

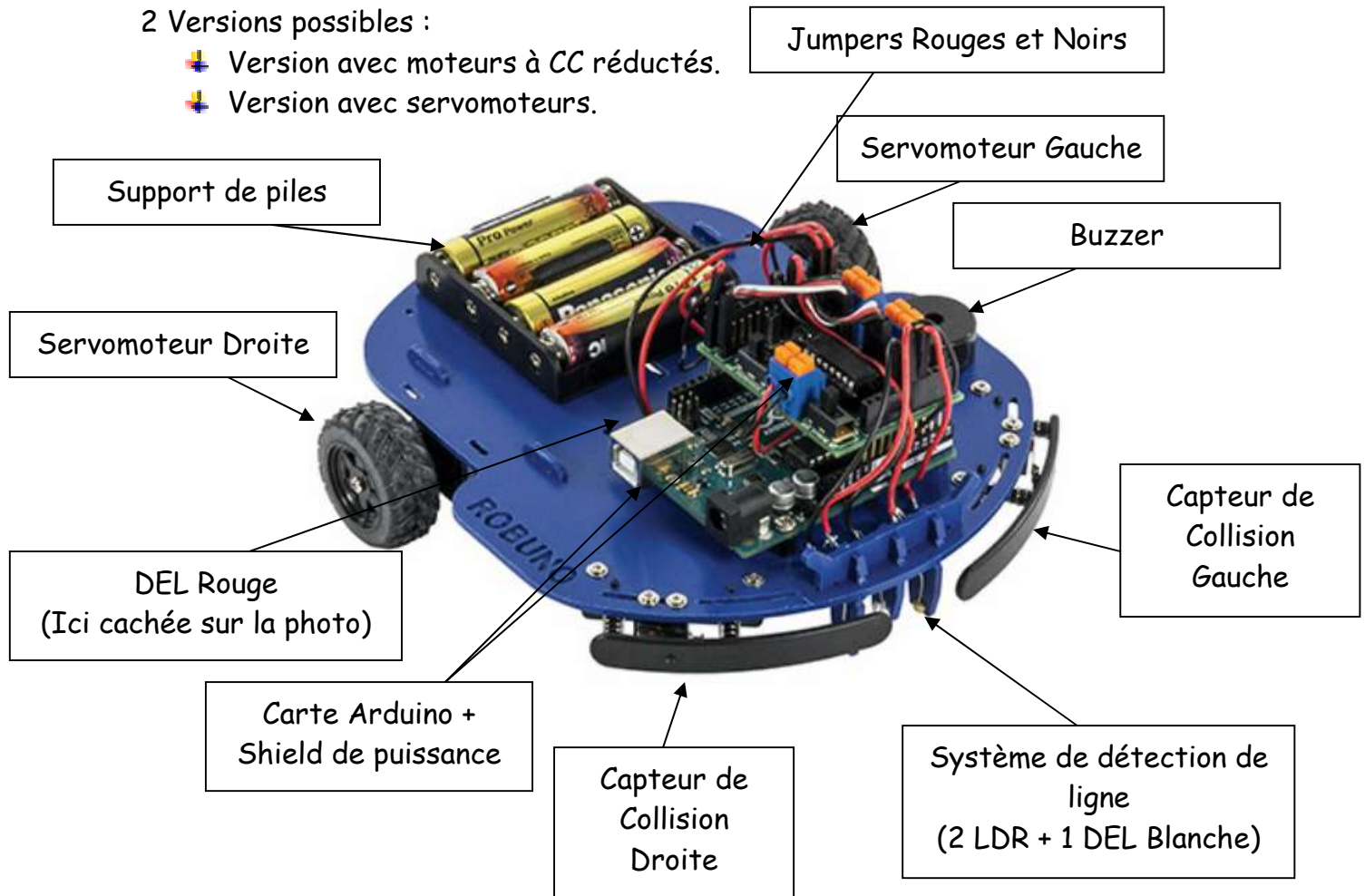
RobUno et Blockly

1) Présentation du robot mobile :

Technologie Services, distributeur de matériels pédagogiques, a sorti un nouveau Robot Mobile appelé RobUno.

2 Versions possibles :

- 🔧 Version avec moteurs à CC réduits.
- 🔧 Version avec servomoteurs.



2) Quelques points d'analyse après quelques premiers essais :

- 🔧 Ultra facile à monter.
- 🔧 Mode d'emploi et gamme de montage disponible sur leur site Internet :
<http://www.technologieservices.fr/fr/a-a1000022617/ressource/1001727/Dossier-d-aide-au-montage-et-programme-RUSA.html?typeArti=ressource>
- 🔧 Le shield de puissance permet de connecter au choix 2 moteurs à CC ou 2 servomoteurs ; divers capteurs ; juste à raccorder... Et ça marche !
- 🔧 L'utilisation avec le système suivi de ligne nécessite (d'après mes premiers tests) une alimentation annexe pour le shield. En effet, l'alimentation principale s'écroule complètement en utilisant les LDR, la diode blanche et les servomoteurs.
- 🔧 Les « moustaches » pour la détection d'obstacle sont plutôt bien pensées tant dans leur fonctionnement que dans leur installation.

3) Affectation des capteurs et actionneurs sur la carte Arduino :

Matériels	Broche Arduino
Capteur de collision Gauche	2
Capteur de collision Droite	3
Servomoteur Gauche	5
Servomoteur Droite	6
DEL Rouge	7
DEL Blanche	8
LDR Gauche	A1
LDR Droite	A0

4) Dans Blockly@rduino :

J'ai réalisé des blocs de programmation pour ce robot dans Blockly@rduino. Notre collègue S.Canet a validé mes blocs et les a intégré à son github :

<https://github.com/technologiescollege/Blockly-at-rduino>

Pour les retrouver, une fois Blockly@rduino ouvert, cliquer sur l'onglet « **Configurer les blocs** », puis dans la fenêtre de dialogue, cocher « **RobUno** » ; puis « **Valider** ».

LES CAPTEURS

The screenshot shows the Blockly@rduino web interface. At the top, it says "Blockly@rduino : éditeur graphique pour aider à la programmation des interfaces". On the left, there's a sidebar with a dropdown menu for "carte" set to "Arduino UNO". Below this are several buttons: "afficher la carte", "configurer les blocs", "supervision de la carte", "blocs", "code Arduino", "console Série", "traduction XML", "sauver en fichier XML", "charger un fichier XML", and "ouvrir un exemple". In the center, there's a category menu with "logique", "boucles", "maths", "variables", "fonctions", "arduino", "RobUno", "capteurs", and "actionneurs". The "RobUno" category is expanded, showing "capteurs" and "actionneurs". On the right, there's a workspace with four green blocks. The first two blocks are "Le capteur de collision Gauche" and "Le capteur de collision Droite", both set to "sur la broche 1" and "est appuyé ?". The next two blocks are "la luminosité lue sur la LDR Gauche" and "la luminosité lue sur la LDR Droite", both set to "sur la broche Analogique 1".

LES ACTIONNEURS



5) Le programme dans Blockly@rduino :






Objectif: Programmer RobUno pour qu'il puisse se déplacer dans un environnement en détectant les obstacles via ses capteurs de collision.

Lors de sa manœuvre d'évitement (recul puis pivot à gauche ou à droite), suite à une détection d'obstacles, le robot doit émettre un son et allumer son voyant Rouge.

Vidéo de démonstration ici : <https://youtu.be/ekx3U1PyiO4>

Etape n°1 :

Créer les fonctions :

-  Avancer
-  Reculer
-  Pivoter sur la gauche
-  Pivoter sur la droite
-  Signal et alarme collision



Pour ne pas surcharger le programme final, j'appelle dans les fonctions **Reculer**, **Pivoter à gauche** et **Pivoter à droite** la fonction : **Signal et alarme collision**



Etape n°2 :

Créer le programme :



6) Le Code Arduino généré par Blockly@rduino :

```
#include <Servo.h>
```

```
Servo servo_5;
```

```
Servo servo_6;
```

```
void Avancer() {  
  servo_5.write(90 - map(90, 0, 255, 0, 90));  
  servo_6.write(90 + map(90, 0, 255, 0, 90));  
}
```

```
void Reculer() {  
  servo_5.write(90 + map(90, 0, 255, 0, 90));  
  servo_6.write(90 - map(90, 0, 255, 0, 90));  
  Signal_et_alarme_collision();  
  delay(1000);  
}
```

```
void Pivoter_sur_la_gauche() {  
  servo_5.write(90 + map(90, 0, 255, 0, 90));  
  servo_6.write(90 + map(90, 0, 255, 0, 90));  
  Signal_et_alarme_collision();  
  delay(1000);  
}
```

```
void Pivoter_sur_la_droite() {  
  servo_5.write(90 - map(90, 0, 255, 0, 90));  
  servo_6.write(90 - map(90, 0, 255, 0, 90));  
}
```

```
    Signal_et_alarme_collision();  
    delay(1000);  
}
```

```
void Signal_et_alarme_collision() {  
    tone(13,10000,1000);  
    digitalWrite(7,HIGH);  
    delay(1000);  
    digitalWrite(7,LOW);  
}
```

```
void setup() {  
    servo_5.attach(5);  
  
    servo_6.attach(6);  
  
    pinMode(7, OUTPUT);  
    pinMode(2, INPUT);  
    pinMode(3, INPUT);  
}
```

```
void loop() {  
    if (digitalRead(2)==1) {  
        Reculer();  
        Pivoter_sur_la_droite();  
  
    } else if (digitalRead(3)==1) {  
        Reculer();  
        Pivoter_sur_la_gauche();  
    } else {  
        Avancer();  
  
    }  
  
}
```