

## Structure d'un programme

```
void setup() {  
  // fonction exécutée à l'initialisation de la carte  
}  
  
void loop() {  
  // fonction exécutée en boucle infinie  
}
```

## Test interrupteur avec résistance pull up

```
int boutonPin = 2;
int etat;

void setup() {
  // fonction exécutée à l'initialisation de la carte

  pinMode(boutonPin, INPUT); //le pin du bouton est en entrée
  digitalWrite(boutonPin, HIGH); //active la résistance pull up

  Serial.begin(9600); // initialise la communication série à 9600 bauds
}

void loop() {
  // fonction exécutée en boucle infinie

  etat = digitalRead(boutonPin); //lit l'état du bouton poussoir
  Serial.println(etat); //envoie l'état sur la communication série
  delay(300); //attend 300 millisecondes
}
```

résultats du moniteur série (outils/moniteur série sur l'ide Arduino) :

1  
1  
1  
0  
0  
1  
0  
1  
1  
1

0 correspond au bouton appuyé et 1 au bouton relâché

## Marche arrê – version 1

```
int boutonPin = 2; //numéro de pin du bouton
int ledPin = 3; //numero de pin de la led d'état
int etatPrecedent ; //état précédent
int etat; //variable d'état du bouton poussoir
int marche = 0; //bouton marche 0 / arrêt 1

void setup() {
  // fonction exécutée à l'initialisation de la carte
  pinMode(boutonPin, INPUT);
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
  digitalWrite(boutonPin, HIGH);
  etatPrecedent = digitalRead(boutonPin);
  Serial.begin(9600);
}

void loop() {
  // fonction exécutée en boucle infinie

  etat = digitalRead(boutonPin);
  if ((etatPrecedent == 1) && (etat ==0)) { //le bouton vient d'être pressé
    marche = 1 - marche; //inversion du bouton
    digitalWrite(ledPin,marche);
    Serial.println(ledPin);
  }
  etatPrecedent = etat;
  delay(20); //attente pour éviter les phénomènes de rebond
}
```



