

I. Présentation des outils

1. Le robot BOE Shield-Bot

Le robot que nous allons utiliser se compose de trois parties. La première est le châssis correspondant à l'armature et donc la base du robot. La deuxième partie héberge une carte arduino Uno. Celle-ci est composée principalement d'un micro-contrôleur programmable (qui contiendra notre programme), d'un port série (connexion PC-robot) et des broches de communication. La dernière partie héberge une carte composée de broches de connexion, ainsi que d'une plaquette. La composition du robot est montrée en figure 1.

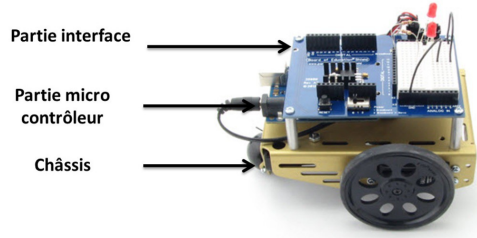


Figure 1 : Détail du robot

Nous nous intéresserons uniquement à la dernière carte permettant d'augmenter les fonctionnalités du robot. Nous utiliserons principalement la plaquette pouvant accueillir des LEDs, un buzzer, des détecteurs infrarouges... La gestion de l'alimentation se fait par le biais de la carte arduino Uno. Il existe deux façons d'alimenter la carte, via le port USB ou cinq piles de type LR6 qui seront placées dans le réceptacle sous l'armature du robot.

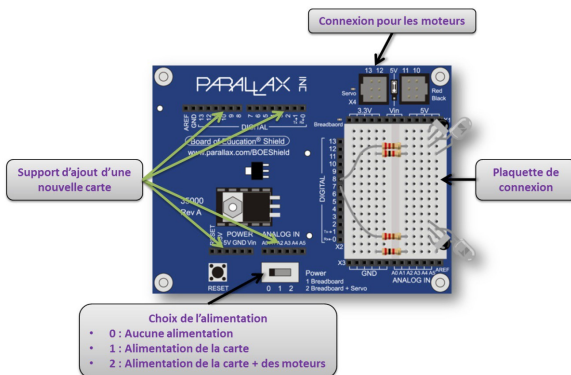


Figure 2 : Schéma de l'interface

La figure 2 montre la carte et ses composants.

Dans un premier temps, nous utiliserons le châssis connecté directement par USB à l'ordinateur.

Intérêt du robot Shield-Bot :

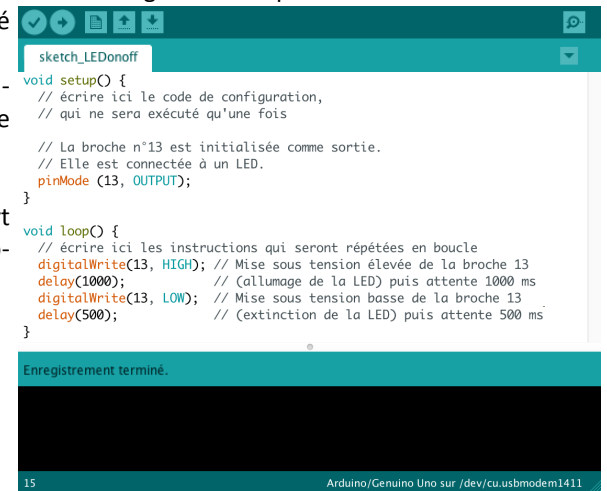
Une fois le code téléchargé sur la carte du châssis, le code peut continuer à s'exécuter. Ainsi, à condition de l'équiper d'une source autonome d'énergie (les piles LR6), il est possible d'envisager avec ce système la programmation de systèmes embarqués. Exemples : engins spatiaux, télévision, disque dur externe, imprimante multifonction, console de jeux, avionique, automobile,...

2. Le logiciel Arduino

Nous allons dans cette partie présenter le logiciel permettant de développer des programmes pour le robot BOE Shield-bot. Le logiciel est disponible sur les ordinateurs du lycée dans le répertoire partagé ISN/Logiciels.

L'environnement de développement se présente sous la forme donnée par la figure 3

- ✓ vérification du programme
- ➔ compilation du code et export du programme vers le micro-contrôleur
- 📄 création d'un nouveau fichier
- 📂 ouverture d'un fichier
- 💾 sauvegarde du fichier
- 🔍 ouverture du moniteur série



Arduino utilise le langage de programmation C++, avec deux fonctions :

- la fonction `setup()` ne s'exécute qu'une seule fois
- la fonction `loop()` boucle indéfiniment sur elle-même

Pour téléverser (exporter) un programme sur le micro-contrôleur, celui-ci doit être branché *via* un port USB à l'ordinateur.

II. Manipulation du robot et développement

1. Afficher du texte dans la console

Copier le code suivant dans l'interface de développement Arduino :

```
void setup() {
    Serial.begin(9600);           // vitesse de communication
    Serial.print("Bonjour");     // affichage dans la console
}
void loop() {
```

Compiler et téléverser le code vers le micro-contrôleur. Ouvrir la console et vérifier que la vitesse de traitement de la console vaut bien 9600 baud.

Modifier ce code afin de faire afficher :

```
Bonjour
tout le monde
```

Remarque : Pour imposer un retour à la ligne à la fin du texte affiché, remplacer la commande `print` par `println`.

2. Faire clignoter les DEL

Des circuits électriques contenant des diodes électroluminescentes (DEL) sont connectés à votre plaquette, respectivement sur les broches 13 et 12. Pour que la DEL connectée à la broche 13 s'allume, il faut donner l'ordre de mettre la broche 13 en tension élevée. Pour qu'elle s'éteigne, la broche 13 doit être mise en tension faible.

Copier le code suivant dans l'interface de développement Arduino :

```
void setup() {
  pinMode(13, OUTPUT); // la broche 13 est connectée en tant que porte de sortie
  pinMode(12, OUTPUT); // la broche 12 est connectée en tant que porte de sortie
}
void loop() {
  digitalWrite(13, HIGH); // allumage de la DEL 13
  delay(500); // attente de 500 ms avant l'instruction suivante
  digitalWrite(13, LOW); // extinction de la DEL 13

  digitalWrite(12, HIGH); // allumage de la DEL 12
  delay(500); // attente de 500 ms avant l'instruction suivante
  digitalWrite(12, LOW); // extinction de la DEL 12
}
```

Modifier ce code pour que la LED 13 reste allumée pendant 1000 ms, puis que la LED 12 reste allumée pendant 100 ms.

Selon le même principe de branchement que les deux DEL déjà en place, ajouter deux couples DEL/résistance sur les broches 11 et 10. Modifier le code pour faire en sorte que la lumière semble se déplacer de DEL en DEL de manière fluide.

3. Faire avancer le robot

Un servomoteur est associé à chaque roue du châssis. La plaquette commande chaque servomoteur en choisissant le sens et la vitesse de rotation.

Pour cela, on utilise une bibliothèque d'instructions dédiées aux servomoteurs (*library* en anglais).

Copier le code suivant dans l'interface de développement Arduino :

```
#include <Servo.h> // utilisation de la bibliothèque

Servo servoLeft; // déclaration du moteur gauche
Servo servoRight; // déclaration du moteur droit

void setup() {
  servoLeft.attach(13); // communication avec le moteur gauche via la broche 13
  servoRight.attach(12); // communication avec le moteur droit via la broche 12
  // *
}

void loop() {
}
```

Le code précédent permet de déclarer les moteurs et de les connecter logiquement à leur broche. Il est maintenant possible de leur donner des instructions :

Au niveau de la ligne * du code précédent, ajouter le code suivant :

```
servoLeft.writeMicroseconds(1300);
servoRight.writeMicroseconds(1700);
delay(500);

servoLeft.writeMicroseconds(1700);
servoRight.writeMicroseconds(1300);
delay(500);

servoLeft.writeMicroseconds(1500);
servoRight.writeMicroseconds(1500);
delay(500);
```

Que fait le robot ?

Compléter le tableau suivant :

servoLeft broche 13	servoRight broche 12	Comportement
1700	1300	
1300	1700	
1500	1500	
1300	1300	
1700	1700	
1480	1520	
1500	1700	

Mesurer la distance parcourue par le robot pour l'exécution de la commande suivante :

```
servoLeft.writeMicroseconds(1300);
servoRight.writeMicroseconds(1700);
delay(2000);
```

III. Premiers défis

1. Programmer le robot afin qu'il avance de 10 cm.
2. Programmer le robot afin qu'il effectue un quart de tour à gauche.
3. Programmer le robot afin qu'il effectue un rectangle de 10 cm par 20 cm.

Références :

<https://www.arduino.cc/>

Initiation à la programmation du robot BOE Shield-Bot – Sébastien Bindel (LABRI)