

Systèmes intelligents

CAHIER DES CHARGES

Réalisé par : GOIRE Antoine et DUCHENE Roy

Professeur : Mme Slama et Mme Liberatore

Année d’étude : Master 1-finalité automatisation

Année académique 2018-2019.

1. **Description du projet**

**ARDUINO**

Un robot tondeuse, qui circule aléatoirement dans un périmètre définit par une ligne d’une certaine couleur (exemple : le rouge). Cette ligne est détectée par le robot grâce à un détecteur de couleur RGB situé à l’avant de celui-ci. Il y a également un capteur à ultrasons qui détecte si un objet ou autre obstrue le chemin. Un fois qu’il s’en approche de X cm, il fait demi-tour et prend une autre direction. Si cet objectif est atteint c’est-à-dire en bonus, il est prévu d’ajouter une sécurité sur le robot tondeuse c’est-à-dire un capteur qui sera installé vers le sol. Quand une personne soulèvera la tondeuse cela coupera le moteur des lames ainsi que les moteurs des roues et un bip s’enclenchera également.

**RASPBERRY**

Détection de la matière au sol grâce à la caméra qui sera câblé sur la raspberry. Si c’est du gazon on actionne le moteur de la coupe en revanche si ce n’est pas de l’herbe alors le moteur se coupe et un bip s’actionne. On affichera sur l’écran LCD si c’est du gazon ou pas. En bonus, nous ajouterons sur l’écran LCD la couleur du gazon, sa qualité ainsi que si le gazon est bien tondu ou non.

**SCHEMA**

**LEGENDE**

Robot tondeuse

Arbre ou obstacle

Route qui traverse l’herbe (état de surface différente de celle de l’herbe).

Contour dans lequel le robot ne devra pas sortir.

Le robot déclenche un BIP et arrête le moteur de coupe.

**PREMIER PLAN DU ROBOT TONDEUSE**



1. **Description du matériel nécessaire**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre** | **Matériel** | **Code article** | **Lien** | **Prix** |
| 1 | Module à détection US HC-SR04A | 35750 | <https://www.gotronic.fr/art-module-a-detection-us-hc-sr04a-27740.htm> | 3,90 € |
| 1 | Capteur de couleur SEN0101 | 31752 | <https://www.gotronic.fr/art-capteur-de-couleur-sen0101-19374.htm> | 9,90 € |
| 2 | Kit roue+moteur 6VCC-100t/min | 33912 | <https://www.gotronic.fr/art-kit-roue-moteur-23569.htm> | 2x 5,50€ |
| 3 | Régulateurs ajustable 1,25à30VCC GT134 | 35189 | <https://www.gotronic.fr/art-regulateur-ajustable-1-25-a-30-vcc-gt134-26094.htm> | 3x2,50€ |
| 1 | Commande de 2 moteurs SBC-motordriver2 | 35683 | <https://www.gotronic.fr/art-commande-de-2-moteurs-sbc-motodriver2-27418.htm> | 6,90 € |
| 2 | Module relais 5 V GT1080 | 35226 | <https://www.gotronic.fr/art-module-relais-5-v-gt1080-26130.htm> | 2x3,80€ |
| 1 | Kit GPIO pour raspberry PI PI018 | 35118 | <https://www.gotronic.fr/art-kit-gpio-pour-raspberry-pi-pi018-25862.htm> | 10,90 € |
| 1 | moteur pour couper l'herbe | 25350 | <https://www.gotronic.fr/art-moteur-standard-rm1a-12005.htm> | 3,20€ |
| 1 | écran LCD | 34940 | <https://www.gotronic.fr/art-afficheur-lcd-i2c-2x16-caracteres-25650.htm> | 9,90€ |
| 1 | piezzo | 05496 | <https://www.gotronic.fr/art-capsule-piezoelectrique-dp035f-3856.htm> | 1,20€ |
| 1 | caméra raspberry | 35368 | <https://www.gotronic.fr/art-camera-8-mpx-pour-raspberry-rb-camera-v2-26774.htm> | 36,50€ |
| 1 | raspberry | 35790 | <https://www.gotronic.fr/art-carte-raspberry-pi3-b-27826.htm> | 43,95€ |
| 1 | arduino ATmega | 25951 | https://www.gotronic.fr/art-carte-arduino-mega-2560-12421.htm | 38.90€ |
| 1 | roulette (avant) | / | / | Pack |
| 2 | support de pile | / | / | Pack |
| X | piles | / | / | Pack |

[](https://www.gotronic.fr/ori-module-a-detection-us-hc-sr04a-27740.jpg) Module à détection US HC-SR04A

[](https://www.gotronic.fr/ori-capteur-de-couleur-sen0101-19374.jpg) Capteur de couleur SEN0101

[](https://www.gotronic.fr/ori-kit-roue-moteur-23569.jpg)Kit roue+moteur 6VCC-100t/min

[](https://www.gotronic.fr/ori-regulateur-ajustable-1-25-a-30-vcc-gt134-26094.jpg)Régulateurs ajustables 1,25à30VCC GT134

[](https://www.gotronic.fr/ori-commande-de-2-moteurs-sbc-motodriver2-27418.jpg)Commande de 2 moteurs SBC-motordriver2

[](https://www.gotronic.fr/ori-module-relais-5-v-gt1080-26130.jpg) Module relais 5 V GT1080

[](https://www.gotronic.fr/ori-kit-gpio-pour-raspberry-pi-pi018-25862.jpg) Kit GPIO pour Raspberry PI PI018

 Écran LCD

 Arduino

 Raspberry

 Roulette avant du robot

Moteur pour couper l’herbe

1. **Diagramme du projet**

Réflexion et conception du robot

Recherche et documentation technique sur les composants

Choix des composants et commande

Montage du robot + réglages des capteurs +câblage électrique

Programmation RASPBERRY

Programmation Arduino

Test + débogage RASPBERRY

Test + débogage Arduino

1. **Répartition des tâches au sein du groupe**
2. **Time Line avec étapes de validations**

|  |  |  |  |
| --- | --- | --- | --- |
| 05/02 | 06/02 | 12/02 | 13/02 |
| Réflexion robot | Réflexion robot | Préparation de la présentation du robot | Préparation de la présentation du robot |
| Réflexion robot | Réflexion robot | Préparation de la présentation du robot  +cahier des charges | Préparation de la présentation du robot  +cahier des charges |
| 19/02 | 20/02 | 26/02 | 27/02 |
| Choix du matériels  +cahier des charges | Choix du matériels  + cahier des charges | Présentation du robot + passage de la commande | cahier des charges |
| Choix du matériels  +cahier des charges | Choix du matériels  +cahier des charges | Présentation du robot + passage de la commande | cahier des charges |
| 12/03 | 13/03 | 19/03 | 20/03 |
| Codage capteur de couleur | Codage pour actionner les roues | Mise en relation programme + test et débogage | Test et débogage |
| Codage du capteur de proximité | Codage pour actionner les roues | Mise en relation programme + test et débogage | Test et débogage |
| 27/03 | 03/04 | 24/04 | 08/05 |
| Raspberry écran LCD | Communication Arduino Raspberry | Mise en relation du programme + Test et débogage | Test et débogage |
| Raspberry reconnaissance gazon et sol | Raspberry reconnaissance gazon et sol | Mise en relation du programme + Test et débogage | Test et débogage |
| 15/05 |  |  |  |
| Améliorations  (Mise au propre du code + commentaires etc) |  |  |  |
| Améliorations  (Mise au propre du code + commentaires etc) |  |  |  |
| Vacances Carnaval 04/03 => 08/03 | | | |
| Montage robot | Câblage électrique  +réglages des capteurs | Temps pour s’il y a un retard à rattraper |  |
| / | / | Temps pour s’il y a un retard à rattraper |  |
| Vacances Pâques 08/04 => 19/04 | | | |
| Temps pour s’il y a un retard à rattraper |  |  |  |
| Temps pour s’il y a un retard à rattraper |  |  |  |

1. **Etat de l’art de l’existant**

**Code pour moteur driver Arduino**

Code utile pour le déplacement de notre robot tondeuse.

[](https://www.gotronic.fr/ori-commande-de-2-moteurs-sbc-motodriver2-27418.jpg)

int motorpin1 = 3;

int motorpin2 = 4;

int pinPMoteur = 5;                       

void setup () {

  pinMode(motorpin1,OUTPUT);

  pinMode(motorpin2,OUTPUT);

  pinMode(pinPMoteur,OUTPUT);             

}

void loop () {

  digitalWrite(motorpin1,LOW);

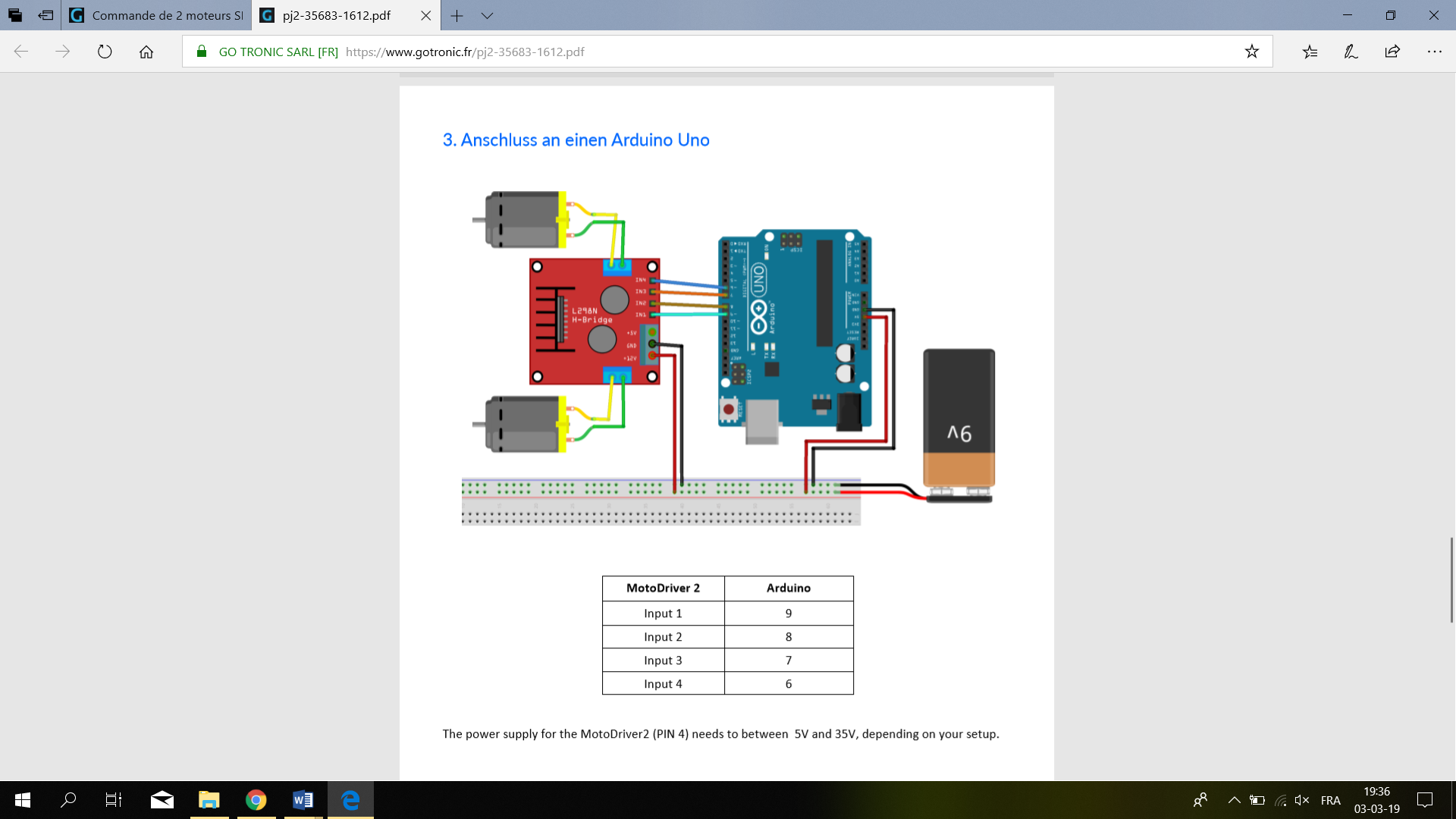
  digitalWrite(motorpin2,HIGH);

  analogWrite(pinPMoteur,255);

delay(5000);

  digitalWrite(motorpin1,HIGH);

  digitalWrite(motorpin2,LOW);

  analogWrite(pinPMoteur,150);

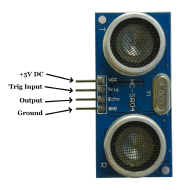
  delay(5000);

}

**Code utile pour le capteur a ultrason**

Permet au robot d’éviter les obstacles présents dans son environnement de travails.

Le fonctionnement du module est le suivant :

Il faut envoyer une impulsion niveau haut (à + 5v) pendant au moins 10 µs sur la broche ‘**Trig Input**’; cela déclenche la mesure. En retour la sortie ‘**Output**’ ou ‘**Echo**’, va fournir une impulsion + 5v dont la durée est proportionnelle à la distance si le module détecte un objet.

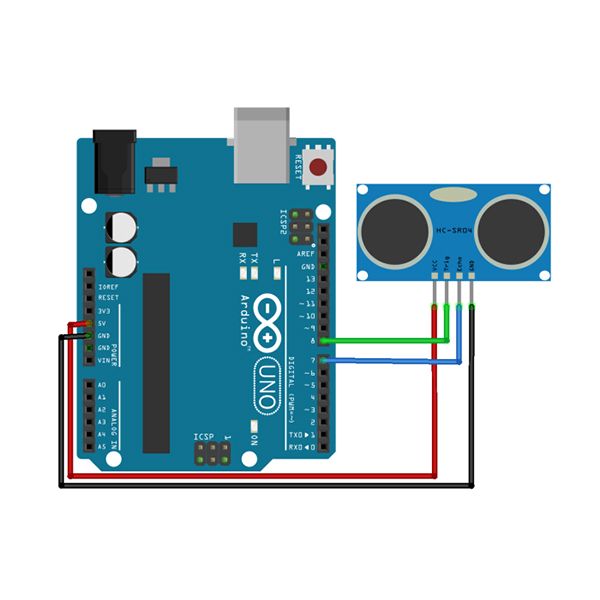
Afin de pouvoir calculer la distance en cm, on utilisera la formule suivante :

**distance = (durée de l’impulsion (en µs) / 58**

/\* Utilisation du capteur Ultrason HC-SR04 \*/

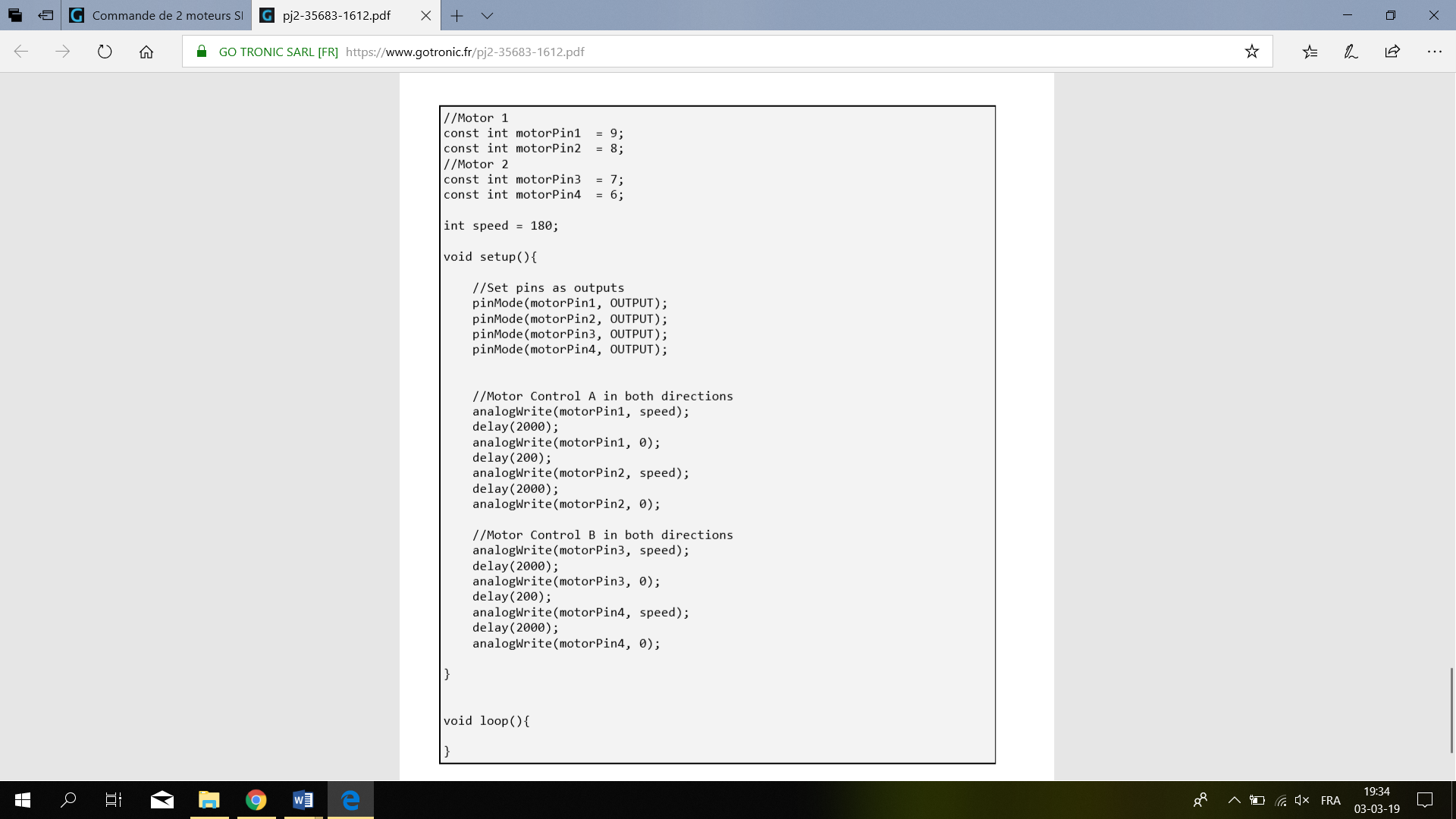
// définition des broches utilisées   
int trig = 8;   
int echo = 7;   
long lecture\_echo;   
long cm;

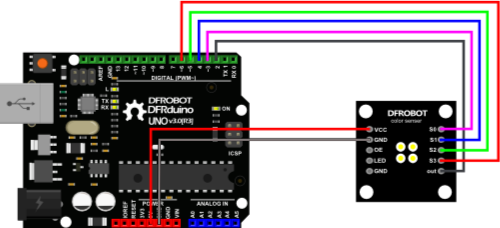
void setup()   
{   
  pinMode(trig, OUTPUT);   
  digitalWrite(trig, LOW);   
  pinMode(echo, INPUT);   
  Serial.begin(9600);   
}

void loop()   
{   
  digitalWrite(trig, HIGH);   
  delayMicroseconds(10);   
  digitalWrite(trig, LOW);   
  lecture\_echo = pulseIn(echo, HIGH);   
  cm = lecture\_echo / 58;   
  Serial.print("Distance en cm : ");   
  Serial.println(cm);   
  delay(1000);   
}

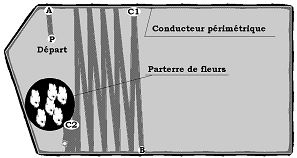
**Code utile pour le capteur RGB**

Permet de détecter les limites établies par une ligne de couleur.

[](https://www.gotronic.fr/ori-capteur-de-couleur-sen0101-19374.jpg)



**Stratégie de tonte**

[](https://wikimeca.org/index.php?title=Fichier:RL_500_strat%C3%A9gie_de_tonte.PNG)

* **T1** : Tâche d'orientation : l'opérateur ayant posé la tondeuse au sol en un point P quelconque de la parcelle, puis appuyé sur le bouton de démarrage, la tondeuse pivote autour d'un axe vertical dans le sens direct (noté Z+), et s'oriente vers le nord géographique à l'aide de la boussole électronique.
* **T2** : Tâche de recherche et de suivi de fil : la tondeuse avance ensuite vers le nord jusqu'à ce qu'elle rencontre le conducteur périmétrique (point A).  
  Elle pivote dans le sens direct, et suit le conducteur périmétrique afin de tondre le pourtour de la parcelle.  
  Lorsqu'elle a bouclé un tour et demi (point B), elle pivote vers l'intérieur de la parcelle et commence un cycle de tonte en "zig-zag".
* **T3** : Tâche de tonte en "zig-zag" : à chaque fois que la tondeuse rencontre le conducteur périmétrique (point C1) ou un obstacle (point C2), elle s'arrête, pivote autour d'un axe vertical d'environ 5 degrés, et repart en sens inverse.
* **T4** : Tâche de pivotement de 60 degrés: si un pivotement de 5 degrés lors de la tâche T3 l'amène à sortir du périmètre, la tondeuse pivote sur elle-même de 60 degrés et repart pour un nouveau cycle de tonte en "zig-zag".

**Raspberry**

<https://www.framboise314.fr/i-a-realisez-un-systeme-de-reconnaissance-dobjets-avec-raspberry-pi/>

<https://gogul09.github.io/software/texture-recognition>

<https://www.bluetin.io/opencv/opencv-color-detection-filtering-python/>

<https://www.pyimagesearch.com/2014/08/04/opencv-python-color-detection/>

<https://stackoverflow.com/questions/50178248/detect-color-in-the-middle-of-image-opencv-python>

1. **Etat d’avancement**
2. **PARTIE 1: Arduino**

**Diagramme Générale du déplacement du robot.**

**ACTION**

**CONDITION** **CONDITION**

Démarrage Robot via l’interrupteur

DEMMARAGE ARDUINO

Attente de X secondes

RGB = Vert ET capteur de proximité >X cm

INITIALISATION ARDUINO

RGB = Pas vert OU capteur de proximité <X cm

Action du robot

RGB = Pas vert OU capteur de proximité <X cm

Le robot n’avance pas

Le robot avance tout droit

Aucune condition

Robot se stop

Aucune condition

Robot attend X secondes

X secondes écoulé OK

RGB = Pas vert OU capteur de proximité <X cm

Robot recule

Robot reculé OK

Robot fait demi-tour

Robot demi-tour OK

Robot stop

Robot stop OK

Robot avance

**Mise en forme du programme Arduino**

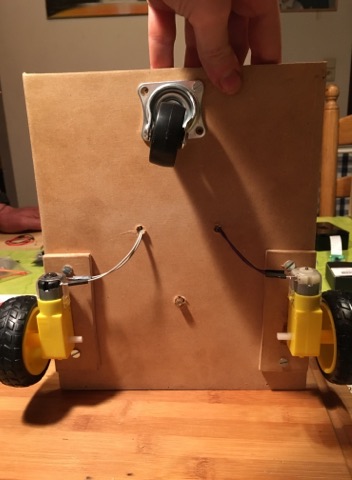
**DEFINE :** Définition des numéros de pins

**VOID SETUP :** Définition des entrées et sorties des pins

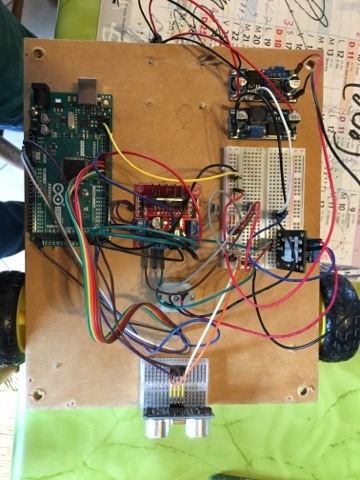
**VOID LOOP :**

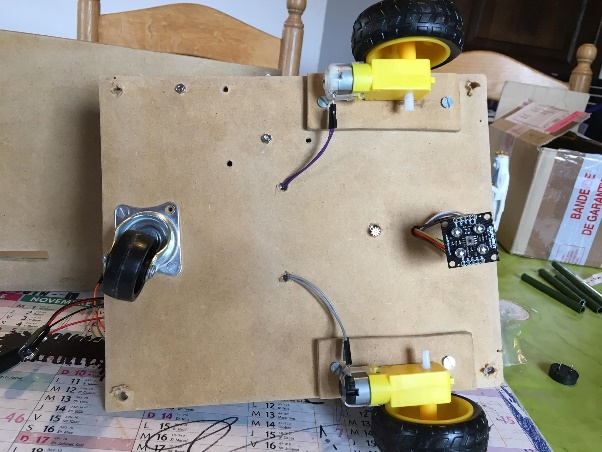
1. Appel de la fonction du calcul de distance
2. Déplacement du robot grâce aux appels des différentes fonctions
3. Fonction du calcul de distance
4. Fonction de la marche avant
5. Fonction de la marche arrière
6. Fonction du demi-tour
7. Fonction du stop
8. Fonction du calcul pour les couleurs RGB

**Montage robot**



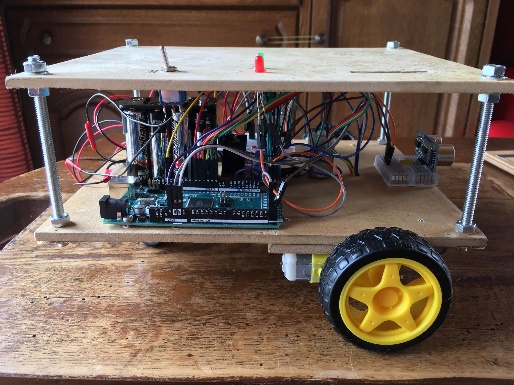


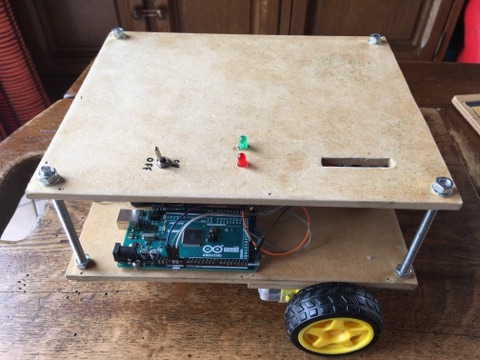












**Schéma électrique et cablage de l’Arduino**

1. **PARTIE 2: Raspberry + reconnaissance**
2. **Conclusion:**