



Lanceur de ballons de football intelligent

PRÉPARÉ PAR :

BELHAJ Hamza : OJ077T

ENCADRÉ PAR :

Pr A. OUAANABI

Plan de la présentation

- **EXPRESSION DU BESOIN**
- **PROBLÉMATIQUE**
- **CAHIER DES CHARGES FONCTIONNEL**
- **OBJECTIFS :**
 - **OBJECTIF 1 : Principe de fonctionnement du système**
 - **OBJECTIF 2 : Étude électrique du système**
 - **OBJECTIF 3 : Commande à distance du robot**
- **CONCLUSION**

Expression du besoin



Problématique

Comment peut-on assurer d'une manière efficace le déplacement du robot sur le terrain, l'orientation précise des balles, tout en assurant la sécurité des personnes ? Et de quelle manière peut-on établir une communication à distance avec le système ?

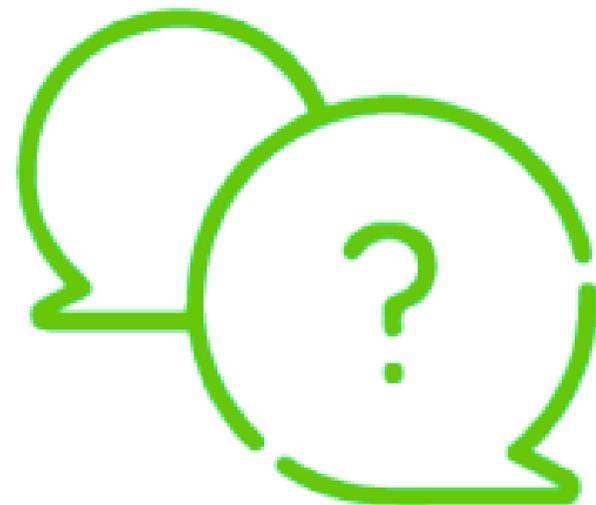


Diagramme des cas d'utilisation

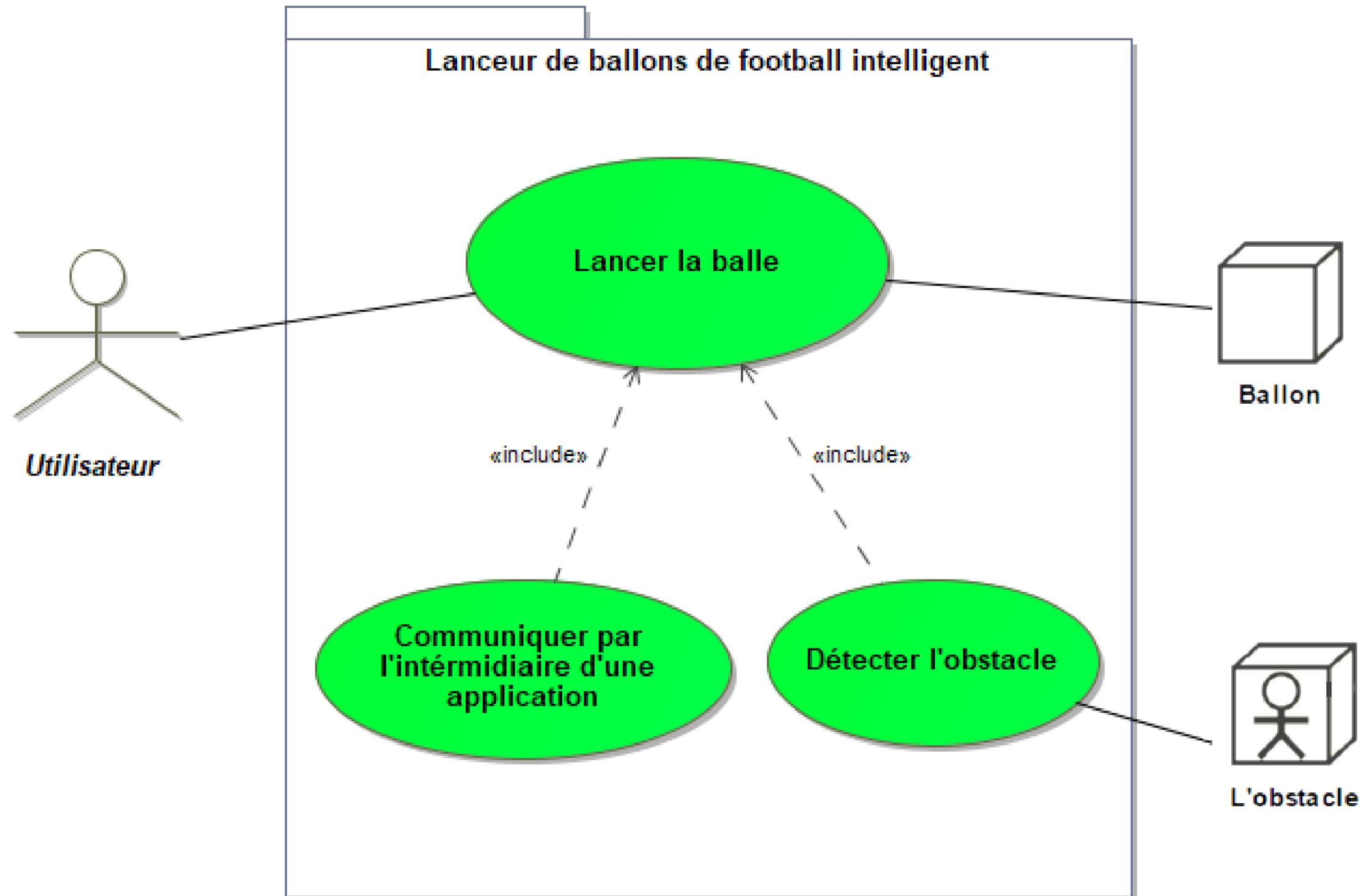


Diagramme des exigences

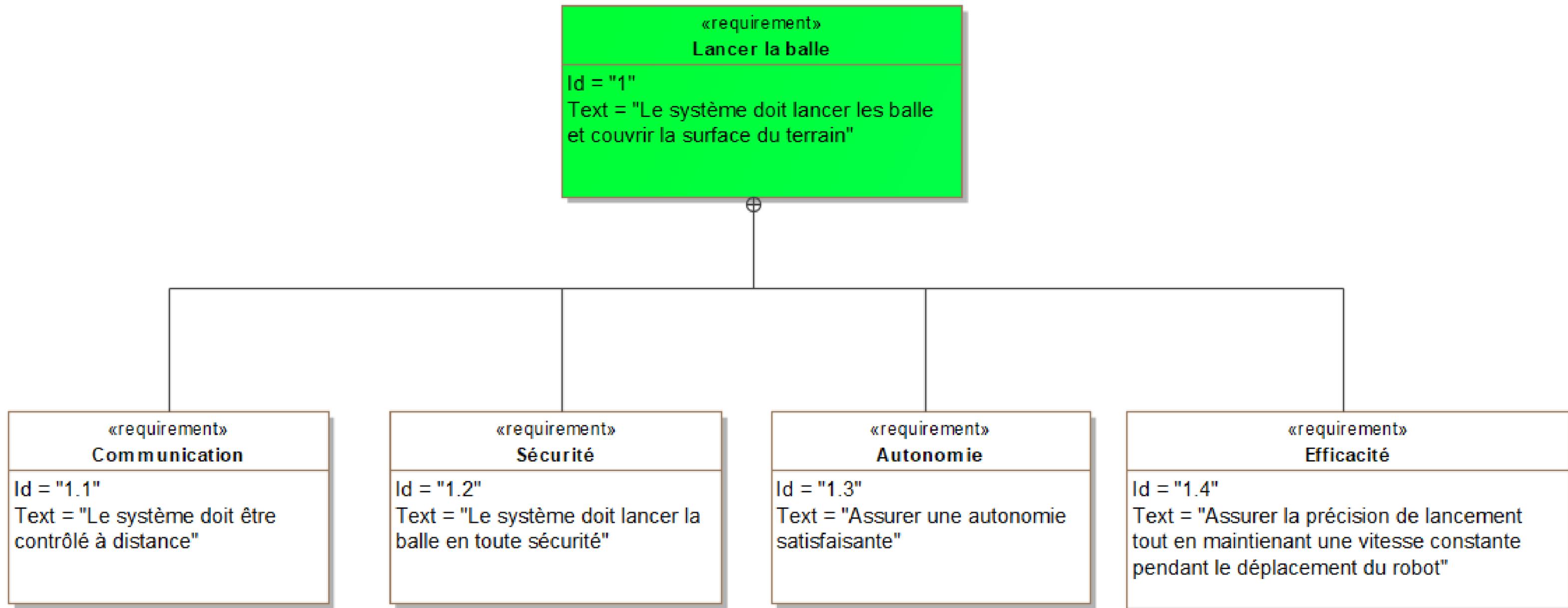


Diagramme des exigences

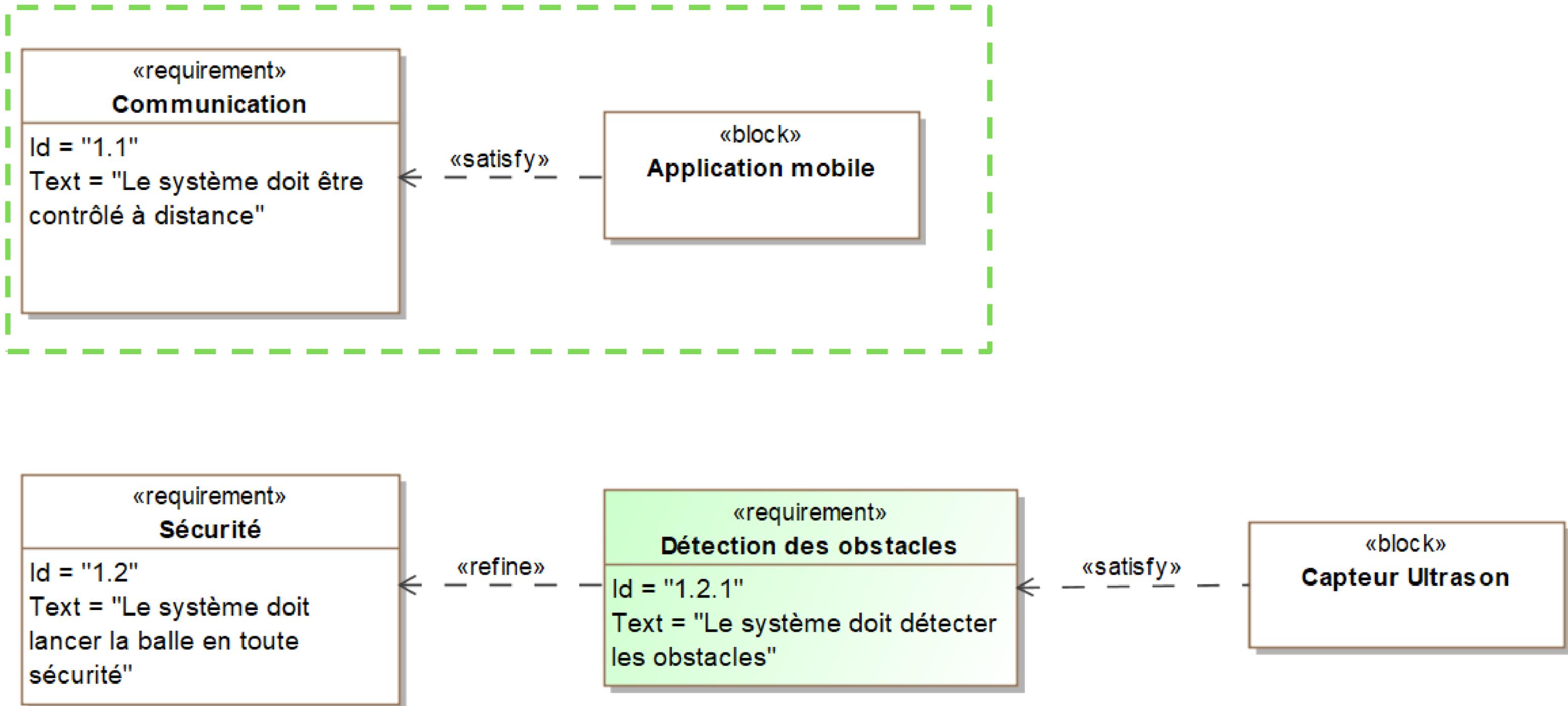
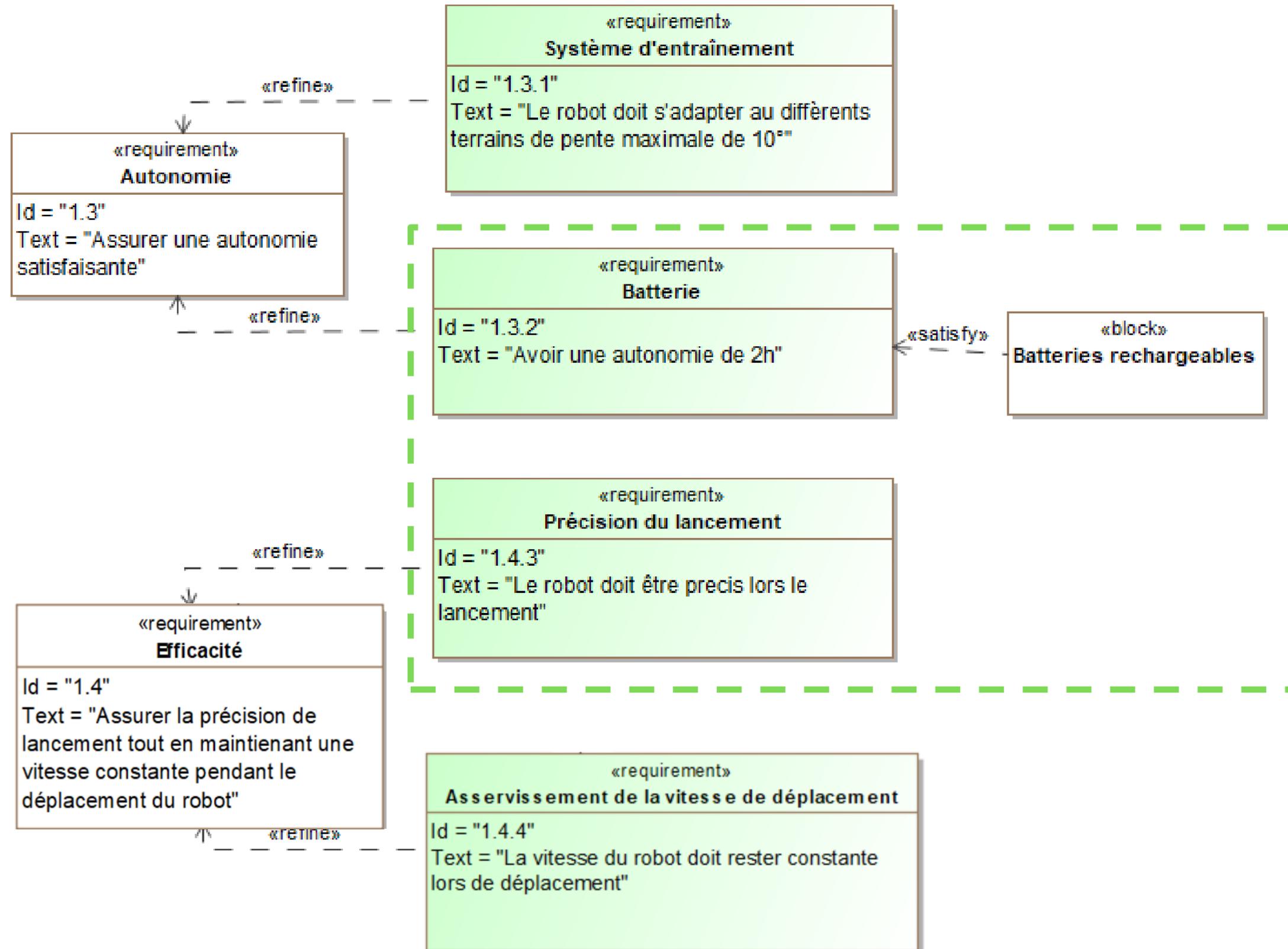


Diagramme des exigences





Premier Objectif:

Principe de fonctionnement du
système

Principe du fonctionnement

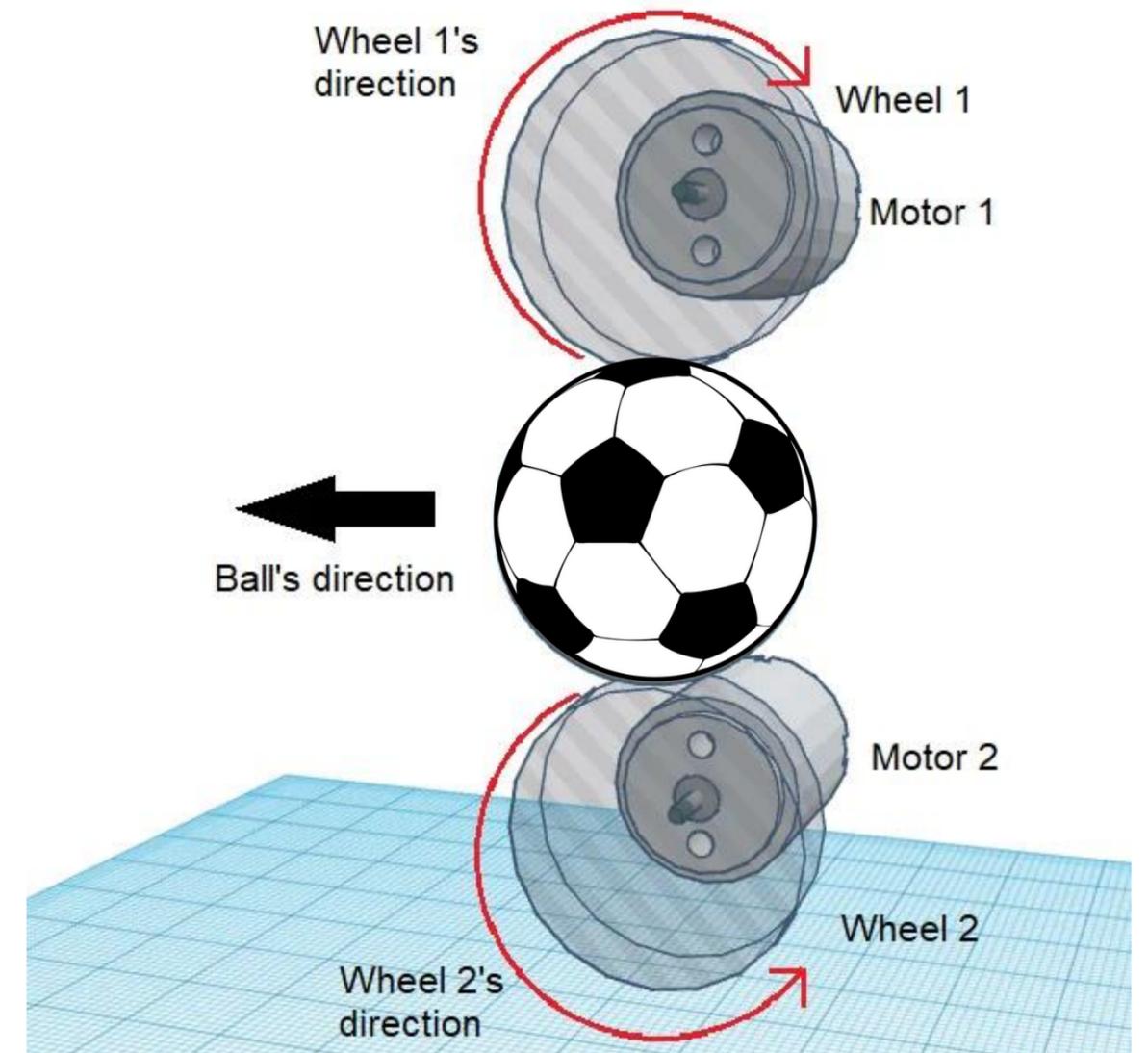
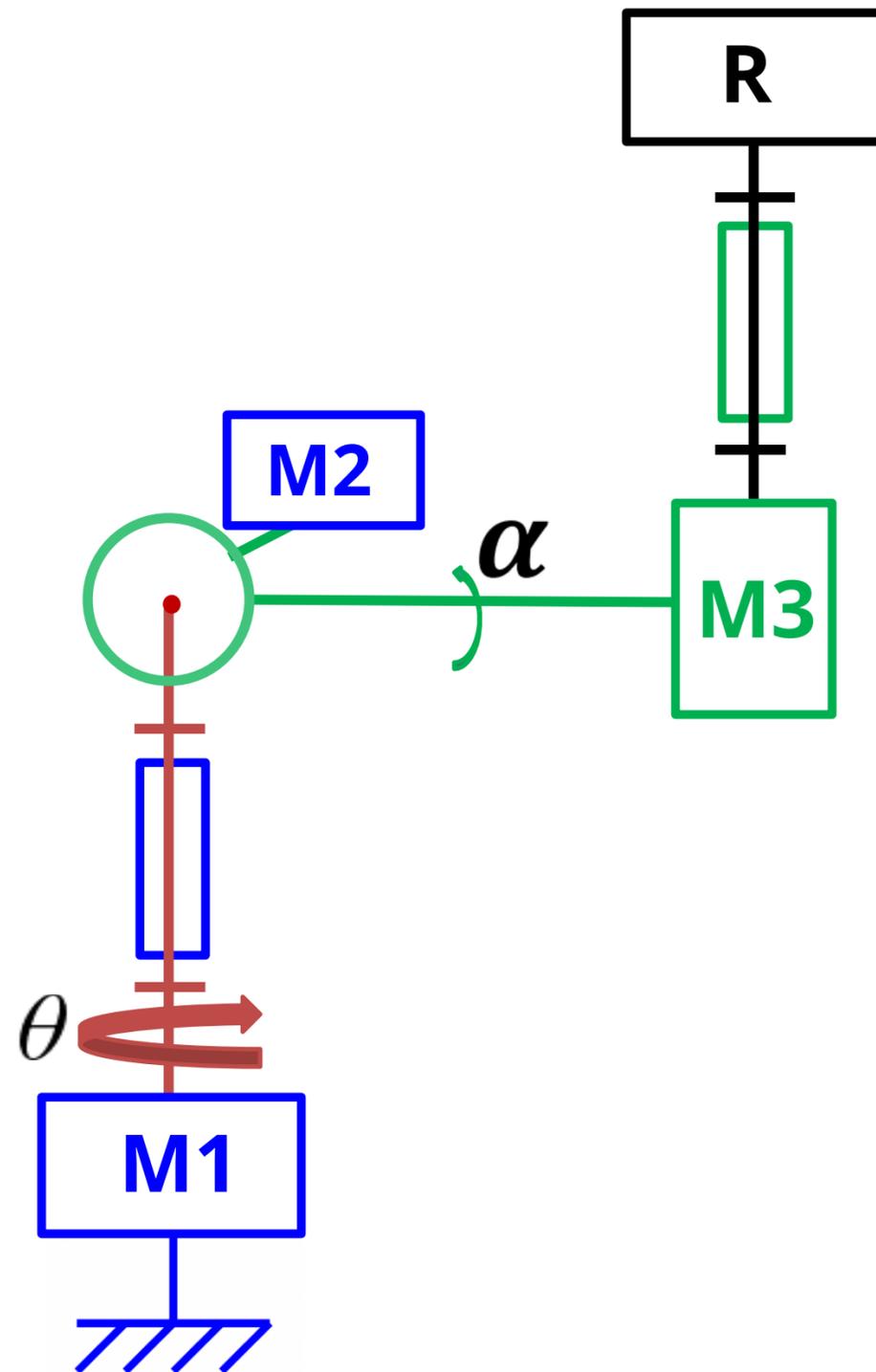


Schéma cinématique:



M1	1er Servo - moteur
M2	2ème Servo - moteur
M3	Moteur de lancement
R	Roue de lancement

Choix du moteur de lancement

□ Des données importantes

- Vitesse maximale du lancement : $V_m = 26,38 \text{ m/s}$ (95 Km/h)
- Masse de la balle : $m = 0,43 \text{ Kg}$
- Rayon de la roue de lancement : $R = 0,1 \text{ m}$

□ Vitesse à l'arbre de la roue

$$\omega_r = \frac{V_m}{R} = \frac{26,38}{0,1} = 263,8 \text{ rad/s}$$

$$\Rightarrow N_r = 2520 \text{ tr/min}$$

□ Puissance nominale

- Énergie cinétique de la balle pour la vitesse maximale :

$$E_c = \frac{1}{2} \times m \times V_m^2 = \frac{1}{2} \times 0,43 \times 26,38^2 = 149 \text{ J}$$

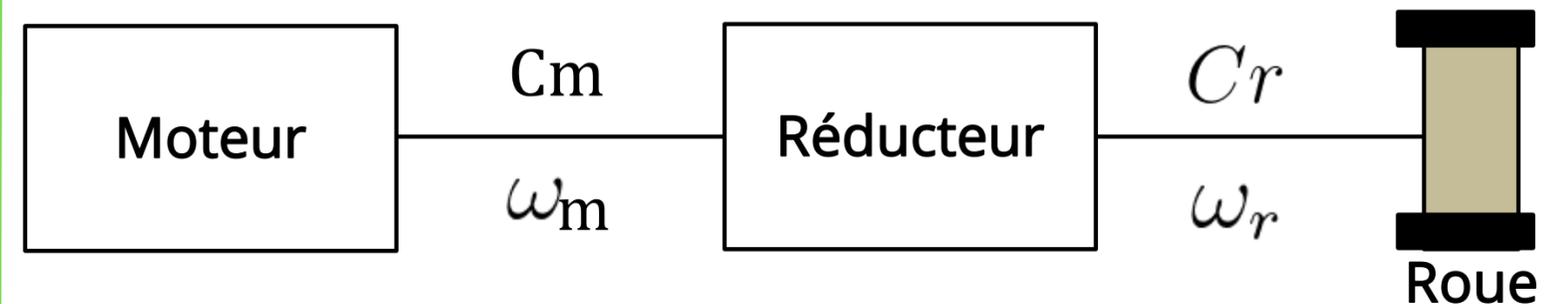
- Si la balle est lancée en 0,43 seconde :

$$P = \frac{E_c}{T} = \frac{149}{0,43} = 346,5 \text{ W}$$

□ Le couple à l'arbre de la roue

$$C_r = \frac{P}{\omega_r} = \frac{346,5}{263,8} = 1,31 \text{ N.m}$$

□ Chaine cinématique



Choix du moteur de lancement

❑ Vitesse à l'arbre de la roue

La vitesse plus rencontrée au marché :
 $N_m = 3000 \text{ tr/min}$

❑ Le rapport de réduction

$$r = \frac{N_m}{N_r} = \frac{3000}{2520} = 1,2$$

❑ Le couple moteur

Puisque : $C_r = 1,31 N.m$ et $r = 1,2$

$$\text{Alors : } C_m = \frac{C_r}{r} = \frac{1,31}{1,2} = 1,1 N.m$$

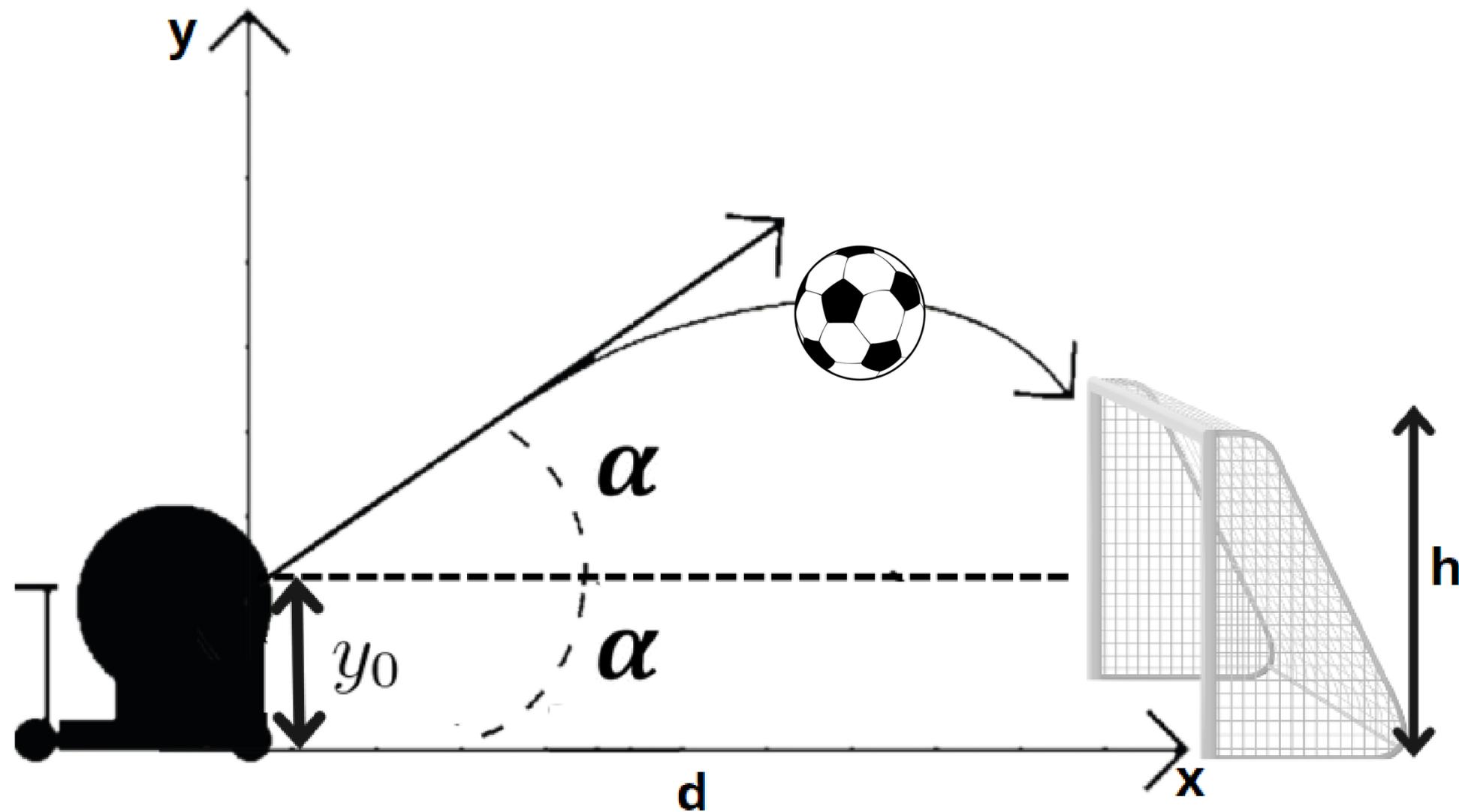
❑ Choix de moteur

Machine à courant continu MY1016 de caractéristiques suivantes :



Model	MY 1016			
Standard	200W 24V/36V	250W 24V/36V	300W 24V/36V	350W 24V/36V
No-load current/A	≤1.5/1.0	≤1.6/1.2	≤1.8/1.4	≤2.0/1.4
No-load rate speed /rpm	3300	3350	3400	3450
Rating Torque/N.m	0.70	0.90	1.04	1.22
Rating speed /rpm	2700	2750	2750	2750
Rating current/A	≤10.6/7.1	≤13.4/8.9	≤16.0/10.7	≤18.7/12.5

Etude de la trajectoire verticale



□ Hypothèses

- Les forces de frottement de l'air ainsi que la poussée d'Archimède sont négligées.
- Système {ballon} de masse m constante et de centre d'inertie G .
- Référentiel terrestre supposé galiléen.

Etude de la trajectoire verticale

- L'expression du vecteur accélération \vec{a} du centre d'inertie du ballon

Le PFD appliqué au ballon s'écrit :

$$\begin{aligned}\sum \vec{F}_{ext} &= m\vec{a}_G \Rightarrow \vec{P} = m\vec{a}_G \\ &\Rightarrow m\vec{g} = m\vec{a}_G \\ &\Rightarrow \vec{a}_G = \vec{g} \quad (1)\end{aligned}$$

- Les équations horaires $x(t)$ et $y(t)$ du mouvement de centre d'inertie G du ballon

• En projetant (1) dans le repère (ox,oy):

$$\Rightarrow a_x = g_x = \frac{dV_x}{dt} = 0$$

$$\Rightarrow a_y = g_y = \frac{dV_y}{dt} = -g$$

• En intégrant deux fois:

$$\Rightarrow x(t) = V_0 \cdot \cos(\alpha) \cdot t \quad (2)$$

$$\Rightarrow y(t) = -\frac{1}{2}gt^2 + V_0 \sin(\alpha)t + y_0 \quad (3)$$

- L'équation de la trajectoire du ballon, dans le plan (xOy), peut s'écrire

• On isole le temps t de l'expression (2) et on remplace l'expression de t dans (3), on obtient :

$$y = -\frac{gx^2}{2V_0^2 \cos^2(\alpha)} + x \tan(\alpha) + y_0 \quad (4)$$

Etude de la trajectoire verticale

□ Etude dans le cas d'un penalty

□ Données nécessaires

- $x=d=11\text{m}$
- $g=9,81\text{N.Kg}^{-1}$
- $y=h=2,44\text{m}$
- $y_0=1,22\text{m}$
- $V_0=26,38\text{m/s}$

□ Recherche de α_{max}

- On remplace les valeurs dans l'expression (4):

$$2,44 = -\frac{9,81 \times 11^2}{2 \times 26,38^2 \times \cos^2(\alpha)} + 11 \times \tan(\alpha) + 1,22$$

- On pose $X=\tan(\alpha)$:

$$1,22 = -0,85 \times (1 + X^2) + 11 \times X$$

$$\Rightarrow -0,85 \times X^2 + 11 \times X - 2,07 = 0$$

- Nous avons une équation quadratique standard de la forme $ax^2 + bx + c = 0$, avec :

$$a=-0,85 \quad b=11 \quad c=-2,07$$

- Le discriminant Δ est donné par : $\Delta = b^2 - 4ac = 114 > 0$
- Les résultats sont données par :

$$X_1 = \frac{-b - \sqrt{\Delta}}{2 \times a} = 12,75 \Rightarrow \tan^{-1}(X_1) \simeq 86^\circ$$

$$X_2 = \frac{-b + \sqrt{\Delta}}{2 \times a} = 0,19 \Rightarrow \tan^{-1}(X_2) \simeq 11^\circ$$

- Le premier angle n'est pas logique, alors on choisissons l'angle : $\alpha_{max} = 11^\circ$

Etude de la trajectoire horizontale

□ Etude dans le cas d'un penalty

□ Données nécessaires

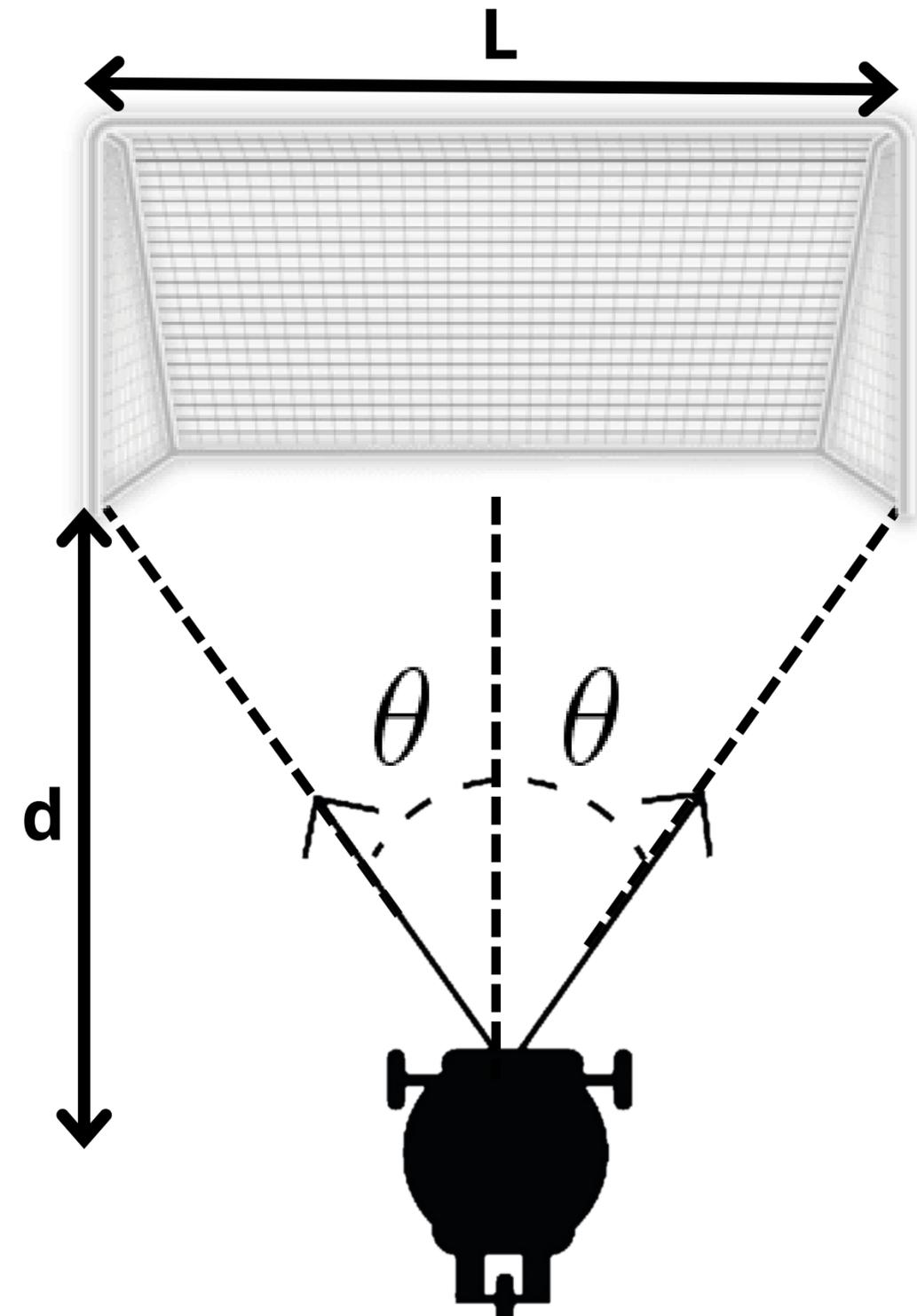
- $d=11\text{m}$
- $L=7,32\text{m}$
- Les forces de frottement de l'air sont négligées.

□ Déterminer l'angle maximal θ_{max}

- On a la moitié de la largeur du but : $\frac{7,32\text{ m}}{2} = 3,66\text{ m}$

$$\text{Alors : } \tan(\theta_{max}) = \frac{3,66}{11} \Rightarrow \theta_{max} = \tan^{-1}\left(\frac{3,66}{11}\right)$$

Donc, nous obtenons : $\theta_{max} \simeq 18^\circ$



Choix des servo moteurs

❑ DYNAMIXEL PM54-060-S250-R



Model Name		PM54-060-S250-R
Series		DYNAMIXEL P
MCU		ARM Cortex-M4 (168 [Mhz], 32 [bit])
Input Voltage	Min. [V]	-
	Recommended [V]	24.0
	Max. [V]	-
Performance Characteristics	Voltage [V]	24.0
	Stall Torque [N·m]	<u>17.7</u>
	Stall Current [A]	7.2
	No Load Speed [rpm]	33.1
	No Load Current [A]	1.25
Continuous Operation	Voltage [V]	24.0
	Torque [N·m]	<u>10.1</u>
	Speed [rpm]	28.3
	Current [A]	3.0

❑ HS-1100WP Servo



Voltage Range	11.1V - 14.8V
No-Load Speed (11.1V)	0.26 sec/60°
No-Load Speed (14.8V)	0.19 sec/60°
Stall Torque (11.1V)	8.24 N.m
Stall Torque (14.8V)	<u>10.8 N.m</u>
Current Draw at Idle	130 mA
Rated Current	3100 mA



Deuxième Objectif:

Étude électrique du système

Choix de batterie

❑ Bilan de courant des équipements du robot

Deux moteurs de lancement (24V)	37,4A
Deux servo moteurs (14,8V) & (24V)	6,1A
Moteur de déplacement (24V)	3,5A
Carte wifi ESP8266 (3.3V)	70 mA
Détecteur d'obstacle (5V)	15 mA
Le courant totale	47,085A

Donc : $C = 47,085 \times 2 = 94,17 \text{ Ah}$

❑ Le batterie choisie

LiTime Mini : LiFePO4 Lithium Battery, 12V 100Ah

Deux batteries en série :

Capacité	100Ah
Tension	12V
Poids	8Kg



Le robot fonctionne en plein régime pendant 2 heures :

On définit la capacité : $C = I \times \Delta t$

- I: Le courant totale consommé par le robot
- Δt : L'autonomie du robot

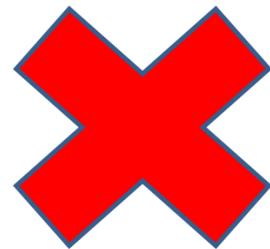
Choix de batterie

❑ Raison de choix



LiTime LiFePO4 Lithium Battery, 24V 100Ah

Capacité	100Ah
Tension	24V
Poids	20Kg



LiTime Mini : LiFePO4 Lithium Battery, 12V 100Ah

Capacité	100Ah
Tension	12V
Poids	8Kg



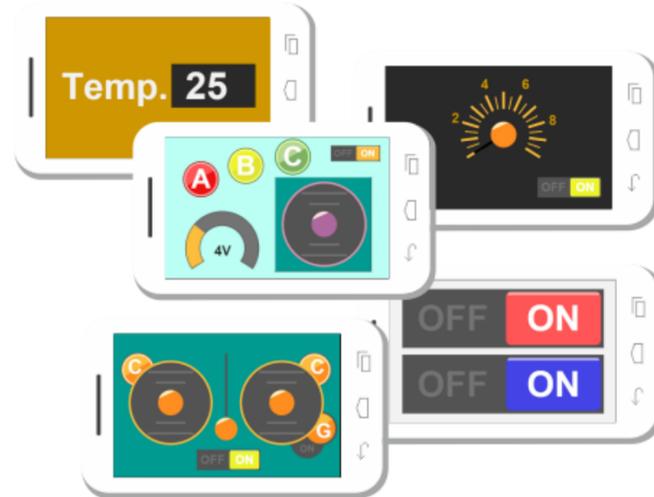


Troisième Objectif:

Commande à distance du robot

Application

❑ Remote XY ?



❑ Configuration au niveau du site

▼ Configuration

-  Wi-Fi access point
-  NodeMCU V3
-  Integrated WiFi
-  Arduino IDE

▼ Settings

Access password:

Wi-Fi access point:
Name (SSID):

Open point

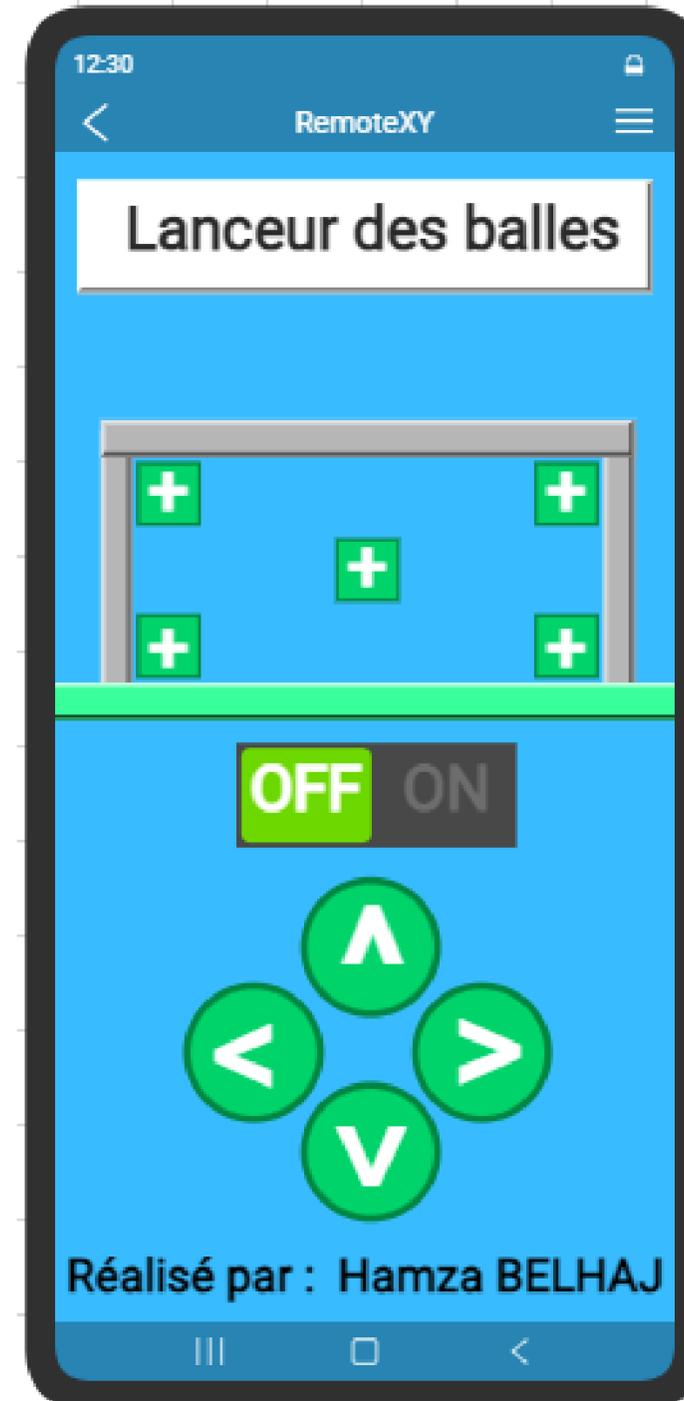
Password (8 or more chars):

Port:

❑ Conception de l'application

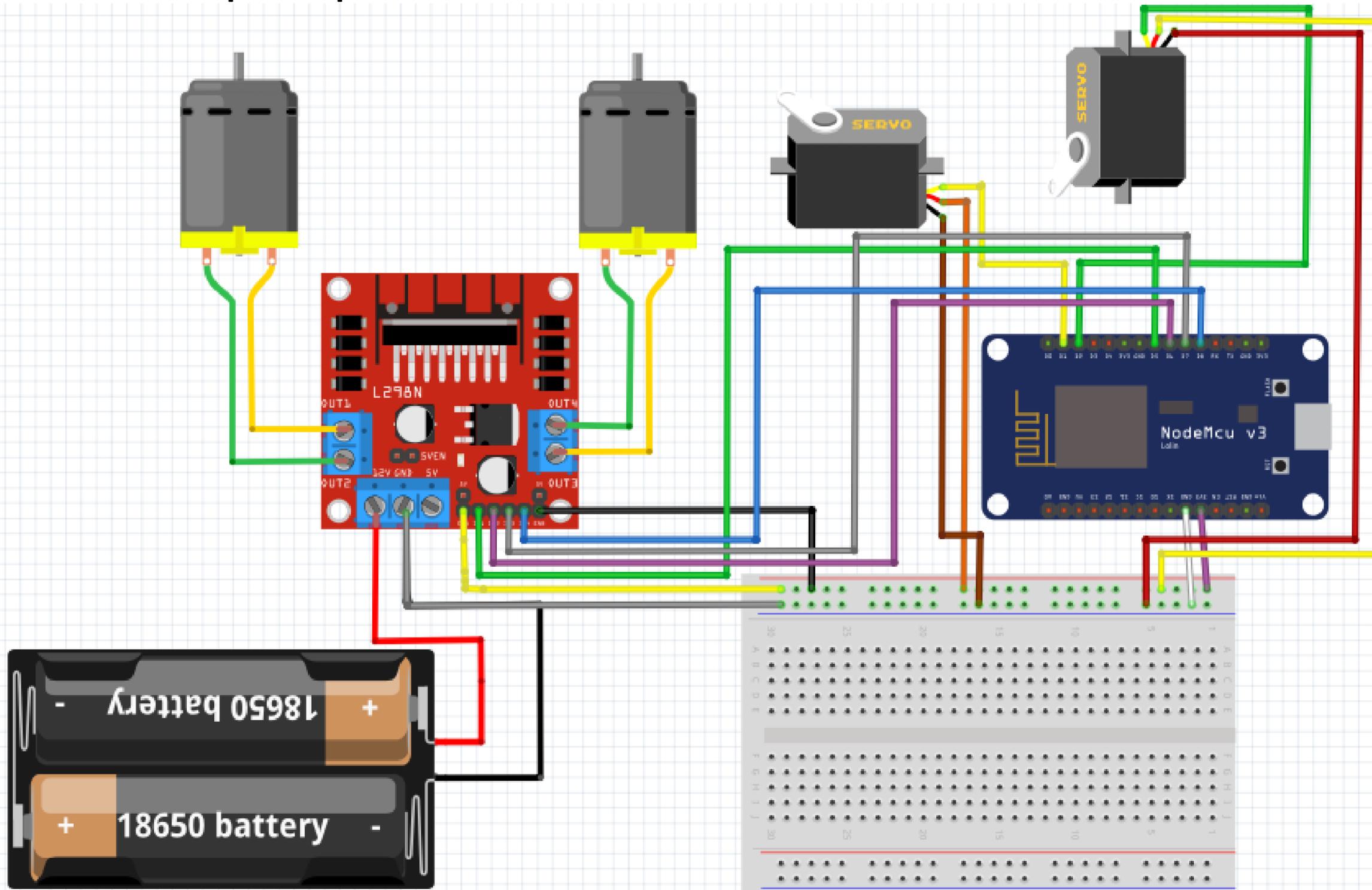
Description :

- Choisir un angle prédéterminé
- Choisir un angle manuellement
- Allumer et éteindre les moteurs de lancement

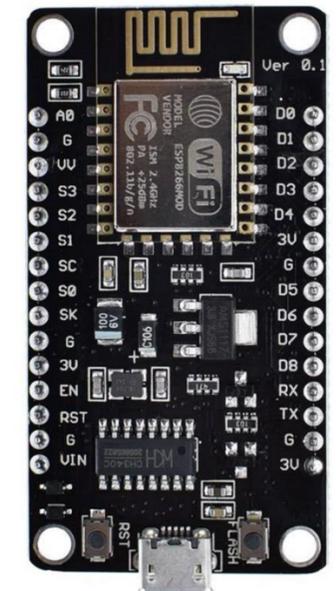


Prototype

❑ Schéma de principe

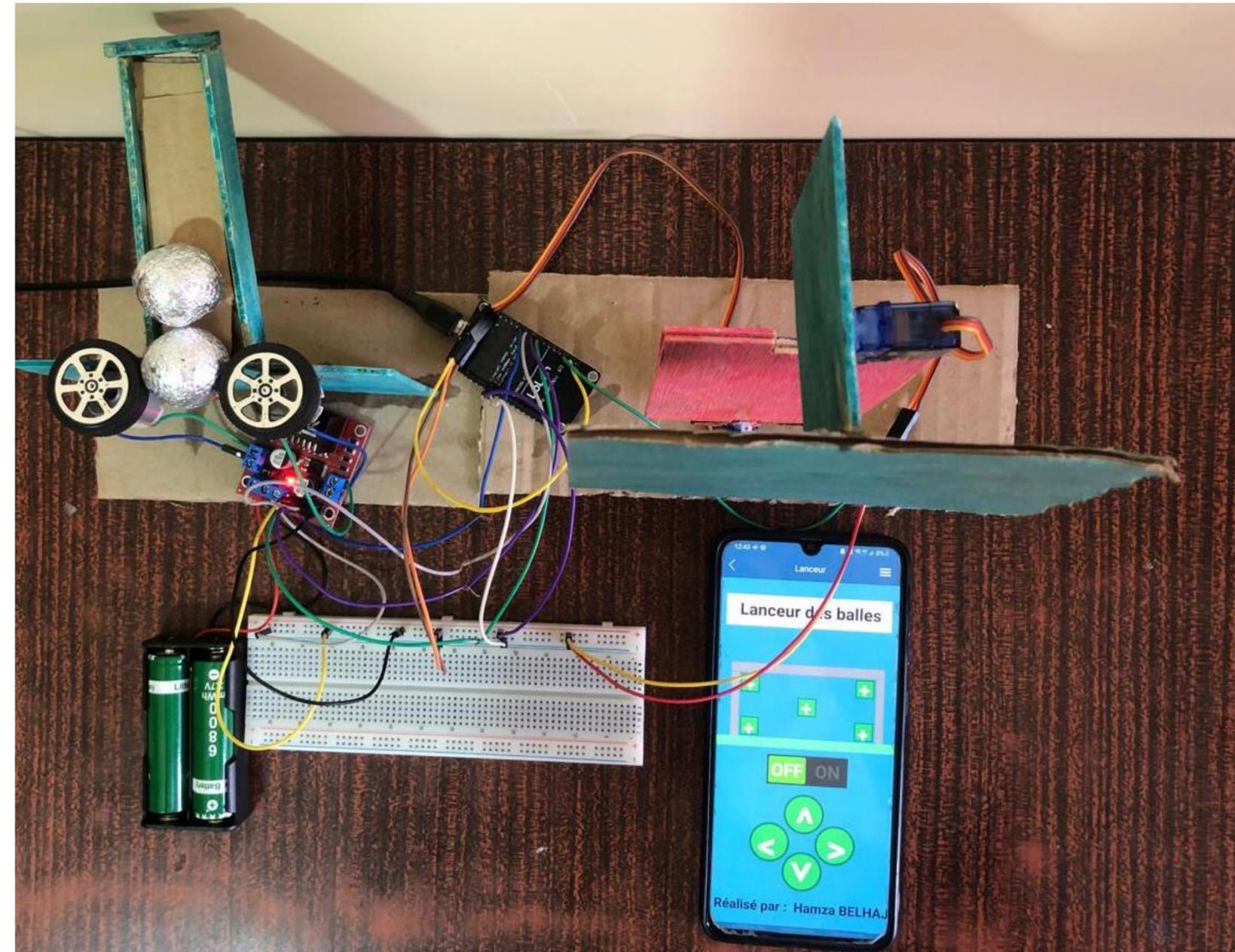
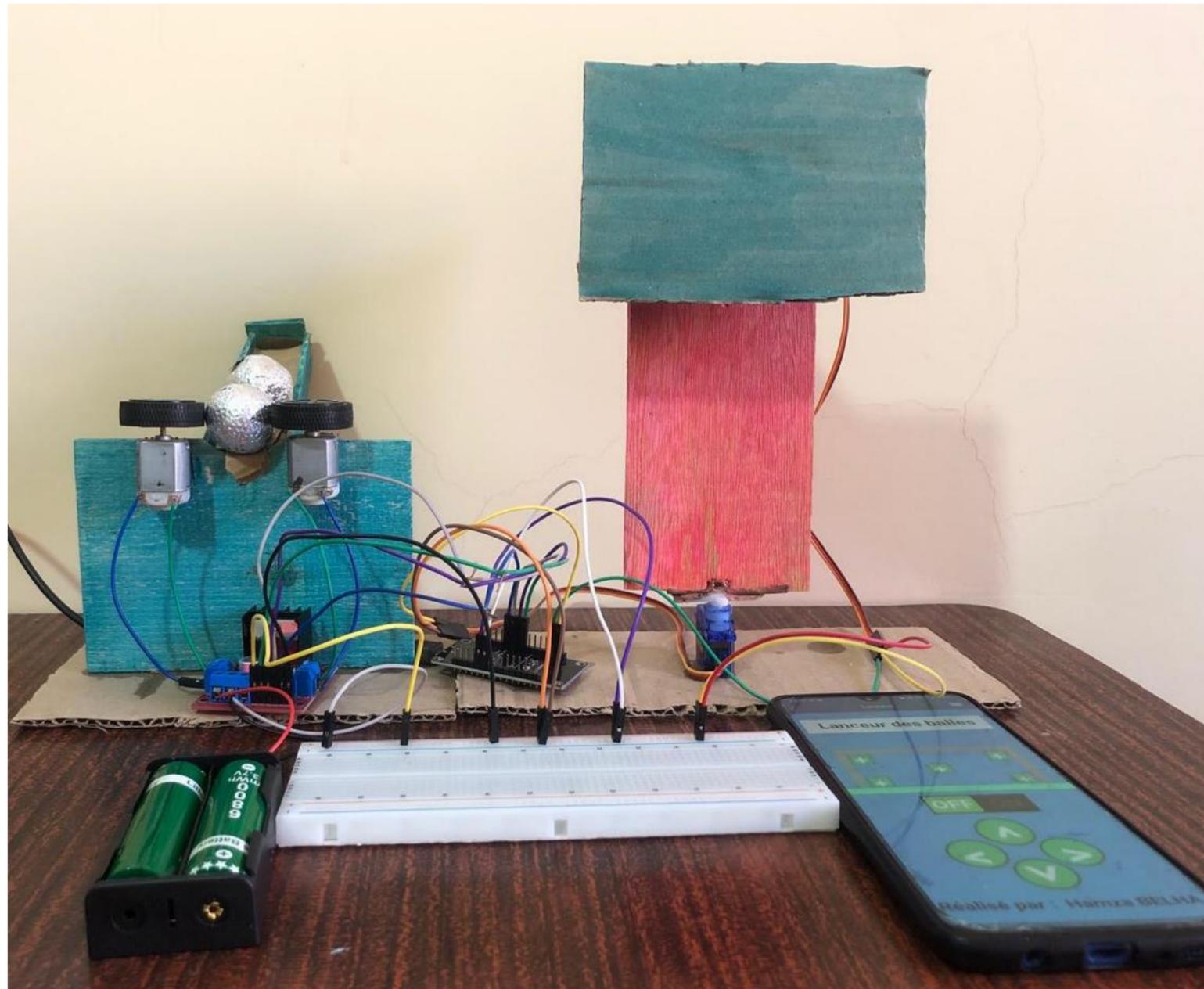


NodeMCU ESP8266



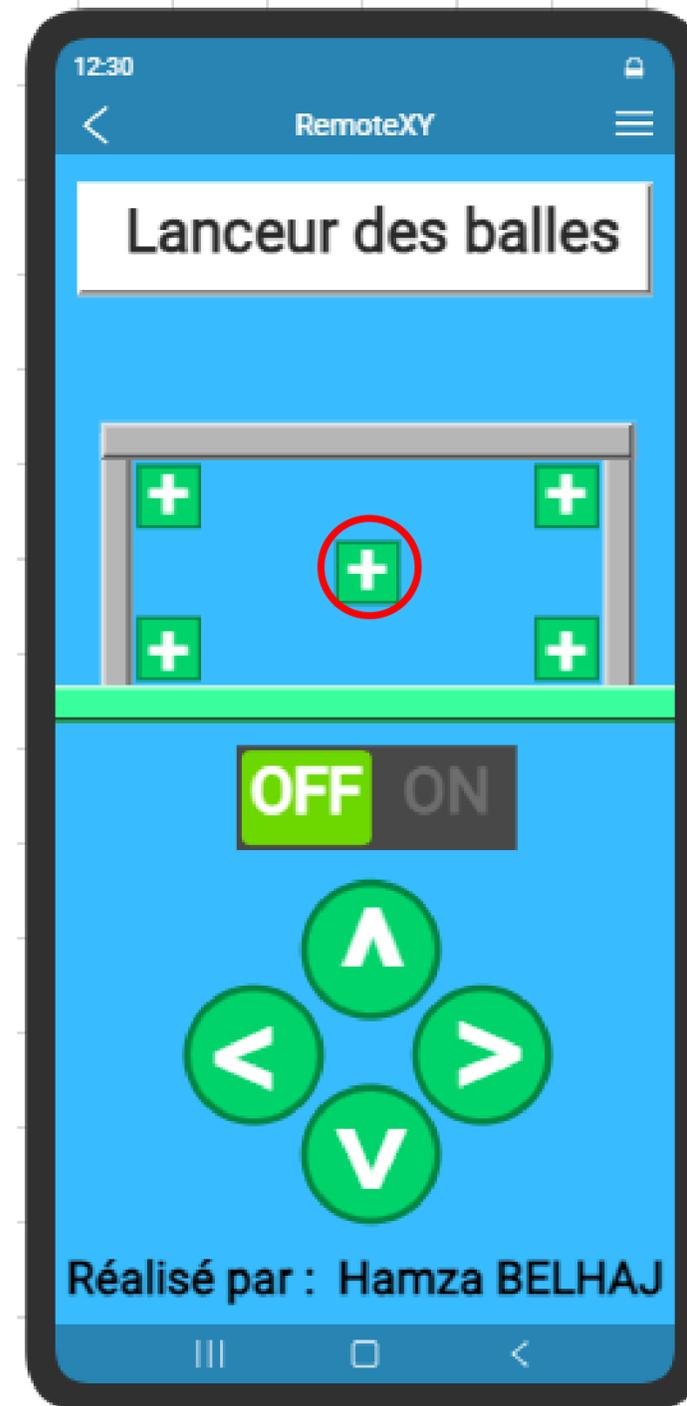
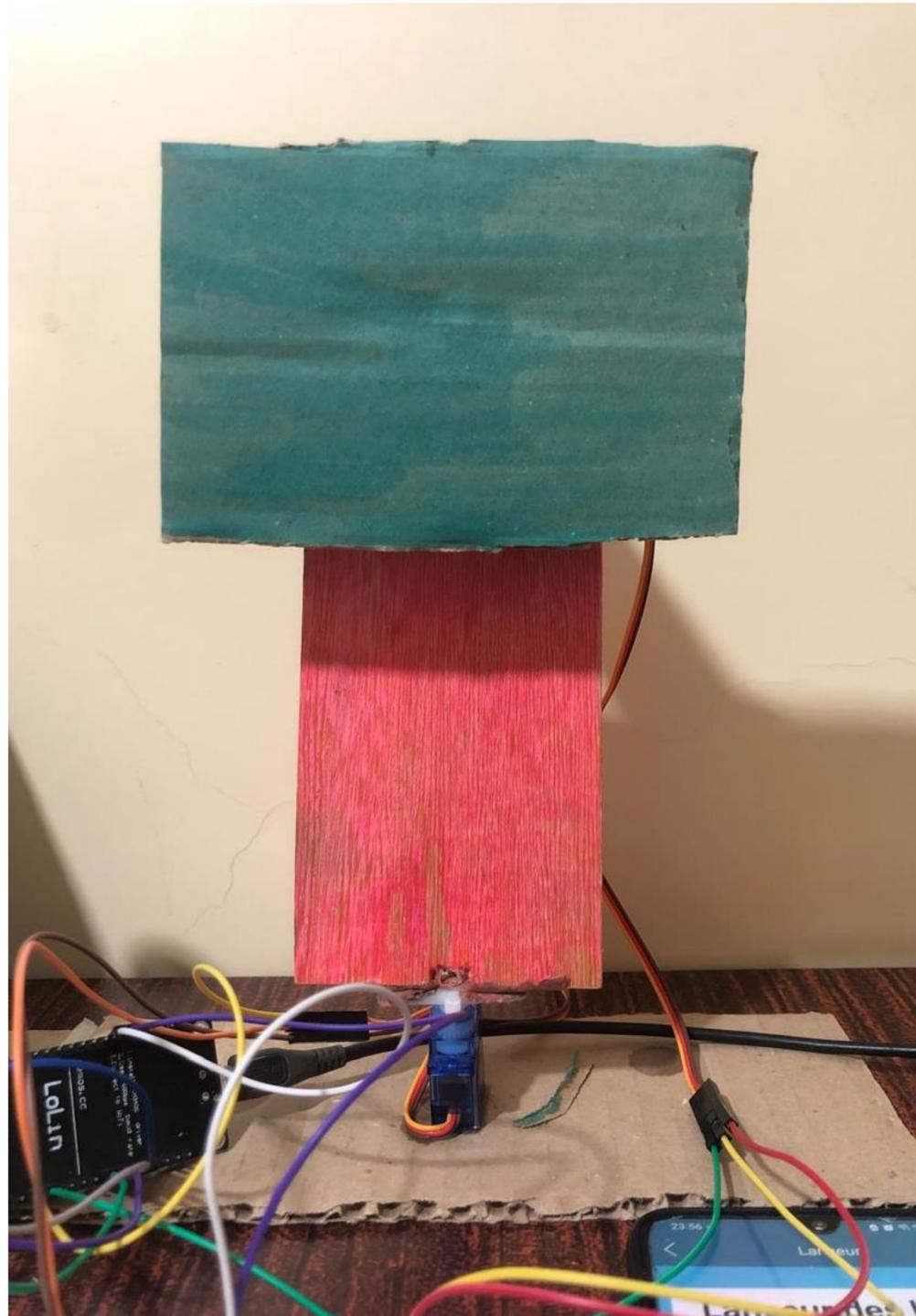
Prototype

□ Schéma de principe



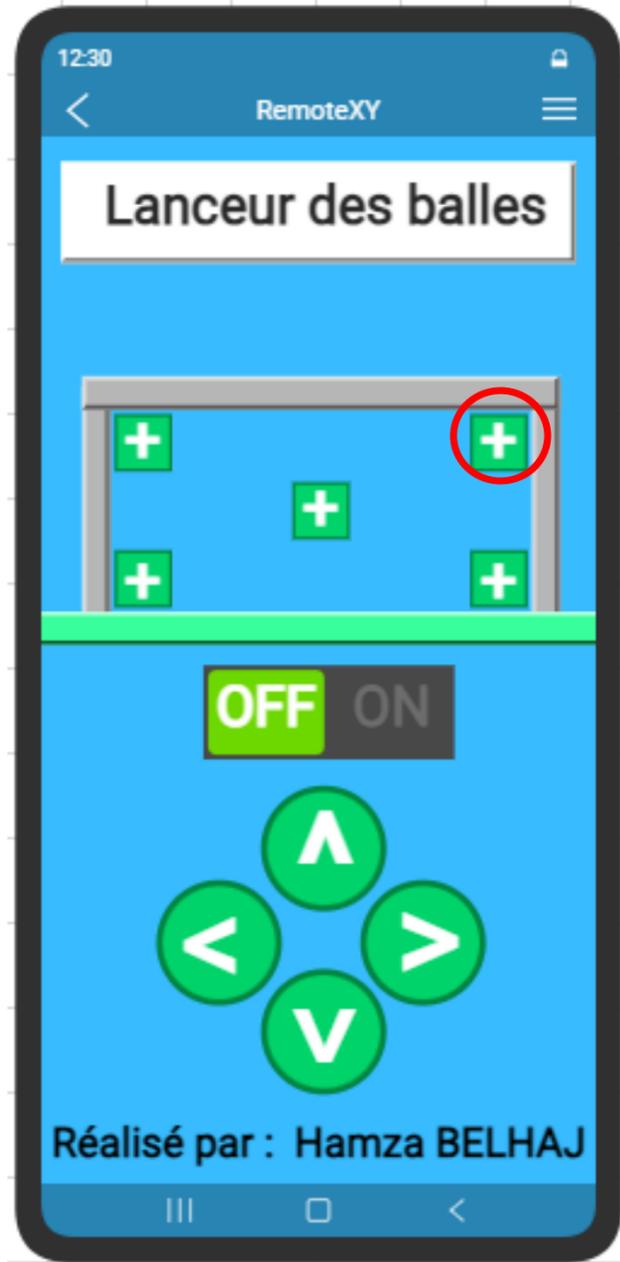
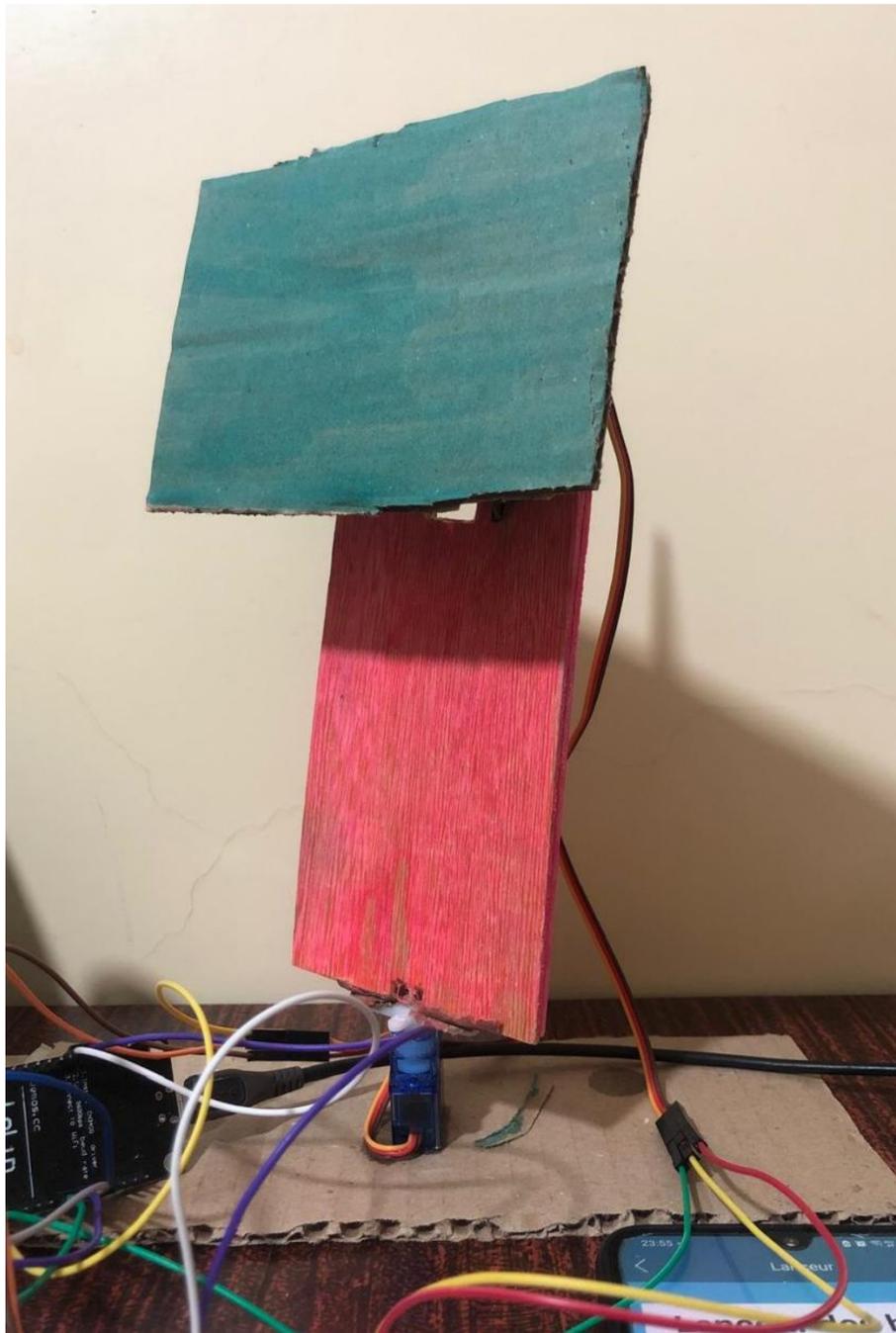
Prototype

- ❑ Viser au centre

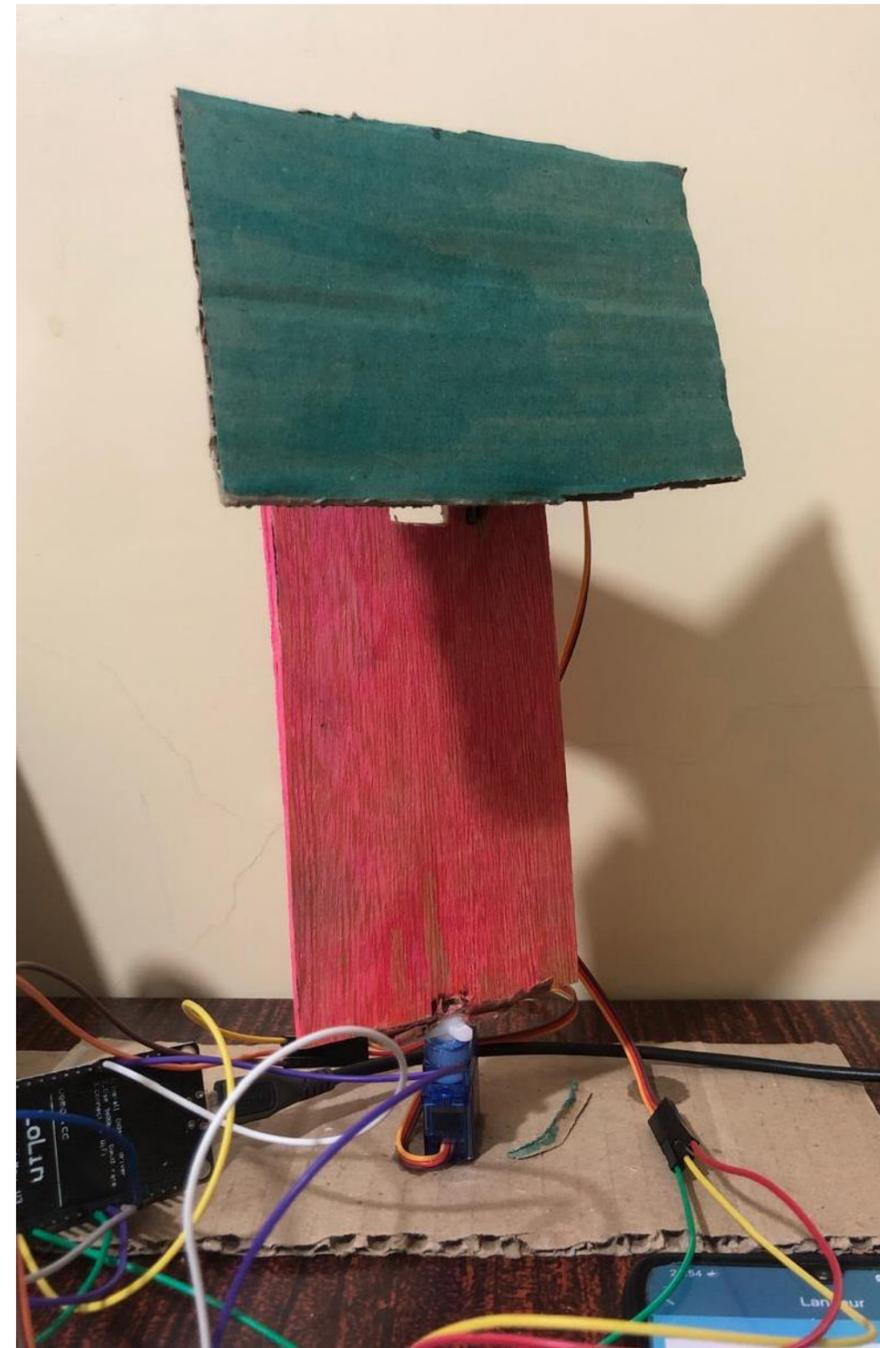


Prototype

☐ Viser en haut à droite

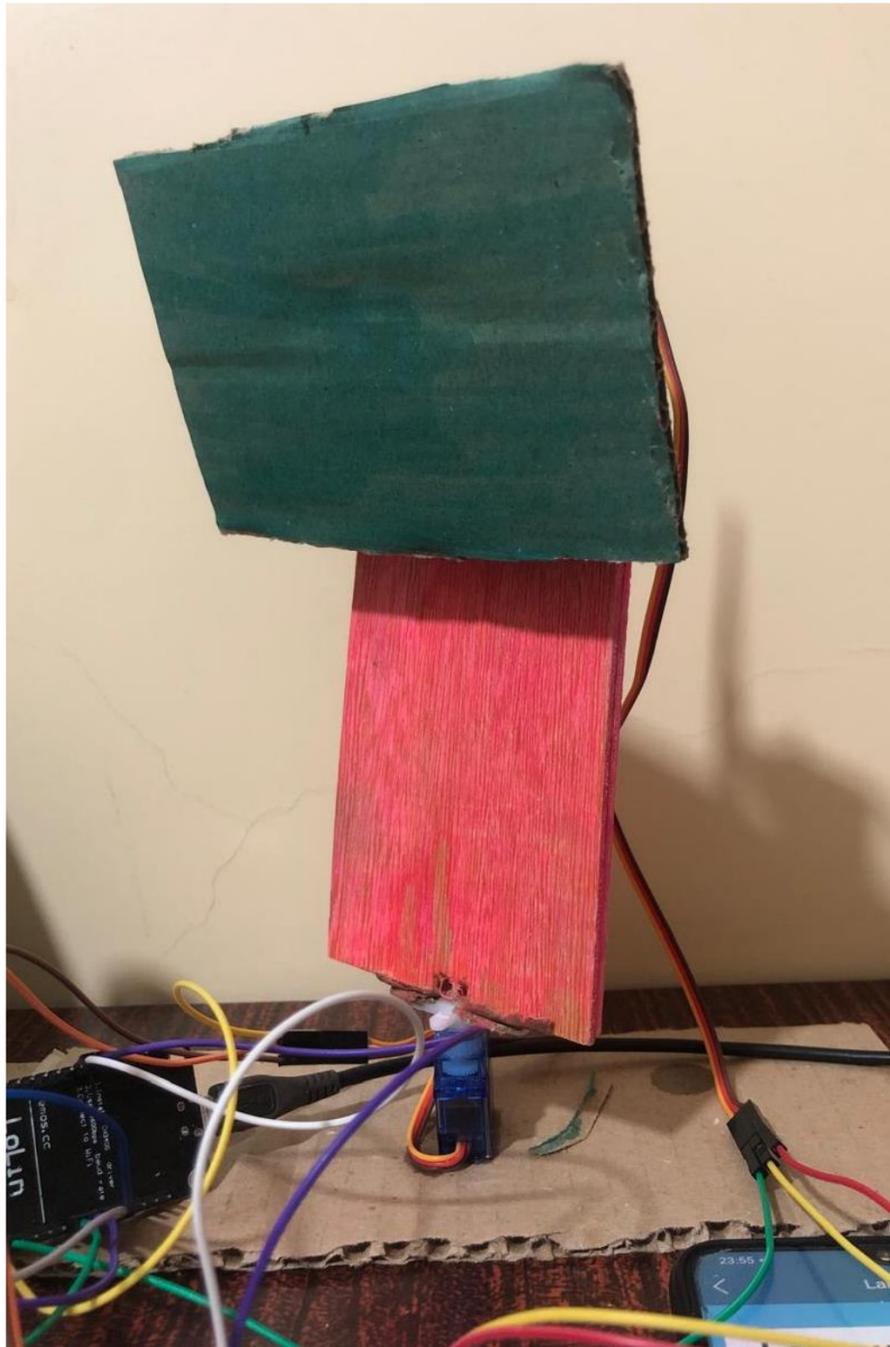


☐ Viser en haut à gauche

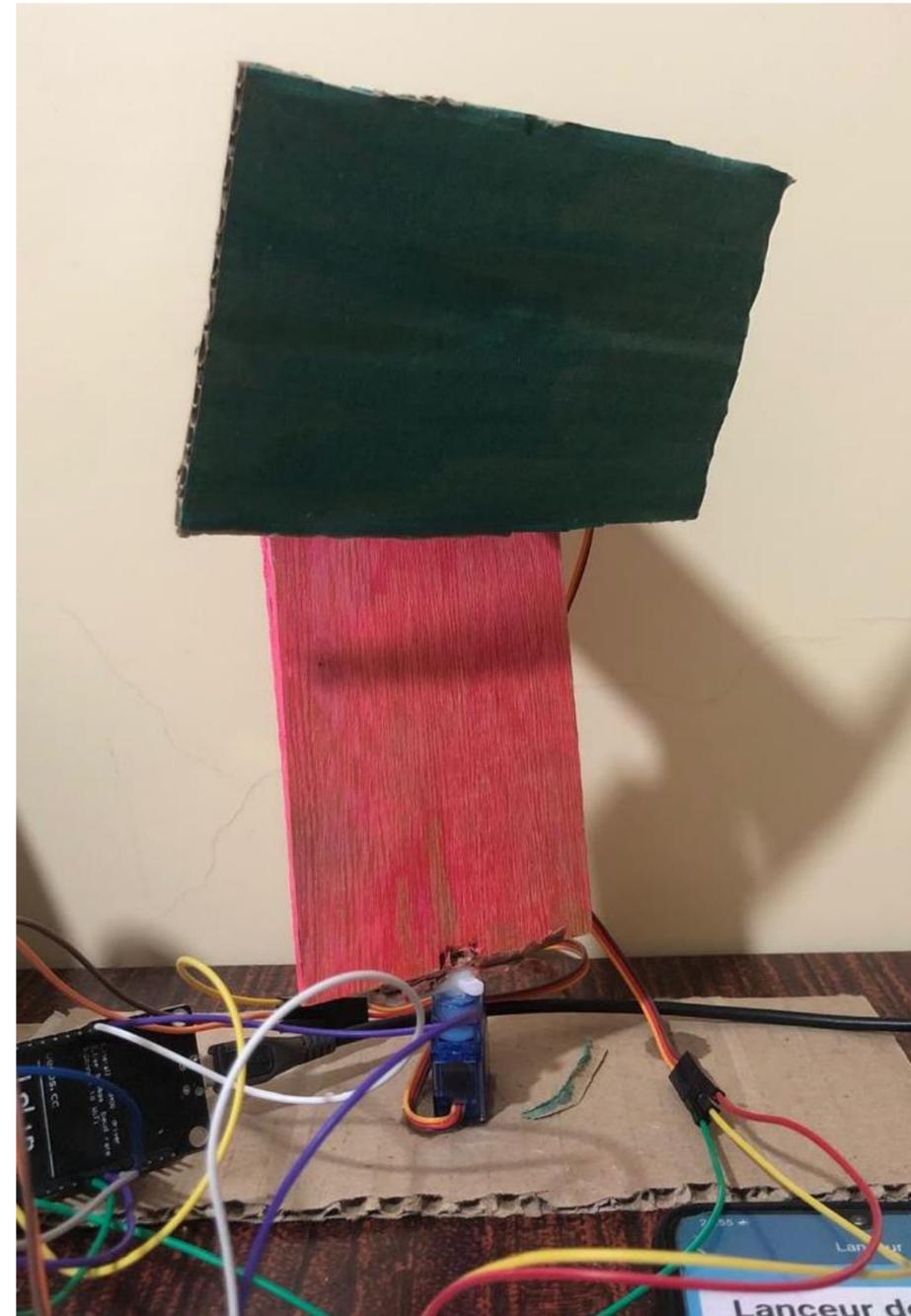


Prototype

❑ Viser en bas à droite

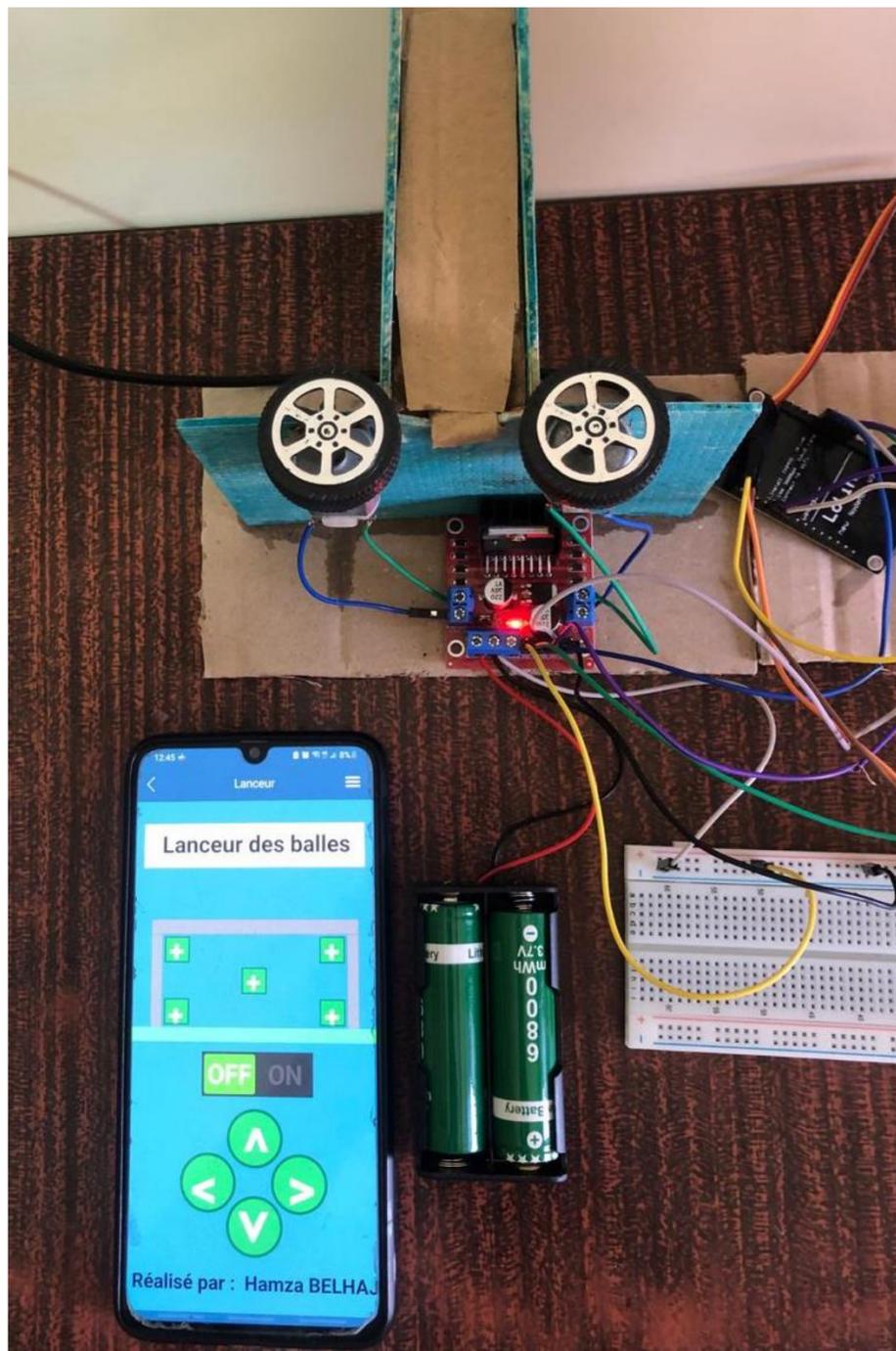


❑ Viser en bas à gauche

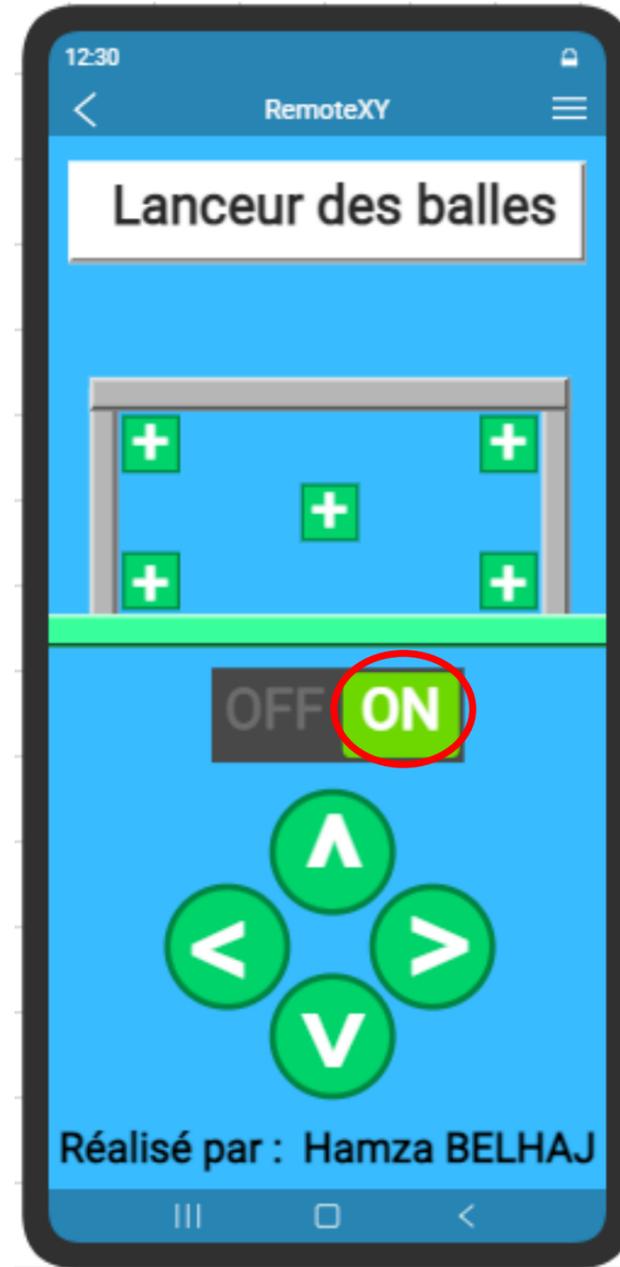
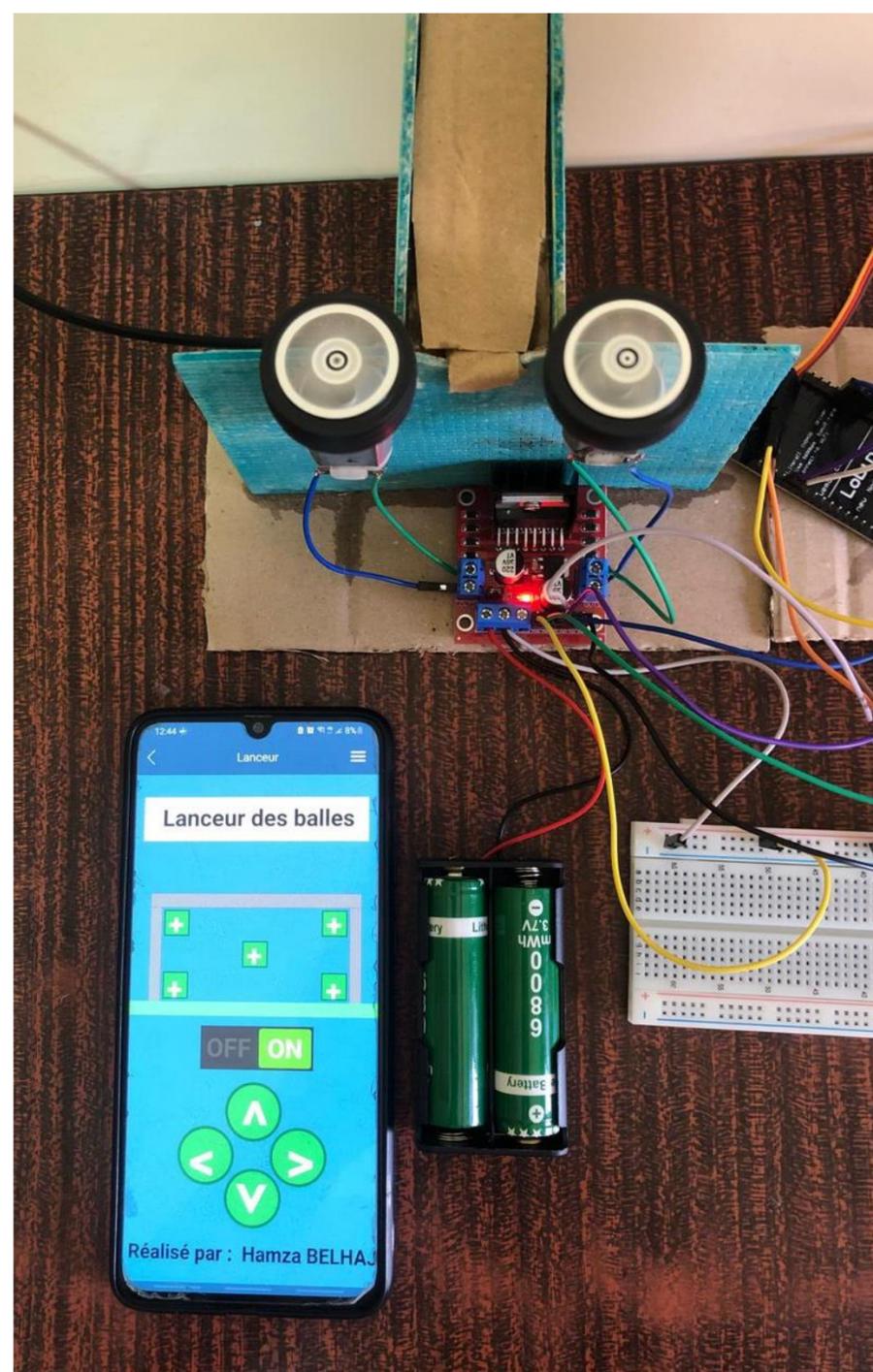


Prototype

❑ Éteindre les moteurs

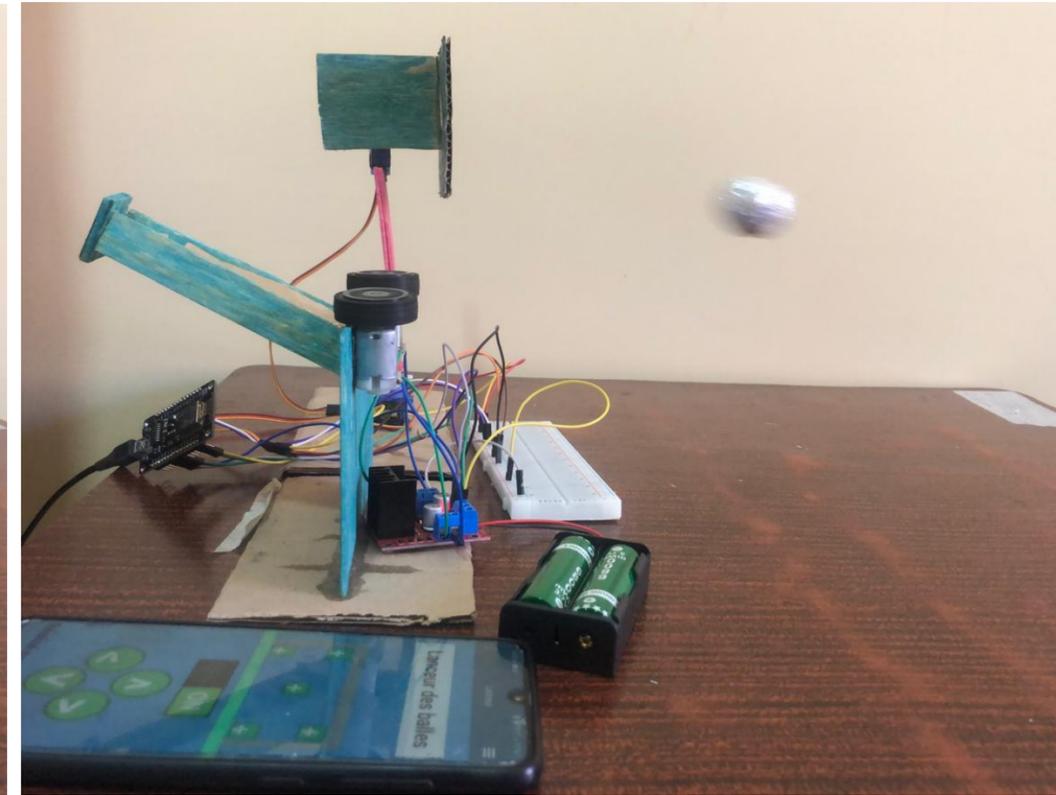
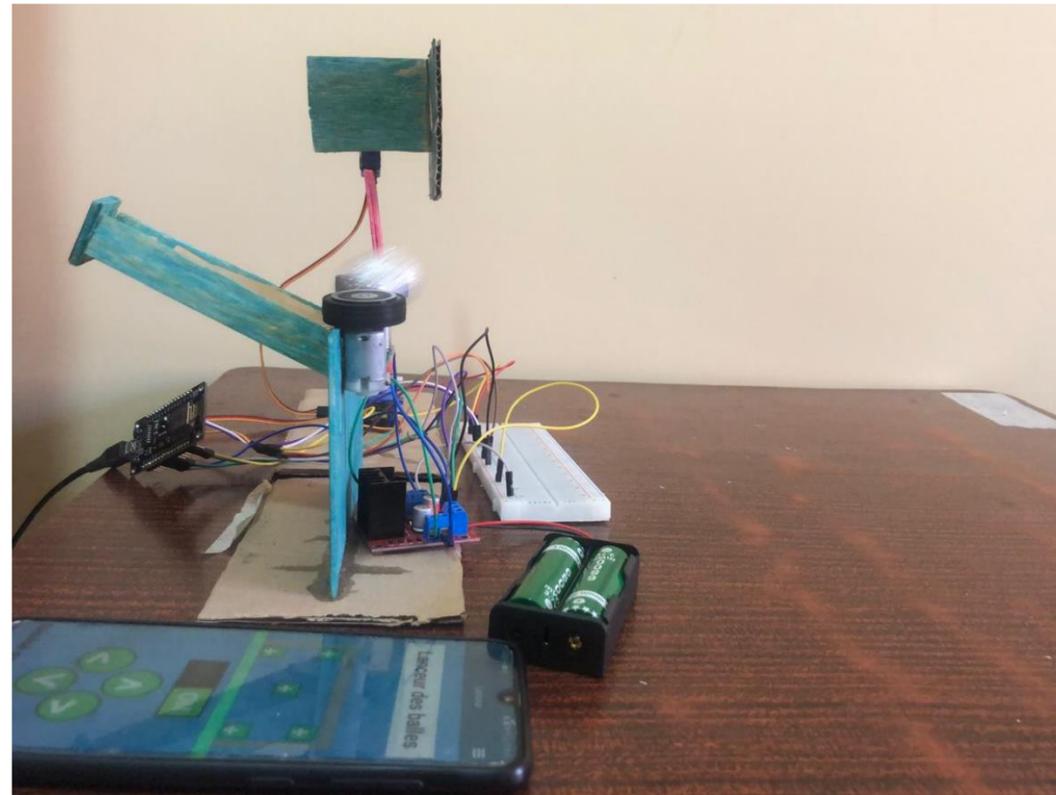
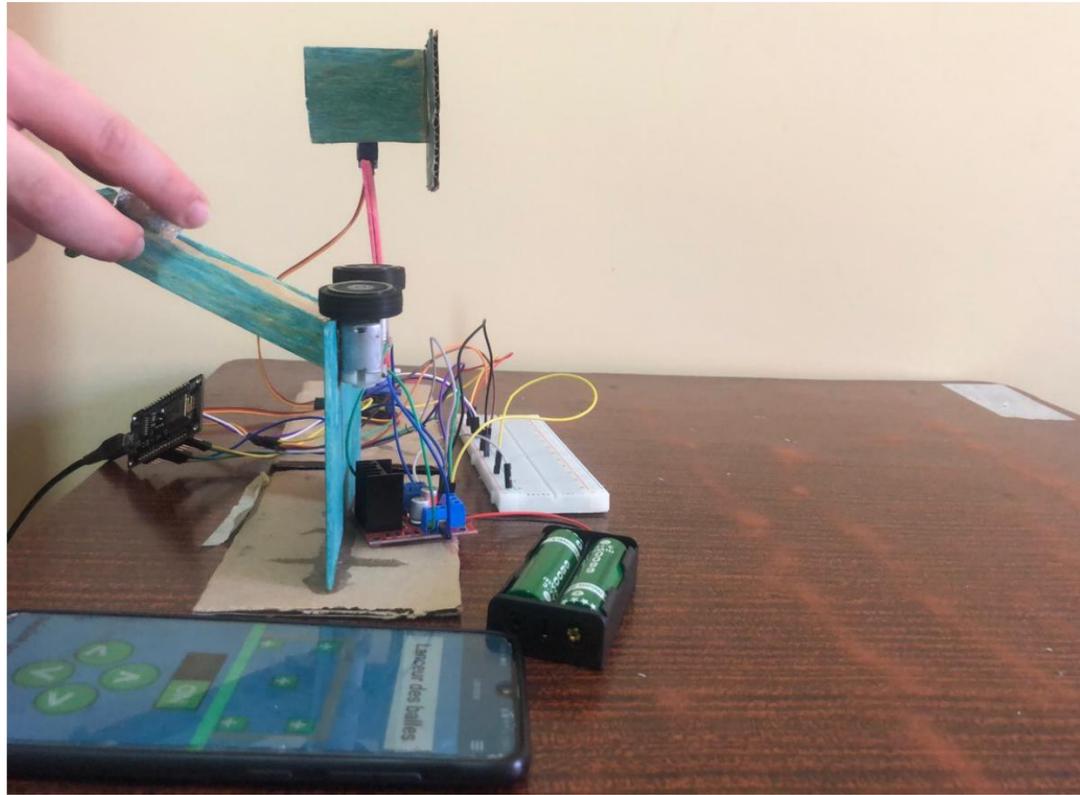


❑ Allumer les moteurs



Prototype

□ Tirer une balle



Conclusion

En conclusion, mon travail sur le "lanceur de balles de football intelligent" a exploré les aspects clés du concept. J'ai étudié comment le lanceur tire et oriente les balles, fait une étude électrique du système et réalisé un prototype qui a prouvé la viabilité du concept. L'ensemble de ces réalisations constitue une avancée majeure vers l'amélioration des compétences des footballeurs et la qualité de leur entraînement.

Merci pour votre attention!



Annexe

❑ Code de l'application :

```
1 // you can enable debug logging to Serial at 115200
2 //#define REMOTEXY_DEBUGLOG
3
4 // RemoteXY select connection mode and include library
5 #define REMOTEXY_MODE__WIFI_POINT
6
7 #include <ESP8266WiFi.h>
8
9 // RemoteXY connection settings
10 #define REMOTEXY_WIFI_SSID "Lanceur"
11 #define REMOTEXY_WIFI_PASSWORD "12345678"
12 #define REMOTEXY_SERVER_PORT 6377
13
14 #include <RemoteXY.h>
15
16 // RemoteXY GUI configuration
17 #pragma pack(push, 1)
18 uint8_t RemoteXY_CONF[] = // 287 bytes
19 { 255,10,0,0,0,24,1,17,0,0,0,178,1,106,200,1,1,26,0,130,
20 94,48,5,44,0,29,130,8,48,5,44,0,29,130,251,91,114,6,0,150,
21 130,8,46,91,6,0,29,130,4,5,98,19,0,31,129,12,9,84,10,8,
22 76,97,110,99,101,117,114,32,100,101,115,32,98,97,108,108,101,115,0,1,
23 14,53,11,11,3,148,31,0,1,14,79,11,11,3,148,31,0,1,48,66,
24 11,11,3,148,31,0,1,82,53,11,11,3,148,31,0,1,82,79,11,11,
25 3,148,31,0,129,84,80,7,12,31,43,0,129,50,66,7,12,31,43,0,
26 129,16,80,7,12,31,43,0,129,16,53,7,12,31,43,0,129,84,53,7,
27 12,31,43,0,1,42,124,24,24,0,148,31,0,1,22,142,24,24,0,148,
28 31,0,1,61,142,24,24,0,148,31,0,1,42,159,24,24,0,148,31,0,
29 129,27,145,12,22,31,60,0,129,49,128,11,26,31,94,0,129,46,150,17,
30 28,31,226,140,132,0,129,68,144,13,23,31,62,0,2,31,101,48,18,0,
31 120,26,31,31,79,78,0,79,70,70,0,129,2,189,104,8,24,82,195,169,
32 97,108,105,115,195,169,32,112,97,114,32,58,32,32,72,97,109,122,97,32,
33 66,69,76,72,65,74,0 };
34
35 // this structure defines all the variables and events of your control interface
36 struct {
37
38     // input variables
39     uint8_t button_01; // =1 if button pressed, else =0
40     uint8_t button_02; // =1 if button pressed, else =0
41     uint8_t button_03; // =1 if button pressed, else =0
42     uint8_t button_04; // =1 if button pressed, else =0
43     uint8_t button_05; // =1 if button pressed, else =0
44     uint8_t button_06; // =1 if button pressed, else =0
45     uint8_t button_07; // =1 if button pressed, else =0
46     uint8_t button_08; // =1 if button pressed, else =0
47     uint8_t button_09; // =1 if button pressed, else =0
48     uint8_t switch 01; // =1 if switch ON and =0 if OFF
49
50     // other variable
51     uint8_t connect_flag; // =1 if wire connected, else =0
52
53 } RemoteXY;
54 #pragma pack(pop)
55
56 ////////////////////////////////////////////////////
57 //                      END RemoteXY include                      //
58 ////////////////////////////////////////////////////
59
60 #include <Servo.h>
61 int servolPin = D1;
62 int servo2Pin = D2;
```

Annexe

❑ Code de l'application:

```
63     Servo servol,servo2;
64
65     int angle = 90;
66     int c=0,i=0;
67     int j=0;
68
69     int motor1Pin1 = D5;
70     int motor1Pin2 = D6;
71     int motor2Pin1 = D7;
72     int motor2Pin2 = D8;
73
74
75 void setup()
76 {
77     RemoteXY_Init ();
78     servol.attach(servolPin);
79     servo2.attach(servo2Pin);
80
81     servol.write(angle);
82     servo2.write(20);
83
84     delay(1000);
85
86     pinMode(motor1Pin1, OUTPUT);
87     pinMode(motor1Pin2, OUTPUT);
88     pinMode(motor2Pin1, OUTPUT);
```

```
89     pinMode(motor2Pin2, OUTPUT);
90 }
91
92
93 void loop()
94 {
95     RemoteXY_Handler ();
96
97     if (RemoteXY.button_04==1) position(18,11); //Haut-droite
98     else if (RemoteXY.button_01==1) position(-18,11); //Haut-gauche
99     else if (RemoteXY.button_05==1) position(18,-11); //Bas-droite
100    else if (RemoteXY.button_02==1) position(-18,-11); //Bas-gauche
101    else if (RemoteXY.button_03==1) position(0,0); //Centre
102
103    else if (RemoteXY.button_08==1){ c=c+10;position(c,i);}
104    else if (RemoteXY.button_07==1){ c=c-10;position(c,i);}
105
106    else if (RemoteXY.button_09==1){ i=i-10;position(c,i);}
107    else if (RemoteXY.button_06==1){ i=i+10;position(c,i);}
108
109    if (RemoteXY.switch_01 == 1) {
110
111        digitalWrite(motor1Pin1, LOW);
112        digitalWrite(motor1Pin2, HIGH);
113        digitalWrite(motor2Pin1, HIGH);
114        digitalWrite(motor2Pin2, LOW);
115    }
116    else if (RemoteXY.switch_01 == 0){
117
```

Annexe

❑ Code de l'application:

```
118     digitalWrite(motor1Pin1, HIGH);
119     digitalWrite(motor1Pin2, HIGH);
120     digitalWrite(motor2Pin1, HIGH);
121     digitalWrite(motor2Pin2, HIGH);
122 }
123 }
124
125
126 void position(int y,int z)
127 {
128     servol.write(angle + y);
129     delay(100);
130     c=y;
131
132     servo2.write(angle + z);
133     delay(100);
134     i=z;
135 }
```