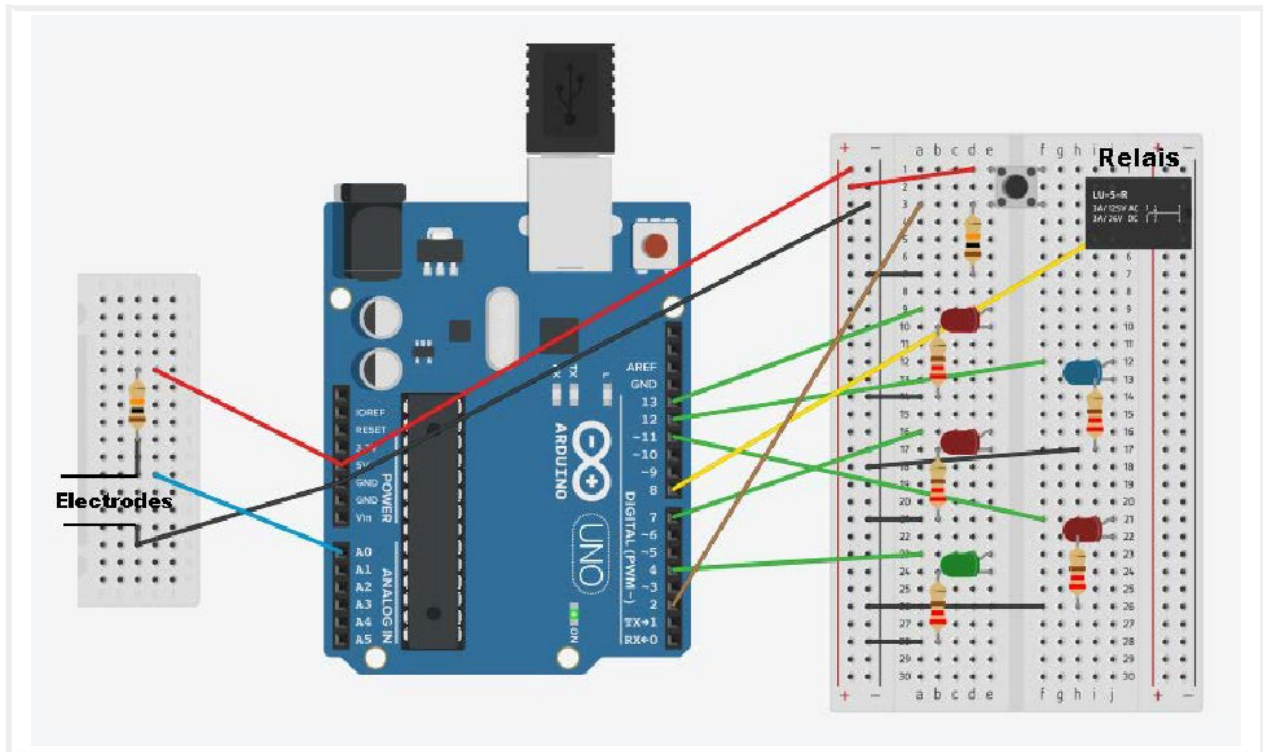


Détecteur de niveau

(Contrôle de l'alimentation électrique d'une électrovanne)



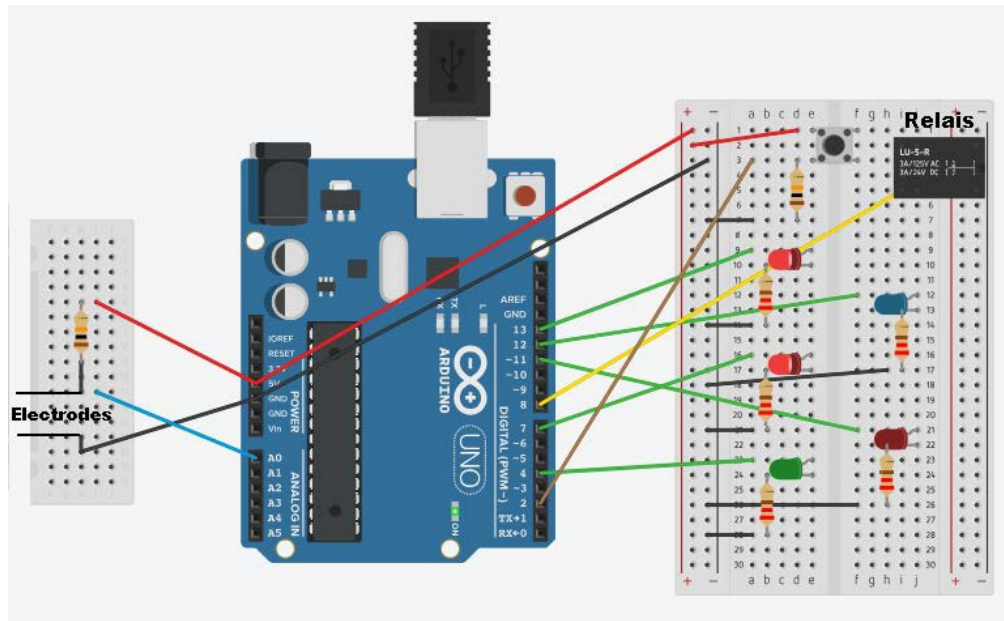
Liste des composants

- . 3 DELs rouges
- . 1 DEL verte
- . 1 DEL bleue
- . 5 résistances de 220 Ω
- . 2 résistances de 10 k Ω
- . 1 Relais SPST
- . 2 plaques d'essais
- . Fils de connexion

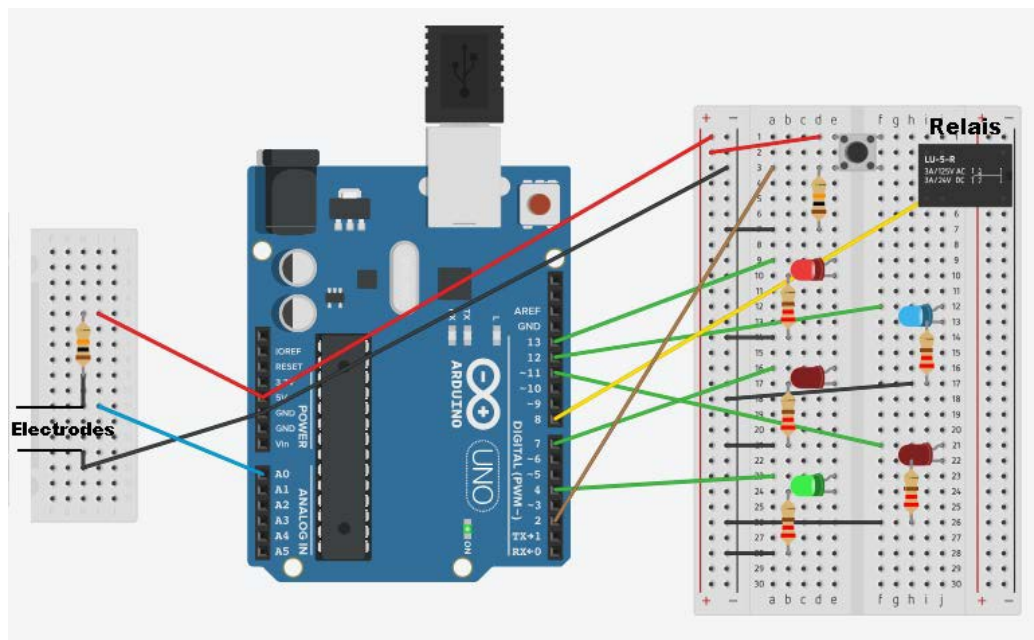
Objectif

L'objectif du montage est de contrôler l'alimentation électrique d'une électrovanne alimentant en eau un dispositif de désionisation, par l'intermédiaire d'un relais en fonction du niveau d'eau détecté par deux électrodes dans une bonbonne d'eau désionisée en sortie du dispositif.

Après avoir appuyé sur le bouton-poussoir (DEL rouge reliée à la borne 13 allumée), si les électrodes ne sont pas en contact avec l'eau, le relais est actif, l'électrovanne est alimentée et est donc ouverte (DEL rouge reliée à la borne 7 allumée).



Quand les électrodes sont en contact avec l'eau, l'alimentation électrique de l'électrovanne est coupée, celle-ci est donc fermée (DEL rouge reliée à la borne 7 éteinte et DEL verte reliée à la borne 4 allumée).



Si la conductivité de l'eau en contact avec les électrodes est bonne, la DEL bleue reliée à la borne 12 s'allume, sinon la DEL rouge reliée à la borne 11 s'allume.

. Le programme

Voici le code de l'activité :

DetecteurNiveau

```
const int switchPin = 2;
const int LedVanneOFFPin = 4;
const int LedVanneONPin=7;
const int LedGoodPin=12;
const int LedBadPin=11;
const int LedOnOffPin=13;
int const PotPin=A0;
int const RelayPin=8;

int PotVal=0;
int switchState = 0;
int prevSwitchState = 0;
int state=0;

void setup() {

    pinMode(LedVanneOFFPin,OUTPUT);
    pinMode(LedVanneONPin,OUTPUT);
    pinMode(LedOnOffPin,OUTPUT);
    pinMode(LedGoodPin,OUTPUT);
    pinMode(LedBadPin,OUTPUT);
    pinMode(switchPin,INPUT);

    Serial.begin(9600);

}

void loop() {

switchState = digitalRead(switchPin);

if ((switchState == HIGH)&&(prevSwitchState == LOW)) {
    state=1-state;
    delay(10);
}
else {
if ((switchState == LOW)&&(prevSwitchState == HIGH)) {
    delay(10);
}
}
prevSwitchState = switchState;
```

```

if (state==1) {

    digitalWrite(LedOnOffPin, HIGH);
    PotVal=analogRead(PotPin);
    Serial.println(PotVal);

    if (PotVal < 1015) {

        digitalWrite(LedVanneONPin, LOW);
        digitalWrite(LedVanneOFFPin, HIGH);
        digitalWrite(RelayPin, LOW);

        if (PotVal > 800) {
            digitalWrite(LedGoodPin, HIGH);
            digitalWrite(LedBadPin, LOW);
        }
        else {
            digitalWrite(LedGoodPin, LOW);
            digitalWrite(LedBadPin, HIGH);
        }
    }
    else {

        digitalWrite(LedVanneONPin, HIGH);
        digitalWrite(LedVanneOFFPin, LOW);
        digitalWrite(LedGoodPin, LOW);
        digitalWrite(RelayPin, HIGH);

    }

    delay(1000);
}

else {
    digitalWrite(LedOnOffPin, LOW);
    digitalWrite(RelayPin, LOW);
    digitalWrite(LedVanneONPin, LOW);
    digitalWrite(LedVanneOFFPin, LOW);
    digitalWrite(LedGoodPin, LOW);
    digitalWrite(LedBadPin, LOW);
}
}

```

Déroulement du programme :

– 1. Déclaration des constantes et variables :

- . const int switchPin = 2 (constante nombre entier correspondant à la broche du bouton poussoir)
- . const int LedVanneOFFPin = 4 (constante nombre entier correspondant à la broche de la DEL verte indiquant que l'électrovanne est fermée)
- . const int LedVanneONPin=7 (constante nombre entier correspondant à la broche de la DEL rouge indiquant que l'électrovanne est ouverte)
- . const int LedGoodPin=12 (constante nombre entier correspondant à la broche de la DEL bleue indiquant que la conductivité de l'eau est bonne)
- . const int LedBadPin=11 (constante nombre entier correspondant à la broche de la DEL rouge indiquant que la conductivité de l'eau n'est pas bonne)
- . const int LedOnOffPin=13 (constante nombre entier correspondant à la broche de la DEL rouge indiquant que le détecteur de niveau fonctionne)
- . int const PotPin=A0 (constante nombre entier correspondant à la broche de l'électrode de mesure)
- . int const RelayPin=8 (constante nombre entier correspondant à la broche de commande du relais)
- . int PotVal=0 (variable nombre entier pour stocker la valeur du potentiel de l'électrode de mesure)
- . int switchState = 0 (variable nombre entier pour stocker la valeur du potentiel de la broche du bouton poussoir)
- . int prevSwitchState = 0 (variable nombre entier pour stocker l'ancienne valeur du potentiel de la broche du bouton poussoir)
- . int state=0 (variable nombre entier correspondant à l'action à effectuer)

– 2. Initialisation des entrées et sorties :

- . **Initialisation des entrées et des sorties (DELs, bouton poussoirs)**
- . **Initialisation de la liaison série à un débit de 9600 bauds**

– 3. Fonction principale en boucle :

- > **Lecture de la valeur de la broche du bouton poussoir,**
- > **Mise à jour de la variable « state » si changement de la valeur du bouton poussoir,**
- > **Mise en route ou arrêt du dispositif en fonction de la variable « state »,**

--> Lecture du potentiel de l'électrode (affichage de la valeur dans le moniteur série),

--> Fermeture ou ouverture de la vanne d'alimentation en eau en fonction de la valeur du potentiel de l'électrode, puis indication sur la qualité de la conductivité de l'eau.