



Département de Physique Appliquée
Master Sciences et Techniques Génie Electrique

Mémoire présenté à
LA FACULTE DES SCIENCES ET TECHNIQUES GUELIZ DE MARRAKECH
Pour obtenir le diplôme de
MASTER ES SCIENCES ET TECHNIQUES
« GENIE ELECTRIQUE »
Option : Automatique et Informatique Industrielle

Par
Sara BAJA
Sous le thème

Perfectionnement pédagogique
&
Réalisation robotique

Soutenu le 18 juillet 2016 devant le jury composé de :

Pr. : A.YOUS
Pr. : A .AOUZAL
Pr. : Abdeljalil EL KARI

Président
Examineur
Encadrant

Organisme d'accueil : L'université polytechnique Opole

N° D'Ordre : MGE2016/

Dédicace

Je dédie ce travail à

Mes parents, pour leur soutien, leur encouragement et pour leurs Sacrifices

qu'ils ont consentis pour m'éduquer, m'instruire et pour le bien être.

Ma famille en général.

Tous mes enseignants de la FSTG.

Tous mes enseignants de l'université Politechnika Opolska.

Tout le personnel qui m'a aidé.

Tous mes amis sans exception.

Veillez accepter mes meilleurs vœux de réussite, de bonheur et de prospérité.

Ce travail est aussi le vôtre.



Remerciements

*Avant d'entamer mon expérience au sein de l'université **Politechnika Opolska**, je tiens d'abord à exprimer ma gratitude et toute ma reconnaissance à Monsieur **Abdeljalil EL KARI** pour sa formation précieuse durant ces deux années, son encadrement et ses conseils avisés durant ces cinq mois, Et au nom de tous les étudiants je le remercie pour l'aide qu'il nous a apportée autant qu'un responsable du Master Génie Electrique.*

*J'adresse mes sincères remerciement à Monsieur **DOUBABI Said** mon enseignant, pour sa formation précieuse, ainsi pour m'avoir accordé sa confiance et me permettre de participer au programme ERASMUS+.*

Je remercie vivement tous mes formateurs qui ont tracé les bonnes conditions et le bon chemin d'apprentissage.

*Mes vifs remerciements à Monsieur **DAICH Said** professeur à l'université des droits à Opole pour l'intérêt qui a apporté afin que des étudiants marocains joignent Erasmus, ainsi pour sa modestie et son accueil.*

Je remercie vivement tous les membres du jury pour avoir accepté de juger mon travail.

*Enfin, je voudrais remercier chaleureusement tous l'organisme de l'université **Politechnika Opolska** pour leur formation, leur aide, leur accueil durant toute ma période en Opole.*

Abstract

The present report is the result of the work i realized during my studies in Poland.

As part of a new agreement between the Cadi Ayyad University and the Opole University of Technology, I did my Final Project within the Opole University of Technology.

The purpose of my Final Project is to develop my scientific knowledge in my field, by studying further modules. It's about renewable energy, Digital Signal Processors (DSP) ,Microprocessor MSP 430(Texas Instruments), AutoCAD ,Diagnosis of high voltage electrical equipment and eventually I wanted to transform my theoretical knowledge into practice by making a robot with two concepts; the first is Obstacle Avoiding Robot and the second is a robot controlled by Smartphone.

Keywords : *renewable energy, Digital Signal Processors (DSP) ,Microprocessor MSP 430(Texas Instruments), AutoCAD ,Diagnosis of high voltage electrical equipment ,Obstacle Avoiding Robot ,robot controlled by Smartphone*

Résumé

Dans le cadre d'un nouveau contrat entre l'Université Cadi Ayyad et l'université Polytechnique Opole, j'ai effectué mon PFE au sein de l'université Polytechnique Opole en Pologne.

L'objectif de mon Projet de Fin d'Etudes consiste à approfondir mes connaissances scientifiques dans mon domaine, en étudiant des modules complémentaires à ma formation. Il s'agit des énergies renouvelables, Digital Signal Processors (DSP), les Microprocesseurs (Texas Instruments), Autocad, diagnostic des équipements électriques de haute tension et finalement j'ai voulu transformer mes acquis théoriques en pratique en réalisant un Robot avec deux concepts ; le premier c'est robot détecteur d'obstacle et le deuxième est un robot contrôlé par Smartphone.

Mots clé : *les énergies renouvelables, Digital Signal Processors (DSP), les Microprocesseurs (Texas Instruments), Autocad, diagnostic des équipements électriques de haute tension , robot détecteur d'obstacle, robot contrôlé par Smartphone.*

Glossaire

DSP : Digital signal Processing.

2D : deux dimensions.

3D : trois dimensions.

ERASMUS : European Action Scheme for the Mobility of University Students.

MAD : dirham marocain.

MASEN : l'Agence Marocaine Pour l'Energie solaire.

CSP : Concentrated Solar Power.

MFLOPS : Million floating-point operations per second

DAO : logiciel de dessin assisté par ordinateur

AC : Tension alternatif.

BDV : Breakdown voltage.

Table des matières

Dédicace	I
Remerciements	II
Abstract.....	III
Résumé	IV
Glossaire	V
Table des matières	VI
Liste des figures.....	VIII
Liste des tableaux	IX
Introduction Générale	1
<i>Chapitre I : Erasmus + et l'Université Polytechnique Opole</i>	3
I. Introduction.....	3
II. Le Programme Erasmus +.....	3
III. Politechnika Opolska (Université de Technologie Opole)	5
IV. Conclusion	7
<i>Chapitre II : l'énergie solaire au Maroc</i>	8
I. Introduction.....	8
II. La Problématique d'électricité au Maroc.....	8
III. La nouvelle stratégie énergétique du Royaume (Energie solaire)	9
IV. L'Agence marocaine pour l'énergie solaire (MASEN).....	10
V. Le projet Noor (Ouarzazate Complexe solaire)	10
VI. Conclusion	12
<i>Chapitre III: Les microprocesseurs MSP 430</i>	13
I. Introduction.....	13
II. La carte MSP430	13
III. Code Composer Studio	16
IV. Les cahiers de charge réalisés	17
V. Conclusion	17
<i>Chapitre IV : Les Processeurs traitement du signal</i>	18
I. Les Processeurs traitement du signal (DSP)	18
II. La carte TMS320C6713 DSP	20

III.	Les programmes simulés sur la carte TMS320C6713	21
IV.	Conclusion	22
<i>Chapitre V : AutoCAD</i>		23
I.	Introduction.....	23
II.	AutoCAD	23
III.	La création d'un plan 2D d'une Villa :	26
IV.	Le plan de la villa en 3D.....	28
V.	Conclusion	28
<i>Chapitre VI : Diagnostic des équipements électriques</i>		29
I.	Introduction générale	29
II.	Transformateur de puissance	29
III.	Les tests de l'huile des transformateurs	30
IV.	Conclusion	34
<i>Chapitre VII : Robotique</i>		35
I.	Introduction.....	35
II.	Les composants du Robot :	35
III.	Robot détecteur d'obstacle.....	40
IV.	Robot contrôlé par Android	42
Conclusion générale.....		48
Annexe A		51
Annexe B		55
Digital Signal Processing (DSP).....		55
Les bibliothèques et les fonctions de DSK6713, codec audio, commutateurs et les LEDS.....		55
Les bibliothèques et les fonctions de DSK6713, codec audio, commutateurs et les LEDS.....		58
Les bibliothèques et les fonctions de DSK6713, codec audio, commutateurs et les LEDS.....		61
Annexe C		66

Liste des figures

Figure 1:Le programme ERASMUS plus	4
Figure 2:L'université Politechnika Opolska	6
Figure 3:: Consommation électrique en GWH	8
Figure 4: Le plan solaire Marocain	9
Figure 5 : Structuration institutionnelle du projet Noor	10
Figure 6:Les composants principales d'un microcontrôleur MSP430	13
Figure 7:L'architecture de la carte MSP430	15
Figure 8:chaîne complète typique d'un système de traitement numérique signal	18
Figure 9:Différence entre la structure « Von Neuman » et « Harvard »	19
Figure 10:la carte TMS320C6713.....	20
Figure 11: caractéristiques matérielles de la carte TMS320C6713 DSK	21
Figure 12 : Figure 12:Interface du logiciel AutoCAD	24
Figure 13 : Figure 13:la barre d'outils et de commande d'AutoCAD 2016	24
Figure 14:Le plan architecte d'une villa	26
Figure 15 : Figure 15:le plan 2D réalisé sur AutoCAD.....	27
Figure 16: le plan de la villa en 3D.....	28
Figure 17:le Figure 17:diagnostic d'huile du transformateur de sa couleur.....	31
Figure 18 :Figure 18: L'indice de neutralisation dans l'huile d'un transformateur	31
Figure 19 : Figure 19:les deux électrodes de la BDV KIT.....	32
Figure 20:Figure 20: La variation de la viscosité avec le changement de la température	33
Figure 21 Figure 21:: La variation de la densité avec le changement de la température	33
Figure 22:Châssis est les roues du robot	35
Figure 23:les composants de la carte Arduino Uno	36
Figure 24:Capteur à ultrasons HC-SR04.....	36
Figure 25: Le principe de fonctionnement du capteur à ultrasons HC-SR04	38
Figure 26:Moteur à courant continu	38
Figure 27:les composants du Moteur Shield L298N	39
Figure 28:le montage du Robot détecteur d'obstacles.....	42
Figure 29 : Figure 29:la fenêtre de la création de l'interface homme machine	43
Figure 30 : Figure 30:La fenêtre de programmation	43
Figure 31:La configuration du ListPicker	45
Figure 32:Contrôle du bluetooth.....	46
Figure 33:La configuration des boutons	46
Figure 34:Brochage du module Bluetooth et l'application Android	47

Liste des tableaux

Tableau 1:les commandes du logiciel AutoCAD	24
Tableau 2:les paramètres techniques de HC-SR04.....	37
Tableau 3: les caractéristiques du moteur DC.....	39

Introduction Générale

Effectuer un PFE à l'étranger pendant un semestre était une expérience intéressante, pour nous, afin d'enrichir nos connaissances. En effet nous avons étudié de nouveaux modules qui se sont avérés complémentaires pour notre formation d'une part et améliorer notre anglais d'autre part puisque tout les modules ont été dispensés dans cette langue.

Dans le cadre de la convention signée avec l'Université Politechnika Opolska , cinq modules et une réalisation nous ont été proposés :

➤ Modules :

- Les énergies renouvelables
- Digital Signal Processors (DSP)
- Microprocesseur MSP 430 (Texas instruments)
- Le logiciel Autocad
- Le diagnostic des équipements électriques de haute tension

➤ Réalisations :

- Robot détecteur d'obstacle
- robot commandé par Android

Dans un premier temps nous avons étudié le Microprocesseur MSP 430 de famille Texas Instrument et le DSP sous forme de cours et de travaux pratiques. Quant aux énergies renouvelables, diagnostic des équipements électriques de haute tension et le logiciel Autocad, ils ont été réalisés sous forme de recherches.

Dans un deuxième temps nous nous sommes intéressés à la commande de deux robots, le premier est un détecteur d'obstacle et le second commandé par un Smartphone.

Le mémoire présenté est organisé en trois chapitres.

Après une introduction générale, le chapitre I présente d'une manière générale le projet Erasmus + et l'Université Polytechnique Opole. Ensuite, nous passerons au deuxième chapitre consacré à une recherche bibliographique sur l'énergie solaire au Maroc. Dans le troisième, quatrième, cinquième et sixième chapitre, nous présentons, respectivement, les

cours qu'il nous ont dispensé à l'université Polytechnique d' Opole à savoir les Microprocesseurs (Texas Instruments), les DSP, Autocad et le diagnostique des équipements électriques de haute tension.

Quant au dernier chapitre, nous allons y présenter les applications réalisées sur un robot détecteur d'obstacle et sur un autre contrôlé par Smartphone. Le mémoire se termine par une conclusion générale.

Chapitre I : Erasmus + et l'Université Polytechnique Opole

I. Introduction

Cette partie est consacrée pour la présentation du programme Erasmus+ autant qu'une organisation crée afin de faciliter la mobilité des étudiants européens et les étudiants d'autres pays qui veulent étudier dans d'autres pays (de l'espace Schengen) , ainsi que l'Université de Technologie Opole (Politechnika Opolska) autant qu'un organisme d'accueil.

II. Le Programme Erasmus +

1. Présentation du programme

Erasmus (European Action Scheme for the Mobility of University Students), le nom du programme vient du moine humaniste et théologien néerlandais Érasme (1469-1536). Erasmus est un programme européen crée dont le but d'échanger les étudiants entre les universités et les grandes écoles européennes. Ce programme fait partie de l'Espace européen de l'enseignement supérieur. C'est un sous-ensemble du programme Éducation et Formation tout au long de la vie [1].

2. L'Histoire d'ERASMUS

Le programme Erasmus a été adopté en 1987 avec un budget de 85 millions d'écus pour la période 1987-1989. Avec Erasmus, les étudiants peuvent effectuer une partie de leurs études dans un autre établissement scolaire européen, pendant trois mois au minimum ou un an au maximum. De sa création jusqu'en 2013, ce programme a permis à 3 millions d'étudiants de participer à des échanges universitaires entre pays européens partenaires. Cet échange est très valorisé par les universités en tant qu'il permet de découvrir de nouveaux horizons [1].

Le programme Erasmus est aujourd'hui ouvert Aux :

- 28 états membres de l'Union européenne :

Allemagne, Autriche, Belgique, Bulgarie, Chypre, Croatie, Danemark, Espagne, Estonie, Finlande, France, Grèce, Hongrie, Irlande, Italie, Lettonie, Lituanie, Luxembourg, Malte,

Pays-Bas, Pologne, Portugal, République tchèque, Roumanie, Royaume Uni, Slovaquie, Slovénie et Suède.

La figure I.1 présente les pays participants à ERASMUS+ :



Figure I.1:Le programme ERASMUS plus

- Deux États en procédure d'adhésion à l'Union européenne :

Turquie, Macédoine (depuis 2014).

De 2003 à 2004 le nombre d'étudiants s'est élevé de 9%, et en 2005 145 000 étudiants ont participé au programme, ce qui représente 1% de la population européenne de ce groupe social. La subvention est d'une centaine d'euros par mois.

Une extension du programme Erasmus appelée Erasmus mundus ouverte à tous les pays du monde a été mise en œuvre à partir de la rentrée universitaire 2004/2005.

3. Le budget d'ERASMUS

Le budget européen pour l'année 2007 à 2013 est de 3,1 milliards d'euros. La Commission européenne a proposé en fin de 2011 une augmentation budgétaire d'environ 64 % par rapport au budget des années précédentes sur sept ans, soit une affectation de 19 milliards d'euros au nouveau programme pour la période 2014-2020. Finalement, le budget est 1,1 milliards d'euros par an mais en 2012 la cessation de paiements se profile, les états contributeurs n'ayant versé que 900 millions¹⁰ Finalement, une augmentation de 40 % est actée soit 14,7 milliards d'euros pour le nouveau programme, baptisé Erasmus+¹¹ (ce qui représente environ 1,5 % du budget global de l'union européenne¹²).

4. Les avantages d'Erasmus+

Le programme Erasmus permet d'améliorer et d'augmenter la mobilité étudiante et enseignante, ainsi que la transparence et la compatibilité des qualifications dans l'enseignement supérieur et la formation professionnelle supérieure en Europe. Les avantages principaux de ce programme sont l'exonération des droits de scolarité de l'université d'accueil, la reconnaissance formelle de la partie des études effectuées à l'étranger, ainsi que le maintien des bourses, prêts et couverture sociale du pays de l'université expéditrice. Afin de valider sa période d'étude à l'étranger, l'étudiant doit choisir un programme d'étude qui fait partie intégrante du programme qu'il suit dans son université d'origine. De plus, l'étudiant concerné doit avoir achevé sa première année d'étude universitaire. Enfin, un contrat d'étude est signé avant le départ de l'étudiant qui fixe la liste des matières qu'il devra suivre, ainsi que le nombre d'ECTS correspondant à sa période de mobilité. Afin de valider sa période d'étude à l'étranger, l'étudiant devra alors obtenir l'ensemble des ECTS prévus dans son contrat d'étude.

III. Politechnika Opolska (Université de Technologie Opole)

1. Informations générales sur l'université

Université de Technologie Opole (voir la figure 2) est un leader universitaire high-tech orientée vers l'international en Pologne. L'université fournit l'éducation principalement dans le domaine de la science technique, ainsi que dans la gestion, le tourisme, le sport et la physiothérapie. Son potentiel scientifique et pédagogique est construit par un personnel universitaire et une équipe de recherche de grande classe. Grâce à une collaboration avec les meilleures compagnies locales et des partenariats étrangers précieux, les étudiants acquissent une connaissance et des compétences pratiques adaptées aux exigences des normes actuelles du marché mondial. La recherche dépasse les domaines traditionnels de l'ingénierie, en combinant les sciences techniques spécialisées avec l'économie, l'administration, le sport et les sciences médicales [2].

L'université est composée de six facultés:

- Génie civil et architecture
- Economie et Management

- L'ingénierie électronique, Automatique et IT
- Ingénierie mécanique
- Ingénierie Production et Logistique
- Education et physiothérapie physique

L'Université de technologie Opole a été créée en 1966 et elle a 50 ans de tradition académique. Avec près de 8 000 étudiants, y compris jusqu'à 400 étudiants étrangers par an. L'Université emploie plus de 900 personnels, y compris 540 enseignants universitaires, 143 professeurs et plus de 20 chercheurs étrangers [2].



Figure I. 2 : L'université Politechnika Opolska

2. Location

L'Université est située dans la belle ville Opole, dans la partie sud-ouest de la Pologne, Europe.

La ville est la capitale de la région d'Opolskie (11 comtés, zone 9,412.5 km³, population: 1.044.346, urbain - 55%, 45% en milieu rural, la densité 110 / km²). Environ 15% des citoyens de la région sont des Allemands de souche.

Comme résultat de nombreuses zones sont officiellement bilingues. Tandis que la langue et la culture allemande jouent un rôle important dans l'éducation dans la région. Opole est une ville européenne de taille moyenne moderne, avec la population de 130 000. Opole est le plus grand centre économique, universitaire, scientifique et culturel de la région [3], [4].

3. Offre pour les étudiants étrangers

L'université offre des études dans un environnement international intéressant, stimulant et inspirant. Une fois inscrits les étudiants deviennent membres de la communauté multiculturelle.

Les programmes et les cours en anglais sont cruciaux pour l'internationalisation de l'Université et donc donné une attention particulière. Les programmes sont dirigés par l'équipe internationale hautement dédiée d'experts et de professionnels universitaires. Les cours sont conçus pour répondre aux futures exigences du marché mondial. Les professeurs bien établis et de renommée internationale, des spécialistes et des conférenciers supérieurs se concentrent en particulier sur l'applicabilité pratique et présenter des solutions concrètes à des situations du monde réel. Les programmes sont exécutés en étroite coopération avec les entreprises internationales.

4. Les Cours pour les étudiants d'ERASMUS

300 cours / sujets en anglais pour les élèves de ERASMUS + . La liste complète est publiée dans le catalogue ECTS disponibles sur la page web Bureau international. Le programme Erasmus offre également des opportunités de stages dans les laboratoires.

IV. Conclusion

ERASMUS (European Action Scheme for the Mobility of University Students) est un programme qui a pour but d'échanger les étudiants entre les pays de l'espace Schengen, dont le but est de dispenser plusieurs modules (normalement ces modules doivent être étudiés dans leurs universités). Le chapitre ci-dessus présente aussi l'Université Polytechnique Opole, le nombre d'étudiants, les enseignants, sa location, les offres qui présente pour les étudiants étrangers, ainsi que les cours offerts aux étudiants du programme ERASMUS.

Chapitre II : l'énergie solaire au Maroc

I. Introduction

Dans cette partie nous allons voir le problème d'électricité au Maroc, et puis présenter l'énergie solaire comme étant la meilleure solution du royaume afin de réduire le coût de ses importations de l'énergie. En plus des grands avantages qu'il devrait tirer de l'exploitation des énergies propres et renouvelables.

II. La Problématique d'électricité au Maroc

Le Maroc importe 95% de sa consommation en énergie, particulièrement le pétrole. En 2011-2013 les principaux exportateurs de pétrole brut vers le Maroc étaient l'Arabie Saoudite, l'Irak et la Russie. La grande majorité du gaz naturel est importé d'Algérie, alors que seulement 7% provient de la production locale. Dans le secteur de l'électricité, les importations en provenance d'Espagne ont fortement augmenté au cours de la dernière décennie, et couvert 2,2% de l'offre marocaine d'énergie primaire en 2012.

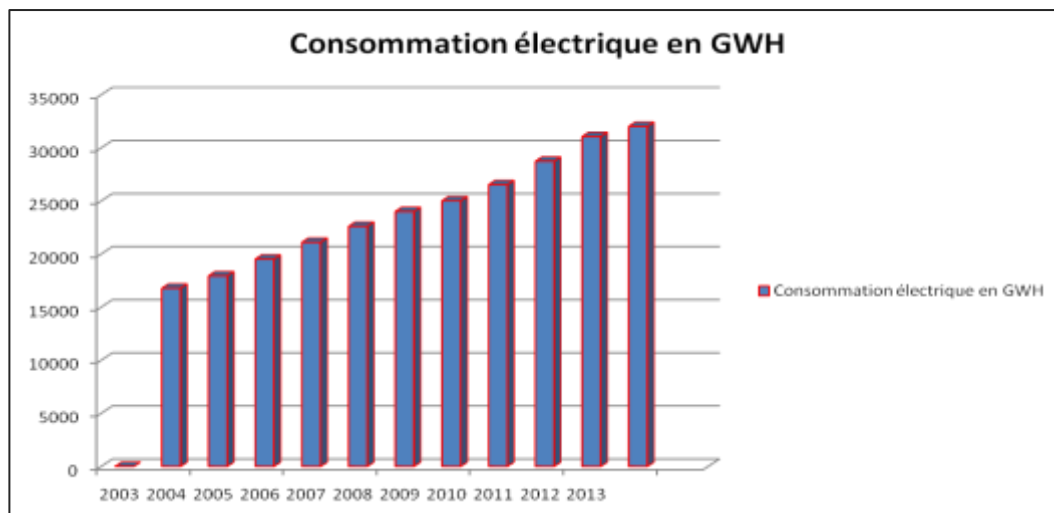


Figure II.1: Consommation électrique en GWH

Vu que la demande de l'électricité augmente de plus en plus (voir la figure II.1), les importations d'énergie (pétrole brut et produits pétroliers, charbon, gaz naturel et électricité) se sont élevées à 102,5 billion MAD en 2013. Cela représente une légère diminution par rapport à 2012 (106,6 billion MAD), mais une très forte augmentation par rapport à 2002 (19,1 billion MAD). Tenant compte des exportations de l'énergie d'une valeur de 9,26 billion MAD en 2013, la "facture énergétique" pour les importations nettes se sont élevées à 93,2

billion MAD en 2012. À moyen et à long terme, le Maroc espère être en mesure de produire plus d'électricité à partir des énergies renouvelables [5].

III. La nouvelle stratégie énergétique du Royaume (Energie solaire)

L'énergie solaire est une composante majeure de la nouvelle stratégie énergétique du royaume, puisque Le Maroc bénéficie d'une capacité considérable de ressources solaires, plus de 3000 heures / an de soleil. Capitalisant sur ce patrimoine, le Maroc a lancé en Novembre 2009, un programme intégré et proactif de la production d'électricité à partir de l'énergie solaire. Ce programme vise à:

- Réduire la dépendance d'énergie primaire en diminuant les importations.
- Encourager le déploiement d'une industrie locale de l'énergie renouvelable du secteur.
- Assurer la recherche, le développement et la formation liée à ces activités.
- mettre en place le Maroc comme le premier développeur d'Afrique du nord dans la production solaire à grande échelle.
- Réduire les émissions de carbone.

Le plan solaire Marocain, qui sera l'un des plus grands projets d'énergie solaire au monde, a été introduit en Novembre 2009 dans le but d'établir 2000MW d'énergie solaire d'ici 2020 (le plan solaire). Cinq sites ont été sélectionnés pour le développement de centrales solaires, avec la 500MW phase un complexe d'énergie solaire à Ouarzazate étant le premier à être mis au point (Programme Ouarzazate) [6].

Cette figure détaille le plan solaire du royaume, ainsi que le but à atteindre d'ici 2020.

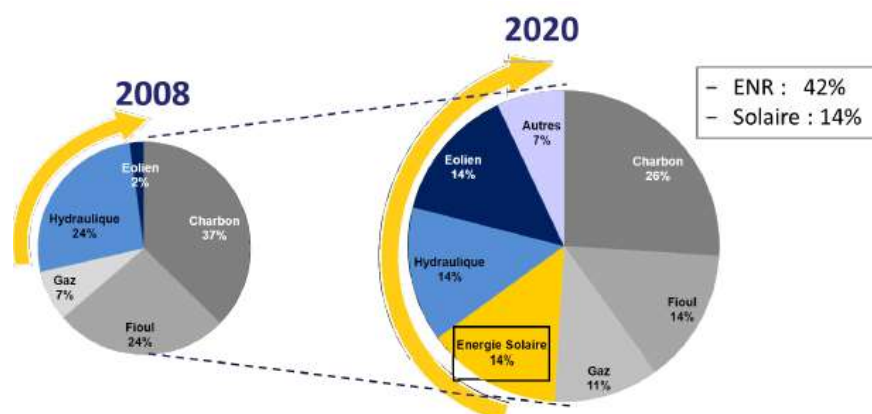


Figure II.2: Le plan solaire Marocain

IV. L'Agence marocaine pour l'énergie solaire (MASEN)

Dans le cadre du loi marocaine, l'Agence marocaine pour l'énergie solaire (MASEN) est responsable de la mise en œuvre du Plan solaire et a le rôle principal de l'organisation des appels d'offres pour les installations des chacun des cinq sites. Sur d'autres projets d'énergie indépendants (IPP) au Maroc, Office National de l'Electricité (ONE) a été chargé d'organiser les appels d'offres. ONE a une participation de 25% dans MASEN .Procédé pour la pré-qualification des entreprises et des Consortia émis par MASEN le 29 Juillet 2010 (mis à jour ultérieurement) (l'invitation pour pré-qualification), indique que le Programme Ouarzazate sera structuré comme un IPP avec un accord d'achat d'électricité de 25 ans (PPA) entre la société de projet et MASEN. En vertu de la structure à double PPA proposé dans l'invitation de pré-qualification, MASEN entrerait alors dans un PPA avec ONE [8] .

Cette électricité sera produite par une société de projet (SPC) détenue par le consortium développeur de la centrale (75%), avec une participation de Masen Capital, la filiale financière de l'agence solaire (25%). La SPC sera en charge de la conception, la construction, le financement, l'exploitation et la maintenance de la centrale (BOOT). [9]

La figure ci-dessous illustre la structuration institutionnelle du projet Noor.

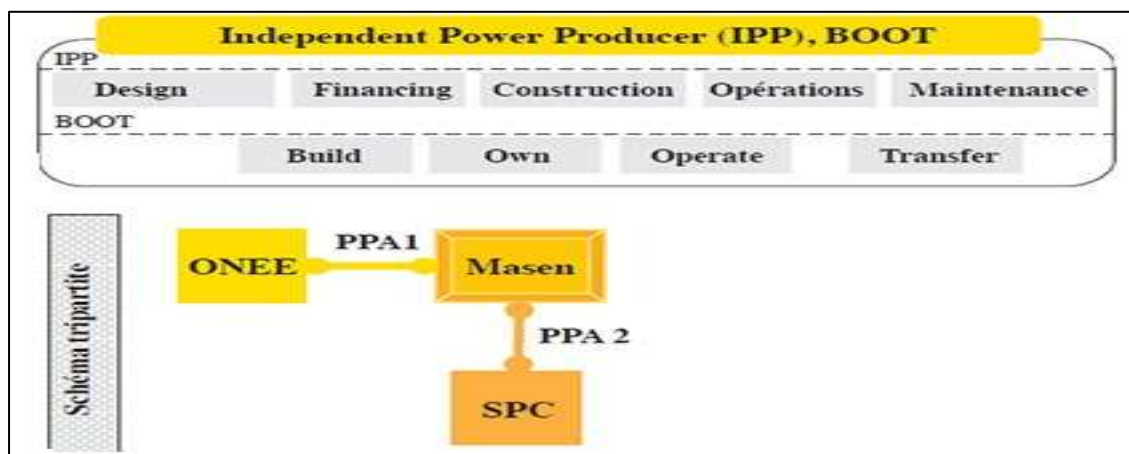


Figure II.3 : Structuration institutionnelle du projet Noor

V. Le projet Noor (Ouarzazate Complexe solaire)

1. La présentation du projet

Le complexe solaire Noor Ouarzazate est développé à 10km au nord-est de la ville de Ouarzazate. La première phase du projet, la construction d'un 160MW concentrated solar

power (CSP) plante nommée Noor I, tandis que la deuxième phase comprend la construction de l'usine Noor II (200MW) CSP et l'usine Noor III (150MW)CSP, et la troisième phase comprendra la construction de l'usine Noor IV .

La première phase des travaux de construction ont commencé en Août 2013 et Noor I a été officiellement mis en service en Février 2016. Noor II et Noor III devraient commencer les opérations commerciales en 2017 et 2018, respectivement [7], [9].

2. La Superficie de Noor

Noor 1, s'étale sur 480 hectares. Mais à terme, c'est-à-dire lorsque les trois autres parties du projet seront terminés, la centrale devrait s'étendre sur 3 093 hectares. Ce qui est équivalent à 3 437 terrains de football, ou bien la superficie de la ville de Rabat. Ce qui devrait en faire, à l'heure actuelle, la plus grande centrale solaire au monde.

3. La capacité de Noor

Cette plantation de capteurs solaire qui comprendra plus de 20 000 collecteurs, qui suivront le mouvement du soleil tout au long de la journée. L'énergie produite est stockée pendant 3 heures, et permet d'alimenter les foyers en électricité même après le coucher du soleil .

MASEN estime que après la construction des deux autre projets , Noor avec une capacité de 580 mégawatts devrait alimenter un million de foyers marocains .

4. Le cout d'investissement

Au total, budget d'investissement pour la création de Noor devrait être de 9 milliards de dollars. Une somme répartie entre plusieurs investisseurs, dont la banque africaine de développement, la banque mondiale, la banque européenne et plusieurs partenaires privés.

5. Les bénéfices du plan solaire Marocain

Grâce à la mise en valeur du potentiel solaire, le projet contribuera à la réduction de la dépendance énergétique, à la préservation de l'environnement, par la limitation des émissions des gaz à effet de serre et à la lutte contre les changements climatiques.

Noor-1 permettra d'éviter le rejet de 240 000 tonnes de CO₂ par an, selon la Banque africaine de développement. Soit 6 millions de tonnes de CO₂ évitées en 25 ans

d'exploitation, précise l'organisation panafricaine. Et lorsque Noor-2 et Noor-3 entreront en fonction, ce seront au total 19 millions de tonnes de CO2 évitées en 25 ans.

Un objectif qui devrait également permettre au Maroc de respecter les promesses faites à Paris lors de la COP 21, à savoir réduire de 32 % ses émissions de gaz à effet de serre d'ici 2030 [7], [9].

VI. Conclusion

Le parc solaire d'Ouarzazate est la vitrine d'un ambitieux projet destiné à faire en sorte que 52% des besoins en énergie du royaume proviennent des énergies renouvelables à l'horizon 2030, Ce qui va diminuer les dépenses des importations d'énergie mensuelles du Maroc .Le plan solaire permet au royaume d'exploiter ses ressources des énergies propres ,ainsi que sa participation dans la bataille mondiale contre le changement climatique et le réchauffement de la planète .

Chapitre III: Les microprocesseurs MSP 430

I. Introduction

Le troisième module que nous avons étudié, A savoir Les microprocesseurs de la famille Texas Instruments MSP430. Dans ce chapitre on va présenter la carte MSP430, sa configuration, ses registres, ses Timers, les différents types de communication, ainsi que quelques programmes que nous avons simulé sur Software Code Composer Studio.

II. La carte MSP430

1. Présentation de la MSP430

La famille Texas Instruments MSP430 des microcontrôleurs ultra-low-power est constituée de plusieurs dispositifs comportant différents ensembles de périphériques ciblés pour diverses applications. L'architecture, combine avec cinq modes de faible puissance, est optimisée pour atteindre une autonomie prolongée dans les applications de mesure portables. L'appareil dispose d'un puissant 16-bit RISC CPU, registres de 16 bits, et des générateurs constants qui contribuent à l'efficacité maximale du code.

L'oscillateur à commande numérique (DCO) permet le Wake-up des modes de faible puissance en mode actif en moins de 1 μ s [10].

La figure ci-dessous présente la carte MSP430, ainsi que ses composants [17].

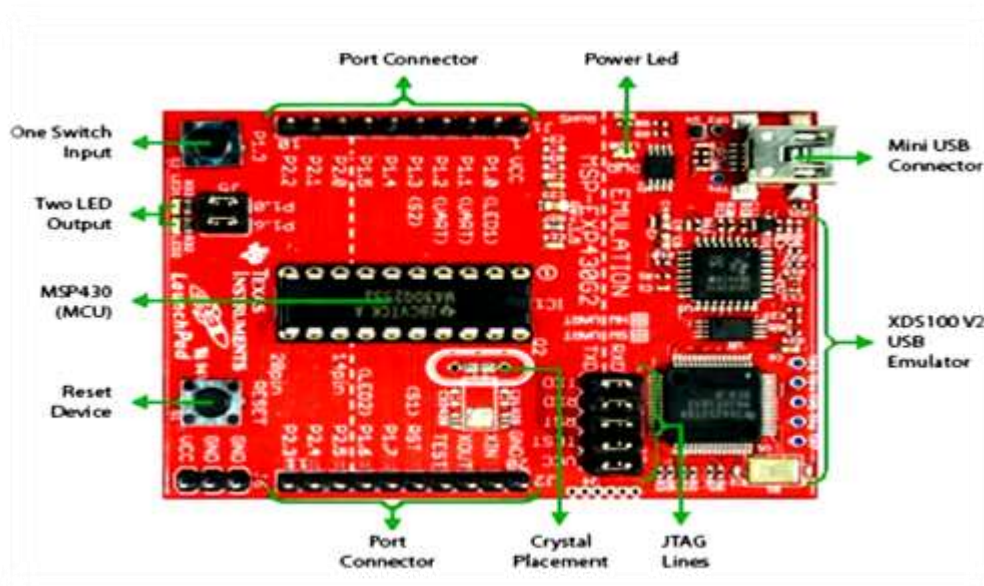


Figure III.1: Les composants principales d'un microcontrôleur MSP430

2. Les caractéristiques principales d'un microcontrôleur MSP430 :

Bien qu'il existe des variantes dans les appareils de la famille, un MSP430 microcontrôleur peut être caractérisé par:

- Faible consommation d'énergie:
 - 0,1 μ A pour la RAM data rétention.
 - 0,8 μ A pour un fonctionnement en mode d'horloge en temps réel.
 - 250 μ A / MIPS à fonctionnement actif.
 - Basse tension de fonctionnement (de 1,8 V à 3,6 V).
 - <1 μ s horloge de démarrage.
 - On-chip analogue devices:
 - 12/10 / 16-bit convertisseur analogique-numérique (ADC).
 - 12-bit dual convertisseur numérique-analogique (CNA).
 - Comparator-gated timers.
 - Amplificateurs opérationnels (OP Amps) .
 - Superviseur de tension d'alimentation (SVS).
 - 16 bit RISC CPU:
 - Instructions de traitement soit sur bits, octets ou des mots.
 - Conception de base compacte réduit la consommation d'énergie et de coût.
 - 27 instructions de base.
 - 7 modes d'adressage.
 - Flexibilité:
 - Jusqu'à 256 kB In-System Programmable (ISP) Flash.
 - Jusqu'à 100 pins options.
 - USART, I2C, Timers.
 - pilote de LCD.
 - émulation intégrée.
- L'architecture de la carte MSP430 est donné dans la figure III.2.

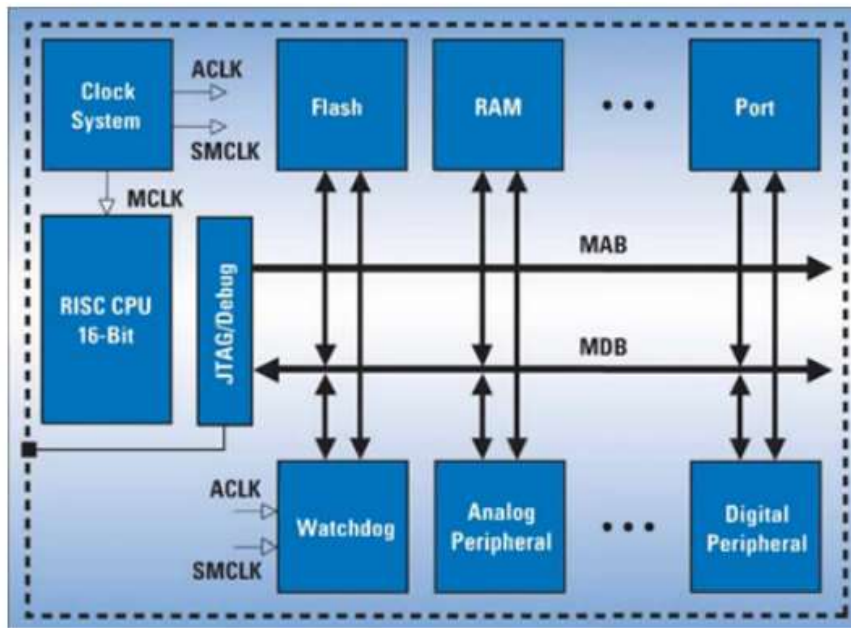


Figure III.2 : L'architecture de la carte MSP430

3. Les registres

Indépendamment du type de port I/O (non-interruptible: P3 et autres) ou interruptible (P1 et P2), la configuration de fonctionnement du port est défini par logiciel en utilisant les registres suivants:

➤ **Registre de direction (*PxDIR*)**

Lire / écrire des registres 8 bits

Configuration de *PxDIR*:

Bit = 1: La broche du port est configurée comme une sortie.

Bit = 0: la broche de port est configurée comme une entrée.

➤ **Registres d'entrée (*PxIN*)**

Configuration de *PxIN*:

Bit = 1: L'entrée est sur 1.

Bit = 0: L'entrée est sur 0.

➤ **Pull-Up/Pull-down resistances (*PxREN*):**

Configuration *PxREN*:

Bit = 1: pull-up/pull-down activée.

Bit = 0: pull-up/pull-down désactivée.



➤ **Registres de sortie (PxOUT)**

Les registres de sortie sont en read /write. Chaque bit de ces registres reflète la valeur écrite à la broche de sortie correspondante.

Configuration de PxOUT:

Bit = 1: La sortie est sur le niveau haut.

Bit = 0: La sortie est le niveau bas.

4. Les ports interruptibles (P1 et P2)

Chaque broche de ports P1 et P2 est capable de générer une demande d'interruption Elle est configuré en utilisant PxIFG, PXIE et PxIES registres. Les ports utilisent tous la même configuration mais avec trois registres complémentaires:

- Activer les interruptions PXIE)
 - Bit = 1: L'interruption est activée.
 - Bit = 0: L'interruption est désactivée.
- Sélectionner le front d'interruption (PxIES)
 - Bit = 1 : Interrupt flag est situé sur un front descendant.
 - Bit = 0: Interrupt flag est situé sur un front montant.
- Interruption Flag Registres (PxIFG) :
 - Bit = 0: Pas d'interruption en attente.
 - Bit = 1: Une interruption en attente.

5. LES TIMERS

MSP430 contient plusieurs timers, chacun d'eux est adapté à une situation. Les timers sont utilisés pour :

- Générer fixed-period time events;
- Autoriser un wake-up périodique du mode sleep;
- Compter les fronts des signaux ;
- remplacer delay loops avec un appel à un timer pour permettre à la CPU de dormir, ainsi que la consommation d'énergie va se diminuer.

III. Code Composer Studio

Code Composer Studio est un environnement de développement intégré (IDE) qui prend en charge les Microcontrôleurs et les processeurs embarqués de TI (Texas Instruments). Code Composer Studio est composé d'une suite d'outils utilisés pour développer et déboguer

des applications intégrés. Il comprend une optimisation C / C ++ compilateur, éditeur de code source, l'environnement de construction du projet, debugger, profiler, et de nombreuses autres fonctionnalités. L'IDE fournit une interface utilisateur unique, vous emmène à travers chaque étape du flux de développement d'applications. Des outils familiers et des interfaces permettent aux utilisateurs de démarrer plus rapidement que jamais. Code Composer Studio combine les avantages de la plate-forme logicielle Eclipse avec des capacités avancées de débogage intégrées de TI résultant dans un environnement de développement convaincant riche en fonctionnalités pour les développeurs embarqués [11].

IV. Les cahiers de charge réalisés

Cahier de charge N°1 : Rédiger un programme qui permet de faire clignoter la led P1.0 si il y a une interruption extérieure sur le bouton P1.3 sinon la led P1 .6 clignote.

Cahier de charge N°2 : Clignoter led en jouant sur pmw, la led devient de plus en plus lumineuse.

Cahier de charge N°3: Rédaction d'un programme qui assure par une communication UART de faire clignoter P1.0 si UCA0RXBUF reçoit la lettre « c » , P1.0 s'arrête de clignoter si UCA0RXBUF reçoit « de »,la led devient plus lumineuse si UCA0RXBUF recoit « f »,la led devient moins lumineuse si UCA0RXBUF reçoit « g »,et finalement pour la lettre « h » et la led clignote afficher *"sara_blinking_%d"* avec %d le taux de clignotement, si la led est éteinte afficher *"sara_notblinking_%d"*

Pour les programmes détaillés (voir Annexe A).

V. Conclusion

Dans ce chapitre nous avons vu le microprocesseur MSP430 de Texas Instrument autant qu'un microprocesseur de faible puissance, Une présentation de la carte, les caractéristiques de MSP430, les registres, les interruptions et les timers.

Cette partie inclut aussi la présentation du Code Composer Studio de la famille Texas Instrument comme étant un compilateur de code et finalement nous avons représenté les programmes que nous avons simulés sur l'annexe A.

Chapitre IV : Les Processeurs traitement du signal

I. Les Processeurs traitement du signal (DSP)

1. Définition

Un DSP est caractérisé par le fait qu'il intègre un ensemble de fonctions spéciales. Ces fonctions sont destinées à le rendre particulièrement performant dans le domaine du traitement numérique du signal. On peut dire qu'un DSP est un type particulier de microprocesseur.

Comme un microprocesseur classique, un DSP est mis en œuvre en lui associant de la mémoire (RAM, ROM) et des périphériques. Un DSP typique a plutôt vocation à servir dans des systèmes de traitements autonomes. Il se présente donc généralement sous la forme d'un microcontrôleur intégrant, selon les marques et les gammes des constructeurs, de la mémoire, des timers, des ports série synchrones rapides, des contrôleurs DMA (Direct Memory Access), des ports d'E/S divers [16].

La figure 8 présente la chaîne complète typique d'un système de traitement numérique du signal :

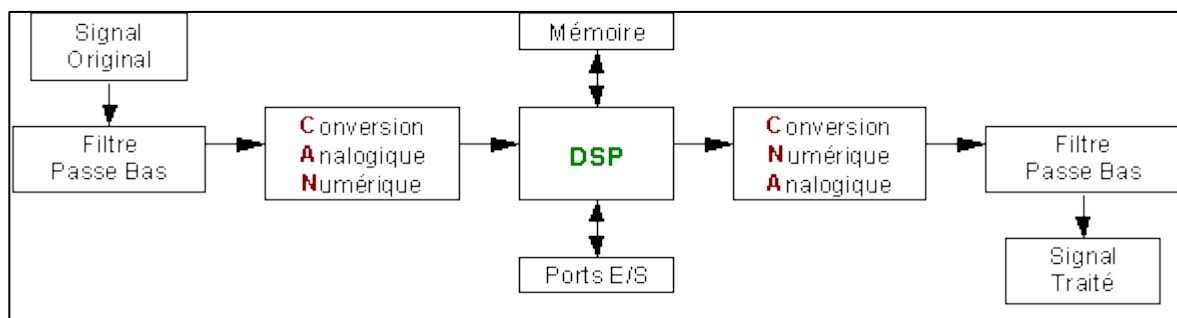


Figure IV.1: chaîne complète typique d'un système de traitement numérique signal

2. Avantage des processeurs traitement du signal

Tous les systèmes à bases de DSP bénéficient des avantages suivants :

- La Souplesse de la programmation Ainsi que la modification par exemple de tel ou tel paramètre d'un filtre numérique ne nécessite pas un changement matériel.

- Implémentation d'algorithmes adaptatifs : Il est possible d'adapter une fonction de traitement numérique en temps réel suivant certains critères d'évolutions du signal (exemple : les filtres adaptatifs).
- Certaines possibilités propres au système de traitement numérique du signal. En analogique sont difficiles à implanter, voire irréalisables (exemple : un filtre à réponse en phase linéaire).
- Stabilité : les composants en numérique sont stable par rapport à ceux en analogique, Ils sont toujours plus ou moins soumis à des variations de leurs caractéristiques en fonction de la température, de la tension d'alimentation, du vieillissement, etc.
- Reproductibilité : les valeurs des composants analogiques sont définies avec une marge de précision plus ou moins grande. Dans ces conditions, aucun montage analogique n'est strictement reproductible à l'identique, il existe toujours des différences qu'il convient de maintenir dans des limites acceptables. Un programme réalisant un traitement numérique est par contre parfaitement reproductible, « à l'infini ».

3. Structures Von Neuman et Harvard

L'architecture d'un DSP est un élément important qui conditionne directement les performances d'un processeur. Il existe deux types fondamentaux de structures, dites « Von Neuman » et « Harvard », telles que présentées par la figure suivante :

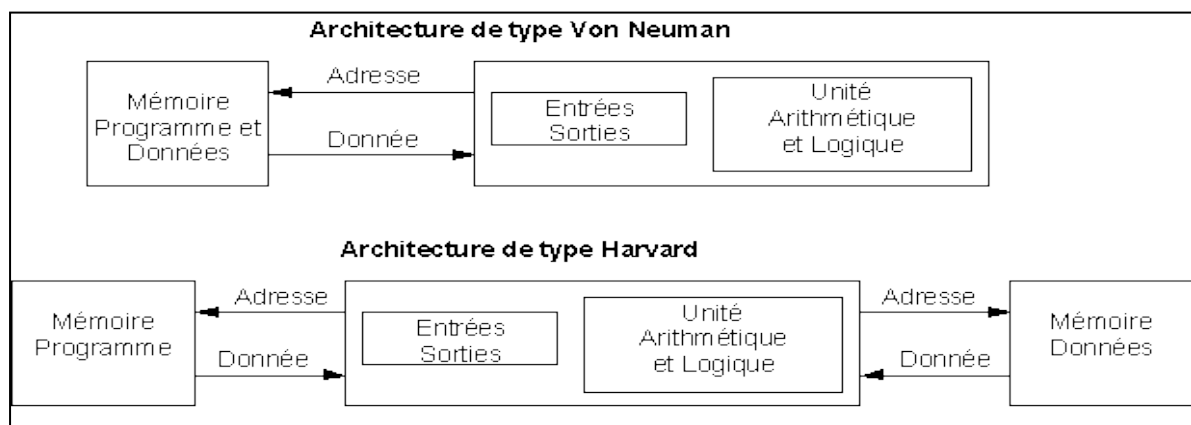


Figure IV.2: Différence entre la structure « Von Neuman » et « Harvard »

L'architecture généralement utilisée par les microprocesseurs est la structure Von Neuman (les programmes et les données dans la même zone mémoire). L'architecture Harvard (les mémoires programmes et données sont séparées) est plutôt utilisée dans des microprocesseurs spécialisés pour des applications temps réels, comme les DSP.

Il existe cependant quelques rares DSP à structure Von Neuman. La raison de ceci est liée au coût supérieur de la structure de type Harvard.

II. La carte TMS320C6713 DSP

TMS320C6713 DSP est un dispositif de 225 MHz offrant jusqu'à 1800 millions d'instructions par seconde (MIP) et 1350 MFLOPS. Cette génération de DSP est conçue pour les applications nécessitant une grande précision. Le C6713 est basé sur la plate-forme TMS320C6000 DSP conçu pour les besoins des applications de haute performance de haute précision tels que pro-audio, médical et diagnostique.

La figure 10 illustre la carte TMS320C6713, ainsi que ses principaux composants.

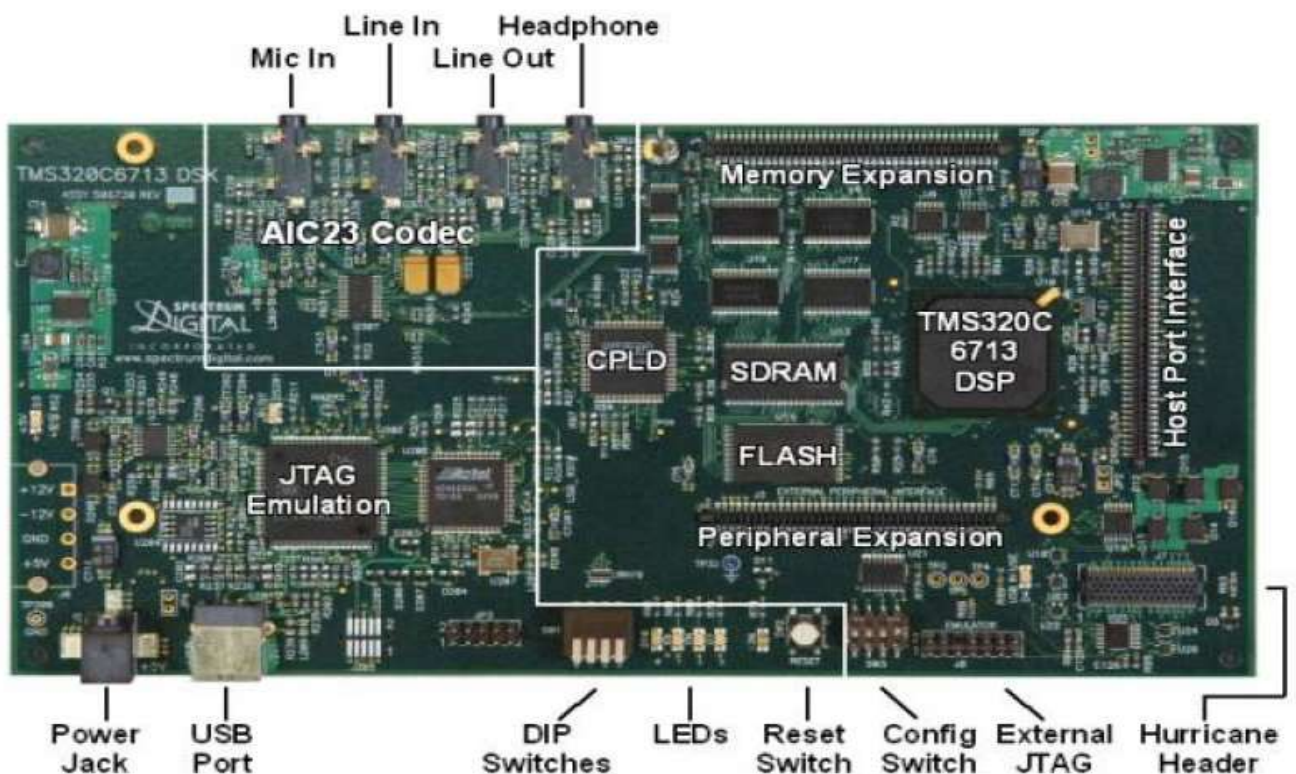


Figure IV. 3 : la carte TMS320C6713

Autres caractéristiques matérielles de la carte TMS320C6713 DSK comprend:

- Support de JTAG embarqué via USB
- Codec stéréo 24-bit de haute qualité
- Quatre 3.5mm jacks audio pour le microphone, ligne, haut-parleur et sortie ligne
- 512K mots de flash et 16 Mo de SDRAM
- connecteur de port d'extension pour modules de plug-in
- Sur la carte d'interface IEEE JTAG norme
- + 5V alimentation universelle

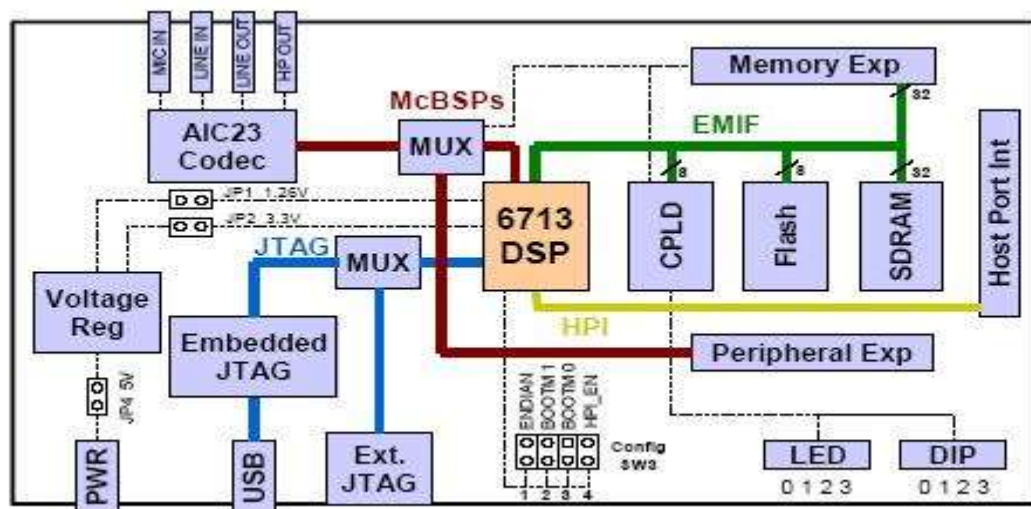


Figure IV.4: caractéristiques matérielles de la carte TMS320C6713 DSK

III. Les programmes simulés sur la carte TMS320C6713

Cahier de charge N°1 : Rédiger un programme qui permet un jeux de lumière en utilisant la directive `#define DELAY_TIME 100000` (Delay de 100 ms).

Cahier de charge N°2: Rédiger un programme qui permet le jeu de lumière on calculant le temps entre le clignotement de chaque LED et sans l'utilisation de #define
DELAY_TIME 100000.

Cahier de charge N°3 : Rédiger un programme qui permet le jeu de lumière en calculant le temps entre le clignotement de chaque LED et la boucle while .

Cahier de charge N°4 : Rédiger un programme qui recueille 44 échantillons à partir du canal gauche et droit du signal audio provenant de la carte son de l'ordinateur et filtrer le canal gauche.

Modifier le programme afin qu'il permet de fileter les deux canaux du signal.

Pour les programmes détaillés (voir l'annexe B).

IV. Conclusion

Ce chapitre introduit le DSP comme un type particulier de microprocesseur, ses avantages par rapport aux systèmes analogiques, la différence entre la structure Von Neumann et Harvard en précisant que dans un DSP on utilise la structure Harvard puisque un DSP est conçu pour les systèmes compliqués.

Cette partie présente aussi la carte TMS320C6713, ses composants et ses caractéristiques. Finalement on mentionne les programmes simulés sur le Code Composer Studio.

Chapitre V : AutoCAD

I. Introduction

Dans cette partie, On m'a demandé de choisir un plan architecte d'une villa de l'un des sites d'architecture, et de créer le plan 2D sur Autocad en respectant les exigences d'architecte telle que les nombres des chambres, les dimensions etc.

II. AutoCAD

1. Présentation du logiciel AutoCAD

AutoCAD est un logiciel de dessin assisté par ordinateur (DAO) créé en décembre 1982 par Autodesk. Bien qu'il ait été développé à l'origine pour les ingénieurs en mécanique, il est aujourd'hui utilisé par de nombreux corps de métiers. Il est actuellement, selon Forbes, le logiciel de DAO le plus répandu dans le monde [12]. C'est un logiciel de dessin technique pluridisciplinaire :

- industrie,
- système d'information géographique, cartographie et topographie,
- électronique,
- électrotechnique (schémas de câblage),
- architecture et urbanisme,
- mécanique.

2. L'interface du logiciel :

Ouvrir l'AutoCAD, une fenêtre s'affiche sur votre écran (voir figure 10) et créer un nouveau projet en cliquant sur (Start Drawing) :



Figure V.1 : Figure 3:Interface du logiciel AutoCAD

Pour commencer la création de votre projet, il faut tout d'abord connaître les composants de la barre d'outils et de commande qui est présenté dans la figure suivante :

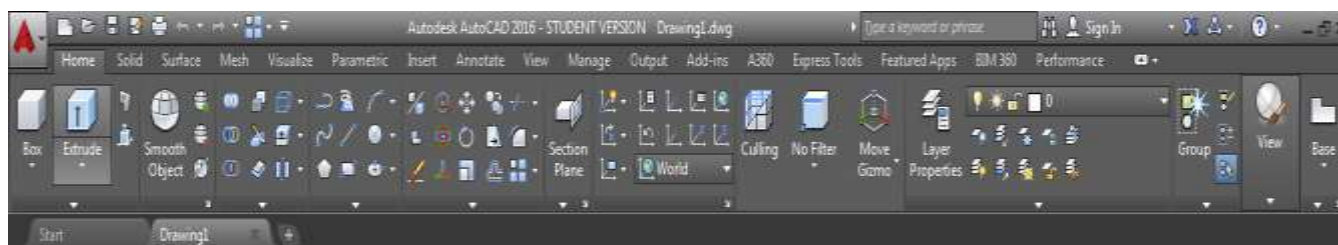









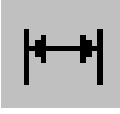




Figure V.2:la barre d'outils et de commande d'AutoCAD 2016

La barre d'outil et de commande contient beaucoup des options qu'on ne peut pas les représenter toutes, on va s'intéresser juste aux commandes qu'on a exploité afin de réaliser ce projet. Le tableau ci- dessous illustre quelques commandes nécessaires pour la création d'un projet sous AutoCAD :

Tableau 1:les commandes du logiciel AutoCAD

<i>Symbole de commande</i>	<i>Le nom de commande</i>	<i>La description de la commande</i>
	<i>Line</i>	Créer une série de segments de ligne contigus. Chaque segment est un objet de ligne qui peut être édité séparément.
	<i>Polyline</i>	Créer une polyligne 2D, un seul objet qui est composé de ligne et d'arc segments
	<i>Mirror</i>	Donner l'image à un objet

	<i>Move</i>	Déplacer les objets créés.
	<i>Copy</i>	Permettre de copier un objet, sans le recréer une autre fois.
	<i>Offset</i>	Décaler un objet à une distance déterminée ou à partir d'un point. Cette commande es très importante pour la création du projet
	<i>Fillet</i>	un arc qui est créé à la tangente entre deux objets en 2D.
	<i>Rotate</i>	Cette commande est pour la rotation des objets sélectionnés.
<i>Scale</i>	<i>Scale</i>	Cette commande est pour redimensionner un objet, spécifiez un point de base et un facteur d'échelle. Un facteur d'échelle supérieur à 1 agrandit l'objet. Un facteur d'échelle entre 0 et 1 rétrécit l'objet..
	<i>Trim</i>	Permettre de supprimer une section d'une entité entre deux intersections
	<i>Linear Dimensions</i>	Vous pouvez créer des dimensions horizontales, verticales, alignées, et radiales avec la commande DIM.
	<i>Layers</i>	Utilisez des Layers pour contrôler la visibilité des objets et d'affecter des propriétés telles que la couleur et le type de ligne. Les objets dans un Layer subissent normalement aux propriétés de ce-dernier.
	<i>EXTRUDE</i>	Créer un solide 3D à partir d'un objet qui délimite une zone ou une surface 3D à partir d'un objet avec des extrémités ouvertes.

III. La création d'un plan 2D d'une Villa :

1. Travail demandé

Sur le site z500.pl , il y a plusieurs plans architecte dont j'ai choisi le projet ci-dessous.

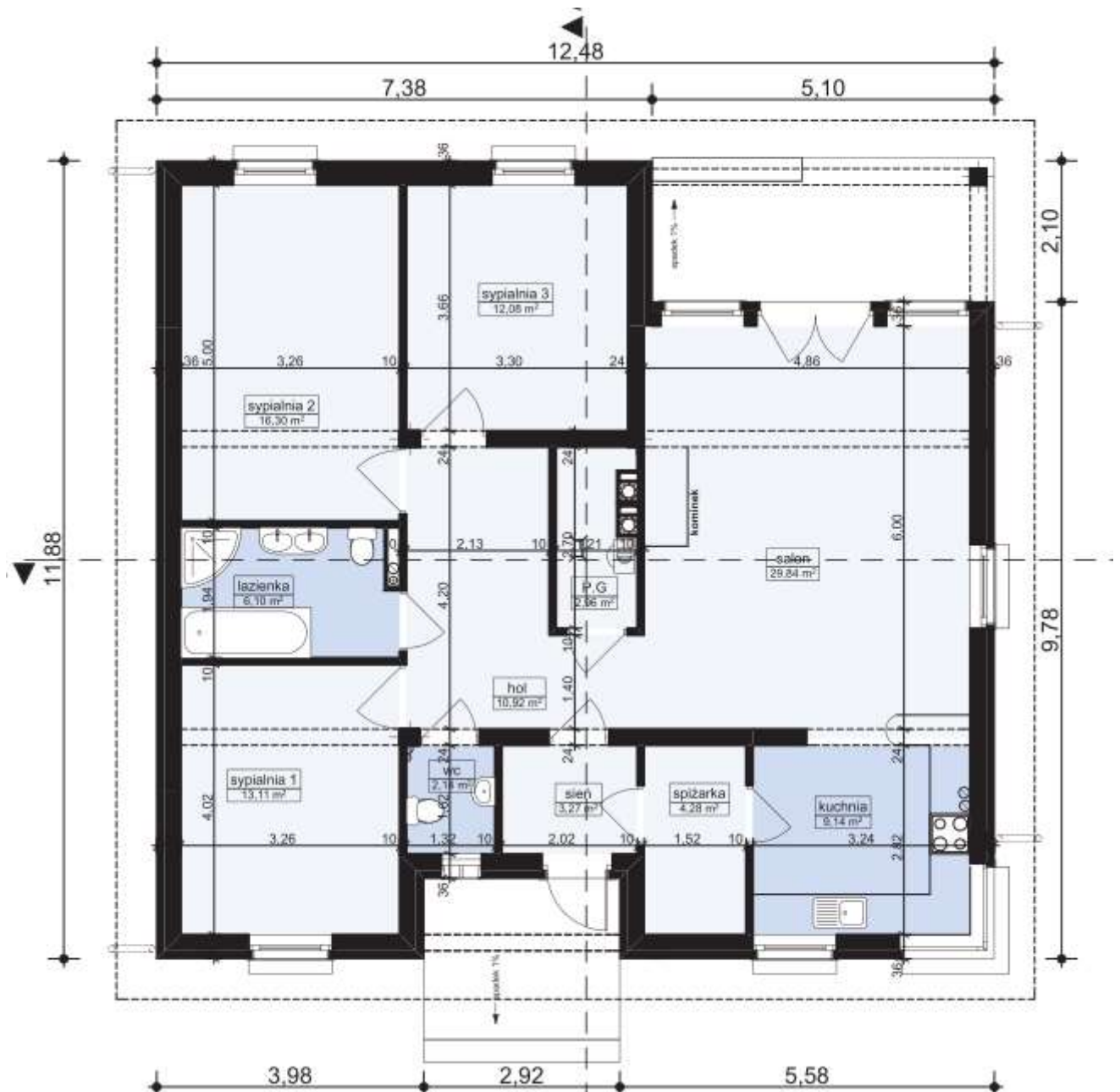


Figure V.3: Le plan architecte d'une villa

2. Résultat du travail

Le plan s'est constitué sur 7 étapes :

- ✓ Dessiner la forme du plan en respectant la forme et les dimensions du plan original.
- ✓ Création des chambres en respectant leurs formes .
- ✓ La Création des portes en utilisant la commande PLINE.
- ✓ La Création des fenêtres.
- ✓ Ajouter des layers au projet en précisant à chaque couleur un élément.
- ✓ Ajouter les meubles par la commande Design Center.
- ✓ Ajouter des textes pour nommer les parties du plan.



Figure V.4 : Figure 4:le plan 2D réalisé sur AutoCAD

IV. Le plan de la villa en 3D

Grace à la commande EXTRUDE nous avons pu réaliser le plan en 3D, comme c'est montré dans la figure ci-dessous [18] :

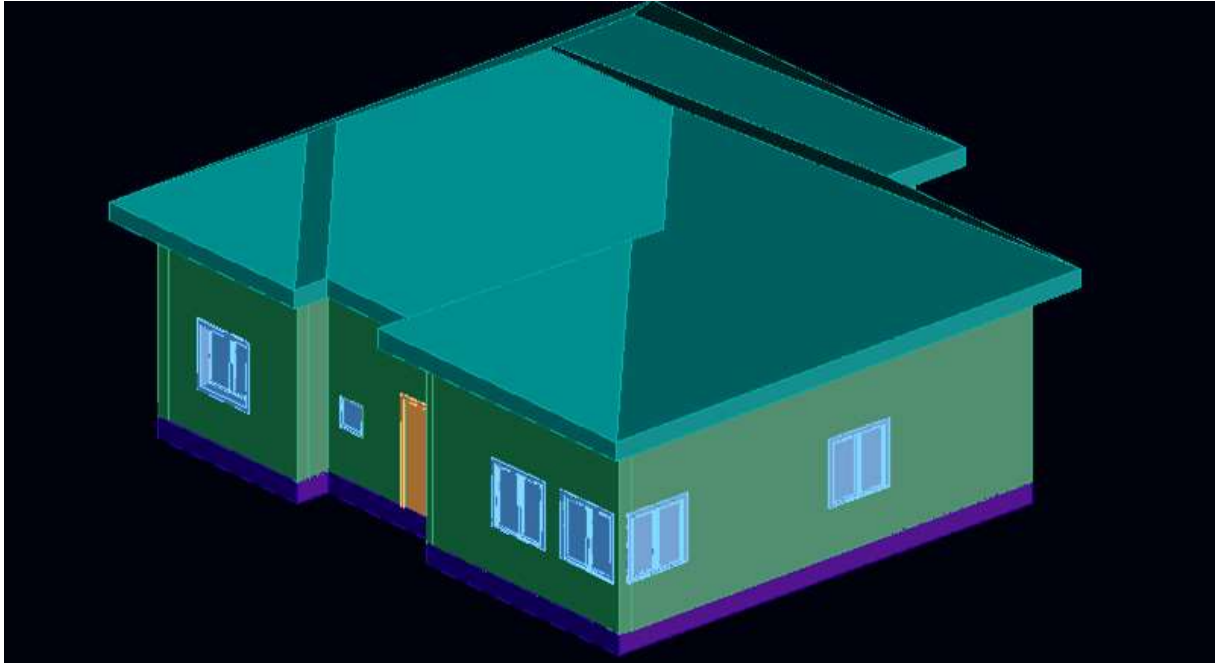


Figure V.5 : le plan de la villa en 3D

V. Conclusion

Ce chapitre donne au lecteur une vision générale sur l'AutoCAD autant qu'un logiciel de dessin assisté par ordinateur, il présente aussi les commandes utilisées pour la réalisation du projet.

La deuxième partie montre le plan sur lequel nous avons travaillé, le résultat du travail en deux dimensions puis le plan de la villa en trois dimensions.

Chapitre VI : Diagnostic des équipements électriques

I. Introduction générale

Dans le cadre du module « Diagnostic des équipements électriques de haute tension », le sujet de recherche qui nous a été proposé concerne l'élément indispensable de toutes industries, c'est celui du transformateur et l'analyse de ses défaillances.

Dans cette partie On va présenter une méthode très utilisée dans les industries pour le diagnostic des transformateurs, celle de l'analyse d'huile du transformateur.

II. Transformateur de puissance

1. Le rôle des transformateurs de puissance (AC /AC)

Un Transformateur de puissance c'est un Appareil statique à deux enroulements ou plus qui, par induction électromagnétique, transforme un système de tension et courant alternatif en un autre système tension et de courant de valeurs généralement différentes à la même fréquence dans le but de transmettre de la puissance électrique.

Les transformateurs sont réversibles et permettent, soit l'élévation, soit l'abaissement de la tension ; d'où la possibilité de choisir la meilleure tension pour la production, le transport, la distribution, l'utilisation de l'énergie électrique, et de passer facilement de l'une à l'autre. Un transformateur électrique permet donc de faire transiter la puissance électrique en courant alternatif (AC) depuis la centrale de production d'électricité jusqu'à son utilisateur final avec un minimum de pertes, à différents niveaux de tensions. Le transformateur est l'élément du réseau qui permet le passage d'une tension alternative donnée à une autre : c'est un convertisseur AC/AC à fréquence fixe, celle du réseau [13], [14], [15].

2. Fiabilité et taux de défaillance des transformateurs

Les transformateurs se caractérisent par leur fiabilité, leur durée de vie est de l'ordre de quarante ans dans des conditions normales d'utilisation. Les pannes qui causent un arrêt immédiat du transformateur sont rares, mais ils sont souvent problématiques à cause de l'indisponibilité du matériel, les défaillances qui entraînent un arrêt des transformateurs de puissances sont de l'ordre de 2% de la totalité des transformateurs dans le monde entier.

Parmi les problèmes qu'on trouve dans un transformateur et qui peuvent causer un dysfonctionnement ou parfois un arrêt immédiat de ce dernier, on trouve un problème dans l'huile d'isolation du transformateur. D'où vient la nécessité de surveillance et de contrôle d'huile fréquemment.

3. Huile isolante du transformateur

L'huile isolante dans un transformateur de puissance électrique est connue sous le nom d'huile du transformateur. Il est généralement obtenu par la distillation fractionnée et le traitement du pétrole brut. Voilà pourquoi cette huile est également connue comme huile minérale isolante. L'huile de transformateur sert principalement à deux choses, utilisée autant qu'un liquide isolant dans un transformateur de puissance électrique et il dissipe la chaleur du transformateur agit comme réfrigérant. En plus de cela, cette huile sert à deux autres objectifs, elle contribue à préserver le noyau et les enroulements qui sont complètement immergés à l'intérieur de l'huile et un autre objectif important de cette huile est, elle empêche le contact direct de l'oxygène atmosphérique avec de la cellulose du papier faisant l'isolation des enroulements, ce papier est sensible à l'oxydation [13], [14], [15].

4. Défauts causés par une mauvaise huile

Défaut d'isolement de la partie active

Ce type de défaut se caractérise par la diminution ou la perte de l'isolement d'un circuit électrique par rapport à son environnement.

Décharges Partielles

Une décharge partielle (DP) est une « décharge électrique localisée qui court-circuite partiellement l'intervalle isolant séparant des conducteurs. Ces décharges sont rarement destructives, mais l'évolution peut l'être car à terme le carbone, conducteur, qu'elles génèrent peut se propager et diminuer la tenue diélectrique entre certains éléments sous tension et ainsi aller jusqu'à provoquer un amorçage.

III. Les tests de l'huile des transformateurs

1. Le visuel et la couleur d'huile

Ce test vérifie la turbidité, la nébulosité, les particules suspendus et la couleur. La nouvelle huile est lumineuse et claire, sans impuretés visuelles et de couleur jaune pâle comme la montre la figure 17. Ainsi la couleur est vérifiée par une valeur numérique qu'on la compare avec une série standard des valeurs.



Figure VI.1: diagnostic d'huile du transformateur de sa couleur

2. L'indice de neutralisation (acide)

Ce test mesure l'indice de neutralisation. Lorsque l'huile s'oxyde dans un transformateur, des acides et des luges sont produits avec de l'eau. Une augmentation importante de l'indice de neutralisation peut être nuisible au système d'isolation. L'indice de neutralisation de l'huile utilisée est inférieur ou égal 0,3. Par conséquent, si l'indice de neutralisation de l'échantillon d'huile est supérieur à 0,3, il indique que l'huile de transformateur doit être changée, comme illustre la figure ci-dessous [19] :

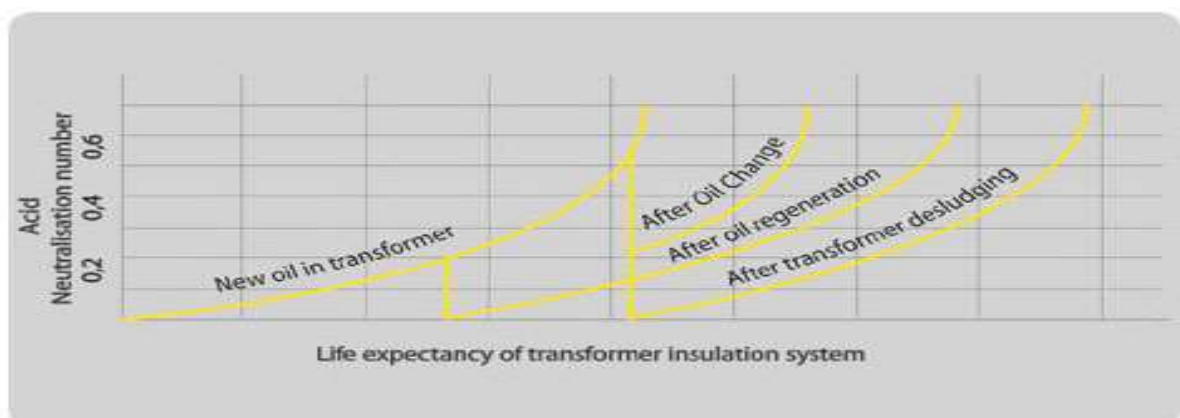


Figure VI.2: L'indice de neutralisation dans l'huile d'un transformateur

3. Tension de claquage

La tension de claquage de l'huile de transformateur est également connu comme Breakdown voltage ou BDV. BDV est mesurée en observant à quelle tension, on aura des étincelles entre les deux Électrodes immergés dans l'huile (voir figure 19), séparés par espace spécifique. Une faible valeur de BDV indique la présence de l'humidité et des substances conductrices dans l'huile. Pour mesurer BDV d'huile de transformateur, on utilise un kit de mesure BDV portable [13], [14], [15]..

La tension monte lentement est appliquée entre les électrodes. Taux de croissance de la tension est généralement contrôlée à 2 KV / s et d'observer la tension à laquelle étincelles commence entre les électrodes.



Figure 5 les deux électrodes de la BDV KIT

4. Humidité

C'est l'eau dans l'huile du transformateur est fortement indésirable car elle affecte négativement les propriétés diélectriques du pétrole. La teneur en eau dans l'huile affecte également l'isolation en papier des enroulements du transformateur. Le Papier absorbe quantité maximale d'eau qui affecte la propriété du papier d'isolation, ainsi que de réduire sa durée de vie. Ainsi, la température de l'huile au moment du prélèvement d'échantillon pour le test est très importante. La teneur en eau dans l'huile est mesurée en parts per million (ppm). La valeur maximum acceptée est de 35ppm, une valeur supérieure montre que l'huile doit être changée [20].

5. La viscosité d'huile du transformateur

On peut dire que la viscosité est la résistance à l'écoulement, à l'état normal. Une bonne huile doit présenter une faible viscosité. Une faible viscosité de l'huile du transformateur est indispensable, mais il est tout aussi important que la viscosité de l'huile devrait augmenter le moins possible avec la diminution de la température, la figure 20 décrit le comportement de viscosité en fonction de la température. Chaque liquide devient plus visqueux si la température diminue.

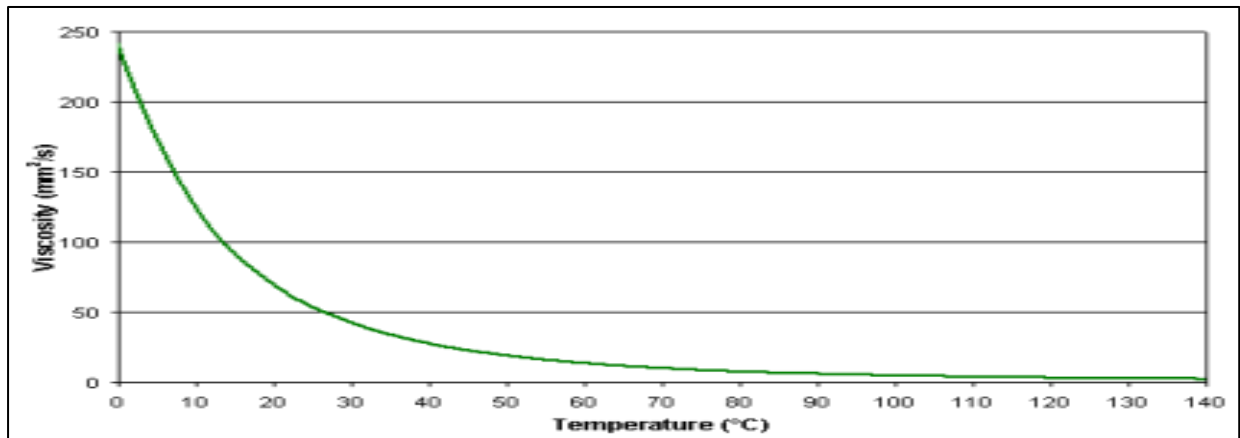


Figure VI.4: La variation de la viscosité avec le changement de la température

6. Densité

La densité de l'huile d'un échantillon est également à prendre en compte pour l'analyse. La valeur préférée de la densité est inférieure à 0,89 g / cm³. Par conséquent, lors de la lecture de la valeur de l'essai pour l'huile ,si la densité est supérieure à 0,89 g / cm³, il indique que l'huile doit être changée. La figure suivante montre la variation de la densité avec le changement de la température [20]:

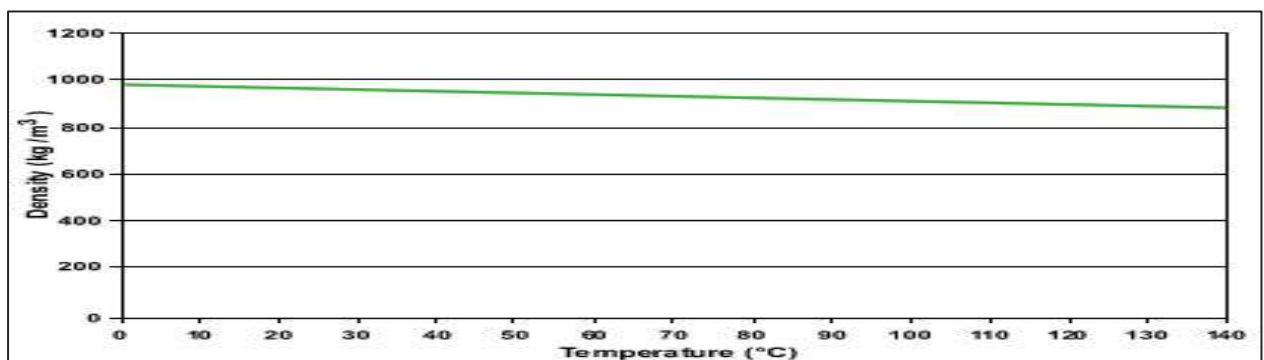


Figure VI.5: La variation de la densité avec le changement de la température

7. Résistance spécifique

À 90 ° C, la résistivité de l'huile du transformateur est censé $0,1 \times 10^{12} \text{ohm-cm}$. Si la résistance d'huile de transformateur à 90 ° C est inférieure à $0,1 \times 10^{12} \text{ohm-cm}$ c-à-dire l'huile est de mauvaise qualité, ce qui nécessite un remplacement [13], [14], [15]..

8. Facteur de puissance

Le facteur de puissance de l'huile isolante est égal au cosinus de l'angle de phase entre une tension alternative appliquée et le courant qui en résulte .

Ce test mesure le courant de fuite qui passe grâce à l'huile. Il est devenu un indicateur très sensible en ce qui concerne la détérioration, ainsi que l'un des tests utile dans l'industrie. Plus le facteur de puissance. Cependant, il peut être analysé à partir de l'évaluation du facteur de dissipation. Un facteur de puissance élevé indique une détérioration, la contamination, ou les deux avec l'humidité, de carbone. Pour l'huile minérale, le facteur de puissance de la nouvelle huile ne doit pas dépasser 0,05 pour cent à 25°C.

IV. Conclusion

Les transformateurs de puissance sont des appareils complexes, qui doivent tenir de nombreuses et fortes contraintes, de natures diverses, durant leur cycle de vie. De plus, dans le contexte actuel, avec un parc d'appareils vieillissant et un réseau de plus en plus chargé, les transformateurs seront soumis à des contraintes de plus en plus importantes, pendant encore plusieurs années à priori. Bien que leur fiabilité soit plutôt bonne, les transformateurs restent des éléments critiques de tout réseau électrique. Il est donc très important d'être capable d'en assurer un diagnostic de l'huile du transformateur et le dépannage au mieux, afin d'optimiser l'exploitation du matériel lui-même, et surtout de garantir la transmission, importante, d'énergie électrique qui transite par lui.

Chapitre VII : Robotique

I. Introduction

Cette partie décrit la réalisation d'un robot programmé par Arduino avec deux concepts, le premier c'est un robot détecteur d'obstacle, et le deuxième robot c'est un robot contrôlé par Smartphone dont on décrira toutes les étapes dans ce chapitre.

II. Les composants du Robot :

1. La partie mécanique

En effet, au vu des composants disponibles ainsi que la petite taille des composants électroniques, notre choix se porta sur un véhicule basique, de forme rectangulaire avec d'un côté les deux moteurs assemblés en une pièce et de l'autre une simple roue pouvant tourner à 360 degré, laissant ainsi la direction aux deux moteurs. La photo ci-dessous présente le châssis ainsi que les roues :



Figure VII.1:Châssis et les roues du robot

2. La carte Arduino

Arduino est une plate-forme open-source utilisé pour la construction de projets d'électronique. L'Arduino est constitué à la fois d'une carte physique programmable de circuit (souvent désigné comme un microcontrôleur) et un morceau de logiciel, ou IDE

(Integrated Development Environment) qui fonctionne sur votre ordinateur, utilisé pour écrire et télécharger le code d'ordinateur à la carte physique.

La figure 23 illustre les composants d'une carte Arduino Uno.

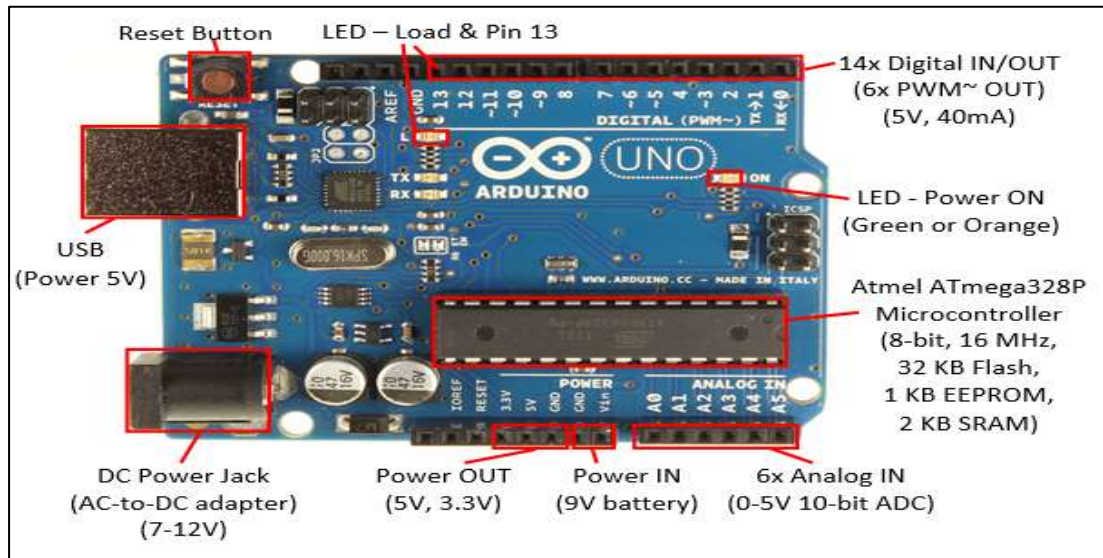


Figure VII.2 : les composants de la carte Arduino Uno

3. Capteur à ultrasons HC-SR04

Un capteur à ultrasons HC-SR04 est capable de mesurer la distance qui le sépare d'un objet quelconque. Pour le robot réalisé ce capteur sert à calculer précisément distance le séparant de l'obstacle, il fonctionne par émission d'ultra-son et détection de l'écho en retour.



Figure VII .3 : Capteur à ultrasons HC-SR04

Le capteur à ultrasons HC-SR04 contient 4 connexions (Voir la figure VII.3) :

- 1. VCC
- 2. Trig (T)
- 3. Echo (R)
- 4. GND

Tableau 2 : les paramètres techniques de HC-SR04

Paramètres techniques	Les valeurs
Tension de fonctionnement	5V (DC)
Consommation de courant	moins de 2 mA
Signal de sortie	5V haut niveau, 0V niveau bas
Angle de visée du détecteur	moins de 15 degrés
Distance de détection	de 2 cm à 450 cm
Haute précision	jusqu'à 0.3 cm
Signal d'écho en retour	signal de sortie TTL en PWL
Signal d'entrée de déclenchement (trigger)	impulsion TTL de 10us

Utilisation du module

Alimentez le module en +5V, la sortie sera alors 5V s'il y a un obstacle dans la portée de détection, et de 0V sinon.

Principe de fonctionnement

- Envoyez un signal de déclenchement IO qui fournit un signal d'au moins 10us sur le niveau haut.
- Le module va alors automatiquement envoyer huit impulsions d'un signal ultrasonore de fréquence 40 kHz de forme carrée.
- Il va détecter automatiquement le signal de retour de ces impulsions en écho.
-
- S'il y a un signal de retour en écho, la sortie passe au niveau élevé, et son temps de maintien (en s) correspond au délai de retard de réception de l'écho (modulation de largeur d'impulsion).

On calcul de la distance de l'obstacle (m) :

Distance = (temps pendant lequel la sortie est au niveau haut * (340 m/s) / 2.

(le son se propage à la vitesse $C = 340\text{m/s}$ et fait un aller-retour).

La figure ci-dessous décrit le fonctionnement du Un capteur à ultrasons HC-SR04.

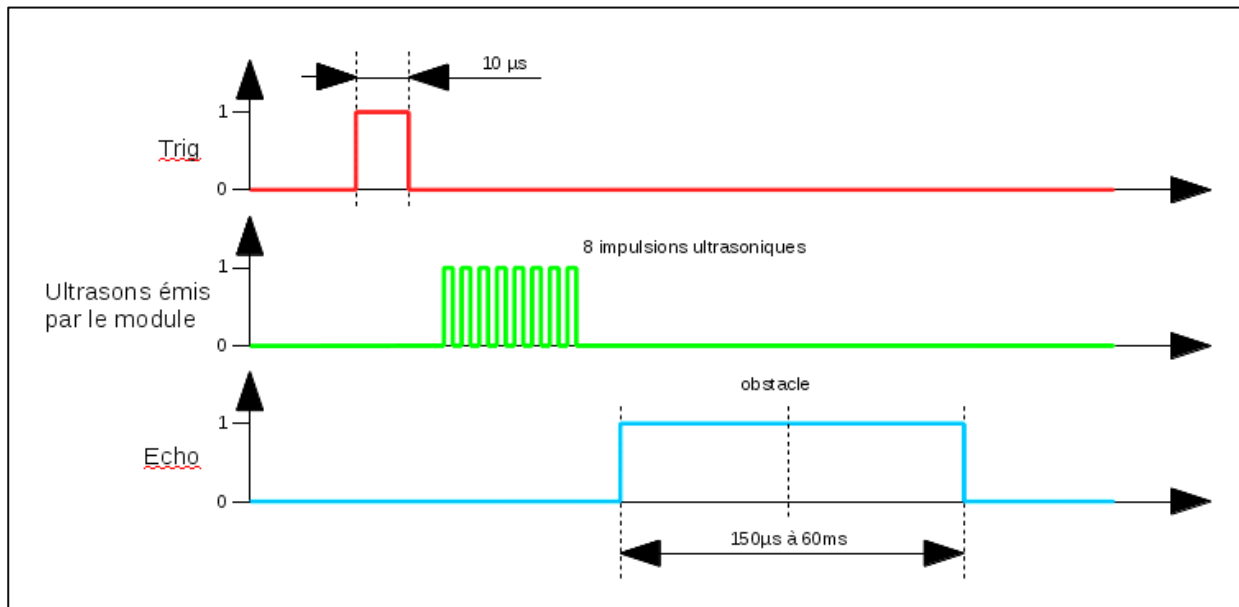


Figure VII .4: Le principe de fonctionnement du capteur à ultrasons HC-SR04

3. moteurs à courant continu

Les moteurs DC sont des moteurs électriques qui sont alimentés par un courant continu (DC), comme à partir d'une batterie ou alimentation en courant continu. La vitesse d'un moteur à courant continu balayé peut être commandée en changeant la tension seule.

Le moteur utilisé dans le projet est illustré dans la figure suivante :



Figure VII .5 : Moteur à courant continu

Le tableau ci-dessous présente les caractéristiques du moteur DC :

Tableau 3: les caractéristiques du moteur DC

Les caractéristiques du Moteur DC	Les Valeurs
Voltage	3 - 12 V
courant	70 mA (typical) - 250 mA (max)
vitesse	Jusqu'à 170 tr /min
Torque	jusqu'à 0.8 Kg

4. Moteur shield L298N

Moteur Shield est un module de commande du moteur propre sur la base du L298N H-bridge IC qui peut vous permet de contrôler la vitesse et la direction des deux moteurs à courant continu, ou le contrôle d'un moteur pas à pas bipolaire avec facilité.

Le module L298N peut être utilisé avec des moteurs ayant une tension comprise entre 5 et 35V en courant continu.

Les éléments de Moteur Shield L298N sont mentionnés dans la figure ci-dessous :

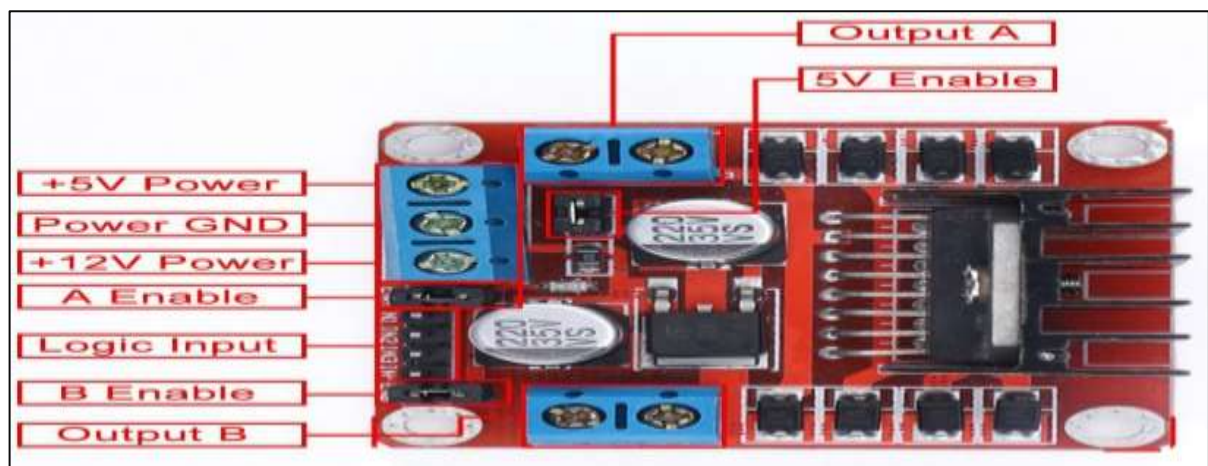


Figure VII .6 : les composants du Moteur Shield L298N

5. Module HC-05

Un Module HC-05 est un outil Bluetooth SPP (Serial Port Protocole) facile à utiliser, conçu pour une configuration transparente de connexion série sans fil.

Le branchement du HC-05 :

RXD->11

TXD->10

GND->GND côté digital

VCC -> 5V

III. Robot détecteur d'obstacle

Le premier robot détecteur d'obstacle est équipé de plusieurs éléments, on cite parmi ces éléments :

- La carte Arduino.
- Un capteur HC-SR04.
- 2 moteurs DC.
- Moteur Shield .
- Un châssis à 2 roues.
- Un buzzer, breadbord , des leds, des résistances ,2 batteries 9V et des fils.

1. Principe du Robot détecteur d'obstacle

Le principe de ce Robot est d'éviter les obstacles .Il doit suivre son chemin en avançant, la led verte est allumé et le buzzer est arrêté. Quand le capteur détecte un obstacle, la led verte s'éteint et la led rouge s'allume ainsi que le buzzer commence à sonner. Le robot recule pendant un temps prédéfinis (100 ms) puis il tourne à droite et suivre son chemin en avançant [21], [22].

2. Le montage du Robot détecteur d'obstacle

La réalisation du Robot s'est effectuée sur 3 étapes :

- a) Le montage des roues avec châssis, ainsi que la fixation des roues avec avec les moteurs.
- b) Le brochage des deux moteurs au driver (Motor Shield), et la connexion du Shield avec l'Arduino afin de contrôler les deux moteurs du robot.

La figure ci-dessous présente la connexion entre Arduino et Moteur Shield.

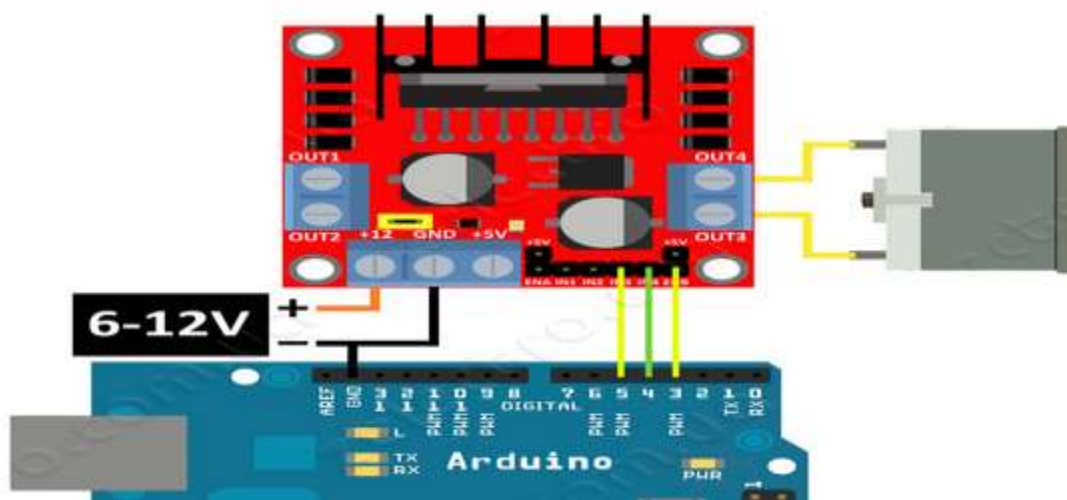


Figure VII .7: le brochage du moteur shield avec le moteur et l'Arduino

- c) Le Brochage d'arduino avec le capteur à ultrason ,ainsi que le brochage des leds, et le buzzer (n'est pas mentionné dans la figure ci-dessous)

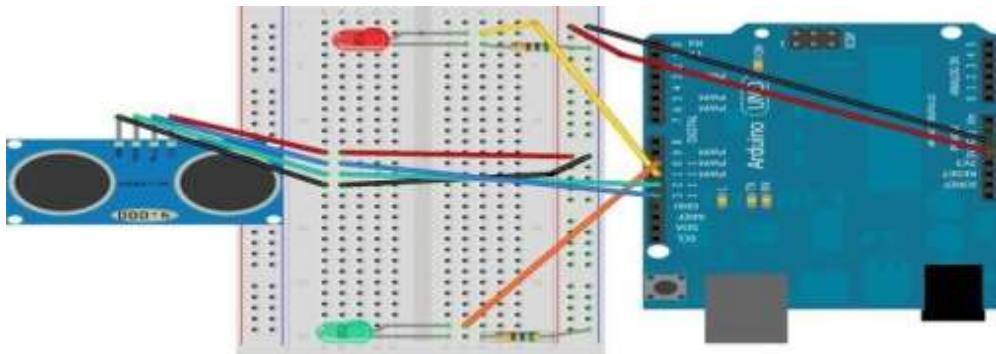


Figure VII .8 : le Brochage du capteur avec l'arduino

Après on alimente la carte Arduino et les moteurs avec 9v.

La figure ci-dessous présente le montage du capteur avec l'arduino ainsi que le montage final.

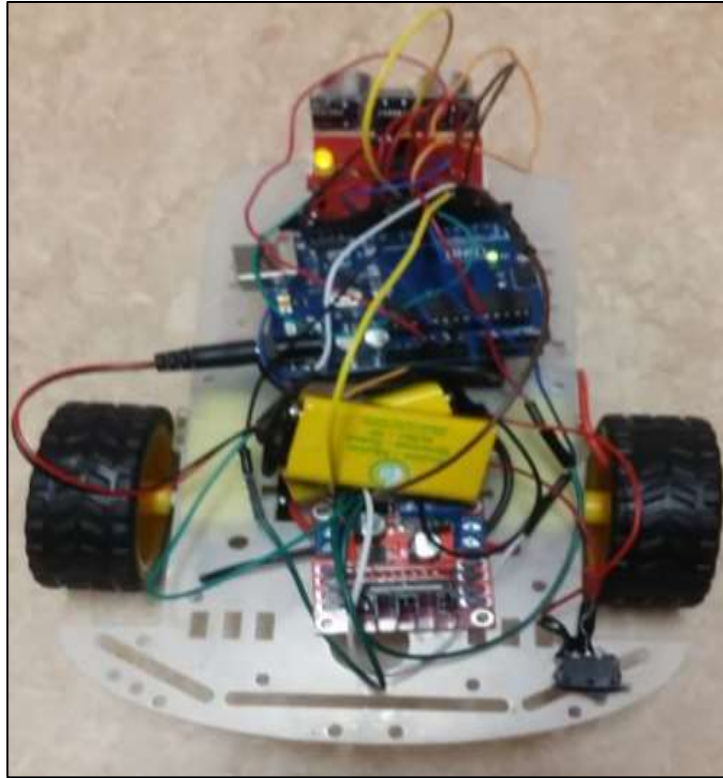


Figure VII .9 : le montage du Robot détecteur d'obstacles

Pour le programme d'arduino du robot détecteur d'obstacle (voir l'annexe C).

IV. Robot contrôlé par Android

Le deuxième robot est un robot programmé par Arduino et contrôlé par une application créée sous Android. Premièrement nous allons voir la création de l'application Android ,puis nous allons passer au programme Arduino et finalement le montage du robot et le brochage du module Bluetooth avec l'Arduino.

1. La présentation d'App Inventor

App Inventor est un outil de développement en ligne pour les téléphones et les tablettes sous Android. App Inventor est un OS (operating system) créé par Google, et concurrent de l'Ios d'Apple qui équipe l'iPad, iPodTouch et iPhone. La plateforme de développement est offerte à tous les utilisateurs possédant un compte Gmail.

La programmation se réalise en ligne, à l'aide de son navigateur préféré, et sous l'environnement logiciel de son choix (Mac, Linux ou Windows). Seules contraintes : avoir

un compte Gmail pour pouvoir y accéder, et un accès à internet évidemment. Les informations sont stockées sur des serveurs distants.

2. Le concept d'App Inventor

Pendant le développement de votre projet ,vous allez rencontrer 3 fenêtres :

- Une fenêtre sert à la création de l'interface homme machine a pour but de représenter l'allure de l'application (voir la figure ci-dessous) :



Figure VII .10 : la fenêtre de la création de l'interface homme machine

- Une fenêtre pour la programmation, elle permettra l'assemblage de blocs et la création du comportement de l'application (voir la figure suivante):



Figure VII . 11 : La fenêtre de programmation

- Un émulateur qui permettra de tester l'application. L'émulateur permet de remplacer un Smartphone ou une tablette afin de s'assurer que l'application fonctionne avant la télécharger sur un téléphone.

3. la programmation événementielle

La programmation événementielle celle utilisée par App Inventor consiste sur l'utilisation des boutons, et par la suite définir le comportement du programme quand un bouton est cliqué. Il n'est donc pas nécessaire de tester son état en permanence, c'est lui qui réagit quand se produit l'événement prévu par le programmeur. Encore faut-il prévoir cet événement et surtout le traitement à opérer quand il survient.

4. Les étapes de réalisation d'une application sur App Inventor

1ère étape du projet : « Dessiner l'interface »

La création de l'interface est la première phase de la création d'une application sur App Inventor. Pour cela, le site affiche un écran de téléphone dans lequel nous pouvons placer, en les faisant glisser, les éléments que nous voulons utiliser.

Il y a des éléments graphiques comme des boutons, des labels... et des éléments non graphiques comme des capteurs (ex. géo localisation...) ou des fonctions permettant d'effectuer des actions : lecteur audio, appareil photos

Pour notre application :

- a. Afin de créer l'application, un label s'est ajouté pour afficher si le Bluetooth est connecté ou déconnecté.
- b. Des photos ajoutés à l'interface qui va être affiché sur le téléphone pour contrôler le robot aisément ;des flèches (haut, bas, gauche, droite) .. .
- c. une table pour arrangement, dans laquelle on a ajouté 5 boutons, dans chacune d'entre eux est inséré une image parmi les images téléchargés pour contrôler le robot.
- d. L'ajout de d'un Bluetooth client pour connecter le module Bluetooth au téléphone.
- e. Il faut ensuite définir les propriétés de chaque élément : choix de texte, la taille, couleur de texte, ajout des images...

2ème étape du projet : « passer à la programmation »

Pour cela, il faut cliquer sur « Blocks » en haut et à droite de la page. Dans cette partie est décrit le comportement de chaque élément.

Voici l'algorithme résumant ceci :

QUAND le Bouton1 est cliqué, faire en sorte que :

- La propriété Visible du Label2 soit VRAIE (donc afficher le label2)
- La propriété Visible du Label1 soit VRAIE (donc afficher le label1)
- Le label 1 affiche le texte de la TextBox (donc afficher le texte

Pour Notre Application, la programmation est divisée en 3 partie :

- a. L'ajout d'un ListPicker gère tous les détails de la sélection sur l'écran comme ci-mentionné dans la figure suivante. Tout ce qui est nécessaire est d'aller chercher l'élément qui a été sélectionné. Voilà pour le fonctionnement de base de ListPicker. Si vous avez de nombreux éléments - trop nombreux pour tenir sur un seul écran - la liste gère automatiquement le défilement. Il suffit d'utiliser votre doigt pour faire glisser vers le haut ou en bas de la liste. Vous pouvez également modifier les propriétés ListPicker telles que la couleur de la liste et plus encore.

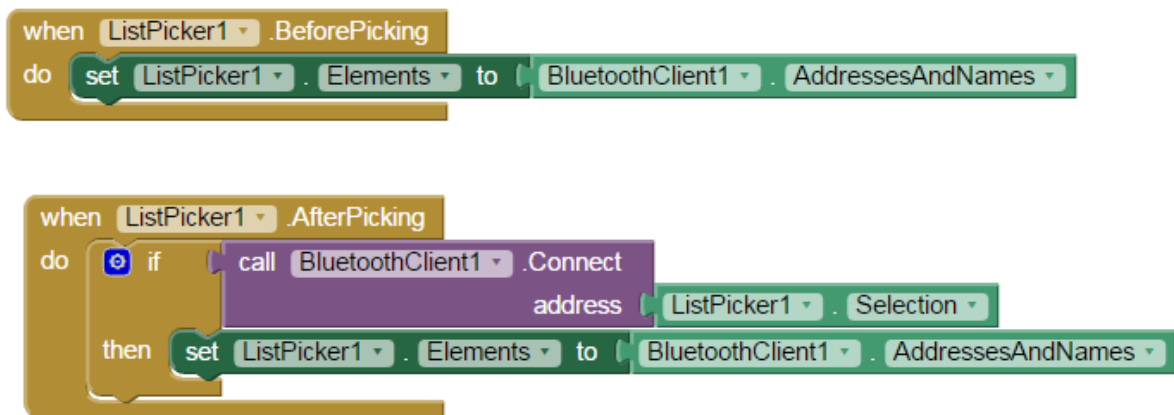


Figure VII .12 : La configuration du ListPicker

- b. Ici on a ajouté Timer « Clock1 », pour tester si le Bluetooth est connecté ou déconnecté, si le Bluetooth est connecté il s'affiche 'connected' sur l'écran avec une couleur verte,

sinon il s'affiche 'not connected' avec une couleur rouge sur l'écran.

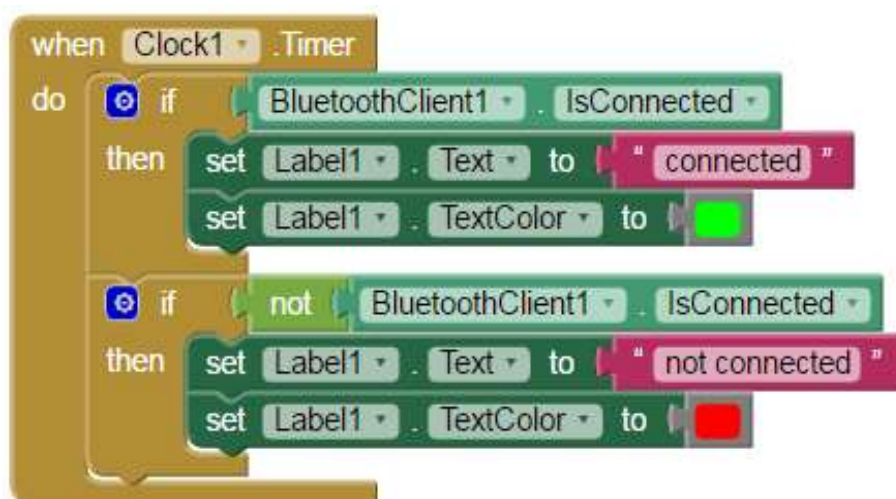


Figure VII .13 : Contrôle du bluetooth

Pour le teste des boutons, on prend l'exemple du bouton 2. Quand on clique sur le bouton2, via Bluetooth « BAS » est envoyé à l'Arduino afin de commander les moteurs de reculer.



Figure VII .14 : La configuration des boutons

3ème étape du projet : « Tester l'application sur l'émulateur ou le smartphone»

Enfin, pour tester l'application sur votre téléphone l'application MIT AI2 Companion doit être installée à télécharger sur le PlayStore. En choisissant AI Companion : un Code de 6 caractères est généré ainsi qu'un QR Code. Il suffit de saisir le code ou de scanner le QR Code pour que l'application soit visible sur le smartphone et ensuite la tester.

La première figure à gauche présente le brochage du module Bluetooth avec l'Arduino, la deuxième illustre l'application qui s'affiche sur l'écran afin de contrôler le robot à distance via Smartphone.

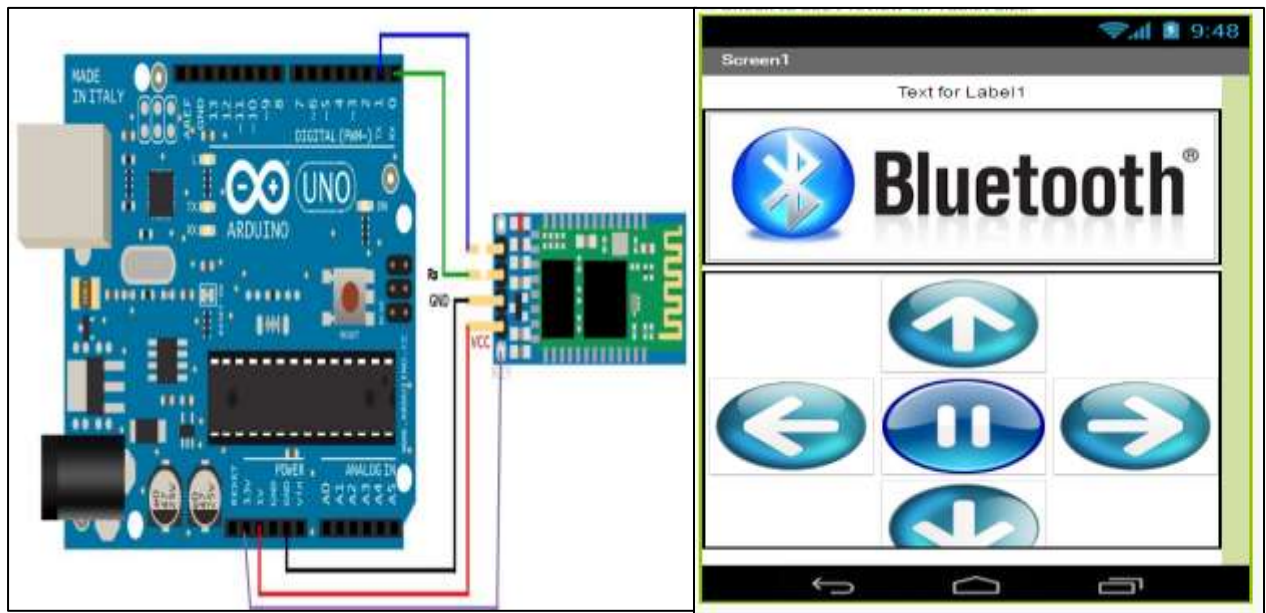


Figure VII .15: Brochage du module Bluetooth et l'application Android

IV. Conclusion

Dans ce chapitre, nous avons réalisé et programmé deux robots en utilisant la solution Arduino. Pour Ce projet nous a permis, d'une part, d'améliorer grandement nos compétences scientifiques, que ce soit du côté programmation, ou du côté électronique.

Conclusion générale

Grace au programme Erasmus + nous avons pu effectuer notre PFE à l'Université Polytechnique Opole en Pologne pour un semestre, en bénéficiant d'une bourse chaque mois.

Notre projet de fin d'étude à l'Université Polytechnique Opole est un peu spécial, puisque il n'avait pas un nom spécifique, j'ai du chercher dans les modules proposés et choisir les modules que nous jugions les plus intéressants.

Ce PFE orienté enseignement et recherche m'a permis tout d'abord de mettre en œuvre mes capacités d'analyse et d'adaptation, fruits de la formation pluridisciplinaire qui nous a été donnée durant nos deux années d'études à la FSTG. En outre étant qu'élève Master aspirant à la recherche scientifique et technologique, cette expérience fut fort enrichissante pour nous.

En plus du côté académique, l'expérience d'ERASMUS m'a permis de contacter plusieurs personnes de différents pays, d'améliorer mon anglais, ainsi que nous avons suivi des cours en langue chinoise.

Nous avons bénéficié aussi d'une attestation de stage que nous avons effectué à l'office des relations internationales autant qu'une responsable de créer des relations avec les universités françaises.

Références

- [1] <https://fr.wikipedia.org/wiki/Erasmus>. Consulté le 15/04/2016.
- [2] Piotr Kokorczak | Head of International Relations Office “Opole University of technology”, site consulté le 1/04/2016.
- [3] https://en.wikipedia.org/wiki/Opole_University_of_Technology, site consulté le 1/04/2016
- [4] <http://www.po.opole.pl/en/>, site consulté le 1/04/2016
- [5] Mem.gov.ma « Les Energies renouvelables au maroc : stratégie et plan d’action », consulter le 30 Mars 2016.
- [6] https://energypedia.info/wiki/Morocco_Energy_Situation, consulté le 30 Mars 2016.
- [7] <http://mobile.leconomiste.com/article/984037-noor-ouarzazatece-que-sera-la-place-du-maroc-au-soleil>, consulté le 30 Mars 2016.
- [8] MASEN « Les énergies renouvelables, un secteur d’avenir » consulté le 30 Mars 2016
- [9] <http://www.power-technology.com/projects/noor-ouarzazate-solar-complex/> consulté le 28 Mars 2016.
- [10] www.ti.com « MSP430G2x53 » consulté le 10 Mai 2016.
- [11] <https://cnx.org/contents/crSsWGqo@2/Code-Composer-Studio-v4-DSPBIO>, consulté le 20 Avril 2016.
- [12] <https://fr.wikipedia.org/wiki/AutoCAD>, consulté le 20 Mars 2016.
- [13] Mohammed Abdul Rahman Uzair, Mohammed Mohiuddin, Mohammed Khaja Shujaiddin « Failure Analysis of Power Transformers », International Journal of Emerging Technology and Advanced Engineering septembre 2013.
- [14] AMBUJ KUMAR « Root-Cause Analysis of Transformer Failure Scenario at Power Sub-Station », Advances in Environmental and Agricultural Science, ISBN: 978-1-61804-270-5, Octobre 2012.
- [15] <http://www.electrical4u.com/transformer-insulating-oil-and-types-of-transformer-oil/> consulté le 25 Mars 2016.
- [16] <http://b.l.free.fr/Page5.html>, consulté le 28 Mars 2016.
- [17] <http://www.ti.com/tool/tmdsdsk6713>, consulté le 28 Mars 2016.
- [18] <http://z500.pl/>, consulté le 25 Mars 2016.

[19] <http://www.midel.com/products/midel-7131/thermal-properties/kinematic-viscosity>, consulté le 18 Avril 2016.

[20] <http://www.midel.com/products/midel-7131/thermal-properties/density>, consulté le 18 Avril 2016.

[21] <https://itechnofrance.wordpress.com/2013/03/12/utilisation-du-module-ultrason-hc-sr04-avec-larduino/> consulté le 15 Mai 2016.

[22] <http://tronixstuff.com/2014/11/25/tutorial-l298n-dual-motor-controller-modules-and-arduino/> consulté le 16 Mai 2016.

Annexe A

Microprocesseur MSP 430

Cahier de charge N°3: Rédaction d'un programme qui assure par une communication UART de faire clignoter P1.0 si UCA0RXBUF reçoit la lettre « c », P1.0 s'arrête de clignoter si UCA0RXBUF reçoit « de », la led devient plus lumineuse si UCA0RXBUF recoit « f », la led devient moins lumineuse si UCA0RXBUF recoit « g », et finalement pour la lettre « h » et la led clignote afficher *"sara_blinking_%d"* avec %d le taux de clignotement, si la led est éteinte afficher *"sara_notblinking_%d"*

```
#include "msp430g2553.h"
#include "stdio.h"
#define PER 50000
#define COUNT 10

//*****
// Global data
//*****
char string1[50];
int counter=COUNT;
unsigned int i;
unsigned int blink=0;
unsigned int true=0;
unsigned int brightness;
// Function prototypes
void UARTSetup(void);
// Main routine
void main(void)
{
    WDTCTL = WDTPW | WDTHOLD;    // Stop WDT
    //----- Ports -----
    P1DIR |= 0x41;    //P1.0 and P1.6 as digital outputs - LED1 and LED2
    P1DIR &= ~0x08;    //P1.3 as input - SW1

    P1SEL = BIT1 | BIT2 | BIT6;    // P1.1 = RXD, P1.2=TXD
    P1SEL2 = BIT1 | BIT2 ;    // P1.1 = RXD, P1.2=TXD

    P1REN |= 0x08;    //Enable PULL UP/DOWN resistor
    P1OUT |= 0x08;    //PULL UP resistor set

    //----- Basic Clock -----

    if (CALBC1_1MHZ == 0xFF || CALDCO_1MHZ == 0xFF)
    {
        while(1);    // If calibration constants erased
                     // do not load, trap CPU!!
    }
    //1Mhz
    BCSCTL1 = CALBC1_1MHZ;    // Set range
```

```

DCOCTL = CALDCO_1MHZ;                                // Set DCO step + modulation */

//LFXTAL
// BCSCCTL1 |= DIVA_0;
// BCSCCTL2 = SELM_2 | DIVM_0;
// BCSCCTL3 = LFXT1S_0 | XCAP_1;
//blinking initial configuration
P1DIR |= 0x09; //xxxx 1xx1
P1DIR &= ~0x08; //xxxx 0xx1
P1REN |= 0x08;
P1OUT |= 0x08;

P1OUT &= ~0x09;
P1OUT |= 0x08;

P1IFG = 0x00;
P1IES |= 0x08;
P1IE |= 0x08;
//----- Timer A -----
TA0CTL = TASSEL_2 | MC_1 | ID_0; // SMCLK, up mode
CCTL0 = CCIE; // CCR0 interrupt enabled
CCR0 = PER;
CCTL1 = OUTMOD_6; // CCR1 int. enabled
CCR1 = 25000; //HIGH_STATE_INI;
//----- USCI -----
UARTSetup();

__bis_SR_register(LPM1_bits + GIE); // Enter LPM3 w/ int until Byte RXed
}

//*****
// Interrupt service routine:
//*****
#pragma vector=TIMER0_A0_VECTOR
__interrupt void Timer_A_CC0 (void)
{
    if(blink==1){
        counter--;
        if(counter == 0) { // && counter1==0){
            P1OUT ^= 0x01; // overflow
            counter=COUNT;
        }
    }
}

#pragma vector=USCIAB0TX_VECTOR
__interrupt void USCI0TX_ISR(void)
{
    UCA0TXBUF = string1[i++]; // TX next character

    if (i == sizeof string1 - 1){ // TX over?
        IE2 &= ~UCA0TXIE; // Disable USCI_A0 TX interrupt
    }
}

#pragma vector=USCIAB0RX_VECTOR
__interrupt void USCI0RX_ISR(void)
{
    UCA0TXBUF=UCA0RXBUF;
    if (UCA0RXBUF == 'c') // 'u' received?

```

```

{
    blink=1;
    //i = 0;
    //IE2 |= UCA0TXIE;                // Enable USCI_A0 TX interrupt
    //UCA0TXBUF = string1[i++];
}
else if (UCA0RXBUF == 'd')
    true=1;
else if((UCA0RXBUF == 'e') && (true=1))
{
    blink=0;
}
else if (UCA0RXBUF == 'f') {
    CCR1=0;
    CCR1 = CCR1 + 5000; //HIGH_STATE_INI;
    if (CCR1==50000)
    {
        CCR1 = 0 ;
    }
}
else if (UCA0RXBUF == 'g') {
    CCR1=50000;
    CCR1 = CCR1 - 5000; //HIGH_STATE_INI;
    if (CCR1==0)
    {
        CCR1 = 50000 ;
    }
}

else (UCA0RXBUF == 'h'){
    brightness=(CCR1/500);
    if(blink==0){
        sprintf(string1,"Sara_not_blinking_%d",brightness);
    }
    else if(blink==1){
        sprintf(string1,"sara_blinking_%d",brightness);
    }
}

    i = 0;
    IE2 |= UCA0TXIE;                // Enable USCI_A0 TX interrupt
    UCA0TXBUF = string1[i++];
}
}

//*****
//      UARTSetup
// *****
void UARTSetup(void)
{
    //1) -> initialization/re-configuration process <-BEGIN
    UCA0CTL1 |= UCSWRST; //Set UCSWRST -- needed for re-configuration process
    //1) END

    //2) -> Initialize all USCI registersSet <- BEGIN
    // CONTROL REGISTERS

```

```

//UCA0CTL0 -> Control Register 0
// 7 | 6 | 5 | 4 | 3 | 2-1 | 0 |
//-----
//UCPEN|UCPAR|UCMSB|UC7BIT|UCSPB|UCMODEx|UCSYNC|
//UCPEN (Parity) = 0b -> Parity disabled
//UCPAR (Parity select) = 0b -> Odd parity "Not importante"
//UCMSB (MSB first select) = 0b -> LSB first
//UC7BIT (Character length) = 0b -> 8-bit data
//UCSPB (Stop bit select) = 0b -> One stop bit
//UCMODEx (USCI mode) = 00b -> UART Mode
//UCSYNC (Synchronous mode enable) = 0b -> Asynchronous mode
UCA0CTL0 = 0x00;
//-----
//-----
//UCA0CTL1 -> Control Register 1
// 6-7 | 5 | 4 | 3 | 2 | 1 | 0 |
//-----
//UCSSELx |UCRXEIE|UCBRKIE|UCDORM|UCTXADDR|UCTXBRK|UCSWRST|
//-----
//UCSSELx (USCI clock source select)= 01b -> ACLK 10b -> SMCLK
//UCRXEIE (interrupt-enable) = 0b -> Erroneous characters rejected
//UCBRKIE (interrupt-enable) = 0b -> Received break characters set
//UCDORM (sleep mode) = 0b -> Not dormant
//UCTXADDR (Transmit address) = 0b -> Next frame transmitted is data
//UCTXBRK (Transmit break) = 0b -> Next frame transmitted is not a
break
//UCSWRST (Software reset) = 1b -> normally Set by a PUC
UCA0CTL1 = 0x81;
//-----
// BAUD RATE GENERATION
// Prescaler N = 1MHz/(9600) = 104.16
// UCA0BRx = INT(N/16) = 6
// UCBRFx = round( ( N/16 - INT(N/16) ) x 16 ) = 8
//-----
UCA0BR0 = 6;
UCA0BR1 = 0x00;
//UCAxMCTL Modulation Control Register
// 7-4 | 3-1 | 0 |
//UCBRFx|UCBRSx|UCOS16|
//UCBRFx (First modulation stage) = 1000b -> from table 19-4
//UCBRSx (Second modulation stage) = 000b -> from table 19-4
//UCOS16 (Oversampling mode) = 1b -> Enabled
UCA0MCTL = UCBRF_8 | UCOS16;

//-----
//2) END

//3) Clear UCSWRST via software -> BEGIN
UCA0CTL1 &= ~UCSWRST; // **Initialize USCI state machine**
//3) ->END

//4) Enable interrupts -> BEGIN
IE2 |= UCA0RXIE; // Enable USCI_A0 RX interrupt
//4) -> END
}

```


Annexe B

Digital Signal Processing (DSP)

Cahier de charge 3 :Le programme ci dessous recueille 44 échantillons à partir du canal gauche et droit du signal audio provenant de la carte son de l'ordinateur et filtre le canal gauche

Modifier le programme afin qu'il permet de fileter les deux canaux du signal

// Les bibliothèques et les fonctions de DSK6713, codec audio, commutateurs et les LEDS

```
#include "C6713_DSK_FIRcfg.h"
```

```
#define CHIP_6713 1
```

```
#include "dsk6713.h"
```

```
#include "dsk6713_aic23.h"
```

```
#include "dsk6713_dip.h"
```

```
#include "dsk6713_led.h"
```

```
short FILTR_L (short, short*) // la déclaration d'une fonction qui renvoie une valeur courte  
en recevant deux arguments
```

```
/* configuration de codec */
```

```
DSK6713_AIC23_Config config = { // normal mode line in , mic off
```

```
0x0017, // 0 DSK6713_AIC23_LEFTINVOL volume d'entrée du canal gauche
```

```
0x0017, // 1 DSK6713_AIC23_RIGHTINVOL volume d'entrée du canal droit
```

```
0x00F9, // 2 DSK6713_AIC23_LEFTHPVOL volume de casque du canal gauche
```

```
0x00F9, // 3 DSK6713_AIC23_RIGHTHPVOL volume de casque du canal droit
```

```
0x0012, // 4 DSK6713_AIC23_ANAPATH contrôler la piste audio analogique
```

```
0x0000, // 5 DSK6713_AIC23_DIGPATH contrôler la piste audio numérique
```

```
0x0002, // 6 DSK6713_AIC23_POWERDOWN les dispositifs de commande sont activés
```

```
0x0043, // 7 DSK6713_AIC23_DIGIF format d'interface audio numérique
```

```
0x0023, // 8 DSK6713_AIC23_SAMPLERATE contrôle la fréquence d'échantillonnage
```

```
0x0001, // 9 DSK6713_AIC23_DIGACT l'activation de l'interface numérique
```

```
} ;
```

```
Short L_in[ 44 ] ; // déclaration d'un tableau pour recueillir les échantillons du canal gauche
```

```
Short h1 [ ] =
```

```
{ -77, -226 , -99 , 329, 533, 108, -327, -76, 207, -483, -1293, -504, 1214, 1407, 217, 448,  
1903, 335, -4825, -6690, -275, 8154, 8154, -275, -6690, -4825, 335, 1903, 448, 217, 1407,  
1214, -504, -1293, -483, 207, -76, -327, 108, 533, 329, -99,-226,-77} ; // déclaration des  
coefficients de filtre
```

```
Void main ( )
```

```
{
```

```
DSK6713_AIC23_CodecHandle hCodec ;
```

```
Int16 OUT_L , OUT_R ;
```

```
Unit32 IN_L , IN_R;
```

```
/* initialisation des bibliothèques */
```

```
DSK6713_init ( ) ;
```

```
DSK6713_LED_init ( ) ;
```

```
DSK6713_DIP_init ( ) ;
```

```
// Configuration des paramètres de codec
```

```
hCodec = DSK6713_AIC23_openCodec ( 0 , & config ) ;
```

```
// Définition de la fréquence de Codec 44 Khz
```

```
DSK6713_AIC23_setFreq ( hCodec , DSK6713_AIC23_FREQ_44KHZ ) ;
```

```
While (1)
```

```
{
```

```
// Lire un échantillon de canal gauche
```

```
While ( ! DSK6713_AIC23_read ( hCodec , & IN_L ) ) ;
```

```
// Lire un échantillon de canal droit
```

```
While ( ! DSK6713_AIC23_read ( hCodec , & IN_R ) ) ;
```

```
OUT_L = FILTER_L ( IN_L , h1 ) ;
```

```

OUT_R = IN_R ;

// Envoyer un échantillon de canal gauche
While ( ! DSK6713_AIC23_write ( hCodec , & OUT_L ) ) ;

// Envoyer un échantillon de canal droit
While ( ! DSK6713_AIC23_write ( hCodec , & OUT_R ) ) ;

}

// Désactiver le codec
DSK6713_AIC23_closeCodec ( hCodec ) ;

}

Short FILTR_L ( short input , short *h )
{
    Int i ;
    Short output ;
    Int acc = 0 ;
    Int prod ;
    L_int [ 0 ] = input ;
    acc = 0 ; // remettre l'accumulateur à 0
    for ( i = 0 ; i < 44 ; i++ )
    {
        prod = ( h [ i ] * L_in [ i ] ) ;
        acc = acc + prod ;
    }
    output = ( short ) ( acc >> 15 ) ; // la modification de la valeur de sortie en 16-bits
    for ( i = 43 ; i > 0 ; i-- )
        L_in [ i ] = L_in [ i-1 ] ; // décalage des échantillons
    return output ; // retour des échantillons de filtrés
}

```

Programme de filtrage des deux canaux :

// Les bibliothèques et les fonctions de DSK6713, codec audio, commutateurs et les LEDS

#include "C6713_DSK_FIRcfg.h"

#define CHIP_6713 1

#include "dsk6713.h"

#include "dsk6713_aic23.h"

#include "dsk6713_dip.h"

#include "dsk6713_led.h"

short FILTR_L (short, short*) // la déclaration d'une fonction qui renvoie une valeur courte
en recevant deux arguments

short FILTR_R (short, short*)

/* configuration de codec */

DSK6713_AIC23_Config config = { // normal mode line in , mic off

0x0017, // 0 DSK6713_AIC23_LEFTINVOL volume d'entrée du canal gauche

0x0017, // 1 DSK6713_AIC23_RIGHTINVOL volume d'entrée du canal droit

0x00F9, // 2 DSK6713_AIC23_LEFTHPVOL volume de casque du canal gauche

0x00F9, // 3 DSK6713_AIC23_RIGHTHPVOL volume de casque du canal droit

0x0012, // 4 DSK6713_AIC23_ANAPATH contrôler la piste audio analogique

0x0000, // 5 DSK6713_AIC23_DIGPATH contrôler la piste audio numérique

0x0002, // 6 DSK6713_AIC23_POWERDOWN les dispositifs de commande sont activés

0x0043, // 7 DSK6713_AIC23_DIGIF format d'interface audio numérique

0x0023, // 8 DSK6713_AIC23_SAMPLERATE contrôle la fréquence d'échantillonnage

0x0001, // 9 DSK6713_AIC23_DIGACT l'activation de l'interface numérique

};

Short L_in[44]; // déclaration d'un tableau pour recueillir les échantillons du canal gauche

Short R_in[44];

Short h1 [] =

```
{ -77, -226 , -99 , 329, 533, 108, -327, -76, 207, -483, -1293, -504, 1214, 1407, 217, 448,
1903, 335, -4825, -6690, -275, 8154, 8154, -275, -6690, -4825, 335, 1903, 448, 217, 1407,
1214, -504, -1293, -483, 207, -76, -327, 108, 533, 329, -99,-226,-77} ; // déclaration des
coefficients de filtre
```

```
Void main ( )
```

```
{
```

```
DSK6713_AIC23_CodecHandle hCodec ;
```

```
Int16 OUT_L , OUT_R ;
```

```
Unit32 IN_L , IN_R;
```

```
/* initialisation des bibliothèques */
```

```
DSK6713_init ( ) ;
```

```
DSK6713_LED_init ( ) ;
```

```
DSK6713_DIP_init ( ) ;
```

```
// Configuration des paramètres de codec
```

```
hCodec = DSK6713_AIC23_openCodec ( 0 , & config ) ;
```

```
// Définition de la fréquence de Codec 44 Khz
```

```
DSK6713_AIC23_setFreq ( hCodec , DSK6713_AIC23_FREQ_44KHZ ) ;
```

```
While (1)
```

```
{
```

```
// Lire un échantillon de canal gauche
```

```
While ( ! DSK6713_AIC23_read ( hCodec , & IN_L ) ) ;
```

```
// Lire un échantillon de canal droit
```

```
While ( ! DSK6713_AIC23_read ( hCodec , & IN_R ) ) ;
```

```
OUT_L = FILTER_L ( IN_L , h1 ) ;
```

```
OUT_R = FILTER_R ( IN_R , h1 ) ;
```

```
// Envoyer un échantillon de canal gauche
```

```
While ( ! DSK6713_AIC23_write ( hCodec , & OUT_L ) ) ;
```

```
// Envoyer un échantillon de canal droit
```

```

While ( ! DSK6713_AIC23_write ( hCodec , & OUT_R ) ) ;

}

// Désactiver le codec

DSK6713_AIC23_closeCodec ( hCodec ) ;

}

Short FILTR_L ( short input , short *h )

{

Int i ;

Short output ;

Int acc = 0 ;

Int prod ;

L_int [ 0 ] = input ;

acc = 0 ; // remettre l'accumulateur à 0

for ( i = 0 ; i < 44 ; i++ )

{

    prod = ( h [ i ] * L_in [ i ] ) ;

    acc = acc + prod ;

}

output = ( short ) ( acc >> 15 ) ; // la modification de la valeur de sortie en 16-bits

for ( i = 43 ; i > 0 ; i-- )

    L_in [ i ] = L_in [ i-1 ] ; // décalage des échantillons

return output ; // retour des échantillons de filtrés

}

Short FILTR_R ( short inputR , short *h )

{

Int k ;

Short outputR ;

```

```

Int accR = 0 ;

Int prodR ;

R_int [ 0 ] = inputR ;

accR = 0 ; // remettre l'accumulateur à 0

for ( k = 0 ; k < 44 ; k++ )

{

    prodR = ( h [k ] * R_in [ k ] ) ;

    accR = accR + prodR ;

}

outputR = ( short ) ( accR >> 15 ) ; // la modification de la valeur de sortie en 16-bits

for ( k = 43 ; k > 0 ; k- - )

    R_in [ i ] = R_in [ k-1 ] ; // décalage des échantillons

return outputR ; // retour des échantillons de filtrés

}

```

Rédiger un programme qui permet de réaliser la fonction d'un filtre coupe bande et qui filtre les deux canaux :

```

// Les bibliothèques et les fonctions de DSK6713, codec audio, commutateurs et les LEDS

#include "C6713_DSK_FIRcfg.h"

#define CHIP_6713 1

#include "dsk6713.h"

#include "dsk6713_aic23.h"

#include "dsk6713_dip.h"

#include "dsk6713_led.h"

short FILTR_L (short, short*) // la déclaration d'une fonction qui renvoie une valeur courte
en recevant deux arguments

short FILTR_R (short, short*)

/* configuration de codec */

DSK6713_AIC23_Config config = { // normal mode line in , mic off

```

```

0x0017, // 0 DSK6713_AIC23_LEFTINVOL volume d'entrée du canal gauche
0x0017, // 1 DSK6713_AIC23_RIGHTINVOL volume d'entrée du canal droit
0x00F9, // 2 DSK6713_AIC23_LEFTHPVOL volume de casque du canal gauche
0x00F9, // 3 DSK6713_AIC23_RIGHTHPVOL volume de casque du canal droit
0x0012, // 4 DSK6713_AIC23_ANAPATH contrôler la piste audio analogique
0x0000, // 5 DSK6713_AIC23_DIGPATH contrôler la piste audio numérique
0x0002, // 6 DSK6713_AIC23_POWERDOWN les dispositifs de commande sont activés
0x0043, // 7 DSK6713_AIC23_DIGIF format d'interface audio numérique
0x0023, // 8 DSK6713_AIC23_SAMPLERATE contrôle la fréquence d'échantillonnage
0x0001, // 9 DSK6713_AIC23_DIGACT l'activation de l'interface numérique
} ;

Short L_in[ 100 ] ; // déclaration d'un tableau pour recueillir les échantillons du canal
gauche

Short R_in[ 100 ] ;

Short h1 [ ] =
{ 0 , 0 , 0 , -1 , -3 , -3 , 0 , 6 , 11 , 8 , -6 , -33 , -61 , -79 , -76 , -50 , -17 , 0 , -22 , -86 , -173 , -
241 , -250 , -179 , -48 , 88 , 162 , 130 , 0 , -158 , -240 , -159 , 106 , 482 , 827 , 986 , 877 , 548 ,
174 , 0 , 219 , 866 , 1759 , 2534 , 2770 , 2160 , 659 , -1455 , -3648 , -5296 , 26586 , -5296 , -
3648 , -1455 , 659 , 2160 , 2770 , 2534 , 1759 , 866 , 219 , 0 , 174 , 548 , 877 , 986 , 827 ,
482 , 106 , -159 , -240 , -158 , 0 , 130 , 162 , 88 , -48 , -179 , -250 , -241 , -173 , -86 , -22 , 0
, -17 , -50 , -76 , -79 , -61 , -33 , -6 , 8 , 11 , 6 , 0 , -3 , -3 , -1 , 0 , 0 , 0 } ; // déclaration des
coefficients de filtre

Void main ( )
{
DSK6713_AIC23_CodecHandle hCodec ;

Int16 OUT_L , OUT_R ;

Unit32 IN_L , IN_R ;

/* initialisation des bibliothèques */

DSK6713_init ( ) ;

```



```

DSK6713_LED_init ( ) ;

DSK6713_DIP_init ( ) ;

// Configuration des paramètres de codec

hCodec = DSK6713_AIC23_openCodec ( 0 , & config ) ;

// Définition de la fréquence de Codec 44 Khz

DSK6713_AIC23_setFreq ( hCodec , DSK6713_AIC23_FREQ_44KHZ ) ;

While (1)

{

// Lire un échantillon de canal gauche

While ( ! DSK6713_AIC23_read ( hCodec , & IN_L ) ) ;

// Lire un échantillon de canal droit

While ( ! DSK6713_AIC23_read ( hCodec , & IN_R ) ) ;

OUT_L = FILTER_L ( IN_L , h1 ) ;

OUT_R = FILTER_R ( IN_R , h1 ) ;

// Envoyer un échantillon de canal gauche

While ( ! DSK6713_AIC23_write ( hCodec , & OUT_L ) ) ;

// Envoyer un échantillon de canal droit

While ( ! DSK6713_AIC23_write ( hCodec , & OUT_R ) ) ;

}

// Désactiver le codec

DSK6713_AIC23_closeCodec ( hCodec ) ;

}

Short FILTR_L ( short input , short *h )

{

Int i ;

Short output ;

Int acc = 0 ;

```

```

Int prod ;

L_int [ 0 ] = input ;

acc = 0 ; // remettre l'accumulateur à 0

for ( i = 0 ; i < 100 ; i++ )

{

    prod = ( h [ i ] * L_in [ i ] ) ;

    acc = acc + prod ;

}

output = ( short ) ( acc >> 15 ) ; // la modification de la valeur de sortie en 16-bits

for ( i = 101 ; i > 0 ; i- - )

    L_in [ i ] = L_in [ i-1 ] ; // décalage des échantillons

return output ; // retour des échantillons de filtrés

}

Short FILTR_R ( short inputR , short *h )

{

Int k ;

Short outputR ;

Int accR = 0 ;

Int prodR ;

R_int [ 0 ] = inputR ;

accR = 0 ; // remettre l'accumulateur à 0

for ( k = 0 ; k < 100 ; k++ )

{

    prodR = ( h [k] * R_in [ k ] ) ;

    accR = accR + prodR ;

}

outputR = ( short ) ( accR >> 15 ) ; // la modification de la valeur de sortie en 16-bits

```

```
for ( k = 101 ; k > 0 ; k- - )  
    R_in [ i ] = R_in [ k-1 ] ; // décalage des échantillons  
return outputR ; // retour des échantillons de filtrés  
}
```

Annexe C

I. Programme du Robot détecteur d'obstacle

```
const int echoPin=9;
const int trigPin=10;
const int ledgreen=11;
const int ledred=12;
const int buzzer=13;
const int Motor1Pin1 = 8;
const int Motor1Pin2 = 7;
const int Motor2Pin2 =4;
const int Motor2Pin1 = 2;
int speedmotor1=5;
int speedmotor2=6;
int distance;
int temps;
void setup() {
  Serial.begin(9600);
  pinMode(Motor1Pin1, OUTPUT);
  pinMode(Motor1Pin2, OUTPUT);
  pinMode(Motor2Pin1, OUTPUT);
  pinMode(Motor2Pin2, OUTPUT);
  pinMode(speedmotor1, OUTPUT);
  pinMode(speedmotor2, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(ledgreen, OUTPUT);
  pinMode(ledred, OUTPUT);
  pinMode(buzzer, OUTPUT);
```

```

pinMode(speedmotor1, OUTPUT)
pinMode(speedmotor2, OUTPUT); }

void loop()
{
digitalWrite(trigPin,LOW);
delayMicroseconds(2);
digitalWrite(trigPin,HIGH);
delayMicroseconds(10);
digitalWrite(trigPin,LOW);
temps=pulseIn(echoPin,HIGH);
distance=temps/58;
if (distance >= 30){
digitalWrite(ledgreen,HIGH);
digitalWrite(ledred,LOW);
digitalWrite(buzzer,LOW);
}
else {
digitalWrite(ledgreen,LOW);
digitalWrite(ledred,HIGH);
digitalWrite(buzzer,HIGH);
MoveBackward();
MoveRight();
MoveForward();
}
}

void MoveForward(){
digitalWrite(Motor1Pin1, HIGH);
digitalWrite(Motor1Pin2, LOW);
analogWrite(speedmotor1,80);

```

```

digitalWrite(Motor2Pin1, HIGH);
digitalWrite(Motor2Pin2, LOW);
analogWrite(speedmotor2,80);
}

void MoveBackward(){
    digitalWrite(Motor1Pin1, LOW);
    digitalWrite(Motor1Pin2, HIGH);
    analogWrite(speedmotor1,80);
    digitalWrite(Motor2Pin1, LOW);
    digitalWrite(Motor2Pin2, HIGH);
    analogWrite(speedmotor2,80);
    delay(1000);
}

void Stop(){
    digitalWrite(Motor1Pin1, LOW);
    digitalWrite(Motor1Pin2, LOW);
    digitalWrite(Motor2Pin1, LOW);
    digitalWrite(Motor2Pin2, LOW);
}

void MoveRight(){
    digitalWrite(Motor1Pin1,LOW);
    digitalWrite(Motor1Pin2, LOW);
    analogWrite(speedmotor1,80);
    digitalWrite(Motor2Pin1, HIGH);
    digitalWrite(Motor2Pin2, LOW);
    analogWrite(speedmotor2,80);
    delay(500);
}

```

Annexe D

II. Programme du robot contrôlé par Bluetooth

```
#include <SoftwareSerial.h>

int speedmotor1=5;

int speedmotor2=6;

SoftwareSerial BT(10, 11); //TX, RX respetively

String readdata;


void setup() {
  BT.begin(9600);
  Serial.begin(9600);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(speedmotor1, OUTPUT);
  pinMode(speedmotor2, OUTPUT);
}

//-----//

void loop() {
  while (BT.available()){ //Check if there is an available byte to read
    delay(10); //Delay added to make thing stable
    char c = BT.read(); //Conduct a serial read
    readdata += c; //build the string- "forward", "reverse", "left" and "right"
  }
  if (readdata.length() > 0) {
    Serial.println(readdata);
    if(readdata == "HAUT")
    {
      digitalWrite(3, HIGH);
    }
  }
}
```

```

digitalWrite (4,LOW);
digitalWrite(5,HIGH);
digitalWrite(6,LOW);
delay(100);
}
else if(readdata == "BAS")
{
digitalWrite(3, LOW);
digitalWrite(4, HIGH);
digitalWrite(5, LOW);
digitalWrite(6,HIGH);
delay(100);
}
else if (readdata == "DROITE")
{
digitalWrite (3,HIGH);
digitalWrite (4,LOW);
digitalWrite (5,LOW);
digitalWrite (6,LOW);
delay (100);
}
else if ( readdata == "GAUCHE")
{
digitalWrite (3, LOW);
digitalWrite (4, HIGH);
digitalWrite (5, LOW);
digitalWrite (6, LOW);
delay (100);
}

```



```
}  
else if (readdata == "PAUSE")  
{  
    digitalWrite (3, LOW);  
    digitalWrite (4, LOW);  
    digitalWrite (5, LOW);  
    digitalWrite (6, LOW);  
    delay (100);  
}
```