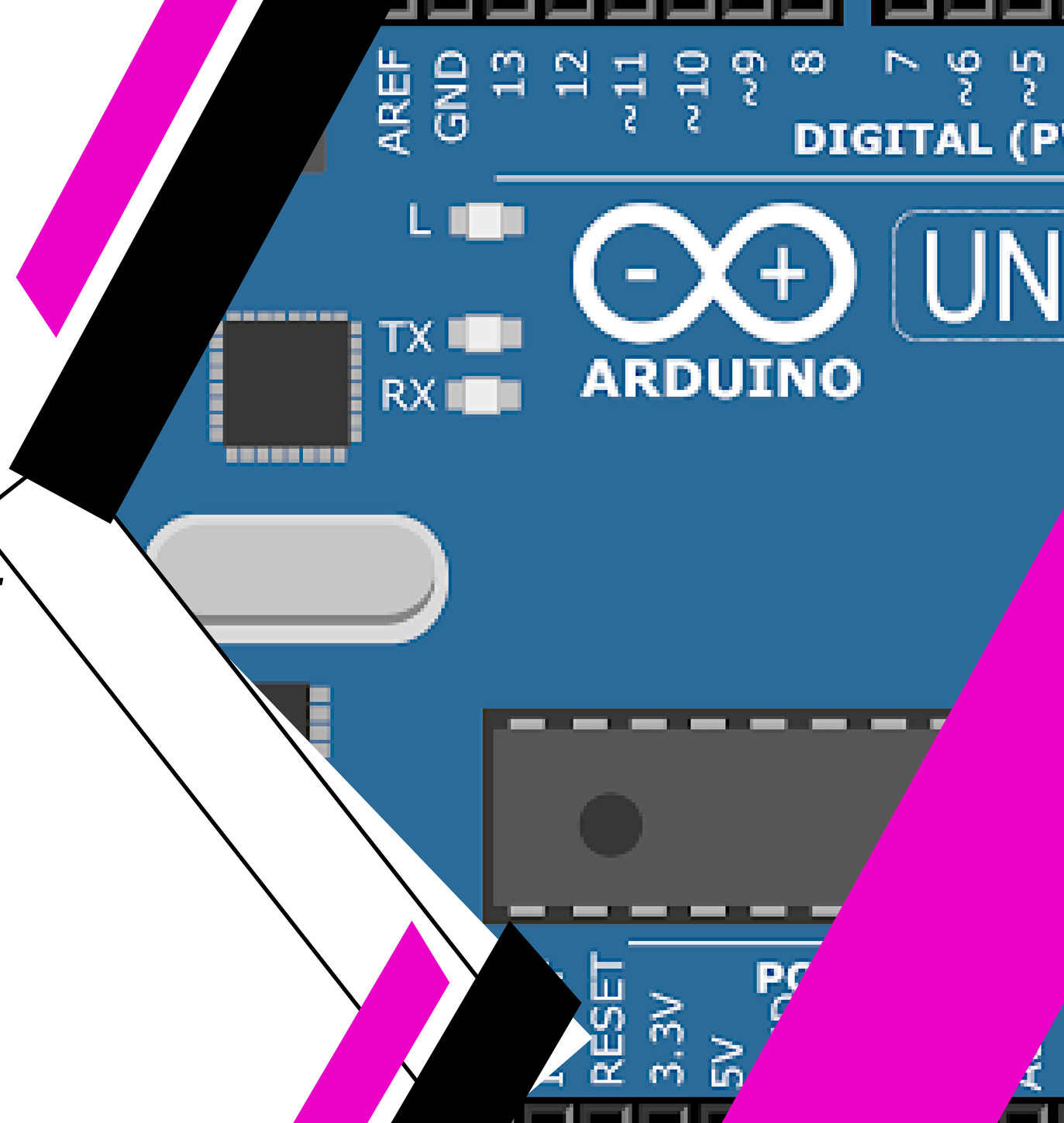
ARDUINO

INITIATION À LA ROBOTIQUE



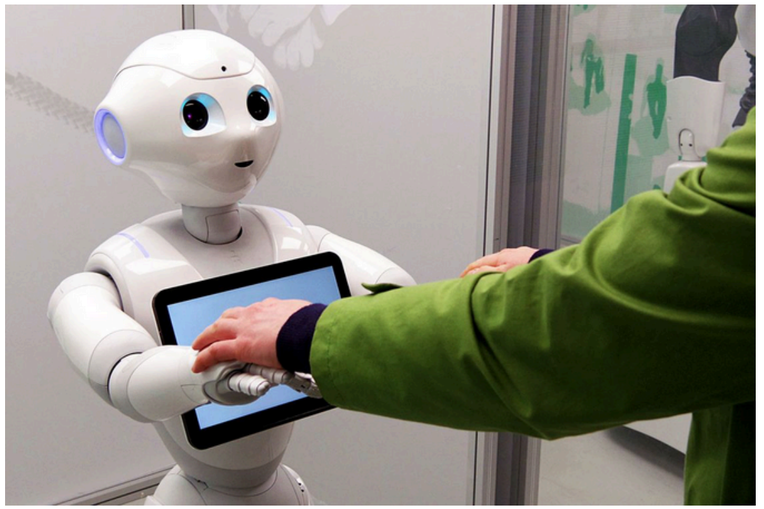
FÉDÉRATION VAR
la ligue de l'enseignement
un avenir par l'éducation populaire

CC BY SA

La robotique c’est quoi?

La robotique C’est quoi?
La robotique est un domaine passionnant qui connaît une forte croissance. Beaucoup de personnes veulent apprendre à créer des robots, mais il est parfois difficile de savoir où commencer.

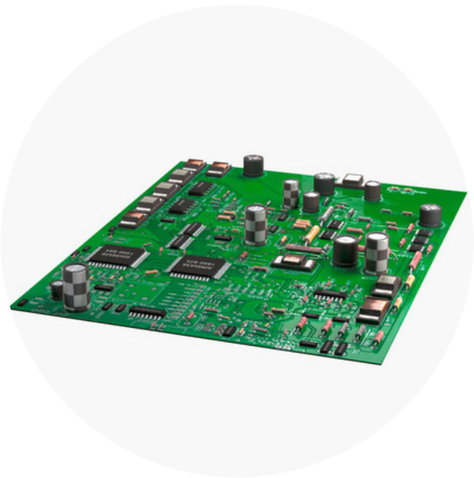
La formation robotique est une formation qui permet de construire des robots afin de répondre à différentes tâches. Cela n’est pas si simple car il faut concevoir un programme différent à chacun mais aussi trouver des solutions pour l’entretenir.



Les trois domaines de la robotique.

1 - Mécanique

La mécanique constitue le corps du robot.



2 - Eléctronique

Constitue la partie circuit imprimé ainsi que le cablage du robot.

3 - Programmation

Défini la partie code du robot. (Intelligence Artificielle)

L'Arduino fait partie du domaine 2 et 3 de la robotique.

Déroulement de l'activité

01

Présentation du matériel fournie avec le kit Arduino.

02

Premier projet: Journal du vaisseau spacial.

03

Deuxième projet:
Love O Meter.

04

Troisième projet:
Thérémine lumineux.

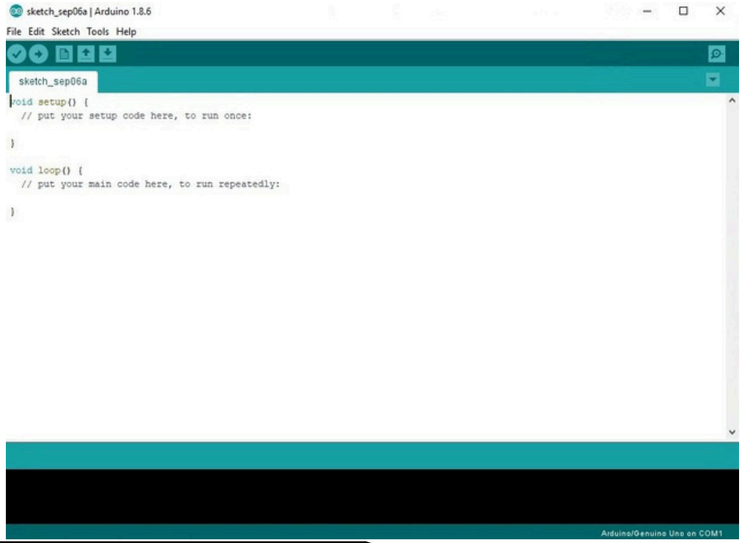
Partie 1 :
Matériel Arduino

Starter kit Arduino UNO

https://store.arduino.cc/collections/kits/products/arduino-starter-kit-multi-language

Logiciel Arduino IDE

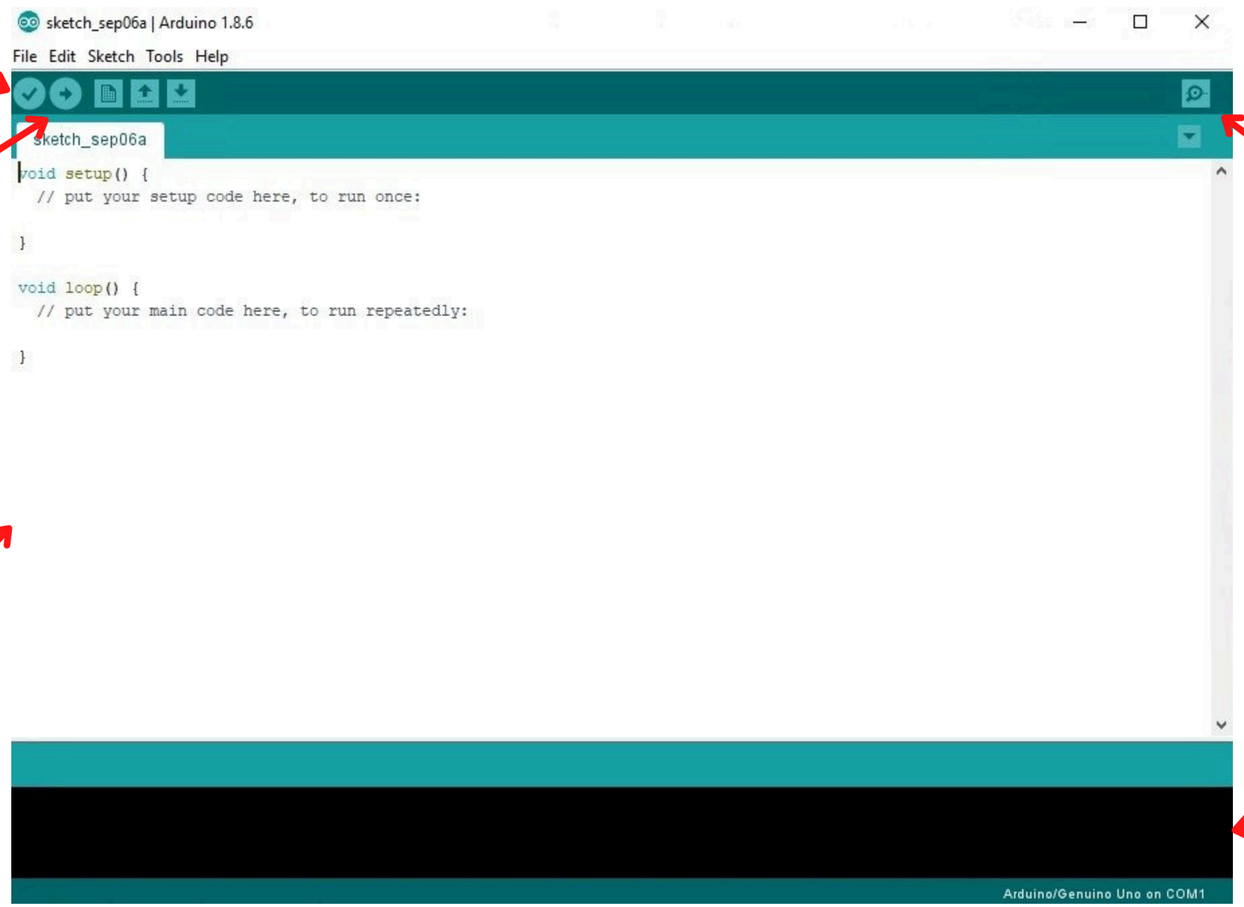
https://apps.microsoft.com/store/detail/arduino-ide/9NBLGGH4RSD8?hl=fr-fr&gl=FR



Logiciel Arduino IDE

Vérifié le code

Téléverser le code dans la carte Arduino



Espace réservé au code

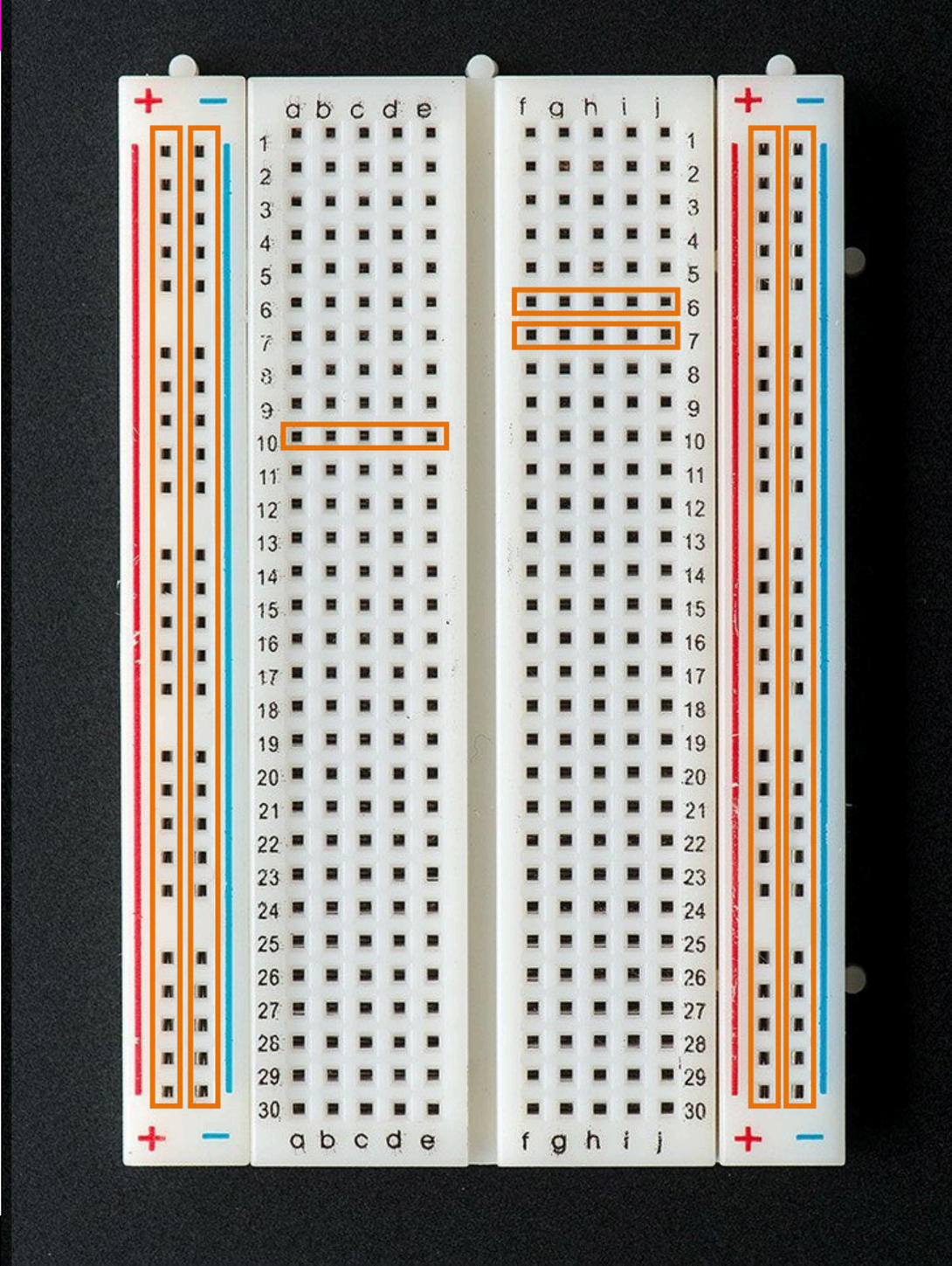
Moniteur série /debug nomé serial dans le code pour afficher les valeurs.

Console où seront afficher les erreurs.

La Breadboard

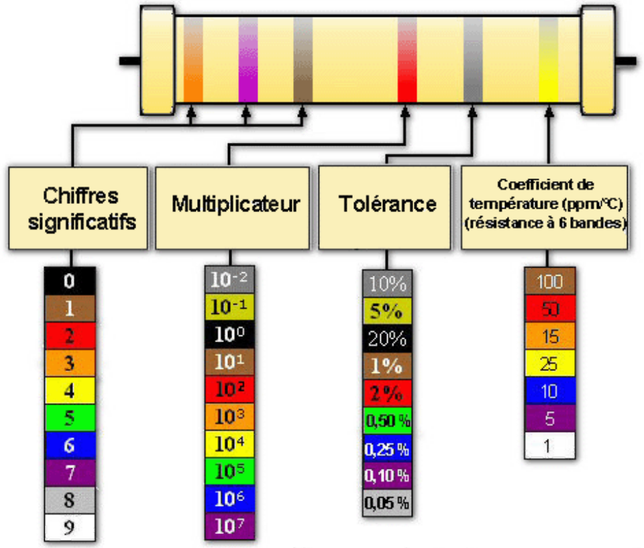
La breadboard est utilisée pour expérimenter des montages électroniques sans avoir besoin de souder les composants.

Sur la face avant, la plaque est constituée de plusieurs trous. Ces trous sont reliés entre eux , grâce à des barrettes métalliques. Le courant éléctrique progresse dans le sens des rectangles orange sur l'image. *(Verticalement pour les bornes + et - et horizontalement pour le centre)*.



Le code couleur des résistances

Chaque couleur correspond à un chiffre et chaque résistance à 4 ou 5 bandes. Pour celle à 4 bandes , les deux premières correspondent aux deux premiers chiffres de leur valeur et la troisième au nombre de zéros qui suit.



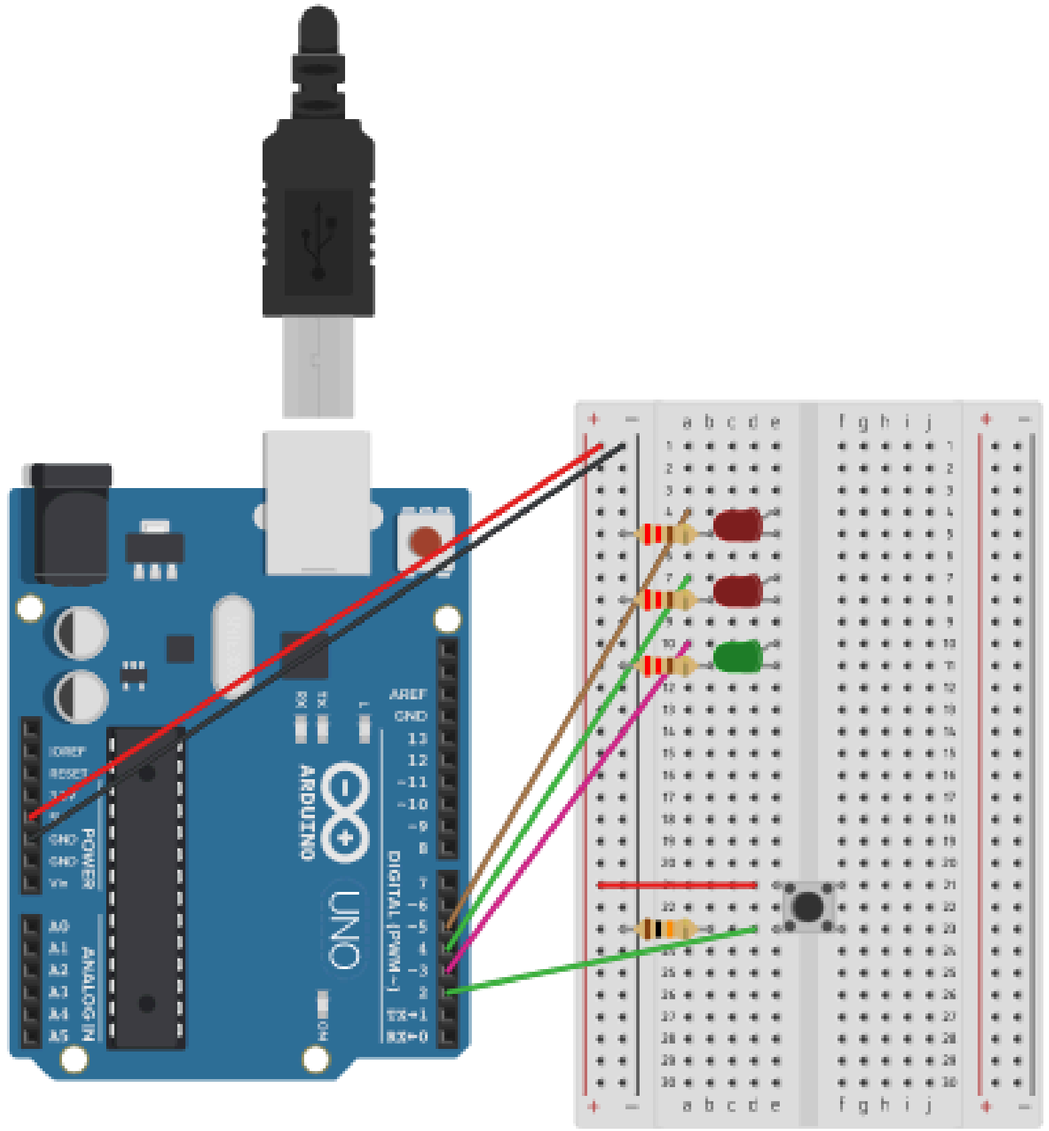
Partie 2 :

Vaisseau spacial - page 33 du manuel

C’est un montage composé de 3 circuits montés en série, chacun relié en dérivation.

Materiel utilisé :

- 1 Carte Arduino
- Le cable USB
- Des fils
- 1 LED verte
- 2 LEDs rouge
- 1 bouton
- 3 résistances 220 ohms (LEDs)
- 1 résistance 27 ohms (bouton)



Code du vaisseau spacial

vaisseau_spacial | Arduino 1.8.9

Fichier Édition Croquis Outils Aide

Téléverser

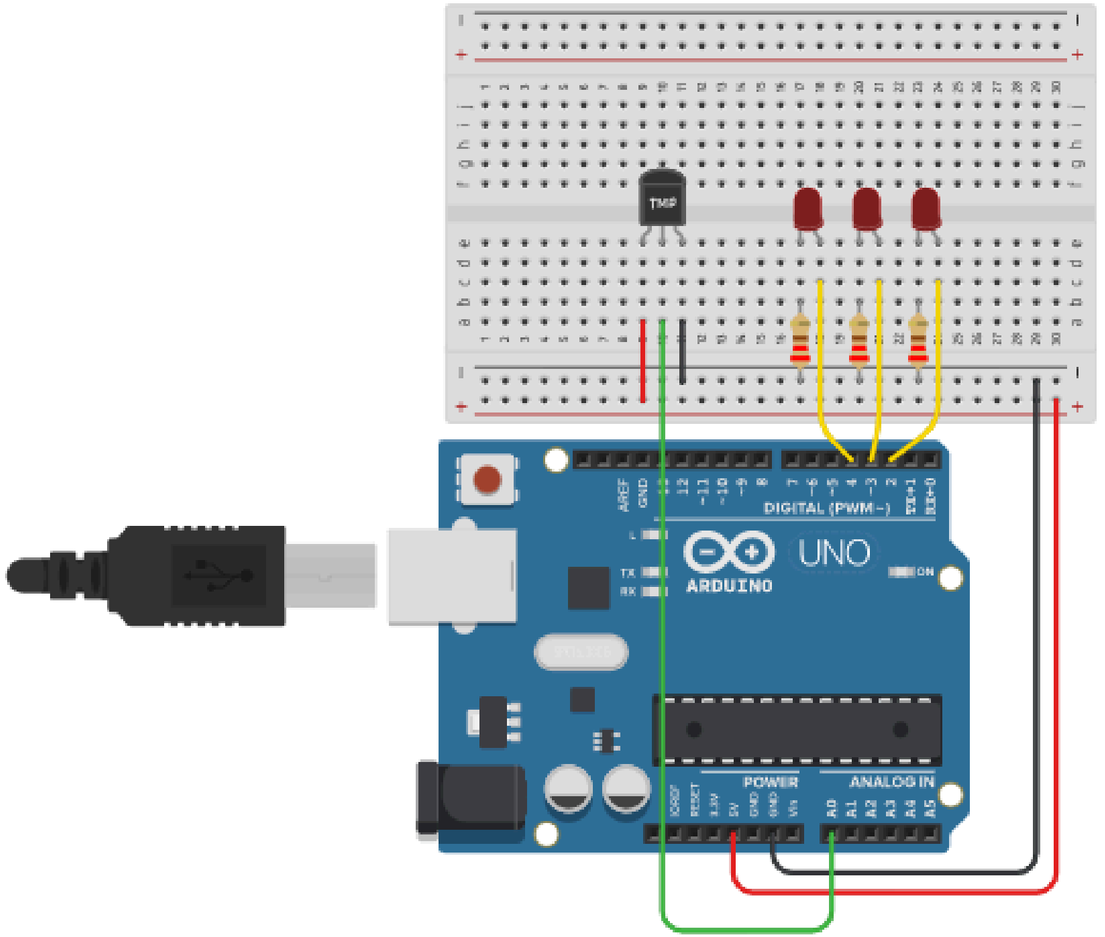
vaisseau_spacial

```
int changementEtat = 0;     // déclaration d'une variable

void setup() {
  // déclaration des LED en tant que sortie (OUTPUT)
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);

  // déclaration du bouton comme entré (INPUT)
  pinMode(2, INPUT);
}

void loop() {

  //stock la valeur lu dans la pin 2 par la fonction digitalRead() dans la variable
  changementEtat = digitalRead(2);


  if (changementEtat == LOW) {     // si le bouton est pas pressé
    digitalWrite(3, HIGH); // allumer la LED verte sur le pin 3
    digitalWrite(4, LOW);  // éteindre la LED rouge sur le pin 4
    digitalWrite(5, LOW);  // éteindre la LED rouge sur le pin 5
  }

  else {     // sinon (le bouton est pressé), alterner les LEDs rouge
    digitalWrite(3, LOW);  // éteindre la LED verte sur le pin 3
    digitalWrite(4, LOW);  // éteindre la LED rouge sur le pin 4
    digitalWrite(5, HIGH); // allumer la LED rouge sur le pin 5

    delay(250);   // Attendre 250 milliseconds

    digitalWrite(4, HIGH); // allumer la LED rouge sur le pin 4
    digitalWrite(5, LOW);  // éteindre la LED rouge sur le pin 5

    delay(250);
  }
}
```

Partie 3 :

Love O Meter - page 43 du manuel

Notre Arduino va se transformer en love machine pour observer à quelle point vous êtes amoureux. Un capteur de température mesure la chaleur.



Matériel utilisé :

- 1 Carte Arduino
- Le cable USB
- Des fils
- 1 TMP36 (capteur de temperature)
- 3 LEDs rouge
- 3 résistance de 220 ohms (LEDs)

Code du Love O Meter

LoveOMeter | Arduino 1.8.9

Fichier Édition Croquis Outils Aide

LoveOMeter

```
const int sensorPin = A0;   // variable pour la broche connecté au capteur (TMP36)
const float baselineTemp = 20.0;    // variable pour stocker la temperature moyenne

void setup() {
  Serial.begin(9600);   // ouvre le debug pour afficher les valeurs
  // on defini les LEDs comme sortie
  // la boucle for() économide un peu de codage suplémentaire
  for (int pinNumber = 2; pinNumber < 5; pinNumber++) {
    pinMode(pinNumber, OUTPUT);
    digitalWrite(pinNumber, LOW);
  }
}
void loop() {
  int sensorVal = analogRead(sensorPin);    // lecture de la valeur du capteur sur le pin A0
  Serial.print("sensor Value: ");   // afficher la valeur dans le debug
  Serial.print(sensorVal);

  float voltage = (sensorVal / 1024.0) * 5.0;   // convertiion de la valeur lu en tension
  Serial.print(", Volts: ");    // afficher la valeur dans le debug
  Serial.print(voltage);

  Serial.print(", degrees C: ");    // convertion de la tension en temperature(°C)
  float temperature = (voltage - 5) * 100;
  Serial.println(temperature);

  if (temperature < baselineTemp + 2) {    // si la temperature est inferieur à la temperature moyenne
    digitalWrite(2, LOW); // LED éteinte(LOW)
    digitalWrite(3, LOW); // LED éteinte(LOW)
    digitalWrite(4, LOW); // LED éteinte(LOW)
  } // si la temperature augmente de 2 à 4 degrés
  else if (temperature >= baselineTemp + 2 && temperature < baselineTemp + 4) {
    digitalWrite(2, HIGH);  // LED allumé(HIGH)
    digitalWrite(3, LOW); // LED éteinte(LOW)
    digitalWrite(4, LOW); // LED éteinte(LOW)
  } // si la temperture augmente de 4 à 6 degrés
  else if (temperature >= baselineTemp + 4 && temperature < baselineTemp + 6) {
    digitalWrite(2, HIGH);  // LED allumé(HIGH)
    digitalWrite(3, HIGH);  // LED allumé(HIGH)
    digitalWrite(4, LOW);   // LED éteinte(LOW)
  }
  else if (temperature >= baselineTemp + 6) {   // si la temperature augmente de 6 ou plus
    digitalWrite(2, HIGH);  // LED allumé(HIGH)
    digitalWrite(3, HIGH);  // LED allumé(HIGH)
    digitalWrite(4, HIGH);  // LED allumé(HIGH)
  }
  delay(1); // attendre 1 millisecondes
}
```

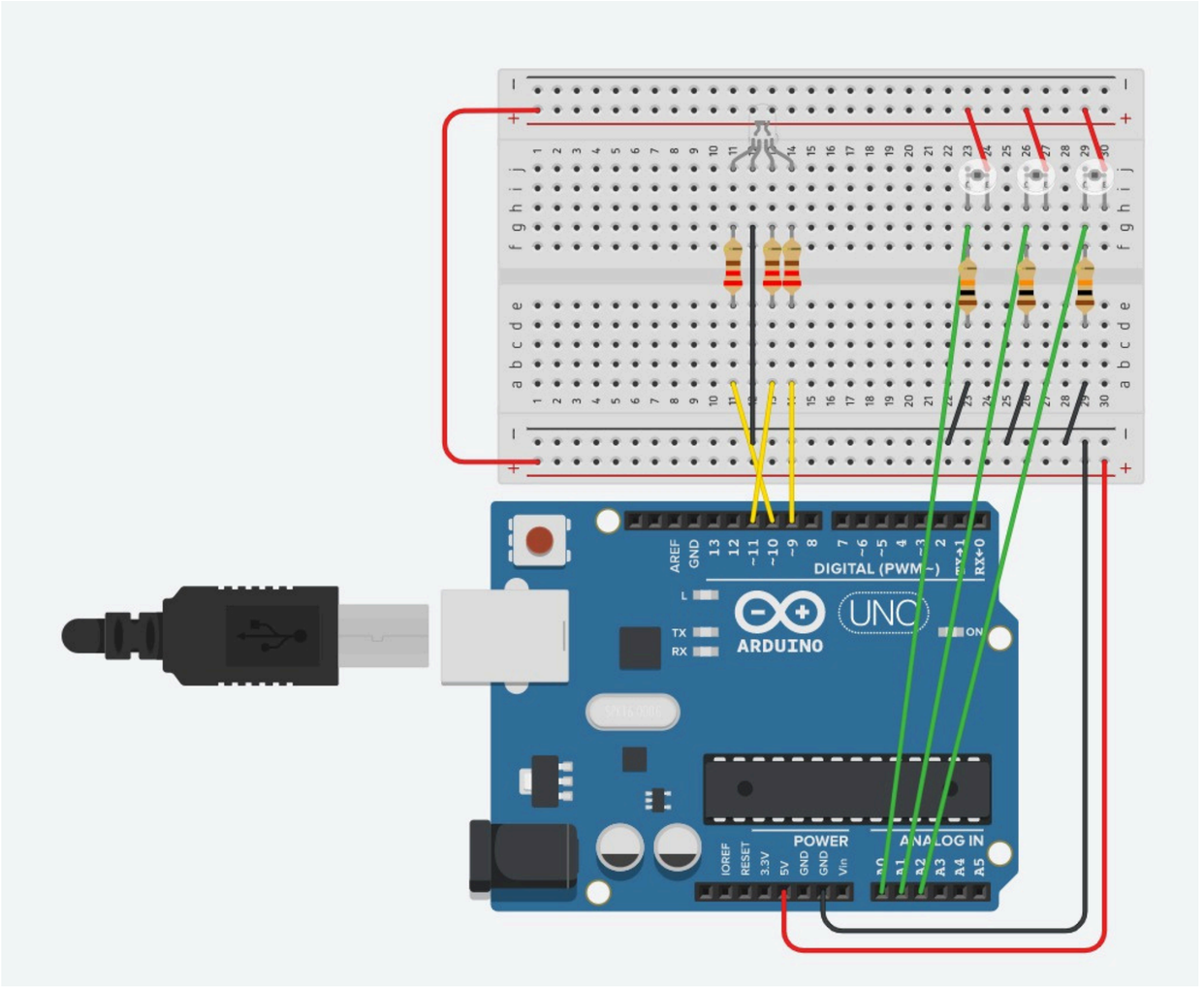
Partie 4 :

Lampe d'ambiance - page 52 du manuel

En utilisant une LED de trois couleurs (RVB) et trois phototransistor, vous pouvez céer une lampe qui change de couleur selon les conditions de l'éclairage externe.

Matériel utilisé :

- 1 Carte Arduino
- Le cable USB
- Des fils
- 1 LED trois couleur (RVB)
- 3 phototransistors
- 3 résistance de 220 ohms (RVB)
- 3 résistance de 10 kilo ohms (phototransistor)



Code de la lampe d'ambiance partie 1

RVB | Arduino 1.8.9

Fichier Édition Croquis Outils Aide

Téléverser

RVB §

```
const int greenLEDPin = 9;     // LED connecté au pin 9 (vert)
const int redLEDPin = 10;      // LED connecté au pin 10 (rouge)
const int blueLEDPin = 11;     // LED connecté au pin 11 (bleu)

const int redSensorPin = A0;     // pin avec le phototransistor pour la couleur rouge
const int greenSensorPin = A1;   // pin avec le phototransistor pour la couleur verte
const int blueSensorPin = A2;    // pin avec le phototransistor pour la couleur bleu

int redValue = 0; // vairiable pour la valeur de la couleur rouge
int greenValue = 0; // vairiable pour la valeur de la couleur verte
int blueValue = 0; // vairiable pour la valeur de la couleur bleu

int redSensorValue = 0; // variable pour garder la valeur du phototransistor rouge
int greenSensorValue = 0; // variable pour garder la valeur du phototransistor verte
int blueSensorValue = 0; // variable pour garder la valeur du phototransistor bleu

void setup() {
  Serial.begin(9600);   // initalisé le debug à 9600 bauds
  pinMode(greenLEDPin, OUTPUT);   // défini un des pin de la LED RVB en sortie (vert)
  pinMode(redLEDPin, OUTPUT);   // défini un des pin de la LED RVB en sortie (rouge)
  pinMode(blueLEDPin, OUTPUT);    // défini un des pin de la LED RVB en sortie (bleu)
}

void loop() {
  // on lit la valeur donnée par les capteur
  redSensorValue = analogRead(redSensorPin);    // lecture de la valeur du phototransistor rouge
  delay(5);                                     // attendre 5 millisecondes (pour ne pas surcharger de valeur le debug)
  greenSensorValue = analogRead(greenSensorPin);    // lecture de la valeur du phototransistor vert
  delay(5);                                         //  attendre 5 millisecondes
  blueSensorValue = analogRead(blueSensorPin);    // lecture de la valeur du phototransistor bleu
```

Code de la lampe d'ambiance partie 2

```
  // affichage des valeurs dans le debug
  Serial.print("raw sensor Values \t red: ");
  Serial.print(redSensorValue);
  Serial.print("\t green: ");
  Serial.print(greenSensorValue);
  Serial.print("\t Blue: ");
  Serial.println(blueSensorValue);

  // on effectue des calculs, pour convertire les tension donnée par les capteur en valeur réel
  redValue = redSensorValue / 4;
  greenValue = greenSensorValue / 4;
  blueValue = blueSensorValue / 4;

  // affichage des valeurs dans le debug
  Serial.print("Mapped sensor Values \t red: ");
  Serial.print(redValue);
  Serial.print("\t green: ");
  Serial.print(greenValue);
  Serial.print("\t Blue: ");
  Serial.println(blueValue);

  // on allume la LED RVB selon les valeurs entré
  analogWrite(redLEDPin, redValue);
  analogWrite(greenLEDPin, greenValue);
  analogWrite(blueLEDPin, blueValue);
}
```

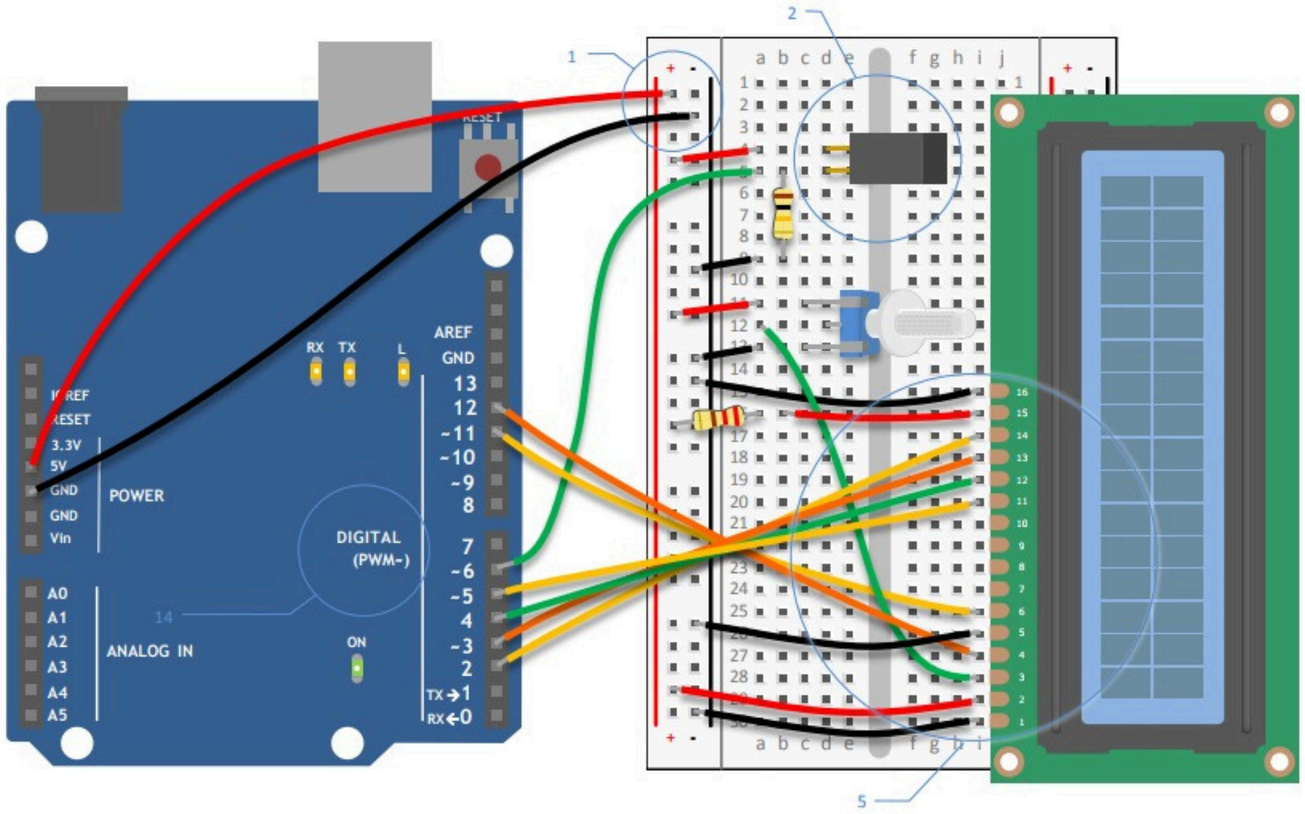
Opptionnel

Boule de cristale - page 115 du manuel

Arduino va devenir une boule de Cristal capable de prédire le futur en répondant à vos questions. Après avoir posé vos questions. Le tilt détecte les mouvements pour afficher une réponse sur l’écran.

Matériel utilisé :

- 1 Carte Arduino
- Le cable USB
- Des fils
- 1 interupteur
- 1 écran LCD
- 1 potentiomètre
- 1 résistance de 10 kilo ohms (interupteur)
- 1 résistance de 220 ohms (écran LCD)



Code de la boule de crystale partie 1

p11_CrystalBall | Arduino 1.8.9

Fichier Édition Croquis Outils Aide

p11_CrystalBall §

```
#include <LiquidCrystal.h>    // inclure la librairie pour l'écran LCD

// initialiser la bibliothèque avec les numéros des pins d'interface
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

const int switchPin = 6;     // défini le pin 6 de la carte à l'interupteur
int switchState = 0;         // attribution de la valeur de la valeur de l'interupteur
int prevSwitchState = 0;     // variable pour maintenir la valeur précédente de la broche d'interrupteur
int reply;    // une variable pour choisir quelle réponse de la boule de cristal

void setup() {
  lcd.begin(16, 2);   // créer le nombre de colone et de lignes sur l'écran LCD

  pinMode(switchPin, INPUT);     // définir l'interupteur comme entré

  lcd.print("Demandez a la");      // écrire un message sur l'écran
  lcd.setCursor(0, 1);             // mettre le curseur colone 0 et ligne 1 (comptage commençant par 0)
  lcd.print("boule magique !");    // écrire la seconde ligne
}
```

Code de la boule de crystale partie 2

```
void loop() {
  switchState = digitalRead(switchPin);   // vérification de la position de l'interupteur
  if (switchState != prevSwitchState) {   // si l'interrupteur change d'état
    // si l'état est passé de HIGH à LOW vous savez que la balle a été
    // incliné d'une direction à l'autre
    if (switchState == LOW) {
      reply = random(8);    // choisi au hasard une réponse
      lcd.clear();          // nettoie l'écran
      lcd.setCursor(0, 0);  // mettre le curseur colone 0 et ligne 1
      lcd.print("la boule dit :");    // écrire du text
      lcd.setCursor(0, 1);    //mettre le curseur colone 0 et ligne 1
      switch (reply) {    // choisir un dicton à imprimer en fonction de la valeur de la réponse
        case 0:
          lcd.print("Yes");
          break;
        case 1:
          lcd.print("Most likely");
          break;
        case 2:
          lcd.print("Certainly");
          break;
        case 3:
          lcd.print("Outlook good");
          break;
        case 4:
          lcd.print("Unsure");
          break;
        case 5:
          lcd.print("Ask again");
          break;
        case 6:
          lcd.print("Doubtful");
          break;
        case 7:
          lcd.print("No");
          break;
      }
    }
  }
  prevSwitchState = switchState;    // sauvgarder l'état actuelle
}
```

Les formations

Si vous êtes intéressés par la robotique et aimez concevoir des robots, voici quelques formations qui pourraient vous intéresser:

- Ingénieurs en robotique
- Classe préparatoire en robotique
- Dut informatique
- BTS électrotechnique, IRIS , CRSA,CPI ...)

LES SAVANTURIERS DE LA HIGH TECH

DIPLOME
DE ROBOTIQUE - NIVEAU 1

Vu le procès-verbal de l'examen établi le 24 juin 2016
par le président du jury, examinateur, ayant autorité sur les diplomes en ligne.
Le Diplome de Robotique - Niveau 1
du site www.mon-diplome.fr
est conféré à la classe de CM2B de l'école élementaire Paul Langevin

pour en jouir avec les droits et prérogatives qui y sont attachés

Signature du titulaire

Les Savanturiers de la High Tech

www.mon-diplome.fr

N° 0000000645157

Robot Arduino

Le robot Arduino permet de créer des circuits électroniques et faire du prototypage rapide. Arduino constitue une technologie de choix pour débuter la programmation d’un petit robot mobile, grâce à la carte Arduino vous contrôlez les moteurs et les capteurs.

Kit robot

Robot Arduino

Livre Arduino

Aller plus loin

Si vous souhaitez faire d'autre projets, nous vous invitons a faires les autre activités marqué dans le manuel et a consulté les sites suivant :

- Github pour les codes : https://github.com/arduino/Arduino
- Developpez.com où vous trouverez des cours sur l'Arduino avec les codes et les branchements : https://arduino.developpez.com/tutoriels/arduino-a-l-ecole/?page=projet-8-les-inputs-numeriques
- Vous pourez utilsé le site Tinkercad our faire des simulations d'arduino avec le montage et le codage (possibilité de faire du codage avec des boites style Scratch) : https://www.tinkercad.com