

Compte-rendu n°4 Projet

Objectif de la séance : Pouvoir faire en sorte que chaque moteur de l'arrière du châssis s'active à partir du capteur à ultrason associé respectivement. Si cette étape est achevée alors la motorisation du train avant du robot sera similaire (on branchera deux moteurs et deux capteurs ultrasons sur une nouvelle carte arduino uno (deux au total pour la motorisation)).

Lors de la séance précédente, les capteurs à ultrasons renvoyaient une valeur nulle ce qui pouvait supposer deux choses :

- Soit le code était incomplet
- Soit le capteur était défectueux ou les branchements incorrèctes.

I-Debugage des capteurs

J'ai commencé par tester chacun des capteurs (arrière-gauche et arrière-droit) avec un code nous permettant d'obtenir la distance en cm entre le capteur respectif et un obstacle quelconque en utilisant uniquement des fonctions non bloquantes (millis et micros tout en supprimant tous les delays()).

Voici le code correspondant.

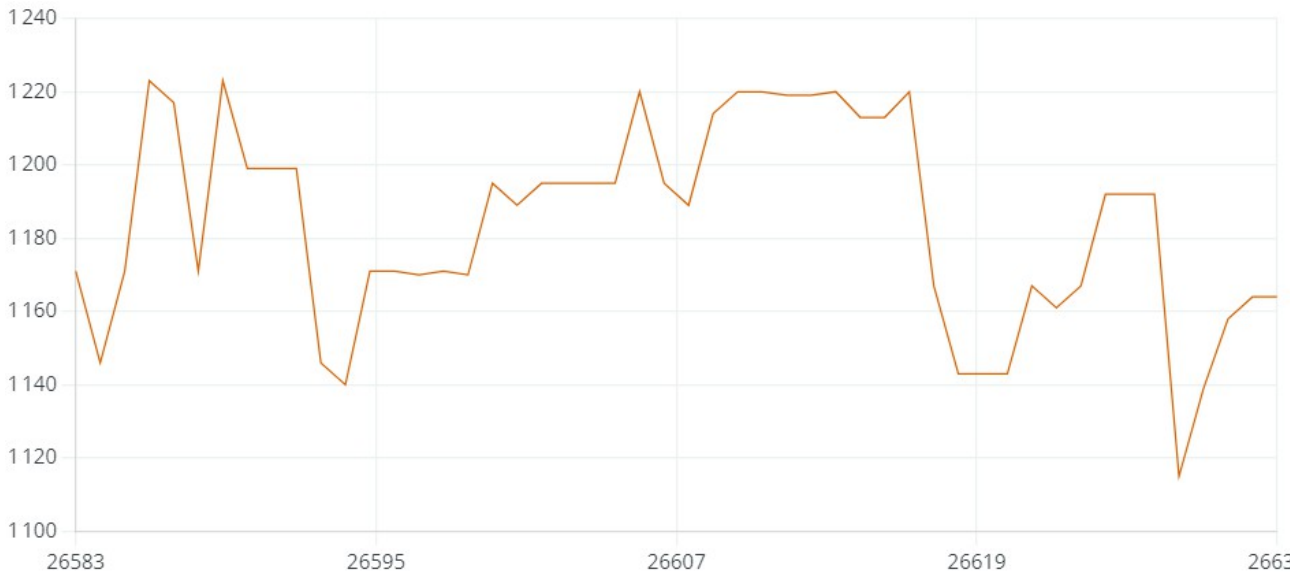
```
1  #define trig1  9
2  #define echo1  10
3
4  #define trig2  11
5  #define echo2  12
6
7  // Définir la distance minimale pour activer les moteurs (en cm)
8  #define DISTANCE_THRESHOLD 10
9
10 // Définir les broches de contrôle des moteurs
11 #define MOTOR_PIN_1 3
12 #define MOTOR_PIN_2 5
13 int impulseTime1;
14 int impulseTime2;
15 int distance1;
16 int distance2;
17 int lectureAD;
18 int lectureAG;
19 int initialMillis;
20 int initialMicros=0;
21 unsigned long currentMillis;
22 unsigned long currentMicros;
23
24
25 void setup() {
26     // put your setup code here, to run once:
27     Serial.begin(9600);
28     pinMode(trig1, OUTPUT);
29     pinMode(trig2, OUTPUT);
30     pinMode(echo1, INPUT);
31     pinMode(echo2, INPUT);
32     initialMillis=0;
33 }
```

```

35 void loop() { // put your main code here,
36   currentMillis=millis();
37   currentMicros=micros();
38   if ((currentMillis-initialMillis)>60){
39     digitalWrite(trig1,LOW);
40     if ((currentMicros-initialMicros)>2){
41       digitalWrite(trig1,HIGH);
42       initialMicros=currentMicros;
43     }
44     if ((currentMicros-initialMicros)>10){
45       digitalWrite(trig1,LOW);
46       initialMicros=currentMicros;
47     }
48     lectureAG=pulseIn(echo1,HIGH);
49     long distanceAG=lectureAG*0.017;
50     initialMillis=currentMillis;
51     Serial.print("AG ");
52     Serial.println(lectureAG);
53     //Serial.println(currentMillis);
54   }

```

Voici le graphique nous renvoyant le temps de propagation de l'onde ultrasonore entre deux obstacles en millisecondes en fonction du temps s'écoulant depuis le lancement du programme (millis()).



A partir de cette opération j'ai pu tester deux choses : Que d'une part les capteurs utilisés étaient fonctionnels et que d'autre part mon nouveau programme à base de millis fonctionnait.

J'ai modifié mon programme pour pouvoir cette fois ci supporter deux capteurs à ultrasons : la seule partie du programme modifiée (hors désignation de nouvelles variables) est la boucle loop().

En voici une capture d'écran :

```
35 void loop() { // put your main code here, to run repeatedly.
36   currentMillis=millis();
37   currentMicros=micros();
38   if ((currentMillis-initialMillis)>60){
39   |
40   ✓ digitalWrite(trig1,LOW);
41     if ((currentMicros-initialMicros)>2){
42       digitalWrite(trig1,HIGH);
43       initialMicros=currentMicros;
44     }
45     if ((currentMicros-initialMicros)>10){
46       digitalWrite(trig1,LOW);
47       initialMicros=currentMicros;
48     }
49   lectureAG=pulseIn(echo1,HIGH);
50   long distanceAG=lectureAG*0.017;
51   initialMillis=currentMillis;
52   Serial.print("AG ");
53   Serial.println(lectureAG);
54   //Serial.println(currentMillis);
55
56   ✓ digitalWrite(trig2,LOW);
57     if ((currentMicros-initialMicros)>2){
58       digitalWrite(trig2,HIGH);
59       initialMicros=currentMicros;
60     }
61     if ((currentMicros-initialMicros)>10){
62       digitalWrite(trig2,LOW);
63       initialMicros=currentMicros;
64     }
65
66   lectureAD=pulseIn(echo2,HIGH);
67   long distanceAD=lectureAD*0.017;
68   Serial.print("AD ");
69   Serial.println(lectureAD);
70 }
```

Les lignes 40 à 53 représentent l'activation du capteur arrière-gauche

Les lignes 56 à 68 représentent l'activation du capteur arrière-droit

Voici un échantillon des valeurs renvoyées par les deux capteurs à ultrasons représentant la distance en cm séparant le capteur respectif d'un obstacle (AG pour le capteur associé au moteur arrière gauche et AD pour le capteur associé au moteur arrière droit)

```
AG 20
AD 5
AG 24
AD 6
AG 39
AD 6
AG 22
AD 5
AG 20
```

Les capteurs ultrasons et le programme à base de millis étant fonctionnels, on ajoute les moteurs.

II-Activation des moteurs en fonction des capteurs d'obstacles

Le nouveau bout de code correspondant au moteur est ajouté dans loop(), on ne modifie pas le reste du code (en dehors de l'ajout des nouvelles variables et des nouveaux pins utilisés)

Voici l'ajout de code :

```
76  if (distanceAG == 0 || distanceAG > 10) {
77      // Aucun obstacle détecté à moins de 10 cm, activer le premier moteur
78      digitalWrite(moteurAG,HIGH);
79      }
80  else {
81      // Obstacle détecté à moins de 10 cm, désactiver le deuxième moteur
82      digitalWrite(moteurAG,LOW);
83      }
84
85
86  // Vérifier le deuxième capteur
87  if (distanceAD == 0 || distanceAD > 10) {
88      // Aucun obstacle détecté à moins de 10 cm, activer le deuxième moteur
89      digitalWrite(moteurAD,HIGH);
90      }
91  else {
92      // Obstacle détecté à moins de 10 cm, désactiver le deuxième moteur
93      digitalWrite(moteurAD,LOW);
94      }
95  }
96
97
98  } //Renvoie la distance détectée entre l'obstacle et le capteur ultrasonore
99  }
```

Le code fonctionne parfaitement et répond aux attentes : Les deux capteurs et les moteurs fonctionnent de manière simultanée : il n'y a plus de blocage dû aux delay().

Voici une vidéo résumant l'activation et la désactivation des moteurs en fonction de la distance détectée par les capteurs d'obstacles: (voir dans le github)

III-Revissage/ remplacement de certaines vis et collage

Les boulons fixés au châssis ne tenaient pas après activation des moteurs (ils se dévissaient en permanence). Certaines n'étaient pas adaptées (vis trop courtes utilisées dû à un manque de vis dans le colis lors de la livraison en raison d'une fouille à la douane) et j'ai donc cherché de nouvelles vis au fablab pour pouvoir les remplacer. Après avoir simplement revissé plusieurs fois les boulons, j'ai décidé cette fois-ci de fixer ces derniers avec de la colle forte en tant que solution provisoire.