



Travaux Pratiques de Systèmes Embarqués

Filière: Sécurité IT et confiance numérique (S2)

TP1: Initiation à l'Arduino

Objectif du TP:

Ce TP a pour objectif d'initiation à l'arduino. Il s'agit de manipuler les entrées et les sortie d'une carte arduino en et de se familiariser avec le langage de programmation embarqué.

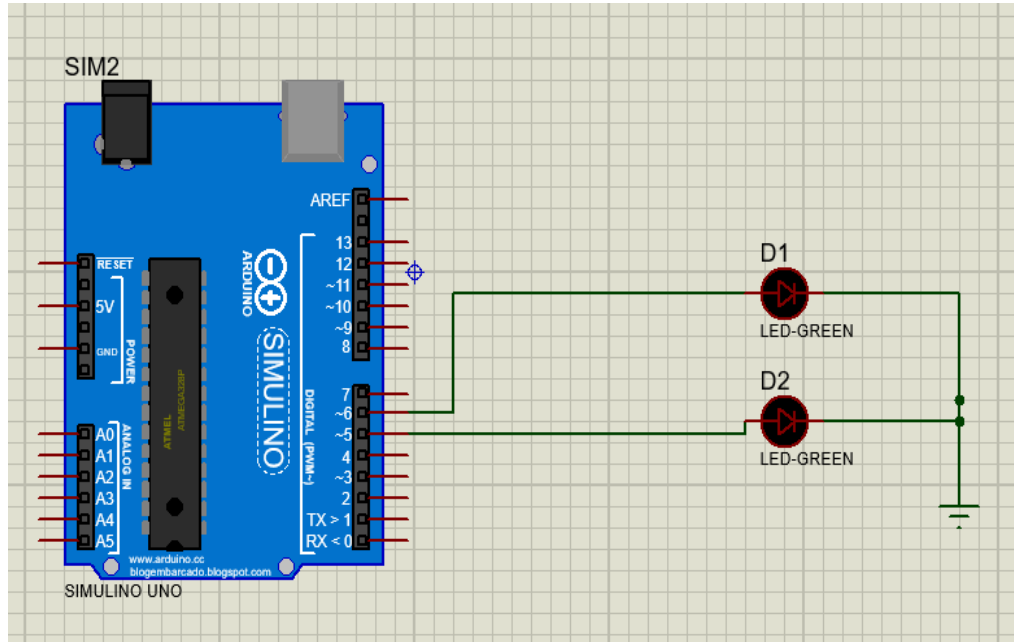
Ce TP est composé des parties suivantes:

- ❖ *Installation et utilisation du logiciel ISIS Proteuse 8*
- ❖ *Réalisation d'un chenillard de LEDs en manipulant les sorties numériques d'une carte Arduino ;*
- ❖ *Acquisition et traitement des informations issues d'un capteur analogique numérique;*
- ❖ *Acquisition des informations issues d'un capteur Ultrason*

Activité 1:

Réaliser le montage suivant sous le logiciel ISIS proteuse

Développer le code Arduino suivant, télécharger le dans la carte Arduino. Exécuter et vérifier le fonctionnement



```
#define led1 5 // led1 branchée sur la broche 2
#define led2 6 // led2 branchée sur la broche 3

void setup() // setup est déroulé une seule fois après la remise à zéro
{
    pinMode(led1, OUTPUT); // la broche led1 (2) est initialisée en sortie
    pinMode(led2, OUTPUT); // la broche led2 (3) est initialisée en sortie
}

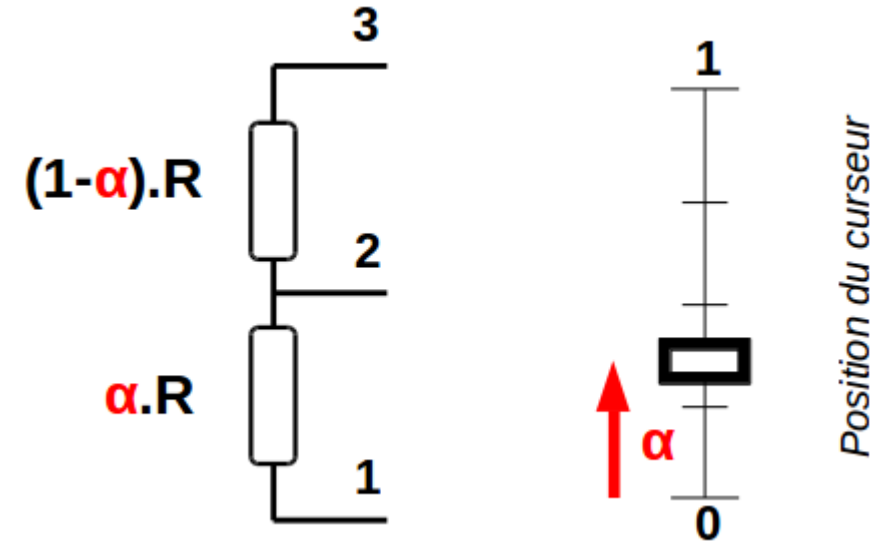
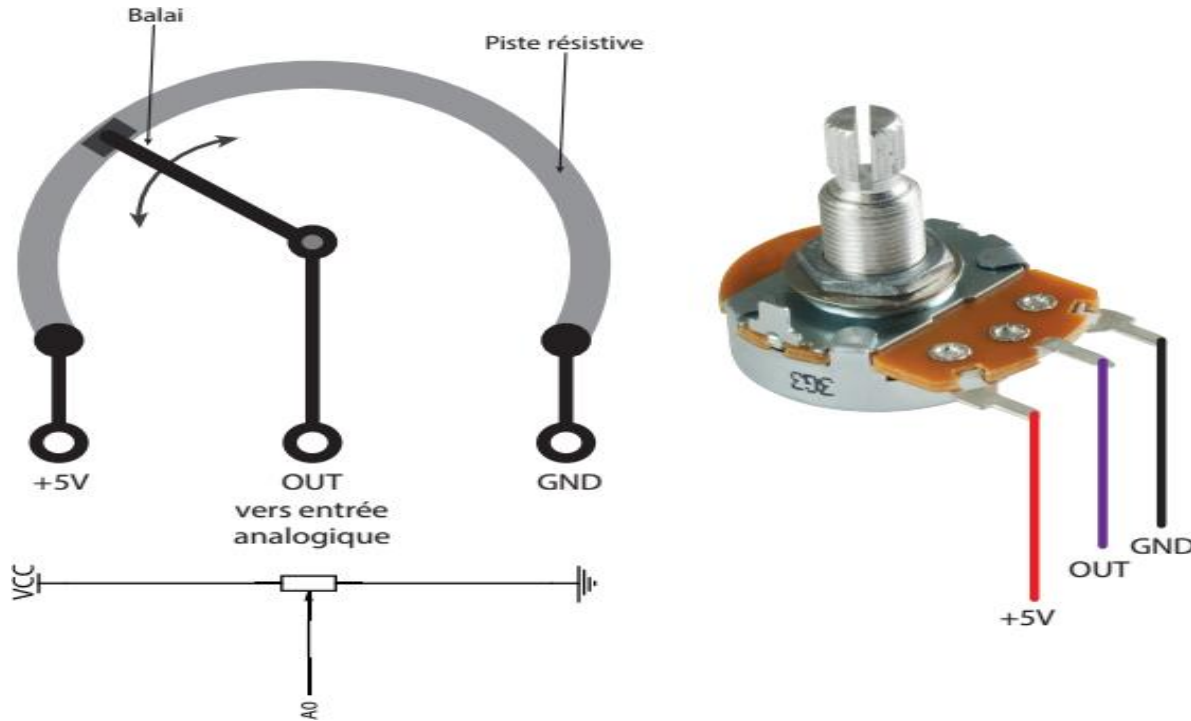
void loop() // loop est déroulé indéfiniment
{
    digitalWrite(led1, HIGH); // allume la LED1 on aurait aussi pu écrire 1 à la place de HIGH
    digitalWrite(led2, LOW); // éteint la LED2 on aurait aussi pu écrire 0 à la place de LOW
    delay(500); // attente de 1/2 seconde
    digitalWrite(led1, LOW); // éteint la LED1
    digitalWrite(led2, HIGH); // allume la LED2
    delay(500); // attente de 1/2 seconde
    digitalWrite(led1, LOW); // éteint la LED1
    digitalWrite(led2, LOW); // éteint la LED2
    delay(500); // attente de 1/2 seconde
    digitalWrite(led1, LOW); // éteint la LED1
    digitalWrite(led2, LOW); // éteint la LED2
    delay(500); // attente de 1/2 seconde
}
```

Challenge 1:

Modifier le programme et le montage de manière à réaliser un chenillard de 5 LEDS

❑ Entrée analogique

Potentiomètre : Il s'agit d'une résistance variable, il délivre sur sa borne de sortie (curseur) une tension entre 0 et VCC, il joue donc le rôle d'un capteur analogique.



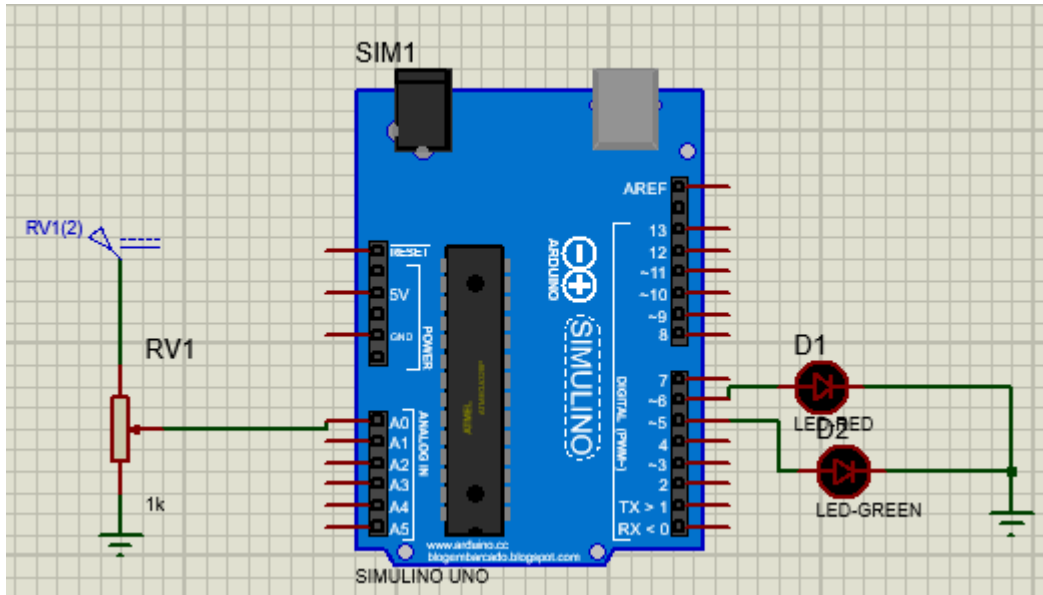
La carte arduino contient un Convertisseur analogique numérique de 10bits, en conséquence lorsqu'une valeur analogique (0 à Vcc) apparaît en entrée, le programme affiche une valeur image entre 0 et 2023, pour afficher la valeur réelle il suffit d'utiliser la règle de trois (3).

❑ Entrée analogique

Activité 2:

Réaliser le montage suivant sous le logiciel ISIS proteuse

Développer le code Arduino suivant, télécharger le dans la carte Arduino. Exécuter et vérifier le fonctionnement



```
float Pot=A0;
float val=0.0;
int L1=5;
int L2=6;
void setup() {
  pinMode(L1, OUTPUT);
  pinMode(L2, OUTPUT);
  pinMode(Pot, INPUT);
}
void loop() {
  val = analogRead(Pot); // Lire la valeur du potentiomètre
  if (val < 100) {
    digitalWrite(L1, HIGH);
    digitalWrite(L2, LOW);
  } else {
    digitalWrite(L1, LOW); // Correction de la logique pour L1
    digitalWrite(L2, HIGH); // Correction de la logique pour L2
  }
  delay(500); // Attendre 500 ms
}
```

Challenge 1:

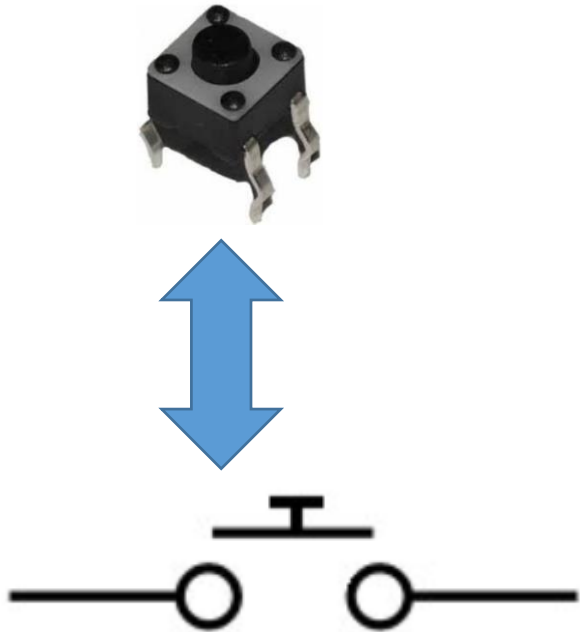
Modifier le programme et le montage pour utiliser trois LEDs. Proposer un scénario pour clignoter ces LEDs en fonction de la valeur du capteur.

Activité 3: LED et bouton poussoir

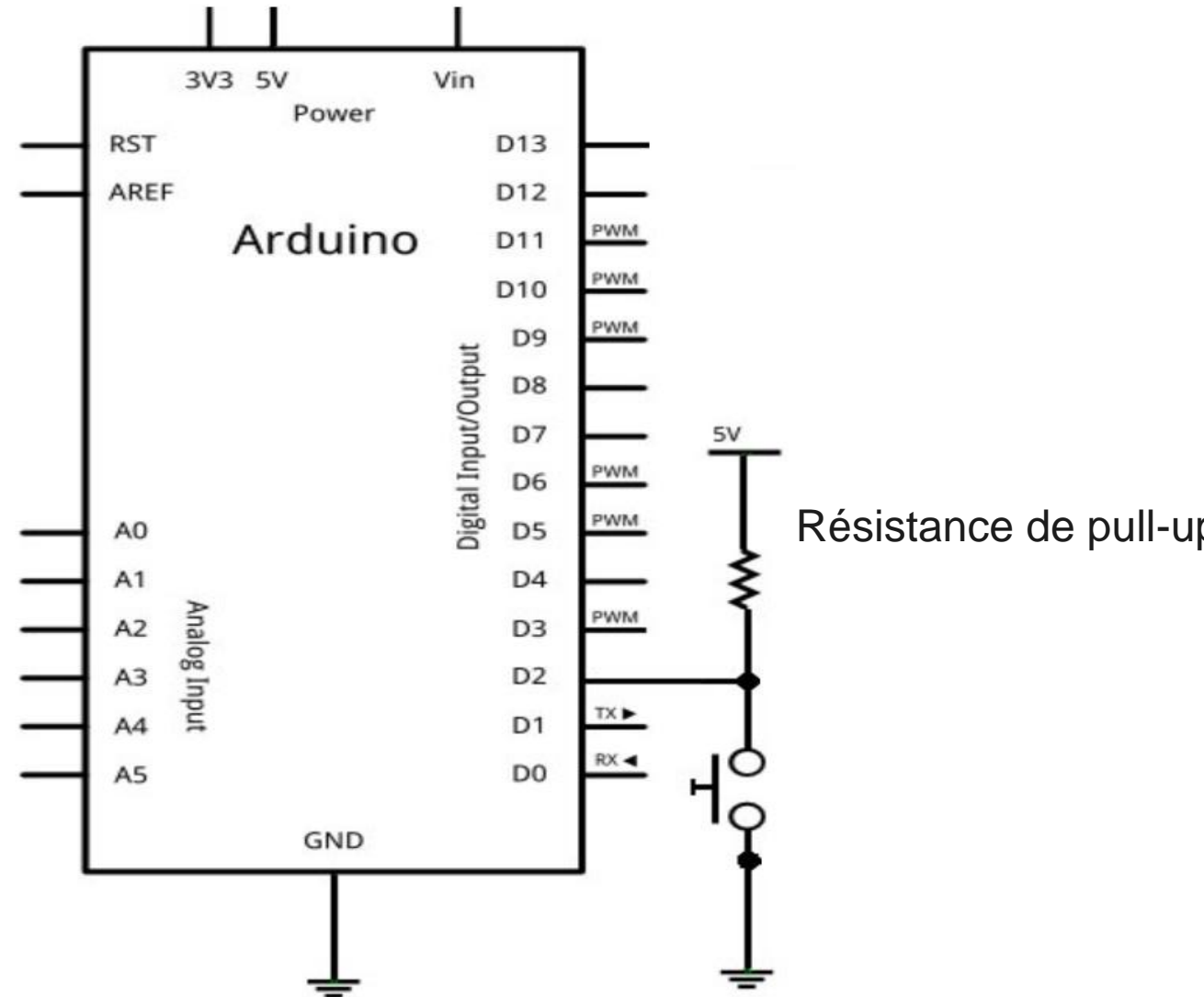
Développez un programme qui permet de contrôler l'état d'une LED à l'aide d'un bouton poussoir

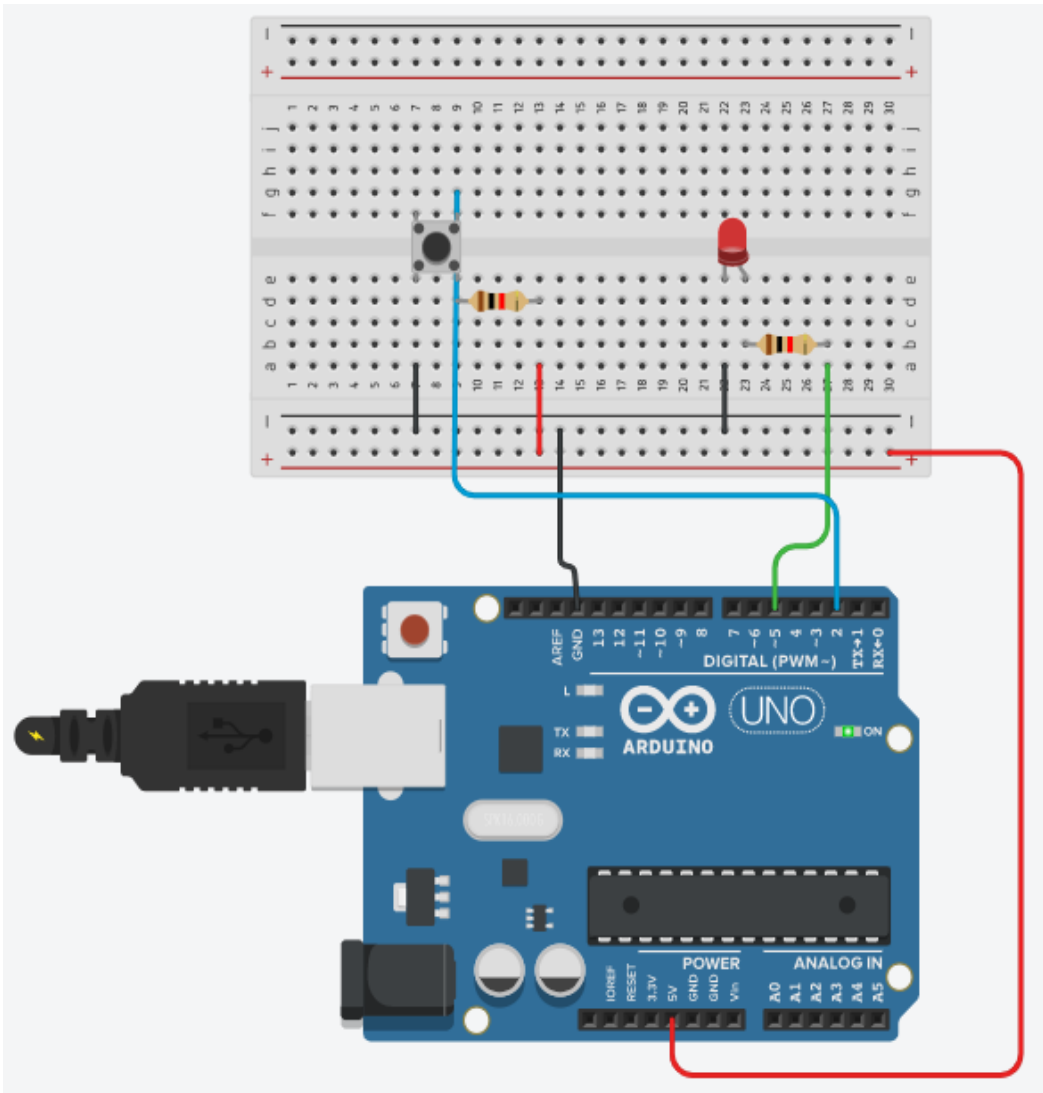
- *allumer la led si le bouton est appuyé*
- *Eteindre la led si le bouton est relâché*

Fonctionnement: bouton poussoir (BP)



Le BP est un interrupteur fermé lorsque il est appuyé et ouvert lorsque il est relâché.





```
const int boutonPin = 2; // crée un identifiant pour la broche utilisée avec le BP
const int ledPin = 5; // crée un identifiant pour la broche utilisée avec la LED
int boutonEtat = 0; // variable pour mémoriser l'état du bouton

void setup() {
    // configure la broche numérique en SORTIE
    pinMode(ledPin, OUTPUT);
    // configure la broche numérique en SORTIE
    pinMode(boutonPin, INPUT);
}

void loop() {
    // lit la valeur de l'état du bouton et la mémorise dans la variable
    boutonEtat = digitalRead(boutonPin);

    // Teste si le bouton est appuyé
    // c'est à dire si la variable boutonEtat est à 1

    if (boutonEtat == LOW) {
        // allume la LED
        digitalWrite(ledPin, HIGH);
    }
    else { // sinon
        // éteint la LED
        digitalWrite(ledPin, LOW);
    }
}
```

Challenge 3: Développer un programme qui permet d'utiliser deux boutons pour contrôler les états de deux LEDs