

# INITIATION A L'INFORMATIQUE EMBARQUÉE : MICROPROCESSEUR/MICROCONTRÔLEUR ET OUTILS LOGICIELS

**Étudiants :****Atef RGHIOUI****Bénédicte LETERME****Mélodie NONNON****Baptiste LECHAT****Mouad LEMHADARE****Saad LAKRAMTI**

---

Date de remise du rapport : **11/06/2022**

Référence du projet : **STPI/P6/2022 – 046**

Intitulé du projet :

***Initiation à l'informatique embarquée : microprocesseur/microcontrôleur et outils logiciels***

Type de projet : ***informatique, électronique***

Objectifs du projet :

- ***Acquérir des compétences en informatique embarquée :***
  - ★ ***Se familiariser avec l'électronique***
  - ★ ***Acquérir les bases du langage C***
  - ★ ***Comprendre le fonctionnement d'une carte Explorer 16/32***
- ***Réaliser un robot serveur télécommandé à l'aide de notre téléphone :***
  - ★ ***Gérer le contrôle des moteurs***
  - ★ ***Gérer le signal bluetooth pour télécommander le robot***
  - ★ ***Faire éviter les obstacles au robot à l'aide d'ultrasons***
  - ★ ***Tout relier pour avoir un robot fonctionnel***

Mots-clefs du projet : ***Robot télécommandé à distance.***

## REMERCIEMENTS

Tout d'abord, nous tenons à remercier l'INSA de nous avoir introduit à la gestion de projets dès la deuxième année de STPI, grâce à non seulement le projet de P6 mais aussi les projets en mathématiques et informatiques du premier semestre.

Ensuite, nous souhaitons remercier M. Richard Grisel d'une part, pour son encadrement de qualité ainsi que pour son aide. D'autre part, d'avoir proposé un sujet intéressant qui se distingue significativement des autres, beaucoup plus orientés physique.

Enfin, nous souhaitons aussi remercier Mme Hélène Rade de nous avoir fourni les composants nécessaires, ainsi que pour son aide à construire notre robot.



## TABLE DES MATIÈRES

<b>Introduction</b>	<b>6</b>
<b>Méthodologie / Organisation du travail</b>	<b>7</b>
2.1 Le déroulement	7
2.2 Le matériel utilisé	8
<b>Travail réalisé et résultats</b>	<b>9</b>
3.1 Branchements et contrôle du moteur	9
3.1.1. Branchements et circuits des moteurs	9
a- Le circuit électrique	9
b- Le branchement	10
c- Le code	11
3.2 Contrôle du robot	12
3.2.1 Les branchements	12
3.2.2 Le code	13
3.3 La conception du robot	13
3.4 Capteur à ultrasons	14
3.4.1 Intérêt d'un capteur à ultrasons et choix du capteur	14
3.4.2 Principe de Fonctionnement	14
3.4.3 Mise en fonctionnement du capteur	15
3.4.4 Expérience et bilan	16
<b>Conclusions et perspectives</b>	<b>17</b>
<b>Bibliographie</b>	<b>18</b>

## NOTATIONS, ACRONYMES

**Broche** : Pièce métallique, destinée à en assurer la connexion électrique et éventuellement la fixation sur la carte électronique.

**CC** : Courant Continu

**Signal PWM**: Pulse Width Modulation en anglais, il s'agit d'un signal dont le rapport cyclique varie.

**BLE** : Bluetooth Low Energy en anglais, il s'agit d'une technologie Bluetooth à faible consommation d'énergie.

**GND** : Signifie Ground en anglais et fait référence à la masse électrique.

**VCC** : Voltage Common Collector en anglais, fait référence à la tension d'alimentation.

## 1. INTRODUCTION

Au cours du projet P6 qui est un enseignement de projet en physique faisant partie des cours du quatrième semestre à l'INSA, nous avons abordé par groupe de 6 l'Informatique Embarquée. Étant tous passionnés par l'informatique nous avons choisi ce sujet afin de découvrir une autre facette de ce domaine très intéressant. Effectivement, nous avons suivi depuis notre arrivée à l'INSA plusieurs cours de programmation et d'algorithmique mais nous n'avions jamais étudié l'informatique embarquée. Ainsi, nous avons découvert, grâce à ce projet, plusieurs notions fondamentales en informatique notamment l'architecture 32-bit, la gestion de mémoire et le langage assembleur. Ce travail a également été une opportunité pour nous de nous familiariser avec l'électronique, une matière qui avait toute son importance au sein de ce projet.

Concrètement, nous avons travaillé sur la construction et la programmation d'un petit robot télécommandé. Notre robot est contrôlé par téléphone grâce à un signal Bluetooth et est également capable de détecter des obstacles grâce à un signal ultrason. Nous avons rencontré plusieurs difficultés, principalement techniques, au cours de la réalisation de notre projet mais nous avons pu les surmonter grâce à l'aide de notre encadrant M. Grisel ainsi que de nos recherches personnelles.

Ce projet enrichissant nous a permis d'une part d'acquérir de nouvelles connaissances dans une branche de l'informatique que l'on ne connaissait pas. D'autre part, de développer des compétences en travail d'équipe et en gestion de projet qui nous seront très utiles non seulement pour la suite de nos études mais également dans le monde professionnel.

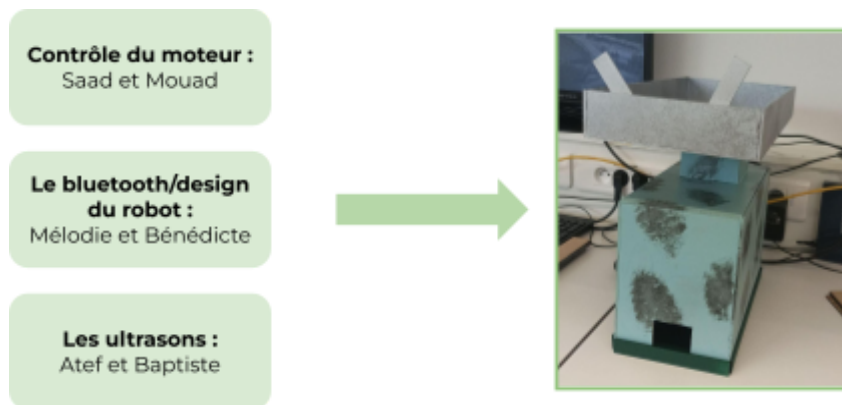
## 2. MÉTHODOLOGIE / ORGANISATION DU TRAVAIL

### 2.1 LE DÉROULEMENT

Le projet s'est déroulé en deux phases. Dans un premier temps, lors des 5 premières séances, nous sommes passés par une phase d'initiation au cours de laquelle nous avons assisté tous ensemble à des cours théoriques d'introduction à l'architecture des ordinateurs, au langage C et au langage assembleur. Ces cours nous ont permis d'acquérir les notions élémentaires qui nous ont permis d'avancer vers des applications pratiques. On a appris à brancher la carte à l'ordinateur et à manipuler l'éditeur de texte MPLab afin d'injecter notre programme sur la carte. Une des applications pratiques que l'on a effectuées consistait à faire clignoter des LEDs grâce à un timer.

Dans un second temps, une fois que nous avons abordé les notions théoriques de base et appris à manipuler le matériel, nous sommes passés à la partie principale du projet. Nous nous sommes mis en groupe de 6 et nous avons entamé une réflexion autour d'un projet d'informatique embarquée que l'on aimerait réaliser et qui surtout était réalisable dans la période de temps qui nous restait. Une fois que notre idée projet avait été validée par M. Grisel nous avons commencé à travailler sur sa réalisation.

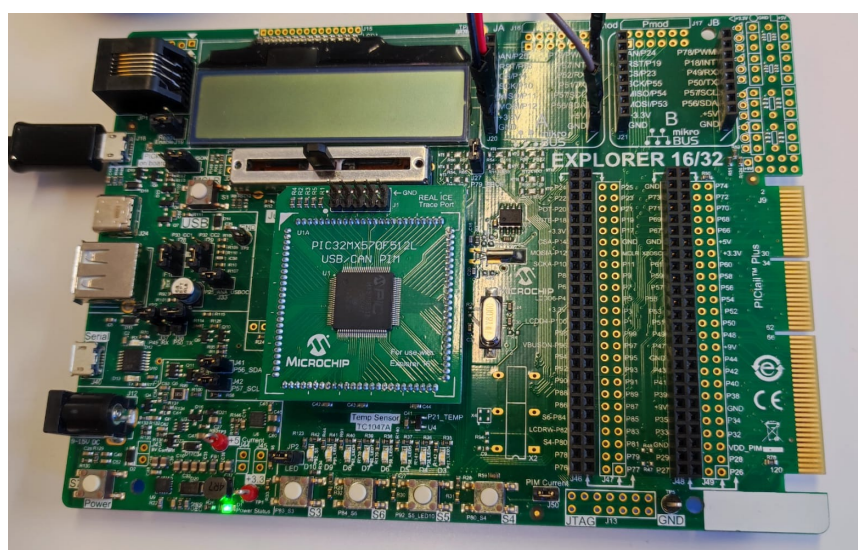
L'idée que nous avons eu était de faire un robot télécommandé, contrôlé par téléphone grâce à un signal Bluetooth et également capable de détecter des obstacles grâce à des ondes ultrasonores. Notre robot consiste essentiellement en une carte électronique mobile à laquelle on a ajouté un capteur Bluetooth et un capteur ultrason, le tout alimenté par des piles. Les deux principaux enjeux du projet étaient d'une part de construire le robot : mise en lien des composants électroniques, compréhension et branchement du circuit électrique et conception et design du corps. D'autre part, la programmation du robot : vitesse et direction du moteur et interprétation et traitement des signaux. Pour mener à bien ce projet nous nous sommes répartis les tâches en fonction des compétences et des préférences de chacun. Finalement, nous avons divisé notre groupe de 6 en 3 groupes de 2. Mélodie et Bénédicte ont travaillé sur la conception et le design du corps du robot, ainsi que le traitement du signal bluetooth, Atef et Baptiste se sont chargés du traitement du signal ultrason et enfin Saad et Mouad ont travaillé sur le contrôle du moteur.



## 2.2 LE MATÉRIEL UTILISÉ

Une carte électronique programmable est un support plan possédant des pistes électriques qui permettent la mise en relation de plusieurs composants électroniques. Comme son nom l'indique, la carte électronique programmable est capable d'exécuter un programme informatique. Les principaux composants indispensables au fonctionnement et donc à la programmabilité de la carte sont un microcontrôleur ou une unité centrale composée d'un microprocesseur, de la mémoire (vive et morte), et des ports d'entrée et de sortie. Sans programme, la carte programmable ne peut pas fonctionner, elle doit donc être reliée à un ordinateur pour lui injecter un programme.

Lors de ce projet, nous utiliserons la carte " Microchip Explorer 16/32 " Development Board" de la marque Microchip Technology. C'est une carte dont le microprocesseur possède une architecture 32-bit, c'est-à-dire que son registre (espace mémoire) est de largeur 32-bit et qu'il peut traiter n'importe quelle donnée et utiliser des adresses mémoire représentées en 32 bits. Nous utiliserons le langage C afin de programmer notre carte.



*Photo de la carte Microchip Explorer 16/32*

### 3. TRAVAIL RÉALISÉ ET RÉSULTATS

#### 3.1. BRANCHEMENTS ET CONTRÔLE DU MOTEUR

Afin de faire tourner les deux roues de notre robot nous disposons de deux moteurs que nous souhaitons faire tourner dans les deux sens et dont nous souhaitons faire varier la vitesse. Dans cette partie nous détaillerons d'une part comment nous allons effectuer les branchements afin de mettre en lien les moteurs avec la carte. D'autre part, nous porterons notre attention sur la logique derrière le programme informatique qui nous permet de contrôler la vitesse et le sens de rotation des moteurs.

##### 3.1.1. Branchements et circuits des moteurs

Les deux moteurs que l'on utilisera sont des moteurs à courant continu directement liés aux roues. Puisque nous cherchons à faire tourner chaque moteur dans les deux sens et qu'il se pourrait que nos deux moteurs tournent simultanément dans des sens différents nous placerons nos moteurs dans deux circuits électriques identiques mais indépendants.

#### A- LE CIRCUIT ÉLECTRIQUE

Afin de faire varier le sens de rotation d'un moteur, il faut varier le sens du courant qui le parcourt. De ce fait, nous utiliserons un circuit H-bridge ou pont en H qui permet de modifier le sens du courant grâce à 4 interrupteurs.

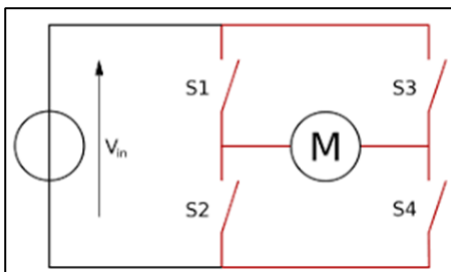


Figure 1: Schéma du pont en H.

Si on ferme les interrupteurs S1 et S4 et qu'on ouvre S2 et S3 alors le moteur tourne dans le sens horaire.

Si on ouvre les interrupteurs S1 et S4 et qu'on ferme S2 et S3 alors le moteur tourne dans le sens trigonométrique.

Lors de ce projet, la composante électronique qui fera office de pont en H sera PmodHb5.



*Figure 2: Photo de la pièce  
Pmod Hb5*

## B- LE BRANCHEMENT

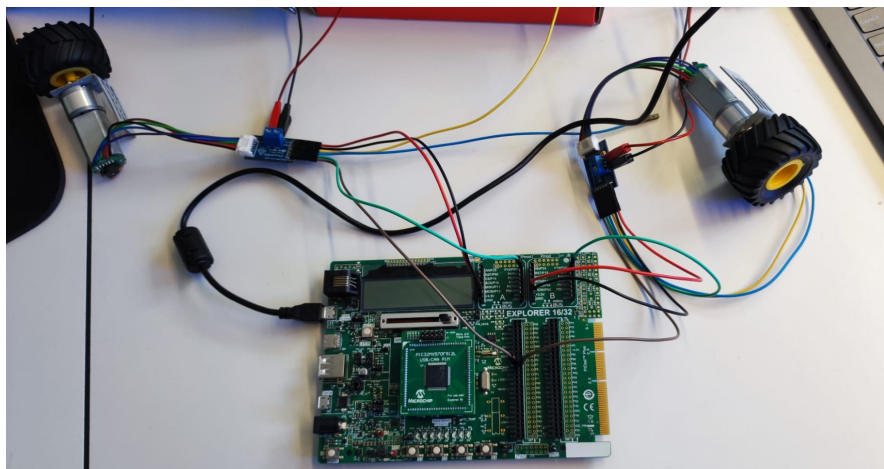
PmodHb5 sert de liaison entre le moteur et la carte électronique, nous utiliserons donc 2 Pmod Hb5, un pour chaque moteur. C'est une pièce qui comporte 6 ports, ceux que nous utiliserons sont : Dir, EN, GND et VCC. Dir et EN sont respectivement les ports qui permettent de contrôler le sens de rotation et la vitesse de rotation du moteur et GND et VCC font respectivement office de masse et d'alimentation électrique. Nous détaillerons le branchement d'un seul moteur puisque nos deux moteurs sont branchés exactement de la même manière. Nous utiliserons des câbles colorés afin de brancher le moteur au PmodHb5 et ensuite le PmodHb5 a la carte et au générateur.

La broche de vitesse EN doit être branchée sur un port PWM de notre carte, dans ce cas soit P72 soit P77. Ce port sert à fournir un signal PWM au moteur afin de faire varier sa vitesse de rotation. Il s'agit d'un signal de fréquence constante et de rapport cyclique variable, et plus le rapport est élevé, plus le moteur tourne vite.

photo

Ensuite, on branche la broche de direction Dir sur un port de sortie, nous avons choisi le port C donc les sorties P6 et P7.

Enfin, on branche la masse GND et l'alimentation VCC aux ports du même nom sur la carte



*Figure 3: Photo du branchement des  
moteurs avec la carte.*



## C- LE CODE

Le code est écrit en langage C embarqué. Il gère les deux moteurs aussi bien que d'autres composantes utilisées pour le fonctionnement du robot ( capteur ultrason, récepteur signal bluetooth). Ce code sert à gérer :

- La vitesse d'avancement du robot.
- Le changement d'orientation (avant, arrière, droite, gauche).
- Récupération du signal envoyé par le téléphone par Bluetooth.
- Interpréter ce signal et l'utiliser pour orienter le robot.
- Interpréter les données fournies par l'ultrason principalement la distance entre le robot et un obstacle.
- Utiliser cette distance comme seuil pour éviter les obstacles ( remettre la vitesse à 0 pour arrêter le Robot ).

Toutes ces fonctionnalités semblent très compliquées à coder au début car il faut gérer les broches en entrée, les broches en sortie, aussi bien que les signaux captés par l'ultrason et le BLE, qu'on doit finalement traduire ce signal pour en déduire la direction ainsi que la distance entre le robot et les obstacles. Le code qu'on a créé est finalement une boucle qui se répète et qui va récupérer à la fois le signal du BLE et celui de l'ultrason et puis les traduire afin de les interpréter, avant de rentrer dans une boucle conditionnelle qui va gérer chaque cas en particulier.

On va contrôler le robot grâce à une application connectée par Bluetooth à notre carte. L'application récupère les appuis sur des touches de directions puis envoie un signal qu'on va ensuite traduire pour pouvoir l'interpréter selon les situations. On a choisi de traduire les signaux en caractères numériques 0 , 1 , 2 , 3 ... Lorsque l'utilisateur de l'application appuie sur un bouton un premier signal est envoyé par bluetooth, une fois que l'utilisateur n'appuie plus sur le bouton un signal différent est envoyé indiquant la fin d'appui.

Le cas '**en avant**' : il faut vérifier que la distance récupérée par l'ultrason est inférieure à une distance de sécurité choisie (**10cm pour laisser le temps de freinage**) . Si c'est le cas, on fixe la direction vers l'avant et on attend pour un délai de 5 ms puis on fait tourner les roues à une vitesse donnée par le rapport cyclique  $OC1RS = OC3RS = 1800$  . Sinon on arrête le robot , et donc le rapport cyclique sera  $OC1RS = OC3RS = 0$ .

Le cas '**en arrière**' : malheureusement on ne possède pas un deuxième capteur ultrason pour vérifier l'existence des obstacles en arrière. Donc cette boucle va juste arrêter le robot  $OC1RS = OC3RS = 0$  , changer la direction des roues vers l'arrière attendre un délai de 5 ms puis reculer à une vitesse  $OC1RS = OC3RS = 1800$ .

Le cas '**à droite**' : pour tourner à droite il suffit d'arrêter la roue droite du robot  $OC1RS = 0$  et de maintenir la vitesse de la roue de gauche  $OC3RS = 1800$ .

Le cas '**à gauche**' : même chose que pour tourner à droite, en inversant le rôle de chaque roue.



## 3.2 CONTRÔLE DU ROBOT

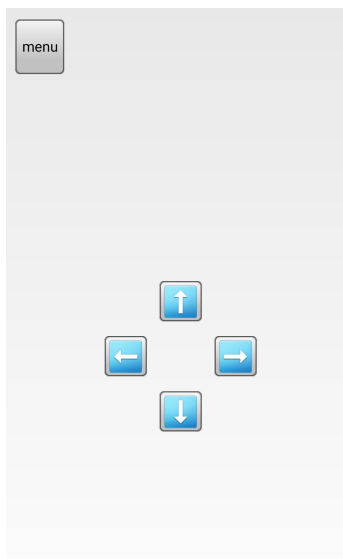


Figure 4 : Application de commande du robot

La première question que nous nous sommes posée pour contrôler le robot est la suivante : par quel moyen le faire avancer selon l'itinéraire voulu ? En premier lieu, nous pensions lui faire suivre un parcours prédéfini au choix afin qu'il puisse servir plusieurs tables. Le choix du parcours aurait été fait grâce à un pavé numérique sur le corps du robot. Après réflexion, nous avons trouvé cela plus pratique de pouvoir directement diriger le robot depuis notre téléphone. La solution choisie est donc la création d'une application qui se connecte à la carte Explorer 16/32 avec le bluetooth. Pour cela, nous avons téléchargé l'application mobile RoboRemoFree, qui permet de créer des boutons et d'envoyer les informations via bluetooth. Une fois notre application créée, il nous a fallu mettre en place le bluetooth sur la carte. Pour ce faire, nous avons dû développer un programme en C.

### 3.2.1 Les branchements

Avant d'installer le code sur la carte, nous avons dû brancher le module bluetooth à celle-ci. Pour cela nous nous sommes aidé de la [documentation du PmodHB5](#).

Comme on peut le voir sur la photo, les branchements sont les suivants : la masse sur la masse de la carte, le "Positive Voltage Supply" sur le "+3,3V", et finalement le RXD sur le TX de la carte et le TXD sur le RX de la carte. En effet, ces 2 ports correspondent à la réception et à la transmission des données et c'est donc important de croiser les deux paramètres pour que la communication puisse se faire. Une fois les bons branchements effectués et la carte connectée à l'ordinateur, le module bluetooth clignote.

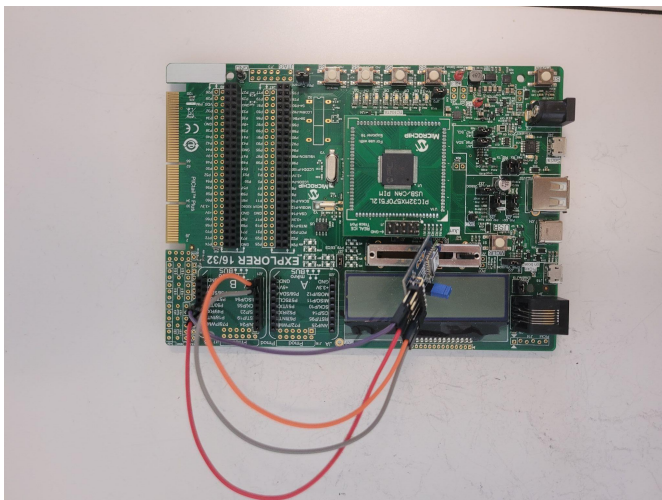


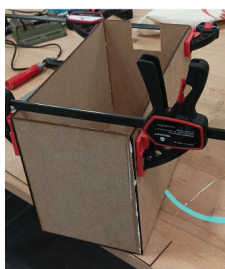
Figure 6: Photo de la pièce PMOD BLE

### 3.2.2 Le code

Le code consiste en plusieurs fichiers de langage C, nous allons parler dans cette partie du fichier Main, décomposable en trois grandes parties. Les 150 premières lignes de code permettent de définir et initialiser tous les paramètres nécessaires à la mise en place du bluetooth. La deuxième partie est une boucle qui attend et stocke les données entrantes jusqu'à ce qu'un retour chariot soit reçu ou que le nombre de caractères entrés (num\_chars) dépasse la taille maximale (max\_size). Elle retourne ensuite la chaîne de caractères. Enfin, en troisième partie, le programme principal initialise la carte puis lis les valeurs entières que nous rentrons dans le terminal.

### 3.3 LA CONCEPTION DU ROBOT

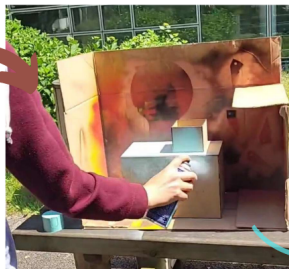
Le robot, représentant une vache, a été réalisé avec des planches de bois découpées au laser. Dans un premier temps, nous avons réalisé les plans du robot. Une planche a été prévue pour supporter la carte et ses composants ainsi que les boîtiers de piles. Cette planche est légèrement plus grande que le corps du robot, et est pourvue de petits rebords afin que les deux pièces puissent s'emboîter. Le corps du robot permet de cacher les différents composants. Sur le dessus, vous pouvez admirer sa tête, coiffée d'un plateau afin de pouvoir y mettre des verres (car ce robot est destiné à servir les clients dans un restaurant). Le plateau est lui aussi pourvu de rebords, afin que les verres ne puissent pas tomber au cours du voyage, au vu de la vitesse fulgurante de notre vachotide. Nous tenons à remercier chaleureusement Hélène RADE, qui nous a grandement aidées à la conception de cette vache, en découpant les planches aux dimensions souhaitées et en nous aidant au montage et à la décoration du robot.



**1ERE ÉTAPE : COLLER LES  
CARTONS**



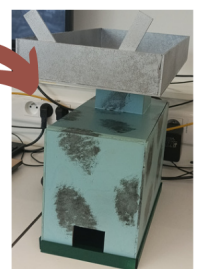
**LE ROBOT EST ASSEMBLÉ,  
DIRECTION LA PEINTURE !**



**ÉTAPE 2 : LA PEINTURE**



**ÉTAPE 3 : LA SÉCHAGE**



**ÉTAPE 4 : ASSEMBLAGE DE  
TOUTES LES PARTIES**

## 3.4 CAPTEUR À ULTRASONS

### 3.4.1 Intérêt d'un capteur à ultrasons et choix du capteur

Nous avons fait le choix d'intégrer un moyen à notre robot de réagir au mieux de manière autonome en cas d'obstruction inattendue de son passage, ou bien d'erreur de l'utilisateur. C'est donc pour cela que notre choix s'est porté sur un capteur à ultrasons (modèle HC-SR04) car il est tout d'abord relativement peu cher mais aussi parfait pour notre utilisation, c'est-à-dire la détection d'un obstacle à courte portée (c.f 3.4.2). Un capteur à ultrasons a néanmoins un inconvénient principal, notamment sa sensibilité à la température et à la pression pouvant entraîner des imprécisions (c.f 3.4.2). Il est également peu précis à longue portée.

Le capteur que nous avons choisi d'utiliser est le HC-SR04 car il est assez peu cher et est un exemple basique mais fiable d'un capteur à ultrasons classique. Afin de l'utiliser avec le micro-contrôleur utilisé dans ce projet, nous avons néanmoins dû utiliser un adaptateur PmodLVLSHFT pour pouvoir le connecter sans problème. Ces caractéristiques sont les suivantes :

- une portée de 2 centimètres à 4 mètres
- une fréquence de 40 kHz
- une impulsion de 10 microsecondes

### 3.4.2 Principe de Fonctionnement

Un capteur à ultrasons a un fonctionnement relativement simple. Il est constitué d'une pièce « trig », abréviation de « trigger », qui sert à envoyer une onde ultrason. Cette onde va se déplacer jusqu'à atteindre l'obstacle à détecter. Cette onde va alors rebondir et revenir sur le capteur, et être détectée par la pièce « echo » du capteur, soit le receveur d'onde. Le capteur mesure alors le temps mis par l'onde pour revenir.



Figure 7: Photo de la pièce HC-SR04

A partir de cette durée on peut alors calculer la distance à laquelle se trouve l'obstacle grâce à la formule suivante :

distance = vitesse x temps

En prenant comme vitesse la célérité du son calculée par :

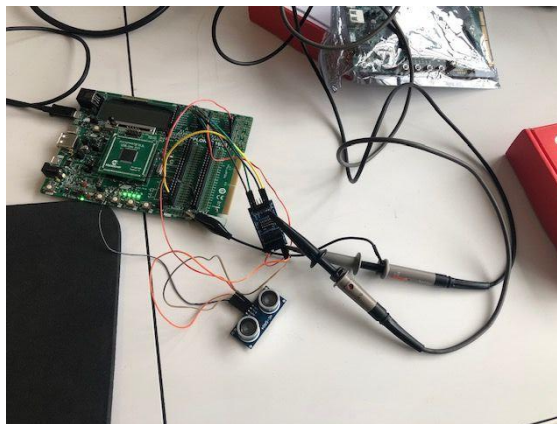
$c = 331,4 + (0,606 * \text{température}) + (0,0124 * \text{humidité}) = 344,2 \text{ m/s}$  dans les conditions d'expérience.

avec 331,4 la vitesse du son à 0°C et une humidité de 0 %.

On remarque que cette distance ne peut être calculée qu'à température et pression constante, sinon la distance calculée ne sera pas exacte. Cependant, le capteur n'a pas besoin d'une précision très élevée pour remplir son rôle.

### **3.4.3 Mise en fonctionnement du capteur**

Afin de faire fonctionner le capteur, nous avons d'abord dû le brancher avec la carte par l'intermédiaire de l'adaptateur comme sur la figure ci-contre : une fois les branchements faits et testés, nous avons pu commencer à programmer le capteur. Nous avons donc commencé par écrire un organigramme avant de passer au code pour plus de clarté.



*Figure 9: Photo des branchements entre carte, capteur et adaptateur*

Début

Émettre une onde ultrasonore pendant 10 microsecondes

On démarre le chronomètre

Le capteur détecte le retour de l'onde

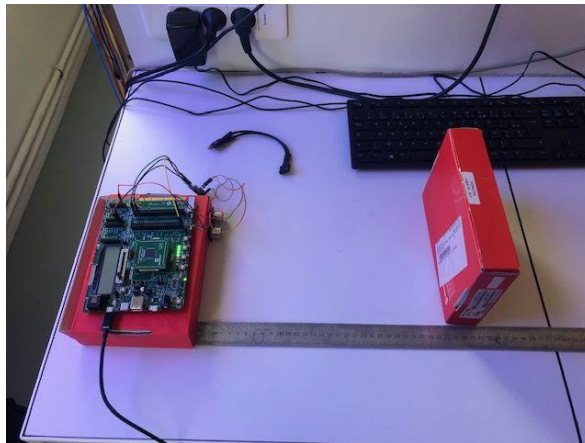
On arrête le chronomètre

On calcule la distance

Fin

### 3.4.4 Expérience et bilan

Après avoir programmé le capteur, nous sommes donc passé à des expériences afin d'en tester la précision. Nous avons donc placé le capteur à l'horizontale sur une paillasse puis avons placé à ses côtés une règle graduée. Tout en laissant le capteur là où il était, nous avons éloigné de plus en plus l'obstacle, sur l'image la boîte rouge, et avons effectué des mesures tous les 5 centimètres.



*Figure 8: Photo des teste sur la précision*

Les résultats obtenus furent les suivants :

<b>Distance mesurée</b>	10,3	15,3	20,4	24,9	28,9	34,9	40,9
<b>Distance réelle</b>	10	15	20	25	30	35	40

On remarque que le capteur est globalement précis, les erreurs de mesures venant probablement de l'erreur humaine et éventuellement d'une variation de la température de la pièce.

#### **4. CONCLUSIONS ET PERSPECTIVES**

La conception de ce petit robot à été une expérience formatrice pour nous six, et chacun d'entre nous en ressort ayant acquis de nouvelles compétences. Malgré le fait que le robot n'a pas pu être assemblé à temps, ce projet peut être considéré comme une réussite, du moins de notre point de vue. Il a fallu apprendre à jongler entre projet et études (et même un autre projet pour trois d'entre nous), apprendre à se coordonner, à diviser les tâches afin d'être le plus efficace possible. Cependant, l'aspect le plus important de ce projet à nos yeux reste tout ce qu'il nous a permis de découvrir.

Ce projet a été pour nous une porte d'entrée vers de nombreux domaines. Puisque la majorité d'entre nous souhaite poursuivre ses études en informatique, découvrir le langage C dans le cadre de ce projet est d'une grande valeur. En parallèle, le support utilisé (carte et logiciel), bien que à but beaucoup plus précis et restreint, a été très intéressant à utiliser.

Beaucoup d'améliorations sont envisageables pour le robot, comme par exemple une façon de changer de vitesse, ou l'ajout d'un capteur anti collisions à l'arrière. De plus, nous sommes loin d'avoir exploité le logiciel au maximum de ses capacités, et nous n'avons eu accès qu'à un nombre limité de composants.

## 5. BIBLIOGRAPHIE

1 - MPLABXIDEPart3\_PWM.pdf

2 - Section44\_Motor\_control.pdf

3- Section16\_output\_PWM.pdf

3 - PIM\_and\_PICtail\_Plus\_Plus\_Pinout\_Explorer16\_and\_Explorer16\_32.pdf

4 - How to Set up an Ultrasonic Range Finder :

<https://www.circuitbasics.com/how-to-set-up-an-ultrasonic-range-finder-on-an-arduino/>

5 - Fonctionnement du BLE :

<https://create.arduino.cc/projecthub/JANAK13/bluetooth-controlled-servo-439997>

6- PMOD HB5 commande de moteur CC:

<https://digilent.com/reference/pmod/pmodhb5/start>

7 - PMOD BLE circuit Bluetooth , connexion par liaison série RS232 :

<https://digilent.com/shop/pmod-ble-bluetooth-low-energy-interface/>

8 - Microchip explorer 16/32 :

<https://microchipdeveloper.com/boards:explorer1632>