



INSTITUT NATIONAL DES POSTES ET TÉLÉCOMMUNICATIONS  
FILIÈRE CSE2

---

Rapport Mini-projet développement :  
Self Balancing Rebot

---

*Réalisé par :*  
Ibrahim JOUNAHI

*Encadré par :*  
Mr. Mohcine CHAMI

9 juillet 2019

### **Remerciements**

Je vous remercie, Monsieur Mohcine, d'avoir partagé avec moi votre passion pour l'enseignement. j'ai grandement apprécié votre soutien, votre implication et votre expérience

# Table des matières

<b>Introduction</b>	<b>1</b>
<b>Mots clés</b>	<b>2</b>
<b>1 Présentation des différents éléments majeurs du projet :</b>	<b>3</b>
1.1 MPU6050 : . . . . .	3
1.2 Régulateur PID : . . . . .	5
1.3 LMD18200 : . . . . .	5
1.4 Arduino : . . . . .	7
1.5 Moteur DC : . . . . .	9
1.6 Alimentation : . . . . .	10
1.7 PWM pulse width modulation : . . . . .	11
<b>2 L'analyse fonctionnelle du robot :</b>	<b>12</b>
2.1 L'analyse du besoin : . . . . .	12
2.2 Fonctions de services : . . . . .	12
2.3 Cahier des charges : . . . . .	13
<b>3 Assemblage et réalisation :</b>	<b>14</b>
3.1 Interconnexion des différents composants : . . . . .	14
3.1.1 MPU6050 avec l'Arduino : . . . . .	15
3.1.2 L298N avec l'Arduino et les moteurs : . . . . .	16
3.1.3 Assemblage complet : . . . . .	16
3.2 Code source : . . . . .	17
<b>Webographie</b>	<b>20</b>

# Introduction

L'équilibrage à pendule inverse est un phénomène répandu dans le monde qui nous entoure. Le corps humain est capable d'équilibrer avec une précision incroyable grâce à nos capteurs biologiques, tels que nos yeux et nos oreilles (internes). Dans le monde des robots, toutefois, un tel équilibrage est en réalité un processus assez compliqué et nécessite des capteurs avancés et une puissance de calcul importante pour ajuster et équilibrer le pendule. Ce projet est notre tentative de recréer un robot d'équilibrage intelligent et l'étude des principes nécessaires pour contrôler efficacement ce système robotique. C'est pour cela, dans le cadre du mini projet développement, j'ai choisi de traiter ce modèle robot et d'élaborer ce mémoire qui présente le concept du robot à équilibrage automatique.

Les systèmes de contrôle sont partout autour de nous. Ils sont utilisés dans des produits commerciaux ainsi que dans des applications industrielles et militaires. Traditionnellement, ces systèmes étaient analogiques et leurs différents composants, tels que les capteurs, les actionneurs et les circuits de commande, étaient interconnectés par des connexions physiques aux signaux analogiques.

Aujourd'hui, avec l'évolution des ordinateurs et des technologies de l'information, la plupart des systèmes de contrôle sont conçus comme des systèmes numériques basés sur des cartes ARDUINO. Ceci permet de connecter leurs différents composants par des connexions logiques sur un réseau de communication.

Dès 1995, les robots prennent de plus en plus d'importance, tant dans le domaine industriel que dans la vie quotidienne. En effet, le robot que nous devons réaliser est une copie d'un robot existant, commercialisé et destiné à un jeune public (Segway).

L'objectif principal de ce projet est de réaliser un système contrôlé par un ARDUINO UNO. Ce dernier sera le cerveau du système, il enverra des requêtes aux différents composants constituant le robot.

# Mots clés

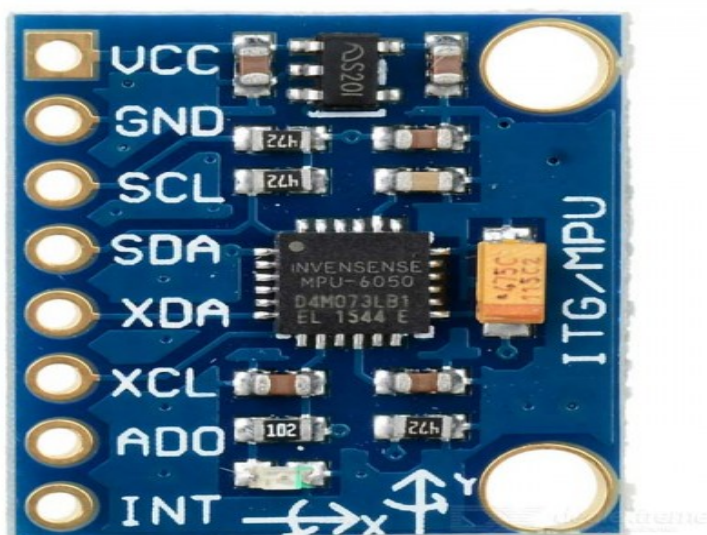
MPU6050  
Accéléromètre  
Gyroscope  
L298N  
Arduino  
DC motors  
PWM

# Chapitre 1

## Présentation des différents éléments majeurs du projet :

### 1.1 MPU6050 :

Le capteur MPU6050 contient à la fois un accéléromètre 3 axes et un gyroscope 3 axes qui communique via i2c il est en général fixé à un mobile ou tout autre objet, permet de mesurer l'accélération linéaire de ce dernier, le gyroscope capte la position angulaire de cet objet



MotionInterface est en train de devenir une fonction «indispensable» adoptée par les fabricants de smartphones et de tablettes en raison de l'énorme valeur qu'elle ajoute à l'expérience de l'utilisateur final. Dans les smartphones, il trouve une utilisation dans des applications telles que les commandes gestuelles pour les applications et le contrôle du téléphone, les jeux améliorés, la réalité augmentée, la capture et la visualisation de photos panoramiques et la navigation pour piétons et véhicules. Grâce à sa capacité à suivre avec précision et précision les mouvements des utilisateurs, la technologie MotionTracking peut convertir les combinés et les tablettes en puissants dispositifs 3D intelligents pouvant être utilisés dans des applications allant de la surveillance de la santé et de la forme physique aux services basés sur la localisation. Les exigences clés pour les périphériques activés par MotionInterface sont petites taille de l'emballage, faible consommation d'énergie, haute précision et répétabilité, tolérance élevée aux chocs et application programmabilité de performance spécifique - le tout à un prix bas pour le consommateur.

## 4 Applications

- *BlurFree™* technology (for Video/Still Image Stabilization)
- *AirSign™* technology (for Security/Authentication)
- *TouchAnywhere™* technology (for "no touch" UI Application Control/Navigation)
- *MotionCommand™* technology (for Gesture Short-cuts)
- Motion-enabled game and application framework
- InstantGesture™ iG™ gesture recognition
- Location based services, points of interest, and dead reckoning
- Handset and portable gaming
- Motion-based game controllers
- 3D remote controls for Internet connected DTVs and set top boxes, 3D mice
- Wearable sensors for health, fitness and sports
- Toys

FIGURE 1.1 – Les applications du MPU6050

## 5 Features

### 5.1 Gyroscope Features

The triple-axis MEMS gyroscope in the MPU-60X0 includes a wide range of features:

- Digital-output X-, Y-, and Z-Axis angular rate sensors (gyroscopes) with a user-programmable full-scale range of  $\pm 250$ ,  $\pm 500$ ,  $\pm 1000$ , and  $\pm 2000^\circ/\text{sec}$
- External sync signal connected to the FSYNC pin supports image, video and GPS synchronization
- Integrated 16-bit ADCs enable simultaneous sampling of gyros
- Enhanced bias and sensitivity temperature stability reduces the need for user calibration
- Improved low-frequency noise performance
- Digitally-programmable low-pass filter
- Gyroscope operating current: 3.6mA
- Standby current: 5 $\mu$ A
- Factory calibrated sensitivity scale factor
- User self-test

### 5.2 Accelerometer Features

The triple-axis MEMS accelerometer in MPU-60X0 includes a wide range of features:

- Digital-output triple-axis accelerometer with a programmable full scale range of  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$  and  $\pm 16g$
- Integrated 16-bit ADCs enable simultaneous sampling of accelerometers while requiring no external multiplexer
- Accelerometer normal operating current: 500 $\mu$ A
- Low power accelerometer mode current: 10 $\mu$ A at 1.25Hz, 20 $\mu$ A at 5Hz, 60 $\mu$ A at 20Hz, 110 $\mu$ A at 40Hz
- Orientation detection and signaling
- Tap detection
- User-programmable interrupts
- High-G interrupt
- User self-test

FIGURE 1.2 – Les caractéristiques de gyroscope et d'accéléromètre

## 1.2 Régulateur PID :

L'élément le plus important de tout robot est le contrôleur. Surtout pour un robot d'auto-équilibrage, le programme de contrôle est vital, car il interprète les données de capteurs et décide combien les moteurs doivent être déplacés pour que le robot ne reste debout. Le contrôleur le plus couramment utilisé pour les systèmes de stabilisation est le régulateur PID.

PID signifie proportionnelle, intégrale et dérivée, en se référant aux équations mathématiques utilisées pour calculer la sortie.

Le P-composant prend simplement dans l'angle actuel du robot et rend les moteurs se déplacent dans la même direction que le robot est en baisse. Par conséquent, la poursuite du robot tombe cible, plus les moteurs se déplacent. Si le P-composant est utilisé seul, le robot pourrait se stabiliser pendant un certain temps, mais le système aura tendance à dépasser, osciller et finalement tomber.

I-composant est utilisé pour accumuler les erreurs. Par exemple, si le robot a tendance à tomber sur un côté, il sait qu'il a besoin de se déplacer dans la direction opposée, afin de maintenir la sur la cible et pour prévenir la dérive vers la gauche ou la droite.

Enfin, le composant D est chargé d'amortir toute oscillation et assure que le robot ne vibre pas trop. Il agit tout simplement contre tout mouvement.

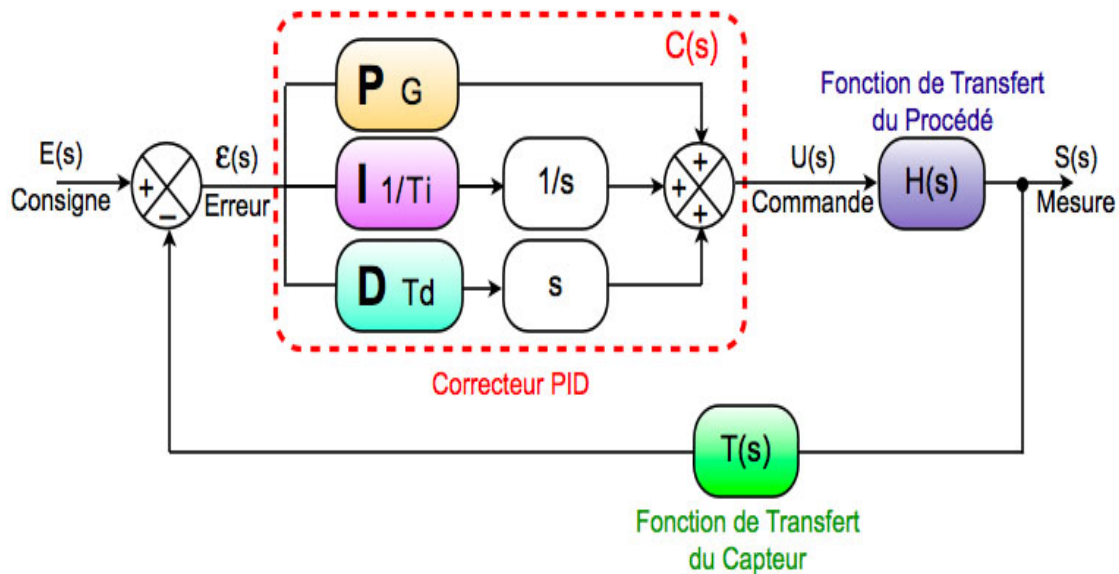


FIGURE 1.3 – Correcteur PID classique

## 1.3 LMD18200 :

Ce breakout board est un Double Pont-H destiné au contrôle de moteur continu (H-Bridge Motor Driver). Il est basé sur le composant L298N qui est un double Pont-H conçu spécifiquement pour ce cas d'utilisation.

C'est un module extrêmement utile pour le contrôler de robots et ensembles mécanisés. Il peut contrôler deux moteur courant continu ou un moteur pas-à-pas 4 fils 2 phases. il est conçu pour supporter des tensions plus élevées, des courants importants tout en proposant une commande logique TTL (basse tension, courant faibles, idéal donc pour un microcontrôleur).

Il peut piloter des charges inductives comme des relais, solénoïdes, moteurs continus et moteurs pas-à-pas. Les deux types de moteurs peuvent être contrôlés aussi bien en vitesse (PWM) qu'en direction. Toutes les sorties en puissance sont déjà protégées par des diodes anti-retour.

Il s'agit d'un module prêt à l'emploi.



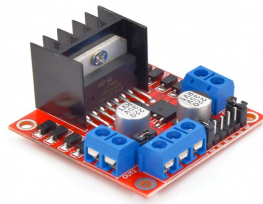


FIGURE 1.4 – L298N

### Caractéristiques

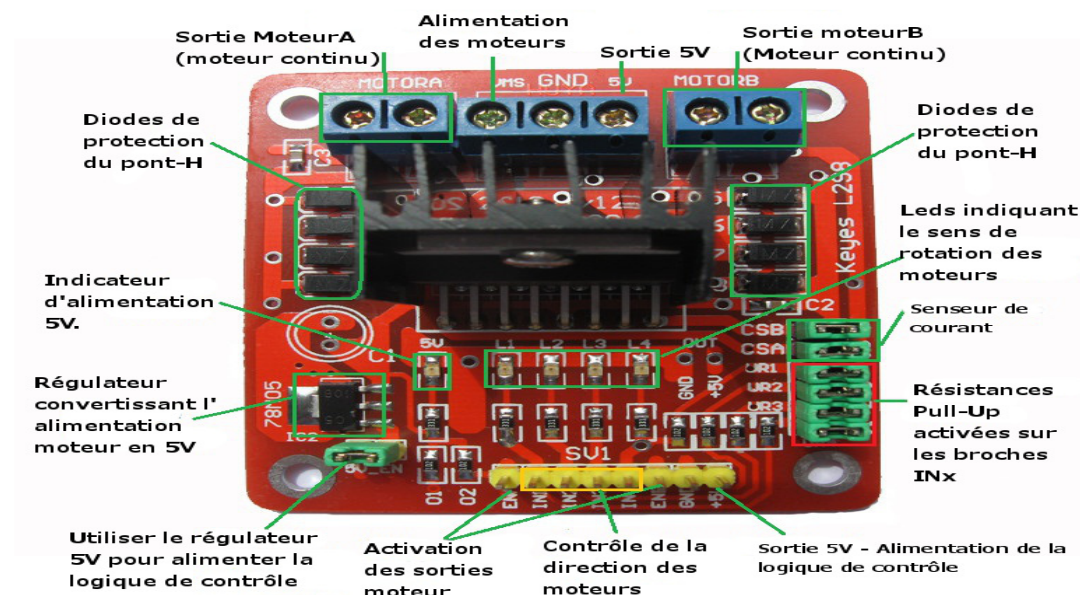
---

- Léger, petit
- Des capacités hors-pair pour contrôle moteur
- Diodes de protections
- Un dissipateur (pour dissiper la chaleur en cas de forte charge)
- Un sélecteur pour sélectionner la source d'alimentation
- 4 Sélecteurs pour les résistances pull up
- Sortie pour 2 moteurs continu/ 1 moteur pas-à-pas (4 bobines, deux phases)
- Indicateur LED pour sens de rotation moteur
- Indicateur LED pour alimentation 5V
- 4 trous de fixation standard

### Spécifications

---

- Composant de contrôle en puissance: L298N
- Alimentation de la charge: de +6V à +35V
- Courant Max (en pointe): 2A
- Tension de commande logique Vss: de +5 à +7V (alimentation interne de +5V)
- Courant de commande logique: de 0 à 36mA
- Tensions pour contrôle du sens: Low -0.3V~1.5V, high: 2.3V~Vss
- Tensions pour contrôle "Enable": Low -0.3V~1.5V, high: 2.3V~Vss
- Puissance Max: 25W (Température 75 °C)
- Température de fonctionnement: de -25°C à +130°C
- Dimensions: 60mm x 54mm
- Poids: ~48g



## 1.4 Arduino :

Le modèle UNO de la société ARDUINO est une carte électronique dont le coeur est un microcontrôleur ATMEL de référence ATmega328. Le microcontrôleur ATmega328 est un microcontrôleur 8bits de la famille AVR dont la programmation peut être réalisée en langage C.

L'intérêt principal des cartes ARDUINO (d'autres modèles existent) est leur facilité de mise en oeuvre. ARDUINO fournit un environnement de développement s'appuyant sur des outils opensource. Le chargement du programme dans la mémoire du microcontrôleur se fait de façon très simple par port USB. En outre, des bibliothèques de fonctions "clé en main" sont également fournies pour l'exploitation d'entrées-sorties courantes : gestion des E/S TOR, gestion des convertisseurs ADC, génération de signaux PWM, exploitation de bus TWI/I2C, exploitation de servomoteurs ... L'objectif de ce document est de mettre en évidence certaines informations techniques concernant l'exploitation des périphériques intégrés, en particulier lorsqu'on n'utilise pas les fonctions "clé en main" d'ARDUINO, dans l'objectif de comprendre comment ça marche.

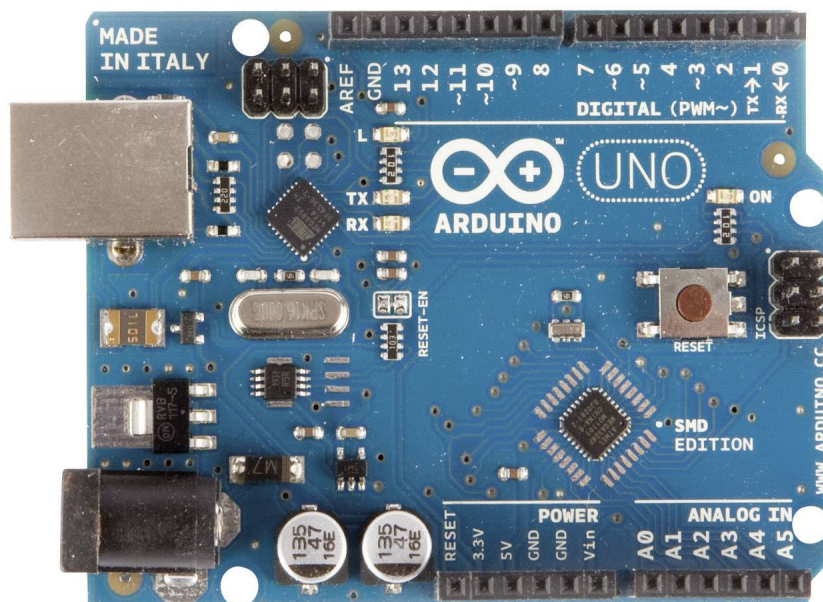


FIGURE 1.5 – L'interface extérieur de la carte arduino

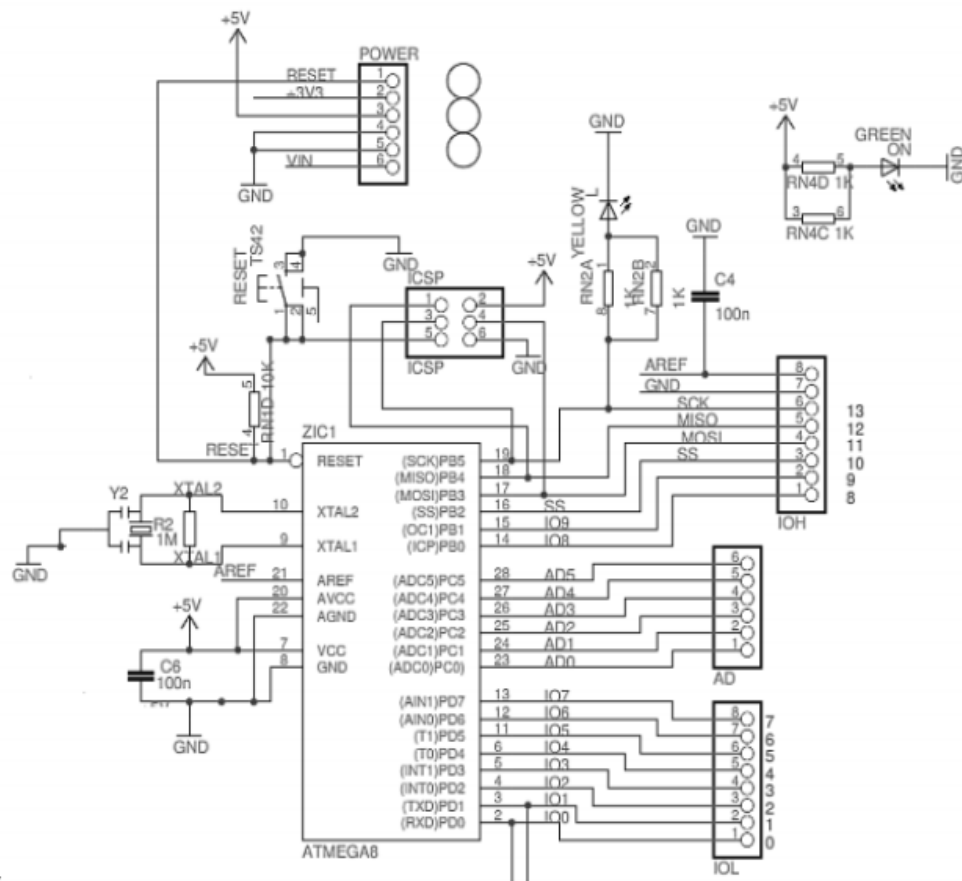


FIGURE 1.6 – Schéma des connexion entrées sorties

### Caractéristiques principales:

- version: Rev. 3
- alimentation:
  - via port USB ou
  - 7 à 12 V sur connecteur alim 5,5 x 2,1 mm
- microprocesseur: ATmega328
- mémoire flash: 32 kB
- mémoire SRAM: 2 kB
- mémoire EEPROM: 1 kB
- 14 broches d'E/S dont 6 PWM
- 6 entrées analogiques 10 bits
- intensité par E/S: 40 mA
- cadencement: 16 MHz
- bus série, I2C et SPI
- gestion des interruptions
- fiche USB B
- dimensions: 74 x 53 x 15 mm

FIGURE 1.7 – Détails techniques de l'arduino Uno

## 1.5 Moteur DC :

Les moteurs DC avec des aimants permanents sont très courants dans différentes applications, avec de petites dimensions, lorsqu'on recherche beaucoup de puissance à bas prix. A cause de leur vitesse assez importante, ils sont utilisés dans plusieurs sortes de transmission (pour obtenir beaucoup de couple avec peu de vitesse).

Les moteurs DC à aimant permanent sont d'une fabrication assez simple et les commander est assez élémentaire. Bien que les contrôler soit assez simple, leur vitesse n'est pas déterminée précisément par le signal de contrôle parce qu'il dépend de nombreux autres facteurs, dont le couple appliqué sur l'arbre et le courant d'alimentation. La relation entre le couple et la vitesse d'un moteur DC parfait est linéaire, ce qui signifie que : plus grand est l'effort sur l'arbre, plus petite est la vitesse de l'arbre et plus grand est le courant dans la bobine.

Les moteurs DC utilisent la tension DC et ne nécessitent pas de contrôle électronique supplémentaire puisque toutes les communications nécessaires sont faites dans le moteur. Lorsque le moteur fonctionne, deux broches statiques glissent dans le commutateur de rotation et garde la tension dans la bobine. La direction de la rotation du moteur est déterminée par la polarité du courant appliqué. Si le moteur ne doit tourner que dans une seule direction, alors le courant peut passer à travers un relai ou tout autre connexion simple. Si le moteur doit tourner dans plusieurs directions, on utilisera un circuit électronique appelé H-bridge (ou "demi-pont").

Dans le H-bridge il y a quatre transistors (ou quatre groupes) qui dirigent le courant pour faire tourner le moteur. Le schéma électrique du H-bridge ressemble à la lettre H et c'est pourquoi on le nomme ainsi. La particularité du H-bridge est la possibilité d'appliquer plusieurs directions de polarité au moteur. L'illustration montre le schéma principal d'un H-bridge avec l'exemple d'interrupteurs. Si deux interrupteurs en diagonale sont fermés, le moteur démarre. La direction de la rotation dépend de laquelle des deux diagonales est fermée. Dans les vrais H-bridge les interrupteurs sont remplacés par des transistors qui sont sélectionnés en fonction du courant du moteur et de la tension.

En plus de changer la direction de la rotation, le H-bridge permet de modifier la vitesse de rotation. Pour ce faire, les transistors doivent être ouverts et fermés constamment en utilisant la modulation de la largeur de l'impulsion (PWM), ainsi l'énergie totale fournie au moteur est quelquepart entre le moteur éteint et le moteur fonctionnant à pleine puissance. Le temps d'ouverture dans toutes les périodes du PWM est appelé cycle de service, donné en pourcent. 0 pour cent signifie que le transistor est constamment fermé – ce n'est pas du courant conducteur. 100 pour cent signifie que le transistor est constamment fermé ainsi le courant conducteur. La fréquence du PWM doit être assez élevée afin d'éviter les vibrations dans l'arbre du moteur. A petites fréquences le moteur produit du bruit, on utilise alors le plus fréquemment une fréquence de modulation de 20 kHz. D'autre part le rendement du H-bridge n'est pas très bon à hautes fréquences. Les vibrations de l'arbre du moteur sont réduites par l'inertie du rotor et l'induction des bobines.



FIGURE 1.8 – Moteur DC

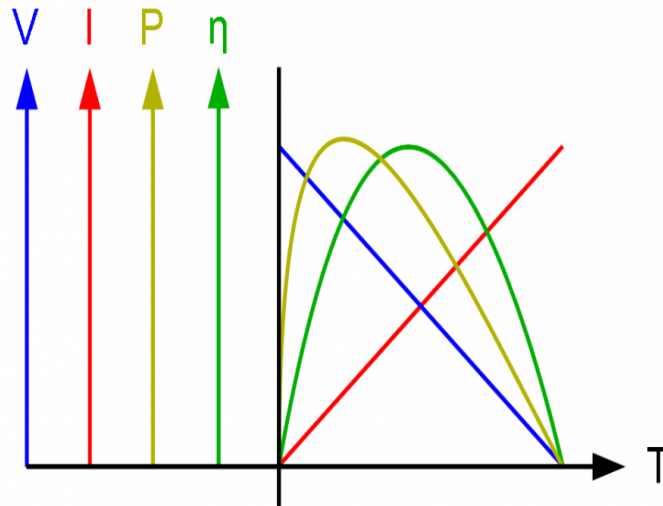


FIGURE 1.9 – Le graph parfait de la relation entre la vitesse ( $V$ ), le courant ( $I$ ), la puissance ( $P$ ), le rendement ( $\eta$ ) et le couple ( $T$ ) d'un moteur DC.

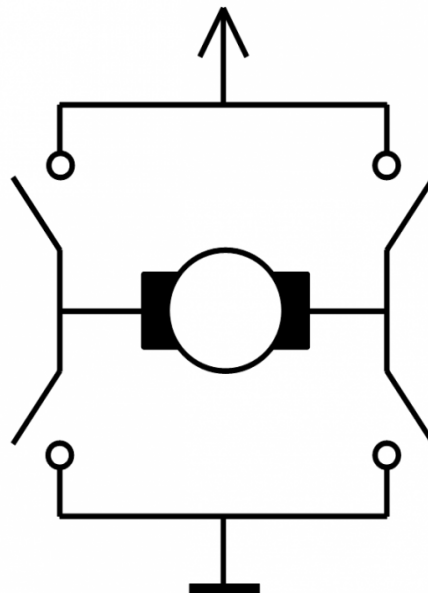


FIGURE 1.10 – Principe de fonctionnement du H-bridge utilisé avec les interrupteurs.

## 1.6 Alimentation :

Le robot a besoin d'une batterie ou générateur de courant comme source d'énergie pour fonctionner .Les caractéristiques principales de cette alimentation sont sa tension de sortie, sa capacité et sa résistance interne. On alimente la carte par une tension DC de 12V délivrée vers les moteurs DC.





FIGURE 1.11 – Le générateur à 12V DC utilisé dans le projet

## 1.7 PWM pulse width modulation :

Dans ce projet, le mouvement des moteurs est engendrés par le signal PWM. En effet, la technique de modulation de largeur d'impulsion (Pulse Width Modulation PWM) consiste à générer un signal carré avec un rapport cyclique modulé en fonction d'un signal de commande. Le signal généré peut servir à commander un circuit de puissance à découpage (pont en H), associé à un filtrage passe-bas inductif, pour générer une onde sinusoïdale ou d'une autre forme. La technique est utilisée dans les onduleurs mono-phasés, diphasés ou triphasés . Le même principe est utilisé dans les amplificateurs Audio. D'où, on fait une approximation analogique au signal numérique.

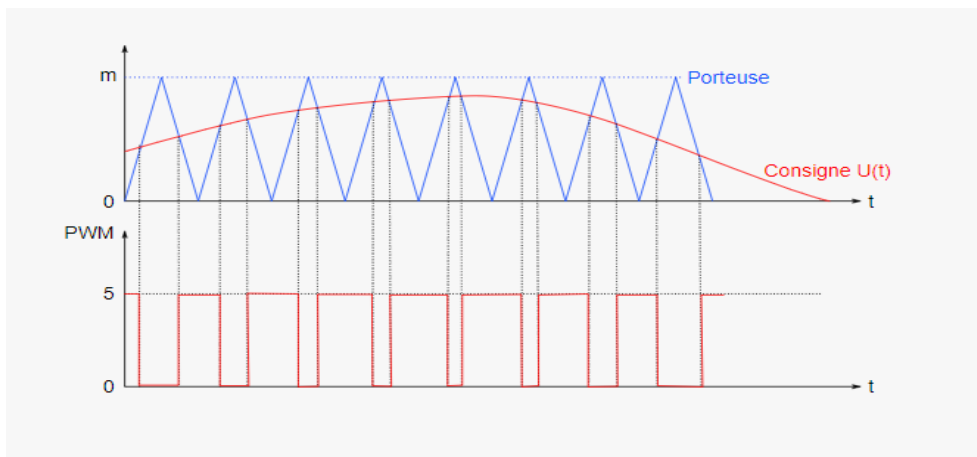


FIGURE 1.12 – Approximation de signal numérique vers un signal analogique.

## Chapitre 2

# L'analyse fonctionnelle du rebot :

### 2.1 L'analyse du besoin :

Verbalisation et schématisation du besoin :

Un des chemins les plus courants pour formaliser le besoin est de répondre aux quelques questions (bête à corne) :

- A qui le produit rend il service : Utilisateur qui peut être un étudiant (comme mini projet d'étude) ou public (inspiration du Segway)
- Dans quel but : Pour conserver l'équilibre du système 2 roues
- Sur quoi le produit agit il : Centre de gravité

Par conséquent, on transforme ces questions en diagramme dit de "bête à corne"

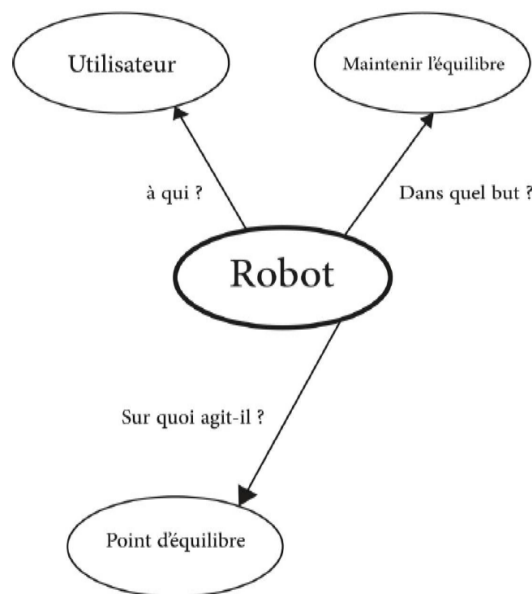


FIGURE 2.1 – Diagramme de "bête à corne"

### 2.2 Fonctions de services :

Elles peuvent être classées en 2 catégories :

- Fonction principale : FP (aussi appelée fonction d'Interaction FI) C'est une fonction de service qui justifie la création du projet.
- Fonction contrainte : FC (aussi appelée fonction d'adaptation FA) C'est une fonction de service qui limite la liberté du concepteur. Pour définir ces fonctions de service, on trace le graphe des interactions.

On représente ces fonctions à l'aide de l'outil diagramme "Pieuvre" ou graphe des interactions, il a pour objectif d'analyser les besoins et identifier les fonctions de service d'un produit.

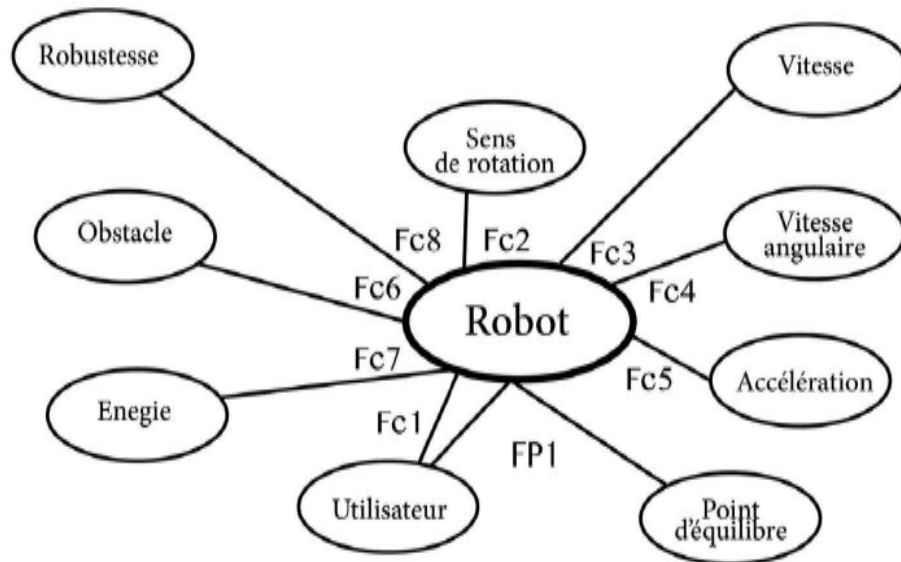


FIGURE 2.2 – Diagramme ‘pieuvre’ du robot

- Fonctions principales :

FP 1 : Etre constamment équilibré par une force extérieure (qui sera 2 moteurs DC).

- Fonction de contraintes :

FC 1 : Etre autonome.

FC 2 : Inverser le sens du courant dans les moteurs = sens de rotation.

FC 3 : Commander la vitesse.

FC 4 : Mesurer la vitesse angulaire.

FC 5 : Mesurer l'accélération.

FC 6 : Eviter les obstacles.

FC 7 : Autonomie en énergie.

FC 8 : Le support doit etre robuste.

## 2.3 Cahier des charges :

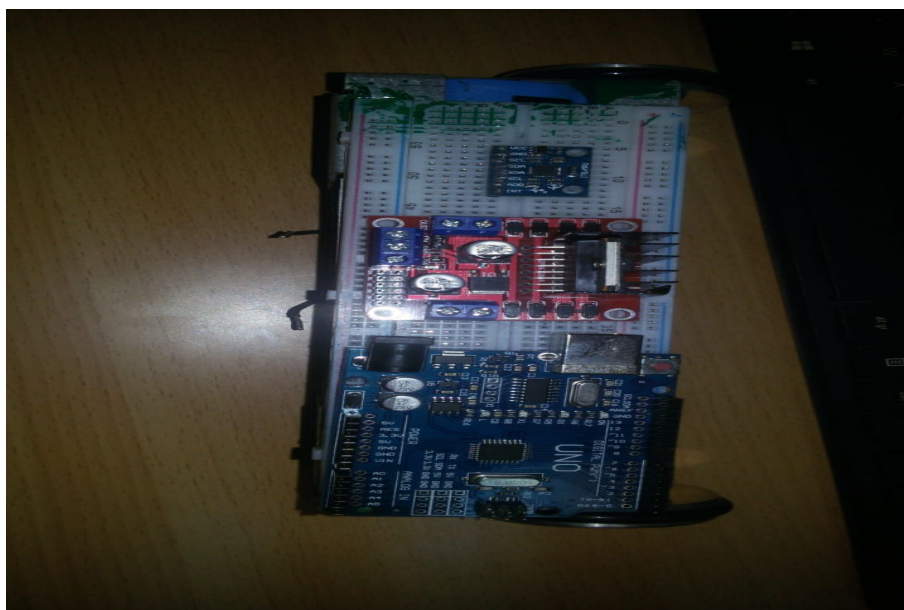
- Corps du robot animé d'un mouvement de translation (avance, recule, tourner)
- L298N
- Utilisation d'un Arduino UNO (11 dollars)
- Utilisation d'un MPU6050 (4 dollars)
- Transportable (hauteur maximale de 17 cm et poids inférieur à 2 Kg)
- Langage de programmation utilisé : C



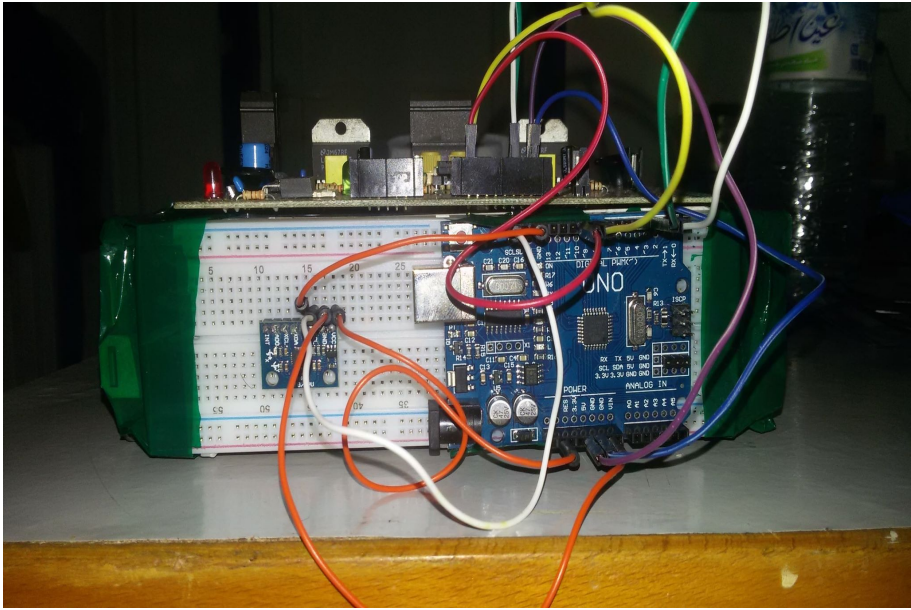
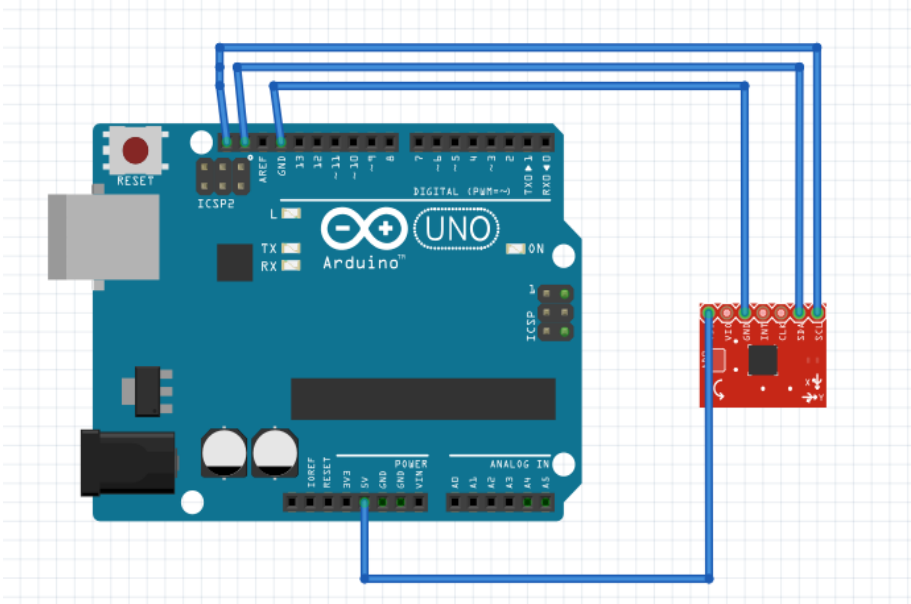
## Chapitre 3

### Assemblage et réalisation :

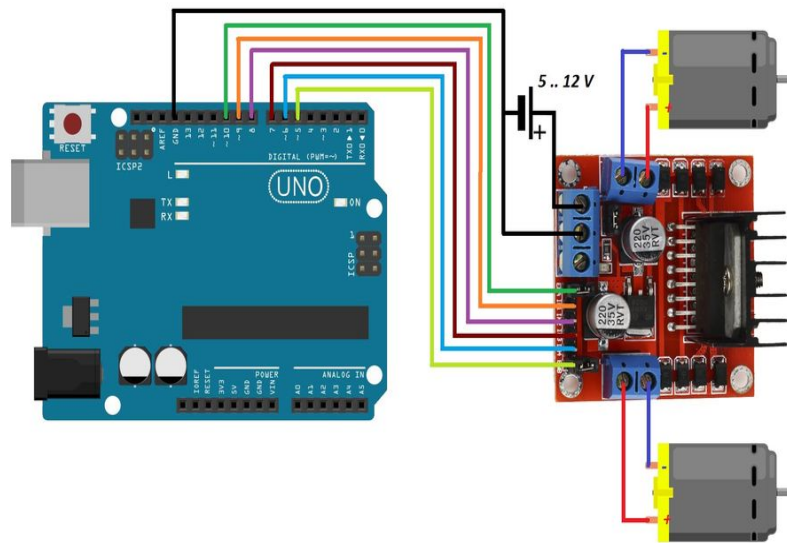
#### 3.1 Interconnexion des différents composants :



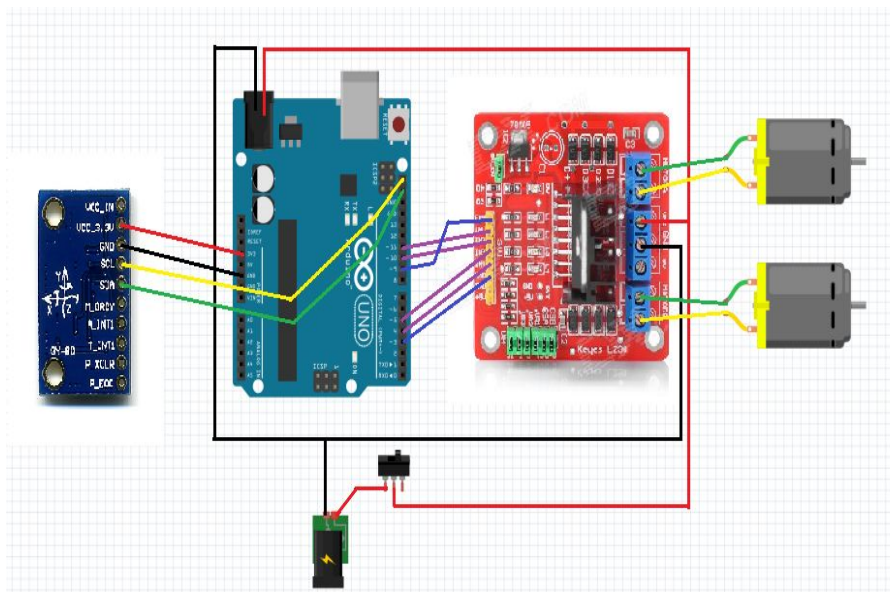
3.1.1 MPU6050 avec l'Arduino :

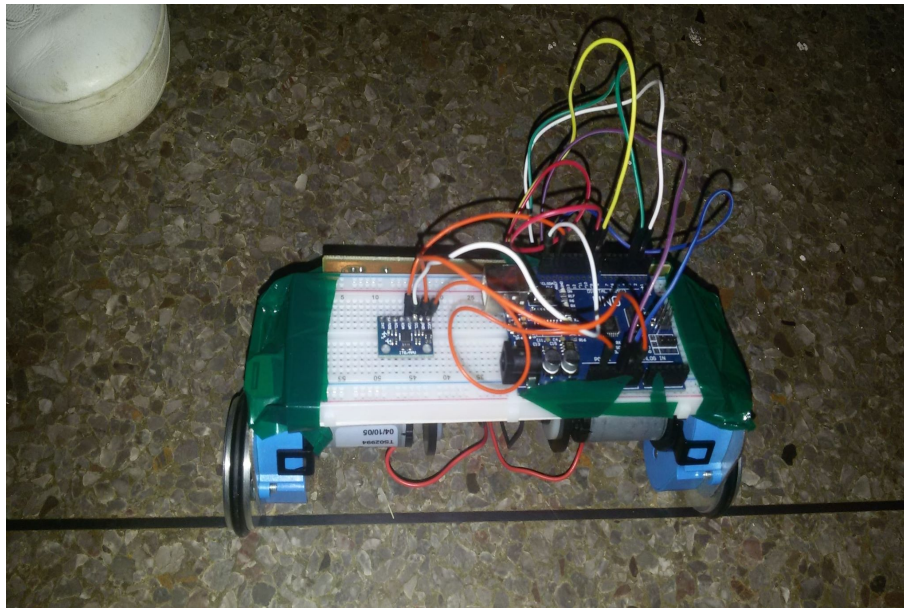


### 3.1.2 L298N avec l'Arduino et les moteurs :



### 3.1.3 Assemblage complet :





## 3.2 Code source :

```

MPUcode | Arduino 1.8.8
Fichier Édition Croquis Outils Aide

#include "Wire.h" // Arduino Wire library
#include "I2Cdev.h" // Installer ces 2 librairies
#include "MPU6050.h"
#include "math.h"

// ADO low = 0x68 (default for InvenSense evaluation board)
// ADO high = 0x69
MPU6050 accelgyro;

int16_t ax, ay, az; //mesures brutes
int16_t gx, gy, gz;
float angle=0;

// motor initialization
int In1 = 6;
int In2 = 7;
int In3 = 9;
int In4 = 8;
int ENA = 10;
int ENB = 5;

void setup() {
  Wire.begin(); //I2C bus
  //Serial.begin(9600);
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB (LEONARDO)
  }

  // initialize device
  accelgyro.initialize();

  // motor
  pinMode(In1,OUTPUT);
  pinMode(In2,OUTPUT);
  pinMode(In3,OUTPUT);
  pinMode(In4,OUTPUT);
}

```

MPUcode | Arduino 1.8.8

Fichier Édition Croquis Outils Aide

✓ ↻ 📄 ⬆ ⬇

MPUcode

```
accelgyro.initialize();

// motor
pinMode(In1,OUTPUT);
pinMode(In2,OUTPUT);
pinMode(In3,OUTPUT);
pinMode(In4,OUTPUT);
pinMode(ENA, OUTPUT);
pinMode(ENB, OUTPUT);

digitalWrite(In1,HIGH);
digitalWrite(In2, LOW);
digitalWrite(In3,HIGH);
digitalWrite(In4, LOW);
}

void negativeState()
{
digitalWrite(In1 , HIGH);
digitalWrite(In2, LOW);
digitalWrite(In3, HIGH);
digitalWrite(In4,LOW);
analogWrite( ENA, 100);
analogWrite( ENB, 100);
}

void positiveState()
{
digitalWrite(In1 , LOW);
digitalWrite(In2, HIGH);
digitalWrite(In3, LOW);
digitalWrite(In4,HIGH);
analogWrite( ENA, 100);
analogWrite( ENB, 100);
}

void loop() {
```



MPUcode | Arduino 1.8.8

Fichier Édition Croquis Outils Aide

✓ ↻ 📄 ⬆ ⬇

MPUcode

```
void negativeState()
{
  digitalWrite(In1 , HIGH);
  digitalWrite(In2, LOW);
  digitalWrite(In3, HIGH);
  digitalWrite(In4, LOW);
  analogWrite( ENA, 100);
  analogWrite( ENB, 100);
}

void positiveState()
{
  digitalWrite(In1 , LOW);
  digitalWrite(In2, HIGH);
  digitalWrite(In3, LOW);
  digitalWrite(In4, HIGH);
  analogWrite( ENA, 100);
  analogWrite( ENB, 100);
}

void loop() {
  accelgyro.getMotion6(&sax, &say, &sz, &sgx, &sgy, &sgz);
  angle= atan2((double)sax, (double)sz )*180/PI; // 180/PI permet d'avoir la valeur en °
  Serial.println(angle);
  delay(10);

  // equilibrage
  if ( angle <= -5 )
  {
    negativeState();
  }
  else if ( angle >= 5 )
  {
    positiveState();
  }
}
```

# Webographie

<https://store.invensense.com/datasheets/invensense/MPU-6050>  
<https://fr.wikipedia.org/wiki/Regulateur-PID>  
<http://www.ti.com/lit/ds/symlink/lmd18200.pdf>  
<http://gilles.thebault.free.fr/spip.php?article32>  
<http://forum.arduino.cc/index.php?topic=60170.0>  
<http://home.roboticlab.eu/fr/examples/motor/dc> <http://www.f-legrand.fr/scidoc/docimg/sciphys/arduino/ondepwm/c>