



TRAVAUX D'INITIATIVE  
PERSONNELLE ENCADRÉS T.I.P.E.

2023

**La ville**



**Sujet :**

**Robot tondeuse intelligent**

**préparé par :**

**Hajar EL HAOUTI**

**I****Introduction**

- *Présentation du système*
- *Problématique et objectifs*

**II****Présentation fonctionnelle du système**

- *Diagramme Sys ML*

**III****Analyse des solutions techniques**

- *Choix des moteur de déplacement*
- *modélisation de la chaîne d'énergie*
- *Asservissement et régulation de vitesse*
- *Mesure de la vitesse de déplacement*
- *Inspection du robot*

**IV****Résultats globale****V****Conclusion**

Les villes de demain doivent avoir des espaces verts propres, esthétiques et bien entretenus. Les robots tondeuses peuvent être une solution pour maintenir ces espaces en bon état sans effort pour les jardiniers..



## **Robot Tondeuse**

Les robots tondeuses sont des tondeuses à gazon autonomes qui utilisent des capteurs et des algorithmes pour tondre l'herbe efficacement et précisément. Ils sont équipés de lames rotatives pour couper l'herbe tout en offrant une tonte régulière et uniforme.

## Robot Tondeuse

### Les avantages :

1. Tonte uniforme
2. Respect de l'environnement
3. Sécurité
4. Économies de coûts
5. automatisation



## **Problématique**

comment assurer une tonte de gazon efficace dans les espaces verts en milieu urbain tout en garantissant la sécurité des biens et des personnes ainsi qu'une bonne communication ? C'est une question cruciale étant donné l'importance des espaces verts en ville



## Objectifs

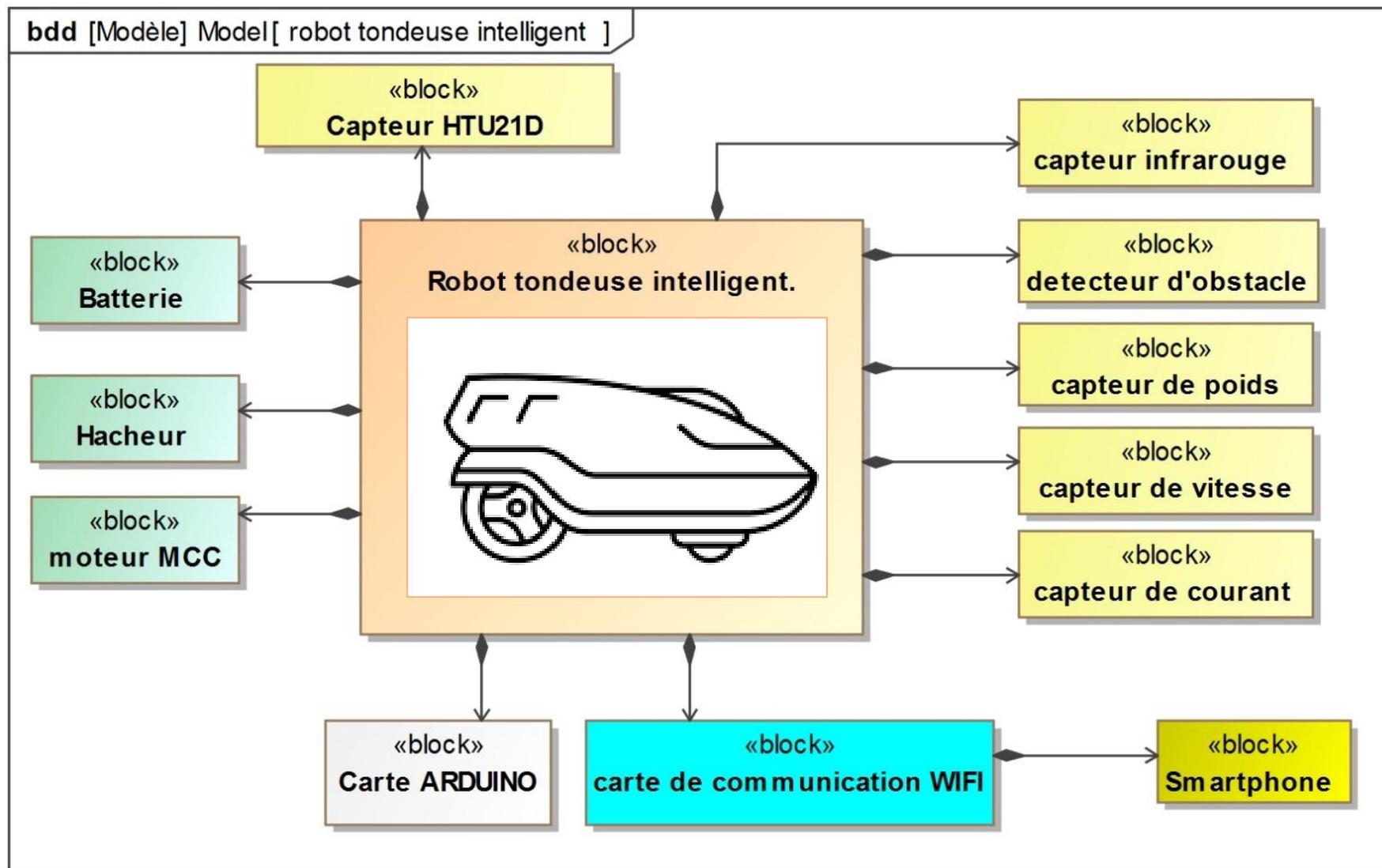
- ❑ *Dimensionnement de la chaîne d'alimentation du robot : Batterie.*
- ❑ *Dimensionnement de la chaîne de conversion électromécanique du robot.*
- ❑ *Etude de régulation et asservissement de vitesse de déplacement du robot.*
- ❑ *Etude et choix de la chaîne d'acquisition : les capteurs et détecteur.*
- ❑ *Conception d'une application mobile pour la communication avec le robot tondeuse.*





## **2 Diagramme Sys ML : exigences Req**

### 3 Diagramme Sys ML : BDD



## 1- Choix de la batterie d'alimentation

### bilan de courant des équipement de robot

Deux moteur MCC (24V)	1000 mA
Capteur de température (5V)	0.15 mA
Détecteur d'obstacle (5V)	15 mA
Capteur de infrarouge (5V)	20 mA
Carte wifi ESP8266 (3.3V)	70 mA
Capteur de courant (5V)	1 mA
Capteur de vitesse (5V)	30 mA.
Capteur de poids (5V)	1,5 mA
<b>Le courant totale</b>	<b>1.138 A</b>

### Capacité théorique demandé

Le robot fonctionne en plein régime pendant 2 heures (autonomie).

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

- $I$ : le courant totale consommé par le robot
- $\Delta t$ : l'autonomie du robot

Donc :  $C = 2.276 \text{ Ah}$

Coefficient de sécurité : 25%



**$C_n = 3 \text{ Ah}$**

### la batterie choisie

***SOLiSE B12003L : Batterie lithium 12V 3,8Ah***

<https://www.masolise.com/industriel/35-b12003l.html>

### Deux batterie (en série) :

Capacité	3.8 Ah
Tension de charge:	14,4V
Tension mini de décharge:	10V
Poids:	380g



## 2 – Détection automatique d'obstacle

### contexte

Le robot s'arrête automatiquement lorsqu'il détecte un obstacle à une distance de 0.5 mètres, et ne reprend son mouvement que lorsque l'obstacle a disparu.

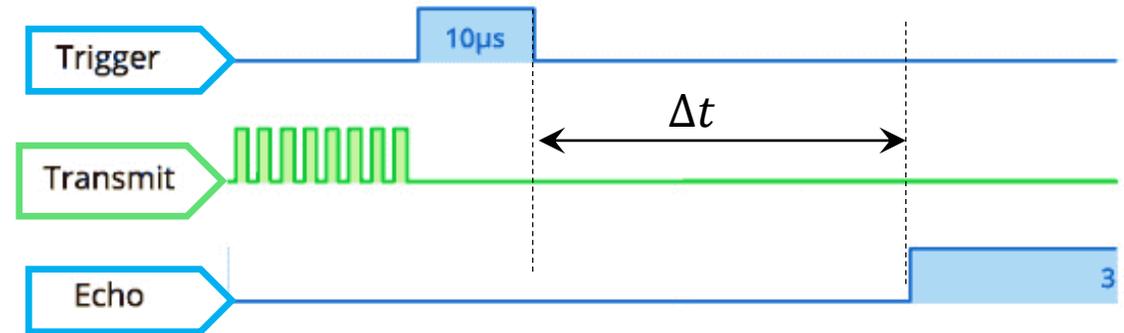
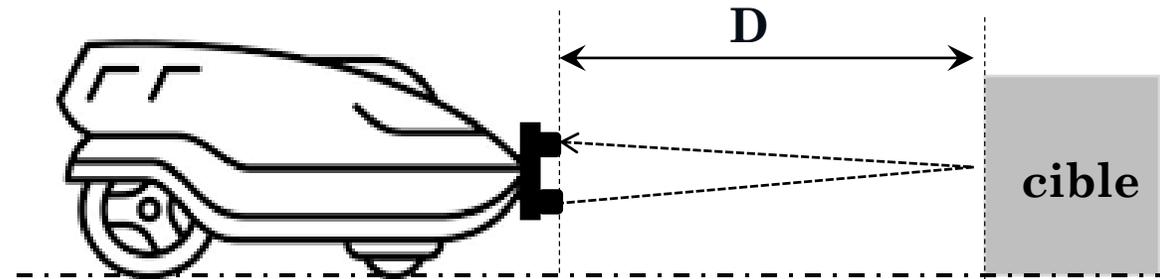
### Capteur utilisée : HC-SR04

HC-SR04 est un capteur à ultrasons qui peut mesurer la distance en utilisant la réflexion d'ondes sonores.



- Alimentation : 5V
- Plage de mesure : 2 cm à 4m

### mesure de la distance



La relation pour calculer la distance  $D$  :

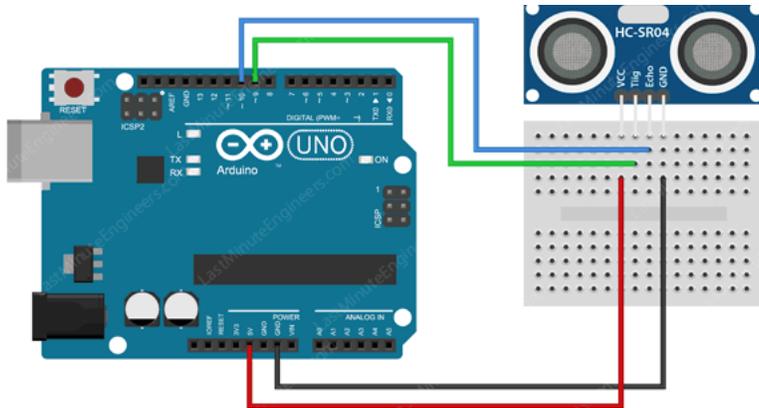
$$D = \Delta t \cdot \frac{v}{2}$$

- $v$  : la vitesse d'onde sonore (340 m/s)
- $\Delta t$  : le temps de propagation

## 2 – Détection automatique d'obstacle

Expérience : mesure de la distance

### ❖ Schéma de principe



On installe une bibliothèque [NewPing](#) pour gérer le capteur HC-SR04: `#include "NewPing.h"`

### ❑ Configuration

`sonar(x, y, z)`

Configuration

`sonar.ping_cm()`

Lire la distance D

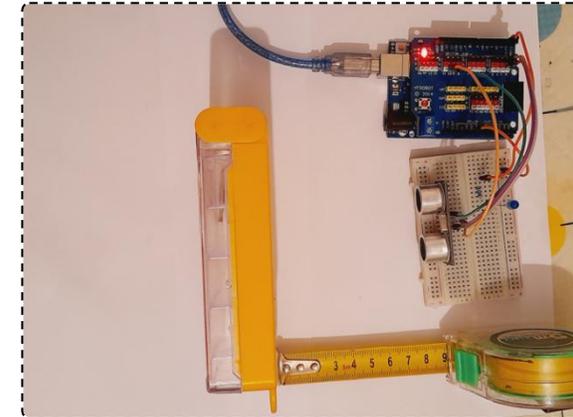
### ❖ Schéma réel et résultat



Exemple : pour le seuil est 10 cm

```
COM3
Distance = 14 cm
```

Le robot en service



```
COM3
Distance = 7 cm
```

Le robot s'arrête

## 2 – Détection automatique d'obstacle

Expérience : mesure de la distance

### ❖ Programme C.

#### ❑ Configuration

```
*main.ino x
1
2 #include "NewPing.h"
3
4 #define TRIGGER_PIN 9
5 #define ECHO_PIN 10
6 #define MAX_DISTANCE 400
7 #define led 4
8
9 NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
10
11 void setup() {
12     Serial.begin(9600);
13     pinMode(led, OUTPUT);
14 }
15
```

- Appel la bibliothèque de gestion de capteur
- Déclaration des variables
- Initialisation de capteur
- Définir la distance maximale à mesurer
- Définir la vitesse de communication.

#### ❑ programme

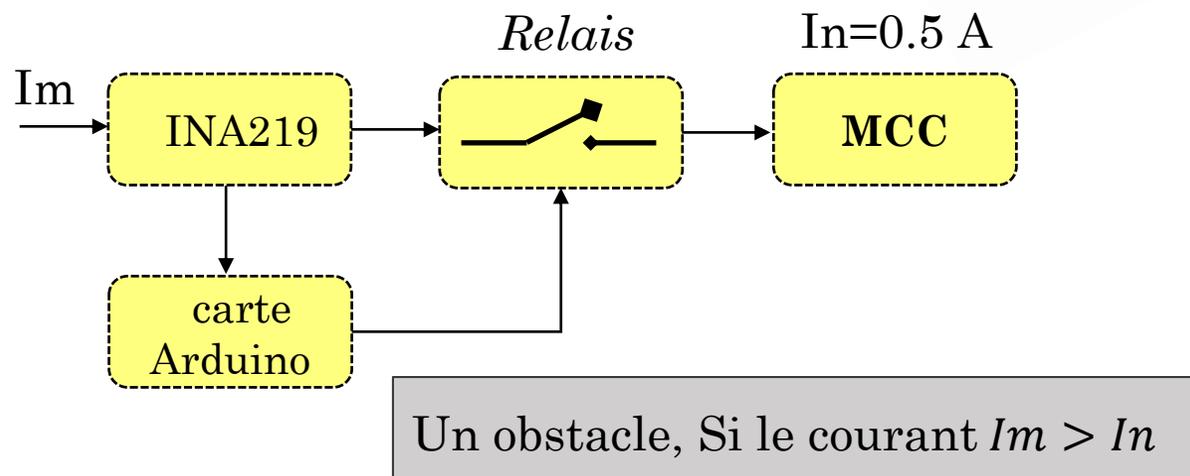
```
*main.ino x
16
17 void loop() {
18
19     float D=sonar.ping_cm();
20
21     Serial.print("Distance = ");
22     Serial.print(D);
23     Serial.println(" cm");
24
25     if (D>= 50) digitalRead(led,HIGH);
26     else digitalRead(led,LOW);
27     delay(500);
28 }
29
```

- mesure et lecture de la distance
- Affichage de la distance
- Une LED s'allume lorsque la distance  $D > 50$  cm

### 3 – Limiteur de couple du moteur de coupe d'herbe

#### Contexte

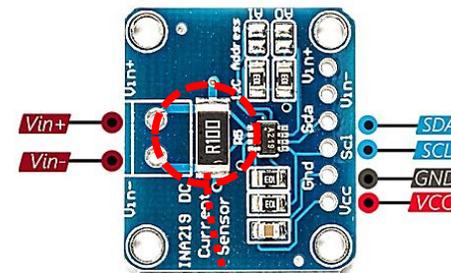
Pour protéger le moteur de la tonte, un capteur de courant mesure la quantité de courant absorbé par celui-ci en cas d'obstacle tel que des rochers ou des fils, qui peuvent bloquer le moteur



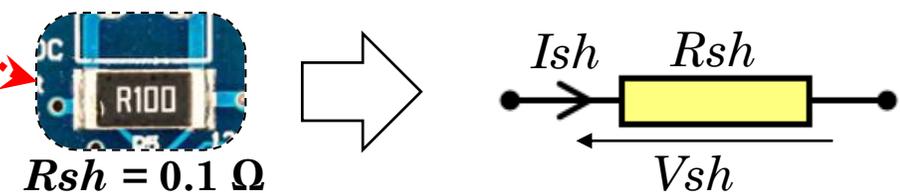
#### Choix de capteur

Le moteur de tonte : 10 W, 0.5 A, 1500 tr/min

❑ Le choix : INA219 (cap. Numérique I2C)



Alimentation	5 V
Tension de shunt DC	26 V
Courant de shunt DC	3.6 A
Résistance de shunt $R_{sh}$	0.1 $\Omega$



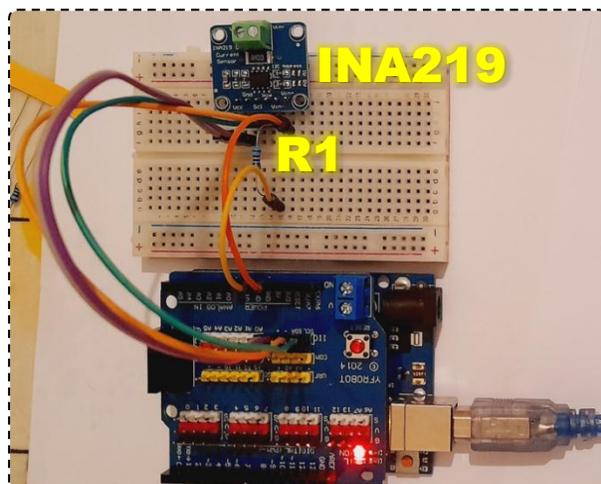
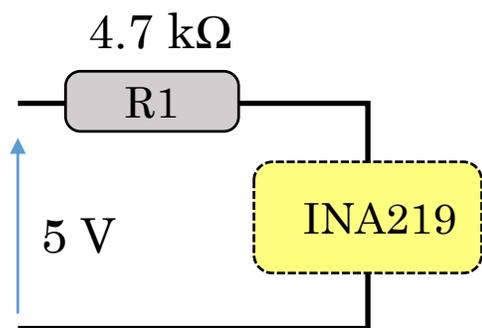
■ Bibliothèque de gestion : [Adafruit\\_INA219](#)

■ Mesures effectuées par ce capteur :

- Le courant
- La tension
- La puissance

## 2 - Limiteur de couple du moteur de coupe d'herbe

### Expérience 1 : validé le capteur



### ❑ Fonctions de mesures

```
33 shuntvoltage = ina219.getShuntVoltage_mV();
34 busvoltage = ina219.getBusVoltage_V();
35 current_mA = ina219.getCurrent_mA();
36 power_mW = ina219.getPower_mW();
37
```

$I = 1.06 \text{ mA}$

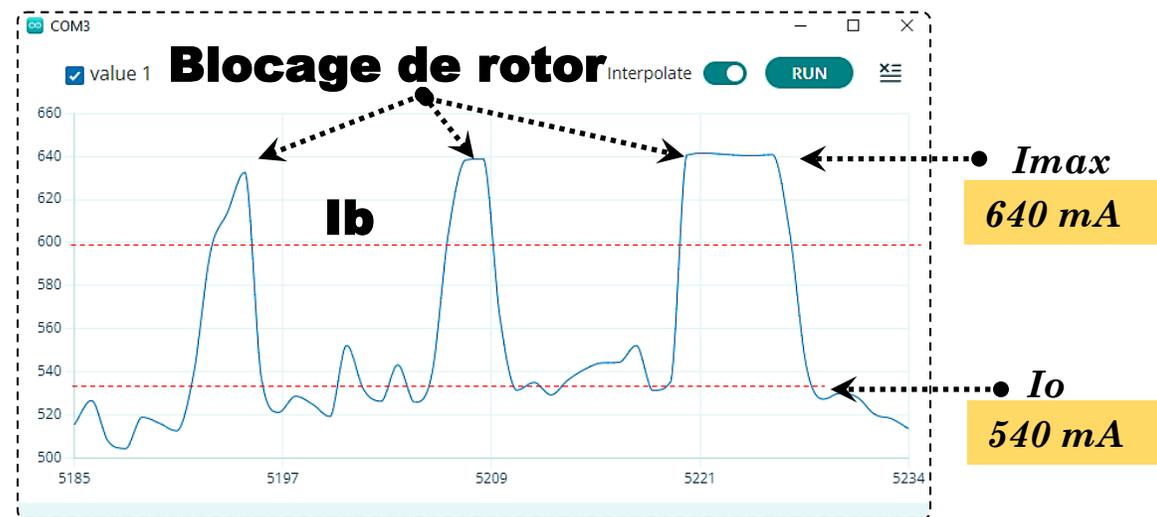
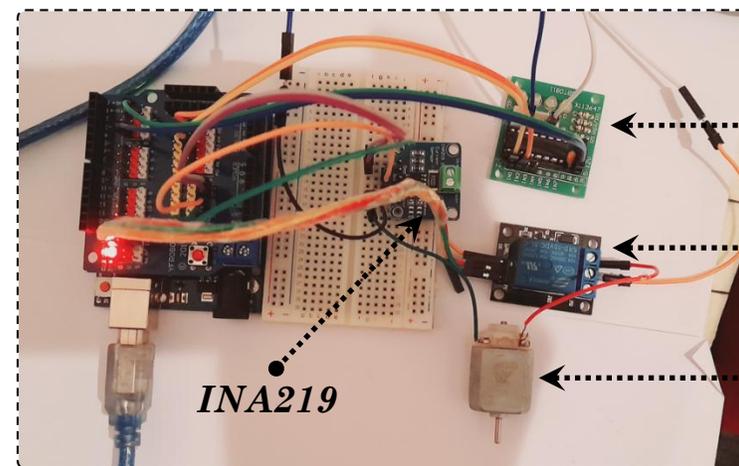
### ❑ Résultats

Sortie Moniteur série x

Message (Enter to send message to 'Arduino

```
tension de branche 4.96 V
tension de shunt: 0.08 mV
courant : 1.10 mA
puissance: 6.00 mW
```

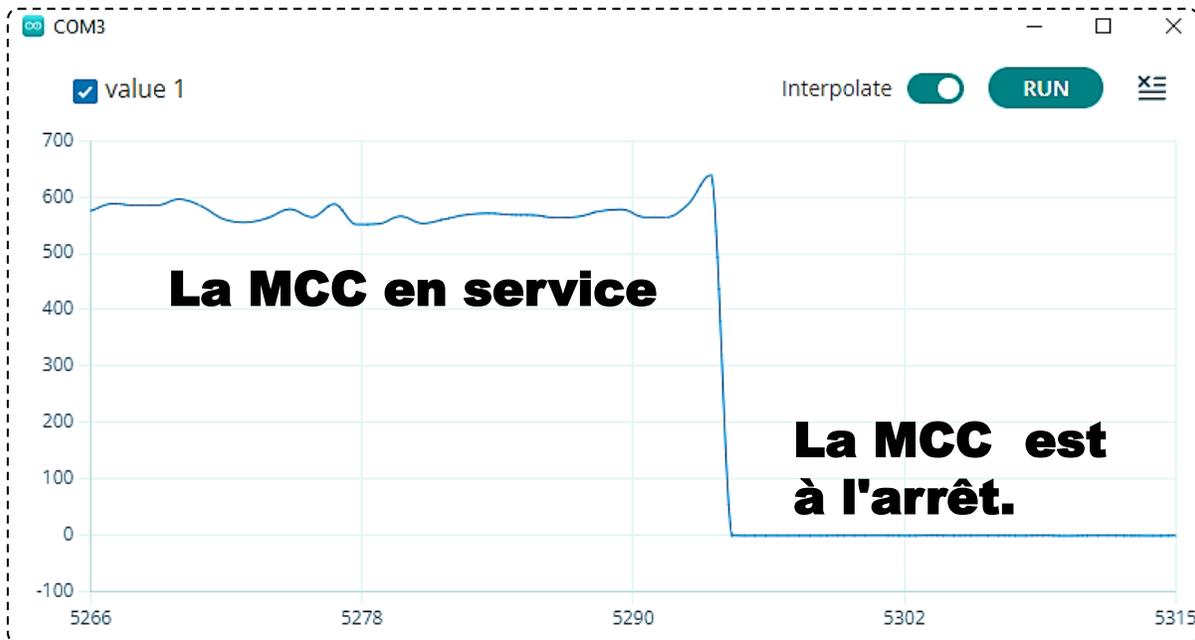
### Expérience N2: extraire le courant de blocage $I_b$



## 2 – Limiteur de couple du moteur de coupe d'herbe

### ❖ L'arrêt de la machine MCC

Si le courant de blocage  $I_b$  est dépassé, le relais est activé et coupe l'alimentation de la machine.



### ❖ Programme C

#### ❑ Configuration

```
*main.ino
1 #include <Wire.h>
2 #include <Adafruit_INA219.h>
3 #define relais 2
4 Adafruit_INA219 ina219;
5
6 void setup(void)
7 {
8     Serial.begin(115200);
9     if (! ina219.begin()) {while (1) { delay(10); }}
10    pinMode(relais,OUTPUT);
11    digitalWrite(relais,LOW);
12 }
13
```

- Appel la bibliothèque de gestion de capteur
- Déclaration des variables
- Initialisation de capteur
- La commande de relais
- Définir la vitesse de communication.

## 2 – Limiteur de couple du moteur de coupe d'herbe

### ❖ Programme C

#### □ programme

```
*main.ino ✖  
14  
15 void loop(void)  
16 {  
17   float Im_mA = 0;  
18   Im_mA = ina219.getCurrent_mA();  
19   Serial.println(Im_mA);  
20  
21   if(Im_mA > 600)   digitalWrite(relais,HIGH);  
22   delay(200);  
23 }  
24
```



## 2 – pesage d'herbe coupé

### Contexte

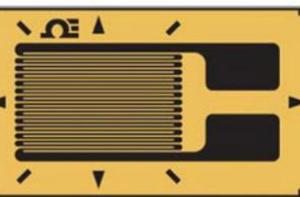
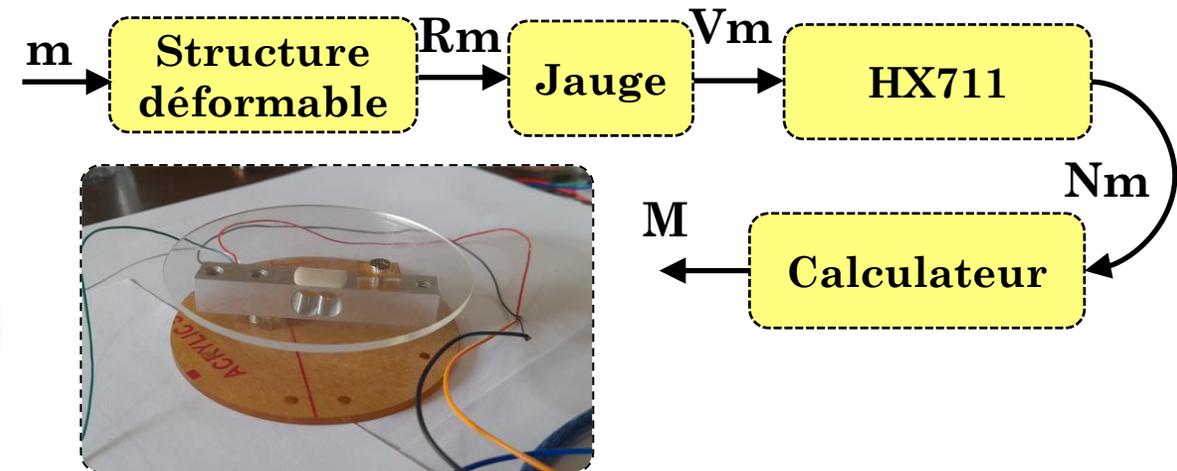
Pour garantir une tonte d'herbe efficace et sûre, il est crucial de contrôler la charge maximale supportée par la tondeuse (5 kg). Le poids d'herbe coupée est souvent utilisé pour déterminer cette limite, car une quantité excessive d'herbe dans le bac de la tondeuse peut déséquilibrer la machine et la rendre instable.



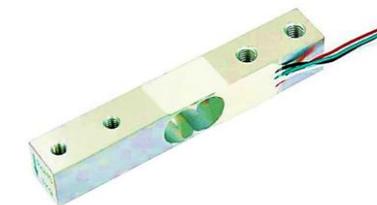
Capteur de poids à jauge

décision

### le capteur à jauge & HX711

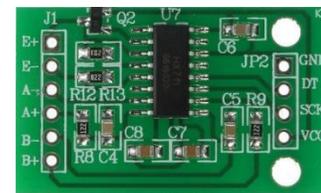


4 jauges



structure déformable

10 Kg



HX711

- AOP
- CAN 12 bits



## 2 – pesage d'herbe coupé

### Programme en C++

#### ❑ Configuration

```
*main.ino x
1  #include "HX711.h"
2  #define DEBUG_HX711
3  #define calibration 20780.0 // le poide M=0 kg
4
5  byte pinData = 2;
6  byte pinClk = 3;
7  HX711 poids;
8
9  void setup() {
10
11  #ifdef DEBUG_HX711
12      Serial.begin(9600);
13      Serial.println("le poids ");
14  #endif
15  poids.begin(pinData, pinClk);
16  poids.set_scale(calibration);
17  poids.tare();
18 }
```

#### ❑ programme

```
*main.ino x
19
20 void loop() {
21     #ifdef DEBUG_HX711
22         Serial.print("Le poids : ");
23         float N=poids.get_units();
24         float M=-0.0763636*N; //étalonnage (0.336 ---> -4.4)
25         Serial.print(M);
26         Serial.print(" Kg");
27         Serial.println();
28     #endif
29 }
30
```

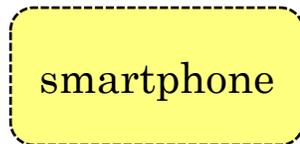
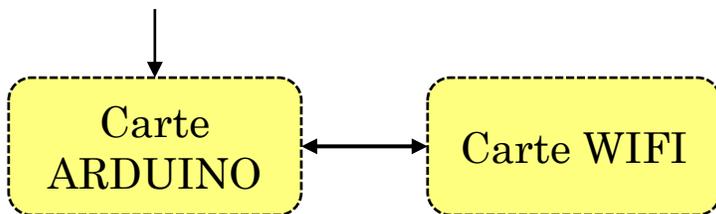
## 4 – Contrôle à distance de robot tendeuse

### Contexte

Le robot est doté d'une unité de communication qui permet de le contrôler à distance via une connexion Bluetooth avec un smartphone :

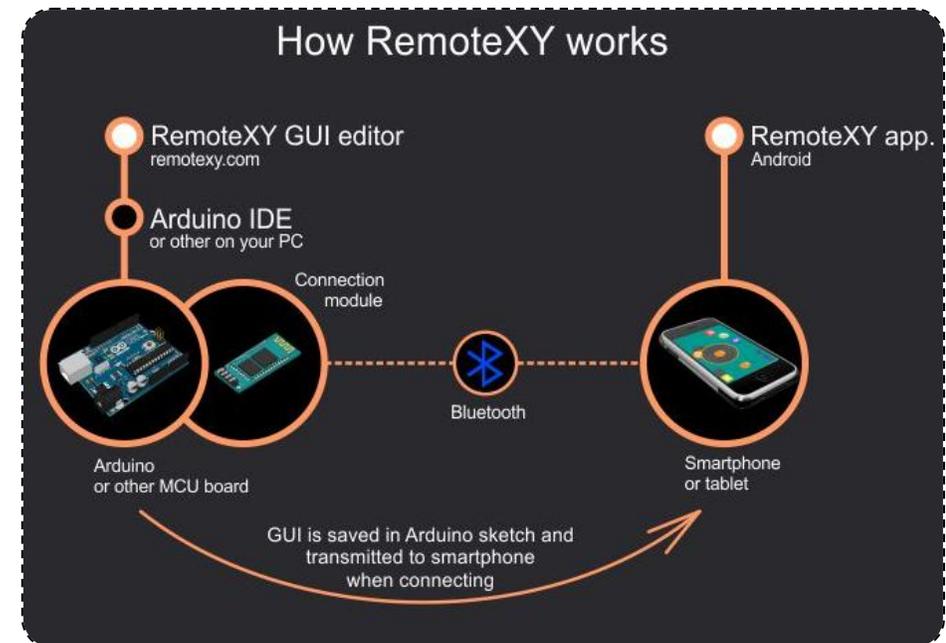
- Contrôle de déplacement
- Affichage des mesures
- Affichage des erreurs

### Les mesure



### plate forme de conception de l'application

**Remotexy** est un outil simple pour concevoir et utiliser une interface utilisateur graphique mobile qui permet de contrôler des cartes de contrôleur à partir d'un smartphone ou d'une tablette.



## 4 – Contrôle à distance de robot tendeuse

plate forme de conception de l'application

Remotexy est un système complet qui comprend:

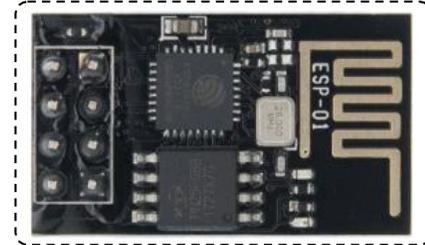
- un éditeur d'interfaces graphiques mobiles pour les cartes contrôleurs disponibles sur le site remotexy.com.
- une application mobile qui permet de se connecter au contrôleur et de le contrôler via une interface graphique.



Les moyennes nécessaires

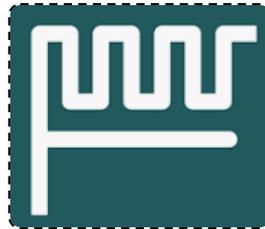
### ■ Module WIFI

- Alimentation: 3,6 à 6 Vcc
- Liaison série: 4800 b/s



### ■ Application mobile

Application est disponible dans Apple store et Google Play



### ■ Bibliothèque de gestion

Bibliothèque de communication série et de Remotexy sont disponible dans le site web remotexy.com:

```
#include <RemoteXY.h>
```

## 4 – Contrôle à distance de robot tondeuse

plate forme RemoteXY

Elements

▼ Controls

- Button
- Push switch
- Switch
- Select
- Slider
- Joystick
- RGB color
- Edit field

▼ Sensors

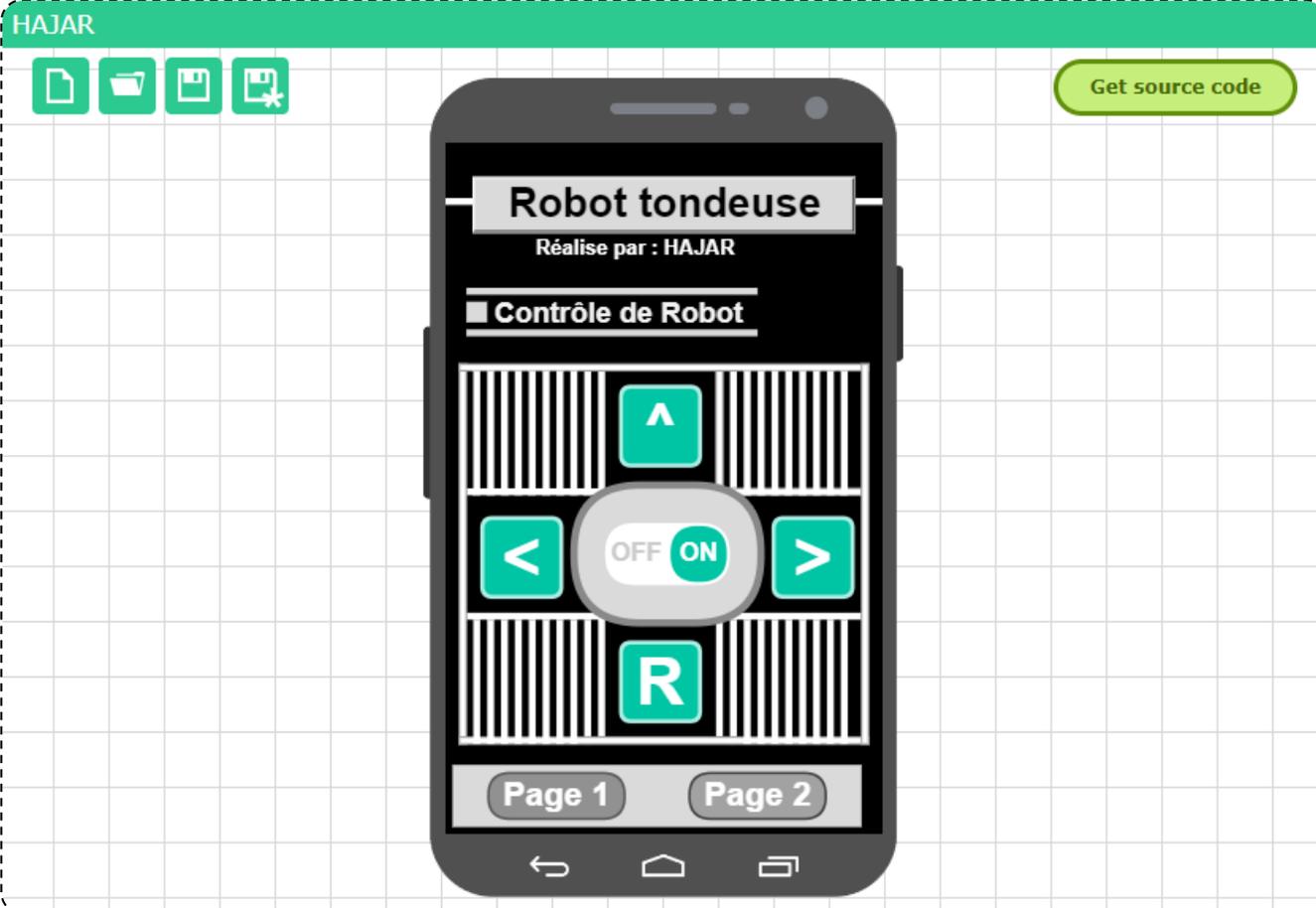
▼ Indication

- Led
- RGB led
- Linear level

Outils

HAJAR

Get source code



Éditeur de l'application

▼ Configuration

- Wi-Fi access point
- Arduino UNO
- ESP8266 Wi-Fi module
- Arduino IDE

▼ Module interface

Connection interface:  
Software Serial

RX pin: 2 TX pin: 3

Speed (baud rate):  
9600

configuration

## 4 – Contrôle à distance de robot tendeuse

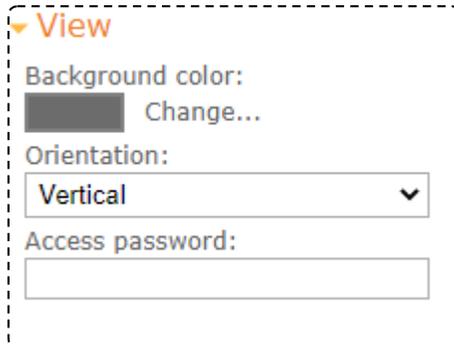
### Configuration logicielle et matérielles

#### ■ Choix matériel

- Transmission WIFI
- Carte Arduino UNO
- Carte wifi ESP8266
- Editeur de programmation  
Arduino



#### ■ Orientation de l'application



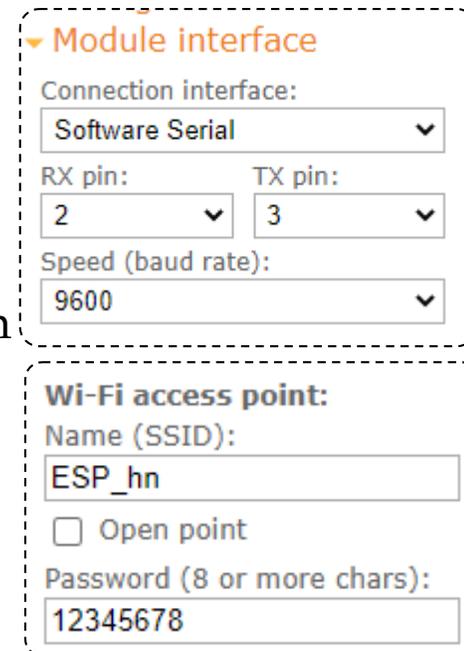
Verticale



Horizontale

#### ■ Connexion et la vitesse de communication

- La transmission série programmé « software »
- La broche de communication avec la carte ARDUINO
  - ❖ Broche 2 : RX en réception
  - ❖ Broche 3 : TX en émission
- La vitesse de communication (ou débit) : la vitesse de communication est réglée à 19200 bits/s



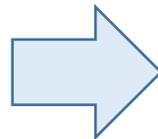
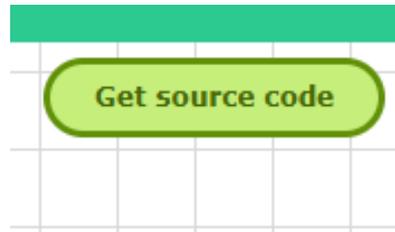
## 4 – Contrôle à distance de robot tondeuse

### Objectifs de l'application



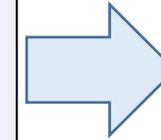
## 4 – Contrôle à distance de robot tendeuse

### Construction de programme



```
ct connection mode and include library
MODE_ESP8266_SOFTSERIAL_CLOUD
<SoftwareSerial.h>
#include <RemoteXY.h>
// RemoteXY connection settings
#define REMOTEXY_SERIAL_RX 2
#define REMOTEXY_SERIAL_TX 3
#define REMOTEXY_SERIAL_SPEED 19200
#define REMOTEXY_WIFI_SSID "ZTE-wifi"
#define REMOTEXY_WIFI_PASSWORD "123456789"
#define REMOTEXY_CLOUD_SERVER "cloud.remotexy.com"
#define REMOTEXY_CLOUD_PORT 6376
#define REMOTEXY_CLOUD_TOKEN "83b5a069b8b023300cc1c7cd1365fbf9"

// RemoteXY configurate
#pragma pack(push, 1)
uint8_t RemoteXY_CONF[] = // 1295 bytes
{ 255,10,0,72,0,8,5,16,31,5,130,3,1,68,60,11,6,123,130,3,
27,50,3,18,6,0,130,3,5,50,3,18,6,0,130,3,5,31,25,21,
6,29,130,2,48,26,13,18,2,16,72,14,5,27,15,15,2,92,27,255,
18,0,0,0,0,0,200,66,0,0,0,0,131,3,3,84,14,4,1,
220,31,115,97,108,111,110,32,0,130,3,252,2,67,7
```

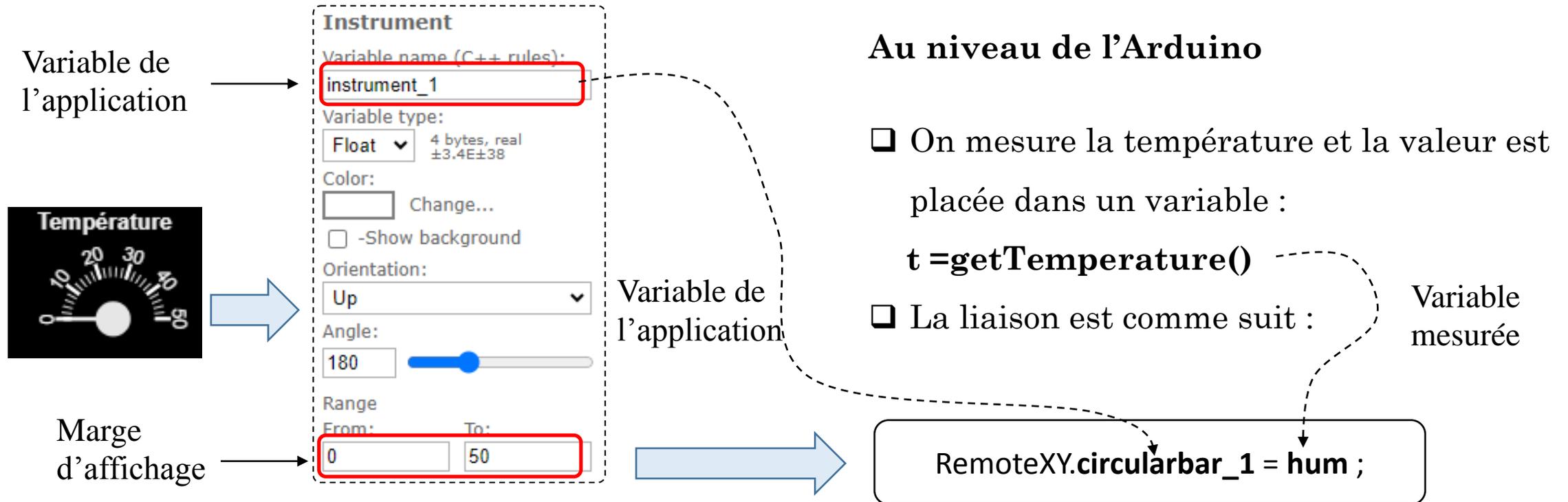


2. Install **RemoteXY library** for Arduino IDE.

Programme incomplet

## 4 – Contrôle à distance de robot tendeuse

liaison entre les variables de l'Arduino et les variable de l'application

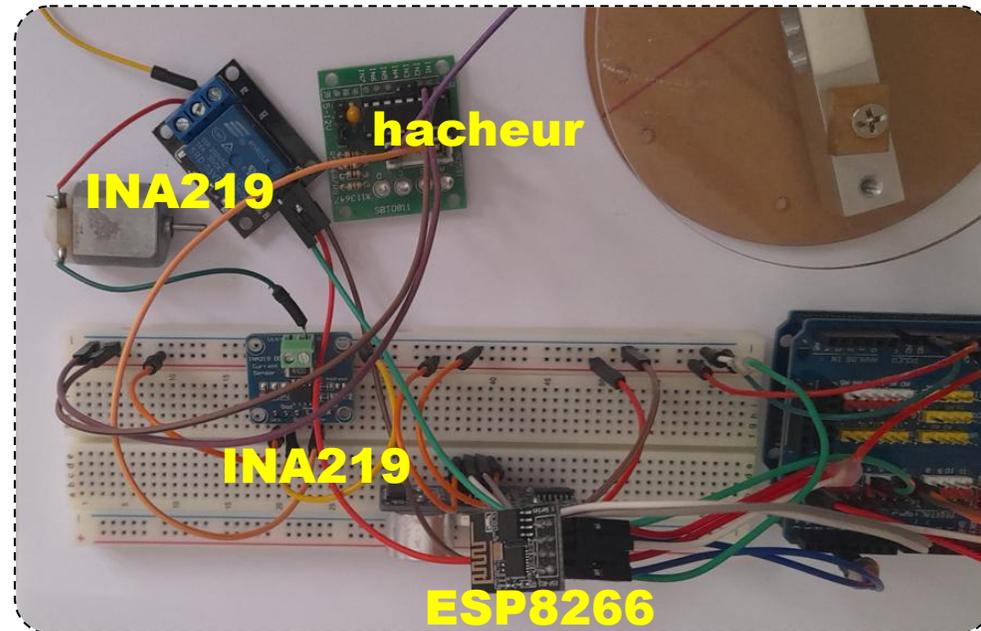
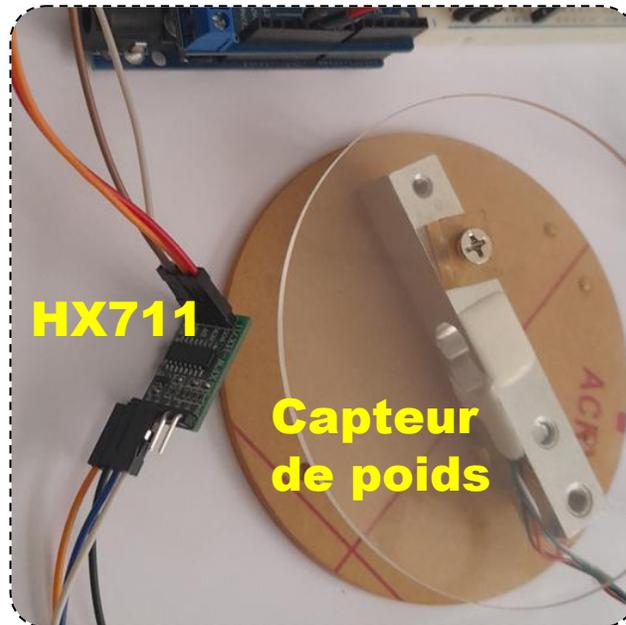
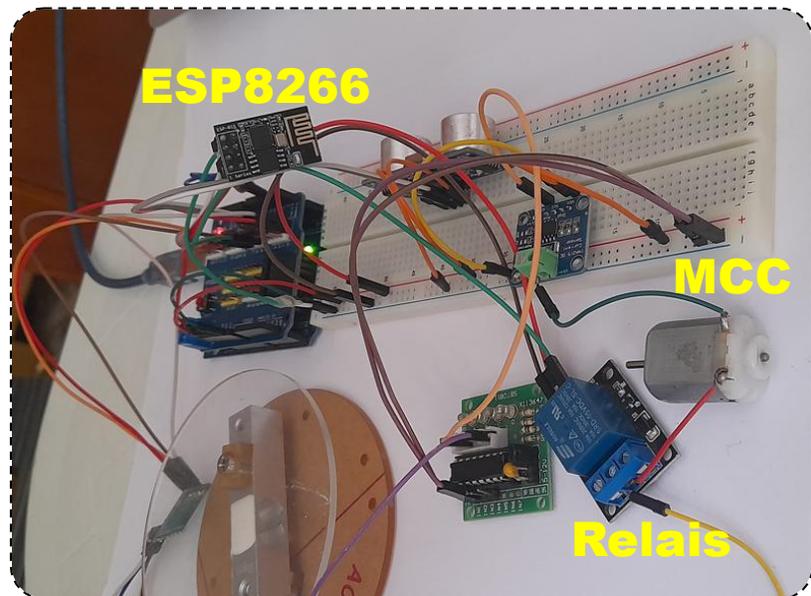


Plus d'information pour cette opération :

<https://remotexy.com/en/help/>

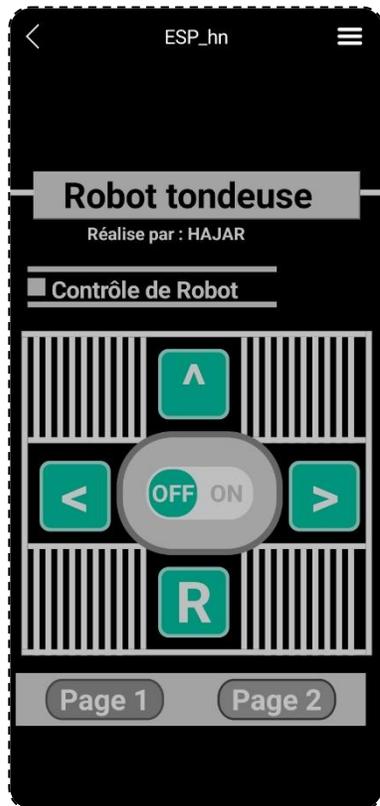
**4 – Contrôle à distance de robot tendeuse**

Résultats :



## 4 – Contrôle à distance de robot tondeuse

### Résultats :



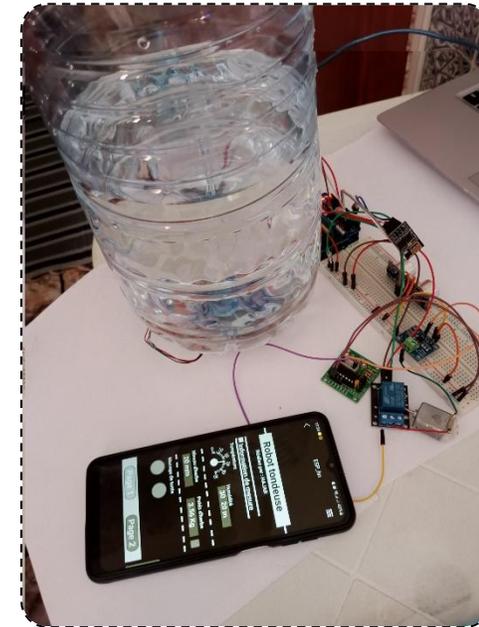
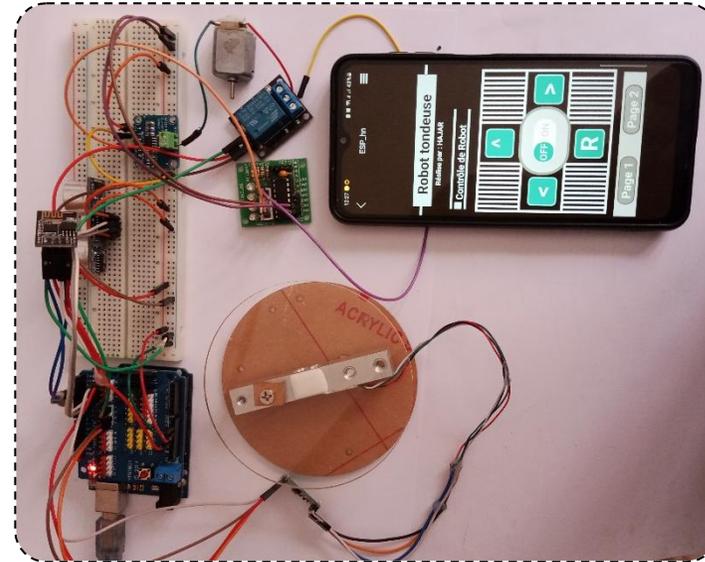
Interface de commande



Blocage de tonte



Masse d'herbe dépassée



Au cours de notre travail sur le TIPE "Robot tondeuse intelligent", nous avons fait preuve d'une grande ambition et d'un esprit d'équipe solide avec mon binôme. J'ai réalisé de ma part plusieurs points importants, tels que l'étude du bilan de courant et le choix de la batterie, l'étude des capteurs (détecteur d'obstacles, limiteur de couple et mesure du poids de l'herbe) et finalement, la conception d'une application de contrôle à distance du robot. Notre collaboration nous a permis d'atteindre ces objectifs de manière efficace et satisfaisante.



**MERCI POUR VOTRE  
ATTENTION**

- [https://www.bosch-diy.com/fr/fr/all-about-diy/tondeuses-robots\\_avantages-inconv%C3%A9nients-co%C3%BBts](https://www.bosch-diy.com/fr/fr/all-about-diy/tondeuses-robots_avantages-inconv%C3%A9nients-co%C3%BBts)
- <https://www.monenergie.be/blog/combien-denergie-une-tondeuse-a-gazon-robotisee-consomme-t-elle/>
- <https://www.bestofrobots.fr/blog/robot-tondeuse/>
- <https://www.lesechos.fr/pme-regions/actualite-des-marches-publics/les-nouvelles-technologies-investissent-les-espaces-verts-149159>
- <https://www.st.com/resource/en/datasheet/vl53l0x.pdf>
- <https://github.com/pololu/vl53l0x-arduino>
- <https://lastminuteengineers.com/electronics/arduino-projects/>
- <https://remotexy.com/en/help/indications/text/>
- <https://www.te.com/usa-en/product-CAT-HSC0004.html>