Start the program

Setup env

```
pip3 install flask
export FLASK_APP=webserver
export FLASK_ENV=development
```

Run the web server

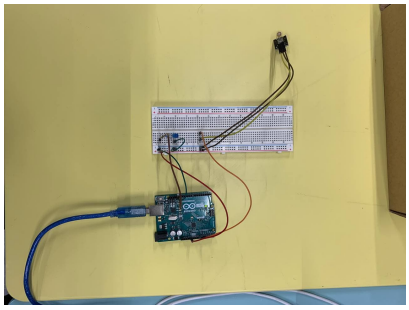
```
flask run
```

Instructions

1 / Capteur sur l'arduino

- Branchement
- Récupération de la valeur toutes les 500ms

Tout d'abord, on effectue les branchements sur notre Arduino pour communiquer entre les 2 périphériques



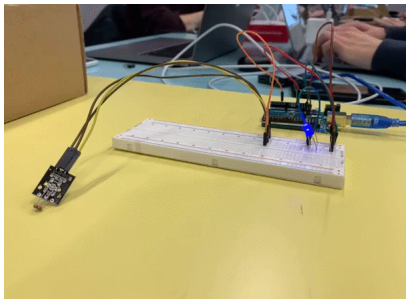
Ensuite, on vérifie que l'on récupère bien les données du capteur

```
09:04:59.775 -> { "lumen" : 202}
09:05:00.283 -> { "lumen" : 202}
09:05:00.758 -> { "lumen" : 205}
09:05:01.297 -> { "lumen" : 204}
09:05:01.766 -> { "lumen" : 202}
09:05:02.267 -> { "lumen" : 202}
09:05:02.788 -> { "lumen" : 202}
09:05:03.289 -> { "lumen" : 203}
09:05:03.773 -> { "lumen" : 202}
09:05:04.265 -> { "lumen" : 203}
09:05:04.770 -> { "lumen" : 203}
09:05:05.271 -> { "lumen" : 203}
```

---

2 / Allumer une LED à chaque mesure pendant 200ms

Une fois les branchements réalisés, on fait clignoter la led à intervalle régulier

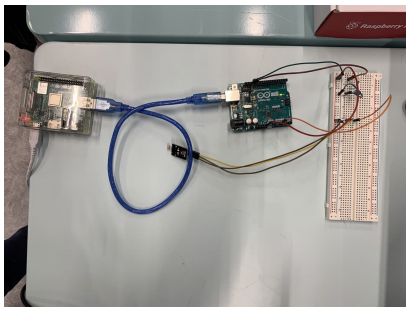


---

3 / Connecter un Raspberry à l'arduino

- Avec un cable USB
- ou par les GPIO
- Transmettre les valeurs au Raspberry

On connecte ensuite le raspberry à l'arduino pour qu'il récupère les données



---

4 / Ecrire un programme dans le langage de votre choix qui lira la valeur du capteur

Pour récupérer ces données, on écrit on programme simple sur l'IDE Arduino

```
#define ledPin 13
#define lumPin A1

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(lumPin, INPUT);
  pinMode(ledPin, OUTPUT);
}

int val = HIGH;

void loop() {
  String json1 = "{ \"lumen\" : ";
  String json4 = json1 + analogRead(lumPin);
  Serial.println(json4 + "}");

  digitalWrite(ledPin, val);
  delay(200);
  digitalWrite(ledPin, !val);

  delay(300);
}

void toggle_val(){
  val = !val;
}
```

---

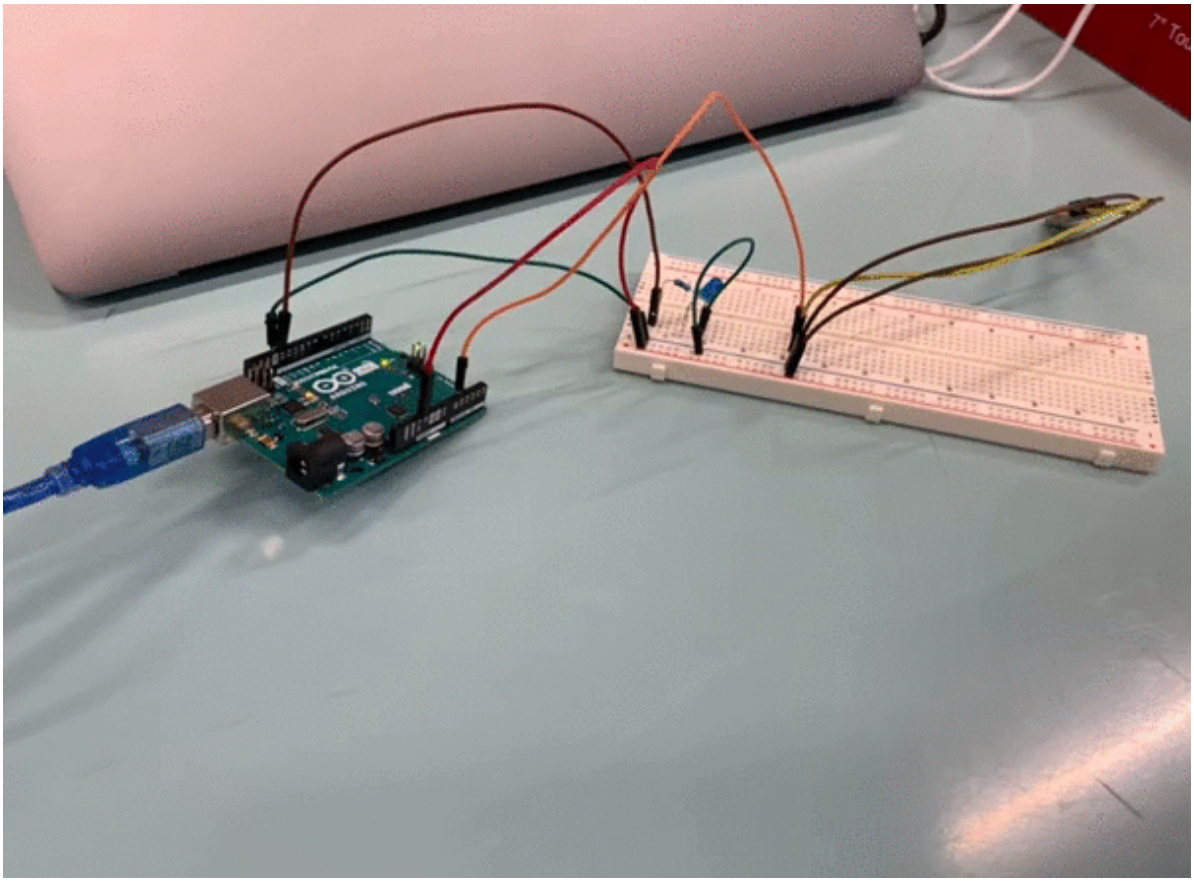
5 / Bonus: Ecrire un serveur Web qui expose la valeur sur une API (langage de votre choix)

Pour le serveur Web, nous avons décidé d'utiliser Flask, qui est un framework Python qui nous permet ici de développer une API et une interface web à partir d'un projet

```
pi@raspberrypi:~/IOT-TP $ wget 127.0.0.1:5000/temperature
--2021-03-26 09:28:51--  http://127.0.0.1:5000/temperature
Connecting to 127.0.0.1:5000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4 [application/json]
Saving to: ‘temperature.1’

temperature.1               100%[=================================================>]       4  --.-KB/s    in 0s

2021-03-26 09:28:51 (75.2 KB/s) - ‘temperature.1’ saved [4/4]

pi@raspberrypi:~/IOT-TP $ cat temperature.1
299
```

6 / Bonus: Permettre via l'API du Raspberry, d'inverser le clignotement de la LED (reste allumée en permanence mais s'éteint 200ms à chaque mesure)

On crée une route pour modifier le clignotement de la led.



Contributors

- @matteolecuit
- @PaulLereverend
- @HugoHUET
- @geoffrey-max
- @adrienvaucard