

REX Minotaure

2019



# Table des matières

<b>I</b>	<b>Introduction</b>	<b>5</b>
0.1	Histoire . . . . .	6
0.2	Rôle de Minotaure . . . . .	6
0.3	Les différentes équipes au cours du temps . . . . .	6
0.3.1	2015-2016 . . . . .	6
0.3.2	2016-2017 . . . . .	6
0.3.3	2017-2018 . . . . .	6
0.3.4	2018-2019 . . . . .	6
0.3.5	2019-2020 . . . . .	6
<b>II</b>	<b>Composants</b>	<b>7</b>
<b>1</b>	<b>Actionneurs</b>	<b>8</b>
1.1	Moteurs pas-à-pas . . . . .	8
1.1.1	Utilisation . . . . .	9
1.2	Moteurs à courant continu . . . . .	10
1.2.1	Utilisation . . . . .	10
1.3	Moteurs brushless . . . . .	11
1.3.1	Critères de sélection . . . . .	11
1.3.2	Utilisation . . . . .	12
1.4	Servomoteurs . . . . .	12
1.4.1	Critères de sélections . . . . .	13
1.4.2	Utilisation . . . . .	13
1.5	Les Dynamixels . . . . .	14
<b>2</b>	<b>Système pneumatique</b>	<b>15</b>
2.1	Pompe . . . . .	15
2.1.1	Utilisation . . . . .	15
2.2	Ventouse . . . . .	16
2.3	Accessoires et autres . . . . .	17
2.3.1	Tubes . . . . .	17
2.3.2	Valves . . . . .	17
<b>3</b>	<b>Communication</b>	<b>18</b>
3.1	Radio . . . . .	18
3.2	Bluetooth . . . . .	18
3.3	Module Xbee . . . . .	19
3.3.1	Architecture réseau . . . . .	19
<b>4</b>	<b>Cartes électroniques</b>	<b>21</b>
4.1	Arduino . . . . .	21
4.2	Raspberry Pi . . . . .	21
4.2.1	Ports USB . . . . .	21
4.3	Teensy . . . . .	21

4.4	Node Mcu . . . . .	22
4.5	Shield . . . . .	22
<b>5</b>	<b>Capteurs</b>	<b>23</b>
5.1	GNSS . . . . .	23
5.1.1	NMEA . . . . .	23
5.1.2	Mode de positionnement . . . . .	23
5.2	LIDAR . . . . .	23
5.2.1	Matériel . . . . .	24
5.2.2	Utilisation . . . . .	24
5.3	Ultrason . . . . .	24
5.4	Infrarouge . . . . .	24
5.4.1	Utilisation . . . . .	25
5.5	Capteurs de fin de course . . . . .	25
5.6	Capteur TOF . . . . .	26
5.7	Codeurs incrémentaux . . . . .	26
5.8	Caméra . . . . .	27
5.8.1	CDD . . . . .	27
5.8.2	CMOS . . . . .	28
5.8.3	DVS . . . . .	28
5.8.4	Matériel . . . . .	28
5.8.5	Utilisation . . . . .	28
5.8.6	Autres . . . . .	28
<b>6</b>	<b>Alimentation</b>	<b>31</b>
6.1	Batterie . . . . .	31
6.1.1	Lipo . . . . .	31
6.1.2	Nimh . . . . .	31
6.1.3	Chargeur . . . . .	32
6.2	Adapter l'alimentation . . . . .	32
6.2.1	BEC . . . . .	32
6.2.2	Convertisseur Buck . . . . .	33
<b>7</b>	<b>Composants électroniques</b>	<b>34</b>
7.1	Fusible . . . . .	34
7.1.1	PTC . . . . .	34
7.1.2	Fusible réinitialisable . . . . .	34
7.1.3	Thermistor . . . . .	34
<b>III</b>	<b>Programmation</b>	<b>35</b>
<b>8</b>	<b>Raspbian</b>	<b>36</b>
8.1	Le Set up . . . . .	36
8.2	Le système d'exploitation . . . . .	36
8.2.1	Les commandes utiles . . . . .	36
8.2.2	Utiliser une clef USB . . . . .	37
<b>9</b>	<b>Le code</b>	<b>38</b>
9.1	Python . . . . .	38
9.2	C/C++ . . . . .	38
9.2.1	Cross-compilation . . . . .	38
<b>10</b>	<b>Traitemet de signaux GNSS : RTK Lib</b>	<b>39</b>

<b>11 Traitement d'images : OpenCV</b>	<b>40</b>
11.1 Systèmes de couleur . . . . .	40
11.2 Odométrie visuel . . . . .	40
11.3 Vision stéréoscopique . . . . .	41
<b>12 ROS</b>	<b>42</b>
<b>13 Utilisation d'un Git</b>	<b>43</b>
13.1 Documenter . . . . .	43
13.1.1 Utilisation de fichier README.md . . . . .	43
13.1.2 Utilisation du Wiki . . . . .	43
13.1.3 Mise en forme du texte . . . . .	43
13.2 Méthodologie . . . . .	44
13.2.1 Bases . . . . .	44
13.2.2 Branches . . . . .	44
13.3 Commandes . . . . .	44
<b>14 Protocole de communication</b>	<b>46</b>
14.1 PWM . . . . .	46
14.1.1 Lire signal PWM . . . . .	46
14.2 Série . . . . .	46
14.2.1 UART . . . . .	47
14.2.2 Utilisation . . . . .	47
14.2.3 TTL . . . . .	47
14.2.4 Utilisation . . . . .	47
14.3 I2C . . . . .	47
14.3.1 Utilisation . . . . .	47
14.4 SPI . . . . .	47
14.4.1 Utilisation . . . . .	47
14.5 CAN . . . . .	47
14.6 Comparaison . . . . .	47
14.6.1 Utilisation . . . . .	47
<b>IV Automatique</b>	<b>48</b>
<b>V Outils de la menuiserie</b>	<b>50</b>
<b>15 Les machines</b>	<b>51</b>
15.1 L'imprimante 3D . . . . .	51
15.1.1 Le logiciel : Cura . . . . .	52
15.1.2 Utilisation . . . . .	52
15.2 La découpeuse laser . . . . .	52
15.2.1 Le logiciel : Inkscape . . . . .	52
<b>VI Conception</b>	<b>54</b>
<b>16 Général</b>	<b>55</b>
<b>17 CAO</b>	<b>56</b>

<b>VII</b>	<b>La Coupe de France de Robotique</b>	<b>57</b>
<b>18</b>	Pendant l'année	<b>58</b>
<b>19</b>	Le départ	<b>59</b>
<b>20</b>	La compétition	<b>60</b>
20.1	Les matchs . . . . .	60
<b>VIII</b>	<b>Autres</b>	<b>61</b>
<b>21</b>	Ressources utiles	<b>62</b>

**Première partie**

**Introduction**

Ce REX a pour rôle d'accumuler les connaissances de générations en générations. Souvent, la coupe de france de robotique demandera l'utilisation d'outils spéciaux comme des ventouses mais celles-ci ne seront pas utilisées les années suivantes et les connaissances dessus seront ainsi perdues.

Dès que vous rencontrez quelque chose qui n'est pas dans ce REX, que vous voulez faire part d'une expérience enrichissante, d'une astuce ou d'un problème récurrent, il ne faut hésiter à l'ajouter à ce REX (Objectif : il faut que ce REX soit plus épais que le poly de MMC).

Minotaure ne pourra s'améliorer que si on accumule l'expérience au fur et à mesure des années (chose difficile avec les césures).

## 0.1 Histoire

Minotaure existe depuis 2004. C'est un P03 qui a créé l'association pour pouvoir continuer à participer à la coupe de france de robotique hors de la mécatronique (il a d'ailleurs créé sa propre entreprise qui sponsorise des équipes de la coupe de france de robotique).

## 0.2 Rôle de Minotaure

Minotaure est une association officielle, au même titre que le BDE ou le TRIUM. Son objectif est de permettre aux étudiants de pratiquer la robotique aux Mines. En pratique, cela passe par gérer le budget du cours de Mécatro. En échange, l'association est financée par l'école pour participer à la coupe de France de Robotique.

## 0.3 Les différentes équipes au cours du temps

### 0.3.1 2015-2016

**Président :** Matthieu Vignes

**Trésorier :**

**Secrétaire générale :**

### 0.3.2 2016-2017

**Président :** Louis Dattin

**Trésorier :** Yoënn Burban

**Secrétaire générale :**

### 0.3.3 2017-2018

**Président :** Maxence Leroy

**Trésorier :** Ulysse Réglade

**Secrétaire générale :**

### 0.3.4 2018-2019

**Président :** Thomas Gossard

**Trésorier :** Léo Chabert

**Secrétaire générale :** Louise Fayole

### 0.3.5 2019-2020

**Président :** Corentin Lemoine

**Trésorier :** Adrien Pauron

**Secrétaire générale :**

# **Deuxième partie**

# **Composants**

# Chapitre 1

## Actionneurs

### 1.1 Moteurs pas-à-pas



FIGURE 1.1: Moteur pas-à-pas

Les moteurs pas-à-pas sont des moteurs synchrones électriques brushless à courant continu. Il en existe 3 types :

**Moteurs pas-à-pas à aimants permanents** Un aimant permanent est solidaire de l'axe du moteur (rotor). Des bobines excitatrices sont placées sur la paroi du moteur (stator) et sont alimentées chronologiquement. Le rotor s'oriente suivant le champ magnétique créé par les bobines.

**Moteurs pas-à-pas à reluctance variable** Il s'agit d'un moteur qui comporte un rotor à encoches se positionnant dans la direction de la plus faible réluctance. Ce rotor, en fer doux, comporte moins de dents qu'il n'y a de pôles au stator.

**Moteurs pas-à-pas hybrides synchrones** Ils combinent les 2 technologies précédentes, et sont plus chers. Leur intérêt réside dans un meilleur couple, une vitesse plus élevée. Vidéos explicatives des moteurs pas-à-pas hybrides : <https://www.youtube.com/watch?v=eyqwLiowZiU>

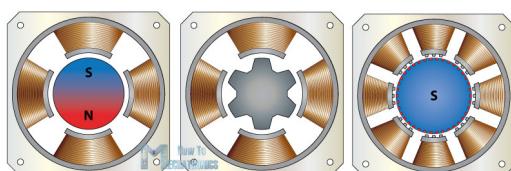


FIGURE 1.2: De gauche à droite : Aimant permanent, Reluctance Variable, Hybride

Le nombre de phase d'un moteur correspond au nombre de bobines indépendantes. Ce nombre est proportionnel à la précision du moteur. Parmi les moteurs pas-à-pas à 2 phases, il existe 2 sous-groupes :

**Unipolaire** : Les plus simples d'utilisation et moins chers. Leur connecteur est constitué de 6 fils. Chaque pôle est constituée de deux bobines enroulées en sens inverse sur les pôles du stator.

Pour changer le sens du champ magnétique dans un pôle, il faut alimenter l'une ou l'autre de ces deux bobines. Le sens du courant est constant.

**Bipolaire** : Les plus puissants. Leur connecteur est constitué de 4 fils. Chaque pôle du stator est constitué d'une seule bobine, et nécessite donc deux fils d'alimentation. Le sens du courant change tout le temps.

Les moteurs les plus courants sont ceux à aimants permanents et les hybrides.

**Utilisation** : Imprimantes (3D ou simple), photocopieurs, robotique, pousse-seringue,...

**Avantages :**

- Contrôle en position possible donc en vitesse sans avoir besoin d'une boucle fermée
- Précision
- Existence d'un couple d'arrêt

**Inconvénients :**

- Le couple maximal est inversement proportionnel à la vitesse maximal, c'est-à-dire qu'un moteur pas-à-pas sera capable de fournir le plus de couple pour de faible vitesse de rotation.
- Lent, ils ne dépassent pas les 3000 tr/min en général.
- Volumineux et lourd
- Fonctionnement plus complexe qu'un moteur à courant continu
- Résonance du moteur

**Précautions :**

- Le bobinage des moteurs pas-à-pas est très fragile. Si les câbles sont mal branchés, il peut y avoir un court-circuit entre les 2 bobinages ce qui "cassera" le moteur pas-à-pas. Il faut utiliser un ohmmètre pour distinguer les fils. 2 fils reliés à la même bobine auront une résistance tandis que d'autres connections afficheront soit rien soit une très grande résistance.
- Toujours manipuler les fils quand il n'y a pas de courant. Un court-circuit peut facilement détruire les moteurs.

Pour distinguer les différents moteurs pas-à-pas, la norme NEMA est utilisée. Cette norme indique la taille du moteur et donc sa puissance. Le nombre représente la surface de la face de l'axe du moteur en pouce carré. Ainsi, un moteur NEMA 11 aura une face d'une surface de 1,1 pouces carrés.

Le couple du moteur est proportionnel au courant.

Une bonne documentation fournira la vitesse et la puissance d'un moteur pas-à-pas. Cependant, il se peut que cela ne soit pas le cas. Dans ce genre de situation, il est possible de calculer des indicateurs sur la vitesse :

$$\text{Speed} = \left[ \frac{\text{Voltage}}{((\text{Inductance}/1000) \times 2 \times \text{Imax} \times \text{Steps})} \right]$$

$$\text{Time/Step} = \text{Inductance} \times \text{Imax} \times \left[ \frac{2}{\text{Voltage}} \right]$$

$$\text{Power} = \text{Voltage} \times \text{Imax}$$

FIGURE 1.3: Calcul de la vitesse d'un moteur pas-à-pas

Une version en ligne permet de faire le calcul directement. <https://www.daycounter.com/Calculators/Stepper-Motor-Calculator.phtml>

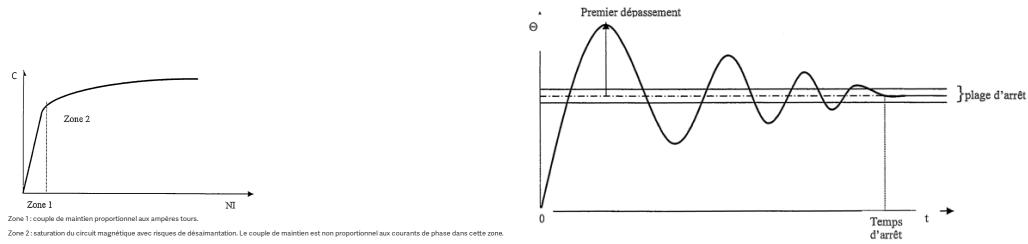
**Fournisseurs** : Stepper Online, Gotronic,...

Pour plus d'information techniques, regarder : <https://www.mdp.fr/documentation/lexique/pas-a-pas/notions-techniques.html>

### 1.1.1 Utilisation

Ce site donne une bonne présentation des moteurs pas-à-pas et comment les utiliser : <https://eskimon.fr/tuto-arduino-603-a-petits-pas-le-moteur-pas-%C3%A0-pas#se-servir-du-moteur>  
Sinon, le contrôleur de moteurs pas-à-pas utilisé pour l'édition 2019 était celui-ci :

FIGURE 1.4: Gauche : Graphique du couple de maintien en fonction du courant consommé - Droite : Réponse en angle à un échelon



<https://fr.rs-online.com/web/p/kits-de-developpement-pour-commande-de-moteur/1646982>  
Un driver en C a été écrit par Matthieu Vignes et est disponible sur le Git.

## 1.2 Moteurs à courant continu

Les moteurs à courant continu sont les plus courants. Ils ne sont cependant utilisés que dans des cas basiques.



FIGURE 1.5: Moteurs à courant continu

### Avantages :

- Très simple d'utilisation
- Taille faible pour un couple puissant

### Inconvénients :

- La loi entre la tension aux bornes du moteur et la vitesse de rotation ne sera pas la même pour des modèles identiques de moteur.

- Aucun retour d'information et contrôle en position ou en vitesse impossible en boucle ouverte.

Fournisseurs : Conrad, GoTronic, Farnell, ...

### 1.2.1 Utilisation

Pour un moteur à courant continu, la vitesse de rotation est proportionnelle à la tension à ses bornes. Cependant, étant donné qu'il est difficile de générer une tension variable, le moteur DC est le plus souvent alimenté à travers un hachoir avec une tension fixe.

Le pont L298N est la meilleure méthode pour contrôler des moteurs à courant continu : [https://wiki.mchobby.be/index.php?title=Pont-H\\_L298N](https://wiki.mchobby.be/index.php?title=Pont-H_L298N)

Il est aussi possible de choisir la tension à fournir pour contrôler le moteur avec des transistors :

<https://openclassrooms.com/fr/courses/2778161-programmez-vos-premiers-montages-avec-arduino-3285333-le-moteur-a-courant-continu-partie-1-transistors-et-sorties-pwm>

Cependant, le moteur ne pourra tourner que dans un sens avec l'utilisation de transistors.

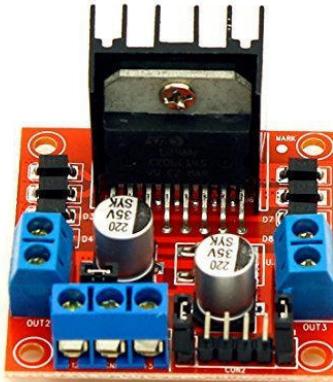


FIGURE 1.6: Pont H L298N

### 1.3 Moteurs brushless

Les moteurs brushless comme leur nom l'indique sont des moteurs à courant continu sans balais. De ce fait, ils s'usent moins rapidement que les moteurs. Ils sont utilisés en aéromodélisme pour leur endurance et la puissance qu'ils peuvent délivrer. A la place des balais, 3 câbles d'alimentations sont nécessaires pour les phases des différentes bobines (alimentation triphasée).

Pour inverser le sens de rotation d'un moteur brushless, il suffit d'inverser la connexion de 2 de ses 3 .



FIGURE 1.7: Moteur brushless

#### Avantages :

- Vitesse asservie
- Fabriqué pour tourné à de grandes vitesses

#### Inconvénients :

- Nécessité d'un ESC

**Fournisseurs :** HobbyKing ou autres sites de modélismes

#### 1.3.1 Critères de sélection

**KV :** C'est le rapport tours par minute(RPM)/tension(V).

**Taille :** Un moteur brushless sera décrit par 4 chiffres, 2208 par exemple. C'est sa taille. Les 2 premiers chiffres sont le diamètre et les 2 suivant la hauteur du moteur.

**Courant consommé :** L'ESC sera à choisir en conséquence

**Batterie :** La notice indique le type de batterie pouvant être utilisé(2S, 3S, ...). Cela revient à indiquer les tensions que le moteur supporte.

**Type :** Il existe 2 grands types de moteurs brushless : outrunner et inrunner. Cela indique si le rotor est à l'extérieur ou à l'intérieur.

**Le nombre de pôles :**

### 1.3.2 Utilisation

Pour utiliser un moteur brushless, il est nécessaire d'utiliser un ESC (Electronic Speed Controller). Un ESC va convertir une tension continu en une alimentation alternative synchrone pour le moteur à travers 3 fils. Ces 3 fils sont interchangeables mais cela changera le sens de rotation du moteur. La tension d'entrée de l'ESC n'est pas unique. Il faut juste qu'elle soit dans l'intervalle demandée. La tension n'aura pas d'influence sur la vitesse de rotation du moteur contrairement au moteur à courant continu. La vitesse de rotation est commandée par un signal PWM en entrée de l'ESC.

Certaines ESC sont équipées de BEC (Battery Eliminator Circuit), c'est à dire qu'il adapte la tension de sortie de la batterie afin de pouvoir fournir une tension utilisable par d'autres éléments électroniques comme un récepteur ou des servomoteurs. En général, un BEC sortira 5V. Voir la section sur les BEC pour plus d'information.

Ce site web explique plus en détail le fonctionnement des moteurs brushless et des ESC :<https://howtomechatronics.com/how-it-works/how-brushless-motor-and-esc-work/>



FIGURE 1.8: ESC

### 1.4 Servomoteurs

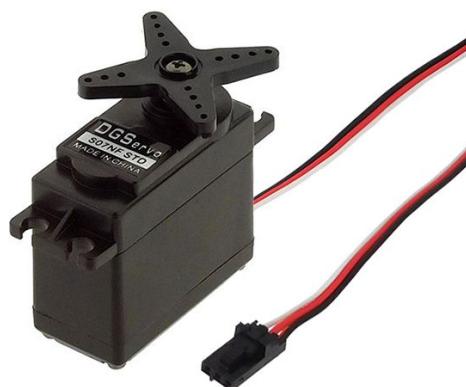
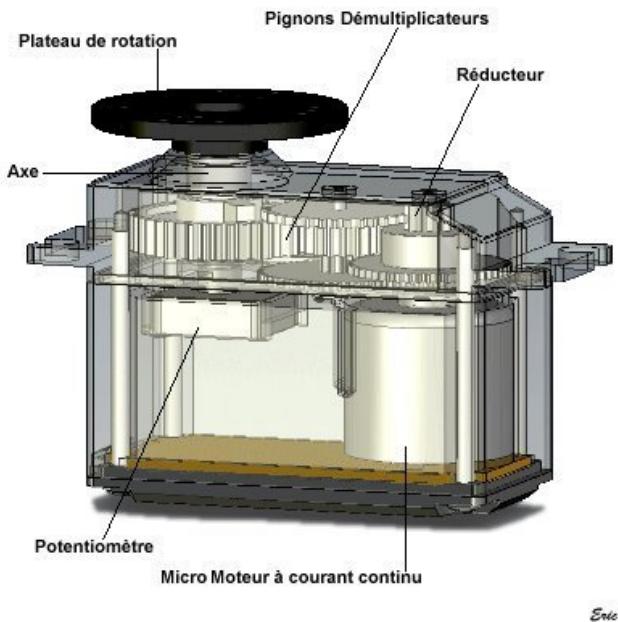


FIGURE 1.9: Servomoteurs

C'est un moteur à courant continu asservi. Il est souvent utilisé pour effectuer de petits mouvements. On le commande en angle et il maintient ensuite sa position peu importe l'effort exercé dessus (bien sûr il casse si on force trop).



Eric G

FIGURE 1.10: Schéma d'un servomoteur

Il est composé d'un moteur à courant continu, d'une réducteur, d'un potentiomètre et d'un circuit intégré. C'est le potentiomètre qui fournit le retour en angle (d'où l'impossibilité de positionner de manière absolue sur plus de 180 °)

#### Avantages :

- Simple d'utilisation
- Contrôle en position

#### Inconvénients :

- Aucun retour sur la position
- Limité à 160-180 ° de rotation (Il existe cependant des servos à rotation continue)

**Fournisseurs :** Hobbyking (et autres sites de modélisme), Gotronic, Conrad,...

### 1.4.1 Critères de sélections

**Le couple :** Celui-ci peut être exprimé dans de nombreuses unités différentes (N.m, kg.cm, oz.inch, ...).

Une fois dans le système d'unité voulu, il suffit de multiplier le couple par le bras de levier (distance entre le centre de rotation et le point d'application de la force).

**Taille :** La taille est normalement proportionnelle au couple mais en y mettant le prix, on peut avoir des servos plus petits mais plus puissants.

**Tension d'alimentation :** Généralement, les servomoteurs sont alimentés par du 5V. Cependant, pour des plus grands modèles la tension peut passer à 7.4V ou 12V.

**Courant :** Le courant consommé par le servo est proportionnel au couple demandé comme pour un moteur à courant continu normal. Il faut prévoir l'alimentation en conséquence. Si jamais il y a un pic de consommation, une chute de tension peut survenir dans tout le circuit ce qui peut provoquer l'arrêt de la carte électronique contrôlant le robot.

### 1.4.2 Utilisation

#### Cablage

Il faut 3 câbles pour utiliser un servomoteur :

- Masse
- V+ : cela dépend des servomoteurs mais ce sera en général 5V
- Signal : un signal PWM est utilisé pour commander

Servo Type	Positive (+)	Signal (S)	Negative (-)
Cirrus	Red	White	Black
Futaba - J	Red	White	Black
JR	Red	Orange	Brown
Hitec	Red	Yellow	Black
Airtronics	Red	Orange	Black
Airtronics	Red	White	Black
	Red	Black	Black
Airtronics - Z	Red	Blue	Black
Fleet	Red	White	Black
KO	Red	White	Black

FIGURE 1.11: Cablage d'un servomoteur

## 1.5 Les Dynamixels

Ce sont les actionneurs le plus souvent utilisés en robotique. Ils sont produits exclusivement par la société Robotis.

En général, ce sont les AX-12 qui sont utilisés. Attention, il faut distinguer 2 modèles d'AX-12 : AX-12W qui ont une plus grande vitesse de rotation mais un plus faible couple et les AX-12A qui sont ont un couple plus grand.



FIGURE 1.12: Dynamixel avec la clé de calibration

### Avantages :

- Retour de beaucoup d'informations : courant consommé, couple appliqué, vitesse et position,...
- Configurable
- Possibilité de les commander avec un seul bus en les connectant en série

### Inconvénients :

- Très difficile d'utilisation. Il faut utiliser un connecteur spécial et coûteux pour les débuguer.
- Plus fragiles que les servomoteurs.
- Prix (60 euros environ l'unité)

**Fournisseur :** Robotis et autres revendeurs

## Chapitre 2

# Système pneumatique

Nous avons vu jusqu'à présent que des actionneurs électriques. Cependant, il est aussi possible d'utiliser des actionneurs pneumatiques.

#### Avantages :

- Rapide
- L'actionneur peut être bloqué sans être endommagé

#### Inconvénients :

- Nécessité d'une source de pression : compresseur, pompe, cartouche de gaz,...
- Tout ou rien
- Risque de fuite ou de bouchage

## 2.1 Pompe

La pompe génère une dépression qui permettra d'aspirer des objets. En général, il s'agit un moteur à courant continu avec un système de soupapes pour générer le vide.

Les pompes trouvées en ligne sont de qualité très inégales et les spécifications décrites peuvent être fausses. Il ne faut pas hésiter à acheter différents modèles en plusieurs exemplaires.

**Fournisseur :** Les modèles déjà utilisés ont été des pompes à air de 12V trouvées sur Amazon par exemple. Roboshop vend un unique modèle qui est n'est pas mal. Cependant, il est relativement massif et surtout bruyant par rapport aux mini-pompes de 12V d'Amazon, qui font aussi bien le travail.

### 2.1.1 Utilisation

Il suffit de l'alimenter comme un moteur à courant continu. On peut très bien simplement la brancher à une alimentation 12V ou bien utiliser un pont en H si on souhaite contrôler la quantité d'air aspirer.



FIGURE 2.1: Compresseur



FIGURE 2.2: Compresseur



FIGURE 2.3: Compresseur

Quand la pompe n'est plus alimenter, elle empêchera légèrement l'air de s'échapper. Ainsi, il ne suffit pas d'arrêter d'alimenter la pompe pour qu'un objet tombe de la ventouse. Pour remédier à ce problème, il y a plusieurs solutions. On peut soit faire un léger trou dans la ventouse ou le tube. La pompe pourra toujours aspirer mais l'objet sera lâché plus rapidement. Sinon, il est possible d'ouvrir le circuit pneumatique avec des valves.

## 2.2 Ventouse

Les ventouses sont très utilisées pour se saisir d'objets ayant des surfaces relativement plane et lisse.

On distinguera surtout 2 types de ventouses : avec et sans soufflet. Le soufflet permet à la ventouse de s'adapter à l'inclinaison de la surface plane de l'objet au moment de le saisir

**Avantage :**

— Rapide

**Inconvénients :**

— Ne fonctionne que sur des surfaces planes

**Fournisseur :** RS

## 2.3 Accessoires et autres

### 2.3.1 Tubes

Il faut bien sûr aussi acheter les tubes pour tout relier. Voici les critères importants à regarder quand on achète des tubes :

- Le diamètre intérieur
- Le diamètre extérieur
- Le rayon de courbure (indique à quel point le tube peut être tordu avant de se boucher)

Le rayon de courbure est très important. Certaines pompes, même si puissantes, ne pourront pas fonctionner correctement si le tube est trop tordu. C'est pourquoi il est parfois nécessaire d'utiliser des raccords qui font des angles droits.

**Fournisseur :**Amazon, RS,...

### 2.3.2 Valves

On peut choisir d'avoir une pompe par ventouse mais cela peut très vite devenir très encombrant. Il faut alors utiliser un système de valves afin de pouvoir bloquer le tube d'une ventouse tout en ouvrant un autre tube. Il existe de vraies valves qu'il est possible d'acheter sur RS. Ces valves seront ensuite actionnées par des servomoteurs. Il est cependant aussi possible de simplement pincer les tubes avec un bras sur un servomoteur. Cela marche aussi très bien.

# Chapitre 3

## Communication

### 3.1 Radio



FIGURE 3.1: Récepteur radio

En modélisme, des radiocommandes sont utilisées pour commander avion, bateau ou voiture. De nos jours, la fréquence utilisée en majorité est la 2.4GHz car plusieurs personnes peuvent utiliser cette fréquence sans risquer d’interférences mais avant, des fréquences de l’ordre du MHz étaient utilisées. Ces fréquences dépendent du pays mais en France, on pouvait utiliser 42Mhz et 58Mhz. Ces fréquences ont l’avantage d’avoir une plus grande portée.

### 3.2 Bluetooth

On peut utiliser le Bluetooth pour faire communiquer deux appareils à courte distance. C'est en général ce qui est fait pour transmettre des informations entre 2 Raspberry Pi.

Le Bluetooth utilise 2.4GHz comme le Wifi mais il n'est pas utilisé pour les mêmes applications. Il permet un débit de données plus faibles et à une plus faible portée. En contrepartie, le Bluetooth consomme moins de courant que le Wifi. Les protocoles de communication ne sont aussi pas les mêmes, tout comme les organismes qui gèrent les normes pour le Bluetooth et le Wifi.



FIGURE 3.2: Radiocommande



FIGURE 3.3: Adaptateur bluetooth

### 3.3 Module Xbee

XBee est une marque qui produit des interfaces réseaux. Il en existe une grande diversité. On peut les distinguer par leur architecture réseau, la fréquence utilisée, le type d'antenne et la porté/puissance du signal.

#### 3.3.1 Architecture réseau

Différentes architectures réseaux ont été développées pour les modules XBee en fonction des besoins.

- 
-



FIGURE 3.4: Module XBee

# Chapitre 4

## Cartes électroniques

### 4.1 Arduino



FIGURE 4.1: Différentes cartes Arduino

Les cartes Arduino sont les cartes de prototypage les plus populaires de part leur simplicité d'utilisation et leur rapport qualité-prix. La plus répandue est la carte Arduino Uno.

### 4.2 Raspberry Pi

C'est un micro-ordinateur. Elle est donc beaucoup plus puissante que la Teensy ou les cartes Arduino. Cependant, elle est aussi limité dans ses capacités.

#### Avantages :

- Ports USB
- Wifi et bluetooth intégrés pour la version 3B+
- Plus de puissance de calcul
- Possibilités de programmer dans n'importe quel langage.

#### Inconvénients :

- Il est difficile de générer des signaux PWM
- Impossible de lire des signaux PWM
- Seulement une vrai liaison série disponible et encore. Elle est normalement dédiée au bluetooth mais on peut l'utiliser pour autre chose si on utilise pas le Bluetooth. Il y a aussi une autre pseudo liaison série mais elle est très limitée.

#### 4.2.1 Ports USB

#### Interface série

#### Courant

### 4.3 Teensy

La teensy est en résumé une version miniature d'une carte arduino, en plus puissante. Elle possède notamment plus de sortie série. Elle se programme exactement de la même manière qu'une carte arduino.

## 4.4 Node Mcu

C'est une carte équipé comme la Teensy mais elle a une puce Wifi en plus. C'est la carte de prototypage utilisée pour les applications IoT.

## 4.5 Shield

Les différentes cartes électroniques ne possèdent pas tous les capteurs ou sorties nécessaires pour un projet. Pour compenser ces problèmes, des Shield ont été créés. Ce sont des extensions des cartes électroniques très faciles d'utilisation. Cela coûte bien sûr moins cher de les faire soit même mais on est certain au moins que c'est fiable avec un shield.

Différents shields :

- GNSS
- Servo

# Chapitre 5

## Capteurs

### 5.1 GNSS

GNSS (Global Navigation Satellite System) désigne un ensemble de composants reposant sur une constellation de satellites artificiels permettant de fournir à un utilisateur par l'intermédiaire d'un récepteur portable de petite taille sa position 3D, sa vitesse et l'heure.

Il est important de distinguer GPS et GNSS. GPS (Global Positionning System) correspond à la constellation de satellites des Etats-Unis qui appliquent le principe de GNSS. C'est un abus de langage.

**Avantages :**

- Position absolue
- Donne aussi la vitesse du véhicule (dans)

**Inconvénients :**

- La précision des mesures dépend fortement de la qualité du signal. En milieu urbain, les immeubles bloquent certains satellites et peuvent générer des multipaths.

**Fournisseurs :** Ublox, Emlid,...

Il existe différents modes de positionnement par GNSS aujourd'hui dont certains peuvent donner une précision allant jusqu'au centimètre.

#### 5.1.1 NMEA

#### 5.1.2 Mode de positionnement

**Single**

**PPP**

**Différentiel**

### 5.2 LIDAR

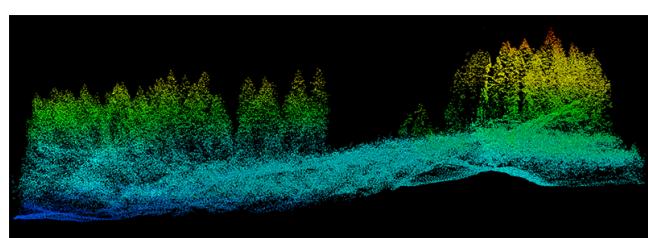


FIGURE 5.1: LIDAR de SLAMTEC

Acronyme pour Light Detection And Ranging, ce capteur permet de générer un nuage de point de l'environnement qui l'entoure. Le LIDAR fait cela grâce à un laser qui mesure la distance avec un obstacle. Contrairement à un capteur infrarouge actif, le laser du LIDAR est très directif et permet

des mesures plus précises. En connaissant la configuration du LIDAR, on connaît la direction du laser et donc la direction de l'obstacle. Les LIDARs que nous utilisons sont font juste des mesures dans un plan mais il existe des LIDARs qui fournissent des nuages de points en 3D.

Un nouveau type de LIDAR est un train de faire son apparition : les Solid State Lidar. Il génère toujours un nuage de points mais il ne fonctionne pas sur le même principe, il n'y pas notamment plus de pièces mobiles.



### 5.2.1 Matériel

**Modèle :** A2M8

**Quantité :** 2

**Fabricant :** Slamtech

**Fournisseur :** Robotshop

### 5.2.2 Utilisation

**Positionnement :** En utilisant des techniques de SLAM, il est possible d'utiliser des balises posées sur le bord du terrain pour obtenir la position et l'orientation du robot. Le problème est que la résolution des LIDAR que Minotaure possède n'est pas suffisante pour détecter les balises partout sur le terrain.

**Détection d'obstacles :** Il est possible de détecter les robots adverses à partir du nuage de points. Pour cela, il faudrait faire une détection de blobs, qui correspondraient aux robots.

## 5.3 Ultrason

## 5.4 Infrarouge

On distingue 2 types de capteurs infrarouges, les capteurs passifs qui mesurent juste la lumière ambiante et les capteurs actifs qui génèrent aussi des ondes infrarouges. On pourrait aussi parler des caméras infrarouges, mais ce sont simplement des caméras qui mesurent captent les infrarouges en plus des ondes du domaine visible.

**Avantages :**

- Peu cher
- Simple à mettre en oeuvre

**Inconvénients :**

- Très sensible à la lumière ambiante, les éclairages traditionnels émettent aussi des infrarouges qui peuvent induire le capteur en erreur.



FIGURE 5.2: Capteurs infrarouges Actif (gauche) et passif (droite)

#### 5.4.1 Utilisation

Le capteur infrarouge actif est en général utilisé en tant que capteur d'obstacle. En fonction de la quantité d'ondes infrarouges reçues, on peut estimer la distance de l'obstacle. Le problème est que cette quantité varie en fonction de l'éclairage de l'environnement (notamment avec des projecteurs). Pour compenser ce problème, il est nécessaire de calibrer les capteurs infrarouges à chaque lancement.

La phase de calibration peut consister définissant le taux moyen d'infrarouge dans l'environnement en mesurant sur une certaine période ou en redéfinissant la valeur limite.

### 5.5 Capteurs de fin de course

Comme son nom l'indique, le capteur de fin de course permet d'observer les contacts. Ils sont aussi appelés interrupteurs ou détecteurs de position. Ce sont en soit des interrupteurs poussoirs avec des

## 5.6 Capteur TOF

Les capteurs TOF (Time Of Flight) regroupent différents types de capteurs. Ils ont tous en commun la mesure de l'aller-retour d'un signal pour mesurer une distance. C'est en soi le même principe que pour les capteurs ultrasons ou infrarouges. Cependant, on qualifie en particulier 2 capteurs de cette manière : des caméras qui mesurent la profondeur sur le même principe que la Kinect 2 et des capteurs de distance utilisant un laser, à distinguer des LIDAR.

ST a développé un capteur sur ce principe à prix très abordable. La technologie s'appelle FlightSense. De part la précision, la taille et la robustesse de ce capteur, il rend les capteurs ultrasons et à infrarouges obsolètes.

World's smallest ToF ranging sensor



FIGURE 5.3: Fonctionnement du capteur FlightSense de ST

## 5.7 Codeurs incrémentaux



FIGURE 5.4: Codeur Incrémental

Les capteurs incrémentaux mesurent une rotation.

Il existe des codeurs incrémentaux absous et relatifs. Cependant, les codeurs absous sont très chers. En effet, chaque position doit avoir son propre code ce qui demande des gravures de très

grandes précisions. En général, un codeur incrémental relatif est utilisé avec une phase d'initialisation lors de son démarrage avec un capteur de buté.

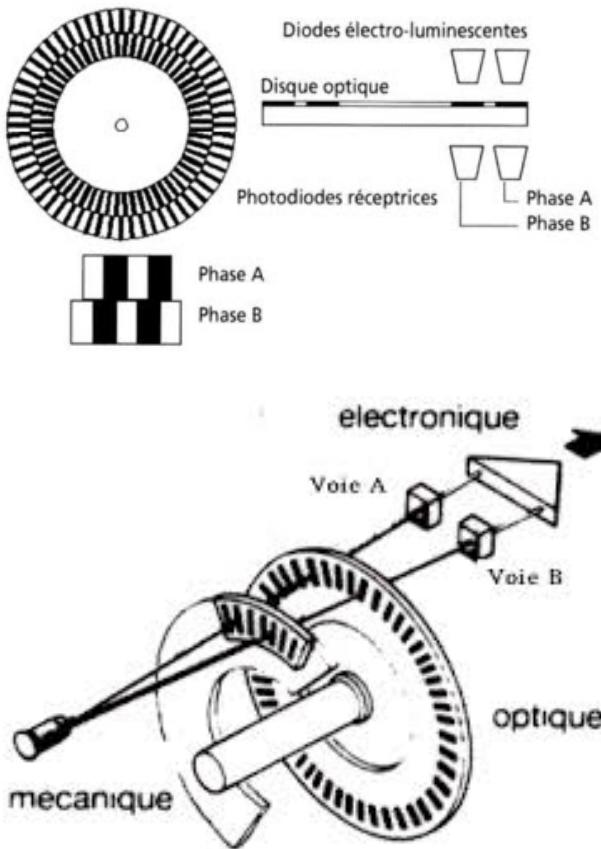


FIGURE 5.5: Schéma de codeurs incrémentaux

## 5.8 Caméra

Il existe 2 types majoritaires de capteurs photographiques : CDD et CMOS. Cependant, un nouveau type de capteur fait se popularise dans le milieu de la robotique : DVS (Dynamic Vision Sensor).

Caractéristiques métrologiques	Capteur CCD	Capteur CMOS
Résolution	Comparable	Comparable
Taille des pixels	Comparable	Comparable
Sensibilité	Elevée	Moyenne
Bruit	Faible	Elevé
Risques de saturation	Elevée	Faible
Rapidité	Grande	Très grande

FIGURE 5.6: Différences entre capteurs CDD et CMOS

Pour plus d'information sur la différence technique entre CMOS et CDD : <https://www.photoniques.com/articles/photon/pdf/2015/06/photon201579p39.pdf>

Pour plus d'information sur les capteurs DVS : <http://rpg.ifi.uzh.ch/docs/EventVisionSurvey.pdf>

### 5.8.1 CDD

CDD signifie Charged Coupled Device. C'est un capteur avec de meilleures performances mais beaucoup plus coûteux. Il est notamment utilisé dans l'industrie du cinéma. Il consomme aussi plus de

puissance ce qui fait qu'il chauffe aussi plus rapidement. Il faut le refroidir à moins de voir apparaître des signaux parasites.

### 5.8.2 CMOS

CMOS signifie Complementary Metal Oxyde Semiconductor. C'est la technologie la plus récente. Au départ, les CDD possédaient de meilleures performances mais de plus en plus, cet écart se réduit. Le facteur le plus distinctif entre les 2 caméras est la sensibilité à la lumière. Dans un endroit peu éclairé, il y aura beaucoup de bruit sur une image prise avec un capteur CMOS.

### 5.8.3 DVS

Un nouveau type de caméra fait son apparition : DVS (Dynamic Vision Sensor). Ce type de caméra n'enregistre pas une image à chaque pas de temps. Elle suit le même principe que les récepteurs lumineux dans nos yeux et enregistrement seulement les variations de luminosité. Ces caméras ont beaucoup de potentiel. En effet, puisque seulement la variation est enregistrée, il n'y a pas d'informations redondantes et les programmes peuvent être plus rapides. De plus, ces caméras sont plus aptes à enregistrer les mouvements rapides de part leur nature. Contrairement aux capteurs CDD et CMOS, il n'y a pas " de perte d'information" entre 2 images. Problème : les DVS coûtent très chères et ont encore une très faible résolution pour le prix demandé. Cependant, en terme d'application pour la robotique, c'est le futur.

### 5.8.4 Matériel

En général, des PiCameras sont utilisés pour les projets demandant de la vision. Il faut distinguer les 1ère versions (V1) et les 2ème versions (V2). Il existe aussi des Picameras qui captent aussi les infrarouges afin d'avoir une vision nocturne (une lampe infrarouge est cependant nécessaire en plus). Dans la plupart des cas, cela suffira.

### 5.8.5 Utilisation

La picamera est facilement utilisable avec Python. Du C++ peut aussi être utilisé pour la commander mais c'est beaucoup plus complexe.

Toutes les informations nécessaires sont présentes sur ce site : <https://picamera.readthedocs.io/en/release-1.13/>

#### Possibilités :

- Choisir la résolution de l'image et le FOV (Field Of View)
- Encodage utilisé
- Paramètre caméra (saturation, nombre d'image par seconde si on prend une vidéo, ...)

### 5.8.6 Autres

Intel a développé des caméras avec des cartes intégrés pour exploiter les données des caméras directement. La marque s'appelle RealSense.

#### T265

Stéréo caméra avec des lentilles fisheyes. Elle ne mesure pas la profondeur d'une image de manière innée. Il faut exploiter les caméras avec OpenCV pour cela. Elle est par contre spécialisé dans le SLAM. La T265 peut mesurer son déplacement et son attitude avec une très grande fiabilité et précision.

#### D435

C'est une stéréocaméra infrarouge qui permet de mesurer la profondeur d'une image. Il y a aussi une caméra RGB pour avoir une image en couleur en plus.

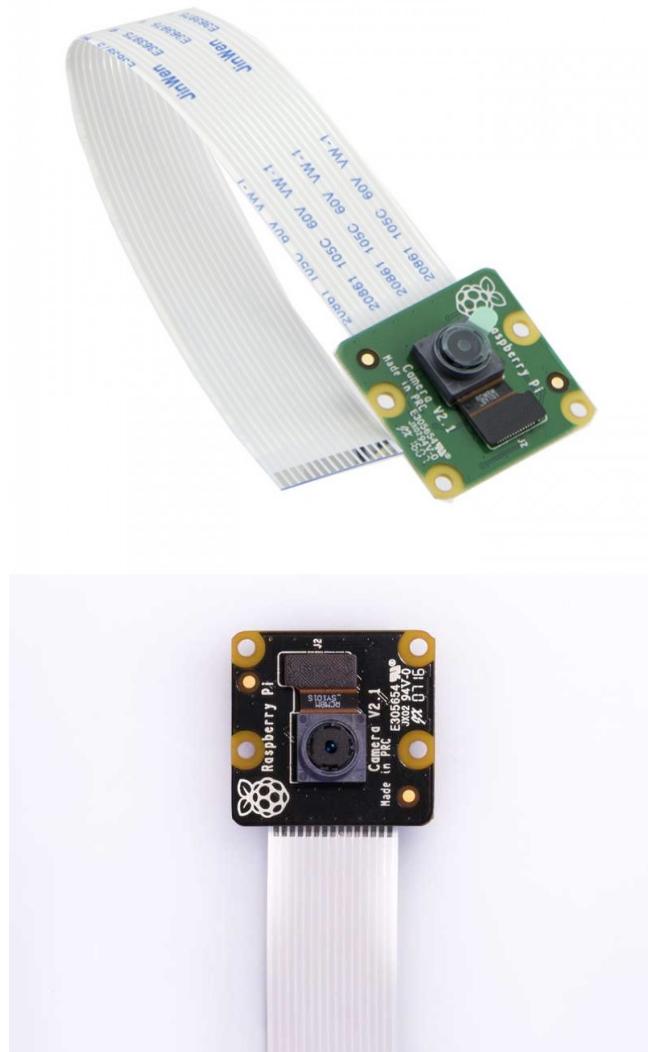


FIGURE 5.7: Picamera et Picamera Noir



FIGURE 5.8: T265 d'Intel RealSense



FIGURE 5.9: D435 d'Intel RealSense

# Chapitre 6

## Alimentation

### 6.1 Batterie

Il existe 2 types majoritaires de batterie chimique. Les Lithium Polymère (LiPo) et les Nickel Mercure (Nimh).

#### 6.1.1 Lipo

Ces batteries présentent la meilleure densité énergétique. Cependant, elles sont aussi très dangereuses. Elles peuvent en effet exploser et provoquer un feu chimique difficilement arrêtable.

Les choses suivantes peuvent causer l'explosion d'une LiPo.

- Mouillée
- Percée
- Reçoit un choc
- Laissée au soleil
- Court-circuit

Si jamais la LiPo est bombée ou a des bosses, il faut absolument s'en débarasser. En attendant, la meilleure solution est de la placer dans un sac en métal ou sur une surface non-inflammable (pierre, métal, ...). Ce type de batterie doit être jeté à la déchèterie dans la section réservée pour.

#### Les critères de sélection

**Taux de décharge :** Indique combien d'ampère la batterie peut fournir en continu. Le taux de décharge est mesuré en C. Il faut appliquer la formule :  $CourantMax = C * \frac{Capacité de la batterie (mAh)}{1000}$ .

Le taux de décharge est souvent surestimé. Prendre un taux de décharge trop petit détériore la batterie et celle-ci pourra gonfler.

**Nombre d'élément :** Correspond à la tension de la batterie. C'est mesuré en S.

#### Les bonnes marques

- Tattu
- Turnigy
- Zippy

Liste non-exhaustive.

#### 6.1.2 Nimh

Les batteries Nimh sont moins fragiles que les batteries Lipo. Cependant, elles ont une plus faible densité énergétique.

### 6.1.3 Chargeur

Les différents types de batterie ne se chargent pas de la même façon. Il y a un mode pour chaque sur le chargeur. Il faut aussi surveiller le taux de charge.



FIGURE 6.1: Chargeur de batterie

Pour les Lipo, il faut aussi brancher la prise d'équilibrage. Ce n'est pas obligatoire mais cela permet de prolonger la durée de vie de la batterie. En effet, au fur et à mesure des différents cycles de charge/décharge, les éléments de la Lipo ne vont pas avoir tous la même tension. La prise d'équilibrage permet d'empêcher cela.

Les batteries ont en général un taux de recharge de 1C. Le courant pour recharger la batterie est à définir sur le chargeur. Il faut appliquer la même formule que pour le courant de taux de décharge.

## 6.2 Adapter l'alimentation

### 6.2.1 BEC

Le BEC (Battery Eliminator Circuit) permet de délivrer une tension. Ce circuit appartient à l'électronique de puissance.

Dans la plupart des ESC de nos jours, un BEC est intégré et fournit du 5V.

Il faut distinguer plusieurs BEC :

**BEC** : Les BEC de base. Ils utilisent des régulateurs de tension comme expliqué dans la section suivante.

**SBEC** : Switch BEC. Ce sont les BEC qui convertissent toutes la puissance. Le Switch fait référence à comment le signal entrant est traité pour être transformé. Ils utilisent un convertisseur Buck.

**UBEC** : Universal BEC. C'était/c'est une marque de BEC. Il fonctionne sur le même principe que les SBEC, mais il peut y en avoir qui fonctionne encore avec des régulateurs de tension.

### Régulateurs de tension linéaire

Les régulateurs permettent comme les BEC de fournir une tension inférieure à la tension d'alimentation. Cependant, contrairement à un BEC, toute la puissance n'est pas convertie. Une grande partie est perdue par effet joule. C'est pourquoi il ne faut l'utiliser que quand la puissance à dissiper est faible ( $P_{dissipée} = (V_{entrée} - V_{sortie}) * I$ ).

**Inconvénients :**

- Pas efficace, la tension a dissipé l'est sous la forme de chaleur
- Cesse de d'alimenter en tension si la température dépasse un certain seuil
- Chauffe beaucoup si il y a beaucoup de puissance à dissiper

**Avantages :**

- Simple à mettre en oeuvre
- Peu cher (de l'ordre de la dizaine de centimes)

### 6.2.2 Convertisseur Buck

C'est ce qu'il faut choisir quand il y a beaucoup de puissance à dissiper (tous les cas sauf quand il faut alimenter un micro-contrôleur en 3.3V en partant de 5V par exemple).

**Inconvénients :**

- Prix de l'ordre de l'euro
- Utilise une bobine donc peut être source d'interférences électromagnétiques

**Avantages :**

- Fort rendement

## Chapitre 7

# Composants électroniques

### 7.1 Fusible

7.1.1 PTC

7.1.2 Fusible réinitialisable

7.1.3 Thermistor

# **Troisième partie**

# **Programmation**

# Chapitre 8

## Raspbian

### 8.1 Le Set up

En règle général, il n'y aura pas d'écrans, de claviers ou de souris à disposition pour pouvoir utiliser directement la raspberry pi avec son ordinateur. Pour contourner ce problème, Matthieu Vignes(P14) a créé une configuration spéciale de l'image Raspbian. Elle fait en sorte que dès le premier démarrage, la raspberry pi émet son propre réseau wifi sur lequel il est possible de se connecter pour pouvoir utiliser la raspberry pi.

La méthodologie pour créer cette image est disponible sur son Github : [https://github.com/matthieuvigne/MiAM\\_eurobot2019/tree/master/ConfigRPi](https://github.com/matthieuvigne/MiAM_eurobot2019/tree/master/ConfigRPi)

**Windows** : Il faut installer Putty pour avoir accès au terminal de la raspberry pi. Sinon, il est possible d'avoir accès à la GUI avec VNC viewer.

**Linux** : Il faut utiliser les lignes de commandes.

**sshfs** : Monter sur son système de fichier, un autre système de fichier distant, à travers une connexion SSH . En gros, il est possible d'avoir accès sur l'ordinateur à tous les fichiers de la raspberry pi. Les modifications sur l'ordinateur seront aussi réalisées sur la raspberry pi.

**ssh** : Crée une liaison SSH avec la raspberry pi, ce qui permet d'avoir accès à son terminal.

**scp** : Copie les fichiers de la raspberry pi à l'ordinateur et vice-versa par liaison SSH.

Matériel utile :

**Clef USB Wifi** : Permettra à l'ordinateur de se connecter à 2 réseaux Wifi en même temps : Celui de la raspberry pi et à une connection internet.

**Cable ethernet** : La raspberry Pi utilise déjà le wifi pour communiquer avec l'ordinateur. Il est alors nécessaire d'utiliser le cable ethernet pour pouvoir réaliser des mises-à-jours logiciel. Il faut pour cela faire un partage de connection par ethernet avec l'ordinateur.

### 8.2 Le système d'exploitation

Raspbian est un système d'exploitation basé sur Linux. Il est possible d'installer 2 versions sur la Raspberry Pi : la version lite (sans interface graphique), la version graphique (beaucoup plus lourde et gourmande en ressource). En règle générale, il es préférable d'installer la version Lite, cela laissera plus de puissance de calcul pour les programmes qui tourneront dessus.

#### 8.2.1 Les commandes utiles

**cd** : Changer de répertoire. Taper juste "cd" permet de répertoire /home/user. Pour accéder à la racine, il faut taper "cd /".

**ls** : Liste tous les fichiers dans un répertoire

**mkdir** : Crée un répertoire

**nano** : Affiche le contenu d'un fichier et permet de le modifier.

**vim** : Même chose que nano mais en plus puissant et moins intuitif.

**chmod** : Change les permissions d'un fichier. Certains fichiers ne pourront pas être modifiés ou lancés car l'utilisateur n'aura pas le droit.

**apt install** : Installe une librairie.

**apt upgrade** : Télécharge les versions actuelles des librairies.

**apt update** : Met-à-jour les librairies.

Pour lancer un programme binaire, il faut "./programme"

### 8.2.2 Utiliser une clef USB

Une clef USB ne peut pas être utilisé directement avec Linux, il faut la monter. Dans les OS avec interface graphique, cette étape se fait automatiquement mais pas si on utilise la version Lite de Raspbian.

Trouver le périphérique : lsblk

Le périphérique SUB apparaîtra dans le dossier /dev en général avec le nom sdb1. Il faut alors créer un dossier dans /media qui correspondra à notre clef USB.

```
sudo mkdir /media/usb  
sudo mount /dev/sdb1 /media/usb  
sudo umount /media/usb
```

# Chapitre 9

## Le code

Quel langage utilisé ?

### 9.1 Python

Python, le langage que tout le monde maîtrise normalement après la prépa. Il est très facile de faire du prototypage avec. C'est d'ailleurs avec ce langage qu'il faudra tester de nouvelles choses. Cependant, en tant que langage interprété il est beaucoup plus lent que les langages compilés (C/C++) .

En général, c'est le langage utilisé lors de la phase de prototypage. Une fois le principe vérifié, le code embarqué est plutôt en C ou C++.

### 9.2 C/C++

En tant que langages compilés, ils sont beaucoup plus rapides que Python. Cependant, ils sont aussi beaucoup plus complexe d'utilisation.

#### 9.2.1 Cross-compilation

Il est possible de compiler les programmes en C et C++ sur la raspberry pi mais cela prends beaucoup de temps en raison du manque de puissance de calcul (de l'ordre des minutes). Cependant, une fois compiler une première fois, le compilateur ne devra recompiler que les fichiers modifiés. La compilation prendra alors moins de temps.

En raison de l'architecture différente des processeurs sur l'ordinateur et sur la Raspberry Pi (ARM), il n'est pas possible de compiler les programmes sur l'ordinateur puis de les copier sur la Raspberry Pi. Il est cependant possible de simuler l'environnement de la raspberry pi pour la compilation sur ordinateur en utilisant la cross-compilation.

## Chapitre 10

# Traitement de signaux GNSS : RTK Lib

RTK Lib est un ensemble de programmes créés pour pouvoir exploiter et analyser les signaux obtenus à partir de récepteurs GNSS. Il existe un ensemble avec une interface graphique utilisable sous Windows. Il existe aussi une version UI utilisable sous n'importe quel système d'exploitation après avoir compiler les programmes en C.

Une branche a été créé à partir de RTK Lib : RTK Lib Explorer. Cette branch a été optimisée pour certains récepteurs et fonctionne en général mieux(en tout cas dans les cas vus).

# Chapitre 11

## Traitements d'images : OpenCV

OpenCV est la bibliothèque par excellence pour tout ce qui concerne le traitement d'images.

### 11.1 Systèmes de couleur

De part la manière dont nous voyons le monde, nous utilisons en général le système RGB pour décrire les couleurs. Cependant, ce n'est pas toujours le système de couleur le plus efficace pour étudier des images. En effet, certains éléments ressortiront plus dans un certain système de couleur. Par exemple, les feux de forêt sont plus distinct avec la luminescence du système LUV qu'avec le rouge de RGB.

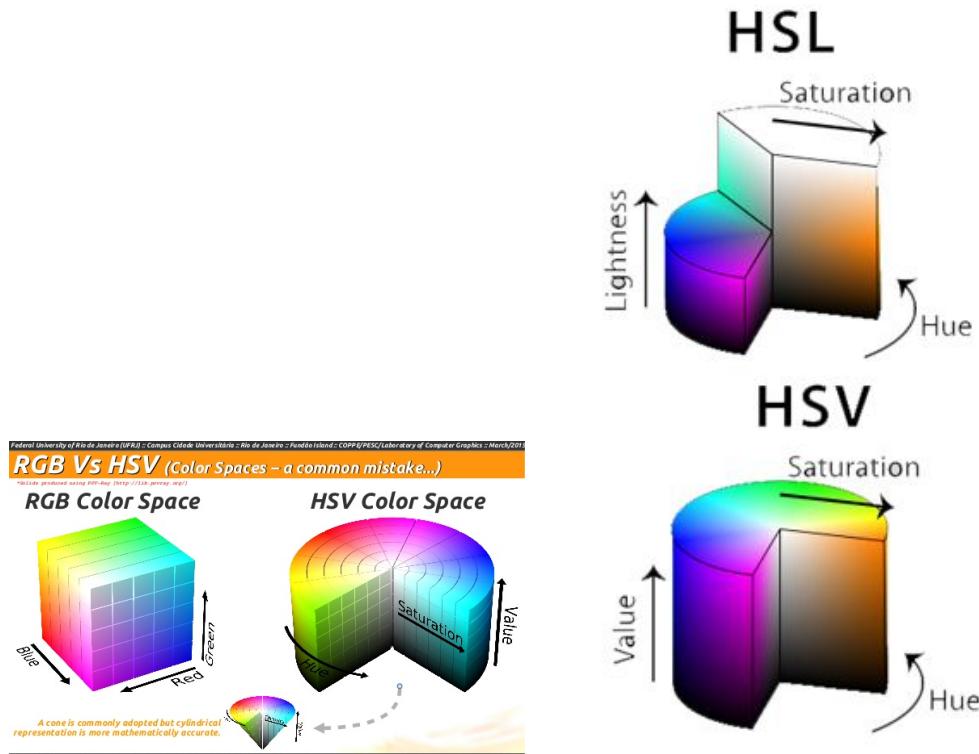


FIGURE 11.1: Systèmes de couleur

### 11.2 Odométrie visuel

Ces algorithmes permettent de mesurer le déplacement effectué entre 2 images. On peut utiliser une seule ou même deux caméras.

<https://link.springer.com/article/10.1186/s40064-016-3573-7>

### 11.3 Vision stéréoscopique

A partir de deux caméras dont on connaît la position relative, il est possible d'obtenir la profondeur de l'image. Cependant, cette méthode pour obtenir la profondeur est très sensible à différents paramètres tels que la calibration des caméras et la distance entre les caméras. Par ailleurs, la carte de profondeur générée est très approximative. Si on veut vraiment mesurer la profondeur d'une image, on utilisera plutôt une caméra infrarouge telle que la Kinect ou la Intel Realsense D435.



FIGURE 11.2: Profondeur d'image avec des stéréos caméras

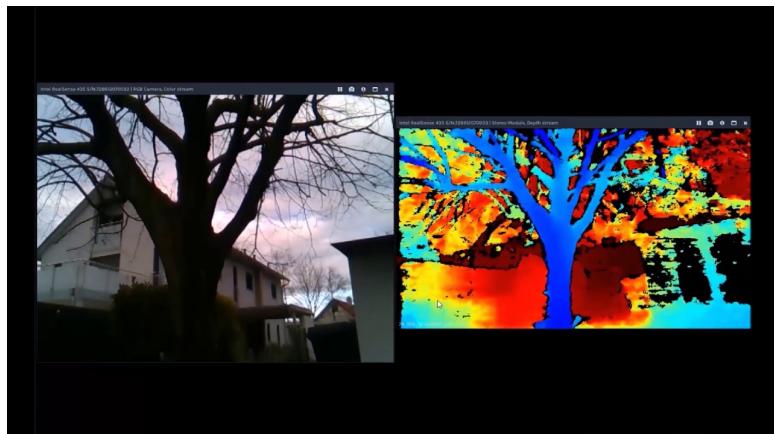


FIGURE 11.3: Profondeur d'image avec RealSense D435 utilisant une caméra infrarouge

# Chapitre 12

## ROS

ROS est une sorte de système d'exploitation pour la robotique.

Pourquoi l'utiliser :

- Une grande communauté
- Beaucoup de bibliothèques déjà développées
- Code beaucoup plus modulable
- Possibilité de programmer un C/C++ ou en python (autres langages supportés aussi de manière non officielle)
- Il est très souvent utilisé dans le milieu de la robotique

Cependant, ROS demande une certaine prise en main. Pour apprendre à l'utiliser, il a plusieurs solutions. Personnellement, je conseille ce livre disponible en version PDF "ROS Robot Programming Book". Il rassemble tout ce qui est nécessaire de savoir.

Cependant, pour avoir de la pratique, investir dans la plateform IgniteAcademy est une très bonne idée. Un abonnement au mois coûte 39 euros. Je l'ai essayé et ça vaut vraiment le coût.

# Chapitre 13

## Utilisation d'un Git

Un Git est très pratique pour coder. Il permet travailler à plusieurs sur un même projet et surtout il permet de gérer l'évolution du code.

Il existe plusieurs services qui propose hébergements en ligne : Github, Gitlab,... Minotaure utilise GitHub.

### 13.1 Documenter

Il existe deux manières de documenter le code sur GitHub. On peut soit écrire des fichiers README.md ou rédiger des pages Wiki. L'utilisation de l'un ou de l'autre va seulement changer ou trouver l'information ainsi que son organisation. En effet, il est seulement possible d'avoir un fichier README.md par répertoire mais on avoir autant de pages wiki organisées comme on veut.

#### 13.1.1 Utilisation de fichier README.md

Il est possible dans chaque répertoire de créer un fichier README.md. Il s'agit d'un fichier texte mis-en-forme (Markdown Documentation). Ainsi, quand le répertoire sera ouvert sur le site Github, le texte mis-en-forme sera affiché automatique sous les fichiers du répertoire.

#### 13.1.2 Utilisation du Wiki

Chaque github offre la possibilité de créer son propre wiki. Il est possible d'agencer les pages wiki comme on le souhaite. La hiérarchie est la même que pour des fichiers normaux : "repertoire1/repertoire2/pageWiki3" par exemple (c'est ce qu'il faut écrire comme nom de page lors de sa création).

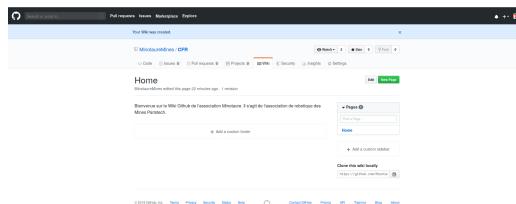


FIGURE 13.1: Wiki du Git

#### 13.1.3 Mise en forme du texte

Il est possible de mettre en forme le texte dans le wiki ou le README. Pour cela, le markup language est utilisé. Il est très simple à utiliser et rend les choses lisibles.

**Titres :** Il faut mettre # avant le titre (Attention de ne pas oublier un espace entre le # et le titre). Le nombre de # indique l'importance du titre : # pour les titres les plus importants, ##### pour les moins importants.

**Liste :** Les listes sont indiquées par des \*, + ou - au début de la ligne pour chaque élément.

**Italique :** \*texte en italique\* ou texte en italique

**Gras :** \*\*texte en gras\*\* ou texte en gras

**Barré :** texte barré

**Liens :** [Nom du lien](lien url)

**Images :** ![nom image](path vers image ou url)

**Code :** “languageDeProgrammation “

## 13.2 Méthodologie

### 13.2.1 Bases

1. Avant de commencer, il faut récupérer la dernière version disponible du code.
2. Une fois satisfait avec les modifications réalisées sur un fichier et avoir vérifier que cela fonctionne, on commit (enregistre) la nouvelle version (on n'enregistre que très rarement un code qui ne fonctionne pas). Il faut mettre un commentaire au commit afin de garder une trace des modifications réalisées.
3. Quand on a finit de travailler sur le code, on push (sauvegarde) les modifications sur le serveur.

Il est très important de qu'un commit corresponde à une seule modification afin de garder une trace des modifications. Cela permettra un débugage plus rapide par la suite. Commiter plusieurs modifications d'un seul coup est à proscrire !

### 13.2.2 Branches

Parfois, on veut tester plusieurs idées pour améliorer un code. Pour éviter de tout mélanger et d'avoir plusieurs fichiers qui font presque la même chose, on utilise des branches. Il s'agit d'une version alternative de tout le code qu'il est possible de développer en parallèle de la branche principale. Une fois satisfait des améliorations réalisées, on peut merger (fusionner) la branche au master (branche principale). Le code des deux branches est alors fusionner.

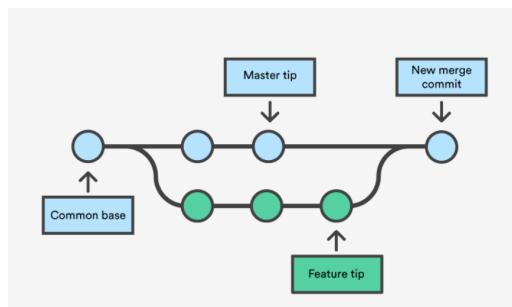


FIGURE 13.2: Principes des branches sur Git

## 13.3 Commandes

Voici une liste non-exhaustive des commandes git de base. Chaque commande possède des options d'utilisation qui ne sont pas décrites ici. Pour plus d'information, chercher sur internet.

**git config :** Définit les paramètres utilisateurs comme le nom d'utilisateur et l'adresse mail.

git config --global user.name “[name]”

git config --global user.email “[email address]”

**git init :** Initialise un répertoire git.

git init [repository name]

**git clone** : Clone un répertoire sur internet à partir du lien url de celui-ci.

git clone [url]

**git add** : Enregistre les modifications des fichiers en vu d'un prochain commit.

git add [file]

**git commit** : Commit les modifications enregistrées avec git add.

git commit -m "[Commentaires]"

git commit -a (Commit tous les fichiers modifiés et déjà suivis par git, pas besoin d'utiliser git add)

**git reset** : git reset [file] (Enlève le fichier des fichiers à commit)

**git staus** : Liste tout les fichiers modifiés qui n'ont pas été enregistrés pour le commit

**git rm** : Efface un fichier

git rm [file]

**git log** : Montre l'historique des commits dans la branche locale.

**git push** : Sauvegarde les commits en ligne.

git push origin master (Si tu travailles sur la branche principale)

**git pull** : Met-à-jour les fichiers locaux par rapport à leur version en ligne.

# Chapitre 14

## Protocole de communication

Pour faire communiquer différents appareils électroniques entre eux, il existe plusieurs méthodes.

### 14.1 PWM

Il ne s'agit pas d'un protocole de communication à part entière. C'est cependant la méthode la plus basique et facile à comprendre.

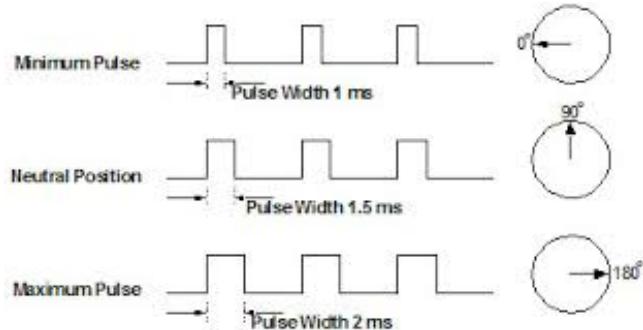


FIGURE 14.1: Signal PWM pour contrôler un servomoteur

La durée d'un signal PWM (Pulse Width Modulation) est proportionnelle à la commande. En général, la fréquence est de 50 Hz pour les servomoteurs mais elle peut être différente pour d'autres applications.

C'est le signal utilisé pour contrôler aussi les ESC des moteurs brushless et beaucoup de capteurs basique codent l'information en PWM.

#### 14.1.1 Lire signal PWM

Générer un signal PWM est relativement simple. Les lire est une autre paire de manches. On ne peut pas en effet prévoir quand le signal changera d'état et les microcontrôleurs n'ont pas d'entrée exprès pour les PWM. Il faut donc avoir recours aux interrupts. Le principe revient d'écouter une entrée et d'attacher une fonction qui se déclenchera dès que l'entrée changera d'état (on peut choisir front montant ou descendant). Afin de pouvoir mesurer le temps sans délais, les interrupts interrompent tous les autres processus ce qui peut faire buguer le programme principal. Si c'est le cas, il faut soit prendre un microcontrôleur plus rapide ou avoir un microcontrôleur dédié pour lire les PWM (les valeurs seront ensuite communiquées au microcontrôleur principal par liaison série).

### 14.2 Série

Il existe 2 protocoles de communications en série qu'il est facile de confondre. Ils fonctionnent en effet sur le même principe mais pas avec les mêmes tension de référence. C'est pourquoi il est important de bien identifier quel est le protocole série utilisé.

#### 14.2.1 UART

#### 14.2.2 Utilisation

#### 14.2.3 TTL

#### 14.2.4 Utilisation

### 14.3 I2C

#### 14.3.1 Utilisation

### 14.4 SPI

#### 14.4.1 Utilisation

### 14.5 CAN

Ce protocole de communication a été développé par Bosch pour réduire la quantité de câble dans les voitures. Il permet en effet de faire communiquer plusieurs appareils avec seulement 2 fils.

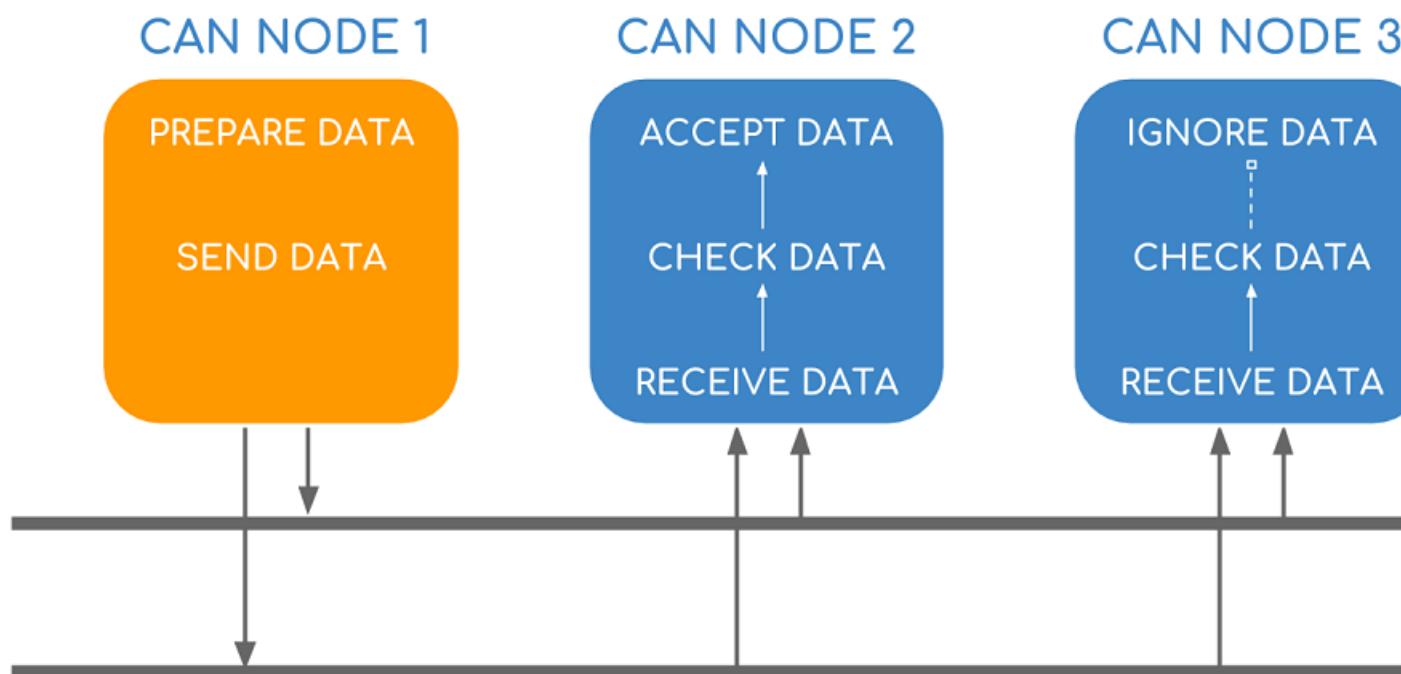


FIGURE 14.2: Principe CAN

### 14.6 Comparaison

#### 14.6.1 Utilisation

# **Quatrième partie**

# **Automatique**

Le livre "Probabilistic Robotics" est excellent pour découvrir les différents problèmes de robotique. Il sert de très bonne introduction aux algorithmes de fusion de données pour estimer la position, au contrôle de robots et au SLAM.

**Cinquième partie**

**Outils de la menuiserie**

# Chapitre 15

## Les machines

La menuiserie possède pas mal d'équipement à la disposition des élèves. Il n'y a pas trop de restrictions sur leur utilisation à condition de tout ranger après à moins de vouloir faire face à Jacky, le responsable de la menuiserie. Il ne faut d'ailleurs pas hésiter à lui demander conseil si besoin est, il est toujours prêt à aider les étudiants dans leurs projets.

Le responsable des outils avancés de la menuiserie est Henri Proudhon. C'est celui qu'il faut contacter si jamais vous souhaitez apprendre à utiliser l'une des machines.

### 15.1 L'imprimante 3D

L'école possède pour l'instant 3 imprimantes 3D. Vous n'utiliserez très probablement jamais la plus ancienne. Les 2 que seront utiles sont :

**l'Ultimaker 2+ :** C'est celle qui peut imprimer le plus grand volume : 223 x 223 x 305 mm

**l'Ultimaker 3 :** La version la plus récente. Elle permet d'imprimer un volume plus faible que l'Ultimaker 2+ (215 x 215 x 200 mm) mais elle possède 2 têtes d'impressions ce qui lui permet de faire des impressions avec 2 matériaux différents.



FIGURE 15.1: Les imprimantes 3D Ultimaker

### 15.1.1 Le logiciel : Cura

Cura est le logiciel qui permet de créer le fichier qui sera utilisé par les Ultimakers. On peut définir les paramètres d'impressions et le positionnement des objets à imprimer dans l'imprimante 3D.

Attention, l'Ultimaker 3 ne fonctionne pas pour l'instant avec la version 3.3.1 de Cura. Il faut utiliser la version 3.2.1.

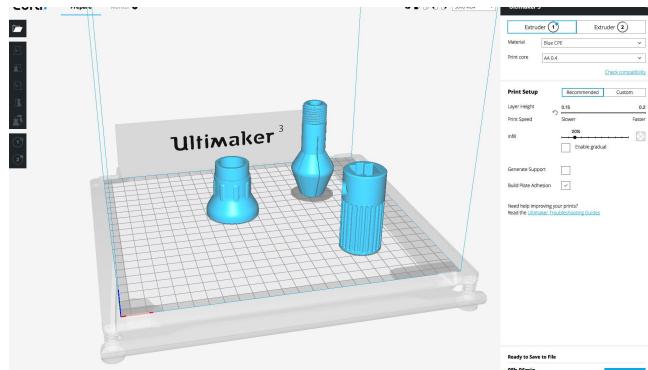


FIGURE 15.2: Cura

### 15.1.2 Utilisation

Avant de lancer l'impression, il faut passer un coup de spray dur le plateau d'impression. Cela évite d'avoir des problèmes quand on décollera l'impression à la fin.

Le menu est très intuitif et bien expliqué. Le changement des bobines est aussi très simple et bien expliqué sur le site web d'Ultimaker.

## 15.2 La découpeuse laser

La découpeuse laser est plus complexe à utiliser que l'imprimante 3D. Il vaut mieux que Henri Proudhon vous montre précisément comment l'utiliser. Il faut en effet choisir des paramètres spécifiques à chaque matériaux. Des paramètres sont déjà définis dans la machine pour certains matériaux mais il faudra très certainement mettre cela à jour avec de nouveaux matières premières. Ensuite, il faut calibrer la distance du laser au matériau à découper afin que cela corresponde à la distance focale. C'est d'ailleurs pour cela qu'il est impossible de couper des matériaux d'une trop grande épaisseur (21mm ne passent pas je crois). On peut contourner ce problème en découpant plusieurs fois le même motif et en les empilant par contre.

La découpeuse laser accepte seulement des matériaux de 40 x 80 cm (à vérifier). Cependant, la découpeuse laser s'ouvre sur un côté. Cela permet d'insérer dedans de pièces qui font plus de 40 cm de largeur.

Certaines matières plastiques ne peuvent être découpées par la découpeuse laser car ils produisent des fumées toxiques lorsqu'ils brûlent. Il faut alors utiliser une fraiseuse CNC (qui est en réparation).

### 15.2.1 Le logiciel : Inkscape

Inkscape est un logiciel de traitement graphique. C'est celui qui produit les fichiers acceptés par la découpeuse laser. Si vous souhaitez découper quelque chose, il faut que votre dessin soit importer sous Inkscape, mis à l'échelle (attention aux unités utilisés) puis il faut que le trait soit en rouge afin qu'il soit reconnu par la découpeuse laser.



FIGURE 15.3: Découpeuse laser

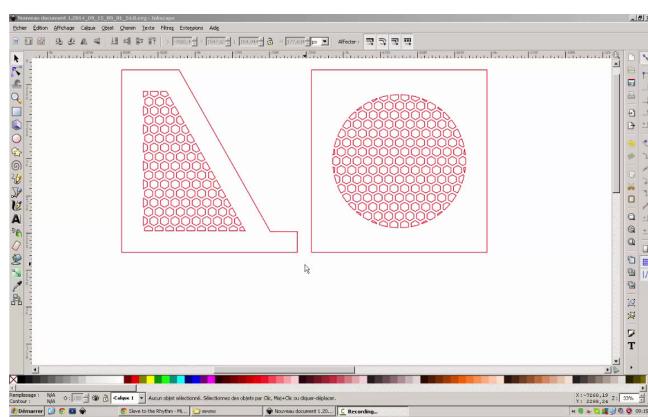


FIGURE 15.4: Inkscape

**Sixième partie**

**Conception**

## Chapitre 16

### Général

La règle d'or doit toujours être la SIMPLICITÉ. Il faut un système robuste et très répétable. Il y a toujours des imprévus (dimension de la table ou des objets à manipuler notamment).

# **Chapitre 17**

## **CAO**

## **Septième partie**

# **La Coupe de France de Robotique**

# Chapitre 18

## Pendant l'année

- Dès que du matériel est nécessaire, le commander immédiatement afin de l'avoir pour la semaine d'après. C'est souvent le matériel qui empêche de progresser sur les robots et de faire des tests.
- Garder l'un des robots de la compétition précédente afin de pouvoir avoir dès le début une base roulante. Cela permettra de commencer directement à coder et à faire des tests pour des algo de pathfinding ou d'évitement d'obstacles.

# Chapitre 19

## Le départ

- Une personne doit être responsable de l'intégralité du chargement des voitures afin d'être certain de ne rien oublier. Si plusieurs personnes s'en occupent et que la communication est mauvaise, il y aura des oubliés (oublier les vis de la table de jeu par exemple)
- Faire une liste des outils et matières premières une semaine à l'avance afin de ne rien oublier. Il faudra cette liste à Mazouz.
- Trouver un moyen pour déplacer les 2 robots et l'équipement nécessaire pour les préparer avec une seule personne. En effet, seulement 2 personnes peuvent aller jusqu'à la table de match et il peut être dur de tout porter à 2. Cela peut être un petit chariot comme le font d'autres équipes.

# Chapitre 20

## La compétition

### 20.1 Les matchs

- Faire une checklist de toutes les tâches à effectuer pour préparer le match. Avec le stress, il n'est pas rare d'oublier quelque chose et de faire rater le match.
- Éviter vraiment de faire des changements de dernière minute. En 2017, RCV a perdu en demi-finale car ils ont choisi une stratégie non-testée. Un bug a empêché les 2 robots de démarrer.

## **Huitième partie**

### **Autres**

## Chapitre 21

# Ressources utiles

Site web :

Aéromodélisme et drone : oscarliang.com