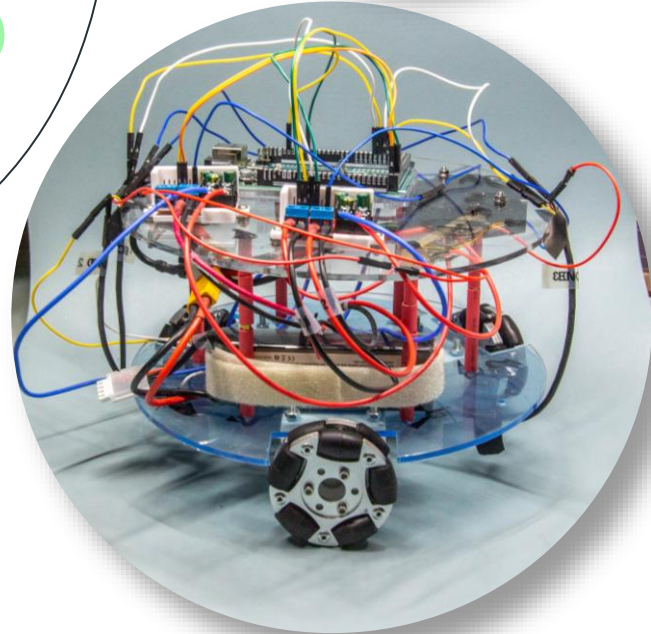
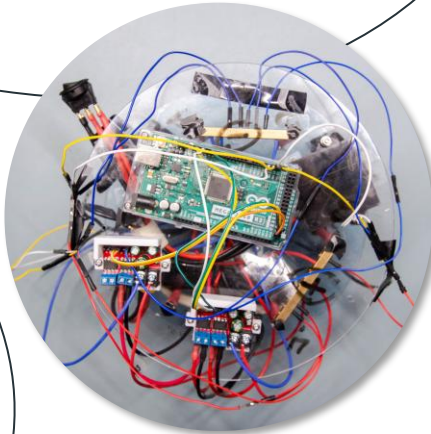
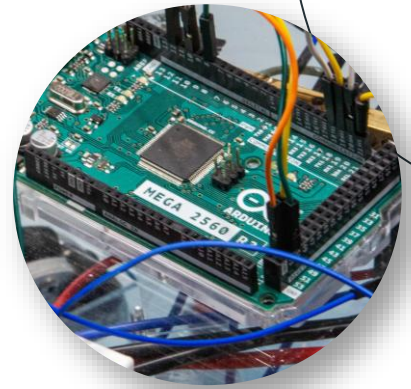


Robot Soccer

Prénom Nom

N° SCEI : *****



Sommaire

Le contexte

1

**Cinématique
d'un robot
omnidirectionnelle**

2

Expérience

3

**Description du
robot**

4

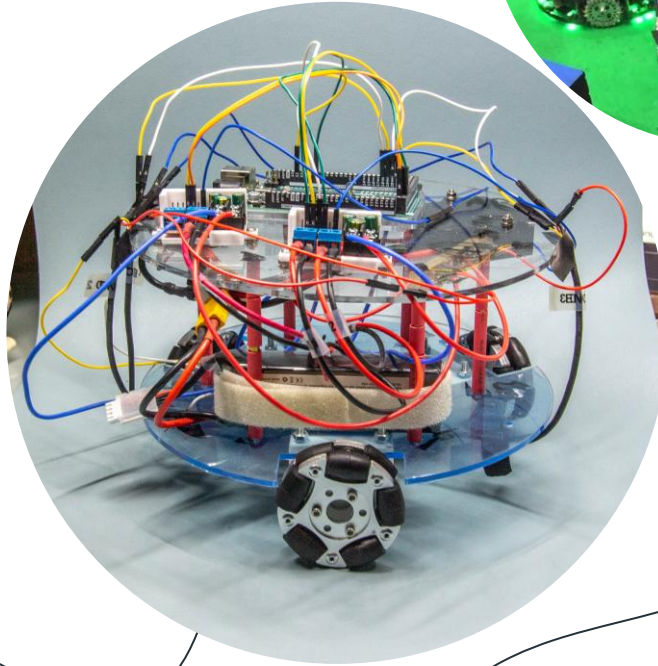
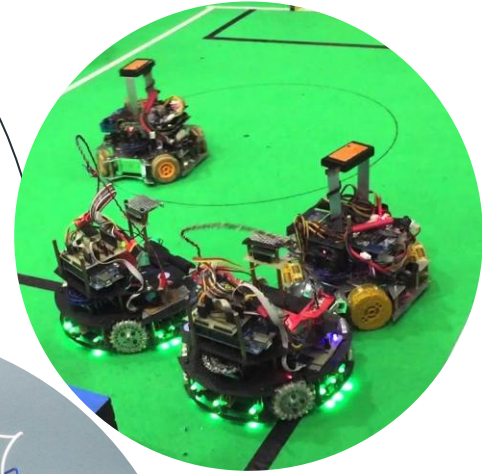
Conclusion

5



Le contexte

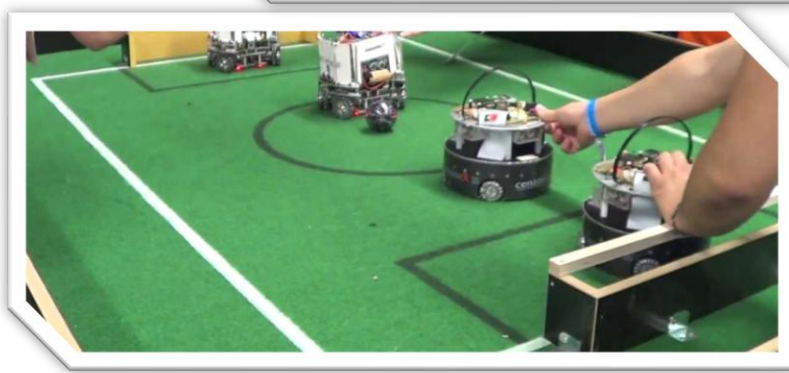
- La Robocup
- Problématique
- Objectifs



Robocup



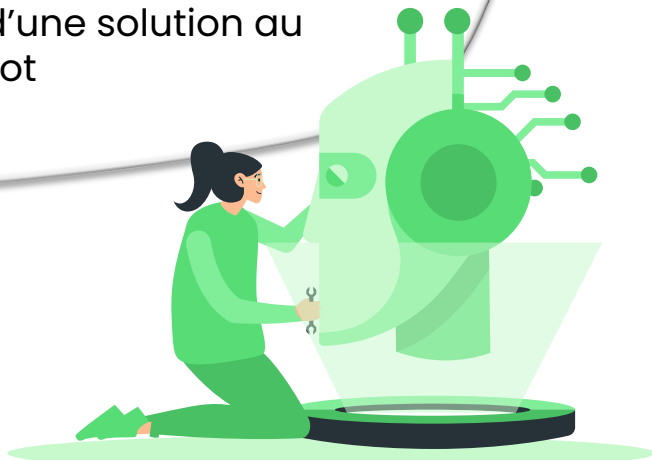
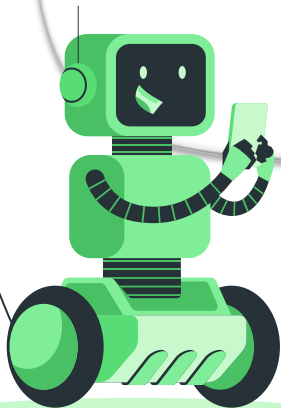
- Compétition internationale de robotique et d'intelligence artificielle
- Catalyseur en R&D
- Développement open source
- Plusieurs ligues :
 - Soccer
 - Industrielle
 - Rescue
 - Home



Problématique

Comment tirer un pénalty capable de tromper le gardien avec un robot soccer ?

Etude et réalisation d'une solution au déplacement du robot



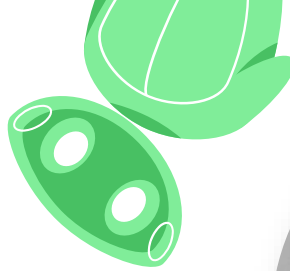
Objectifs

Objectif 1	Objectif 2
Système de déplacement : <ul style="list-style-type: none">• Etude cinématique d'un robot omnidirectionnelle• Réalisation d'un robot holonome	Système de tir : <ul style="list-style-type: none">• Etude de la cinétique d'un tir• Dimensionnement de l'électro-aimant

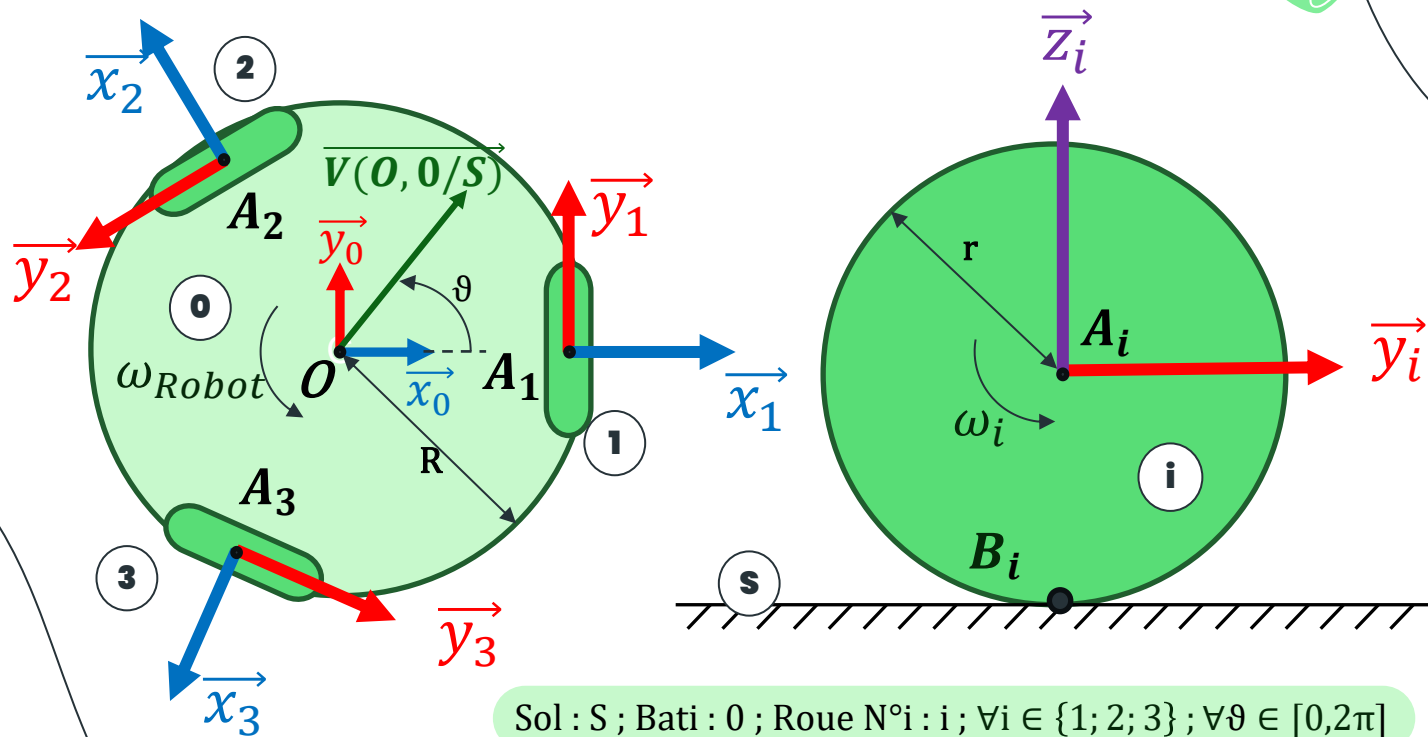
Cinématique du robot omnidirectionnel

Objectif de l'étude :

Déterminer la relation entre la
vitesse du robot et la vitesse de
rotation des moteurs



Paramétrage



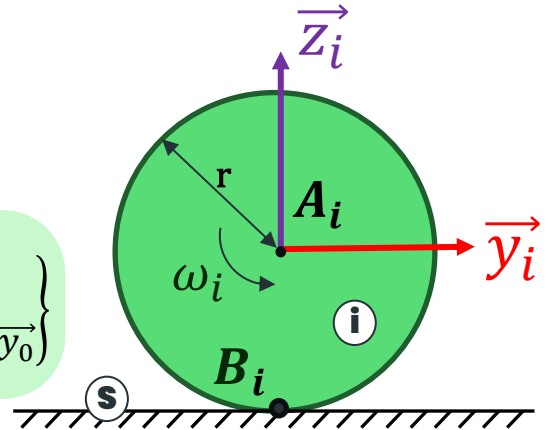
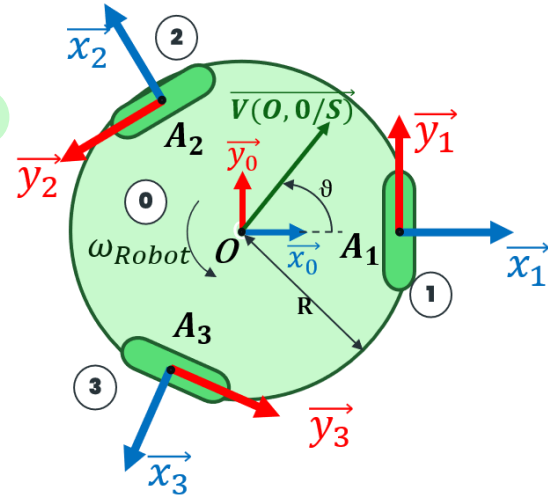
Sol : S ; Bati : 0 ; Roue N°i : i ; $\forall i \in \{1; 2; 3\}$; $\forall \vartheta \in [0, 2\pi]$

$$\alpha_1 = (\vec{y}_0, \vec{y}_1) = (\vec{x}_0, \vec{x}_1) = 0 ; \alpha_2 = (\vec{x}_0, \vec{x}_2) = \frac{2\pi}{3} ; \alpha_3 = (\vec{x}_0, \vec{x}_3) = \frac{-2\pi}{3}$$

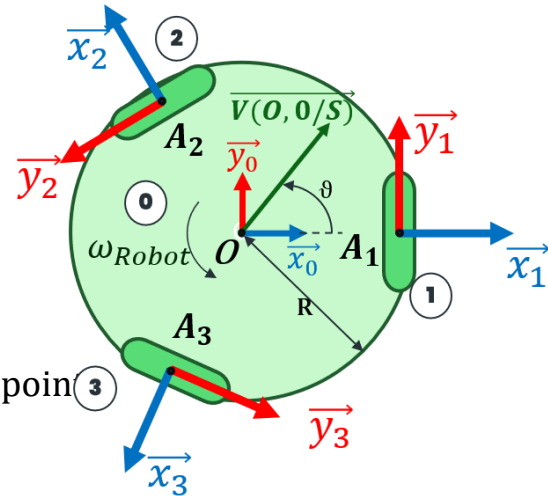
Hypothèses

- $\forall i \in \{1; 2; 3\} ; \forall \theta \in [0, 2\pi]$
- $\alpha_1 = (\vec{x}_0, \vec{x}_1) = 0$
- $\alpha_2 = (\vec{x}_0, \vec{x}_2) = \frac{2\pi}{3}$
- $\alpha_3 = (\vec{x}_0, \vec{x}_3) = \frac{-2\pi}{3}$
- $R0 = (\vec{x}_0, \vec{y}_0, \vec{z}_0)$
- $\dot{\theta} = \omega_{Robot} = \omega_{0/s}$
- Robot en translation plane et en rotation sur \vec{z}_0
- Largeur de la roue négligée, roue circulaire

$$\{v_{Robot/Sol}\} = \{v_{0/s}\} = \begin{cases} \vec{\Omega}_{0/s} = \omega_{0/s} \vec{z}_0 \\ \vec{V}_{O,0/s} = u_{0/s} \vec{x}_0 + v_{0/s} \vec{y}_0 \end{cases}$$



Calculs

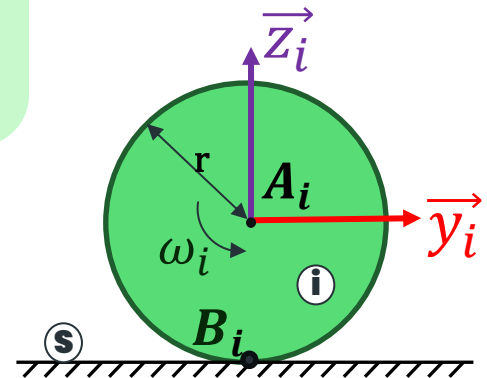


- (1) et (2) : Formule de Varignon (Changement de point)
- A_i et B_i appartiennent à la roue i

$$\overrightarrow{V(A_i, 0/S)} = \overrightarrow{V(O, 0/S)} + \overrightarrow{A_i O} \wedge \overrightarrow{\Omega_{0/S}} \quad (1)$$

$$\overrightarrow{V(A_i, 0/S)} = \overrightarrow{V(A_i, i/S)} \quad (2)$$

$$\overrightarrow{V(B_i, i/S)} = \overrightarrow{V(A_i, i/S)} + \overrightarrow{B_i A_i} \wedge \overrightarrow{\Omega_{i/S}} \quad (3)$$

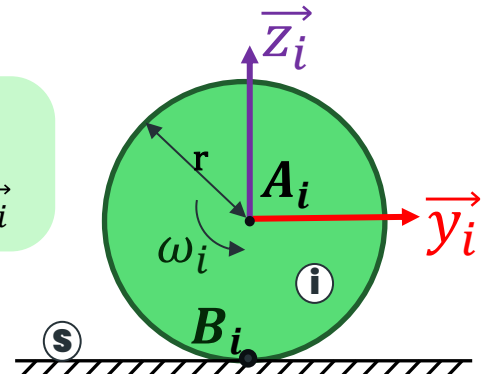
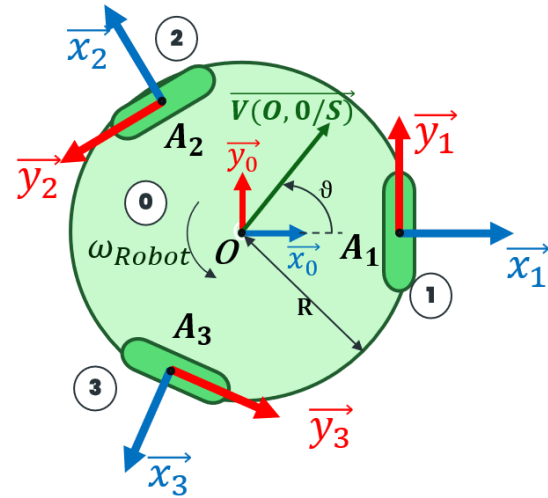


Calculs

- $$\{v_{0/S}\} = \left\{ \begin{array}{l} \overrightarrow{\Omega_{0/S}} = \omega_{0/S} \overrightarrow{z_0} \\ \overrightarrow{V_{0,0/S}} = u_{0/S} \overrightarrow{x_0} + v_{0/S} \overrightarrow{y_0} \end{array} \right\}$$
- $$\left\{ \begin{array}{l} \overrightarrow{V(A_i, 0/S)} = \overrightarrow{V(O, 0/S)} + \overrightarrow{A_i O} \wedge \overrightarrow{\Omega_{0/S}} \quad (1) \\ \overrightarrow{V(A_i, 0/S)} = \overrightarrow{V(A_i, i/S)} \quad (2) \\ \overrightarrow{V(B_i, i/S)} = \overrightarrow{V(A_i, i/S)} + \overrightarrow{B_i A_i} \wedge \overrightarrow{\Omega_{i/S}} \quad (3) \end{array} \right.$$
- $$Or \left\{ \begin{array}{l} \overrightarrow{\Omega_{0/S}} = \omega_{0/S} \overrightarrow{z_0} \\ \overrightarrow{\Omega_{i/0}} = \omega_i \overrightarrow{x_i} \\ \overrightarrow{\Omega_{i/S}} = \overrightarrow{\Omega_{i/0}} + \overrightarrow{\Omega_{0/S}} \end{array} \right.$$

Ainsi,

- $$\left\{ \begin{array}{l} (1) \Leftrightarrow \overrightarrow{V(A_i, 0/S)} = \overrightarrow{V_{0/S}} + R \omega_{0/S} \overrightarrow{z_0} \\ (3) \Leftrightarrow \overrightarrow{V(B_i, i/S)} = \overrightarrow{V(A_i, 0/S)} + r \omega_i \overrightarrow{y_i} \end{array} \right.$$

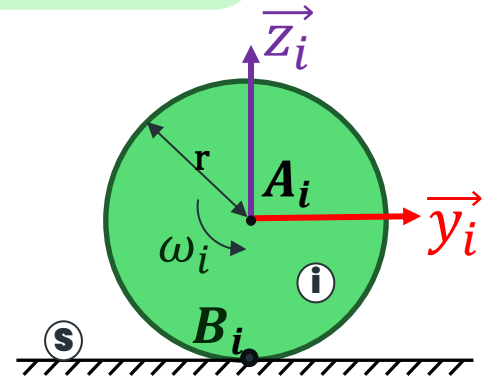
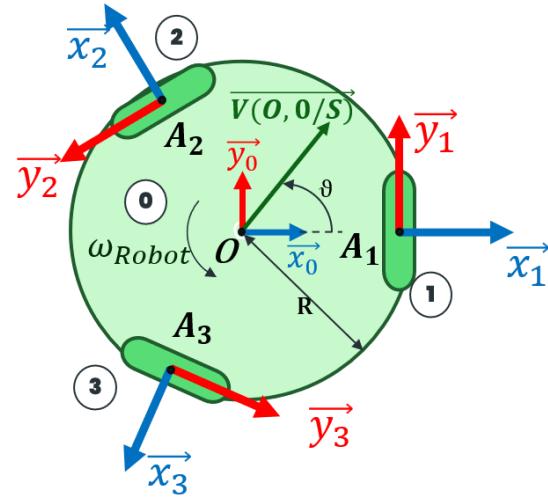


Calculs

- $\{v_{0/S}\} = \left\{ \begin{array}{l} \overrightarrow{\Omega_{0/S}} = \omega_{0/S} \overrightarrow{z_0} \\ \overrightarrow{V_{0/S}} = u_{0/S} \overrightarrow{x_0} + v_{0/S} \overrightarrow{y_0} \end{array} \right\}$
- $$\left\{ \begin{array}{l} (1) \Leftrightarrow \overrightarrow{V(A_i, 0/S)} = \overrightarrow{V_{0/S}} + R \omega_{0/S} \overrightarrow{z_0} \\ (3) \Leftrightarrow \overrightarrow{V(B_i, i/S)} = \overrightarrow{V(A_i, i/S)} + r \omega_i \overrightarrow{y_i} \end{array} \right. \quad (*)$$

Il suit que :

$$(*) \Rightarrow \overrightarrow{V(B_i, i/S)} = u_{0/S} \overrightarrow{x_0} + v_{0/S} \overrightarrow{y_0} + R \omega_{0/S} \overrightarrow{z_0} + r \omega_i \overrightarrow{y_i}$$



Calculs

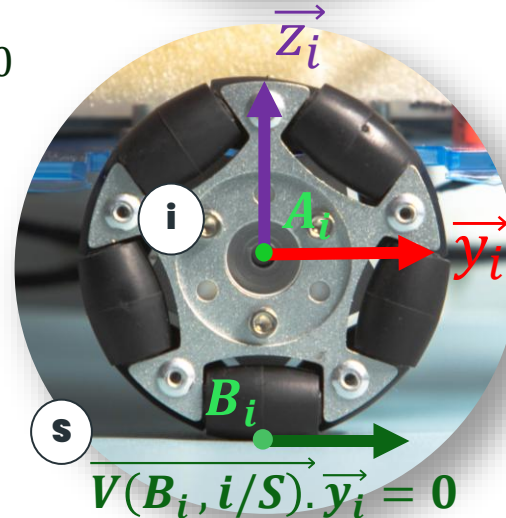
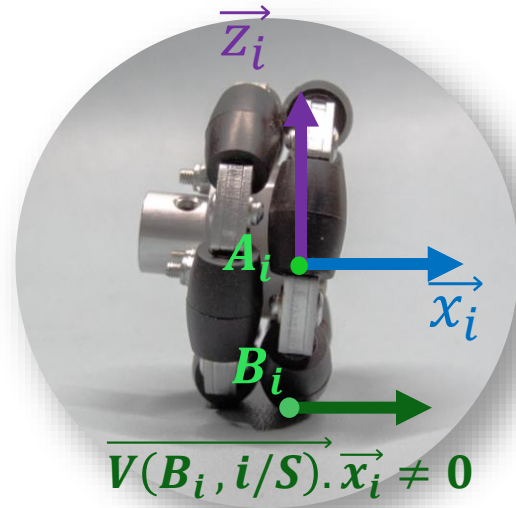
- $\overrightarrow{V(B_i, i/R0)} = u_{0/s} \overrightarrow{x_0} + v_{0/s} \overrightarrow{y_0} + R \omega_{0/s} \overrightarrow{z_0} + r \omega_i \overrightarrow{y_i}$

- Roue i en RSG selon $\overrightarrow{y_i}$:

$$\Rightarrow \overrightarrow{V(B_i, i/S)} \cdot \overrightarrow{y_i} = 0$$

$$\Leftrightarrow -u_{0/s} \sin \alpha_i + v_{0/s} \cos \alpha_i + R \omega_{0/s} + r \omega_i = 0$$

$$\Leftrightarrow \omega_i = \frac{u_{0/s} \sin \alpha_i - v_{0/s} \cos \alpha_i - R \omega_{0/s}}{r}$$

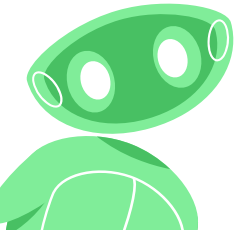
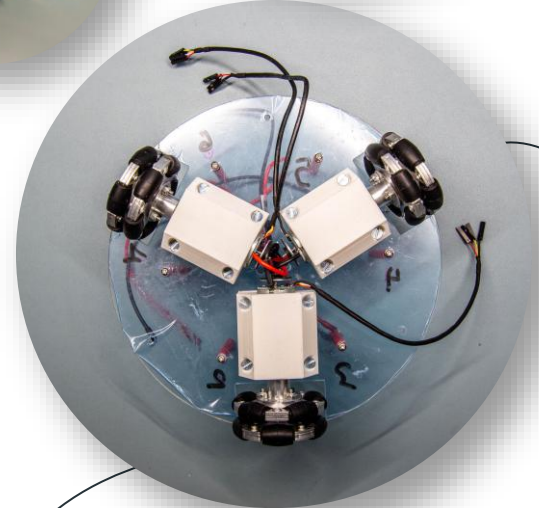
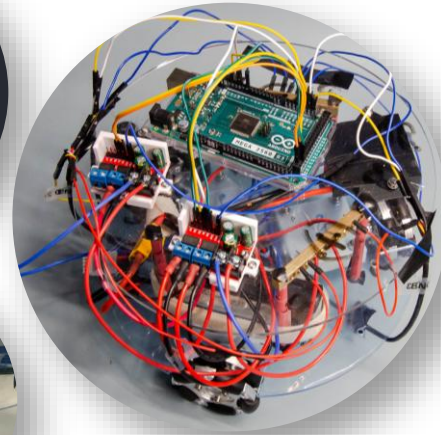
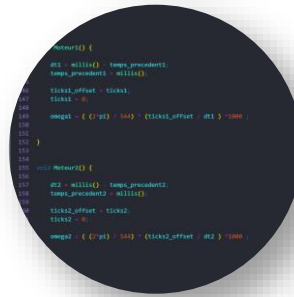


Expérience

Objectif de l'expérience :

Tester la validité de la relation
établie par l'étude théorique

$$\omega_i = \frac{u_{0/s} \sin \alpha_i - v_{0/s} \cos \alpha_i - R \omega_{0/s}}{r}$$



Protocole et hypothèse

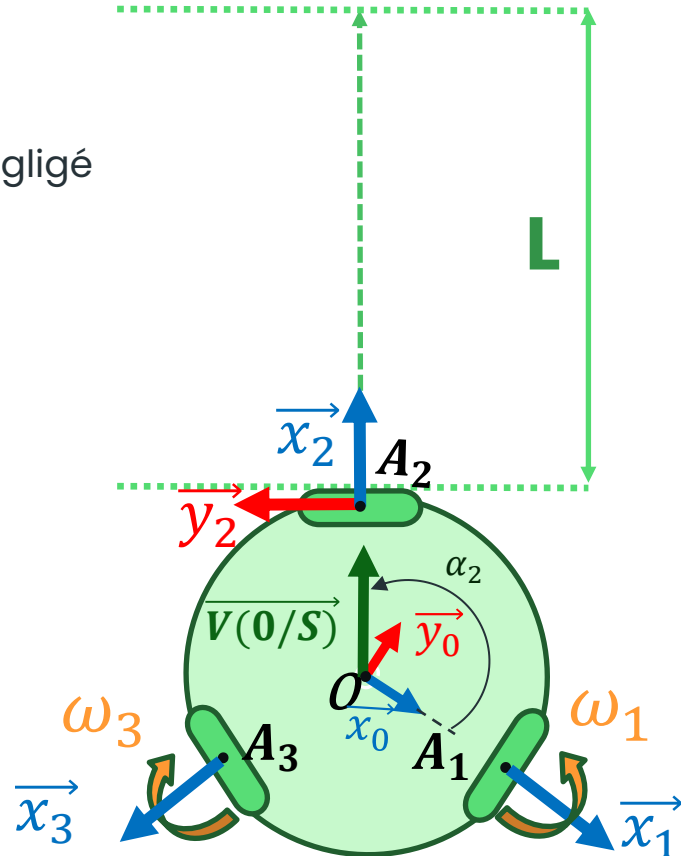
Hypothèses :

- Roulement sans glissement
- Baisse de tension de la batterie négligé

$$\{v_{Robot/Sol}\} = \left\{ \begin{array}{l} \overrightarrow{\Omega_{0/S}} = \vec{0} \\ \overrightarrow{V_{O,0/S}} = v_{Robot} \overrightarrow{x_2} \end{array} \right\}$$

Protocole :

- Faire tradater le robot selon $\overrightarrow{x_2}$ durant T secondes
- Mesurer L : la distance parcourut
- Mesurer ω_1 et ω_3 les vitesses de rotation des roues 1 et 3 par tracking vidéo



Résultats théorique

$$\bullet \quad \{v_{Robot/Sol}\} = \left\{ \begin{array}{l} \overrightarrow{\Omega_{0/S}} = \vec{0} \\ \overrightarrow{V_{O,0/S}} = v_{Robot} \vec{x}_2 \end{array} \right\} \quad (1)$$

$$\bullet \quad \text{Relation : } \omega_i = \frac{u_{0/s} \sin \alpha_i - v_{0/s} \cos \alpha_i - R \omega_{0/s}}{r} \quad (2)$$

$$\bullet \quad u_{0/s} = \frac{-1}{2} v_{Robot}$$

$$\bullet \quad v_{0/s} = \frac{\sqrt{3}}{2} v_{Robot}$$

$$\bullet \quad \omega_{0/s} = 0$$

Finalement,

$$\omega_1 = \frac{-\sqrt{3} v_{Robot}}{2 \times 29} ;$$

$$\omega_2 = 0 ;$$

$$\omega_3 = \frac{\sqrt{3} v_{Robot}}{2 \times 29}$$

Nota bene :

$$\bullet \quad \alpha_1 = 0$$

$$\bullet \quad \alpha_2 = \frac{2\pi}{3}$$

$$\bullet \quad \alpha_3 = \frac{-2\pi}{3}$$

$$\bullet \quad \text{Rayon roue : } r = 29\text{mm}$$

$$\bullet \quad \omega = |\omega_1| = |\omega_3|$$

$$\bullet \quad v_{Robot} = \frac{L}{T}$$

Résultats pratique

Expérience 1

- Revêtement du sol : parquet stratifié
- $T_1 = 5$ secondes
- $L_1 = 1480\text{mm}$

Expérience 2

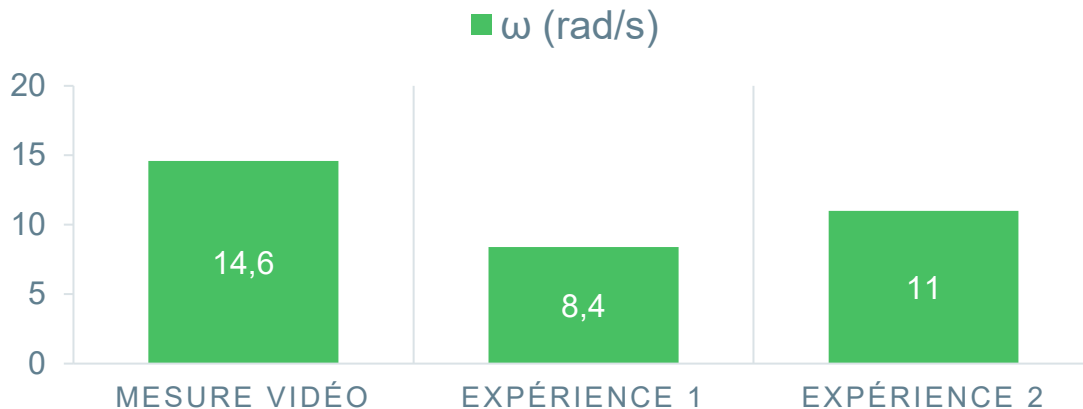
- Revêtement du sol : carrelage
- $T_2 = 3$ secondes
- $L_2 = 1100\text{mm}$

Calcul des vitesses de déplacement du robot

$$v_{E1} = \frac{L_1}{T_1} = 296 \text{ mm/s}$$

$$v_{E2} = \frac{L_2}{T_2} = 367 \text{ mm/s}$$

Résultats pratique



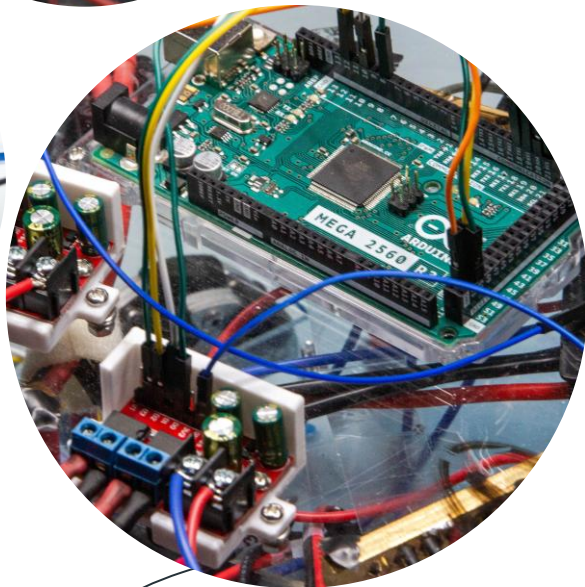
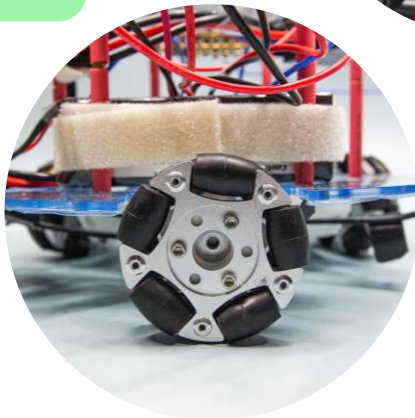
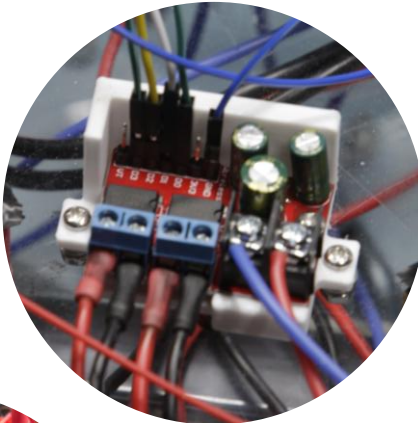
Expérience 1	Expérience 2
Calcul des vitesses de déplacement du robot	
$v_{E1} = \frac{L_1}{T_1} = \frac{1480}{5} = 296 \text{ mm/s}$	$v_{E2} = \frac{L_2}{T_2} = \frac{1100}{3} = 367 \text{ mm/s}$
Calcul des vitesses de rotation des roues	
$\omega_{E1} = \frac{\sqrt{3} v_{E1}}{2 \times 29} = 8,4 \text{ rad/s}$	$\omega_3 = \frac{\sqrt{3} v_{E2}}{2 \times 29} = 11 \text{ rad/s}$
Vitesse de rotation à vide mesurée par tracking vidéo :	
14,6 rad/s	

Conclusion expérience

- D'importantes incertitudes
 - Mesure des distances $\approx 1\text{cm}$
 - Vitesse de rotation mesurée à vide
 - Forte influence du revêtement : augmentation de 35% de vitesse sur le carrelage par rapport au stratifié
 - Problèmes électroniques
- Taux d'erreur :
 - Meilleur cas : 21,9% (expérience 2)
 - Pire cas : 42,4% (expérience 1)
- Conclusion et Axe d'amélioration : Taux d'erreurs importants à cause du glissement et de la mesure de vitesse de rotation mais modèle théorique validé. Par ailleurs, des améliorations sont à prévoir sur le prototype.

Description du Robot

- Les besoins
- Exigence de déplacement
- Exigences du règlement
- BDD
- Système de tir



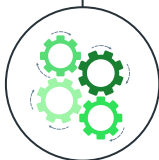
Les besoins

Robot Soccer



Rapidité

Complicquer la
tache du gardien



Agilité

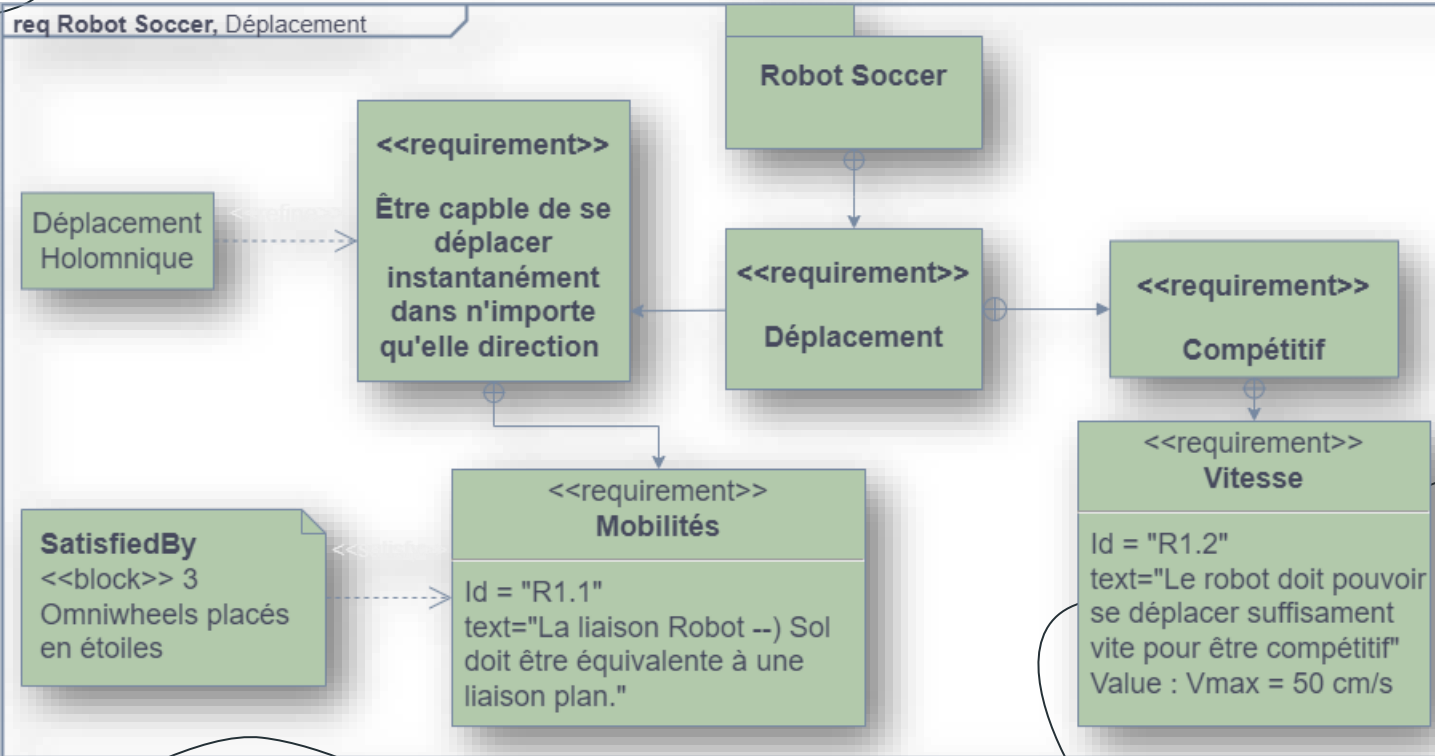
Se déplacer
instantanément dans
n'importe quelle
direction



Respect du Règlement

Avoir un robot crédible
pour participer à la
compétition

Exigences de déplacement



Exigences du règlement

req Robot Soccer, Respect des Normes

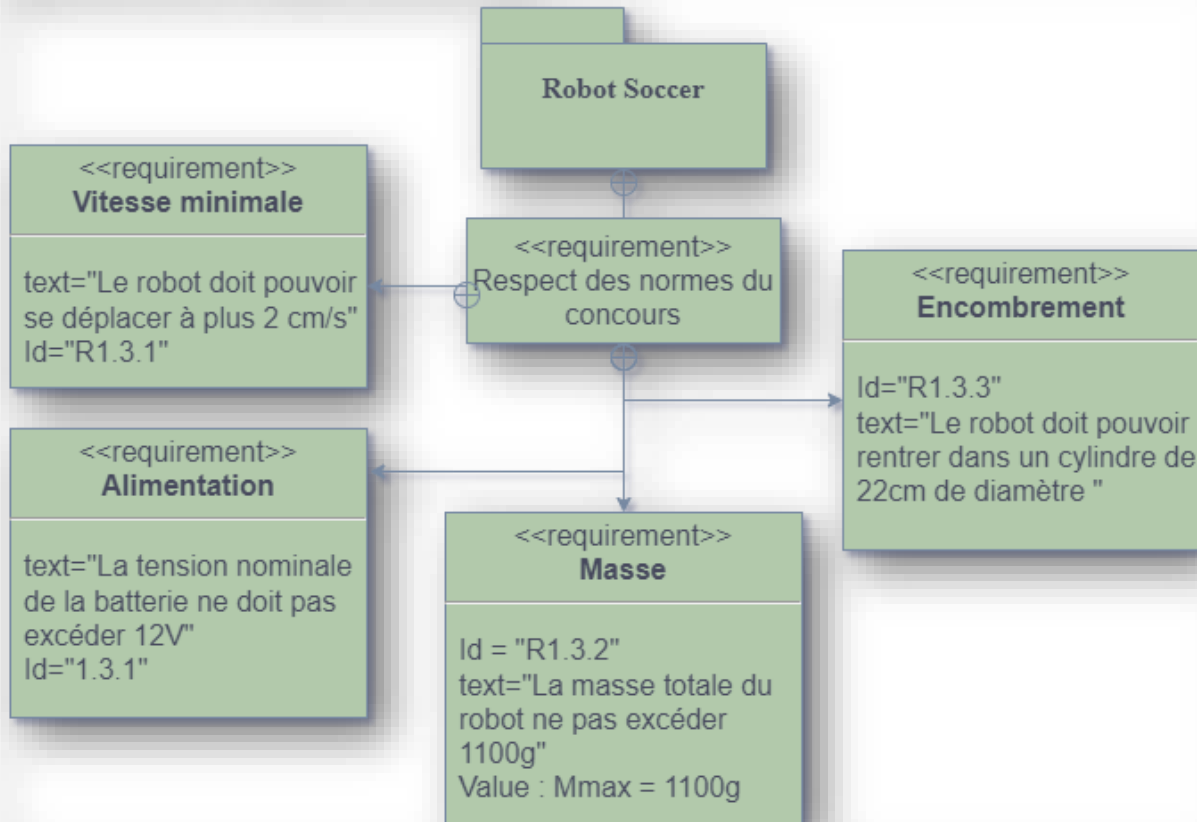
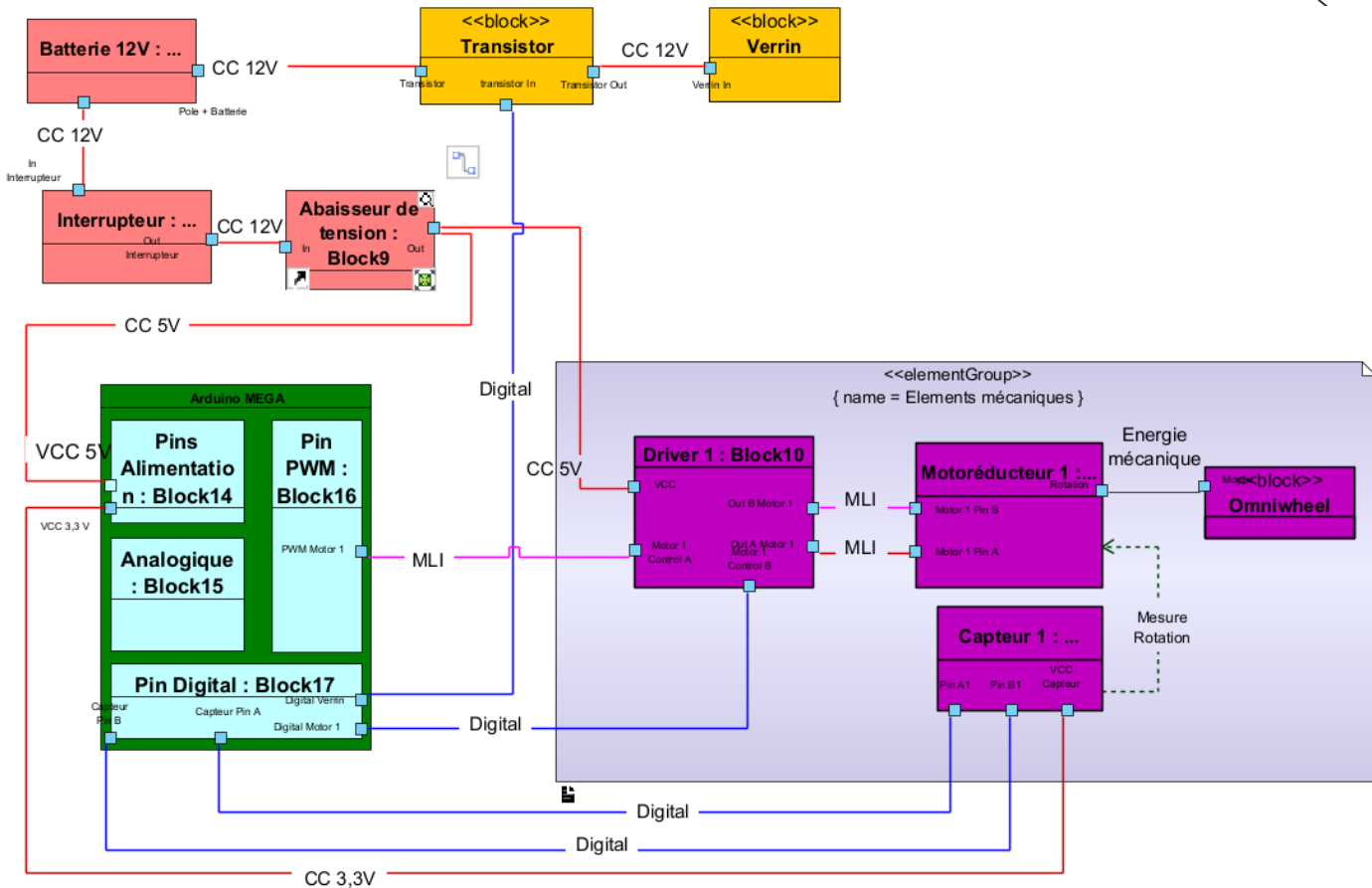
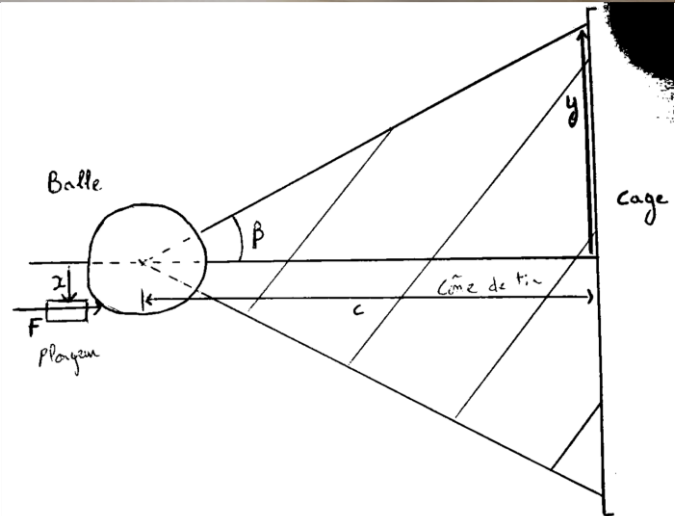
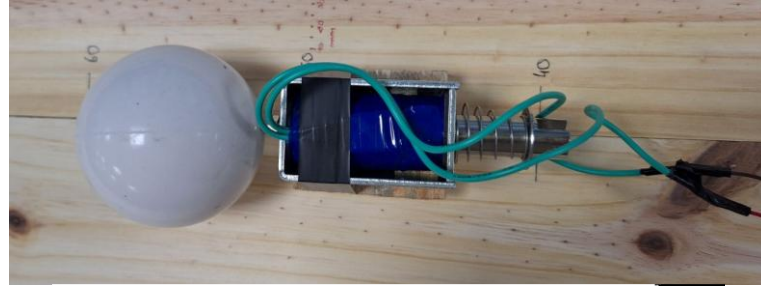


Diagramme de définition de bloc



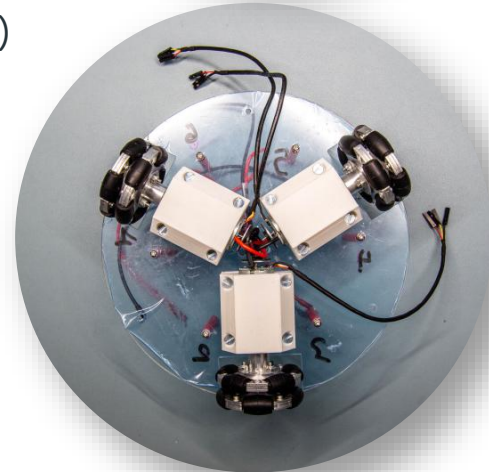
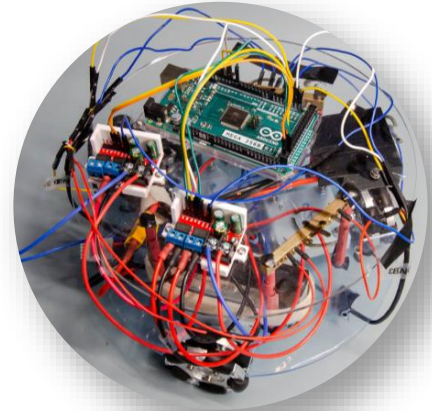
Système de tir

- Electro-aimant à solénoïde
- Certitude d'un tir cadré à une distance de 40cm
- Relation entre la position de la frappe sur la balle et la position de la balle dans les cages

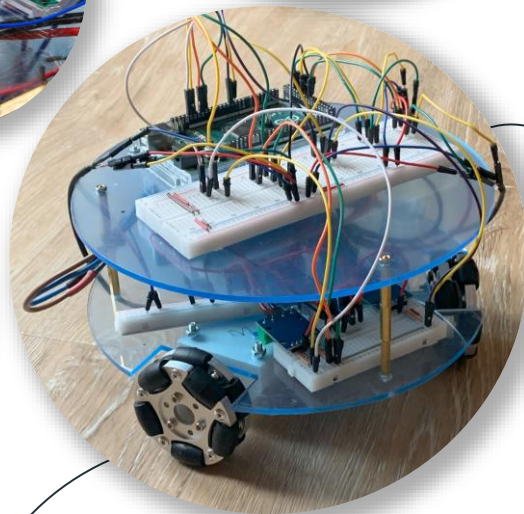
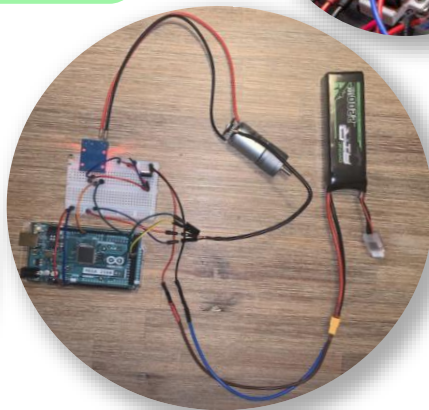
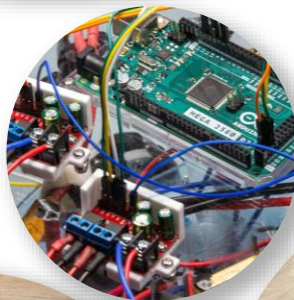
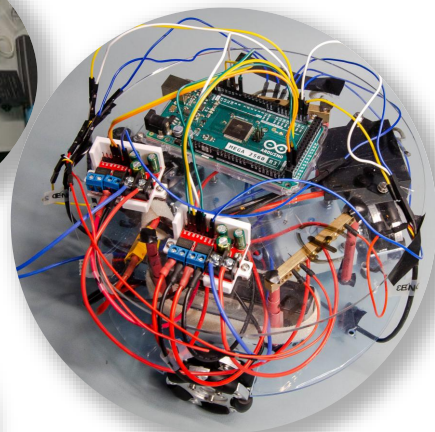
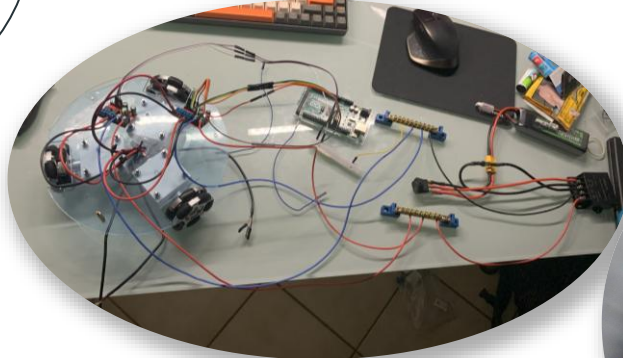


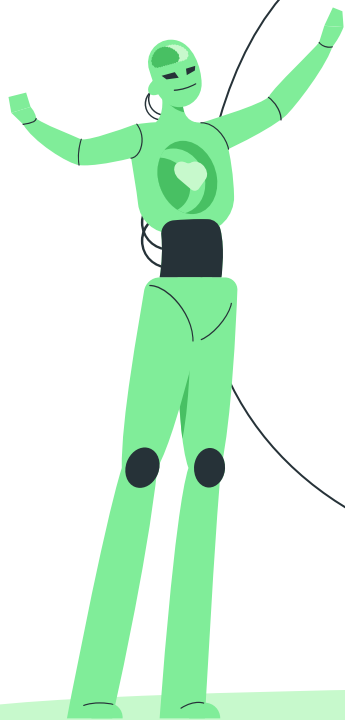
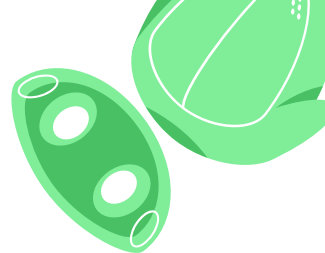
Solution problématique

- Robot à roue omnidirectionnelles
 - ✓ ○ Liaison Robot/Sol \leftrightarrow Liaison plan (CDC)
 - ✓ ○ Loi de commande des moteurs
- Vitesse max
 - ✓ ○ théorique : $770 \text{ mm/s} > 500 \text{ mm/s}$ (CDC)
 - ⊘ ○ Expérimentale : $367 \text{ mm/s} < 500 \text{ mm/s}$ (CDC)
- Système de tir fonctionnel
- Améliorations nécessaires :
 - Montage électrique
 - Raquette de tir

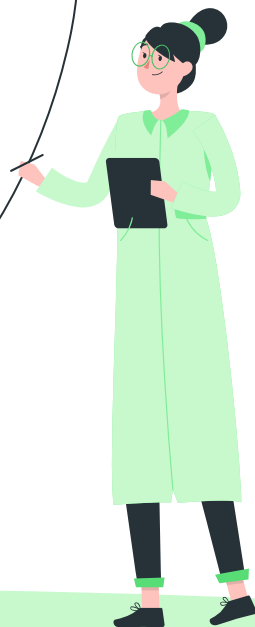


**Merci
pour votre
attention**





Annexes



Les problèmes

Chaine de puissance

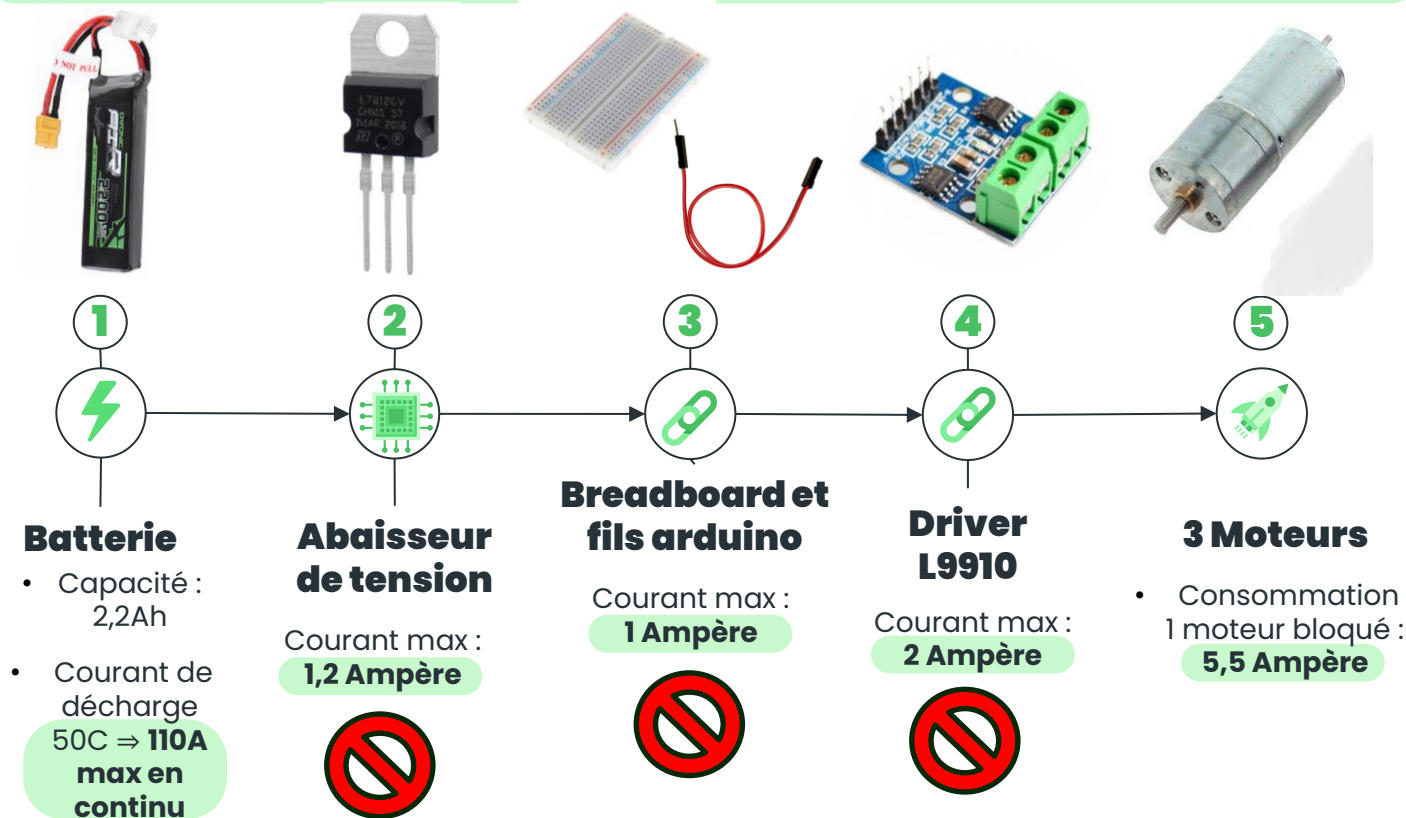
- Tout les moteurs tournent bien à vide
- Difficile de solliciter plusieurs moteurs en même temps
- Ne tournent plus une fois chargé

Moteur

- Importants efforts internes anormaux
- Anomalie absente des autres moteurs
- S'est déclaré après assemblage



Problème : Chaine de puissance



Solution : Chaîne de puissance

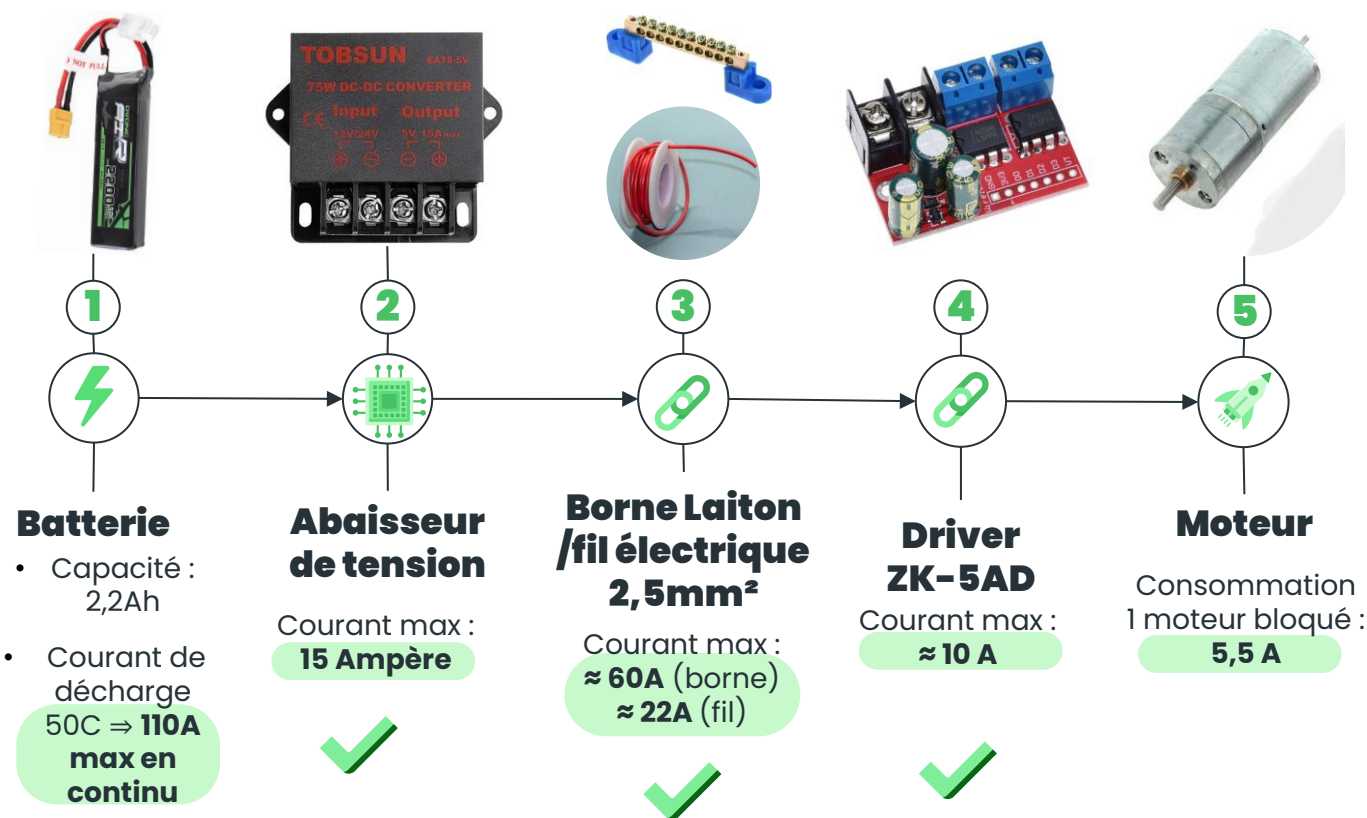
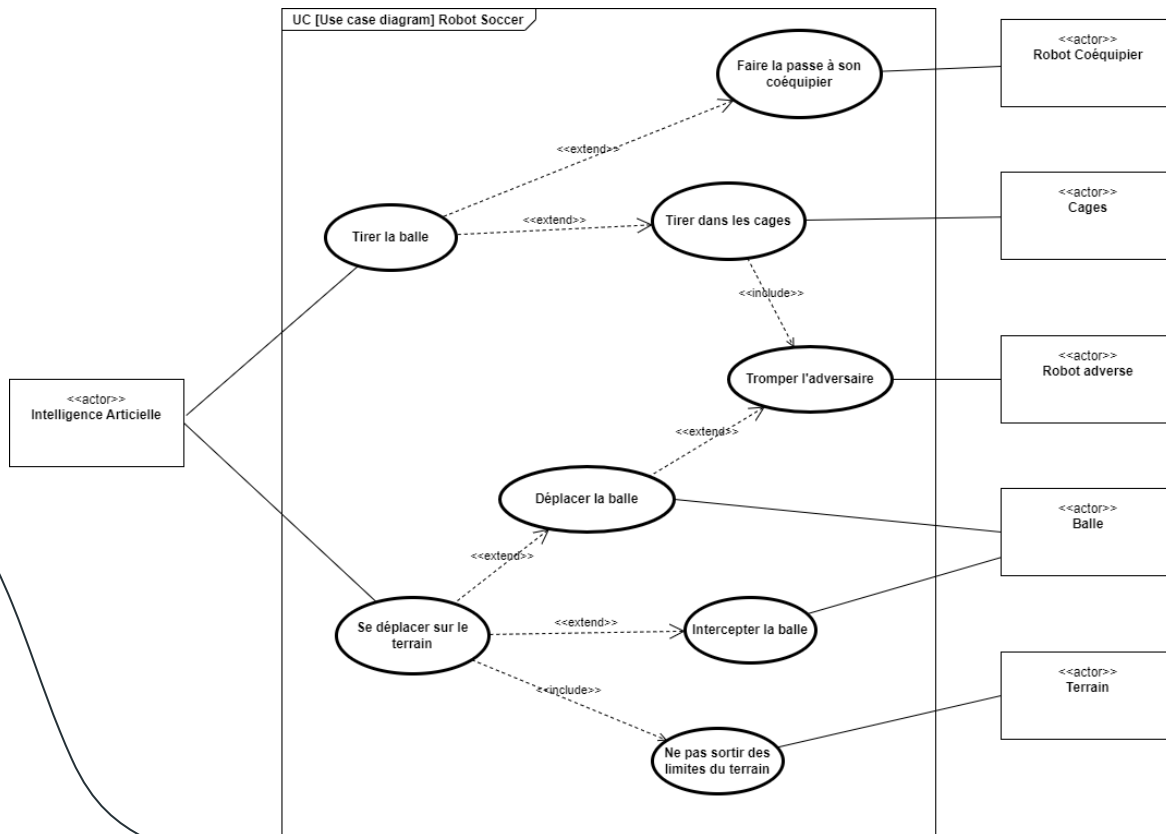


Diagramme des cas d'utilisation



Interrupteur

**Driver 1
(Moteur 3)**

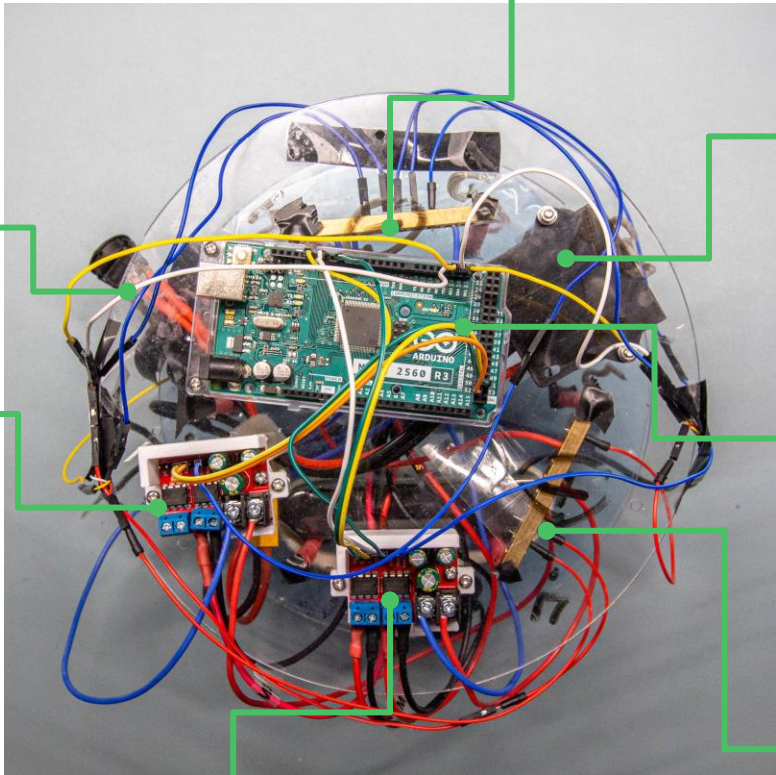
Driver 1 (Moteur 1 et Moteur 2)

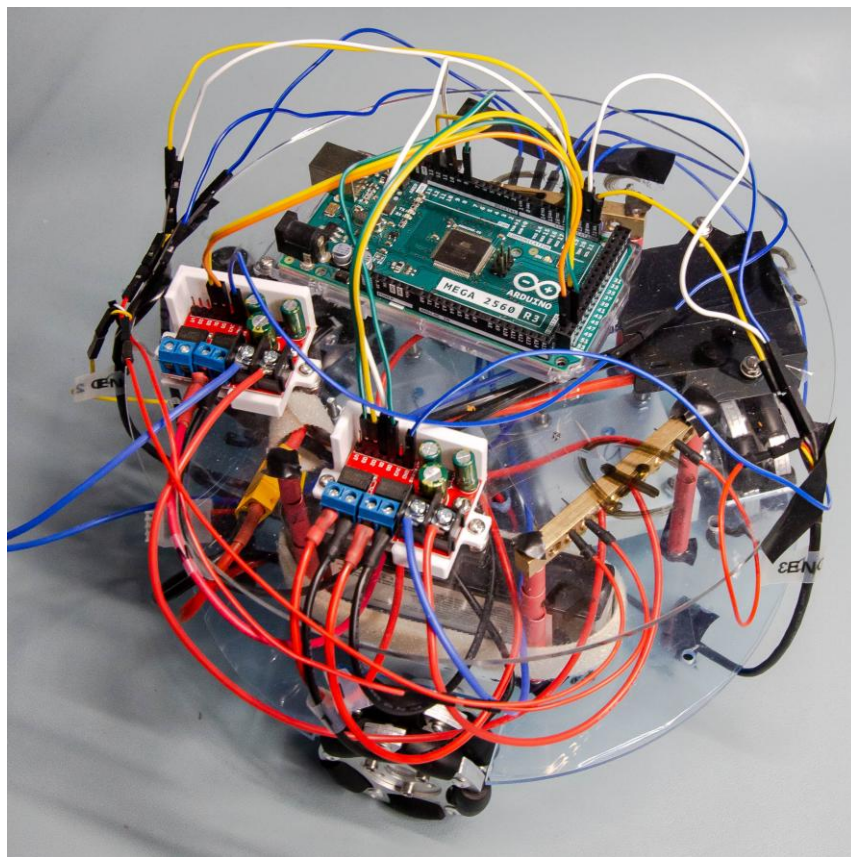
Borne -

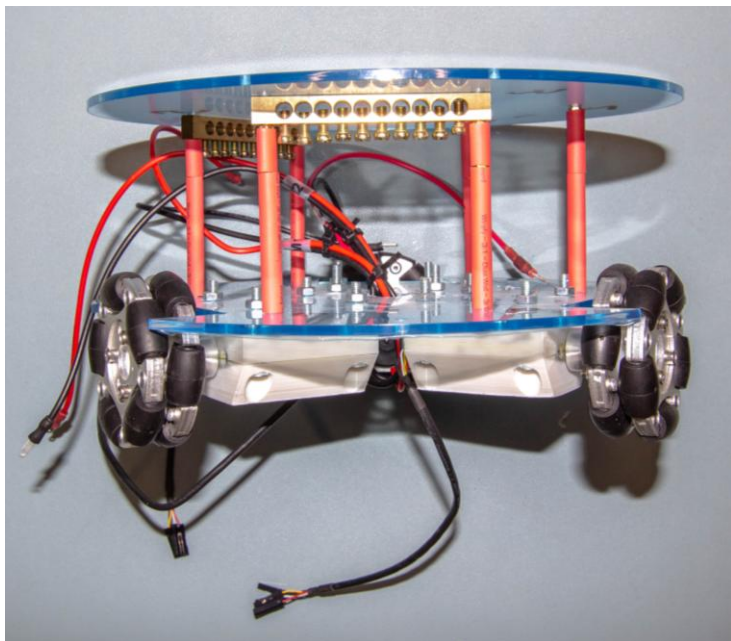
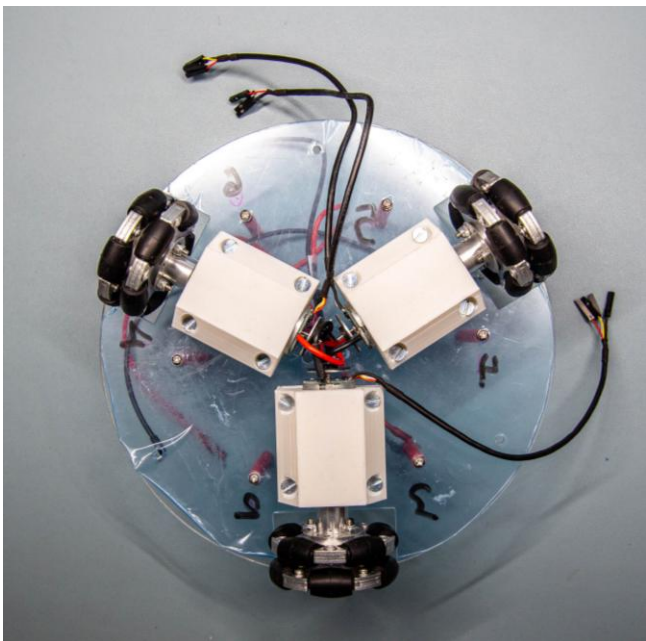
**Abaisseur
de tension**

**Arduino
Mega**

Borne +



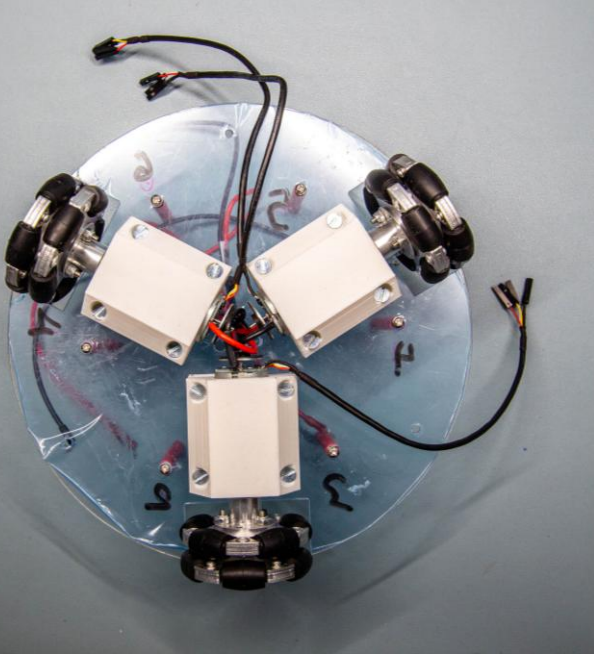




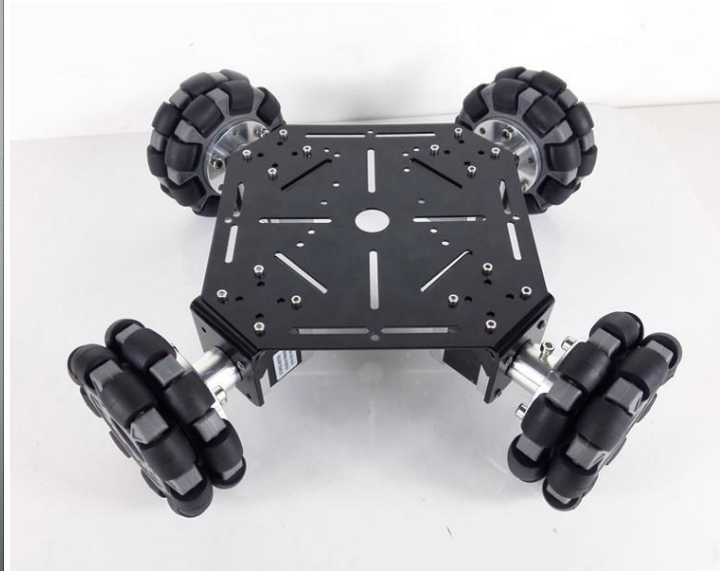
Robot « omni »

Avantages	Inconvénients
<ul style="list-style-type: none">• Liberté de déplacement• Maniabilité	<ul style="list-style-type: none">• Vibration• Manque d'adhérence• Besoin d'un couple plus important qu'un système "classique"• Système plus complexe

Robot « omni »



Configuration 3WD



Configuration 4WD

Robot « omni »

3 roues motrices

- **+ Stable :**
 - 3 points \Rightarrow 1 plan
- **- Code plus complexe :**
 - Cinématique moins intuitive
- **Mécanisme + simple :**
 - Chaines de puissance et d'information plus légères
 - Plus compact

4 roues motrices

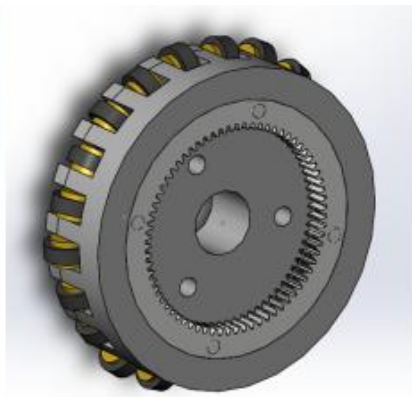
- **+ d'adhérence**
- **+ de couple**
- **Code + simple**
- **- Mécanisme plus complexe :**
 - Problème de planéité
 - 4 moteurs à contrôler
 - Encombrement plus important

Choix : Robot 3 roues motrices

Critères retenus :

- Compacité
- Stabilité
- Simplicité de conception

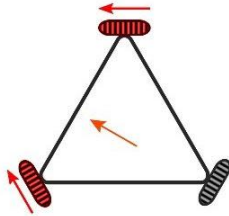
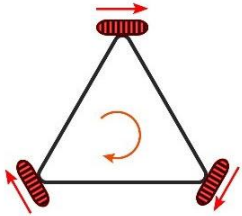
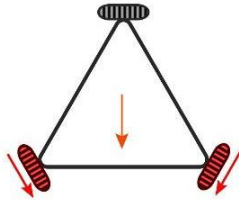
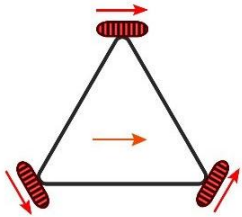
Omniwheel



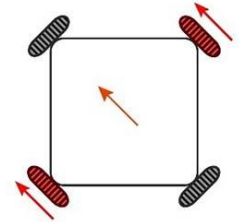
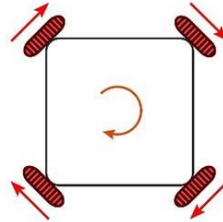
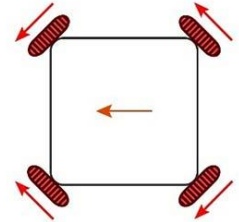
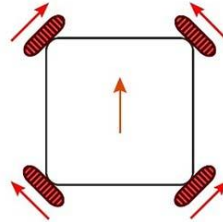
Roue « mécanum »



Déplacement holonome

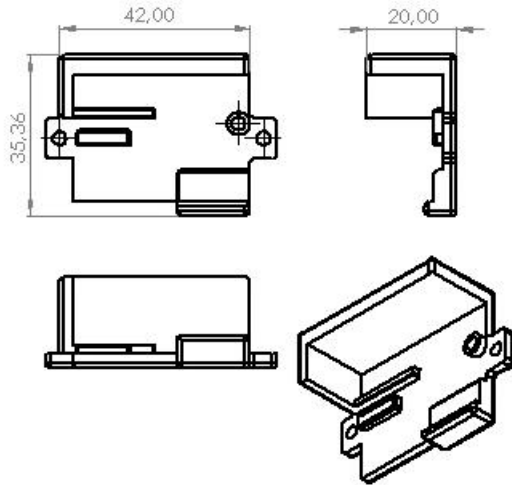


Configuration 3WD

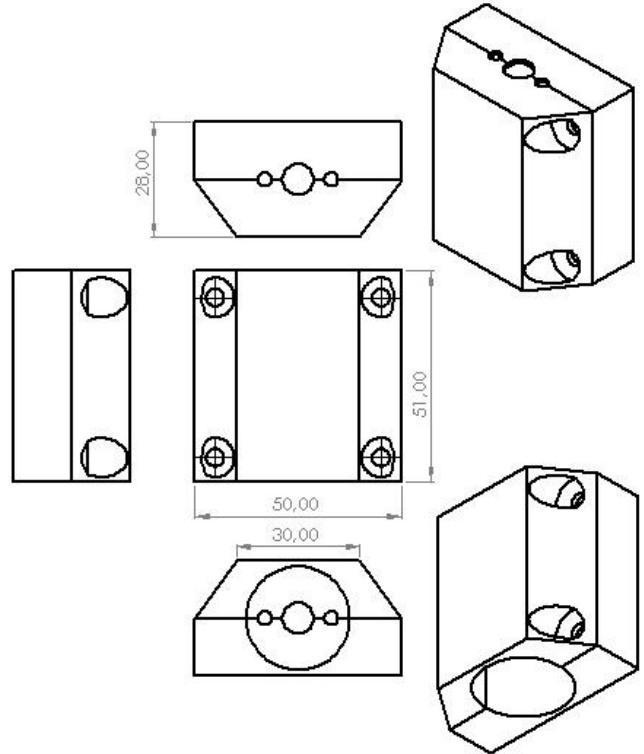


Configuration 4WD

Dessin support driver et moteur

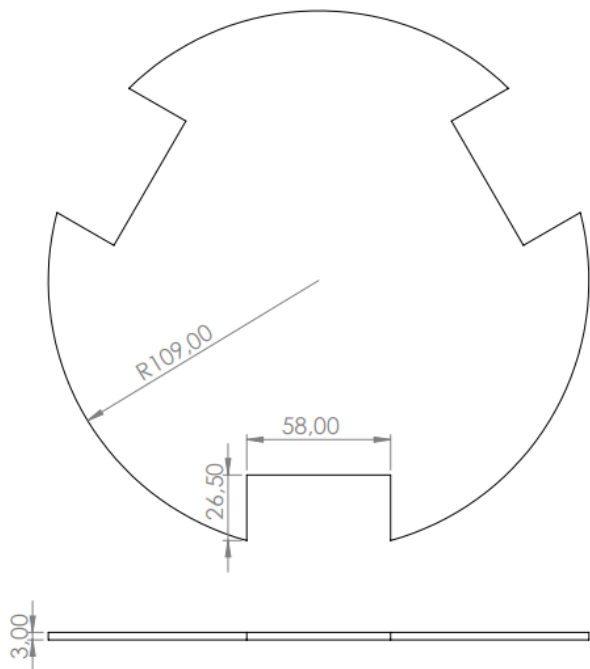


Support driver

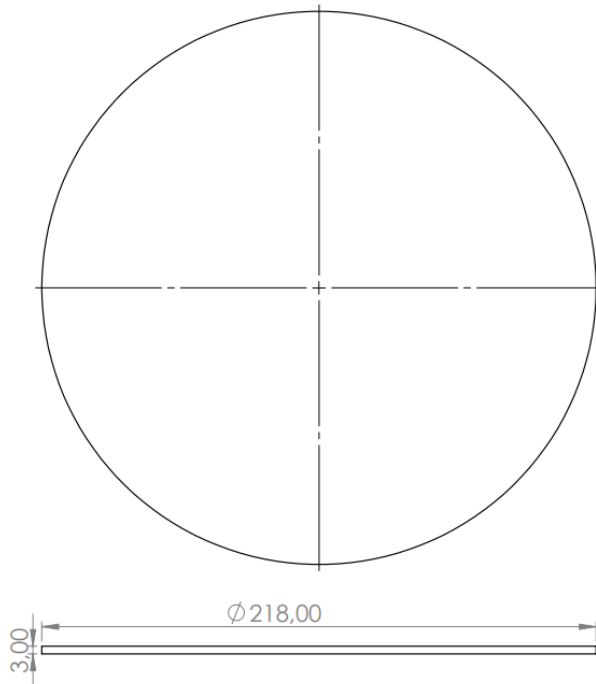


Support moteur

Dessin support driver et moteur



Chassie moteur



**Chassie
supérieur**

Calcul expérience :

$$\bullet \quad \{v_{Robot/Sol}\} = \left\{ \begin{array}{l} \overrightarrow{\Omega_{0/S}} = \vec{0} \\ \overrightarrow{V_{0,0/S}} = v_{Robot} \vec{x}_2 \end{array} \right\} \quad (1)$$

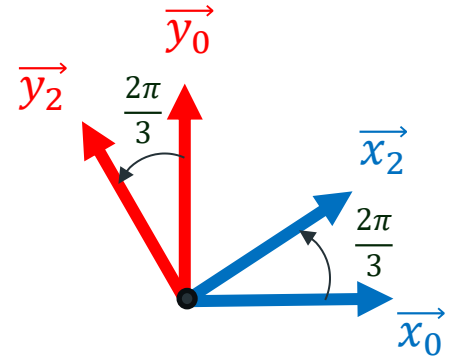
$$\begin{aligned} (1) \Rightarrow \overrightarrow{V_{0,0/S}} &= v_{Robot} (\cos \alpha_2 \vec{x}_0 + \sin \alpha_2 \vec{y}_0) \\ &= v_{Robot} (\cos \frac{2\pi}{3} \vec{x}_0 + \sin \frac{2\pi}{3} \vec{y}_0) \\ &= \underbrace{\frac{-1}{2} v_{Robot}}_{u_{0/s}} \vec{x}_0 + \underbrace{\frac{\sqrt{3}}{2} v_{Robot}}_{v_{0/s}} \vec{y}_0 \end{aligned}$$

Alors :

$$u_{0/s} = \frac{-1}{2} v_{Robot} \quad ; \quad v_{0/s} = \frac{\sqrt{3}}{2} v_{Robot} \quad ; \quad \omega_{0/s} = 0$$

Nota bene :

$$\bullet \quad \alpha_2 = \frac{2\pi}{3}$$



Calcul expérience : ω_1

- Relation : $\omega_i = \frac{u_{0/s} \sin \alpha_i - v_{0/s} \cos \alpha_i - R \omega_{0/s}}{r}$ (2)

- $u_{0/s} = \frac{-1}{2} v_{Robot}$ (3)

- $v_{0/s} = \frac{\sqrt{3}}{2} v_{Robot}$ (4)

Nota bene :

- $\alpha_1 = 0$
- Rayon roue : $r = 29\text{mm}$

Alors,

$$\begin{aligned}\omega_1 &= \frac{1}{29} \left(\frac{-1}{2} v_{Robot} \sin 0 - \frac{\sqrt{3}}{2} v_{Robot} \cos 0 \right) \\ &= \frac{-1}{29} \times \frac{\sqrt{3}}{2} v_{Robot}\end{aligned}$$

Calcul expérience : ω_2

- Relation : $\omega_i = \frac{u_{0/s} \sin \alpha_i - v_{0/s} \cos \alpha_i - R \omega_{0/s}}{r}$ (2)

- $u_{0/s} = \frac{-1}{2} v_{Robot}$ (3)

- $v_{0/s} = \frac{\sqrt{3}}{2} v_{Robot}$ (4)

Nota bene :

- $\alpha_2 = \frac{2\pi}{3}$

- Rayon roue : $r = 29\text{mm}$

Alors,

$$\omega_2 = \frac{1}{29} \left(\frac{-1}{2} v_{Robot} \sin \frac{2\pi}{3} - \frac{\sqrt{3}}{2} v_{Robot} \cos \frac{2\pi}{3} \right)$$

$$= \frac{1}{29} \left(\frac{-1}{2} \times \frac{\sqrt{3}}{2} v_{Robot} - \frac{\sqrt{3}}{2} \times \frac{-1}{2} v_{Robot} \right)$$

$$= 0$$

Calcul expérience : ω_3

- Relation : $\omega_i = \frac{u_{0/s} \sin \alpha_i - v_{0/s} \cos \alpha_i - R \omega_{0/s}}{r}$ (2)

- $u_{0/s} = \frac{-1}{2} v_{Robot}$ (3)

- $v_{0/s} = \frac{\sqrt{3}}{2} v_{Robot}$ (4)

Nota bene :

- $\alpha_3 = \frac{-2\pi}{3}$

- Rayon roue : $r = 29\text{mm}$

Alors,

$$\omega_3 = \frac{1}{29} \left(\frac{-1}{2} v_{Robot} \sin \frac{-2\pi}{3} - \frac{\sqrt{3}}{2} v_{Robot} \cos \frac{-2\pi}{3} \right)$$

$$= \frac{1}{29} \left(\frac{1}{2} \times \frac{\sqrt{3}}{2} v_{Robot} + \frac{\sqrt{3}}{2} \times \frac{1}{2} v_{Robot} \right)$$

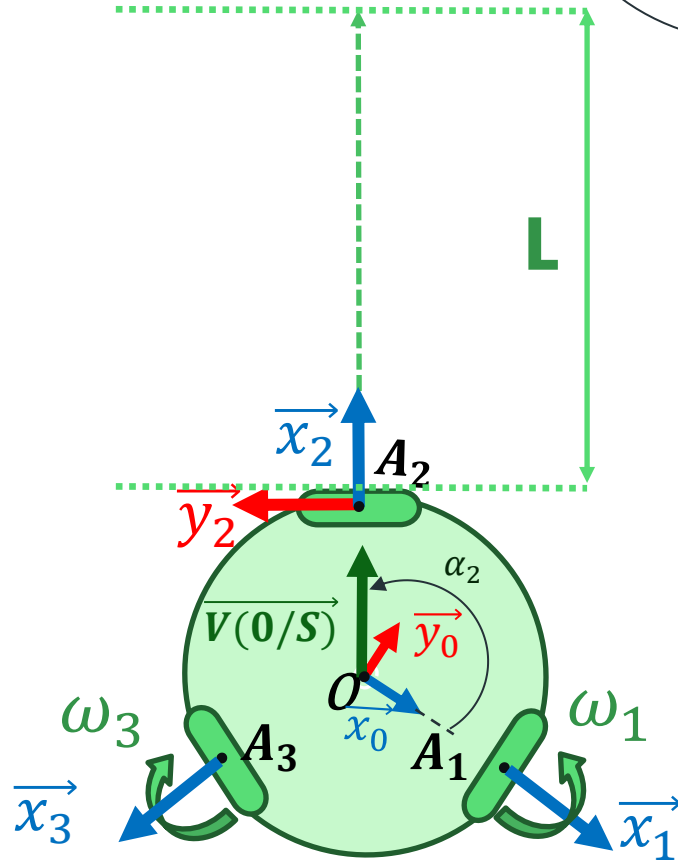
$$= \frac{1}{29} \times \frac{\sqrt{3}}{2} v_{Robot} = -\omega_1$$

Calcul résultat pratique

$$\omega = \frac{\sin \frac{2\pi}{3}}{r} v_{Robot} = \frac{\sqrt{3}}{2 \times 29} v_{Robot}$$

$$\begin{cases} V_{E1} = \frac{L_1}{T_1} = \frac{1480}{5} = 296 \text{ mm/s} \\ V_{E2} = \frac{L_2}{T_2} = \frac{1100}{3} = 367 \text{ mm/s} \end{cases}$$

$$\begin{cases} \omega_{E1} = \frac{\sqrt{3}}{2 \times 29} \times 296 = 8,4 \text{ rad/s} \\ \omega_{E2} = \frac{\sqrt{3}}{2 \times 29} \times 367 = 11 \text{ rad/s} \end{cases}$$



Expérience validation vidéo



Expérience validation vidéo

Moteur 1

Moteur 3

Consigne

250 bits

110 bits

$T_{1 \text{ tour}}$

0,43s

0,43s

ω

14,62 rad/s

14,62 rad/s

- Faire tourner les moteurs à vide aux même consigne que durant l'expérience
- Mesure de temps pour 1 tour
- $$\omega = \frac{2\pi}{T_{1 \text{ tour}}} = \frac{2\pi}{0,43} = 14,61 \text{ rad/s}$$

Expérience encodeur

1 moteur	2 moteurs
<ul style="list-style-type: none">• $T1 = 0,27s$• Vitesse mesuré : 23 rad/s	<ul style="list-style-type: none">• $T1 = 0,29s$• Vitesse mesuré : 11,55 rad/s
Calcul des vitesses de rotations	
$\omega_{1mot} = \frac{2\pi}{0,27} = 23,7 \text{ rad/s} \quad v_{2mot} = \frac{2\pi}{0,29} = 21,66 \text{ rad/s}$	
Conclusion	
Mesure de l'encodeur validé	Mesure de l'encodeur invalidé

Table de vérité Driver Bit de seuil moteurs

	Moteur 1		Moteur 3	
	Sens 1	PWM 1	Sens 3	PWM 3
Rotation Horaire Vmax	LOW	255 bit	LOW	255 bit
Rotation Horaire Vmax	LOW	38 bit	LOW	60 bit
Rotation anti-Horaire Vmax	HIGH	0 bit	HIGH	0 bit
Rotation anti-Horaire Vmin	HIGH	245 bit	HIGH	236 bit

```


1  #include <digitalWriteFast.h>
2
3  //SETUP DES CONSTANTES / VARIABLES
4
5  //Encodeurs : N° de pin Arduino / N° de pin d'interruption
6  const int Encodeur1A = 18;
7  const int Encodeur1B = 19;
8
9  const int Encodeur2A = 18;
10 const int Encodeur2B = 19;
11
12 const int Encodeur3A = 20;
13 const int Encodeur3B = 21;
14
15
16 volatile long ticks1 = 0; volatile long ticks1_offset = 0;
17 volatile long ticks2 = 0; volatile long ticks2_offset = 0;
18 volatile long ticks3 = 0; volatile long ticks3_offset = 0;
19
20 //Moteurs :
21 const int Mot_1_PWM = 8 ;
22 const int Mot_1_Sens = 50 ;
23
24
25 const int Mot_2_PWM = 10 ;
26 const int Mot_2_Sens = 51 ;
27
28 const int Mot_3_PWM = 12 ;
29 const int Mot_3_Sens = 52 ;
30
31
32 volatile double omega1;
33 volatile double omega2;
34 volatile double omega3;
35
36
37 // Variables de temps
38 volatile long temps_initiale = millis(); //(en milliseconde)
39
40 volatile long temps_precedent1; volatile long dt1;
41 volatile long temps_precedent2; volatile long dt2;
42 volatile long temps_precedent3; volatile long dt3;
43
44
45 //Globales
46 volatile float pi = 3.141592653589793;
47

```

```

1  void setup() {
2
3      Serial.begin(9600);
4      Serial.flush();
5
6      //Encodeurs
7      pinMode(Encodeur1A, INPUT);
8      pinMode(Encodeur1B, INPUT);
9
10     //pinMode(Encodeur2A, INPUT);
11     //pinMode(Encodeur2B, INPUT);
12
13     pinMode(Encodeur3A, INPUT);
14     pinMode(Encodeur3B, INPUT);
15
16
17     // activation des résistances de pullup
18     digitalWrite(Encodeur1A, HIGH);
19     digitalWrite(Encodeur1B, HIGH);
20
21     //digitalWrite(Encodeur2A, HIGH);
22     //digitalWrite(Encodeur2B, HIGH);
23
24     digitalWrite(Encodeur3A, HIGH);
25     digitalWrite(Encodeur3B, HIGH);
26
27
28     //Initialiation des interruptions
29     attachInterrupt(digitalPinToInterrupt(Encodeur1A), Codeur_Interupt_1A, CHANGE);
30     attachInterrupt(digitalPinToInterrupt(Encodeur1B), Codeur_Interupt_1B, CHANGE);
31
32
33     //Moteurs
34     pinMode(Mot_1_Sens, OUTPUT);
35     pinMode(Mot_1_PWM, OUTPUT);
36
37     //pinMode(Mot_2_Sens, OUTPUT);
38     //pinMode(Mot_2_PWM, OUTPUT);
39
40     pinMode(Mot_3_Sens, OUTPUT);
41     pinMode(Mot_3_PWM, OUTPUT);
42
43
44     digitalWrite(Mot_1_Sens, LOW);
45     analogWrite(Mot_1_PWM, 250 );
46
47     //digitalWrite(Mot_2_Sens,LOW);
48     //analogWrite(Mot_2_PWM, 0 );
49
50     digitalWrite(Mot_3_Sens, LOW);
51     analogWrite(Mot_3_PWM, 110 );
52
53 }
54
55

```



```
1
2 void loop() {
3
4     Moteur1();
5     //Moteur2();
6     Moteur3();
7
8     if ( (millis() - temps_initiale) > 3000) {
9
10        analogWrite(Mot_1_PWM, 0 );
11        digitalWrite(Mot_1_Sens, LOW);
12
13        analogWrite(Mot_3_PWM, 0 );
14        digitalWrite(Mot_3_Sens, LOW);
15
16    }
17 }
18
```



```
1 void Moteur1() {  
2  
3     dt1 = millis() - temps_precedent1;  
4     temps_precedent1 = millis();  
5  
6     ticks1_offset = ticks1;  
7     ticks1 = 0;  
8  
9     omega1 = ( (2*pi) / 544) * (ticks1_offset / dt1 ) *1000 ;  
10  
11  
12 }
```



```
1  //FONCTION ENCODEURS
2
3  void Codeur_Interrupt_1A() {
4
5      if (digitalReadFast(Encodeur1A) == digitalReadFast(Encodeur1B)) {
6          ticks1--;
7      }
8
9      else {
10         ticks1++;
11     }
12 }
13
14
15
16
17 void Codeur_Interrupt_1B() {
18
19     if (digitalReadFast(Encodeur1A) == digitalReadFast(Encodeur1B)) {
20         ticks1++;
21     }
22
23     else {
24         ticks1--;
25     }
26 }
27
28
29
```



```
1 void print_data() {
2
3     Serial.println("-----");
4     Serial.print("Temps : "); Serial.println( (millis() - temps_initiale) );
5     Serial.println("-----");
6     Serial.println("          | Vitesse");
7     Serial.println("-----");
8     Serial.print("Moteur 1 | "); Serial.print( omega1 ); Serial.println(" rad/seconde");
9     Serial.println("-----");
10    Serial.print("Moteur 3 | "); Serial.print( omega3 ); Serial.println(" rad/seconde");
11    Serial.println("-----");
12    Serial.println(" ");
13
14 }
```



```
1
2 //FONCTIONS MOTEURS
3
4 void Moteur1() {
5
6     dt1 = millis() - temps_precedent1;
7     temps_precedent1 = millis();
8
9     ticks1_offset = ticks1;
10    ticks1 = 0;
11
12    omega1 = ( (2*pi) / 544) * (ticks1_offset / dt1 ) *1000 ;
13
14
15 }
16
17
18 void Moteur2() {
19
20     dt2 = millis() - temps_precedent2;
21     temps_precedent2 = millis();
22
23     ticks2_offset = ticks2;
24     ticks2 = 0;
25
26     omega2 = ( (2*pi) / 544) * (ticks2_offset / dt2 ) *1000 ;
27
28
29
30 }
31
32 void Moteur3() {
33
34     dt3 = millis() - temps_precedent3;
35     temps_precedent3 = millis();
36
37     ticks3_offset = ticks3;
38     ticks3 = 0;
39
40     omega3 = ( (2*pi) / 544) * (ticks3_offset / dt3 ) *1000 ;
41
42
43
44 }
45
```

$$\text{Ainsi, } \{v_{0/s}\} = \begin{Bmatrix} 0 & u_{0/s} \\ 0 & v_{0/s} \\ \omega_{0/s} & 0 \end{Bmatrix}_{R0=(\vec{x}_0, \vec{y}_0, \vec{z}_0)} = \begin{Bmatrix} \overrightarrow{\Omega_{0/s}} = \omega_{0/s} \vec{z}_0 \\ \overrightarrow{V_{0/s}} = u_{0/s} \vec{x}_0 + v_{0/s} \vec{y}_0 \end{Bmatrix}_{R0=(\vec{x}_0, \vec{y}_0, \vec{z}_0)}$$

$$\begin{cases} \overrightarrow{V(A_i, 0/S)} = \overrightarrow{V(O, 0/S)} + \overrightarrow{A_i O} \wedge \overrightarrow{\Omega_{0/s}} \\ \overrightarrow{V(B_i, i/S)} = \overrightarrow{V(A_i, i/R0)} + \overrightarrow{B_i A_i} \wedge \overrightarrow{\Omega_{i/s}} \end{cases} \quad (1)$$

$$(3)$$

$$\bullet \text{ Or } \begin{cases} \overrightarrow{\Omega_{i/0}} = \omega_{0/s} \vec{z}_0 \\ \overrightarrow{\Omega_{i/0}} = \omega_i \vec{x}_i \\ \overrightarrow{\Omega_{i/s}} = \overrightarrow{\Omega_{i/0}} + \overrightarrow{\Omega_{0/s}} \end{cases}$$

$$\text{Relation : } \omega_i = \frac{u_{0/s} \sin \alpha_i - v_{0/s} \cos \alpha_i - R \omega_{0/s}}{r}$$

$$V_1 = \frac{148}{5} \times 10 = 294 \text{ mm/s}$$

$$V_2 = \frac{110}{3} \times 10 = 367 \text{ mm/s}$$

$$\bullet \quad u_{0/s} = \frac{-1}{2} v_{Robot}$$

$$\bullet \quad v_{0/s} = \frac{\sqrt{3}}{2} v_{Robot}$$

$$\bullet \quad \begin{cases} (1) \Leftrightarrow \overrightarrow{V\left(A_i, \frac{0}{S}\right)} = \overrightarrow{\vec{V}_0} + R \omega_{0/s} \vec{z}_0 \\ (3) \Leftrightarrow \overrightarrow{V\left(B_i, \frac{i}{R0}\right)} = \overrightarrow{V\left(A_i, \frac{0}{S}\right)} + r \omega_i \vec{y}_i \end{cases}$$



Table de vérité/seuil Driver

Moteurs 1	Sens	PWM
Rotation Horaire Vmax	LOW	255
Rotation Horaire Vmin	LOW	38
Rotation anti-Horaire Vmax	HIGH	0
Rotation anti-Horaire Vmin	HIGH	245