



Module SCI

Ecole Nationale Supérieure d'Informatique d'Alger (Ex. ini)

Compte rendu TP2

10 Mars 2022

Introduction

Dans ce TP nous nous intéressons aux capteurs, explorant principalement deux en particulier (le bouton poussoir et le capteur de lumière), ainsi que les circuits adéquats pour les faire fonctionner.

Nous explorons de même le code derrière cette manipulation, les instructions et les paramètres réglés.

Objectifs

1. **Partie théorique :** Exploration des capteurs, circuits, branchements sur le breadboard et le code exécutable par le contrôleur.
2. **Partie pratique :** Schémas des circuits, résultats et commentaires.

Partie théorique

B. Exploration des capteurs utilisés :

1. Le capteur de lumière :

Un capteur de lumière (photorésistance) de la lumière est un composant sensible à la lumière et un capteur passif. Lorsque la lumière tombe dessus, la résistance change. La valeur de la résistance diminue quand le niveau de lumière augmente. Cette propriété de sensibilité est dû à l'utilisation des semiconducteurs

2. Le bouton poussoir :

Un capteur numérique à deux états ON/OFF. Il n'est pas un système mécanique parfait, du coup lorsqu'un appui (passer à ON) est effectué il rebondit plusieurs fois avant de se stabiliser à 5V (des signaux parasites), pour y remédier :

- Solution électronique : on utilise une résistance PULL-UP (existe dans la carte) ou une résistance externe qui a une grande valeur
- Solution software : on remarque que les effets parasites ne durent que certaines nanosecondes donc, on peut rajouter un petit délai avant de changer l'état de la LED.

Donc, oui la configuration anti-rebond est nécessaire pour une stabilité, lors de la pratique on peut l'apercevoir en affichant les valeurs de voltage renvoyées à chaque instant par le bouton, car visuellement ça se produit très rapidement.

C/D. Circuits de branchements :

Les composants nécessaires:

- 2 résistances 220 ohm + résistance PULL-UP (dans la carte)
- 1 bouton poussoir + 1 capteur de lumière
- câbles de liaisons

Remarque: pour le deuxième schéma (celui du bouton poussoir), on utilise directement la résistance pull-up du MCU en changeant le mode de la broche avec l'instruction :

`pinMode(pin, INPUT_PULLUP)`

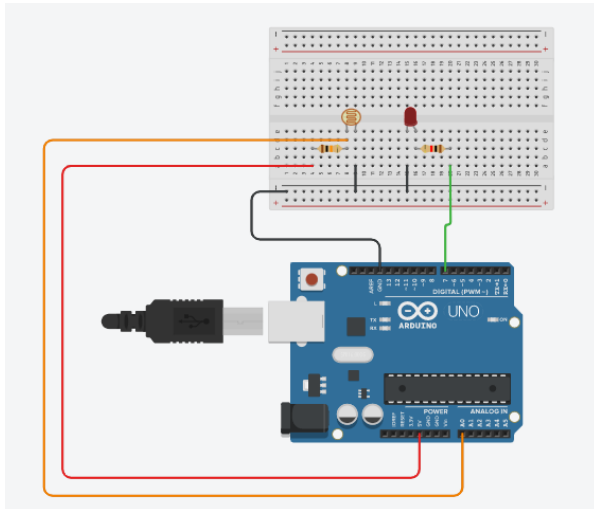


Figure 1 : circuit pour capteur de lumière

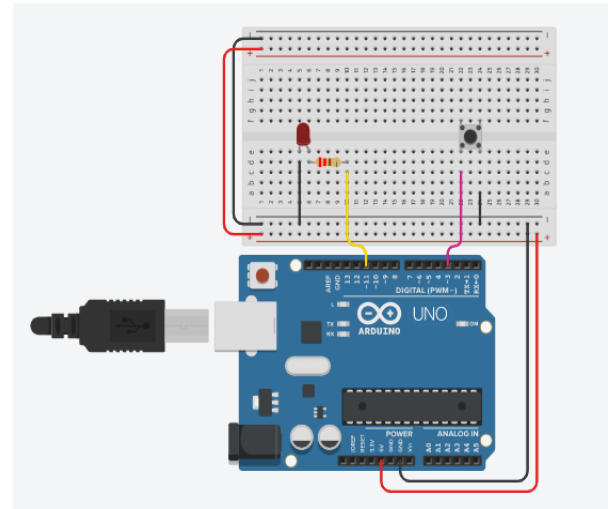


Figure 2 : circuit pour bouton poussoir

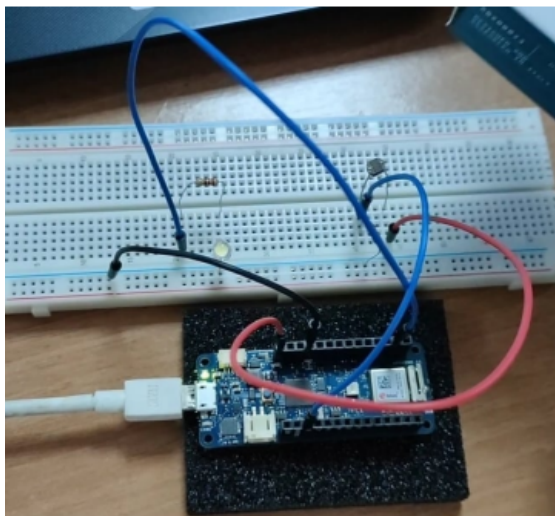


Figure 3 : branchement pratique du circuit 1

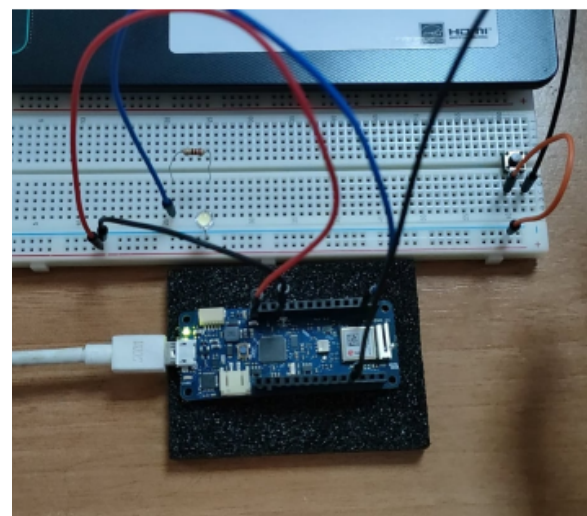


Figure 4 : branchement du circuit 2

Partie pratique:

A. Branchement d'un capteur de lumière (Analogique) :

1. Configuration d'une ligne GPIO (General Purpose I/O) et affichage de la valeur :



```

sketch_apr01a$

const int ldrPin = A0;

void setup() {
  Serial.begin(9600); //Ouvrir un port pour l'affichage

  pinMode(ldrPin, INPUT); //Configurer l'entrée depuis le capteur
}

void loop() {
  int ldrStatus = analogRead(ldrPin); //Lire la valeur captée
  Serial.print("L'intensité lumineuse est = ", ldrStatus);
}

```

- Pour afficher la valeur retournée par l'entrée analogique, d'abord ouvrir un port avec l'instruction `Serial.begin(9600)` et régler la pin du LDR(Light Dependant Résistance) le capteur de lumière en entrée `INPUT`.

- `analogRead(ldrPin)` permet de récupérer cette valeur dans la variable `ldrStatus`.

Figure 5: code lecture de valeur captée par la photorésistance

2. Transformer l'entrée analogique en numérique (commander une LED) :



```

sketch_apr01a$

const int ledPin = 13;
const int ldrPin = A0;

void setup() {
  Serial.begin(9600); //Ouvrir un port pour l'affichage
  pinMode(ledPin, OUTPUT); //Régler la LED en sortie
  pinMode(ldrPin, INPUT); //Régler le LDR en entrée
}

void loop() {
  int ldrStatus = analogRead(ldrPin); //récupérer la valeur
  if (ldrStatus <= 200) { //Si <= seuil
    digitalWrite(ledPin, HIGH);
    Serial.print("Il fait noir, LED allumée !");
    Serial.println(ldrStatus);
  } else {
    digitalWrite(ledPin, LOW);
    Serial.print("Assez de lumière, éteindre la LED !");
    Serial.println(ldrStatus);
  }
}

```

Pour transformer le signal analogique reçu, en commande d'allumage numérique de la LED, on se sert d'un seuil fixé à 200 par exemple. Quand la valeur captée dépasse celle-ci, on éteint la LED, sinon on l'allume.

Figure 6: code pour commander une LED

Résultats :

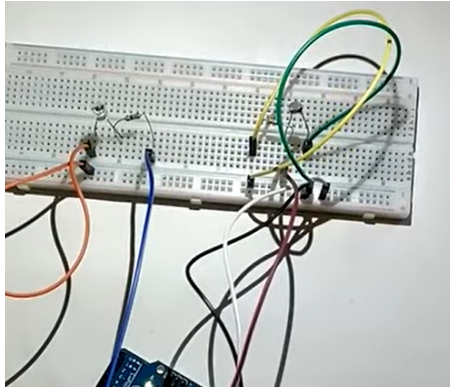


Figure 7 : LED éteinte en présence de la lumière

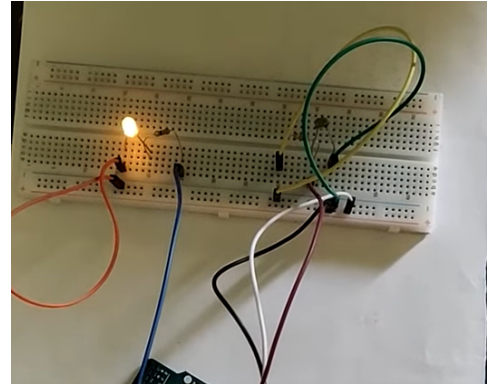


Figure 8 : LED allumée en absence de lumière

3. Application à la commande automatique des lampadaires :

- Les systèmes d'éclairage public actuels se basent sur des périodes d'allumage fixes sans prendre en compte l'influence des saisons de l'année et les variations météorologiques au long de la journée, ce qui mène à des pertes journalières d'électricité.
- Une solution serait de mettre en place un ou plusieurs détecteurs de luminosité pour un ensemble de lampadaires appartenant à la même région pour commander leur allumage

B. Branchement d'un bouton poussoir (Numérique) :



Figure 9 : Résultat du bouton poussoir