à manipuler [4].

Les différentes tailles des puces PICAXE (8, 14, 18, 20, 28 et 40 broches) donnent une grande flexibilité sur la façon dont le système peut être utilisé, il suffit de sélectionner la taille de la puce comme requis pour un projet. Sur les pièces PICAXE presque toutes les broches peuvent être configurées pour être une sortie, une entrée numérique ou analogique. Les fonctionnalités avancées telles que PWM, I2C... peuvent également être facilement atteintes si nécessaire [4].

1.7.1 Définition du PICAXE

Un Picaxe est un microcontrôleur **Pic** auquel est ajouté un logiciel d'amorçage développé par la société **Revolution Education**. Le logiciel d'amorçage (ou bootstrap) est un programme installé dans une zone spécifique du microcontrôleur ce qui permet de ne pas utiliser de programmeur spécifique. Ce mode particulier utilise deux broches du Pic (Serial Out et Serial In) et le dialogue s'effectue via une connexion **série RS232** ou **USB** [8].

Le code d'amorçage pré-programmé contient également des routines communes (comme la façon de générer un retard, de pause ou une sortie son), de sorte que chaque téléchargement n'a pas à perdre du temps en téléchargeant ces données communément requis. Cela rend le téléchargement beaucoup plus rapide [4].

1.7.2 Le microcontrôleur PICAXE 40X2

La **puce du microcontrôleur PICAXE-40X2** est une puce de deuxième génération dotée de davantage de fonctionnalités et d'une flexibilité de broches optimales, ce qui rend le contrôleur idéal pour la formation. Microcontrôleur à 40 broches amélioré avec contrôle individuel des broches, 4 slots de mémoire et d'autres fonctionnalités, comme la prise en charge du bus I2C/spi, les caractéristiques de ce microcontrôleur sont citées en annexe A.

La version PIC18F45K22 du microcontrôleur 40X2 peut fonctionner de 2.1 à 5V par 2 broches de Vcc et 2 broches de GND.

1.7.2.1 Les entrées/sorties

Une des nouvelles fonctionnalités clés de la série X2, c'est que presque toutes les broches sont configurables comme entrées ou sorties. Cela crée beaucoup plus de flexibilité. Naturellement, les broches peuvent être configurées pour la mise en PICAXE par défaut si désiré. La gamme X2 peut avoir jusqu'à 32 broches d'entrées / sorties configurables, qui sont disposées dans 4 ports, étiquetés de A à D. Chaque port a jusqu'à 8 broches (0-7) [7].

Voir le schéma de brochage du microcontrôleur 40X2 (figure 1.6)

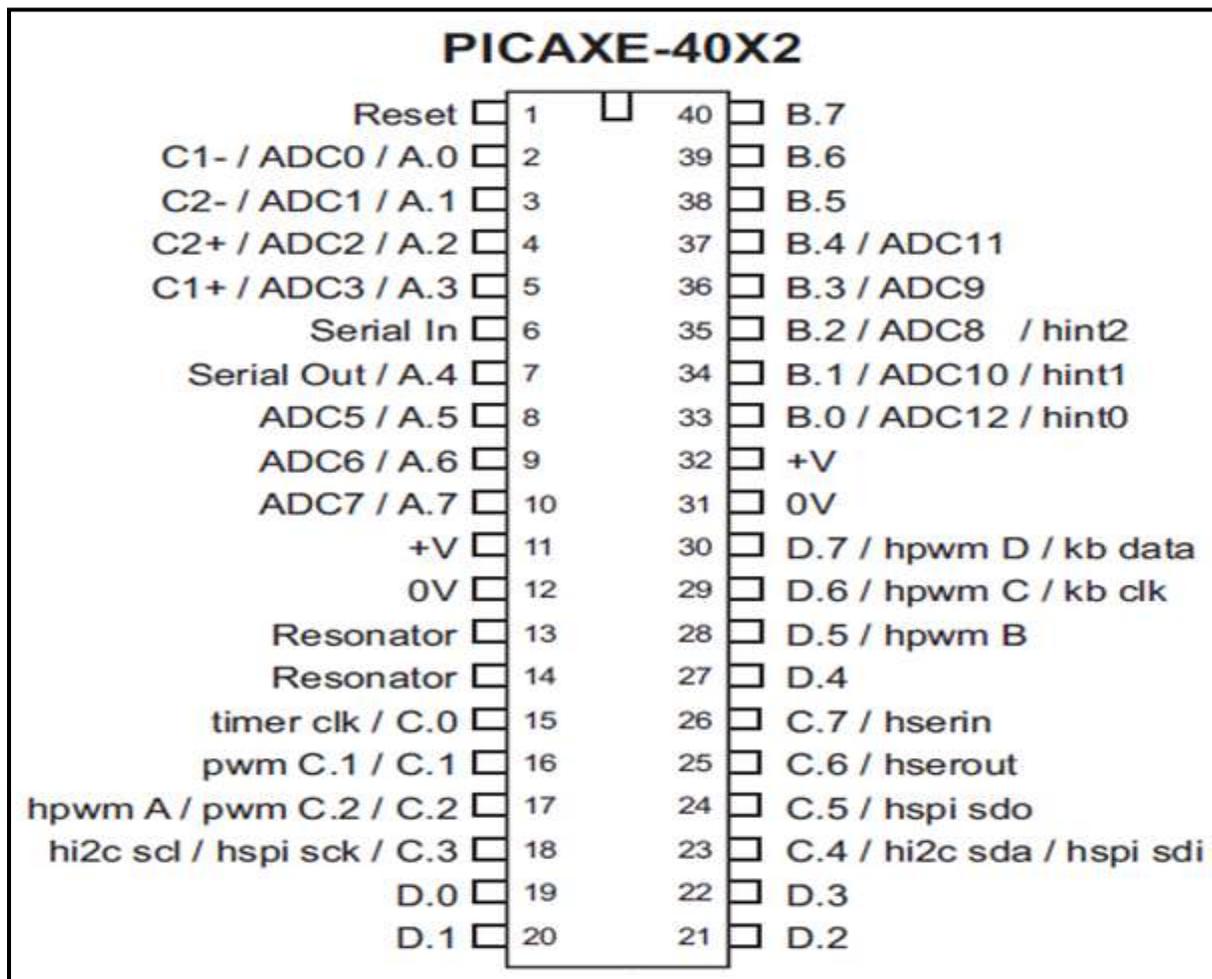


Figure 1.6 : schéma de brochage du microcontrôleur PICAXE 40X2 [7]

1.7.2.2 Entrées analogiques

Dans la série X2 on peut trouver jusqu'à 12 broches pour l'ADC (entrées analogique sur 8 ou 10 bits) qui sont configurables.

En raison du grand nombre de broches ADC disponibles, chaque ADC a un numéro de

canal ADC unique. Pour cela on utilise la commande "readadc" pour sélectionner le canal au quel on va faire la conversion analogique / numérique.

Une nouvelle fonctionnalité de la gamme X2 est l'intégration de 2 comparateurs internes (C1 et C2) qui comparent constamment deux valeurs analogiques. Les deux valeurs peuvent être de deux broches ADC externes, ou d'une broche ADC externe et une valeur interne [7].

1.7.2.3 Fréquence d'horloge

Des fréquences d'horloge très élevées sont disponibles sur le PICAXE. Cela améliore beaucoup la vitesse de traitement.

Le 40X2 a une fréquence interne de 16MHz, en plus il a des options de fréquence d'horloge qui lui permettent d'avoir une fréquence externe de 64MHz. [7]

1.7.2.4 La RAM

Le X2 possède une grande zone de RAM jusqu'à 1280 octets pour l'utilisation générale (Tableau 1.1).

Tableau 1. 1 : La RAM du microcontrôleur 40X2 [7]

Microcontrôleur	General RAM	Scratchpad
40X2	256	1024

1.7.2.4.1 General RAM

Sur les parties X2, il ya jusqu'à 256 variables de caractère général. 56 d'entre eux, connus sous le nom b0 à b55, peuvent être utilisés directement dans n'importe quelle commande.

Tous les octets (0-127 / 0-255) peuvent également être adressées à la fois directement et indirectement [7].

1.7.2.4.2 Scratchpad RAM

C'est une mémoire interne à grande vitesse utilisée pour le stockage temporaire des calculs, des données, et d'autres travaux en cours. Sur le X2, il ya jusqu'à 1024 octets de RAM de bloc-notes [7].

1.7.2.5 Programme

Chaque emplacement de programme peut contenir environ 1000 lignes de code BASIC.

La gamme X2 peut avoir jusqu'à quatre emplacements de programmes internes, numérotés de 0 à 3. Chaque emplacement est complètement indépendant des autres emplacements. Les programmes peuvent être démarrés à l'aide d'une commande 'run' [7].

Le PICAXE 40X2 contient quatre emplacements de programme, la capacité mémoire de chaque emplacement est de 4096 octets [4].

1.7.2.6 Broches d'interruption matérielle

Le X2 a 3 broches qui peuvent être configurés comme des broches d'interruption matérielle. Lorsqu'un déclenchement d'une interruption se produit, un indicateur est défini qui peut être utilisé pour déclencher un événement [7].

1.7.2.7 Boot I2C

La commande boot I2C peut être utilisée pour copier un programme à partir d'un emplacement de mémoire externe dans un emplacement de la mémoire interne [7].

1.7.3 Quelques commandes disponibles sur le PICAXE :

Sur les puces de deuxième génération du PICAXE on peut trouver les commandes suivantes :

➤ **Support UNI/O**

Les commandes « *UNIin/UNIout* » permettent de communiquer avec *UNI/O* d'une mémoire EEPROM de Microchip [7].

➤ **Servo**

La commande « *servo* » qui sert à commander un servomoteur a été entièrement mise à jour, maintenant elle est plus précise et elle fonctionne de 8 à 32 MHz [7].

➤ **Opérateur NOB (Number Of Bits)**

C'est un nouvel opérateur qui compte le nombre de '1' dans une variable [7].

1.7.4 Le microcontrôleur 20X2 :

Le microcontrôleur PICAXE 20X2 est une puce de deuxième génération qui contient 20 broches qui sont disposées dans 3 ports, étiquetés de A à C. Chaque port a jusqu'à 8 broches (0-7), voir le schéma de brochage (figure 1.7). Pour le rendre plus flexible, presque toutes ses broches sont configurables. Il peut être alimenté sur une plage de 1.8 à 5V et sa fréquence de fonctionnement peut atteindre 64 MHz en utilisant des options de fréquence d'horloge interne.

La taille de RAM qu'il possède est montrée sur le tableau 1.2.

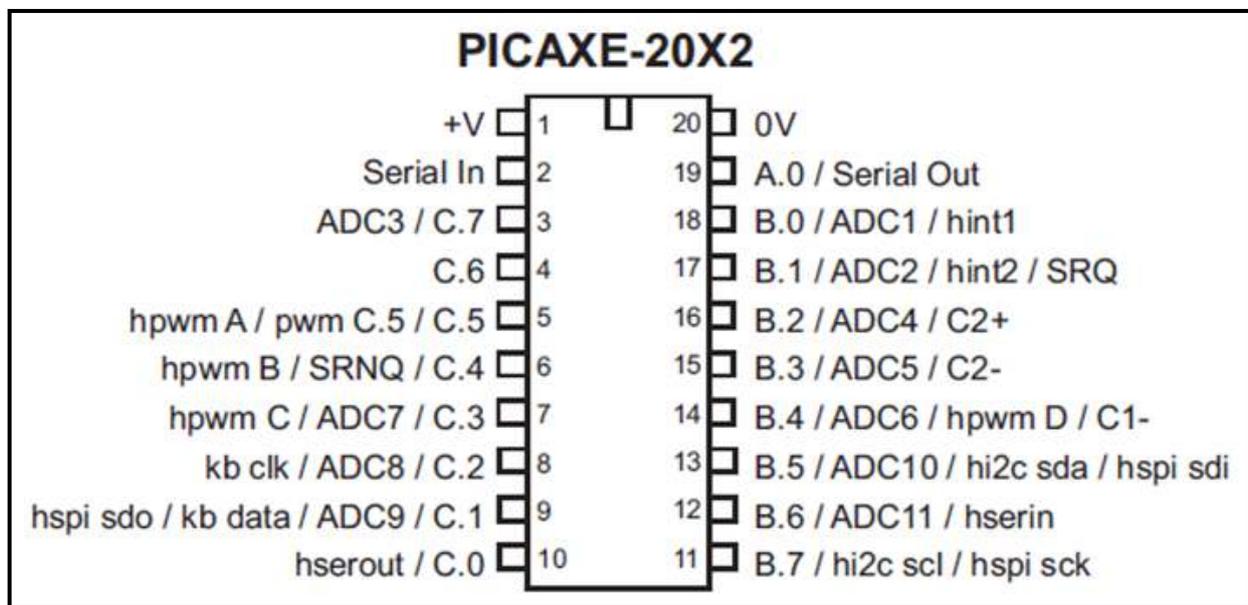


Figure 1. 7 : schéma de brochage du microcontrôleur PICAXE 20X2 [7]

Tableau 1. 2 : La RAM du microcontrôleur 20X2 [7]

Microcontrôleur	General RAM	Scratchpad
20X2	128	128

1.7.5 Programmation du PICAXE

Le PICAXE utilise le langage **BASIC** simple ou des **organigrammes graphiques (Flowchart)** tel que les utilisateurs peuvent générer des programmes facilement. Il est beaucoup plus facile à apprendre et à manipuler que les autres langages de programmation (C ou assembleur) [4].

La programmation du PICAXE est effectuée à l'aide d'un logiciel qui s'appelle « **PICAXE Programming Editor** ».

Le « Picaxe Programming Editor » est un logiciel permettant d'écrire des programmes en BASIC ou sous forme graphique (organigramme de programmation) puis de les transférer dans un microcontrôleur Picaxe [8].

Le circuit téléchargement USB est identique pour toutes les puces PICAXE. Il se compose de 3 fils de la puce PICAXE au câble USB AXE027. Un fil envoie des données à partir de l'ordinateur à l'entrée série du PICAXE, un fil transmet les données de PICAXE à l'ordinateur, et le troisième fil est la masse. Le même circuit peut également être utilisé pour le câble série AXE026 (voir figure 1.8) [3].

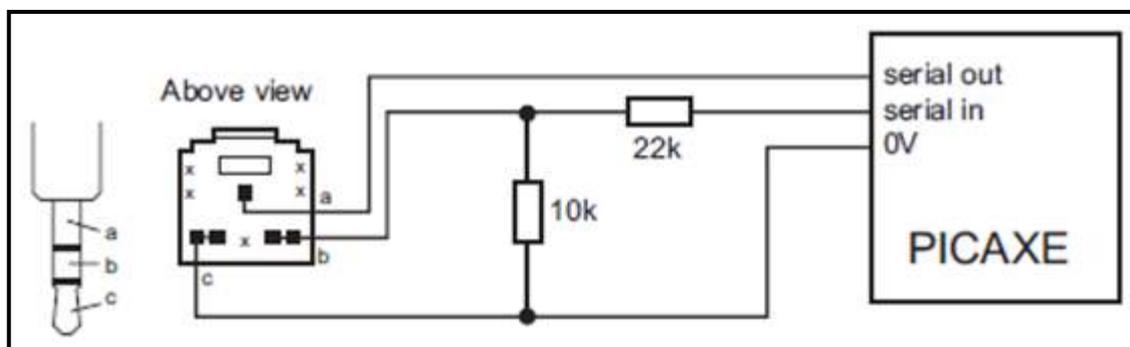


Figure 1. 8 : Circuit de téléchargement PICAXE [3]

1.8 Conclusion :

Pour avoir un produit complet et fini dans une chaîne de production, ça nécessite une grande précision et des fois des tâches dangereuses. Afin de réaliser un produit parfait convainquant et satisfaisant, il faut introduire un ou plusieurs automates programmables dans les systèmes de production.

Pour cela les automates programmables industriels ont un rôle indispensable dans l'industrie, dans ce chapitre on a vu l'architecture, le fonctionnement, le rôle et les applications des A.P.I.

D'autre part, on a parlé sur la série des microcontrôleurs PICAXE et les fonctionnalités présentées dans cette gamme, ainsi que leur flexibilité, rapidité et la capacité des différents types de mémoire. Ces atouts rendent les microcontrôleurs PICAXE plus puissants et performants.

2.1 Introduction

La popularité des microcontrôleurs PICAXE ne cesse de croître. On se propose d'entreprendre la réalisation d'un automate programmable, destiné à la domotique. Pour cela, on s'est inspiré d'un montage présenté dans la magazine d'électronique pratique [9].

Il renferme deux microcontrôleurs PICAXE : un 40X2, le plus puissant de la gamme, ayant en charge toutes les opérations, même les plus complexes et un 20X2 servant de coprocesseur d'affichage « sériel » pour un écran LCD de quatre lignes de vingt caractères.

Notre montage comporte sept entrées analogiques, huit entrées numériques, quatorze sorties numériques, un port I2C, un afficheur LCD de quatre lignes de vingt caractères et un buzzer.

Les connexions aux périphériques s'effectuent à l'aide de prises et embases de type MJ-4P4C, très fiables et pratiques. Elles véhiculent les alimentations, les signaux et sont munies de détrompeurs.

Notre étude porte également sur la réalisation des différents périphériques d'entrées et de sorties (capteurs, touches, moteurs DC, moteurs pas à pas, relais, servomoteur, etc..) offrant à notre automate toute sa puissance.

La programmation se fera via le logiciel « PICAXE Programming Editor », en basic, pour tirer parti de toute sa puissance, ou éventuellement en mode « Flowchart » (diagrammes).

2.2 Caractéristiques de l'automate programmable

En plus de la gestion des automatismes, notre montage offre d'énormes possibilités en vue d'apprendre et de se perfectionner dans le domaine de la programmation des microcontrôleurs PICAXE.

Les principales caractéristiques et équipements de notre automate sont résumées dans le tableau suivant (Tableau 2.1).

Tableau 2. 1 : caractéristique de l'automate programmable

Alimentation par un bloc secteur de 12 V ou à partir de six piles de 1.5V
Gestion « sérielle » de l'afficheur par un coprocesseur PICAXE-20X2
Buzzer piézo
Programmation du microcontrôleur par câble « sériel » ou USB (référence AXE027)
Possibilité de reprogrammer le coprocesseur d'affichage
Pas de programmeur externe
Sept entrées analogiques sur 8 bits ou 10 bits
Huit entrées numériques
Afficheur alphanumérique de quatre lignes de vingt caractères
Quatorze sorties numériques, dont deux destinées aux signaux PWM (MLI en français)
Les signaux PWM sont gérés en multitâches (indépendamment du déroulement du programme)
Port I2C
Toutes les lignes d'entrées sorties sont disponibles sur des connecteurs femelles de type MJ-4P4C
Cinq sorties auxiliaires sur le coprocesseur d'affichage (sur connecteur SIL)
Visualisation par leds des niveaux des entrées et des sorties numériques
Entrées numériques tamponnées
Sorties numériques protégées en courant par résistances à 10 mA
Réalisation de dix périphériques d'entrées sorties
Extension sans limites du type et du nombre de périphériques
Aucun câblage externe

2.3 Présentation des différents blocs

2.3.1 Bloc d'alimentation

Ce bloc d'alimentation consiste à assurer une tension continue de 5V à la sortie du bloc pour alimenter d'autres blocs (voir figure 2.1).

Il s'agit des signaux de programmation (signal input et signal output). Deux cavaliers de configuration permettent de sélectionner le PICAXE-40X2 ou le PICAXE-20X2. Pour d'évidentes raisons de simplification, grâce à ce procédé, une seule interface de programmation suffit.

La broche d'initialisation RST est portée, par défaut, au potentiel positif via la résistance R42. l'appui sur la touche RST force cette broche à la masse, à travers la résistance R43 de faible valeur et, de ce fait, initialise le microcontrôleur.

2.3.3 L'interface de programmation

Ce bloc joue le rôle d'interface entre le PC et les microcontrôleurs à l'aide d'un câble USB AXE027 (voir figure 2.3).

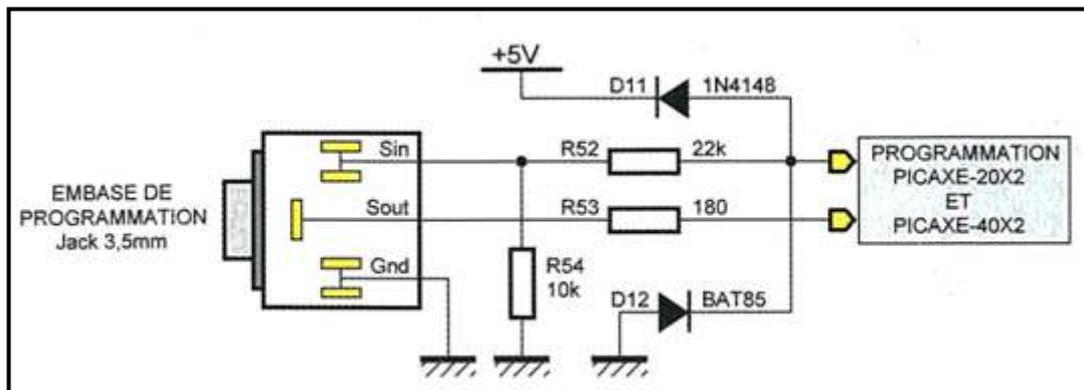


Figure 2. 3 : Module d'interface de programmation

Elle ne présente aucune particularité. Seule l'embase 'jack 3.5 mm' et les trois résistances R52 à R54 sont réellement indispensables. Les deux diodes optionnelles D11 et D12 améliorent la communication dans certains cas.

On connecte cette interface de programmation au PC à l'aide d'un câble AXE027, le choix de picaxe (40X2 ou 20X2) à programmer se fait à l'aide des cavaliers de configuration.

2.3.4 Les sept entrées du convertisseur analogique/numérique

Le Picaxe 40X2 dispose de 7 entrées analogiques. Ces entrées disposent d'un convertisseur permettant de fournir une valeur numérique représentative de la valeur analogique. La valeur analogique doit impérativement évoluer entre 0V et la tension d'alimentation +Vcc (5V) sinon le circuit Picaxe peut être détruit.

La valeur numérique obtenue est codée sur 8 bits (un octet) et peut donc évoluer de 00000000 à 11111111 en binaire donc de 0 à 255 en décimal (la précision obtenue est de $5/255 = 0,0196V$ par pas). Ainsi si la valeur numérique est de 00110111 (55 en décimal), la tension analogique est de 55×0.0196 soit 1,08V (voir figure 2.4).

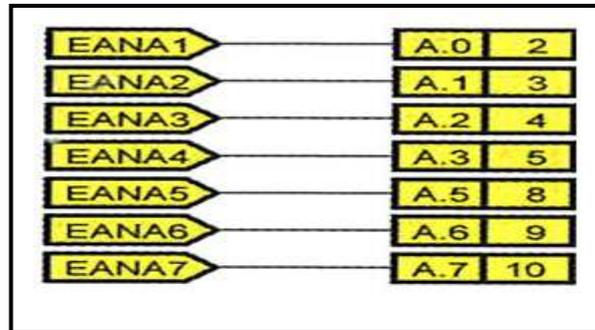


Figure 2. 4 : les sept entrées du convertisseur analogique/numérique

2.3.5 Les huit entrées numériques

Ce bloc consiste à visualiser les entrées numériques de notre automate (voir figure 2.5).

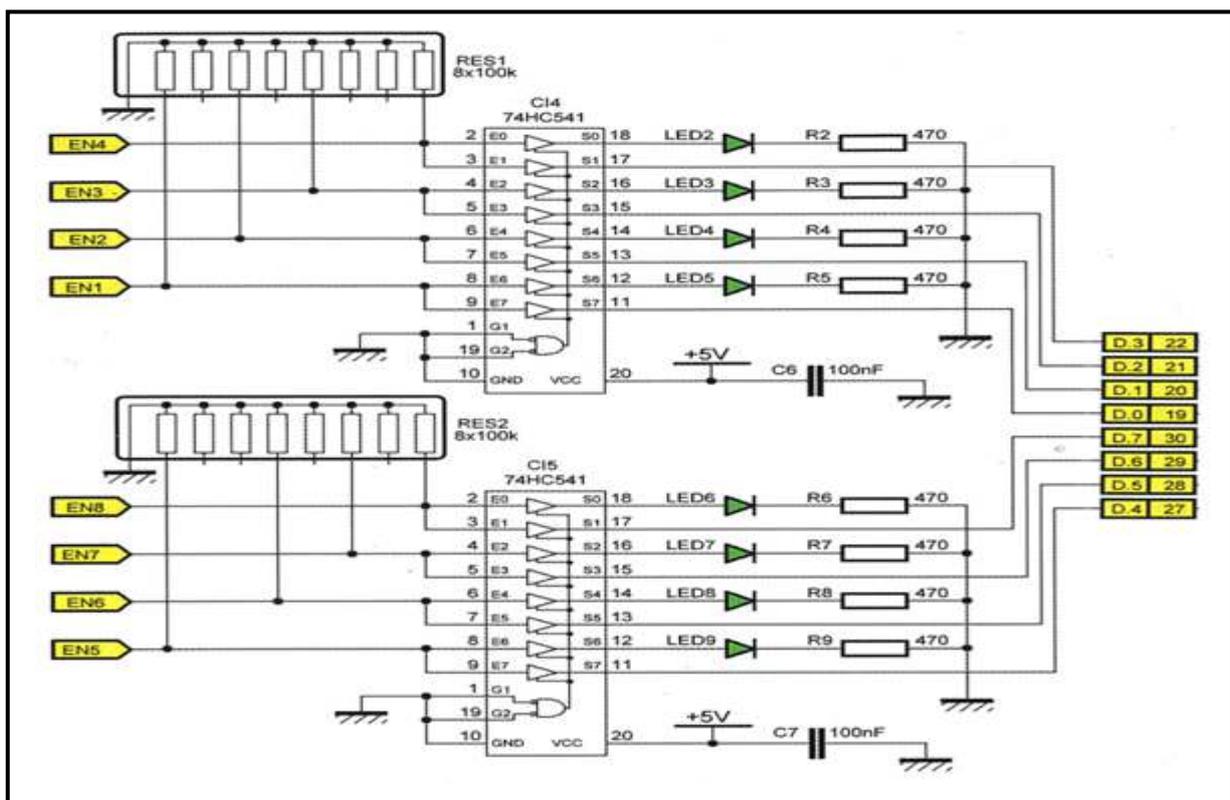


Figure 2. 5 : les huit entrées numériques

Chaque entrée (EN1 à EN8) attaque deux tampons non-inverseurs des circuits CI4 ou CI5. Au repos, ces entrées sont positionnées au niveau logique bas, par une résistance de forte valeur des blocs de résistances RES1 ou RES2. Les tampons des circuits CI4 et CI5 étant

commandés simultanément deux par deux, une des deux sorties aboutit à une des huit entrées numériques du port D, l'autre gère la visualisation de l'état logique au moyen d'une résistance R2 à R9. Les condensateurs C6 et C7 découplent la tension d'alimentation de CI4 et CI5.

Les circuits intégrés CI4 et CI5 de type 74HC541 sont des dispositifs CMOS à grille de Silicium, sont composés de huit portes logiques non-inverseur et un seul porte non-and, ce circuit permet d'amplifier le courant et isoler ou protéger le picaxe 40X2 [10].

2.3.6 Les quatorze sorties numériques

Ce bloc sert à visualiser les sorties numériques de notre automate et le port I2C (voir figure 2.6).

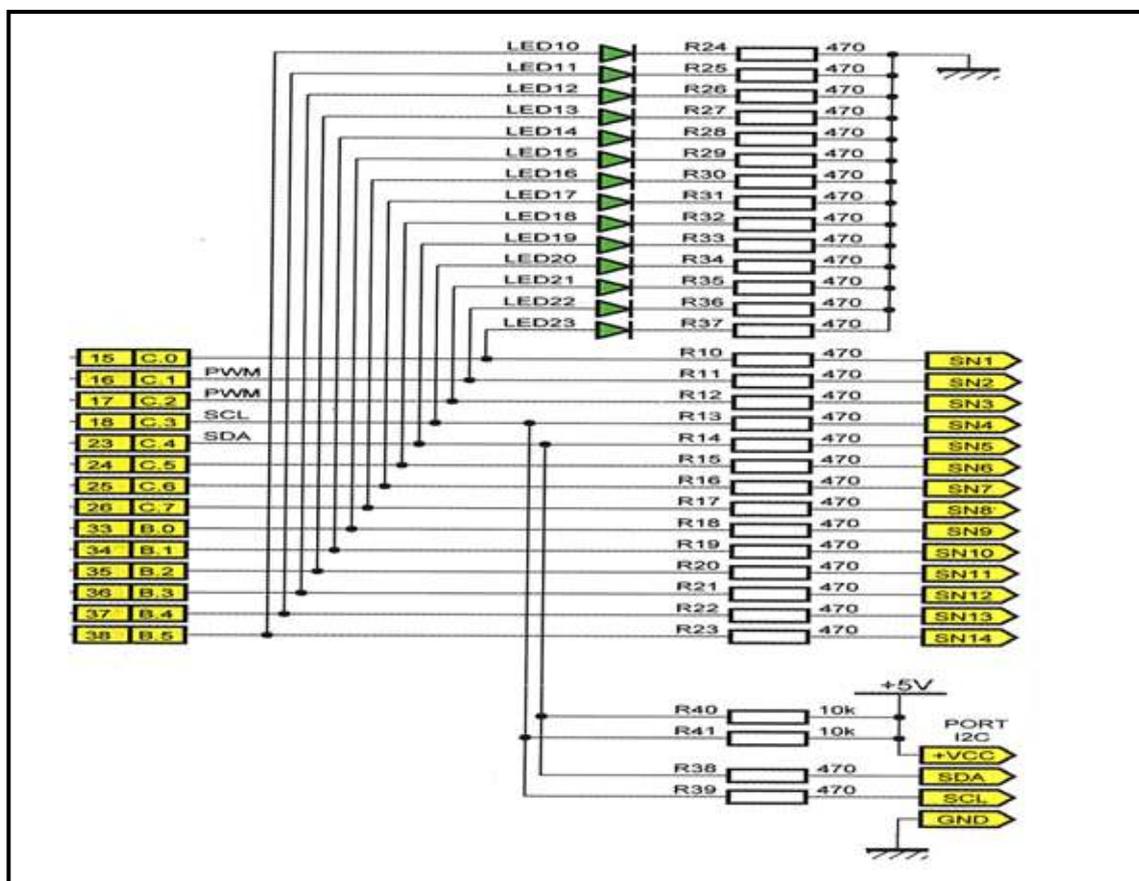


Figure 2. 6 : les quatorze sorties numériques

Les lignes des portes B0 à B5 et C0 à C7 du microcontrôleur arrivent sur les embases MJ-4P4C, via les résistances de protections R10 à R23. Il conviendra d'en tenir compte lors des développements des périphériques (notamment pour la commande d'un transistor ou d'une led par exemple).

Les niveaux logiques des sorties sont visualisés par les led 10 à 23, limitées en courant par les résistances R24 à R37. On Note que les sorties C1 et C2 (SN2 et SN3) peuvent gérer les signaux PWM (ou MLI) en mode multitâche.

Les sorties C3 et C4 se chargent de la communication, selon le protocole I2C. Les données sont véhiculées via la ligne C4 (SDA). Le signal d'horloge est fourni par la ligne C.3 (SCL). Les résistances R38 et R39 protègent ces ports. Les résistances R40 et R41 les positionnent au niveau logique « haut » (+5V), au repos.

2.3.7 L'afficheur LCD et le buzzer

Ce bloc sert à l'affichage à l'aide d'un microcontrôleur PICAXE 20X2 des informations venant du PICAXE 40X2 (voir figure 2.7).

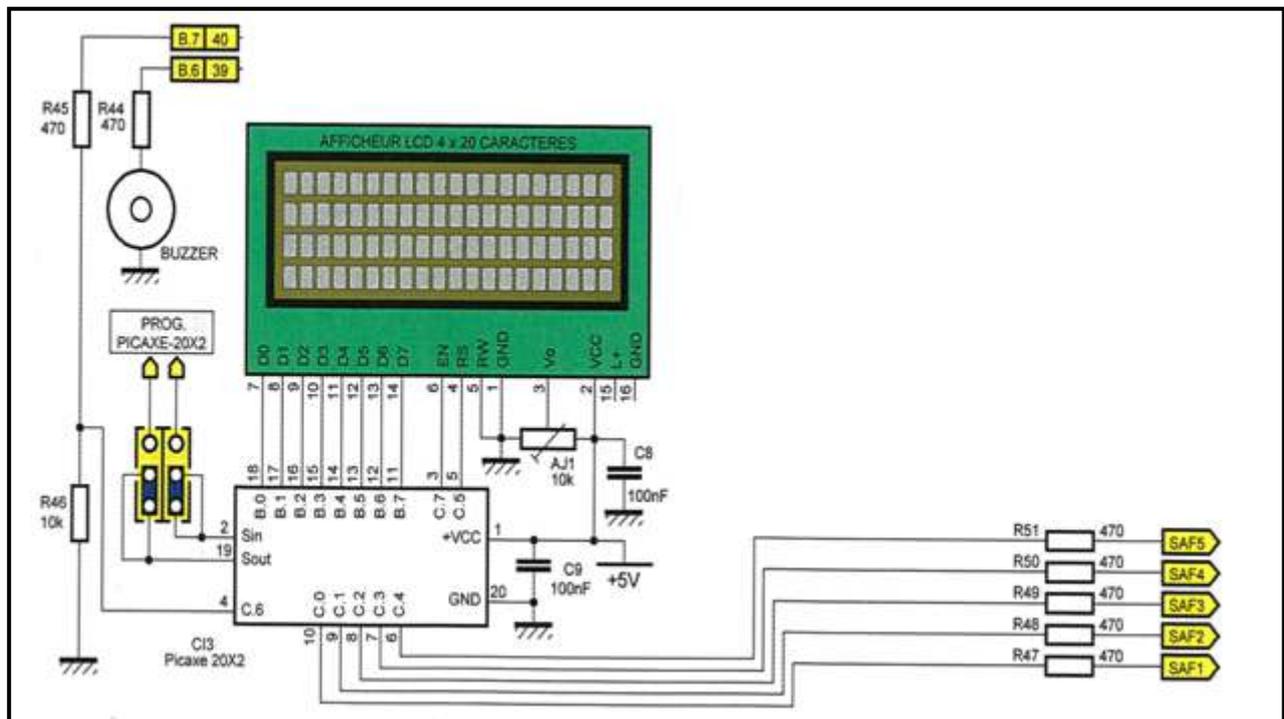


Figure 2. 7 : l'afficheur LCD et le buzzer

Il s'agit d'un modèle courant de quatre lignes de vingt caractères à commande « parallèle », sans rétro-éclairage.

La résistance ajustable AJ1 règle le contraste et le condensateur C8 découple sa tension d'alimentation.

La ligne RW est directement liée à la masse, car nous n'envisageons pas de lire les registres internes de l'afficheur dans cette application. Les entées de données (D0 à D7) et de

gestion (RS et EN) sont reportées sur les sorties de CI3, un microcontrôleur PICAXE-20X2 dédié à l’affichage. Ce coprocesseur nous permet de commander, à moindres frais, l’afficheur en mode « sériel », bien moins gourmand en lignes de communication.

Nous disposons ainsi de presque toutes les lignes du PICAXE-40X2. Le port C6 de CI3 se charge de la communication « sérielle », via la résistance de protection R45 et aboutit à la ligne B7 du PICAXE-40X2. Le niveau logique « bas », nécessaire repos, est assuré via la résistance R46. Le condensateur C9 découple la tension, au plus près de CI3.

Le fait d’employer un coprocesseur d’affichage nous permet de bénéficier de cinq sorties auxiliaires raccordées aux ports C0 à C4 de CI3 et commandées par la même communication « sériel ». Les résistances R47 à R51 assurent la protection de ces lignes. La ligne B6 de CI2 commande le buzzer piézo, via la résistance R44 de faible valeur.

Les schémas de brochage du 40X2 et 20X2 sont donnés en annexe B.

2.4 Les périphériques

Afin de pouvoir tester notre automate, nous avons procédé à la réalisation de quelques périphériques classiques. Pour cela, nous avons utilisés les plus courants. Il est évident que leur nombre n’est pas restrictif, on peut concevoir d’autres avec le même principe.

Chaque périphérique tient sur une petite platine, disposant de sa propre embase et d’un connecteur à trois broches proposant l’alimentation (0V et +5V) et le signal. Certains périphériques utilisent la même platine selon les composants câblés.

D’autres peuvent cohabiter, sans soucis, sur un seul circuit imprimé, comme le buzzer piézo, la led et le servomoteur.

2.4.1 La commande de deux moteurs à courant continu (DC)

Un moteur électrique sert à transformer l’énergie électrique en énergie mécanique. Ce circuit de commande de deux moteur DC consiste à commander la vitesse et le sens des moteurs (voir figure 2.8).

2.4.1.1 Présentation du moteur à courant continu

Le moteur à courant continue est fréquemment employé en automatisme (par exemple, en robotique). Il est alimenté par une tension continue.

2.4.1.2 Constitution d'un moteur à courant continu

Un inducteur, appelé aussi stator, est composé d'enroulements bobinés autour d'un élément immobile du stator. Il crée le champ magnétique dit statorique.

Un induit appelé aussi rotor cylindrique est composé de tôles isolées entre elles et munies d'encoches dans lesquelles sont réparties les conducteurs. Parcourus par un courant, ceux-ci créent le champ magnétique dit rotorique

Un collecteur fixé à l'induit, est en contact avec les charbons. Ces derniers appelés aussi balais, alimentent l'induit par le collecteur sur lequel ils frottent (voir figure 2.10).

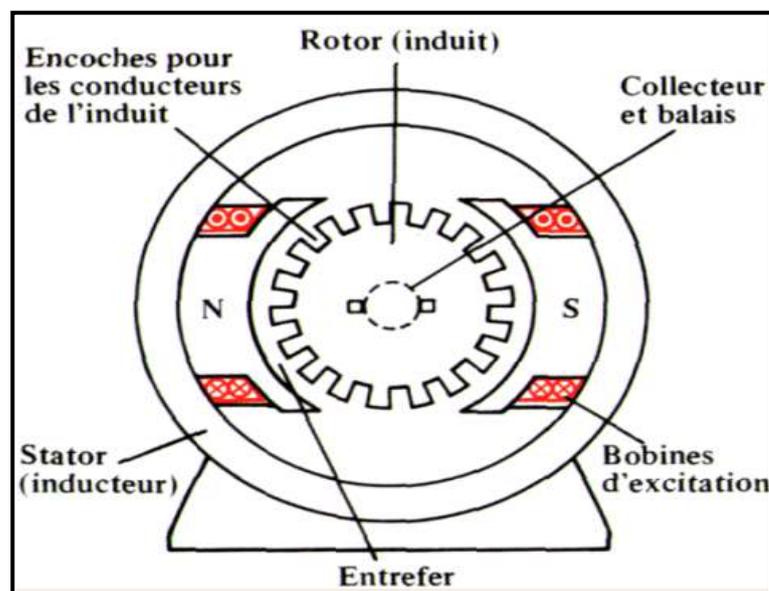


Figure 2.8 : schéma détaillé d'un moteur à courant continu

Dans notre cas le moteur est alimenté par une tension continue variable de 0 à 12 V. Il consomme en fonctionnement nominal un courant de 1A.

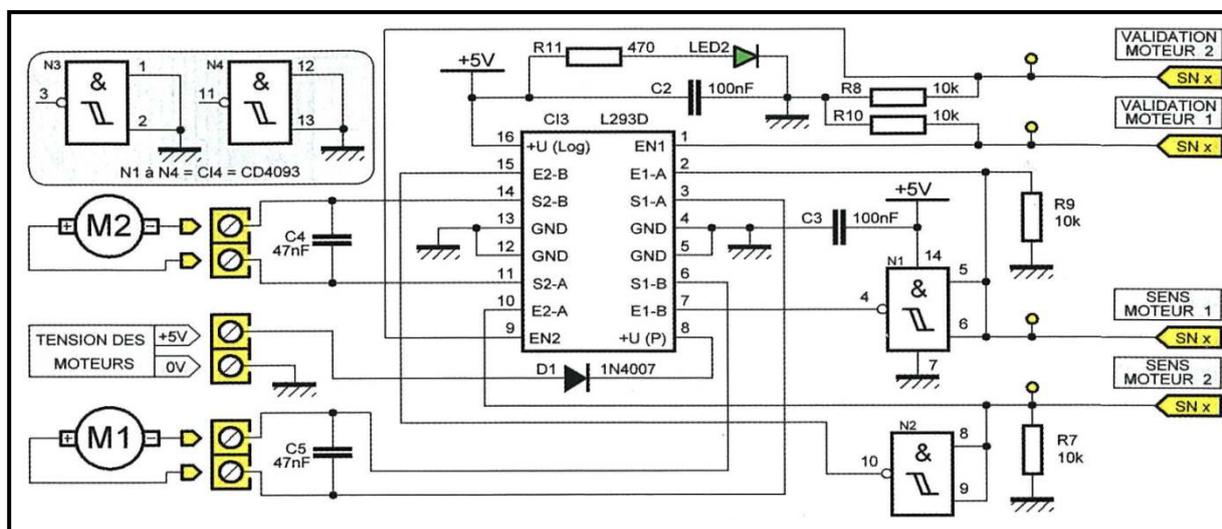


Figure 2.9 : circuit de commande de deux moteurs DC

Pour commander deux moteurs, nous employons le circuit spécifique L293D dédié à cette tâche et intégrant les deux ponts en H ainsi que les diodes de roue libre. Pour fonctionner, chaque pont de puissance requiert trois signaux logiques.

Un premier pour la mise sous tension, les deux autres pour déterminer le sens de rotation du moteur. Afin de réduire le nombre de lignes de commande de trois à deux par moteur, nous employons une porte logique NON-ET montée en « inverseur » pour obtenir directement les niveaux logiques corrects opposés pour sélectionner le sens de rotation.

Le schéma du brochage du circuit intégré L293D est montré sur la figure 2.9.

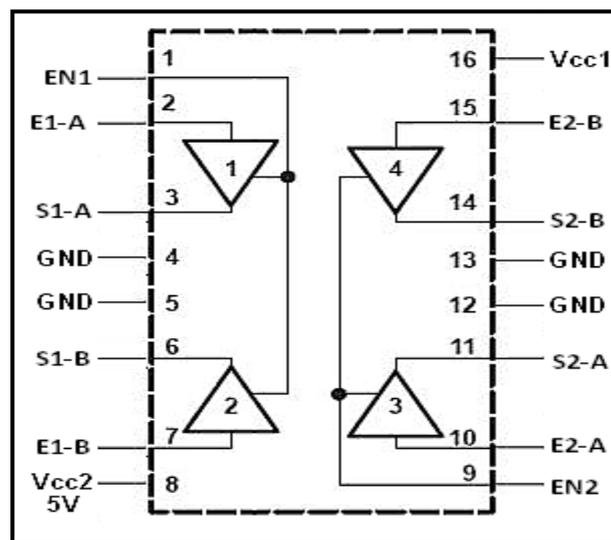


Figure 2. 10 : schéma de brochage du circuit L293D

Les pins Enable 1,2 servent pour sélectionner ou valider les deux moteurs 1,2 et les entrées input 1,2 servent pour choisir le sens de rotation du moteur 1, la même chose pour les entrées input 3,4 pour le deuxième moteur.

Les résistances R7 à R10 positionnent les lignes de commande au niveau logique « bas » au repos.

La tension de puissance destinée aux moteurs provient de l'extérieur et arrive sur la broche 8 de CI3, via une diode de protection.

Nous avons choisi de faire tourner des petits moteurs (200 mA à 300 mA). Si on souhaite travailler avec de plus fortes puissances, il conviendra d'opter pour une diode plus puissante (1N5404), sans jamais dépasser 500 mA par moteur, limite du L293D.

Les condensateurs C2 et C3 découplent la tension au plus près des circuits intégrés. L'antiparasitage des moteurs est assuré par les condensateurs C4 et C5. Nous opterons, de

préférence, pour des modèles « céramique ». Les entrées des deux portes NON-ET non utilisées sont raccordées à la masse. Enfin, la led2, limitée en courant par la résistance R11, visualise la mise en service de ce périphérique.

2.4.2 La commande d'un moteur pas à pas unipolaire

Ce type de moteur se trouve dans un grand nombre de périphériques informatiques : imprimantes, lecteur de disquettes ou disque dur, car il s'agit du composant mécanique par excellence pour tout ce qui demande une grande précision de positionnement (voir figure 2.11).

2.4.2.1 Présentation du moteur pas à pas

Le moteur pas à pas est un moteur qui tourne en fonction des impulsions électriques reçues dans ses bobinages. L'angle de rotation minimal entre deux modifications des impulsions électriques s'appelle un pas. On caractérise un moteur par le nombre de pas par tour (c'est à dire pour 360°). Les valeurs courantes sont 48, 100 ou 200 pas par tour.

2.4.2.2 constitutions d'un moteur pas à pas

Le rotor est constitué d'un aimant permanent, et le stator comporte 2 bobinage (ou 2 groupe de bobinage). En contrôlant l'alimentation des bobines, et le sens du courant dans celles-ci, on peut faire varier le champ dans le moteur (voir figure 2.12).

Dans notre cas on a utilisé un moteur pas à pas 55SMP25D6G avec une tension d'alimentation de 6V et une résistance de phase de 6.5Ω . La Longueur de l'arbre est de 10 mm, le diamètre de l'arbre est de 4 mm, avec pignon à 20 dents.

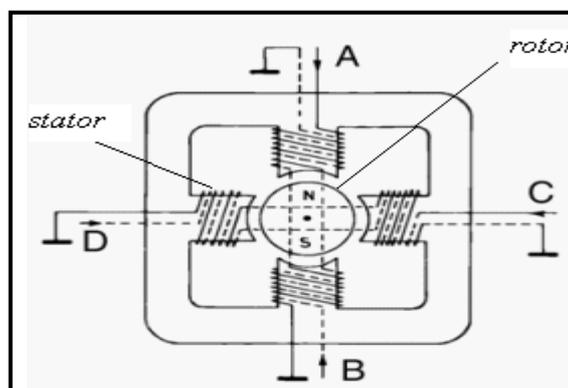


Figure 2. 11 : schéma simplifié d'un moteur pas à pas

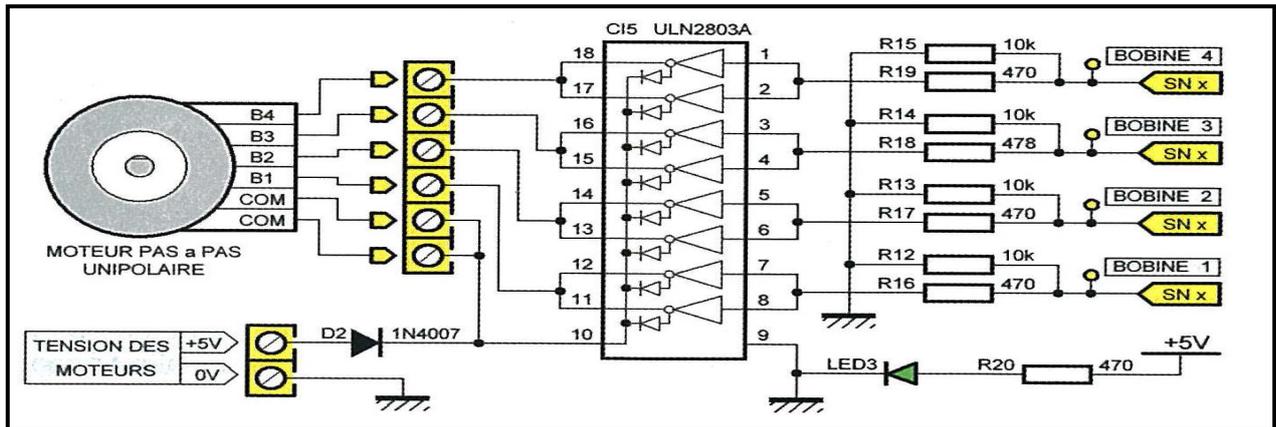


Figure 2. 12 : circuit de commande d'un moteur pas à pas unipolaire

Un tel moteur, contrairement aux moteurs à courant continu, est constitué de quatre bobines câblées en interne, avec un commun par couple. Nous obtenons donc un brochage externe de six fils, dont deux communs à relier au même potentiel.

Les bobines (ou électroaimants) se commandent individuellement pour faire avancer le rotor du moteur, d'un pas, ou d'un demi pas. La vitesse de rotation d'un tel moteur n'est pas très élevée, mais nous y gagnons en couple et en précision.

La faible courant issu des sorties de l'automate ne peut en aucun cas alimenter la bobine d'un moteur. Nous faisons appel au circuit CI5 : un ULN2803 renfermant huit étages « inverseurs » à transistors à collecteurs ouverts, mais intégrant les diodes de protections anti-retour. Afin de commuter plus de puissance, nous avons couplé les étages en « parallèle ».

Les résistances R12 à R15 positionnent les lignes de commande au niveau logique « bas » au repos. Les signaux parviennent aux entrées de CI5, via les résistances R16 à R19. Le commun du moteur est porté au positif d'une alimentation externe, adaptée, via la diode D2 qui supporte que 1 A.

La led3, limitée en courant par la résistance R20, visualise la liaison de ce périphérique à l'automate.

2.4.3 La commande d'un servomoteur

Un servomoteur est un système qui a pour but de produire un mouvement précis en réponse à une commande externe. Ce circuit consiste à commander l'angle de rotation de ce servomoteur (voir figure 2.13).

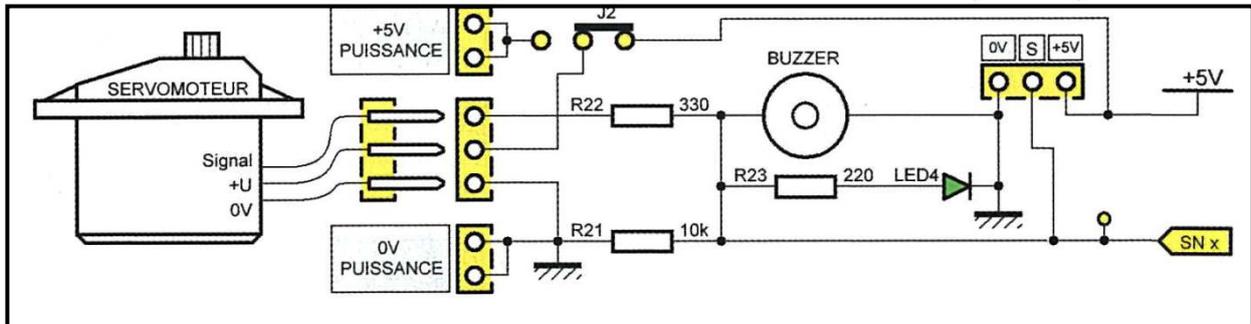


Figure 2. 8 : circuit de commande d'un servomoteur

Un même signal produit par l'automate peut convenir à commander trois périphériques, soit, le servo, le buzzer et la led4, sans conflit.

Le servomoteur requiert des impulsions régulières modulées en largeur. La résistance R22 assure une protection et R21 positionne la ligne de commande au niveau logique « bas » au repos. Un buzzer piézo est déjà intégré à l'automate, mais il peut être utile d'en raccorder un second, sur périphérique. Il s'alimente directement par une sortie produisant une fréquence audible (dans ce cas, il est préférable de ne pas raccorder un servomoteur). La led4, limitée en courant par la résistance R23, se relie directement à une sortie et peut s'accommoder à n'importe quel type de signal.

Dans notre cas on a utilisé le servomoteur HS-75BB dont les caractéristiques sont les suivantes :

- Couple de sortie 6.6Kg.cm à 4.8 V.
- Vitesse de rotation de 0.45 sec/60° à 4.8 V.
- Angle de fonctionnement est de 180°.
- Pignons avec nylon.
- Taille de 44 x 23 x 25 mm.
- Poids de 35 g.

2.4.4 La commande d'un relais

Un relais est un appareil dans lequel un phénomène électrique (courant ou tension) contrôle la communication On/Off d'un élément mécanique (voir figure 2.14).

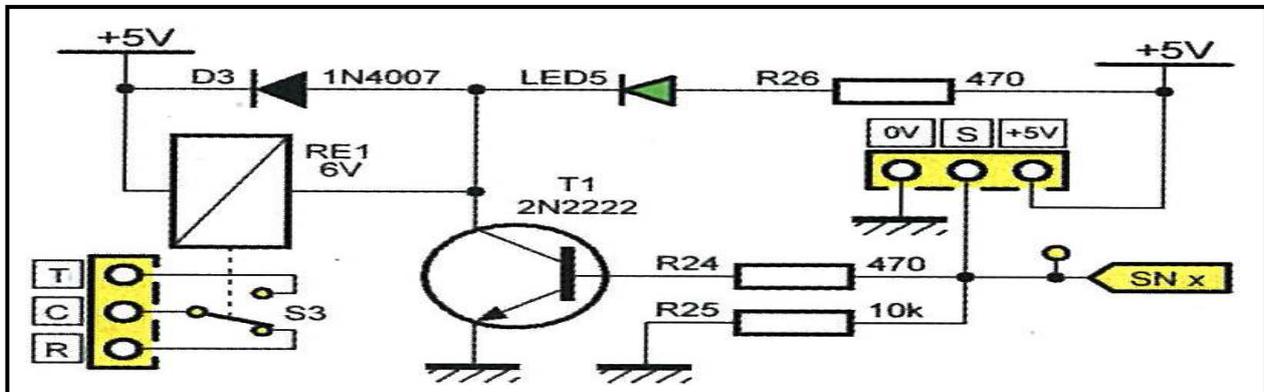


Figure 2. 9 : circuit de commande d'un relais

Ce périphérique permet de commuter toute charge via les contacts secs d'un relais.

Le signal, issu d'une sortie, attaque la base du transistor NPN/T1, via la résistance R24. Au repos, la base est positionnée à la masse par la résistance R25. Le collecteur alimente directement la bobine du relais en 4.5V (un relais de 5V à 6V convient parfaitement).

La diode D3 protège T1 des courants de retour. La led5, limité en courant par la résistance R26, visualise l'activation du relais.

Dans ce montage, on a utilisé un relais de 5 V. Le relais doit se fermer lorsque la sortie de l'automate est égale 1 et il donne une valeur de 3 V à la sortie du périphérique.

2.4.5 La lecture de la valeur d'une résistance ajustable

Une résistance variable est une résistance dont la valeur est variable. Dans cette application on a utilisé cette résistance variable comme une entrée de notre automate (voir figure 2.15).

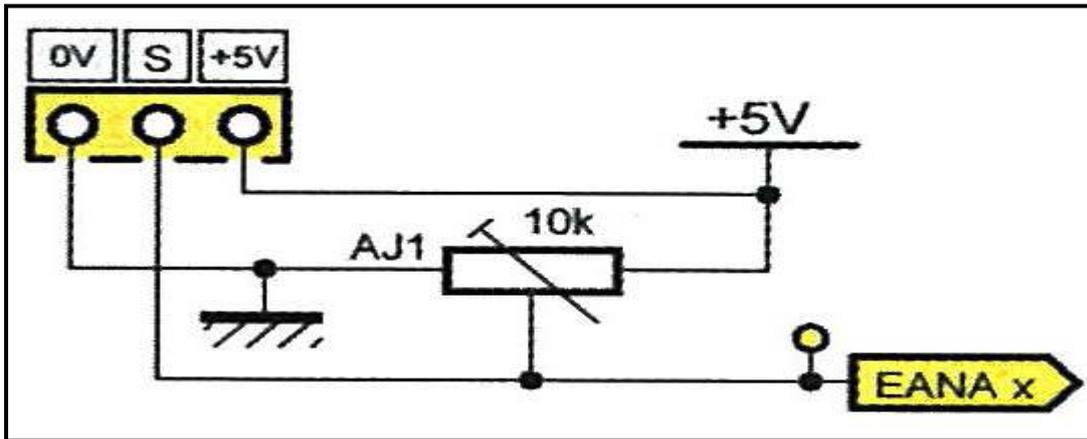


Figure 2. 10 : la lecture de la valeur d'une résistance ajustable

Il s'agit de faire lire à l'automate, par le biais d'une entrée du convertisseur analogique/numérique configuré en 8 ou 10 bits, la position du curseur de la résistance ajustable AJ1, dont les extrémités sont alimentées entre la masse (0V) et le +5V.

2.4.6 La lecture de la valeur d'une photorésistance

Les photorésistances sont des composants photosensibles dont la valeur ohmique varie en fonction de l'intensité de lumière qu'ils reçoivent.

Une photorésistance mesurée dans l'obscurité a une valeur d'environ 1 mégohm. Si elle reçoit un peu de lumière sa valeur s'abaissera immédiatement aux environs de 400 k Ω . Si l'intensité de la lumière augmente, sa valeur tend vers les 80 k Ω . Si elle reçoit une lumière forte, sa résistance descendra jusqu'à quelques dizaines d'ohms (voir figure 2.16) [11].

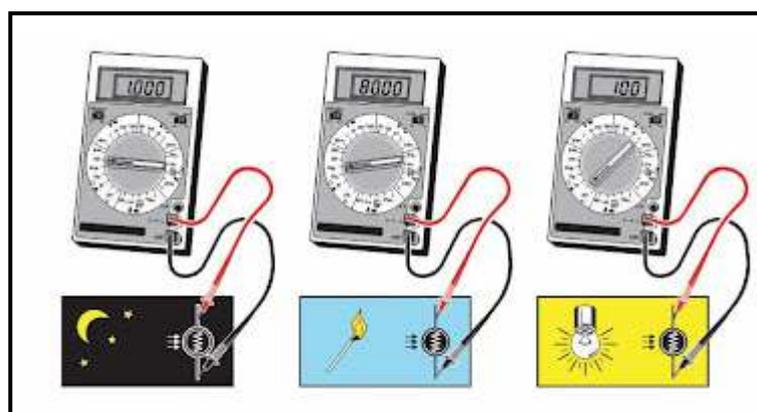


Figure 2. 11 : schéma du teste d'une photorésistance

Les photorésistances sont utilisées pour la réalisation d'automatismes capables de fonctionner en présence d'une source lumineuse. Dans ce montage, on a réalisé le capteur de lumière suivant (voir figure 2.17) :

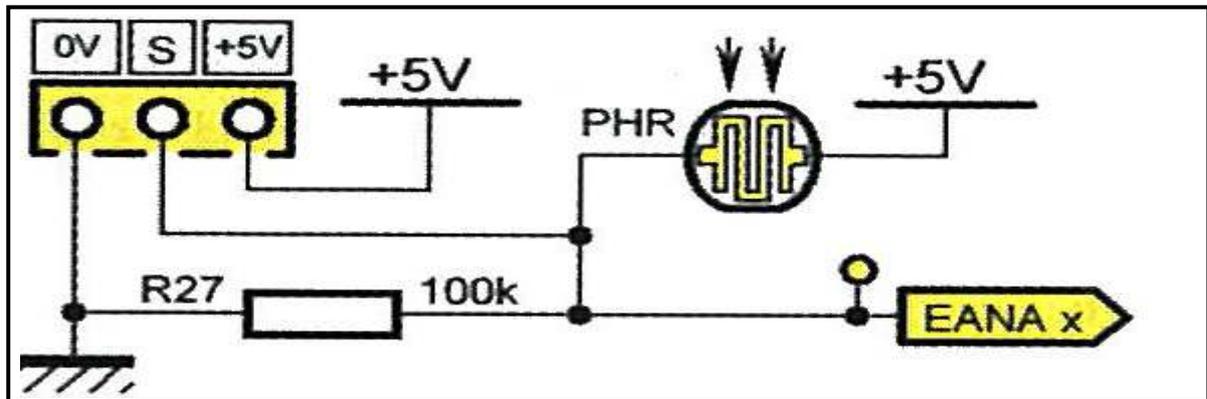


Figure 2. 12 : la lecture de la valeur d'une photorésistance

La photorésistance PHR est alimentée entre +5V et la masse via la résistance R27. Le signal est prélevé au point commun entre la photorésistance et R27. Il peut être lu par n'importe quelle entrée du convertisseur analogique /numérique configuré en 8 ou 10 bits pour la visualisation sur l'afficheur LCD.

2.4.7 La lecture de l'état logique d'une touche

Le rôle de ce montage est de tester une entrée numérique parmi les 8 de notre automate (voir figure 2.18).

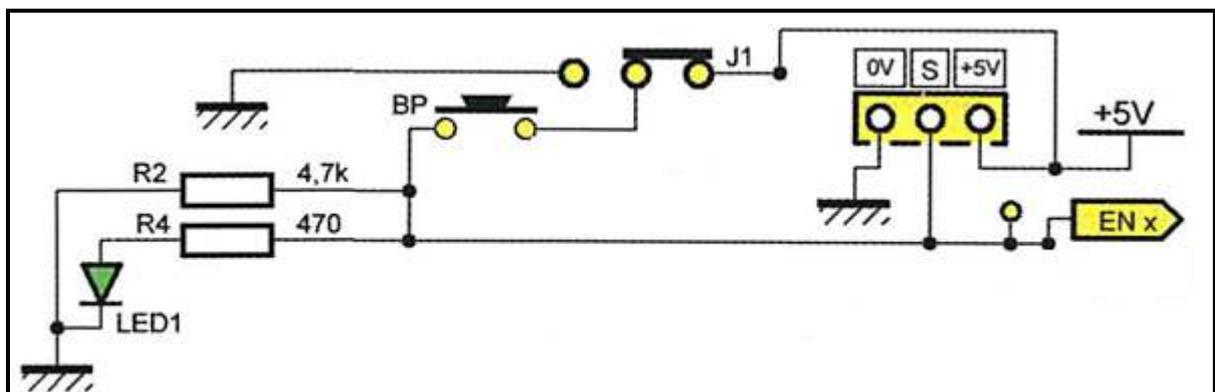


Figure 2. 13 : la lecture de l'état logique d'une touche

La touche BP peut « forcer » une entrée numérique de l'automate au niveau logique « haut » (+5V) ou « bas » (0V) en fonction de la position du cavalier J1. Selon notre cas, il conviendra de câbler la résistance R2. Celle-ci positionne l'entrée de l'automate au niveau logique correct.

Lorsque J1 est orienté vers le +5V, la led1 s'illumine lors de l'appui sur BP. Son courant est limité par la résistance R4.

2.5 Conclusion

Dans ce chapitre, nous avons présenté les différents blocs de notre automate ainsi que les périphériques réalisés.

Une fois le fonctionnement compris, nous allons procéder dans le chapitre suivant à la réalisation de ces différents éléments.

3.1 Introduction

Dans le chapitre deux, nous avons défini les différents blocs constituant notre automate programmable. Dans ce chapitre, nous allons présenter la simulation, la réalisation, la programmation et quelques commandes que nous avons réalisées pour prouver le bon fonctionnement de notre automate programmable.

3.2 Simulation sur PROTEUS et réalisation

3.2.1 Simulation du bloc d'alimentation

Avant de réaliser le circuit imprimé de notre automate programmable, on simule son bloc d'alimentation sur le logiciel de routage PROTEUS. La figure 3.1 montre le circuit simulé.

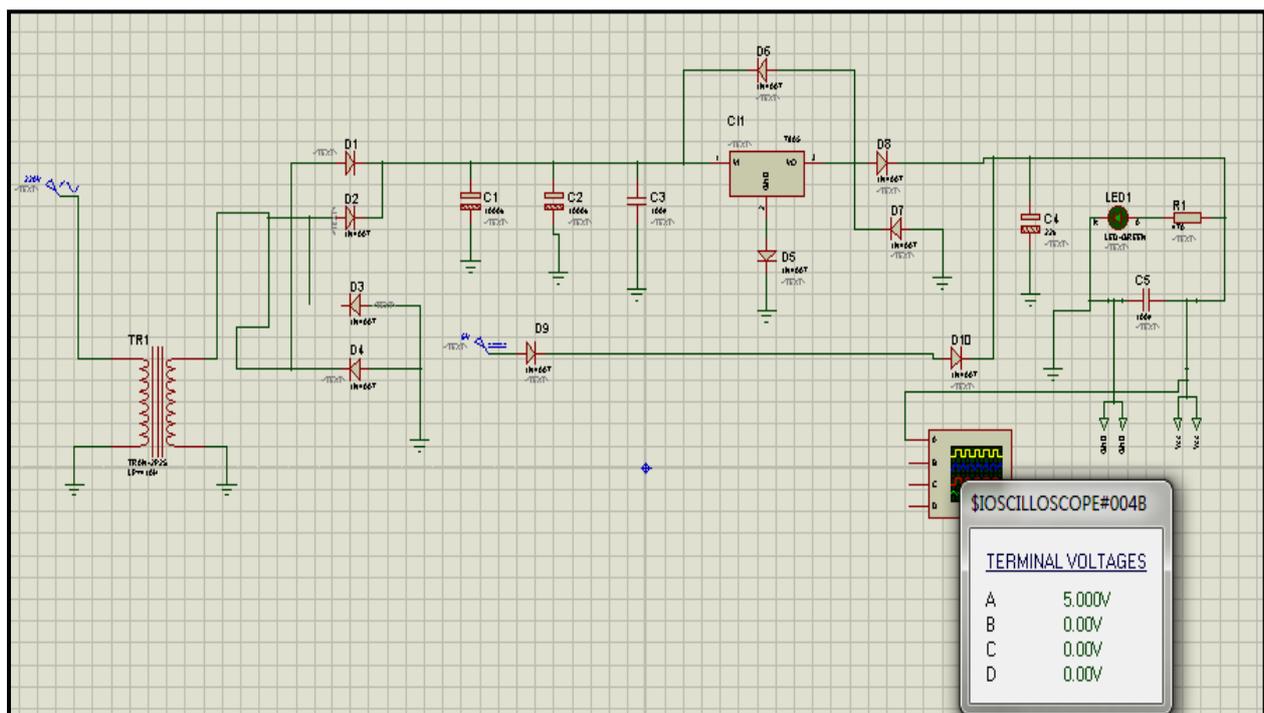


Figure 3. 1 : Bloc d'alimentation sur PROTEUS

Comme il est précisé dans le chapitre précédent, l'alimentation délivre 5V pour alimenter les différentes parties de l'automate programmable, la simulation nous a permis de visualiser la tension de sortie à l'aide de l'oscilloscope (voir figure 3.2).

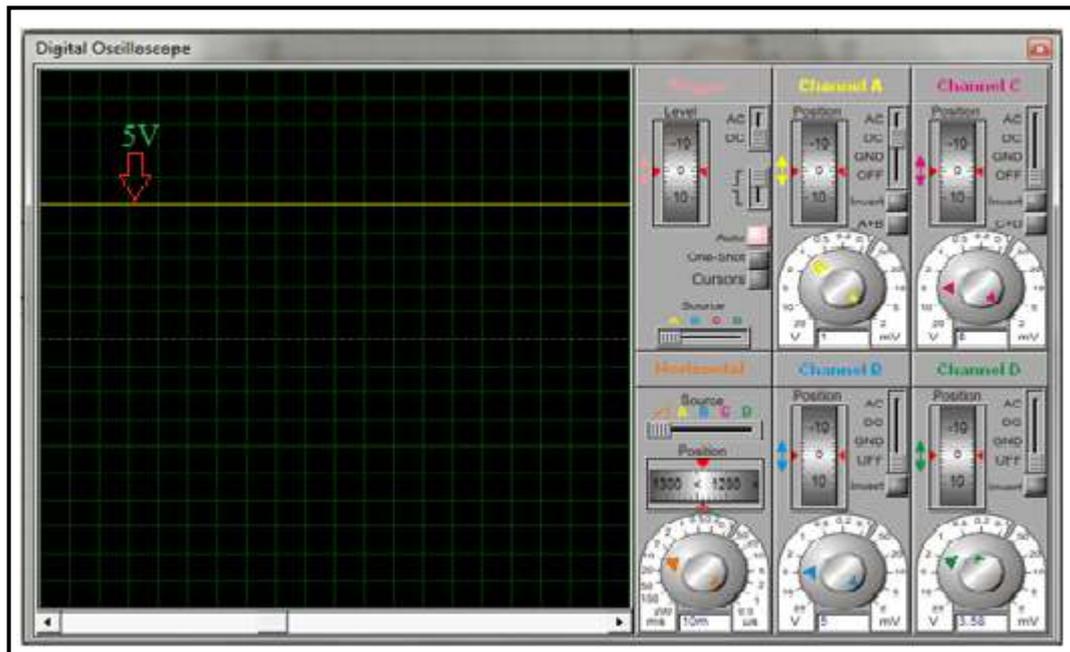


Figure 3. 2 : visualisation de la tension de sortie

3.2.2 La carte principale

Nous avons effectué le raccordement des différentes parties constituant notre automate programmable, la simulation sur PROTEUS de la carte principale est présentée sur la figure 3.3.

ponts de liaisons (straps), suivis par les résistances, les diodes, les supports de circuits intégrés, le connecteur de l'afficheur LCD constitué de quatorze broche de « barrette sécable » femelles, les condensateurs au mylar, la résistance ajustable AJ1, le buzzer piézo, les embases en commençant par celle de programmation et en terminant par les MJ/4P4C, les condensateurs électrochimiques et enfin le régulateur CII sur son dissipateur thermique.

La touche RST est placée 2 cm au dessous de la face avant, afin de pouvoir l'actionneur avec une tige fine par un petit orifice. L'afficheur LCD prend place sur son connecteur. Pour ajuster sa hauteur, il suffit d'embrocher des connecteurs les uns sur les autres.

Après les différentes étapes citées précédemment, on a obtenu la plaque de circuit imprimé montrée sur la figure 3.4, avec tous les composants, l'afficheur LCD et les embases MJ/4P4C.

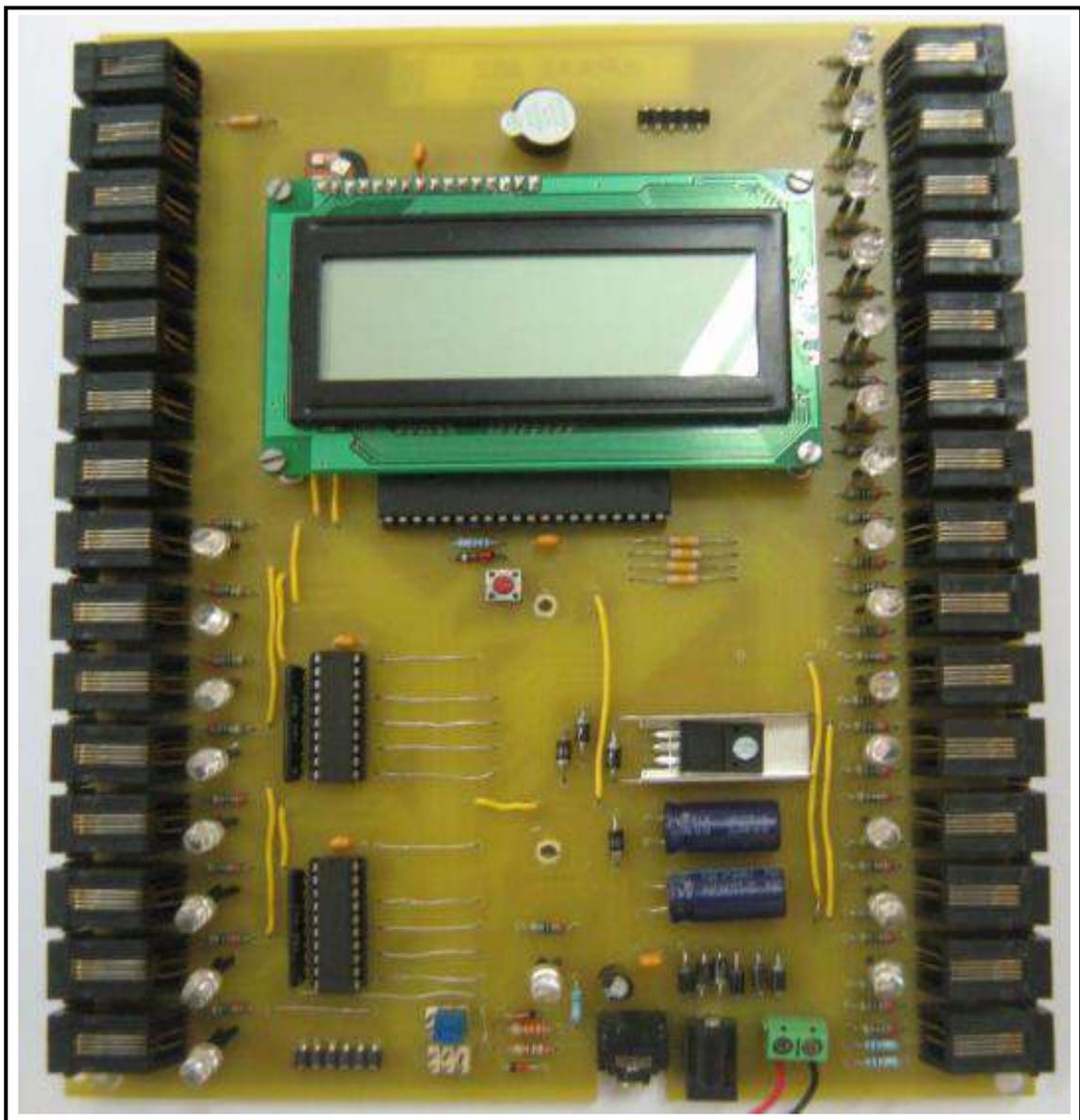


Figure 3. 4 : vue de l'automate réalisé

3.2.3 Les périphériques

Les périphériques sont conçus de manière modulaire, sachant que certains tiennent sur un même circuit câblé différemment. Comme pour l'unité centrale, on doit souder les composants par ordre de taille et de fragilité, en commençant bien évidemment par les ponts de liaisons.

3.2.3.1 Les entrées

Comme nous l'avons dit, notre automate contient des entrées numériques et analogiques, afin de les tester on va réaliser les périphériques suivants :

➤ Résistance ajustable

C'est un circuit simple qui contient une résistance ajustable et une embase MJ/4P4C pour la communication avec l'automate. La simulation de ce circuit est sur la figure 3.5. Le circuit réalisé de cette entrée est montré sur la figure 3.6.

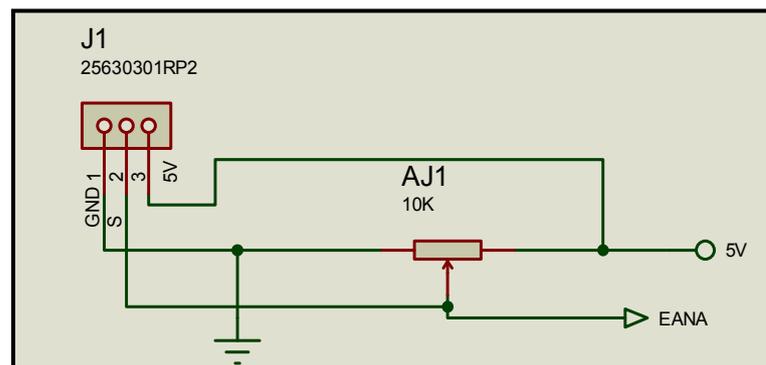


Figure 3. 5 : schéma de câblage de la résistance ajustable sur PROTEUS

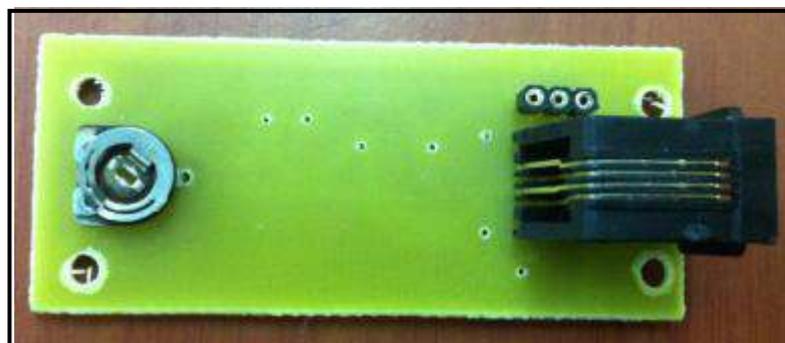


Figure 3. 6 : le circuit réalisé de la résistance ajustable

➤ Photorésistance

Pour la photo résistance, nous réalisons un circuit imprimé identique à celui de la résistance ajustable, mais on va le câbler différemment (voir figure 3.7). La figure 3.8 montre la plaque du circuit réalisé.

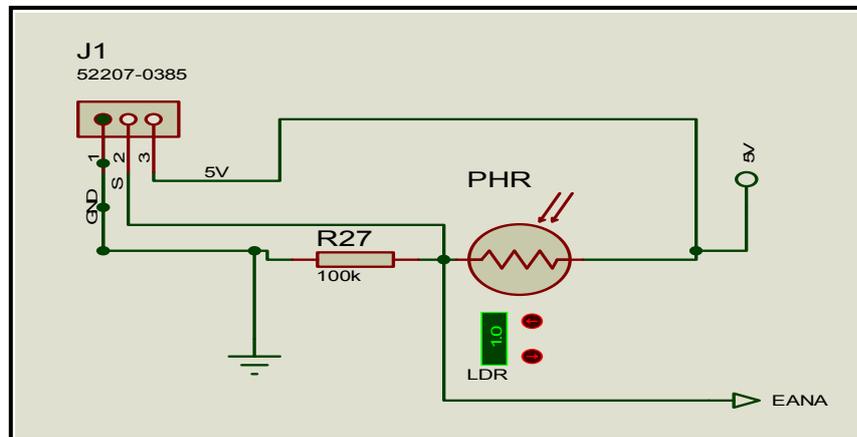


Figure 3. 7 : schéma de câblage de la photorésistance sur PROTEUS

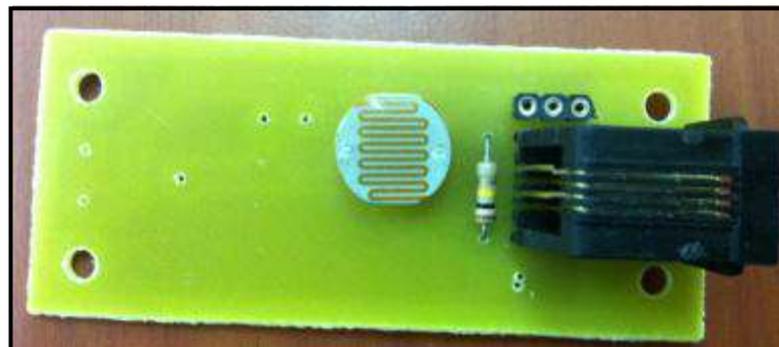


Figure 3. 8 : le circuit réalisé de la photorésistance

➤ Etat logique d'une touche

Notre automate peut avoir une information sur l'état logique d'une touche. Le circuit réalisé contient une touche, un cavalier pour la configuration et une embase pour la connexion avec l'automate (voir figures 3.9 et 3.10).

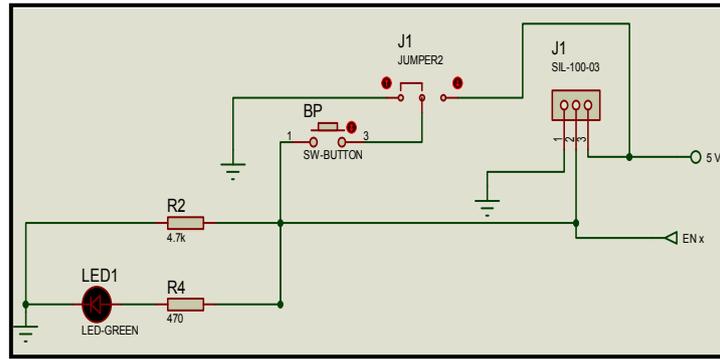


Figure 3. 9 : le schéma de câblage de circuit de l'état logique d'une touche sur PROTEUS



Figure 3. 10 : le circuit réalisé de l'état logique d'une touche

3.2.3.2 Les sorties

Afin de tester les différentes sorties de notre automate, nous avons donc réalisé les différents montages suivant :

➤ **Commande de deux moteurs à courant continu**

Pour commander deux moteurs à courant continu, dont le but est de commander le déplacement d'un robot ou d'un mobile par exemple, nous avons réalisé la carte de puissance détaillée dans le chapitre deux.

La simulation de ce circuit sur PROTEUS apparaît sur la figure 3.11 et la plaque de circuit réalisé est montrée sur la figure 3.12. Le circuit contient les composants nécessaires et quatre embases MJ/4P4C, deux pour la validation des deux moteurs et les deux autres pour le choix de sens.

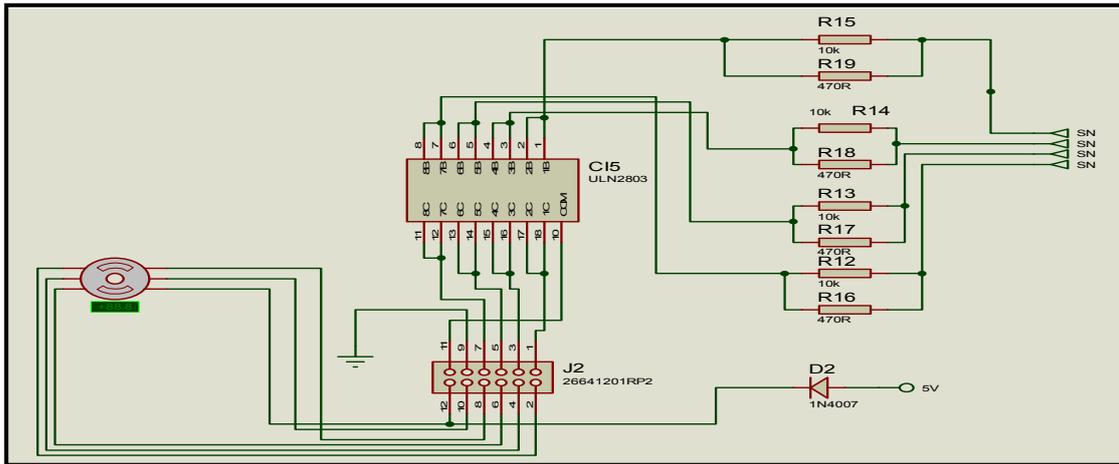


Figure 3. 13 : schéma de câblage de la carte de puissance du moteur pas à pas sur PROTEUS

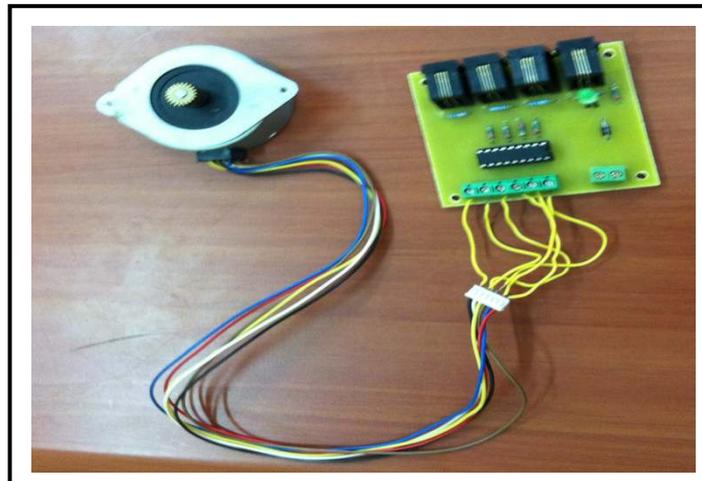


Figure 3. 14 : le circuit réalisé de la carte de puissance du moteur pas à pas

➤ Commande d'un relais

Notre automate peut commander un relais, nous avons détaillé son circuit de commande dans le chapitre deux. La simulation de ce circuit est montrée sur la figure 3.15 et la plaque imprimé est présentée sur la figure 3.16.

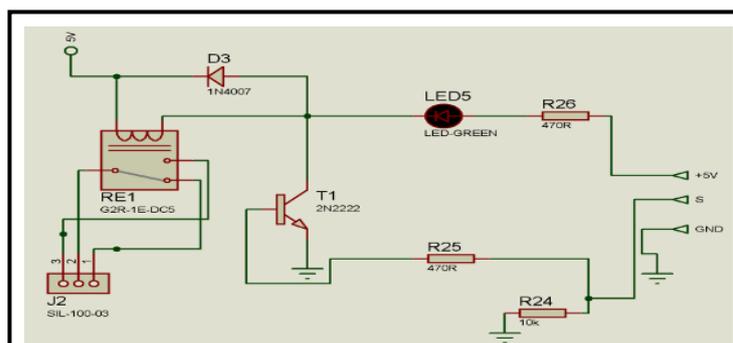


Figure 3. 15 : le schéma de câblage du relais sur PROTEUS



Figure 3. 16 : le circuit réalisé de commande du relais

➤ **Commande d'un servomoteur**

Nous avons réalisé un circuit qui commande un servomoteur, un buzzer et une led. On s'est intéressé au servomoteur. La figure 3.17 décrit la simulation de ce circuit et la figure 3.18 expose la plaque du circuit réalisé.

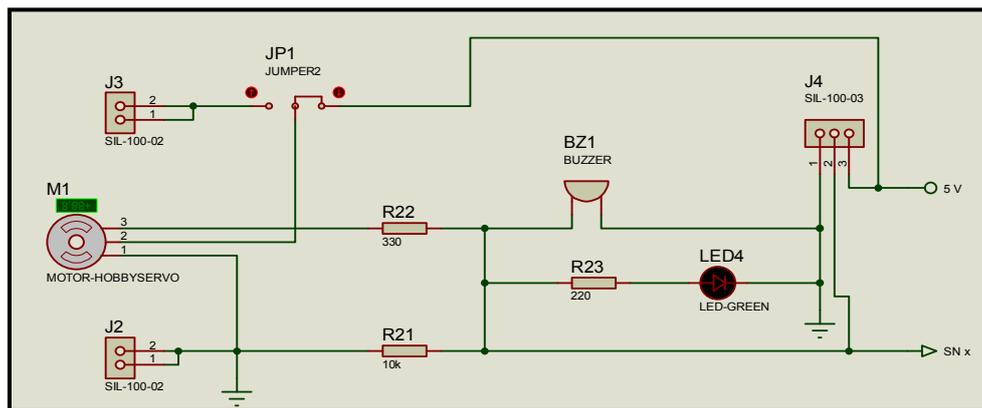


Figure 3. 17 : le schéma de câblage de circuit de commande du servomoteur sur PROTEUS

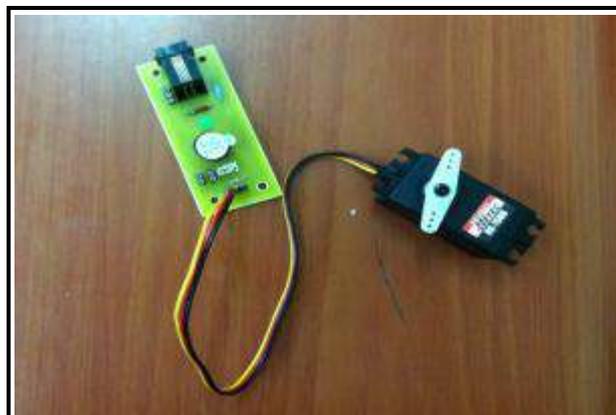


Figure 3. 18 : le circuit réalisé de commande du servomoteur

3.2.4 Connexion entre l'automate et les périphériques

Nous allons parler ici des embases femelles MJ/4P4C (figure 3.19), qui se trouvent sur l'automate programmable jouant le rôle des connecteurs d'entrées/sorties, ainsi que sur les périphériques afin de les relier à l'automate.

Nous avons utilisé des câbles qui contiennent quatre fils à l'intérieur, à l'aide d'une pince spéciale, nous ajoutons aux extrémités de chaque câble des connecteurs mâle MJ/4P4C (figure 3.20), pour mieux décrire les câbles utilisés, on présente sur la figure 3.21 un câble exploité dans notre travail.

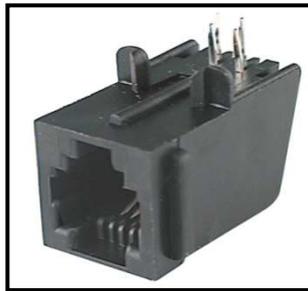


Figure 3. 19 : embase MJ4P4C femelle



Figure 3. 20 : connecteur MJ/4P4C mâle



Figure 3. 21 : câble utilisable

Toutes les entrées sorties aboutissent sur ces connecteurs, chacun d'eux comporte les lignes d'alimentation (0V et 5V), ainsi que le signal.

Notons la différence entre celles de l'automate et celles des périphériques, car la liaison par câble plat, à quatre conducteurs souples sur des prises mâles, n'est pas croisé, la figure 3.22 montre le brochage de ces embases.

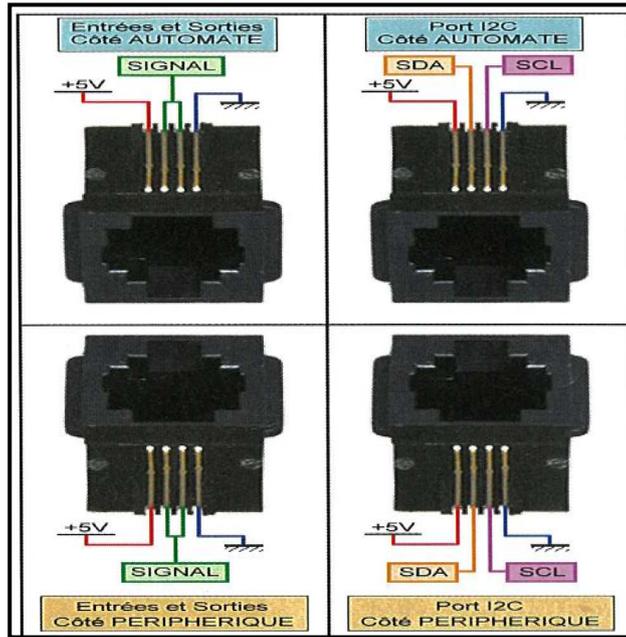


Figure 3. 22 : le schéma de brochage des embases MJ/4P4

La liste des composants utilisés est donnée en annexe C, et les circuits imprimés de toutes les plaques réalisées sont exposés en annexe D. la figure 3.23, montre le montage final de notre réalisation.

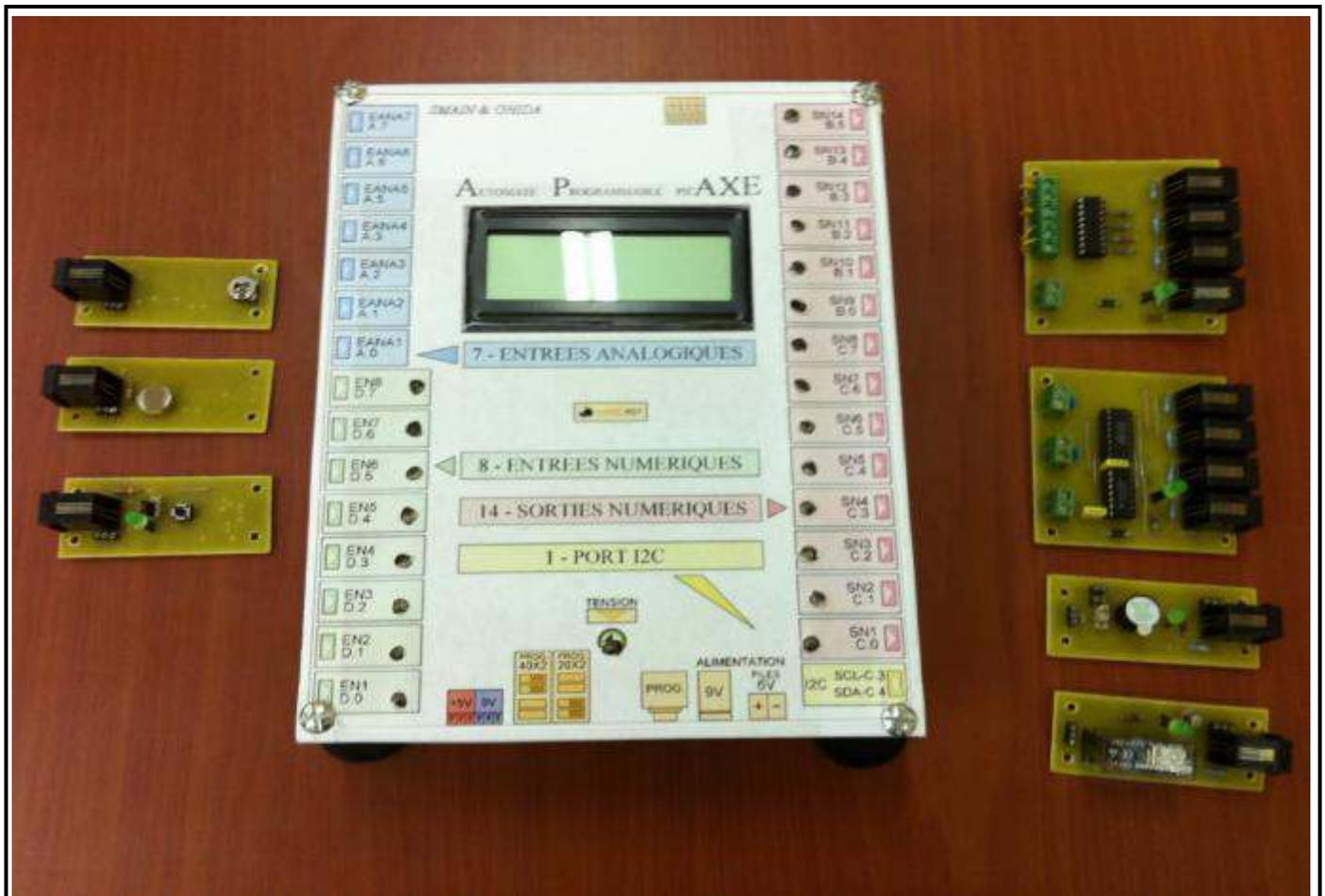


Figure 3. 23 : le montage final de notre réalisation

3.3 Programmation

Pour programmer notre automate programmable, nous avons commencé par télécharger le logiciel *PICAXE Programming Editor*.

Au premier lancement du logiciel, on se trouve devant la fenêtre des options. Si tels n'est pas le cas, on clique sur l'icône « option » pour l'ouvrir.

Certaines options se règlent une fois pour toutes (langue et paramétrage de l'éditeur), d'autres doivent être ajustées à chaque utilisation (microcontrôleur et port « sériel » employés). Un clic sur le bouton « check Firmware Version... » permet de connaître la version du PICAXE, si tous les raccordements sont corrects.

Il existe deux manières de programmer un PICAXE : par diagramme **Flowchart** ou en **Basic**, les figures 3.24 et 3.25 présentent une vue d'écran des deux modes de programmation.

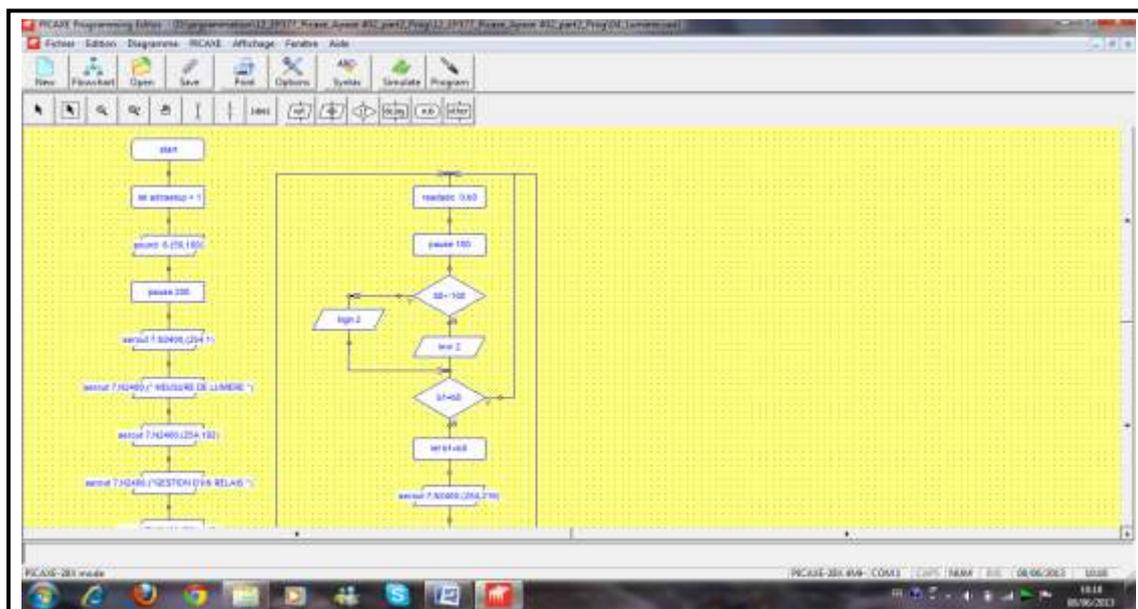


Figure 3. 24 : vue du logiciel de programmation en Flowchart

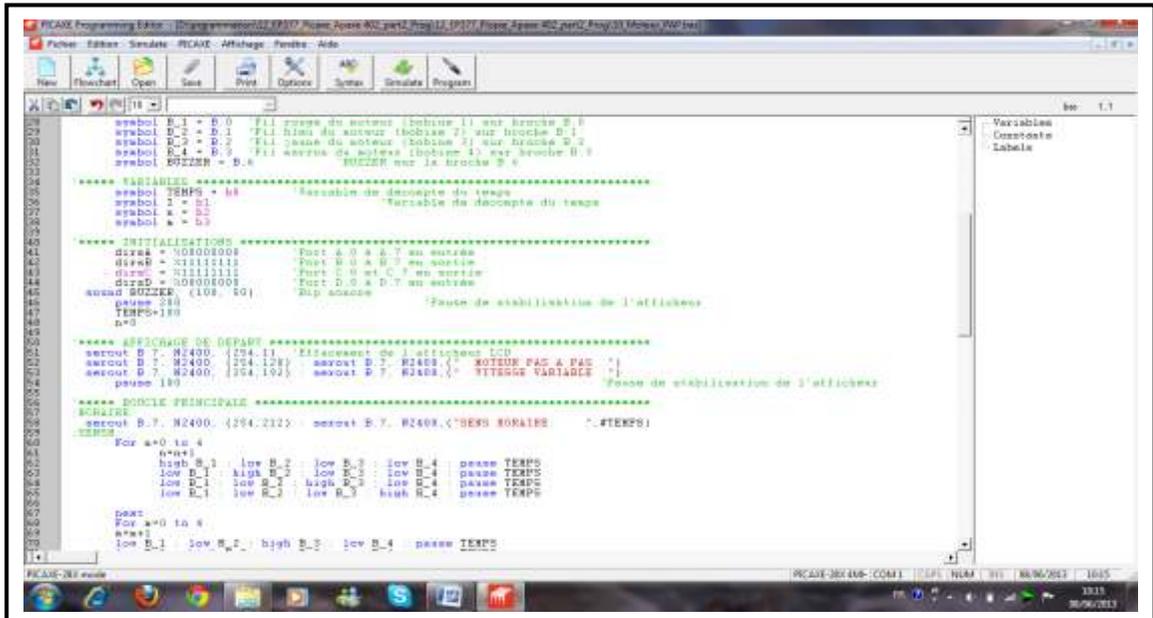


Figure 3. 25 : vue du logiciel de programmation en Basic

La première méthode est destinée à l'apprentissage, car elle offre un environnement visuel dans lequel les instructions prennent la forme des symboles formant un organigramme. En contrepartie, il est difficile d'entreprendre de longs développements et d'accéder à toutes les fonctions. La seconde façon de programmer, bien plus professionnelle, permet d'envisager les applications les plus sophistiquées, sans réelles restrictions, compte tenu des capacités du microcontrôleur utilisé, il est possible de convertir le diagramme, en Basic (menu « PICAXE », et sous-menu « Convertir le Diagramme en Basic.... »).

Comme toutes les broches ne sont pas attribuées en mode Flowchart, car le logiciel considère un PICAXE 40X2 comme un PICAXE 28X2, le tableau 3.1 et le tableau 3.2 indiquent les broches utilisées et leur sens dans les deux modes de programmation.

Tableau 3. 1 : attribution des entrées et sorties du 40X2 dans la programmation en Basic

A.0 : EANA 1	B.0 : SN 9	C.0 : SN 1	D.0 : EN 1
A.1 : EANA 2	B.1 : SN 10	C.1 : SN 2	D.1 : EN 2
A.2: EANA 3	B.2 : SN 11	C.2 : SN 3	D.2 : EN 3
A.3 : EANA 4	B.3 : SN 12	C.3 : SN 4	D.3 : EN 4
A.5: EANA 5	B.4 : SN 13	C.4 : SN 5	D.4 : EN 5
A.6 : EANA 6	B.5 : SN 14	C.5 : SN 6	D.5 : EN 6
A.7: EANA 7	B.6 : Buzzer	C.6 : SN 7	D.6 : EN 7
	B.7 : LCD	C.7 : SN 8	D.7 : EN 8

Tableau 3. 2 : attribution des entrées et sorties du 40X2 dans la programmation en Flowchart

A.0 : EA 1	B.0 : SN 0	C.0 : EN 0	D.0 : Rien
A.1 : EA 2	B.1 : SN 1	C.1 : EN 1	D.1 : Rien
A.2: EA 3	B.2 : SN 2	C.2 : EN 2	D.2 : Rien
A.3 : EA 4	B.3 : SN 3	C.3 : EN 3	D.3 : Rien
A.5: Rien	B.4 : SN 4	C.4 : EN 4	D.4 : Rien
A.6 : Rien	B.5 : SN 5	C.5 : EN 5	D.5 : Rien
A.7: Rien	B.6 : Buzzer	C.6 : EN 6	D.6 : Rien
	B.7 : LCD	C.7 : EN 7	D.7 : Rien

La programmation s’effectue ensuite très simplement grâce au port « sériel » ou USB.

Pour notre cas, nous avons utilisé le câble USB (*AXE027*) montré sur la figure 3.26.



Figure 3. 26 : câble AXE027

Nous positionnons les cavaliers de programmation dans la position de programmer le microcontrôleur (20X2) d’affichage, puis nous raccordons le cordon avec la prise « jack » entre l’automate et l’ordinateur, en lançant le chargement de programme par l’icône « programme » sous la barre des menus, afin de programmer en Basic l’afficheur et les cinq sorties auxiliaires, après on met les cavaliers de configuration sur le 40X2 et on effectue les différentes tests sur notre automate.

3.4 Les tests

3.4.1 En mode Flowchart

Dans le but de montrer le bon fonctionnement de notre automate dans le mode de programmation en Flowchart (diagramme), nous allons proposer quelques tests dans ce mode.

3.4.1.1 Test d'une touche

Nous reprenons le montage de la figure 2.18.

Nous effectuons le raccordement (figure 3.27) entre le périphérique de la touche et l'automate dans l'entrée numérique C0 (voir tableau 3.2).

Pour ce premier montage, il s'agit d'un simple test de l'état logique de la touche. Lors d'un appui sur la touche reliée à l'entrée EN0, la sortie SN0 passe à l'état logique 1. Au repos la sortie SN1 passe à l'état logique 1. Le programme se déroulera selon l'organigramme présenté sur la figure 3.28.

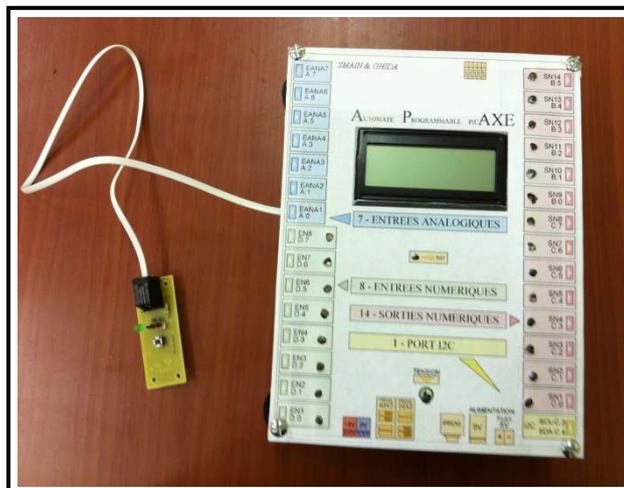


Figure 3. 27 : raccordement entre le périphérique de la touche et l'automate

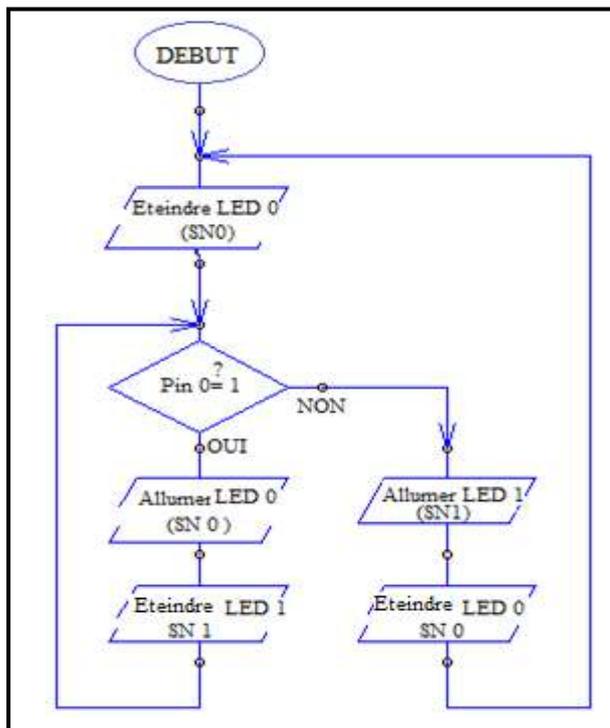


Figure 3. 28 : organigramme expliquant le test d'une touche

3.4.1.2 Lecture de la valeur d'une résistance ajustable

Pour réaliser cette tâche nous reprenons le circuit décrit dans la figure 2.15.

Nous effectuons le raccordement entre le périphérique de la résistance ajustable et l'automate (figure 3.29) dans l'entrée analogique EA2, dans le programme chargé dans le microcontrôleur, on utilise la commande *readadc*, pour lire la valeur analogique de la broche d'entrée EA2 du microcontrôleur, qui est utilisée pour chaque raccordement à une entrée analogique.

Le programme efface l'écran et affiche la valeur qui résulte de la position du curseur de la résistance ajustable, lue par l'entrée analogique n°2. Pour éviter le clignotement de l'affichage, la nouvelle valeur ne s'affiche que si elle diffère de la précédente (figure 3.30).

A ce sujet, notons le rôle des variables : b0 mémorise la valeur lue sur EA0, alors que b1 conserve la dernière donnée afin de permettre la comparaison avec b0.

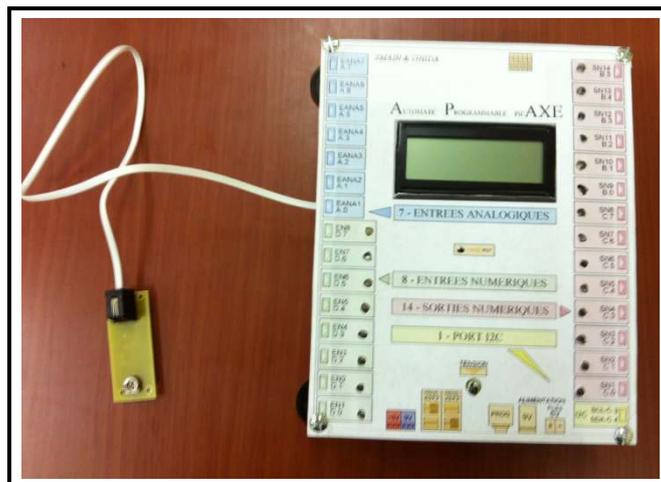


Figure 3. 29 : raccordement entre le périphérique de la résistance ajustable et l'automate

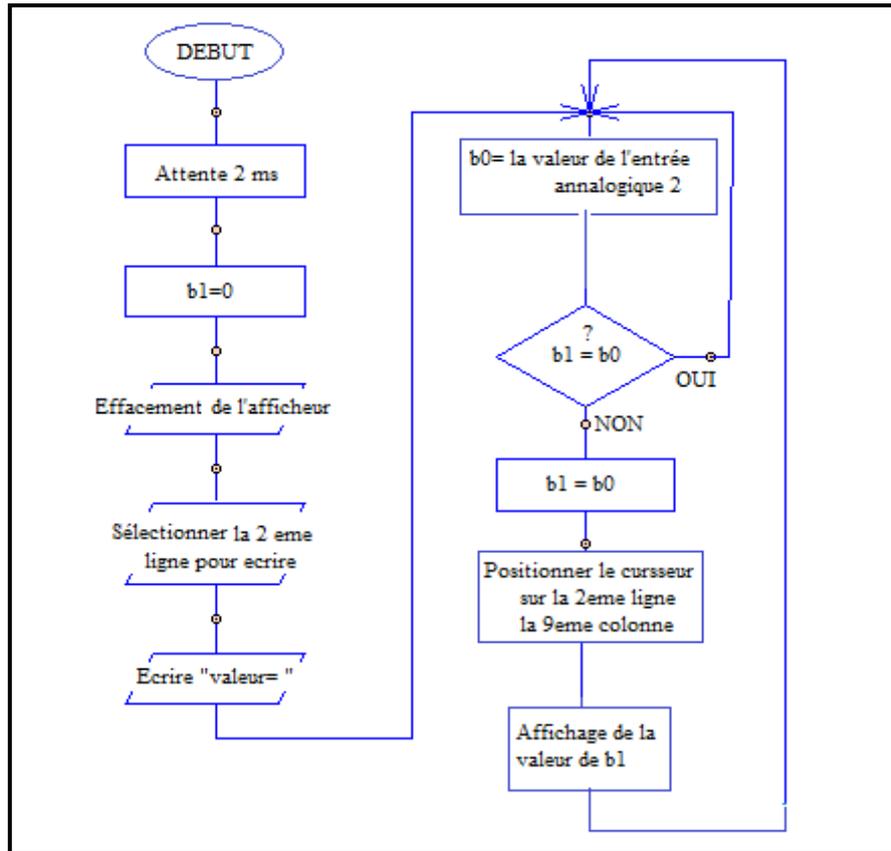


Figure 3. 30 : organigramme la lecture de la valeur de la résistance ajustable

3.4.1.3 Evaluation d'un niveau d'éclairement

Nous effectuons le raccordement entre le périphérique de la photorésistance et celui du relais avec l'automate (figure 3.31), les circuits détaillés de ces deux périphériques sont donnés sur les figures 2.14 et 2.17.

Après le chargement de programme dans le microcontrôleur, le buzzer émet un bip sonore, l'écran s'efface et affiche la valeur résultant de l'éclairement sur la photorésistance, lue par l'entrée analogique EA0, l'organigramme de la figure 3.32 montre le déroulement du programme.

A noter que pour éviter le clignotement de l'affichage, la nouvelle valeur ne s'affiche que si elle diffère de la précédente.

Si l'éclairement tombe sous le seuil de 100 qui correspond à une valeur de résistance égale à 5 k Ω , les contacts du relais raccordés sur la sortie SN2 se ferment.



Figure 3. 31 : raccordement du relais et de la photorésistance avec l'automate

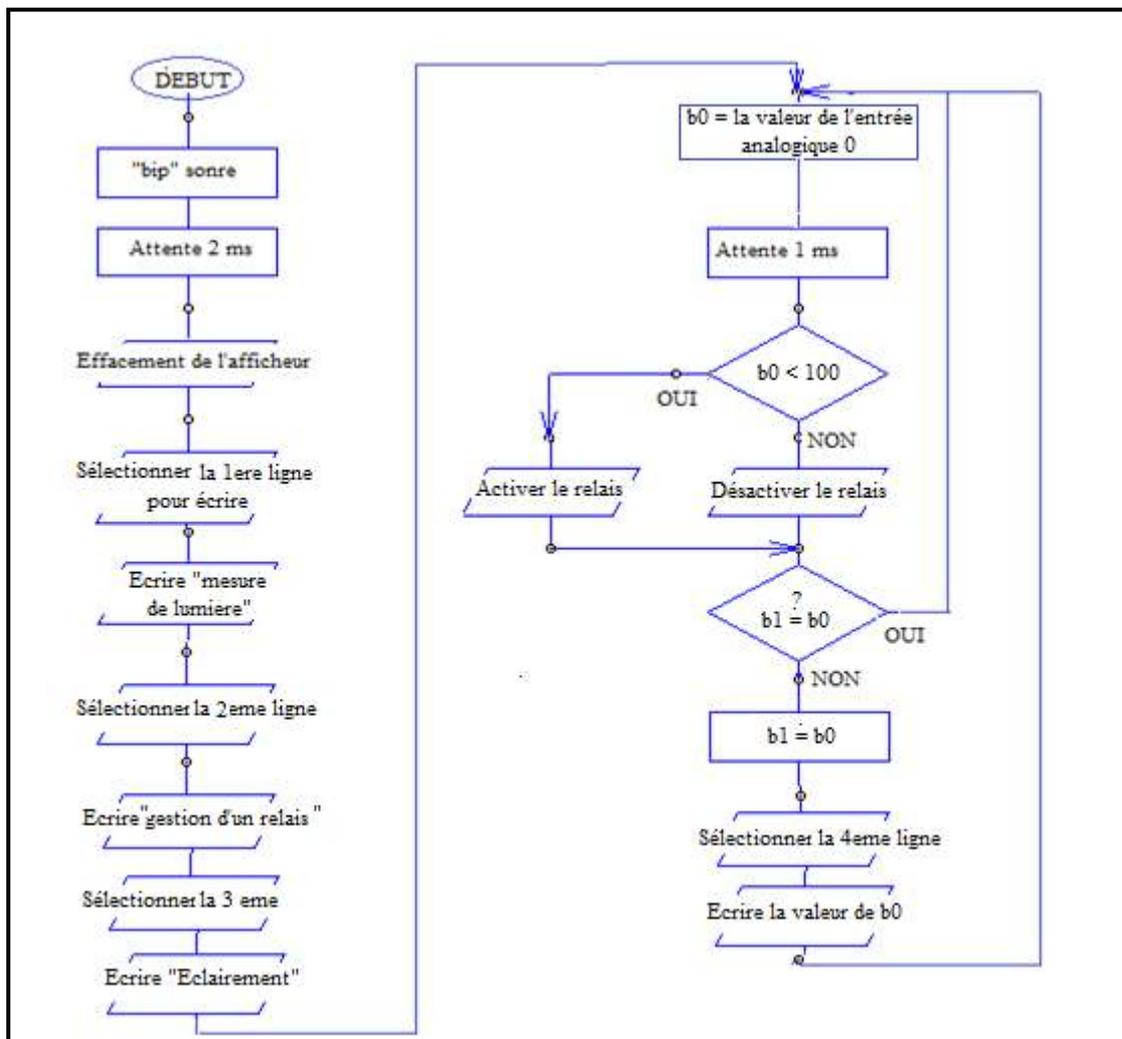


Figure 3. 32 : organigramme d'évaluation du niveau d'éclairment

3.4.1.4 Rotation d'un servomoteur

Comme il est montré sur la figure 3.33, nous effectuons le raccordement entre les périphériques de la résistance ajustable (figure 2.15) et celui destiné au servomoteur (figure 2.14) avec l'automate. On relie le servomoteur sur le connecteur du périphérique, on écrit le programme en utilisant la commande *servo*, puis on charge le fichier de programmation dans le microcontrôleur.

Le programme émet un bip sonore, efface l'écran et affiche un message qui demande d'agir sur la résistance ajustable lue par l'entrée analogique EA0, pour modifier la position du servomoteur relié à la sortie SN5. La valeur lue s'affiche et n'est rafraîchie que si elle diffère de la précédente. L'organigramme de la figure 3.34 décrit le déroulement du programme.



Figure 3. 33 : raccordement du servomoteur et de la résistance ajustable avec l'automate

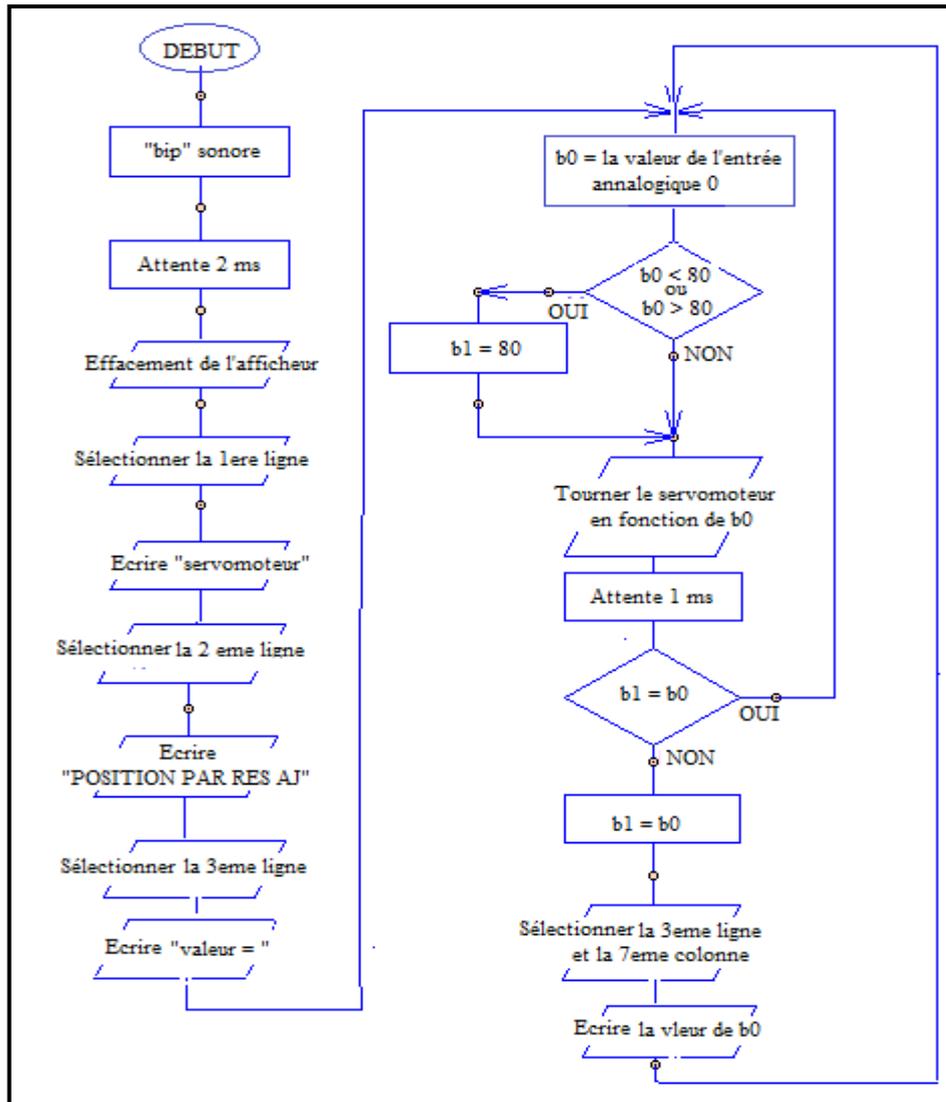


Figure 3. 34 : organigramme de commande du servomoteur

3.4.1.5 Commande de deux moteurs à courant continu

Nous reprenons le montage de la figure 2.9.

Nous effectuons le raccordement entre le périphérique de commande des deux moteurs à courant continu et l'automate (voir figure 3.36). On relie une alimentation en 9V et les deux moteurs sur les borniers à vis du périphérique.

Ce montage est censé montrer par exemple, comment gérer les déplacements d'un robot mobile équipé de deux moteurs DC. Il convient de paramétrer les options (icône « Option », puis onglet « Diagramme » et enfin, bouton « Déplacement ») conformément à la vue d'écran de la figure 3.35.

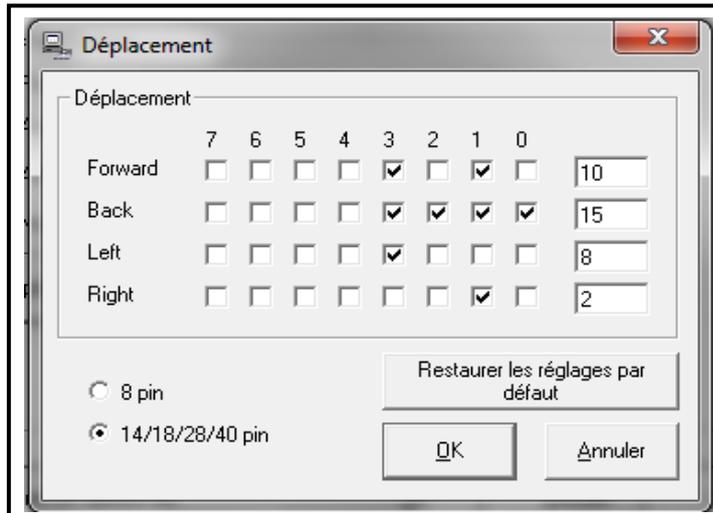


Figure 3. 35 : paramétrage de déplacement

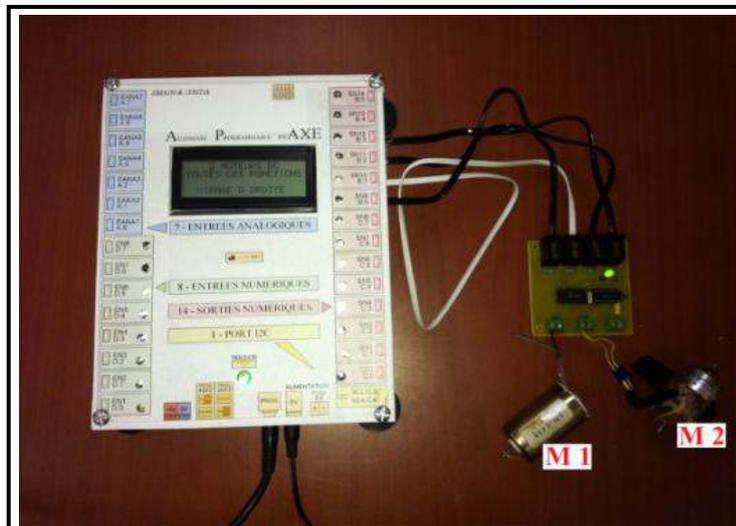


Figure 3. 36 : raccordement des deux moteurs avec l'automate

Le programme efface l'écran, puis affiche un message donnant le sens de rotation du mobile, en temps réel. Le déroulement du programme est décrit par l'organigramme sur la figure 3.37.

La validation du moteur 1 est confiée à la sortie SN3 et son sens de rotation à SN2.

La validation du moteur 2 est confiée à la sortie SN1 et son sens de rotation à SN0.

Les moteurs tournent comme s'ils commanderaient le déplacement d'un mobile, qui se déplace suivant le cycle perpétuel : marche « avant » de 3s, arrêt de 1s, marche « arrière » de 3s, arrêt de 1s, virage à gauche de 3s, arrêt de 1s, virage à droite de 3s et enfin arrêt de 1s.

3.4.2 En mode Basic

3.4.2.1 Commande d'un moteur pas à pas unipolaire

On reprend le montage de la figure 2.12.

Nous effectuons le raccordement entre le périphérique de commande du moteur pas à pas unipolaire et l'automate, nous relierons également une alimentation 12 V et les six fils du moteur, sur les borniers à vis du périphérique, en respectant bien les couleurs des fils du moteur.

Ce montage montre comment gérer la rotation dans les deux sens et en modifiant la vitesse, d'un moteur pas à pas unipolaire (voir figure 3.38).

On charge le programme dans le microcontrôleur, après un bip sonore de validation, le programme efface l'écran, puis affiche un message donnant le sens de rotation du moteur et le délai entre deux pas, en temps réel. Le déroulement du programme est décrit par l'organigramme de la figure 3.39.

La bobine 1 du moteur est commandée par la sortie SN9, la bobine 2 par la sortie SN10, la bobine 3 par la sortie SN11 et la bobine 4 par la sortie SN12.

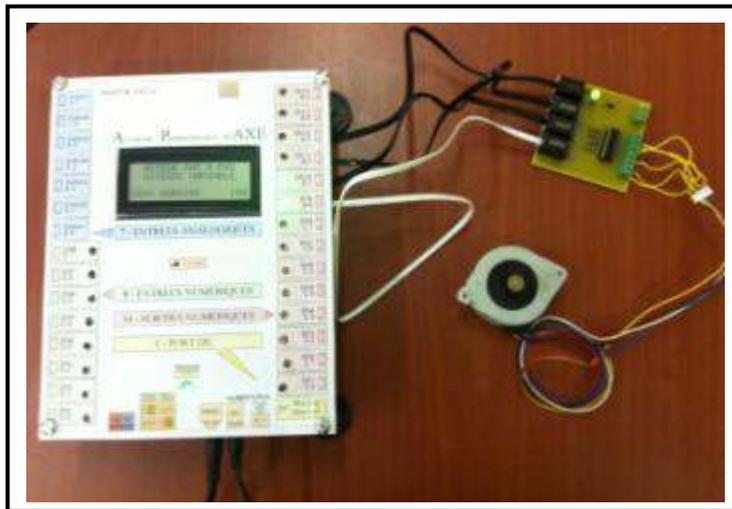


Figure 3. 38 : raccordement du moteur pas à pas avec l'automate

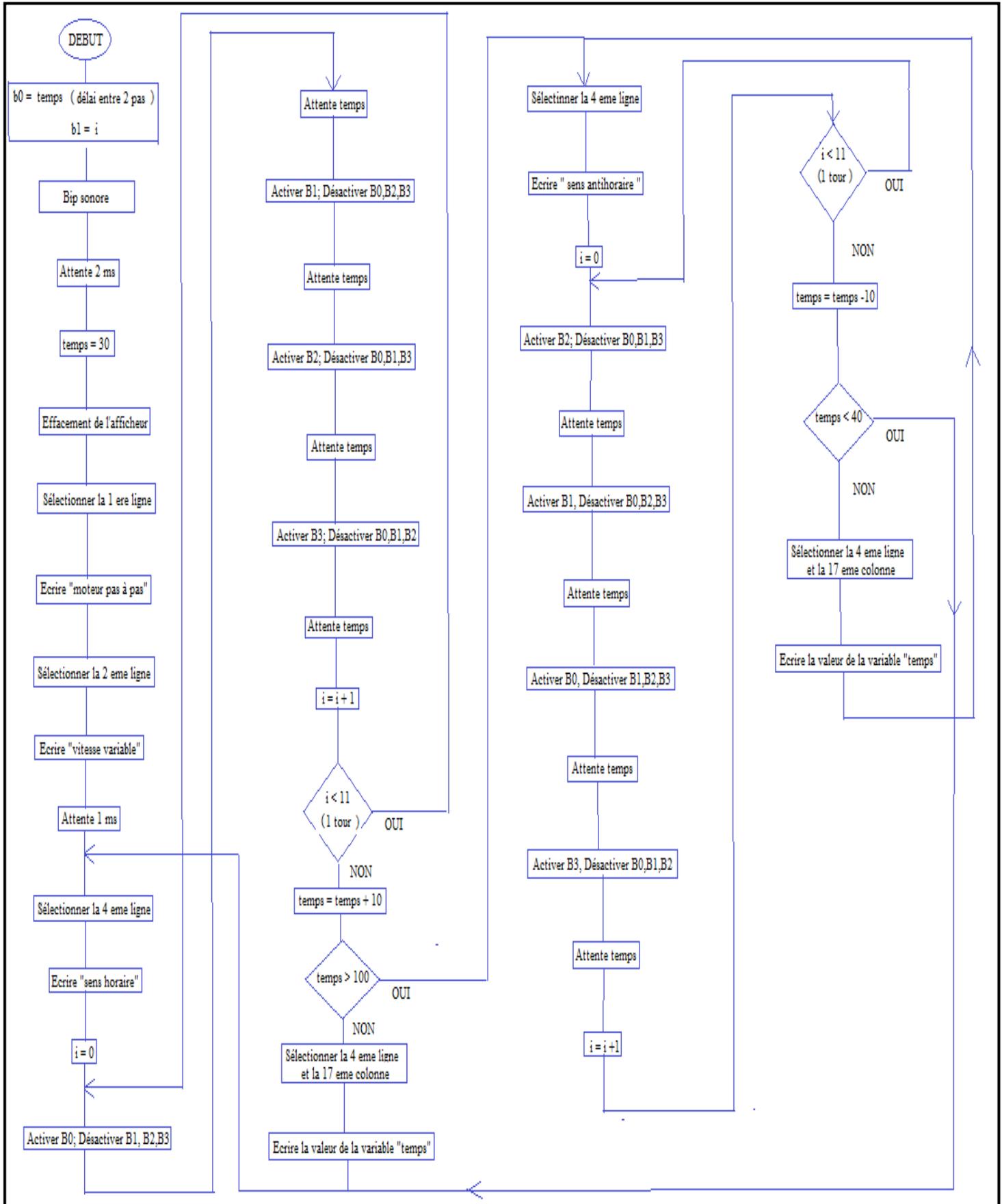


Figure 3. 39 : organigramme de commande de moteur pas à pas

3.4.2.2 Commande de deux moteurs à courant continu

La commande des deux moteurs étant déjà faite en Flowchart, on peut la refaire d'une autre manière en convertissant le programme Flowchart en Basic. On charge le programme obtenu dans le microcontrôleur.

Les moteurs tournent comme s'ils commanderaient le déplacement d'un mobile, qui se déplace suivant le cycle perpétuel : marche « avant » de 3s, arrêt de 1s, marche « arrière » de 3s, arrêt de 1s, virage à gauche de 3s, arrêt de 1s, virage à droite de 3s et enfin arrêt de 1s.

3.5 Conclusion

La stratégie utilisée dans la partie réalisation est basée sur la simplicité des montages et le moindre coût, pour assurer le bon fonctionnement de notre carte et avoir un système pratique et simple à utiliser. Pour cela, on a utilisé des connecteurs d'entrée sortie faciles à manipuler et nous avons ajouté une face avant pour notre automate, afin d'améliorer la précision et la visibilité à l'utilisateur.

Notre système est fonctionnel, différents tests ont été concluants grâce aux périphériques réalisés.

CONCLUSION GENERALE

La conception et la mise en œuvre d'un automate programmable sont des sujets très vastes. Il faut savoir que ce type d'appareil est employé dans l'industrie, dès qu'il s'agit d'automatiser un processus de fonctionnement (robotisation de chaîne de production, contrôle et validation d'une pièce, ascenseurs, escalier mécaniques, régulation climatique de locaux, etc..).

Nous avons donc réalisé un automate programmable basé sur le PICAXE 40X2 ayant les caractéristiques suivantes : sept entrées analogiques, huit entrées numériques, quatorze sorties numériques, un port I2C, un afficheur LCD de quatre lignes de vingt caractères et un buzzer.

Nous lui avons ensuite développés quelques périphériques classiques. Notre montage fonctionne correctement.

Notre projet nous a permis aussi de rentrer dans le vaste et passionnant domaine de l'informatique, c'est grâce à ce projet que nous avons appris d'un peu plus près le langage BASIC, et le logiciel de routage PROTEUS et de nous familiariser avec les nouveaux microcontrôleurs PICAXE et leur programmation.

Finalement, ce travail a été une occasion de concrétiser et de mettre en évidence nos compétences acquises au cours de notre formation académique avec notamment le parcours de thématique à savoir l'électronique, l'automatique, l'instrumentation et de consolider ainsi notre formation.

Liste des figures et tableaux

La liste des figures

<i>Figure 1. 1 : Automate Compact</i>	6
<i>Figure 1. 2 : Automate Modulaire</i>	6
<i>Figure 1. 3 : structure interne de l'automate programmable</i>	8
<i>Figure 1. 4 : fonctionnement de l'A.P.I</i>	9
<i>Figure 1. 5 : structure des systèmes automatisés</i>	10
<i>Figure 1.6 : schéma de brochage du microcontrôleur PICAXE 40X2</i>	13
<i>Figure 1. 7 : schéma de brochage du microcontrôleur PICAXE 20X2</i>	16
<i>Figure 1. 8 : Circuit de téléchargement PICAXE</i>	17
<i>Figure 2. 1 : bloc d'alimentation</i>	20
<i>Figure 2. 2 : circuit de servitude</i>	20
<i>Figure 2. 3 : Module d'interface de programmation</i>	21
<i>Figure 2. 4 : les sept entrées du convertisseur analogique/numérique</i>	22
<i>Figure 2. 5 : les huit entrées numériques</i>	22
<i>Figure 2. 6 : les quatorze sorties numériques</i>	23
<i>Figure 2. 7 : l'afficheur LCD et le buzzer</i>	24
<i>Figure 2. 8 : schéma détaillé d'un moteur à courant continu</i>	26
<i>Figure 2. 9 : circuit de commande de deux moteurs DC</i>	26
<i>Figure 2. 10 : schéma de brochage du circuit L293D</i>	27
<i>Figure 2. 11 : schéma simplifié d'un moteur pas à pas</i>	28
<i>Figure 2. 12 : circuit de commande d'un moteur pas à pas unipolaire</i>	29
<i>Figure 2. 13 : circuit de commande d'un servomoteur</i>	30
<i>Figure 2. 14 : circuit de commande d'un relais</i>	31
<i>Figure 2. 15 : la lecture de la valeur d'une résistance ajustable</i>	32
<i>Figure 2. 16 : schéma du teste d'une photorésistance</i>	33
<i>Figure 2. 17 : la lecture de la valeur d'une photorésistance</i>	33
<i>Figure 2. 18 : la lecture de l'état logique d'une touche</i>	34
<i>Figure 3. 1 : Bloc d'alimentation sur PROTEUS</i>	35
<i>Figure 3. 2 : visualisation de la tension de sortie</i>	36
<i>Figure 3. 3 : la simulation du circuit de la carte principale</i>	37
<i>Figure 3. 4 : vue de l'automate réalisé</i>	38
<i>Figure 3. 5 : schéma de câblage de la résistance ajustable sur PROTEUS</i>	39
<i>Figure 3. 6 : le circuit réalisé de la résistance ajustable</i>	39
<i>Figure 3. 7 : schéma de câblage de la photorésistance sur PROTEUS</i>	40
<i>Figure 3. 8 : le circuit réalisé de la photorésistance</i>	40
<i>Figure 3. 9 : le schéma de câblage de circuit de l'état logique d'une touche sur PROTEUS</i>	41
<i>Figure 3. 10 : le circuit réalisé de l'état logique d'une touche</i>	41
<i>Figure 3. 11 : schéma de câblage de la carte de puissance des deux moteurs DC sur PROTEUS</i>	42
<i>Figure 3. 12 : le circuit réalisé de la carte de puissance des deux moteurs DC</i>	42

<i>Figure 3. 13 : schéma de câblage de la carte de puissance du moteur pas à pas sur PROTEUS</i>	43
<i>Figure 3. 14 : le circuit réalisé de la carte de puissance du moteur pas à pas</i>	43
<i>Figure 3. 15 : le schéma de câblage du relais sur PROTEUS</i>	43
<i>Figure 3. 16 : le circuit réalisé de commande du relais</i>	44
<i>Figure 3. 17 : le schéma de câblage de circuit de commande du servomoteur sur PROTEUS</i>	44
<i>Figure 3. 18 : le circuit réalisé de commande du servomoteur</i>	44
<i>Figure 3. 19 : embase MJ4P4C femelle</i>	45
<i>Figure 3. 20 : connecteur MJ/4P4C mâle</i>	45
<i>Figure 3. 21 : câble utilisable</i>	45
<i>Figure 3. 22 : le schéma de brochage des embases MJ/4P4</i>	46
<i>Figure 3. 23 : le montage final de notre réalisation</i>	46
<i>Figure 3. 24 : vue du logiciel de programmation en Flowchart</i>	47
<i>Figure 3. 25 : vue du logiciel de programmation en Basic</i>	48
<i>Figure 3. 26 : câble AXE027</i>	49
<i>Figure 3. 27 : raccordement entre le périphérique de la touche et l'automate</i>	50
<i>Figure 3. 28 : organigramme expliquant le test d'une touche</i>	50
<i>Figure 3. 29 : raccordement entre le périphérique de la résistance ajustable et l'automate</i>	51
<i>Figure 3. 30 : organigramme la lecture de la valeur de la résistance ajustable</i>	52
<i>Figure 3. 31 : raccordement du relais et de la photorésistance avec l'automate</i>	53
<i>Figure 3. 32 : organigramme d'évaluation du niveau d'éclairage</i>	53
<i>Figure 3. 33 : raccordement du servomoteur et de la résistance ajustable avec l'automate</i>	54
<i>Figure 3. 34 : organigramme de commande du servomoteur</i>	55
<i>Figure 3. 35 : paramétrage de déplacement</i>	56
<i>Figure 3. 36 : raccordement des deux moteurs avec l'automate</i>	56
<i>Figure 3. 37 : organigramme de commande des deux moteurs</i>	57
<i>Figure 3. 38 : raccordement du moteur pas à pas avec l'automate</i>	58
<i>Figure 3. 39 : organigramme de commande de moteur pas à pas</i>	59

La liste des tableaux

<i>Tableau 1. 1 : La RAM du microcontrôleur 40X2</i>	14
<i>Tableau 1. 2 : La RAM du microcontrôleur 20X2</i>	16
<i>Tableau 2. 1 : caractéristique de l'automate programmable</i>	19
<i>Tableau 3. 1 : attribution des entrées et sorties du 40X2 dans la programmation en Basic</i>	48
<i>Tableau 3. 2 : attribution des entrées et sorties du 40X2 dans la programmation en Flowchart</i>	49

ملخص

العمل المقدم في المذكرة هو عبارة عن دراسة و إنشاء مبرمج ألي بواسطة المتحكمين الدقيقين (PICAXE 40X2 و 20X2) و الأجهزة التي يتحكم فيها, نظهر فعالية هذا المبرمج الألي بربطه ببعض أجهزة الاستشعار و المحركات من أجل استخدامها للتحكم في عملية الحكم الذاتي.

Résumé

Le travail présenté dans ce mémoire, est l'étude et la réalisation d'un automate programmable à base de deux microcontrôleurs PICAXE (40X2 et 20X2), ainsi que les périphériques commandés par cet automate, en lui raccordant quelques capteurs et actionneurs afin de le tester dans le but de l'utiliser pour commander un processus autonome.

Abstract

The work presented in this thesis is the study and implementation of a programmable automatic-based on two PICAXE (40X2 and 20X2), and devices controlled by this controller, by connecting him some sensors and actuators to the test in order to use it to control an autonomous process.

Références bibliographiques

- [1] Automates Nano et plate-forme d'automatisme Micro, Schneider Electric 1999.
- [2] Cours Génie Electrique >> Cours Automatisme et Informatique Industrielle >> Automates programmables industriels :
Chapitre 4 : Control Engineering
siemens.co.za/Control-Engineering
Siemens offers a spectrum of products for control engineering.
- [3] Picaxe_manual 1, www.picaxe.com
- [4] Site officiel du PICAXE, <http://www.picaxe.com/What-is-PICAXE/>
- [5] Cours d'automates programmable, université du Grenoble, http://iut-tice.ujf-grenoble.fr/tice-espaces/GEII/EP-od/wupload/File/ARS2_Cours/2_-_Automates_programmables.pdf
- [6] Alain GONZAGA, Les Automates Programmables Industriels, <http://ebookbrowse.com/les-automates-programmables-industriels-pour-geea-pdf-d47786499>
- [7] PICAXE X2 Product Briefing (updated Dec 10). <http://free-doc-lib.com/book/picaxe-x2-product-briefing-updated-dec-10.pdf>
- [8] (André Bernot, Thierry Lancelot, Denis Hoffschir, Dominique Sauzeau) Guide d'utilisation pour la programmation en mode graphique avec « Picaxe Programming Editor»
Édité par la Société a4 Technologie www.a4.fr
- [9] magazine électronique pratique du mois de Novembre 2012 : www.electroniquepratique.com
- [10] datasheet du circuit intégré 74HC541
http://datasheet.seekic.com/PdfFile/74H/74HC541_PHILIPS.pdf
- [11] Principe de fonctionnement d'une photorésistance, article de la société refrobot (entreprise de robotique depuis 2001). <http://johnest.unblog.fr/2012/09/16/photoresistances-introduction/>

Annexe A

Caractéristiques du microcontrôleur 40X2

Micro PIC de base (série PIC18F) : 45K22

Gamme de tensions (V) : 2,1-5,5

Gamme des versions du microprogramme PICAXE : B.3+

Pièce actuelle (en cours de production) : Oui

Fréquence interne max. (MHz) : 16

Fréquence externe max. (MHz) : 64

Prise en charge des capteurs tactiles : Oui

Configuration CAN (convertisseur analogique-numérique) : individuelle

Référence CAN interne (V) : 1 024

RAM des variables (octets) : 256

RAM du bloc-notes (octets) : 1 024

Slots de programme internes : 4

Slots de programme externes : 32

Broches d'interruptions matérielles : 3

Canaux Pwmout : 2

Prise en charge de hpwm : Oui

Mode Direction assistée dans hpwm : Oui

Tirages individuellement contrôlés : Oui

Modules SRLatch, FVR et DAC : Oui

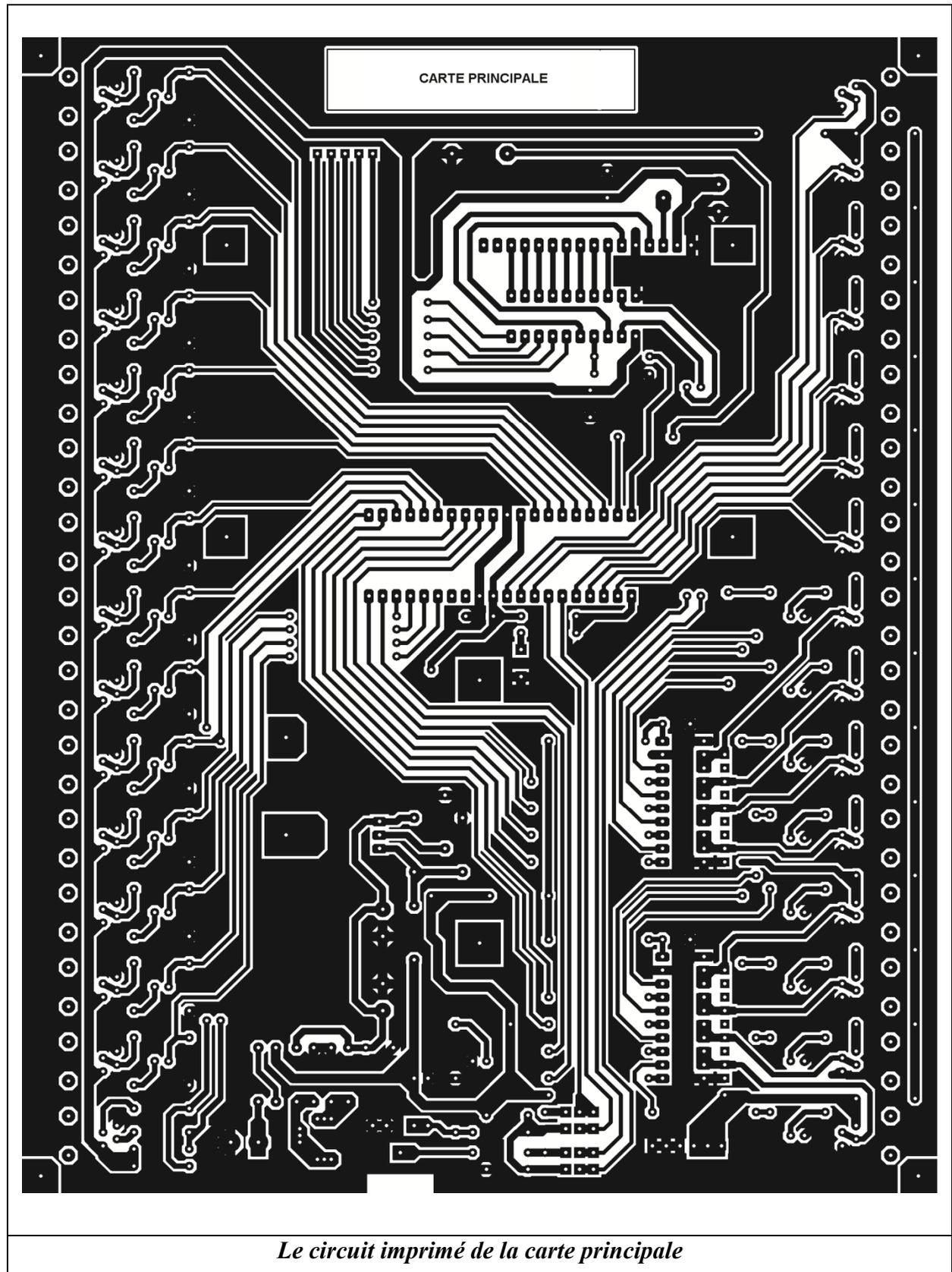
I2C

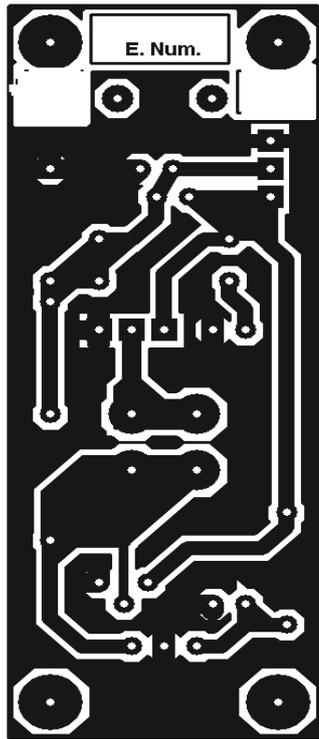
SPI

1 fil

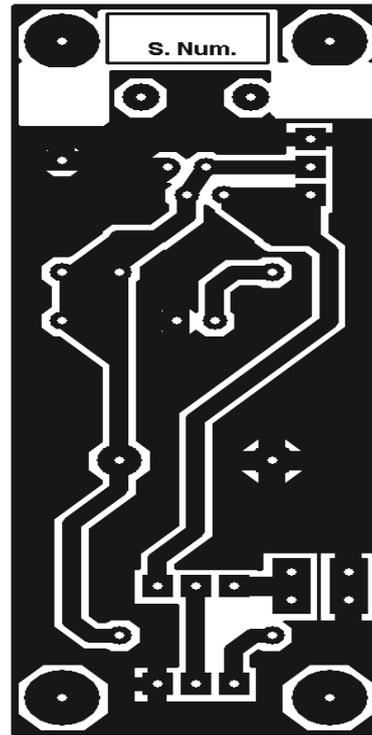
UNI/O

Annexe D

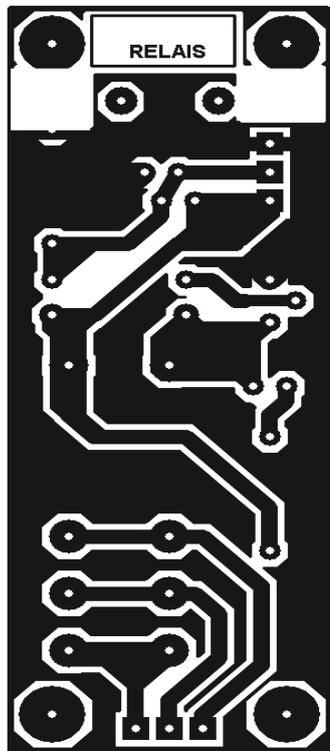




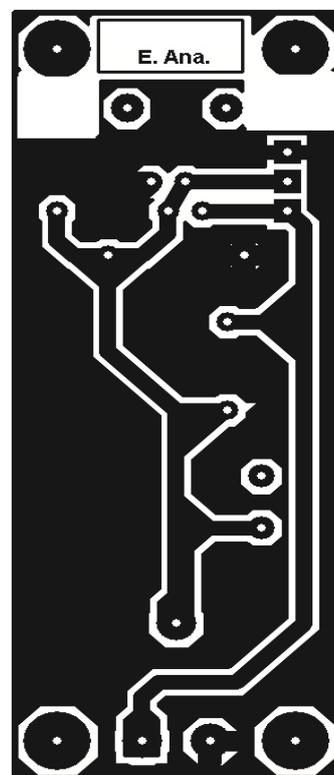
Le circuit imprimé de la touche



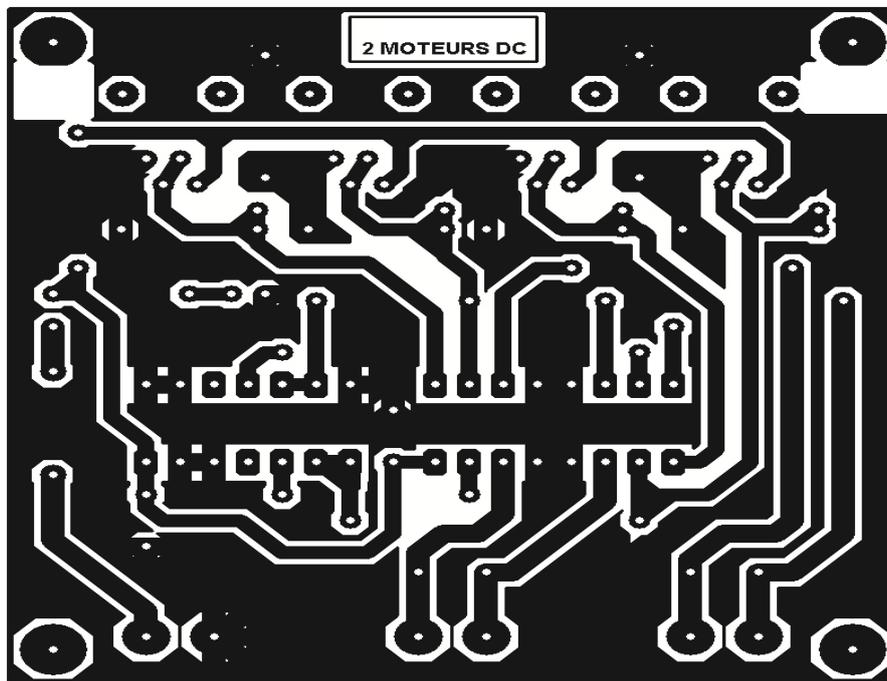
Circuit imprimé de commande du servomoteur



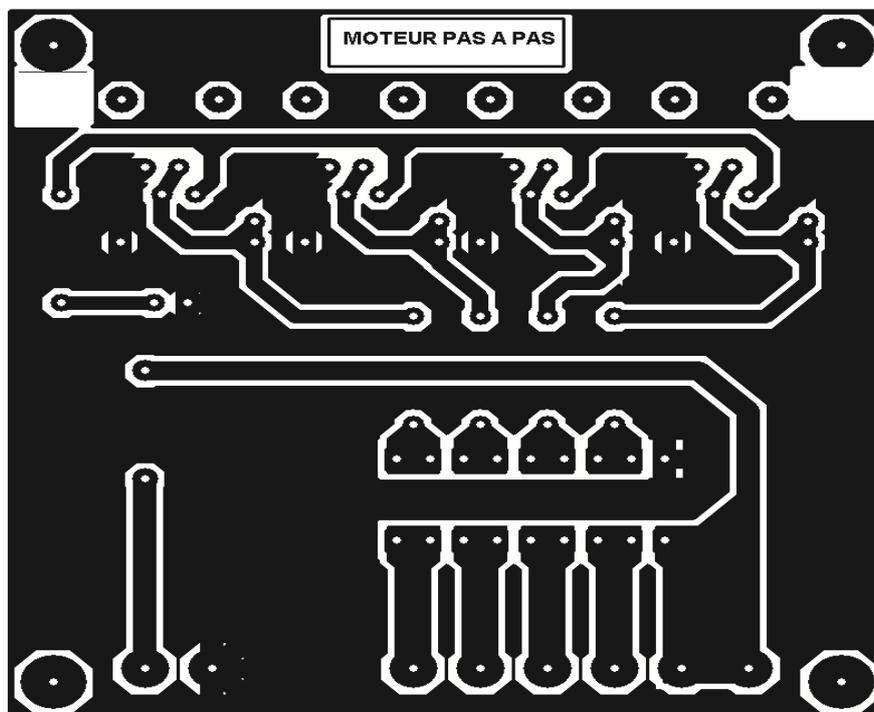
Le circuit imprimé de commande du relais



Le circuit imprimé de la résistance ajustable et de la photorésistance



Le circuit imprimé de la carte de puissance des deux moteurs DC



Le circuit imprimé de la carte de puissance du moteur pas à pas

Annexe B

Schéma de brochage du 40X2

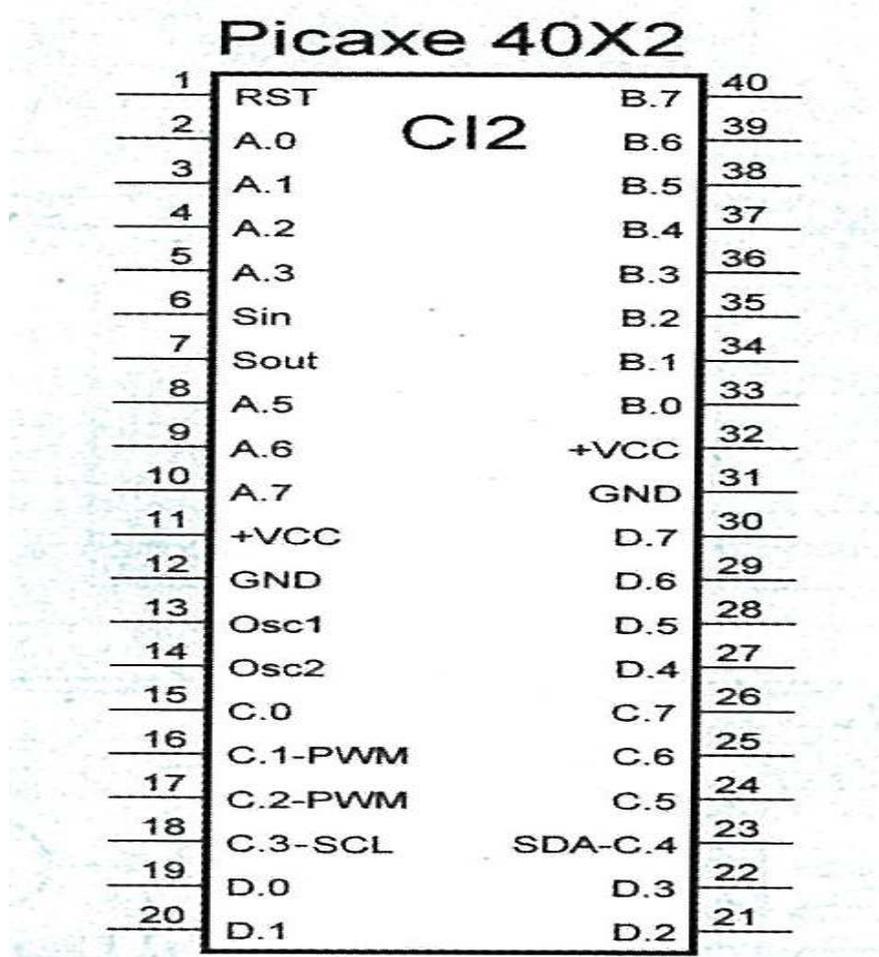
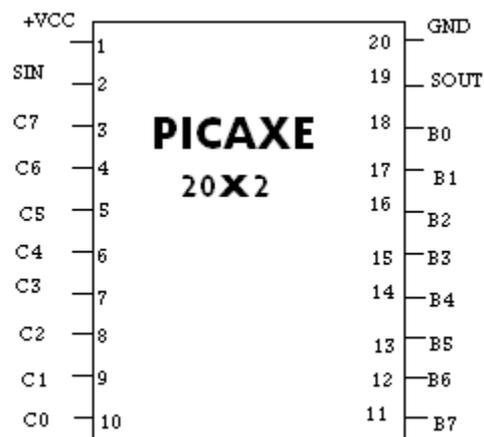


Schéma de brochage du 20X2



Annexe C

Nomenclature

Unité centrale

Résistances 5%

R1 à R39, R44, R45, R47 à R51 : 470 Ω (jaune, violet, marron)

R40, R41, R42, R46, R54: 10 K Ω (marron, noir, orange)

R43: 220 Ω (rouge, rouge, marron)

R52: 22 K Ω (rouge, rouge, orange)

R53: 180 Ω (marron, gris, marron)

RES1, RES2: 8x100 K Ω + 1 broche commune

AJ1 : 10 K Ω , horizontal, 1 tour

Semi-conducteurs

D1 à D10 : 1N4007

D11 : 1N4148

D12 : BAT85

CI1 : 7805

CI2 : PICAXE-40X2

CI3 : PICAXE-20X2

CI4, CI5 : 74HCT541

Led 1 à Led23 : Led verte \varnothing 5mm.

1 afficheur LCD de 4 X 20 caractères.

Condensateurs

C1, C2 : 1000 μ F / 25V (électrochimique à sorties radiales)

C3, C5 à C9 : 100 nF (mylar)

C4 : 22 μ F / 16 V (électrochimique à sorties radiales)

Divers

3 supports de circuits intégrés à 20 broches

1 support de circuits intégrés à 40 broches

2 dissipateurs thermiques pour TO220, type : ML26 (faible hauteur)

1 touche miniature (pour RST)

30 embases MJ/4P4C pour circuit imprimé

Plusieurs connecteurs MJ/4P4C à sertir

Fils rigides fins (pour les straps)

1 embase de programmation pour « PICAXE » (jack stéréo 3.5 pour circuit imprimé).

2 cavaliers de configuration

Barrettes sécables SIL droites et coudées, mâles et femelles

1 connecteur d'alimentation de \varnothing 2.1 mm

1 buzzer piézo de \varnothing 17 mm, sans oscillateur

1 bloc secteur 12V continu

Barrettes sécables femelles type « tulipe » (pour afficheur LCD)

Barrettes sécables mâles pour connecteur de type « tulipe » (pour afficheur LCD)

Barrettes sécables mâles de type « SIL »

Périphériques

Résistances 5%

R1, R22 : 330 Ω (orange, orange, marron)

R2, R3, R5 : 4.7 k Ω (jaune, violet, rouge)

R4, R11, R16 à R20, R24, R26 : 470 Ω (jaune, violet, marron)

R6, R23: 220 Ω (rouge, rouge, marron)

R7 à R10, R12 à R15, R21, R25: 10 k Ω (marron, noir, orange)

R27 : 100 k Ω (marron, noir, jaune)

AJ1 : 10 k Ω , horizontal, 1 tour

Semi-conducteurs

D1 à D3 : 1N4007

CI2 : DS18B20

CI3 : L293D (suffixe « **D** » indispensable)

CI4 : CD4093

CI5 : ULN2803

T1 : 2N2222

Led1 à Led5 : led verte \varnothing 5 mm

Condensateurs

C2, C3 : 100 nF (mylar)

C4, C5 : 47 nF (céramique ou mylar)

Divers

1 support de circuit intégré à 14 broches

1 support de circuit intégré à 16 broches

1 support de circuit intégré à 18 broches

1 touche type « D6 »

1 servomoteur simple de modélisme

2 moteurs pas à pas unipolaires

1 photorésistance

17 embases MJ/4P4C pour circuit imprimé

Plusieurs connecteurs MJ/4P4C à sertir

Nappe de 4 fils souples au pas de 1.27 mm, pour connecteurs MJ/4P4C à sertir

Fils rigides fins (pour les straps)

2 cavaliers de configuration

Barrettes sécables SIL droites et coudées, mâles et femelles

Buzzer piézo de Ø 17 mm

Barrettes sécables femelles type « tulipe »

Barrettes sécables mâles de type « SIL »

Introduction

Générale

Chapitre 1

Généralités

Chapitre 2

Etude Théorique

Chapitre 3

Réalisation et tests

Conclusion Générale

Annexe A

Annexe B

Annexe C

Annexe D

Bibliographie