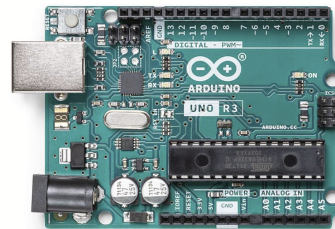
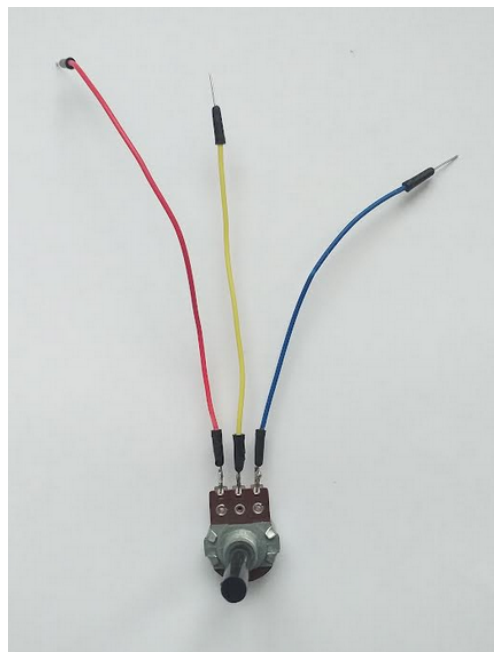


1/ Programme Arduino_v1

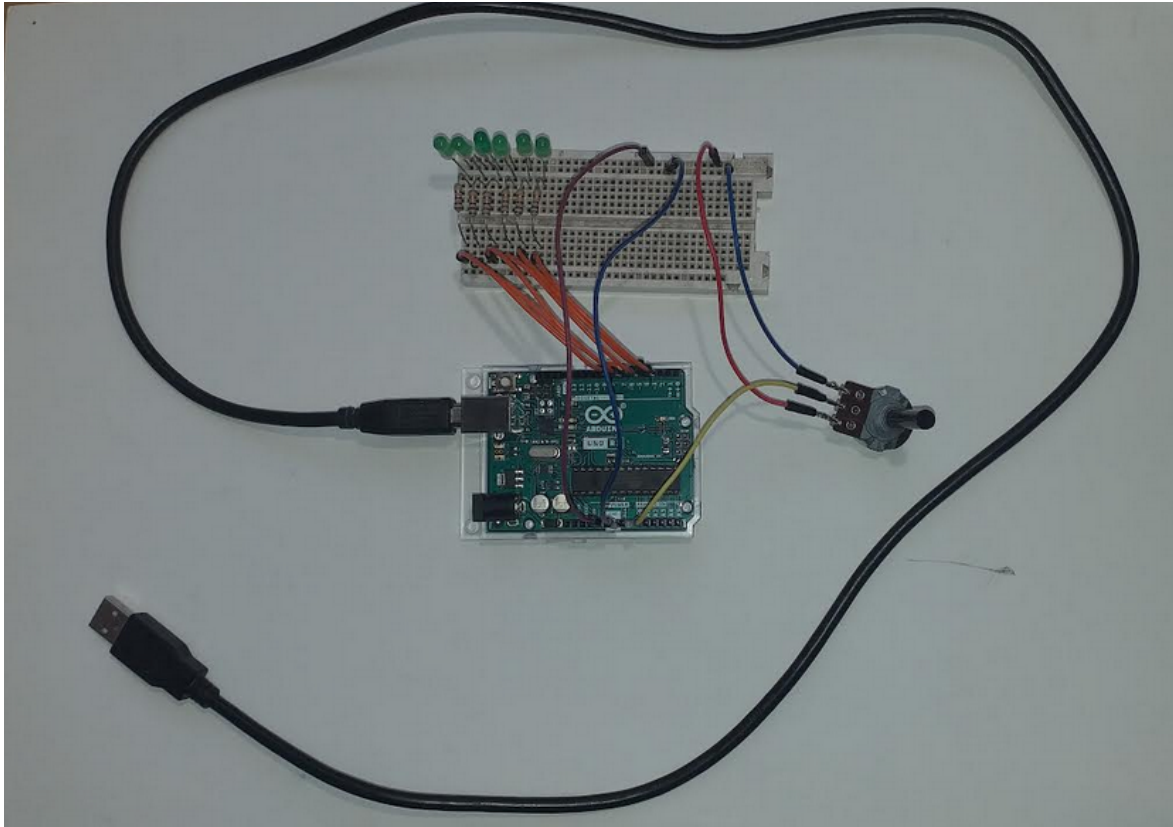
- Soudure des 3 fils du potentiomètre pour la simulation du capteur de tension
- Raccordement des 6 Leds (broches O3 à O8) et du potentiomètre (broche A0)
- Mise en oeuvre d'un programme pour l'allumage des leds en fonction de la tension des batteries
- Mise en oeuvre d'un programme pour la transmission de la valeur de la tension du capteur (broche A0) sur la liaison série depuis la carte Arduino



1.1/ Soudure des 3 fils du potentiomètre pour la simulation du capteur de tension



1.2/ Raccordement des 6 Leds (broches O3 à O8) et du potentiomètre (broche A0)



1.3/ Mise en oeuvre d'un programme pour l'allumage des leds en fonction de la tension des batteries

Hypothèse d'étude: Batterie chargée => potentiomètre régler à 5Vdc

Gemini - Prompt 1 (Google IA):

Propose moi un programme arduino qui prend en compte un potentiomètre connecté sur l'entrée A0 et 6 leds connectées sur les broches 3 à 8. Les leds s'allumeront suivant la valeur de la tension fournie par le potentiomètre (si 0V, aucune led allumée, si 5v, toute les leds sont allumées, pour les valeur de tensions intermédiaires, plus la tension est élevée, plus le nombre de leds sont allumées)

Gemini - Réponse 1:

```
int ledPins[] = {3, 4, 5, 6, 7, 8}; // Tableau des broches des LEDs

int potPin = A0; // Broche du potentiomètre

void setup() { // Déclaration des variables

    for (int i = 0; i < 6; i++) {

        pinMode(ledPins[i], OUTPUT); // Définir les broches des LEDs comme sorties

    }

}

void loop() { // Programme principal

    int sensorValue = analogRead(potPin); // Lire la valeur du potentiomètre

    int ledCount = map(sensorValue, 0, 1023, 0, 5); // Déterminer le nombre de LEDs à allumer

    for (int i = 0; i < 6; i++) {

        if (i <= ledCount) {

            digitalWrite(ledPins[i], HIGH); // Allumer la LED

        } else {

            digitalWrite(ledPins[i], LOW); // Eteindre la LED

        }

    }

}
```

1.4/ Mise en oeuvre d'un programme pour la transmission de la valeur de la tension du capteur (broche A0) sur la liaison série depuis la carte Arduino

Gemini - Prompt 2 (Google IA):

Propose moi une modification du programme pour prendre en compte un module bluetooth connecté aux broches 10 à 13 et pour que la valeur de la tension du potentiomètre soit aussi envoyée sur le port série.

Gemini - Réponse 2:

```
#include <SoftwareSerial.h> // Bibliothèque pour la communication série logicielle
```

```
int ledPins[] = {3, 4, 5, 6, 7, 8}; // Tableau des broches des LEDs
```

```
int potPin = A0; // Broche du potentiomètre
```

```
int rxPin = 10; // Broche RX du module Bluetooth
```

```
int txPin = 11; // Broche TX du module Bluetooth
```

```
SoftwareSerial bluetoothSerial(rxPin, txPin); // Objet SoftwareSerial pour la communication Bluetooth
```

```
void setup() {
```

```
    for (int i = 0; i < 6; i++) {
```

```
        pinMode(ledPins[i], OUTPUT); // Définir les broches des LEDs comme sorties
```

```
    }
```

```
bluetoothSerial.begin(9600); // Définir le débit en bauds pour la communication Bluetooth

Serial.begin(9600); // Définir le débit en bauds pour la communication série

}

void loop() {

    int sensorValue = analogRead(potPin); // Lire la valeur du potentiomètre

    int ledCount = map(sensorValue, 0, 1023, 0, 5); // Déterminer le nombre de LEDs à allumer

    for (int i = 0; i < 6; i++) {

        if (i <= ledCount) {

            digitalWrite(ledPins[i], HIGH); // Allumer la LED

        } else {

            digitalWrite(ledPins[i], LOW); // Eteindre la LED

        }

    }

    // Envoyer la valeur du potentiomètre sur le port série et Bluetooth

    bluetoothSerial.println(sensorValue);

    Serial.println(sensorValue);

}
```