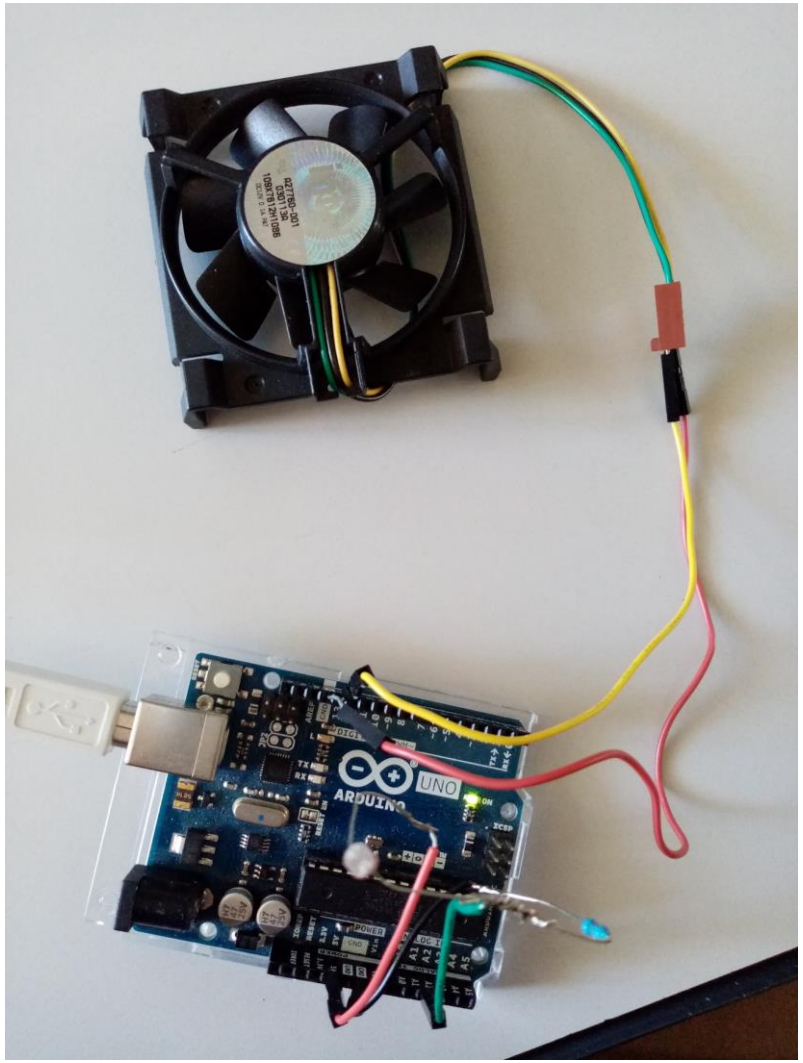
TP Arduino

Julien Zoubian & Pierre Barrillon



Arduino IDE

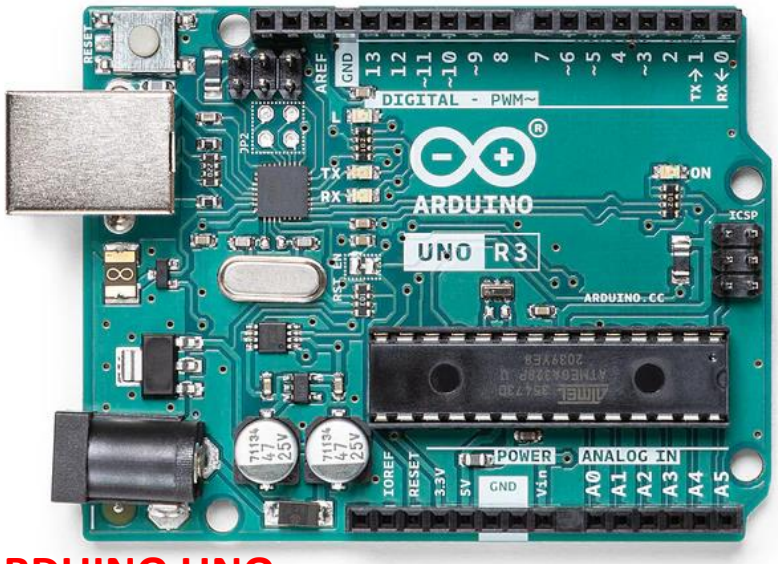
Node-RED

Introduction

- **But**:
  - Utilisation d'un arduino UNO pour récupérer les données mesurées par différents capteurs et/ou piloter des composants.
  - Comprendre le fonctionnement de l'ARDUINO et sa programmation
  - Construire son système avec les éléments disponibles
  - *Développer un interface graphique dans Node-RED* (on verra si on a le temps, mais probablement pas…)

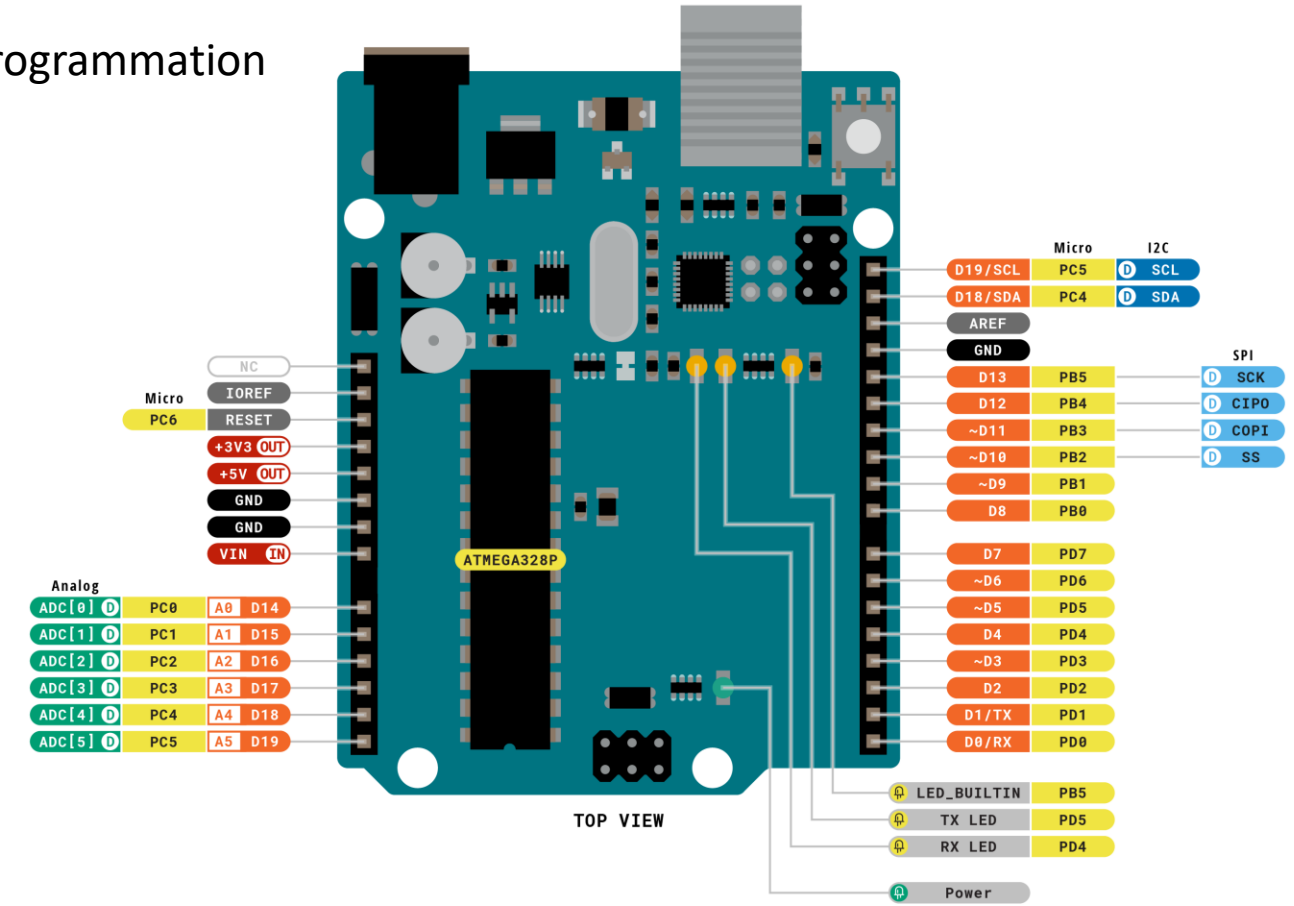
Introduction

- **But**:
  - Utilisation d'un arduino UNO pour récupérer les données mesurées par différents capteurs et/ou piloter des composants.
  - Comprendre le fonctionnement de l'ARDUINO et sa programmation
  - Construire son système avec les éléments disponibles



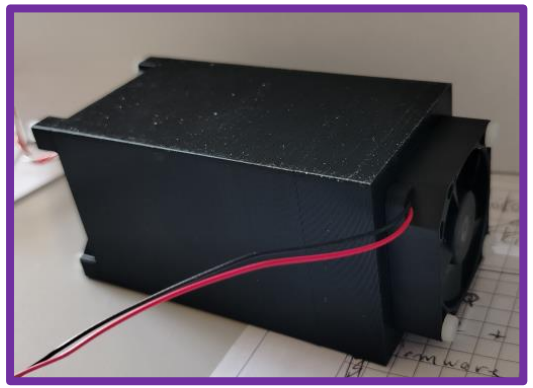
ARDUINO UNO

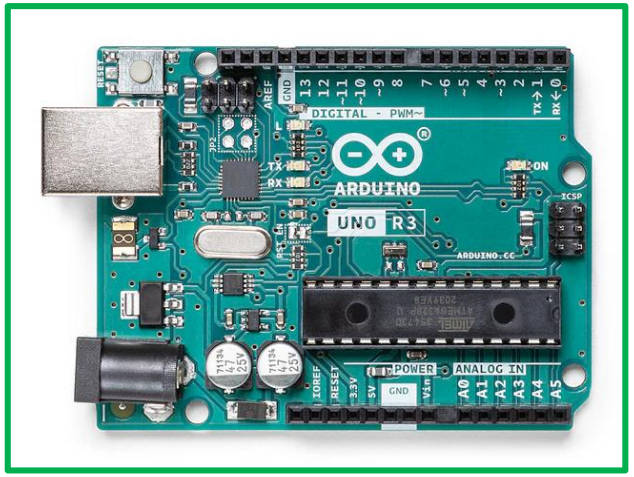
- Carte microcontrôleur ATmega328P.
- 14 pins I/O digital, 6 pins analog Input
- Connection USB
- Produit 5V et 3.3V
- Communications I2C et SPI disponibles

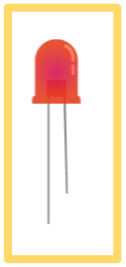


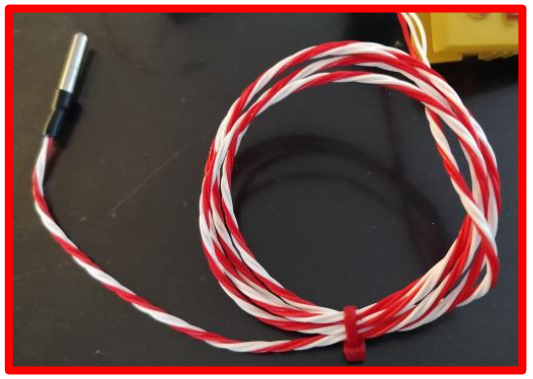
Les jouets à disposition

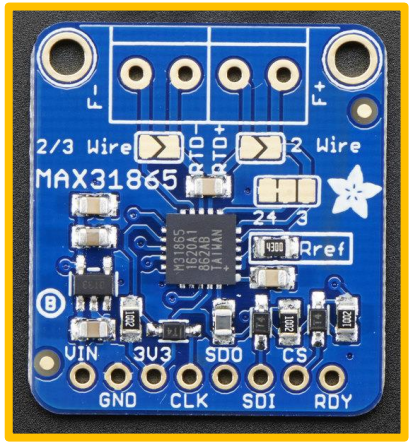
- Arduino UNO
- Câble USB → alimentation et communication avec l'arduino
- PT100 « 4 fils » → mesure de température
- Carte MAX 31865 (lecture PT100)
- Photo-resistor + résitance de 1kOhm → détection lumière
- Ventilateur 5V
- Capteur de flux d'air
- DHT22 → humidité et t°
- HM1500 → humidité
- Buzzer → son/alarme
- LED → lumière/alarme
- Switch → interrupteur
- Câbles
- Structure mécanique

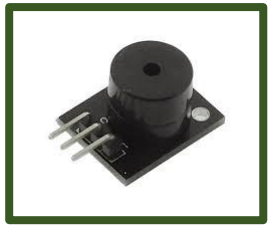


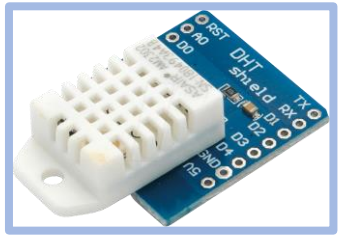


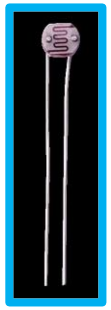


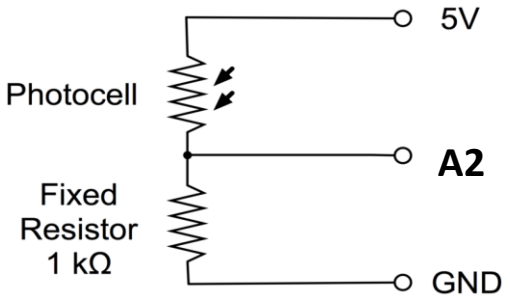






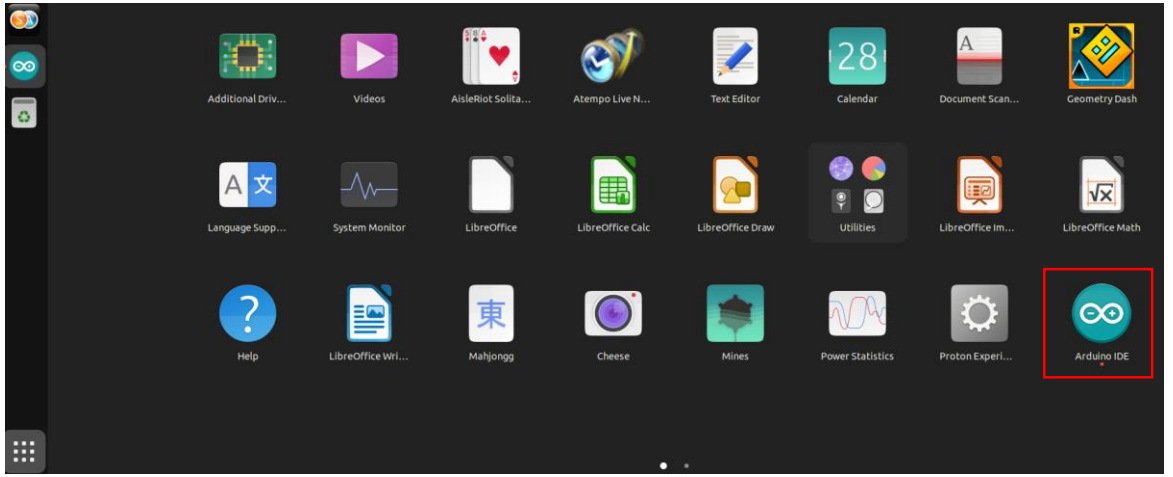


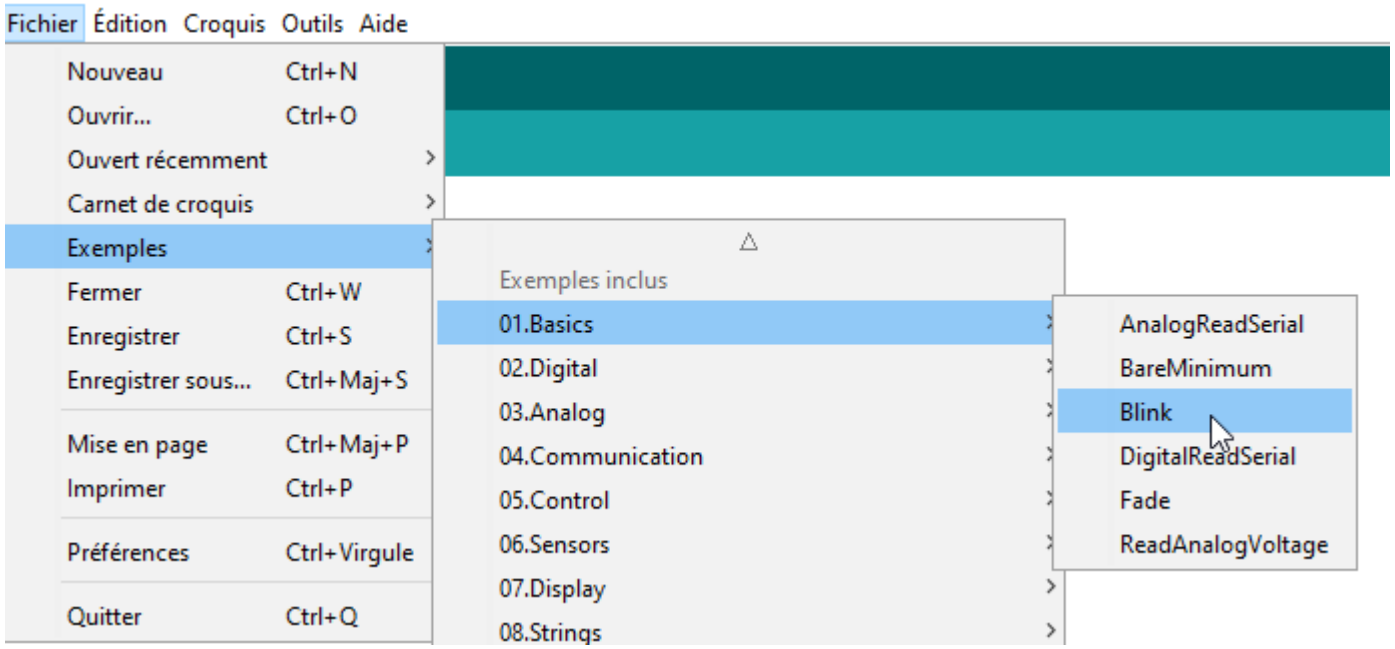




Prise en main

- Chaque binôme récupère un arduino UNO et un câble USB
- Allumage PC. Demandez le mdp.
- Branchement ARDUINO au PC
- Ouverture de l'IDE ARDUINO
- Recherchez l'exemple « blink » et ouvrez le code correspondant





- Les codes ARDUINO sont divisés en 2 parties principales:
  - **Setup**

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}
```

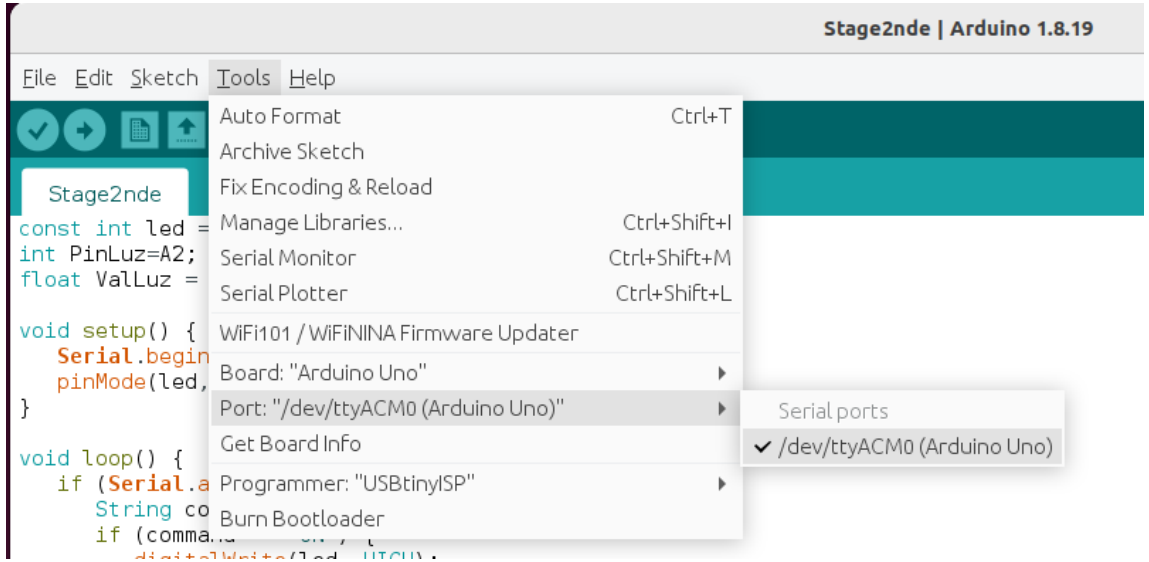
  - **Loop**

```
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

- Dans la plupart des programmes, on a aussi une partie « appel de librairie » et une autre « déclaration ».

Prise en main

- Chaque binôme récupère un arduino UNO et un câble USB
- Allumage PC. Demandez le mdp.
- Branchement ARDUINO au PC
- Ouverture de l'IDE ARDUINO
- Recherchez l'exemple « blink » et ouvrez le code correspondant



- Vérifiez le programme
- Téléversez le programme (sélectionnez le bon « port »).

Prise en main

- Chaque binôme récupère un arduino UNO et un câble USB
- Allumage PC. Demandez le mdp.
- Branchement ARDUINO au PC
- Ouverture de l'IDE ARDUINO
- Recherchez l'exemple « blink » et ouvrez le code correspondant

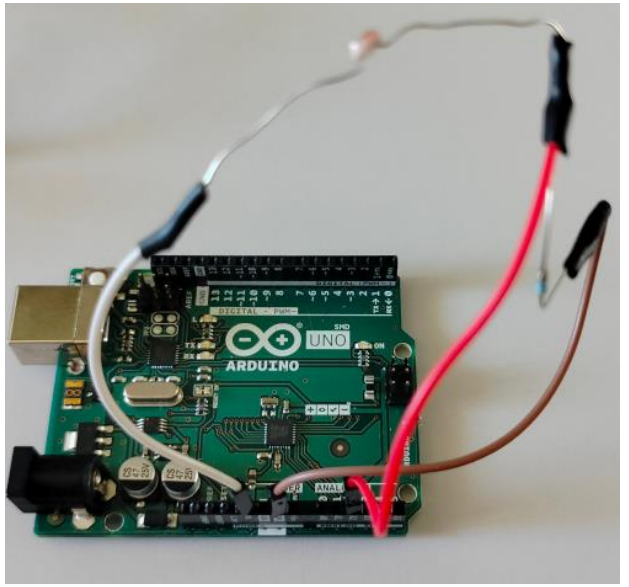
La petite LED sur l'ARDUINO devrait clignoter.
Vous pouvez rajouter une LED plus grosse entre les pins 13 (+: patte longue) et GND (-: patte courte)

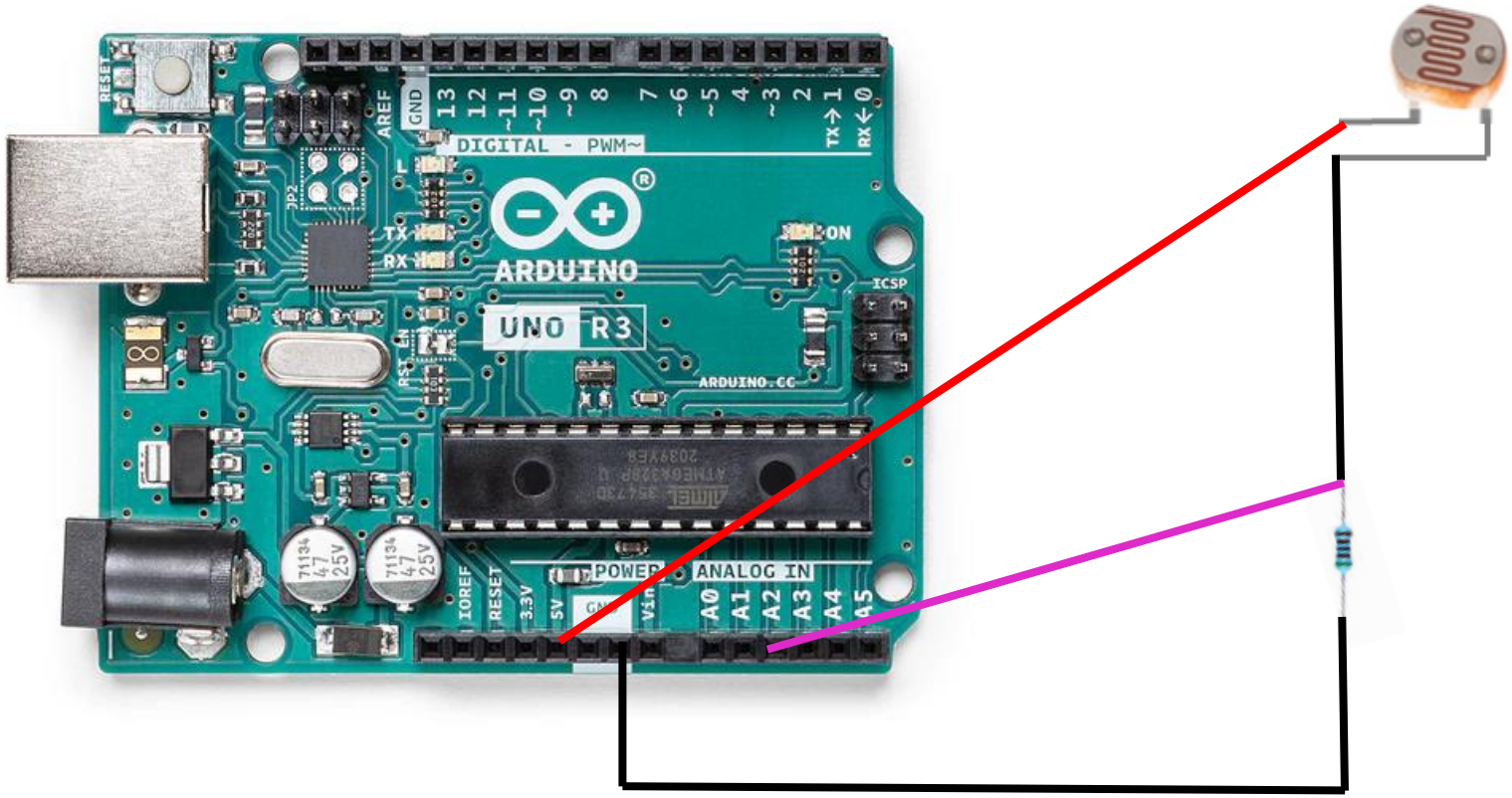
```
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

- Jouez avec la valeur du délai pour faire clignoter plus ou moins la LED
- Vous aurez à sauvegarder le programme dans un autre fichier

Prise en main

- **Lecture d’un capteur analogique**
- Récupérez une photo-résistance + résistance
- Effectuez le branchement suivant:
  - Câble partant de la photo-résistance vers le 5V
  - Câble partant de la résistance vers le GND
  - Câble entre les deux vers l’entrée analogique A2





Prise en main

- **Lecture d'un capteur analogique**
- Récupérez une photo-résistance + résistance
- Effectuez le branchement suivant:
  - Câble partant de la photo-résistance vers le 5V
  - Câble partant de la résistance vers le GND
  - Câble entre les deux vers l'entrée analogique A2
- Créez un nouveau fichier. Il sera pré-rempli

sketch_may28a

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Prise en main

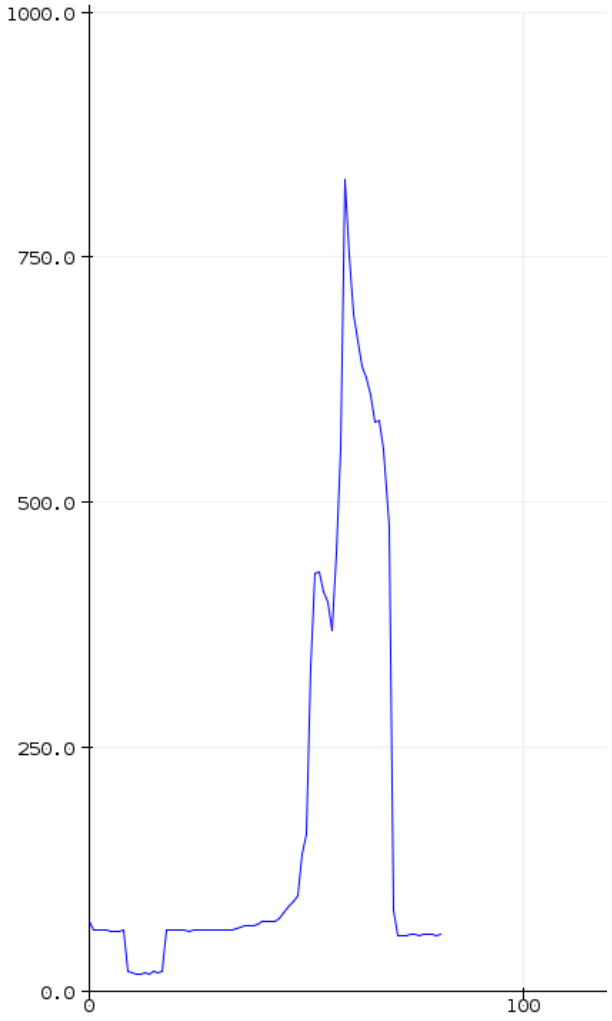
- **Lecture d'un capteur analogique**
- Récupérez une photo-résistance + résistance
- Effectuez le branchement suivant:
  - Câble partant de la photo-résistance vers le 5V
  - Câble partant de la résistance vers le GND
  - Câble entre les deux vers l'entrée analogique A2
- Créez un nouveau fichier. Il sera pré-rempli

- Nous allons écrire notre premier code :
  1. Déclaration des paramètres que nous allons utiliser:
     int PinDiode = A2;
     int LightLevel = 0;
  2. Rajouter dans setup, le démarrage de la liaison série (USB):
     Serial.begin(115200);
  3. Ecrire dans la partie loop, la lecture de la mesure de l'intensité lumineuse et sa sortie sur le port série:
     LightLevel = analogRead(PinDiode);
     Serial.println(LightLevel);
     delay(1000);

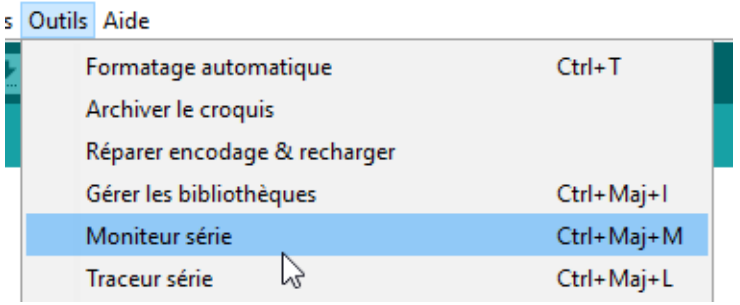
```
int PinDiode = A2;
int LightLevel = 0;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
}

void loop() {
  // put your main code here, to run repeatedly:
  LightLevel = analogRead(PinDiode);
  Serial.println(LightLevel);
  delay(1000);

}
```

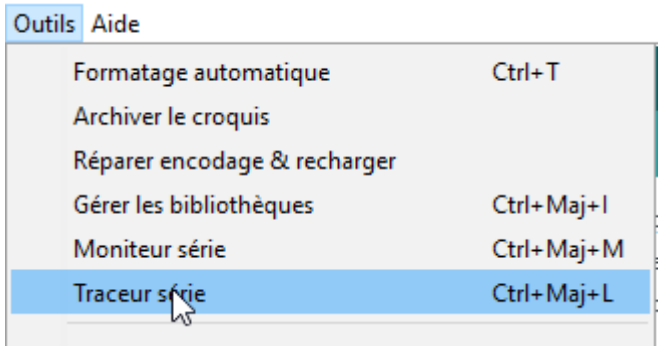
Prise en main

- **Lecture d’un capteur analogique**



- Sauvegardez le fichier, le vérifier puis le téléverser
- **Vérification** sur le moniteur série ou traceur série après téléversement
- Trouvez un moyen de faire varier la valeur mesurée du niveau de lumière.





Prise en main

- **Lecture d’un capteur analogique**
- A quoi cette mesure d’intensité pourrait bien servir ?

Prise en main

- **Lecture d'un capteur analogique**
- A quoi cette mesure d'intensité pourrait bien servir ?
- Et si on essayait de détecter quand elle dépasse une certaine valeur !

- **Programme** pour allumer une LED qui déclenchera son extinction et l'alarme du buzzer:

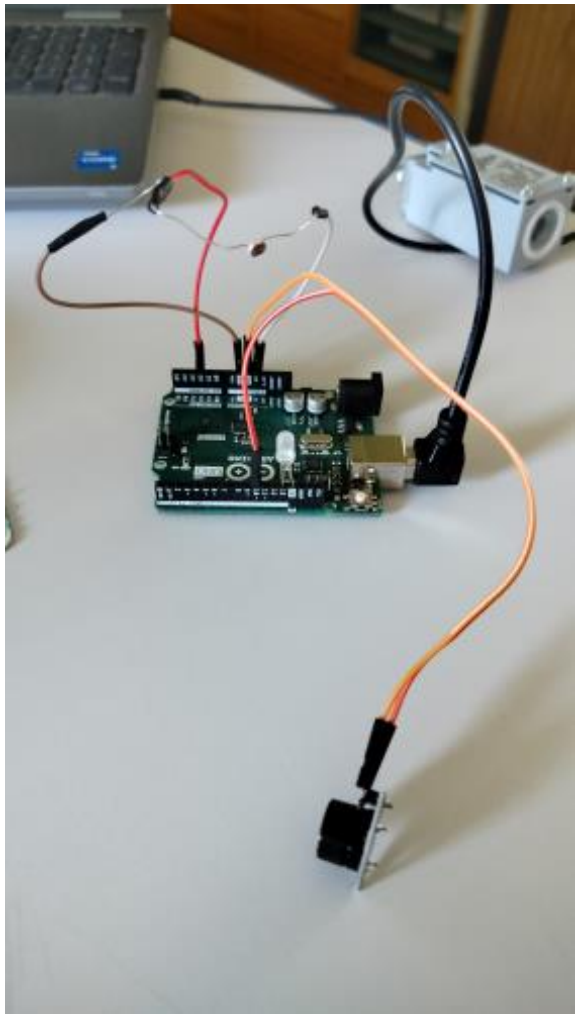
```
int PinDiode = A2;
int LightLevel = 0;
int PinLED = 13;
int PinBuzz = 10;

void setup() {
Serial.begin(115200);
 pinMode(PinLED, OUTPUT);
 pinMode(PinBuzz, OUTPUT);
}

void loop() {
LightLevel = analogRead(PinDiode);
 Serial.println(LightLevel);
 if (LightLevel>100) {
  digitalWrite(PinLED, LOW);
  tone(PinBuzz, 1000, 1000); }
 else {
  digitalWrite(PinLED, HIGH);
 }
 delay(2000);
}
```

Déclaration des pins utilisés

Configuration

Utilisation et changement d'état

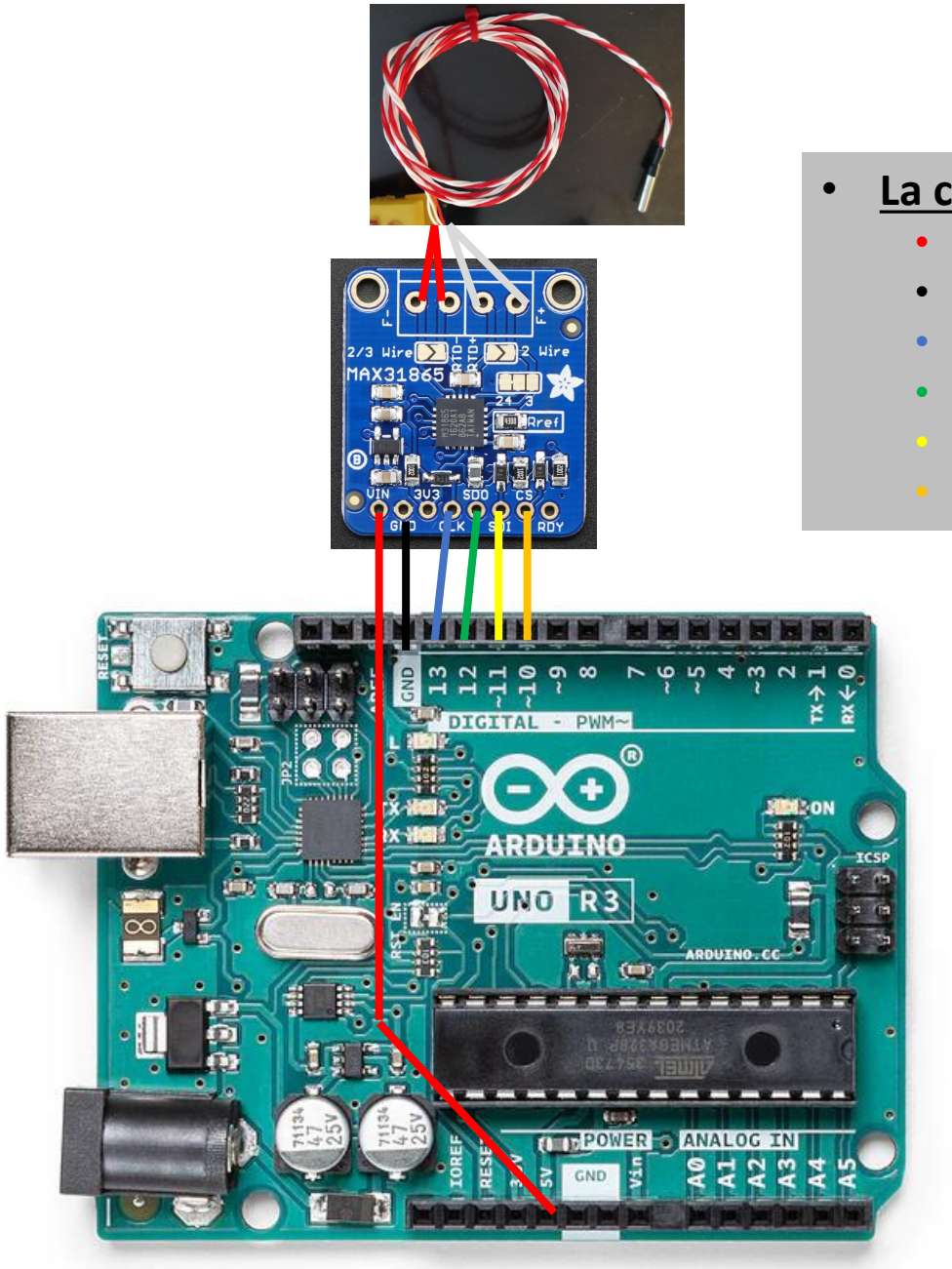
```
int PinDiode = A2;
int LightLevel = 0;
int PinLED = 13;
int PinBuzz = 10;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  pinMode(PinLED, OUTPUT);
  pinMode(PinBuzz, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  LightLevel = analogRead(PinDiode);
  Serial.println(LightLevel);
  if (LightLevel>100) {
    digitalWrite(PinLED, LOW);
    tone(PinBuzz, 1000, 1000);
    delay(1000);
  }
  else {
    digitalWrite(PinLED, HIGH);
  }
  delay(2000);
}
```



NB : tone(pin, frequency, duration)

Next level

- **Lecture d'une PT100**
- Utilisation de la carte MAX31865
- Prenez la carte câblée à la PT100
- Faire le montage



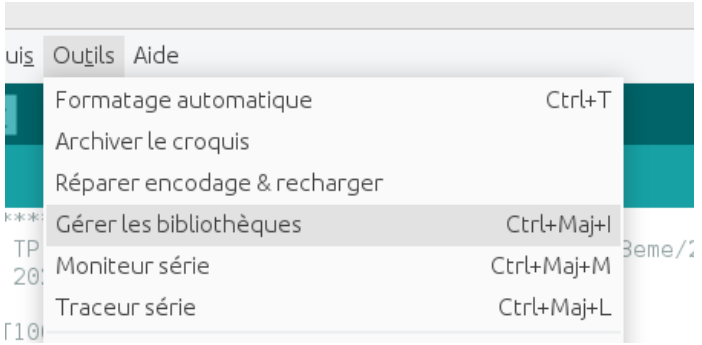
- **La carte MAX31865** :
  - Vin vers 5V
  - gnd vs gnd
  - CLK vers pin Digital #13
  - SDO vers pin Digital #12
  - SDI vers pin Digital #11
  - CS vers pin Digital #10

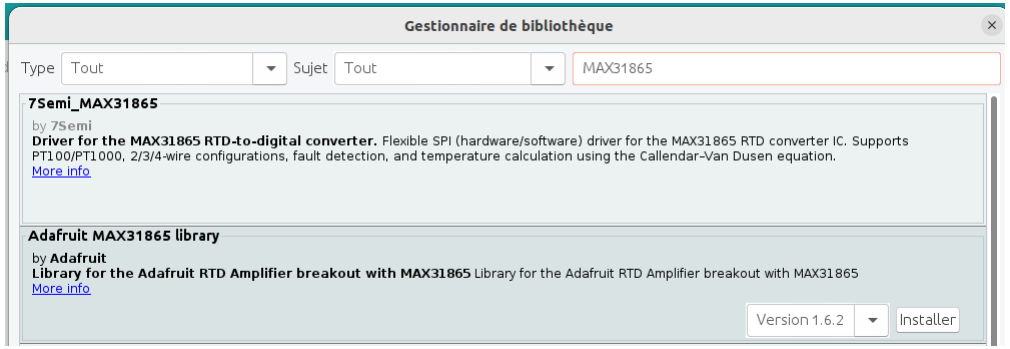
Next level

- **Lecture d'une PT100**
- Utilisation de la carte MAX31865
- Prenez la carte câblée à la PT100
- Faire le montage
- **Le code**

Appel d'une librairie qu'il faudra ptet installer

```
/home/darkside/Arduino/PT100/PT100.ino:8:31: erreur fatale: Adafruit_MAX31865.h : Aucun fichier ou dossier de ce type
compilation terminée.
exit status 1
Erreur de compilation pour la carte Arduino Uno
```



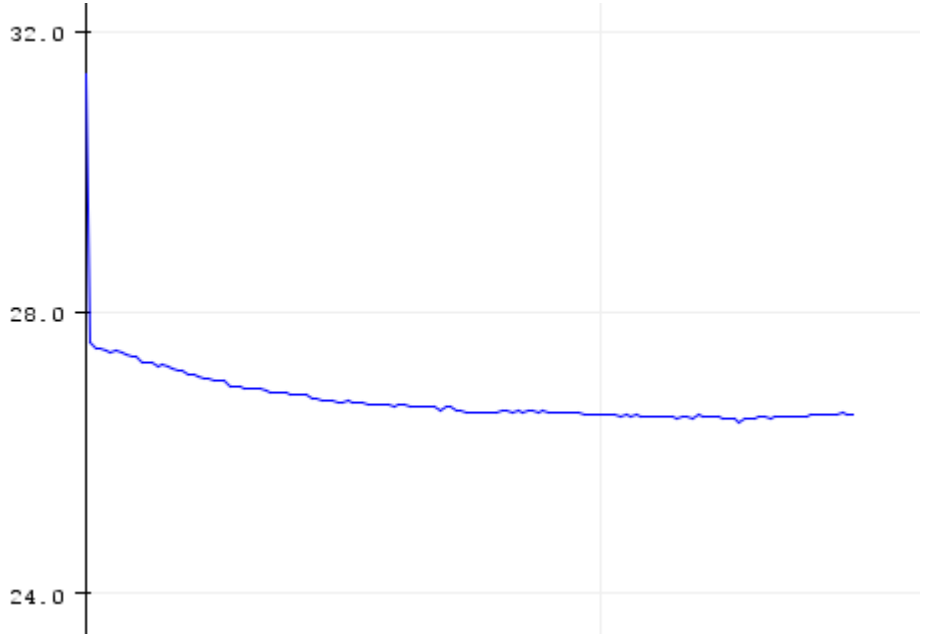



```
#include <Adafruit_MAX31865.h>

// Dans l'ordre les pins du SPI: CS, DI, DO, CLK
Adafruit_MAX31865 thermo = Adafruit_MAX31865(10, 11, 12, 13);

// Caleur de la résistance Rref. 430.0 pour PT100 et 4300.0 pour PT1000
#define RREF      430.0
// Valeur nominale de la résistance à 0°C
// 100.0 pour PT100, 1000.0 pour PT1000
#define RNOMINAL  100.0
float temperature = 0;

void setup() {
   Serial.begin(115200);
   thermo.begin(MAX31865_4WIRE);  // configuration en 4 fils.
   delay(1000);
}


void loop() {

// valeur de la température sur la sortie série.
   temperature = thermo.temperature(RNOMINAL, RREF);
   Serial.println(temperature);

   delay(1000);
}
```

Next level

- **Lecture d’une PT100**
- Utilisation de la carte MAX31865
- Prenez la carte câblée à la PT100
- Faire le montage
- Le code
- **Affichez le traceur série**
- **Jouez avec la PT100 pour faire varier la température**

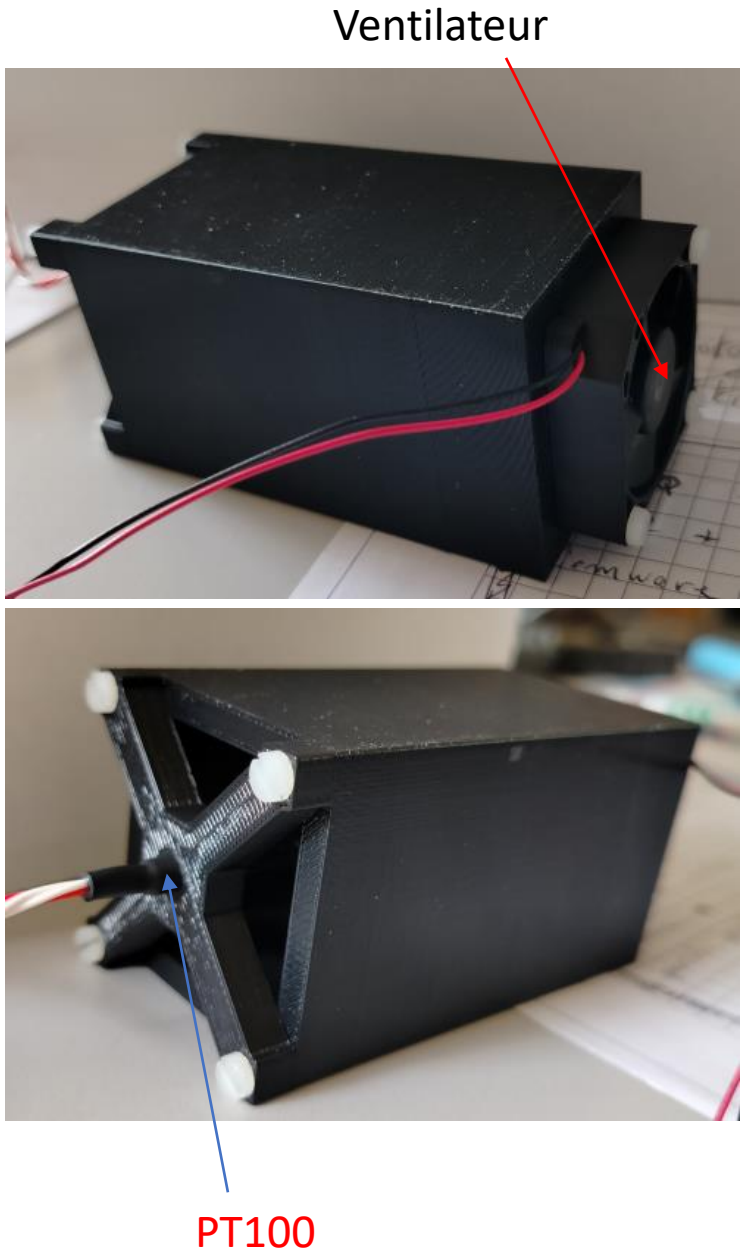
- **A quoi peut nous servir une telle mesure ?**



Next level

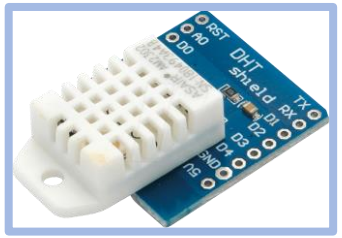
- **Lecture d’une PT100**
- Utilisation de la carte MAX31865
- Prenez la carte câblée à la PT100
- Faire le montage
- Le code
- Affichez le traceur série
- Jouez avec la PT100 pour faire varier la température

- **On pourrait rajouter un ventilo pour essayer de faire baisser la température.**
- **Branchez le câble noir sur gnd de l’ARDUINO et le rouge sur le 3.3V.**
- **Insérez la PT100 dans le trou à l’opposé du ventilo**
- **Qu’est ce que vous observez sur le traceur série ?**



Et le reste ?

- Capteur de flux d’air
- DHT22 → humidité et t°
- HM1500 → humidité
- Switch → interrupteur



Et le reste ?

- **Capteur de flux d'air**

```
// Flow meter parameters
#define RANGE           150   //Measurement Range
#define ZEROVOLTAGE      0.5   //Zero Voltage
#define FULLRANGEVOLTAGE  4.5   //Full scale voltage
#define VREF            5     //Reference voltage
int FlowPin = A0;   // pin of flow meter
float FlowValue = 0;

int LEDpin = 12;

void setup()
{
 Serial.begin(9600);}

void loop()
{
 // Air flow measurement IN
 FlowValue = analogRead(FlowPin)*VREF;
 FlowValue = FlowValue / 1024;
 FlowValue = RANGE*(FlowValue - ZEROVOLTAGE)/(FULLRANGEVOLTAGE - ZEROVOLTAGE);
 Serial.println(FlowValue); delay (500);
}
```

Winsen
炜盛科技Zhengzhou Winsen Electronics Technology Co., Ltd www.winsen-sensor.com

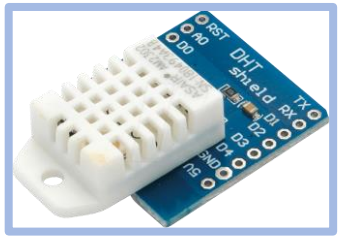
Calculation for Airflow

Actual flow=full scale * (sensor actual output voltage-zero output voltage) / (full scale output voltage-zero output voltage)

Et le reste ?

- **DHT22 → humidité et t°**

```
#include <SPI.h>
#include "DHT.h"
#define DHTPIN1 7     // Digital Pin DHT1
#define DHTTYPE DHT22   // DHT 22  (AM2302)
// Initialize DHT sensor for normal 16mhz Arduino
DHT dht1(DHTPIN1, DHTTYPE);
float h = 0;
float t = 0;

void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  // Init DHT22
  dht1.begin();
}

void loop() {
        // Reading temperature and humidity of the first sensor
        h = dht1.readHumidity();
        t = dht1.readTemperature();
        // Check if any reads failed and exit early (to try again).
        if (isnan(h) || isnan(t)) {
          Serial.println("Failed to read from DHT sensor!");
          return;
        }
        Serial.print(h);
        Serial.print("\t");
        Serial.println(t);
        delay(1000);
}
```



On a besoin d'une librairie spéciale DHT.h
Branchement:

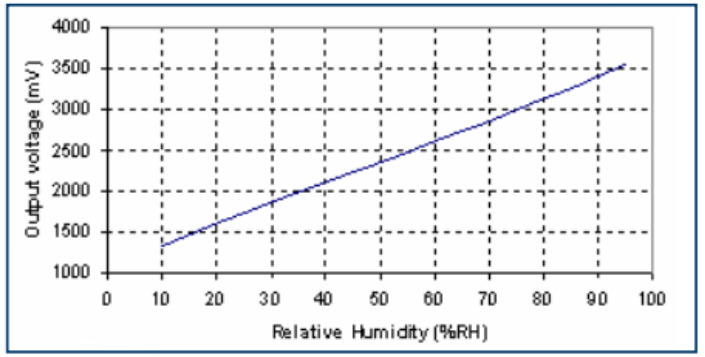
- Rouge sur 5V
- Noir sur gnd
- Vert sur pin 7

Et le reste ?

- HM1500 → humidité

```
// Humidity sensor HM1500LF
int HsensorPin=A4;
float val_h = 0;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  // Reading humidity from HM1500LF
  val_h = analogRead(HsensorPin);
  val_h = (val_h*5/1023) * 38.9 - 41.939;
  Serial.println(val_h);
  delay (500);
}
```

Wire	Color	Function
W1	White	Ground
W2	Blue	Supply Voltage
W3	Yellow	Humidity Output Voltage



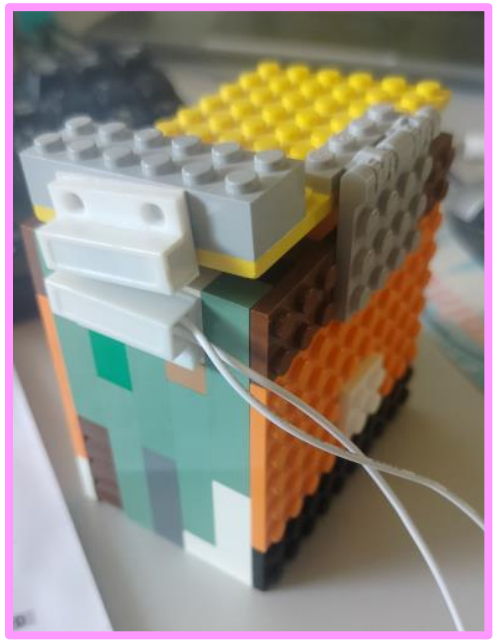
RH (%)	Vout (mV)	RH (%)	Vout (mV)
10	1325	55	2480
15	1465	60	2605
20	1600	65	2730
25	1735	70	2860
30	1860	75	2990
35	1990	80	3125
40	2110	85	3260
45	2235	90	3405
50	2360	95	3555

Et le reste ?

- Switch → interrupteur

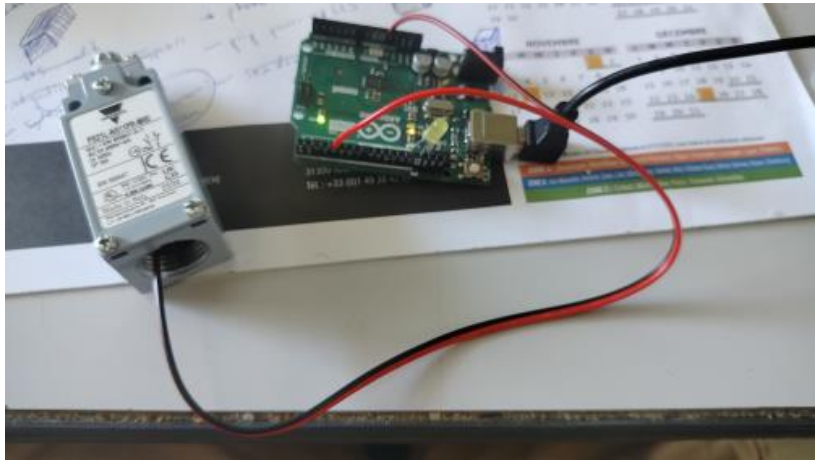
```
int SwitchPin = 4;
bool SwitchAnswer = false;
int LEDpin = 13;

void setup() {
  Serial.begin(9600);
  pinMode(SwitchPin, OUTPUT);
  pinMode(LEDpin, OUTPUT);
}

void loop() {
  SwitchAnswer = digitalRead(SwitchPin);
  Serial.print(SwitchAnswer);
  if (SwitchAnswer == HIGH){
    digitalWrite(LEDpin, HIGH);
    Serial.println(" Switch is opened");
  }
  else {
    digitalWrite(LEDpin, LOW);
    Serial.println(" Switch is closed");
  }
  delay (1000);
}
```

ou



Branchement:

- Un câble sur 5V
- L'autre sur pin digitale 4

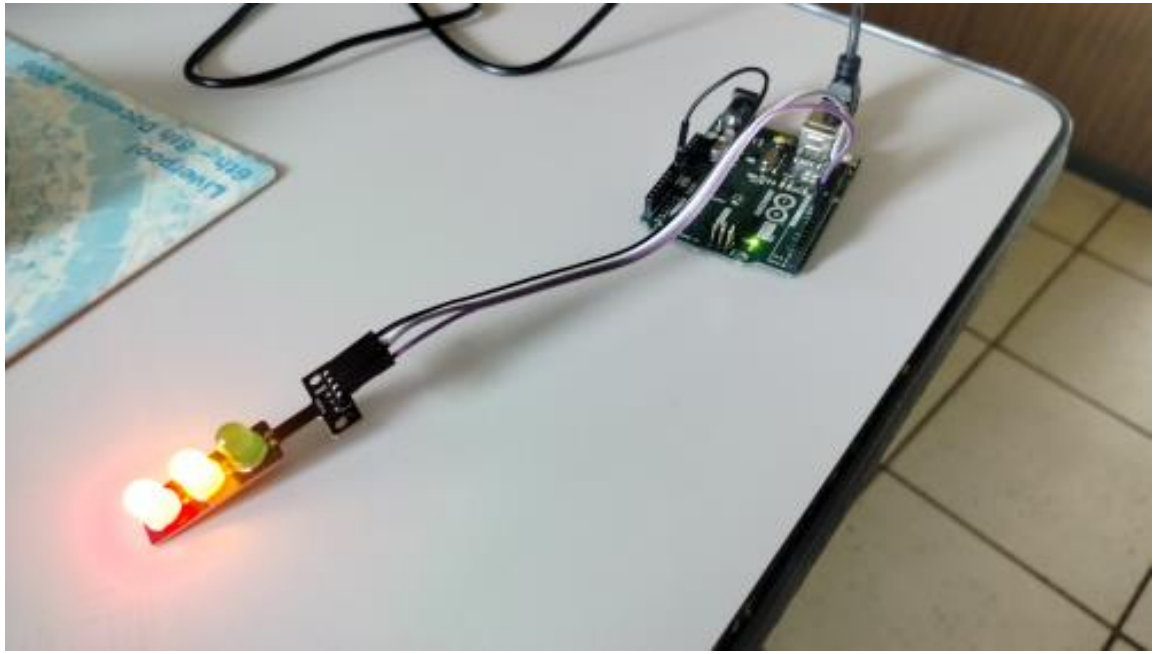


Comment utiliser un tel capteur ?

BONUS

- **Feu tricolore**

```
int pinRed = 10;
int pinYellow = 8;
int pinGreen = 6;

void setup() {
  pinMode(pinRed,OUTPUT);
  pinMode(pinYellow,OUTPUT);
  pinMode(pinGreen,OUTPUT);
}

void loop() {
  digitalWrite(pinRed, HIGH);
  delay(200);
  digitalWrite(pinYellow, HIGH);
  delay(200);
  digitalWrite(pinGreen, HIGH);
  delay(200);
  digitalWrite(pinGreen, LOW);
  delay(200);
  digitalWrite(pinYellow, LOW);
  delay(200);
  digitalWrite(pinRed, LOW);
  delay(200);
}
```

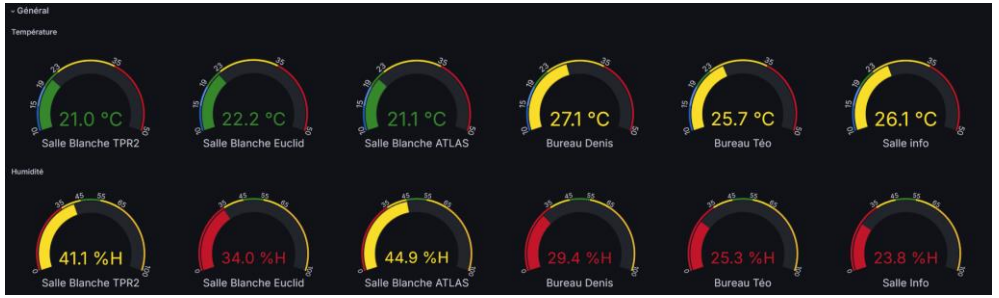


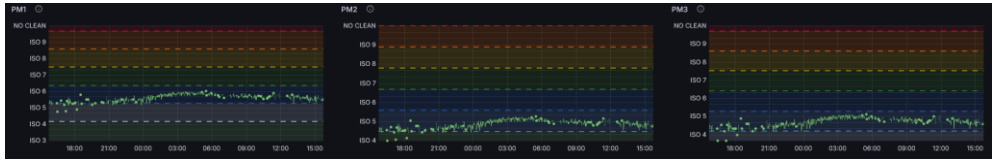
Branchement:

- Gnd carte vers gnd arduino
- G vers pin 6
- Y vers pin 8
- R vers pin 10

Le monitoring à quoi ça sert ?

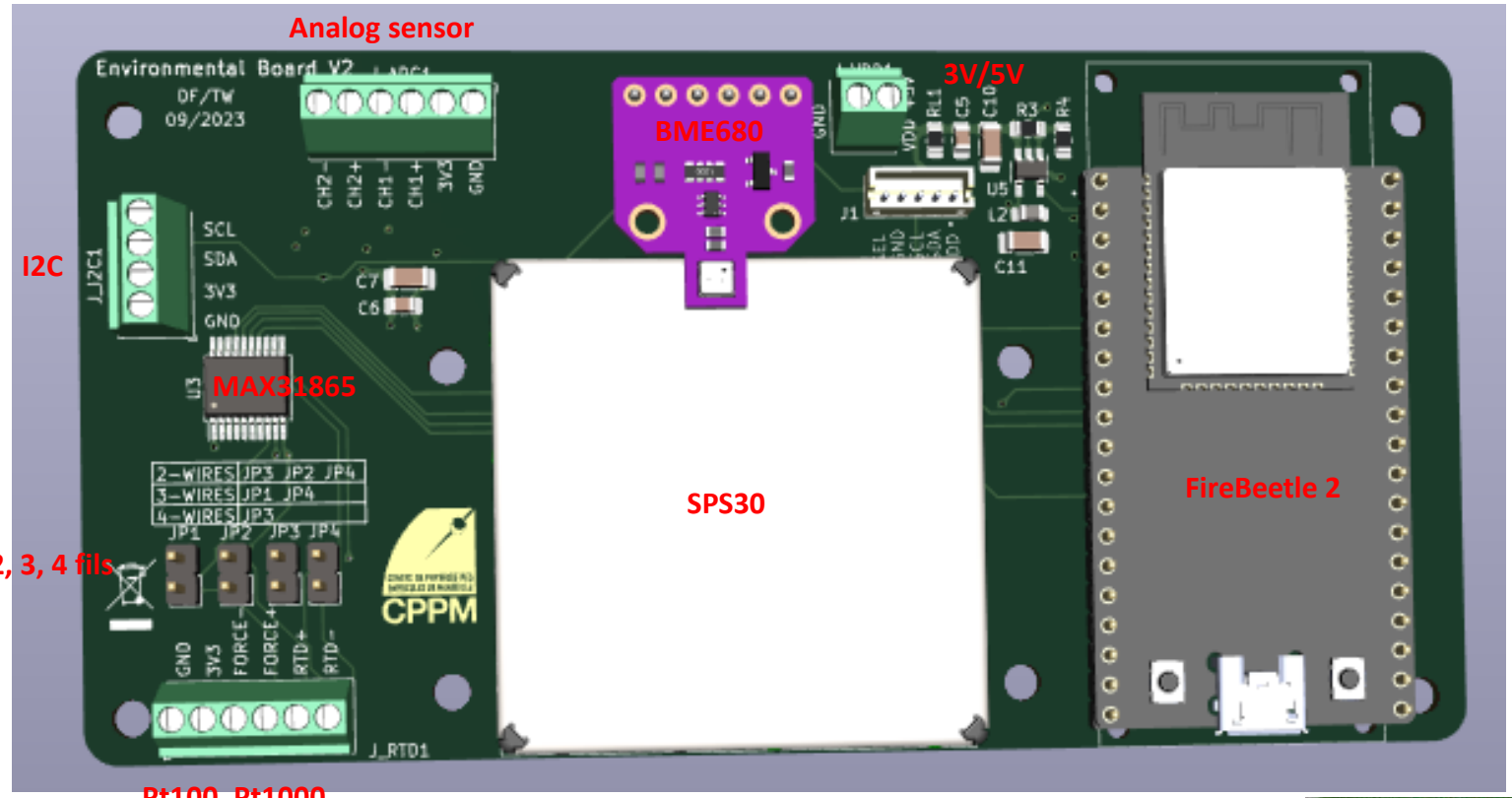
- Surveillance des paramètres environnementaux d'une salle. Typiquement : salle propre.
- Au CPPM : développement d'une carte environnementale qui permet de monitorer température, humidité, niveau de poussières, taux de CO2, etc.
- Récupération des données via le protocole Lora et stockage (base de données).
- Affichage des paramètres sur Grafana

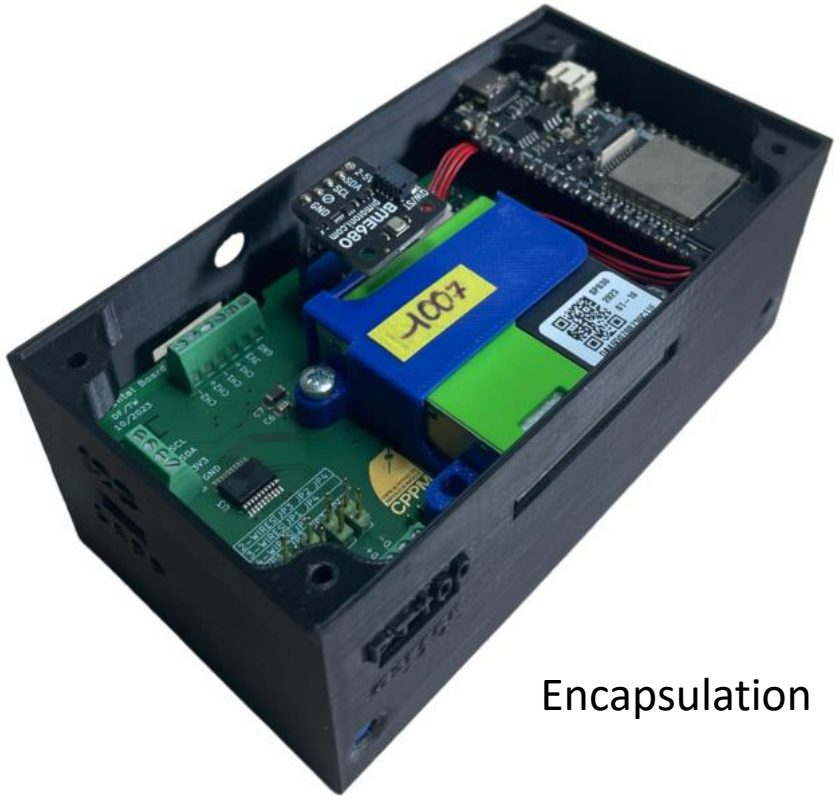




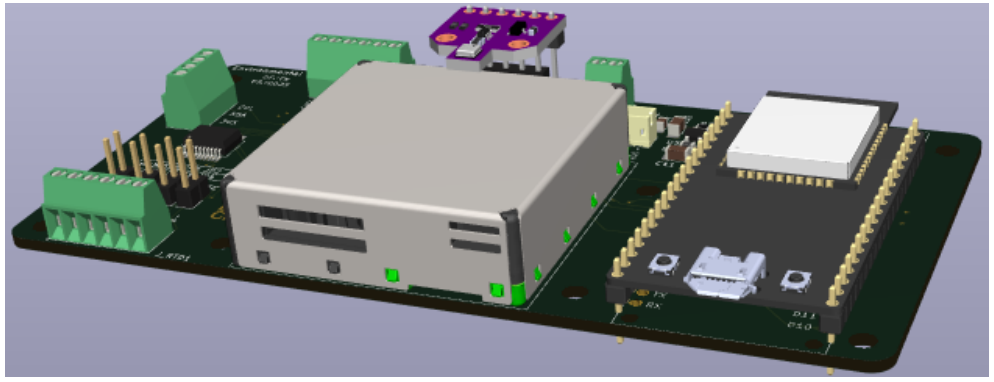
Illustrations

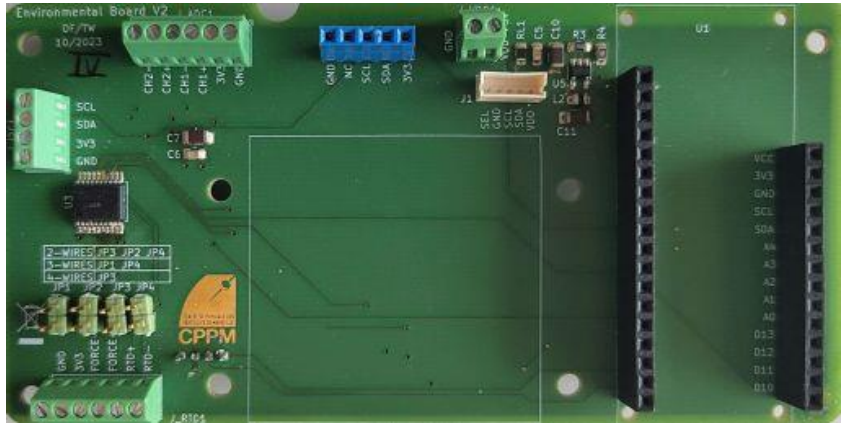
Pictures and CAD





Encapsulation





PCB top

Illustrations

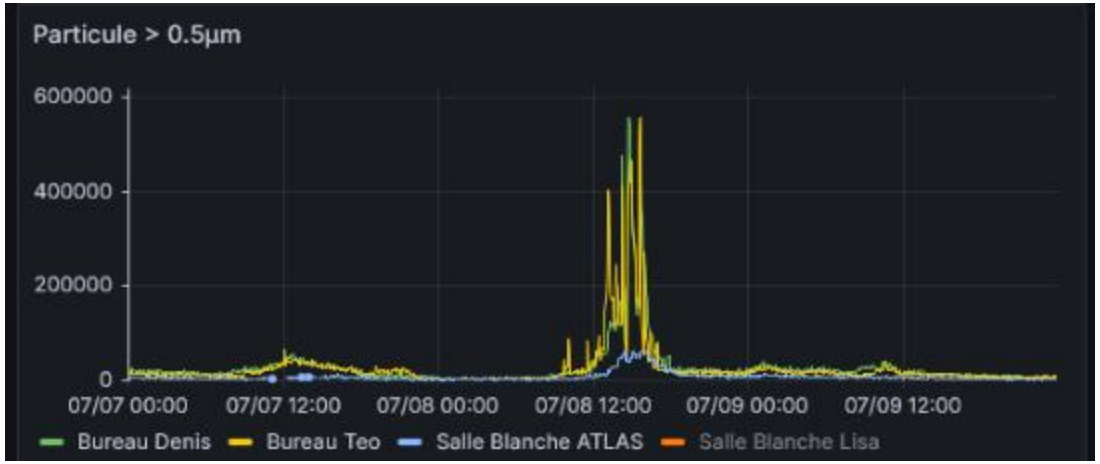
Soldering activities

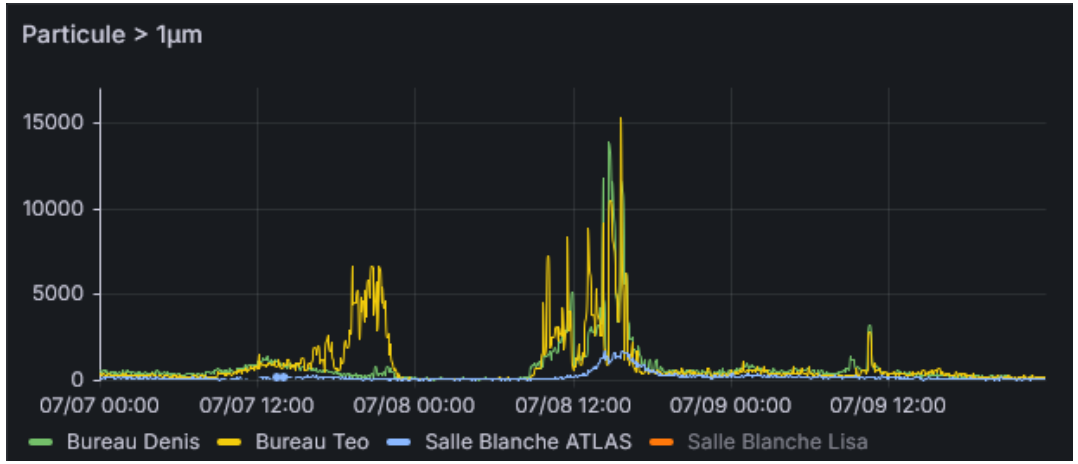
Monitoring from ATLAS-Itk clean room. 2 periods spotted with a notable increase:

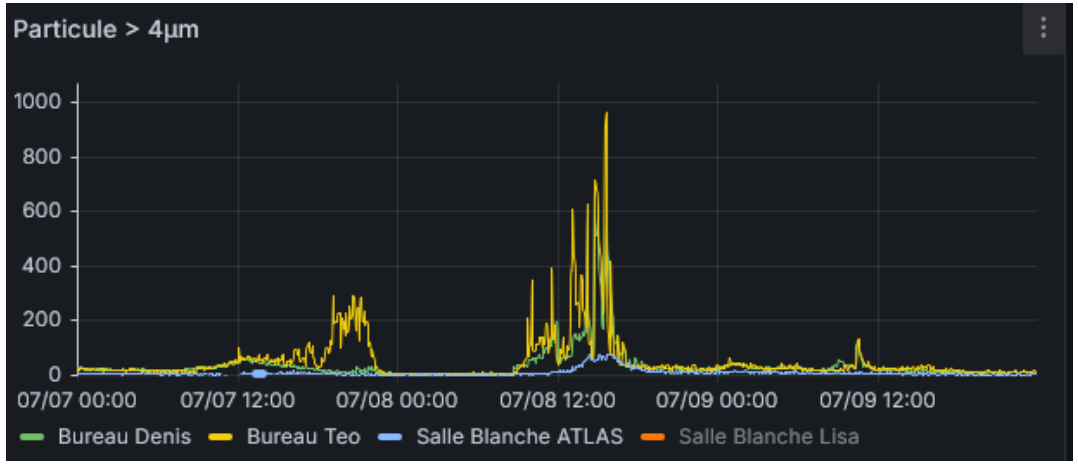
- Mid-may: soldering of some sensors+cables inside the clean room
- Mid-June: soldering in the assembly galery → affecting the CTA used to regulate the temperature inside the clean room

Illustrations

Fire: 7-8 juillet 2025







NODE-RED si on a le temps !

Description rapide de node-red

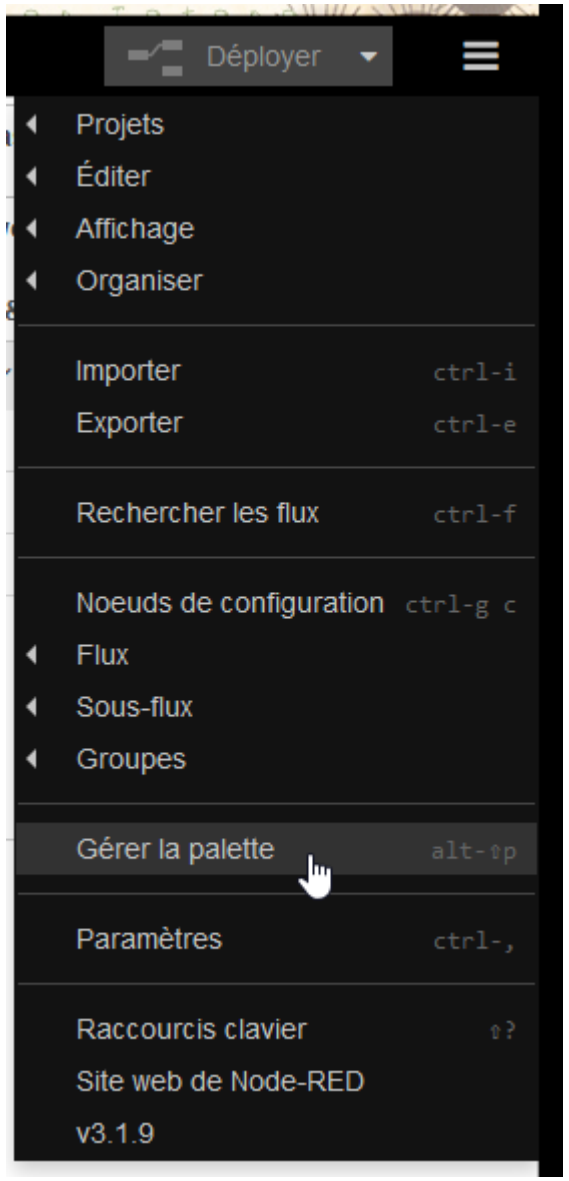
https://nodered.org/

- *Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.*
- *It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click.*
- Une fois installé et exécuté on accède à l'interface pour coder à cette adresse : http://127.0.0.1:1880. On accède au dashboard (visualisation) à celle ci : http://127.0.0.1:1880/ui.
- Utiliser pour du contrôle-commande ou du monitorage.
- Programmation graphique avec des nœuds à connecter + javascript pour rajouter des algorithmes.
- Entre chaque nœud connecté, le paramètre *msg.payload* est transmis

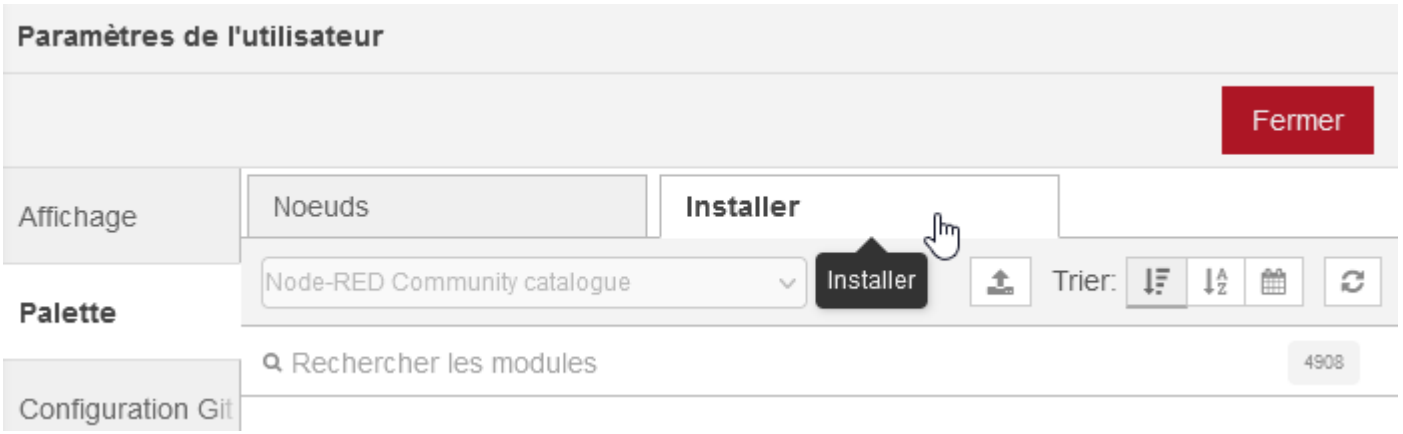
Démarrage node-red

- Dans un terminal taper : *node-red*
- Dans un navigateur rentrer http://127.0.0.1:1880

Palette et installation de nœuds

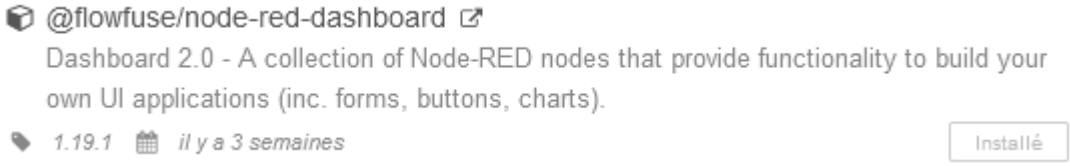


Certains nœuds font partie de librairies à installer via la palette.



Pour nous :

- **node-red-dashboard**
- **node-red-node-serialport**
- **node-red-node-ui-table**
- **node-red-contrib-ui-led**
- **node-red-contrib-boolean-logic-ultimate**
- **@flowfuse/node-red-dashboard**



Nœuds utilisés

NB: utilisez Dashboard2 !

Changer des paramètres

Lien série IN ou OUT

Fonction codée en js

Filtre

Comparaison (valeur > seuil)

Affichage sur dashboard

Texte

Jauge

Graphique

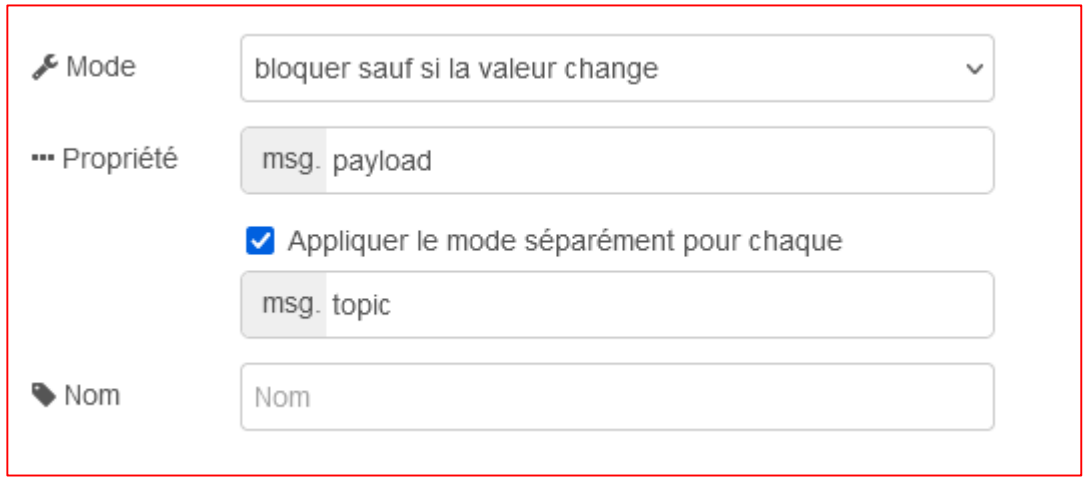
Glissière (input)

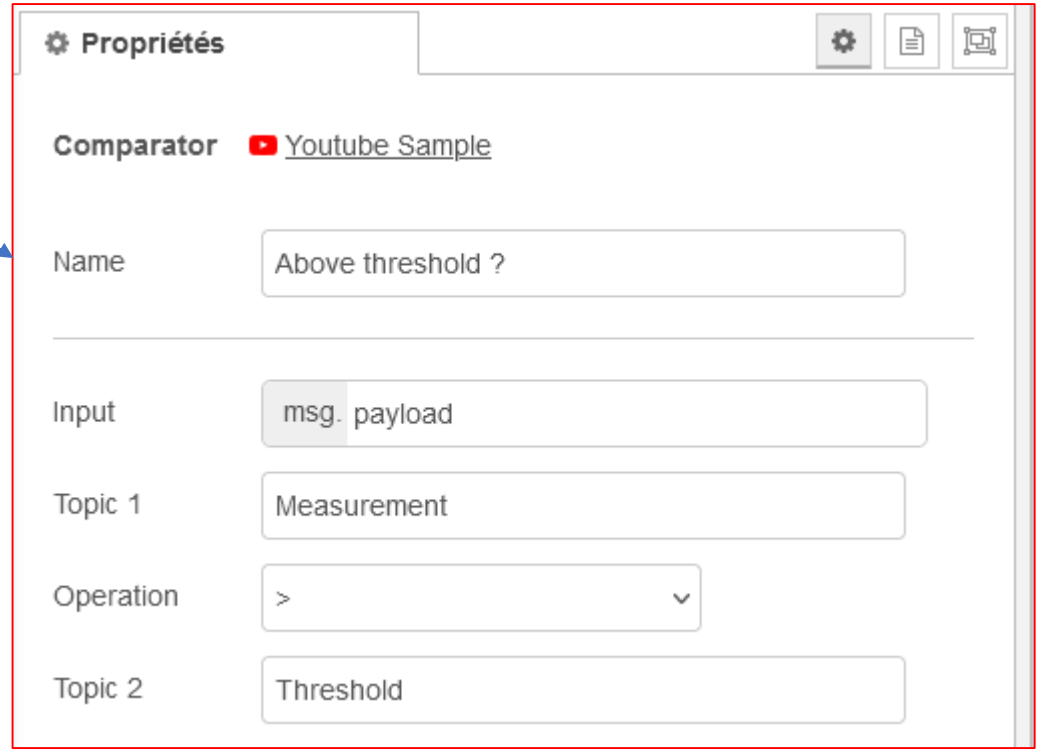
Texte (input)

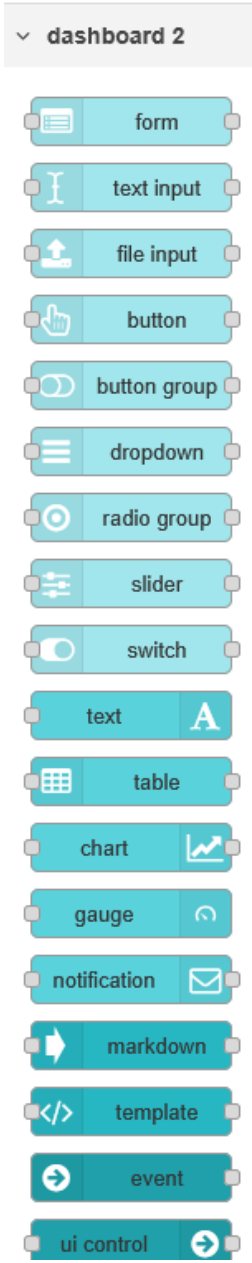
Interrupteur

Injection

Debug output

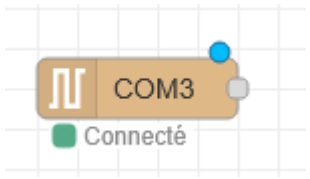


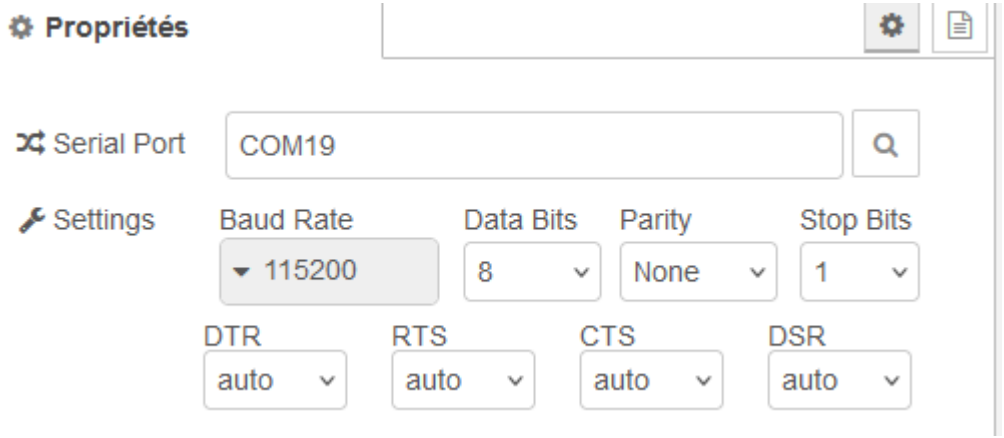




Lecture de l’arduino et affichage

Configuration sur NodeRed du lien série IN

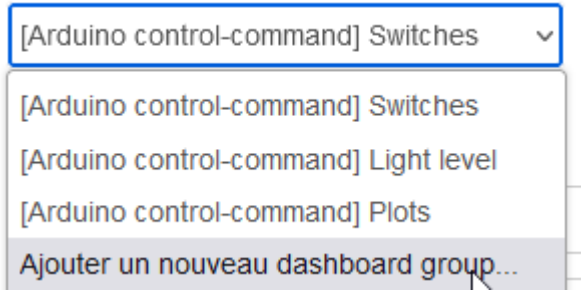




Sélectionner le bon serial port et configurer le lien.

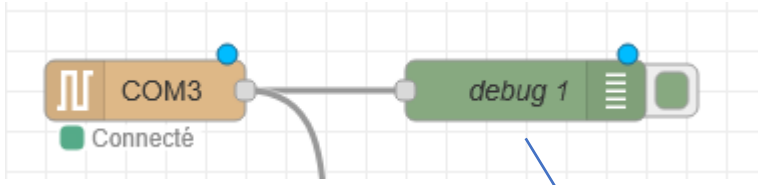
Chaque élément de la dashboard doit être mis dans un groupe et un onglet

Il faut les créer si ils n’existent pas. Ils permettront l’affichage regroupé sur l’interface

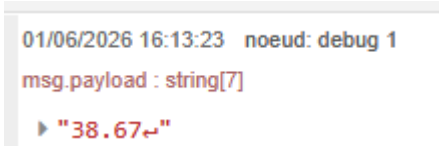


Récupérer les données transmises par l'Arduino

Debug permet d'afficher le
msg.payload en sortie d'un block

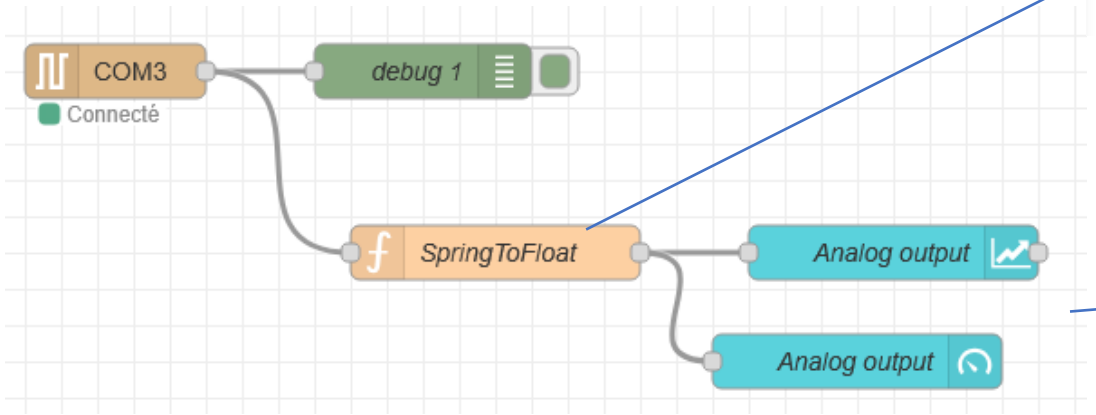


Ce qui sort est une chaîne de caractères (string).



Transformation en nombre entier

```
1  var data=msg.payload;
2  msg.payload = parseFloat(data);
3  return msg;
```



Affichage