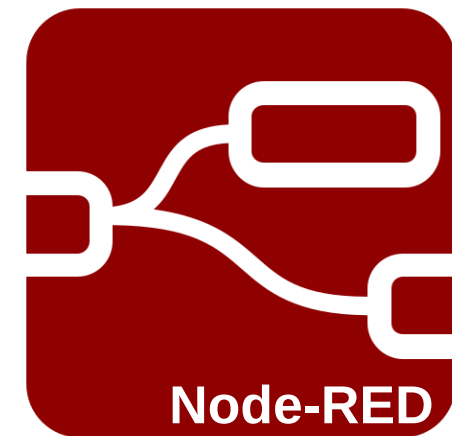
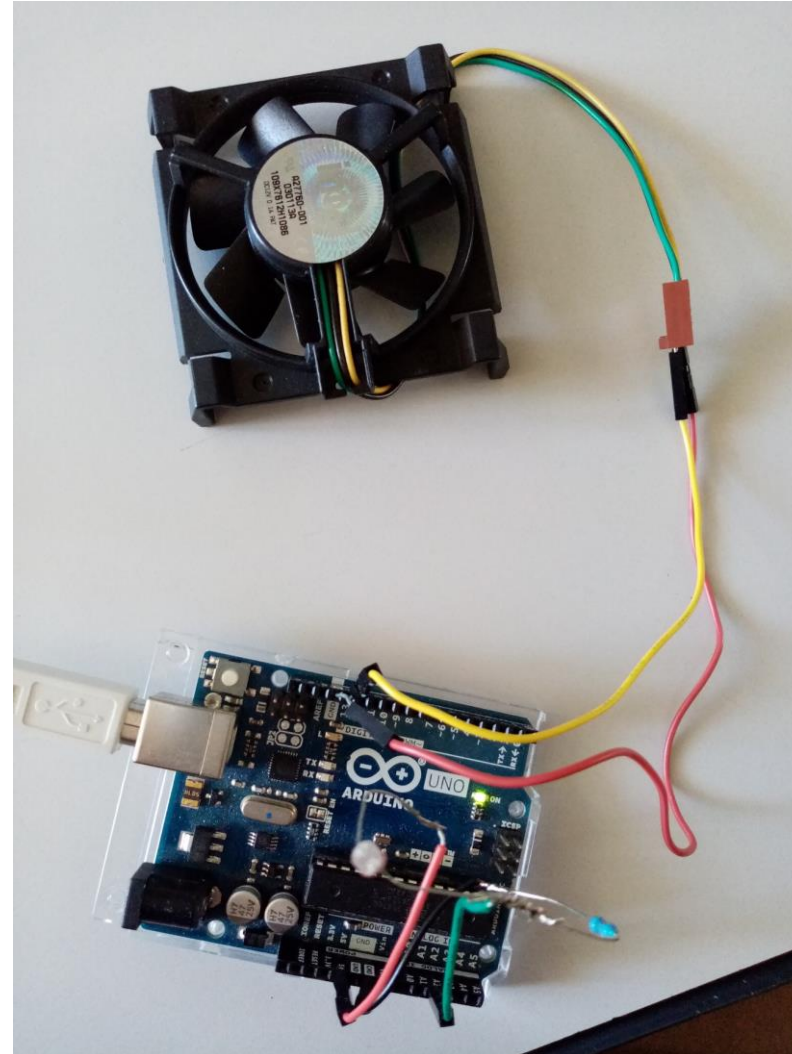


TP Arduino

Julien Zoubian & Pierre Barrillon

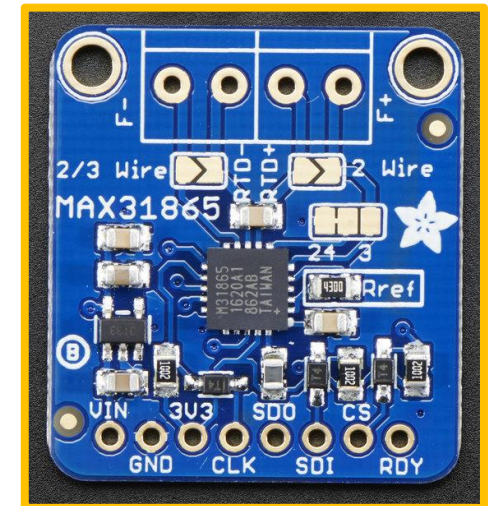
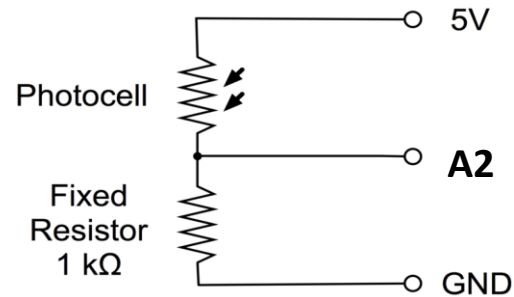
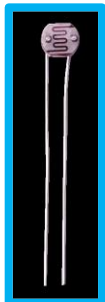
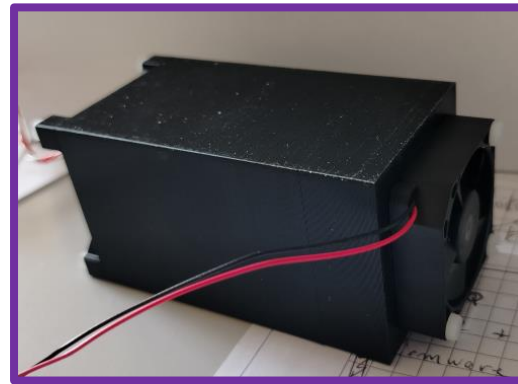
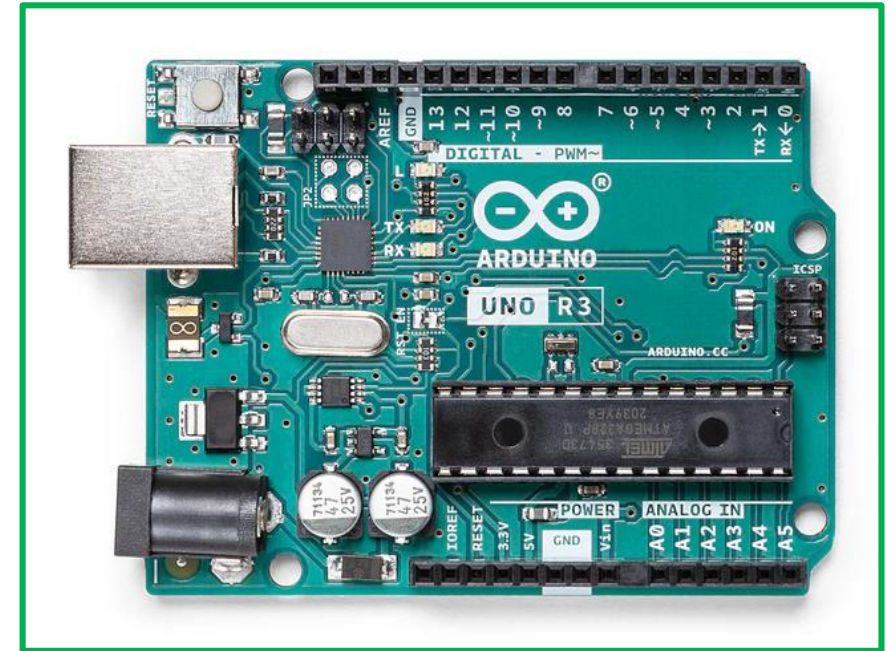


Introduction

- **But:**
 - Utilisation d'un arduino pour récupérer les données mesurées par:
 - Un capteur de température PT100
 - Un capteur de lumière « photo – resistor »
 - Visualisation des données dans une interface graphique Node-red
 - Commande d'un ventilateur

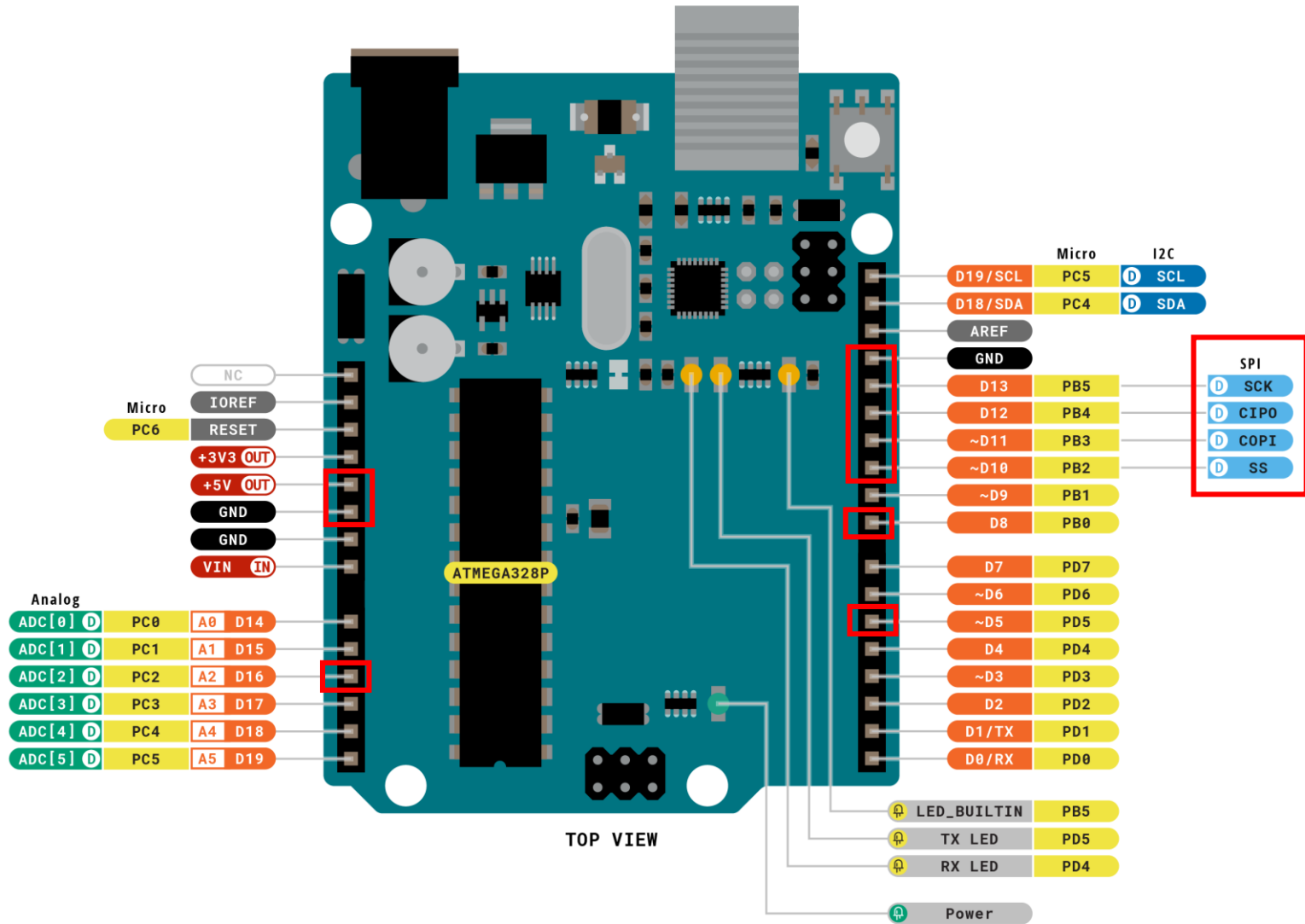
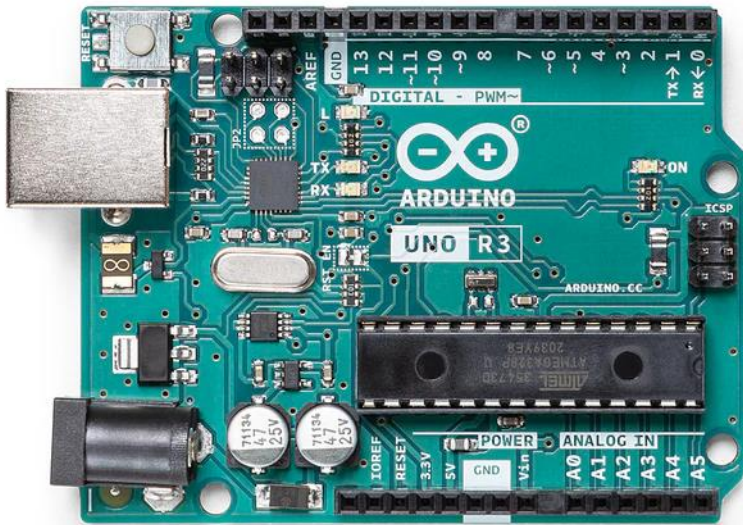
- **Matériel:**

- Arduino UNO
- PT100 « 4 fils »
- Carte MAX 31865 (lecture PT100)
- Photo-resistor + résistance de 1kOhm
- Ventilateur 5V
- Câbles
- Structure mécanique



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



Legend:	Digital	I2C
Power	Analog	SPI
Ground	Main Part	Analog



ARDUINO UNO REV3
SKU code: A000066
Pinout
Last update: 6 Oct, 2022

Câblage - montage

- La carte MAX31865 :

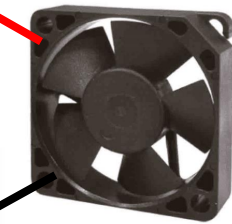
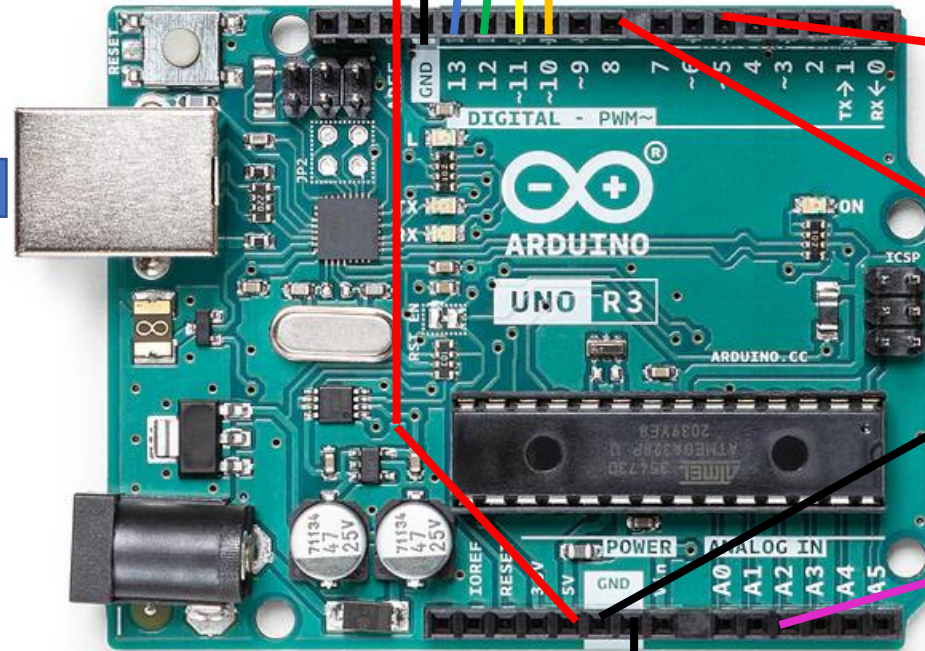
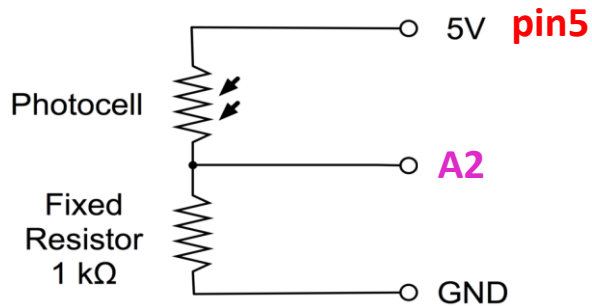
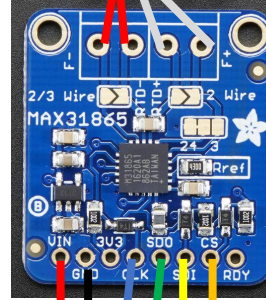
- Vin vers 5V
- gnd vs gnd
- CLK vers pin Digital #13
- SDO vers pin Digital #12
- SDI vers pin Digital #11
- CS vers pin Digital #10

- Le ventilateur :

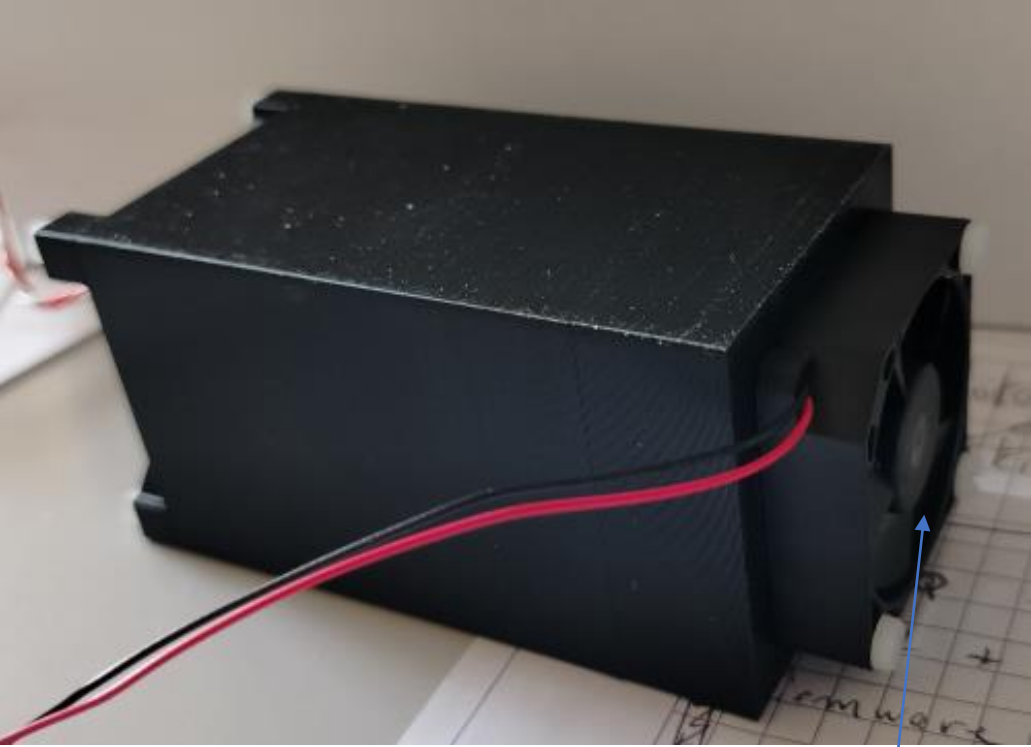
- Câble rouge vers pin Digital #8
- Câble noir vers gnd

- La Photo-resistor :

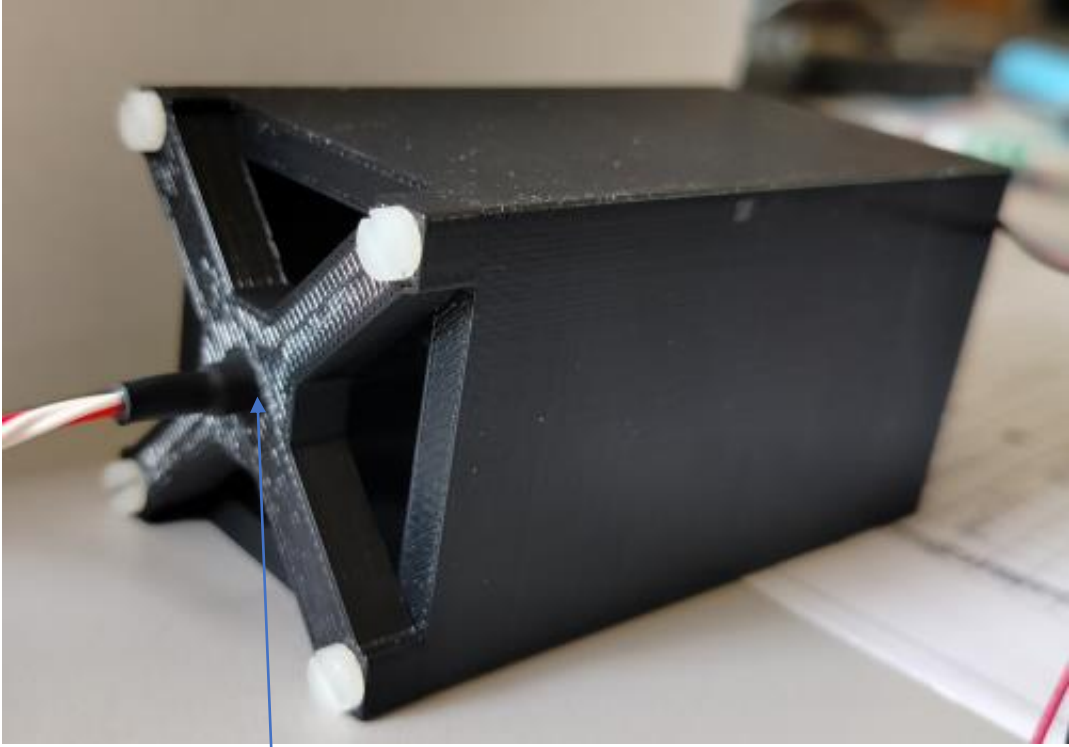
- Câble vers pin Digital #5
- Câble vers pin Analog #A2
- Câble de la résistance vers gnd



Installation



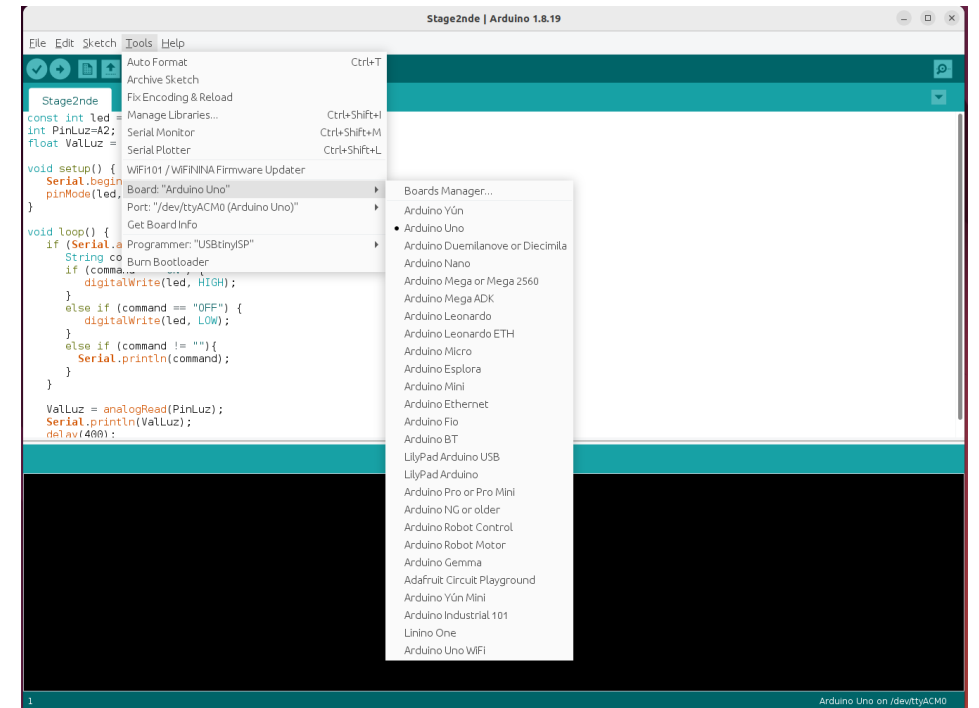
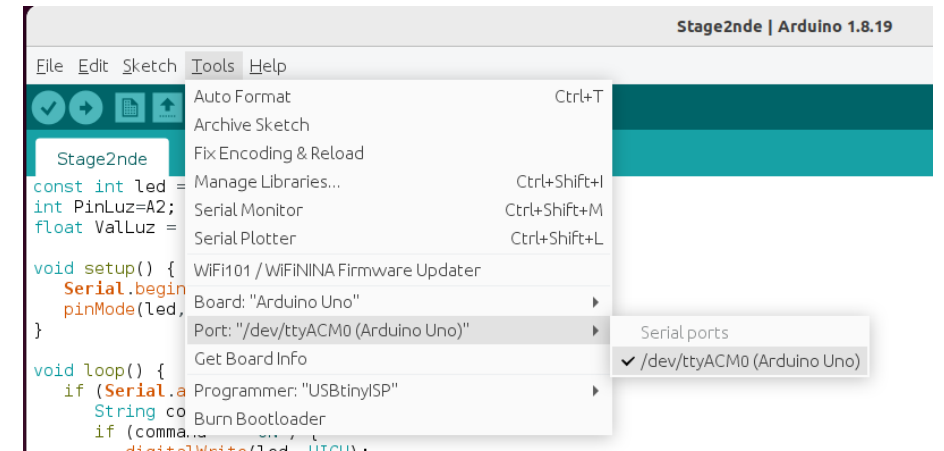
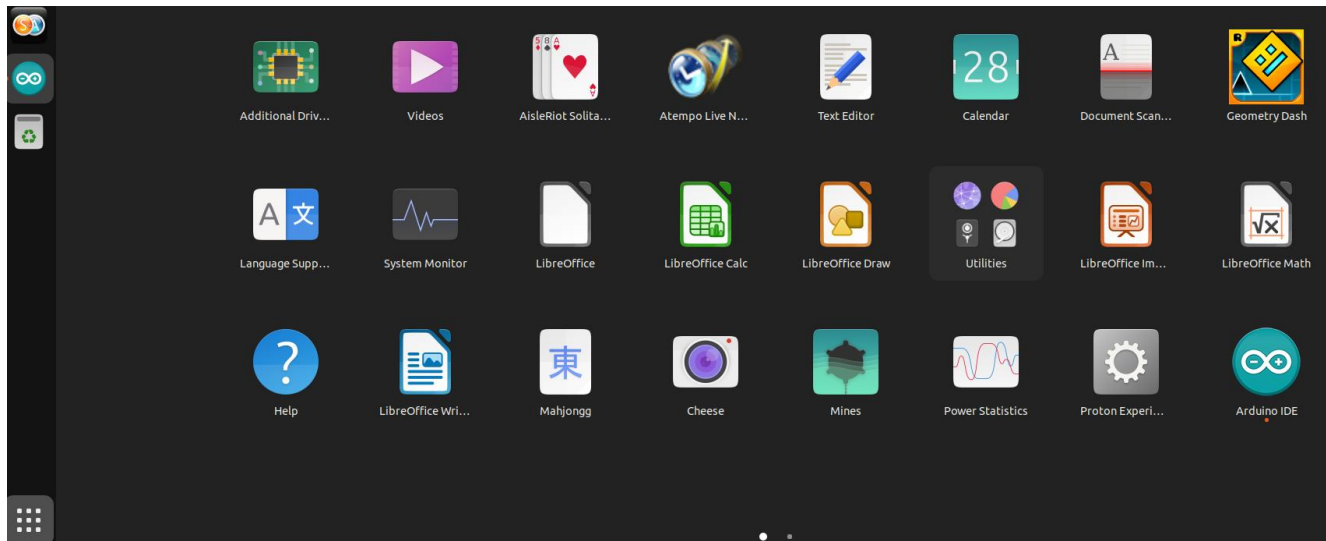
Ventilateur



PT100

Démarrage

- Connection au PC. Demander le mdp.
- Rechercher et lancer le logiciel IDE Arduino (show applications) ou ouvrir le fichier Stage3eme.ino (demandez où il se trouve).
- Le code est décrit juste après.
- Il faudra le « téléverser » dans la carte Arduino en sélectionnant le bon type de carte et le bon port.



Code Arduino

- Entête décrivant brièvement le code
- Appel de la librairie MAX31865 et utilisation pour définir les pins qui seront utilisées (10 à 13).
- Définition des résistances de référence et nominal
- Définition du paramètre correspondant à la température
- Définition de la pin utilisée pour le ventilateur (8)
- Définition des pins utilisés pour la photo-resistor (A2 et 5)
- Définition du paramètre correspondant à l'intensité lumineuse

Stage3eme

```
/*
Programme pour le TP Arduino/Node-red dédié aux stagiaires de 3eme/2nde
Version pour décembre 2025
Inclut :
- lecture d'une PT100 4 fils via la carte MAX31865
- lecture d'une photodiode
- ON/OFF d'un petit ventilateur via commande
*/

#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;

// Pin pour alimenter (ou pas) le ventilateur
const int fan = 8;
// Pins et paramètre correspondant à la photodiode
int PinDiode=A2;
int PinPowerDiode = 5;
float ValLuz = 0;
```

Code Arduino

- Entête décrivant brièvement le code
- Appel de la librairie MAX31865 et utilisation pour définir les pins qui seront utilisées (10 à 13).
- Définition des résistances de référence et nominal
- Définition du paramètre correspondant à la température
- Définition de la pin utilisée pour le ventilateur (8)
- Définition des pins utilisés pour la photo-resistor (A2 et 5)
- Définition du paramètre correspondant à l'intensité lumineuse

- Démarrage de la liaison série
- Déclaration du type de pin pour le ventilateur
- Configuration de la lecture de PT100 en 4 fils
- Réglage de la pin d'alimentation de la photo-resistor

```
void setup() {  
  Serial.begin(115200);  
  pinMode(fan, OUTPUT); // déclaration du type de pin digitale  
  thermo.begin(MAX31865_4WIRE); // configuration en 4 fils.  
  digitalWrite(PinPowerDiode, HIGH); // alimentation de la photodiode  
  delay(1000);  
}
```

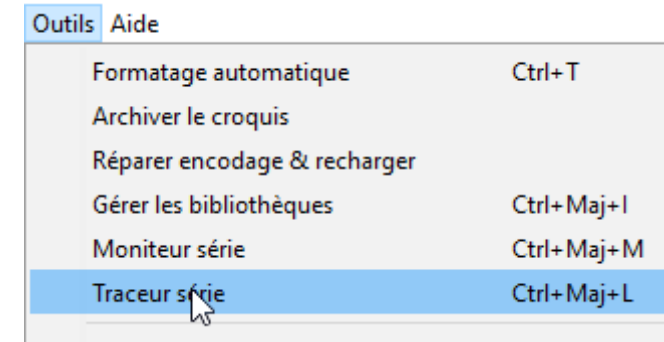
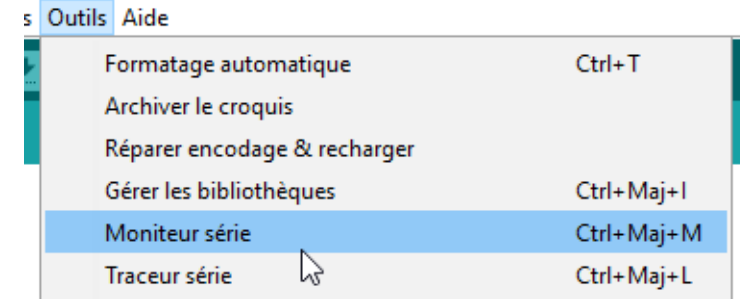
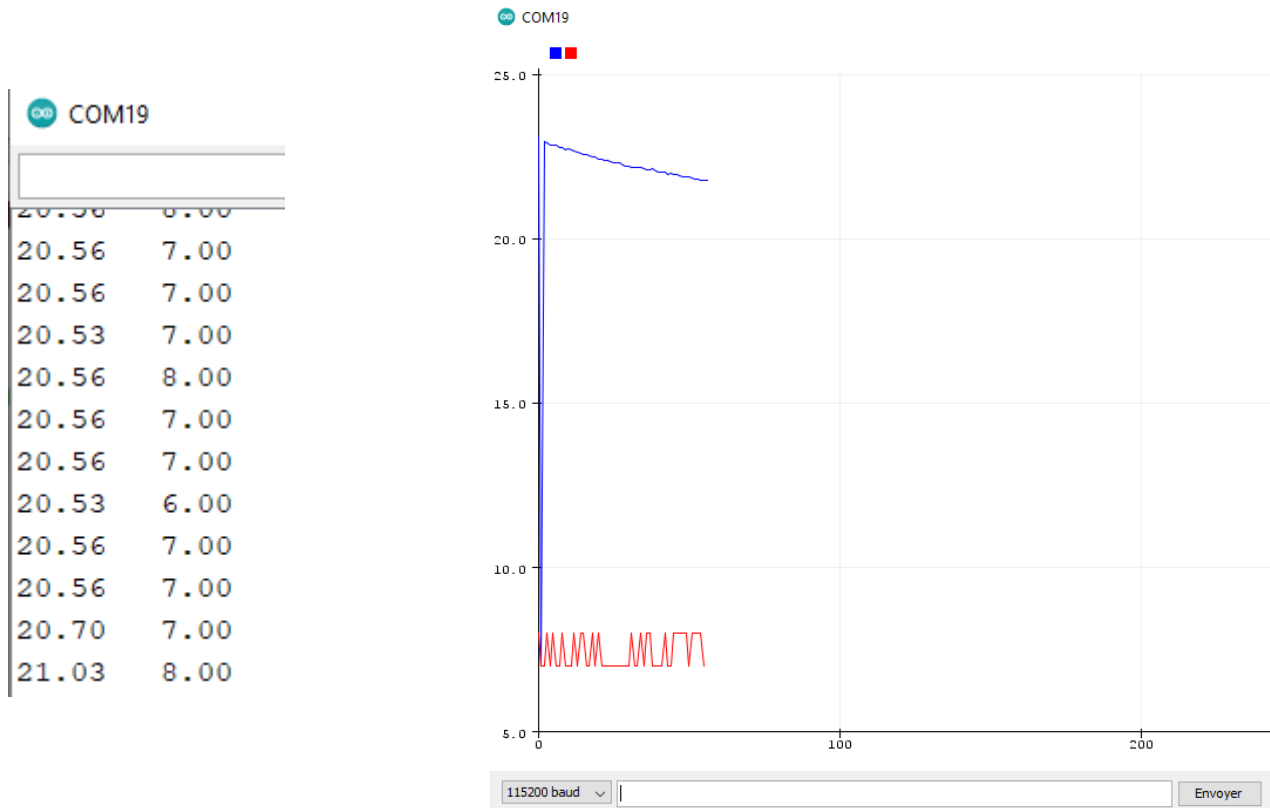

Code Arduino

- Entête décrivant brièvement le code
 - Appel de la librairie MAX31865 et utilisation pour définir les pins qui seront utilisées (10 à 13).
 - Définition des résistances de référence et nominal
 - Définition du paramètre correspondant à la température
 - Définition de la pin utilisée pour le ventilateur (8)
 - Définition des pins utilisés pour la photo-resistor (A2 et 5)
 - Définition du paramètre correspondant à l'intensité lumineuse
-
- Démarrage de la liaison série
 - Déclaration du type de pin pour le ventilateur
 - Configuration de la lecture de PT100 en 4 fils
 - Réglage de la pin d'alimentation de la photo-resistor
-
- Opération en boucle infinie
 - Lecture sur la liaison série pour détecter l'allumage/extinction du ventilateur via une commande ON/OFF
 - Récupération de la valeur de température et de l'intensité lumineuse
 - Ecriture sur la liaison série de ces valeurs
 - Attente de 1000 ms avant la prochaine mesure

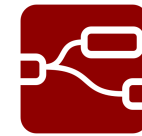
```
void loop() {  
  
    if (Serial.available()) {  
        String command = Serial.readString();  
        if (command == "ON") {  
            digitalWrite(fan, HIGH);  
            // digitalWrite(10,HIGH);  
        }  
        else if (command == "OFF") {  
            digitalWrite(fan, LOW);  
            // digitalWrite(10,LOW);  
        }  
        else if (command != ""){  
            Serial.println(command);  
        }  
    }  
  
    // valeur de la température sur la sortie série.  
    temperature = thermo.temperature(RNOMINAL, RREF);  
    ValLuz = analogRead(PinDiode);  
    Serial.print(temperature);  
    Serial.print("\t");  
    Serial.println(ValLuz);  
  
    delay(1000);  
}
```

Code Arduino

- **Vérification** sur le moniteur série ou traceur série après téléversement
- Utilisez la lampe de votre téléphone
- Réchauffez la sonde de t°



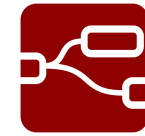
Description rapide de node-red



<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 monitoring.
- 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



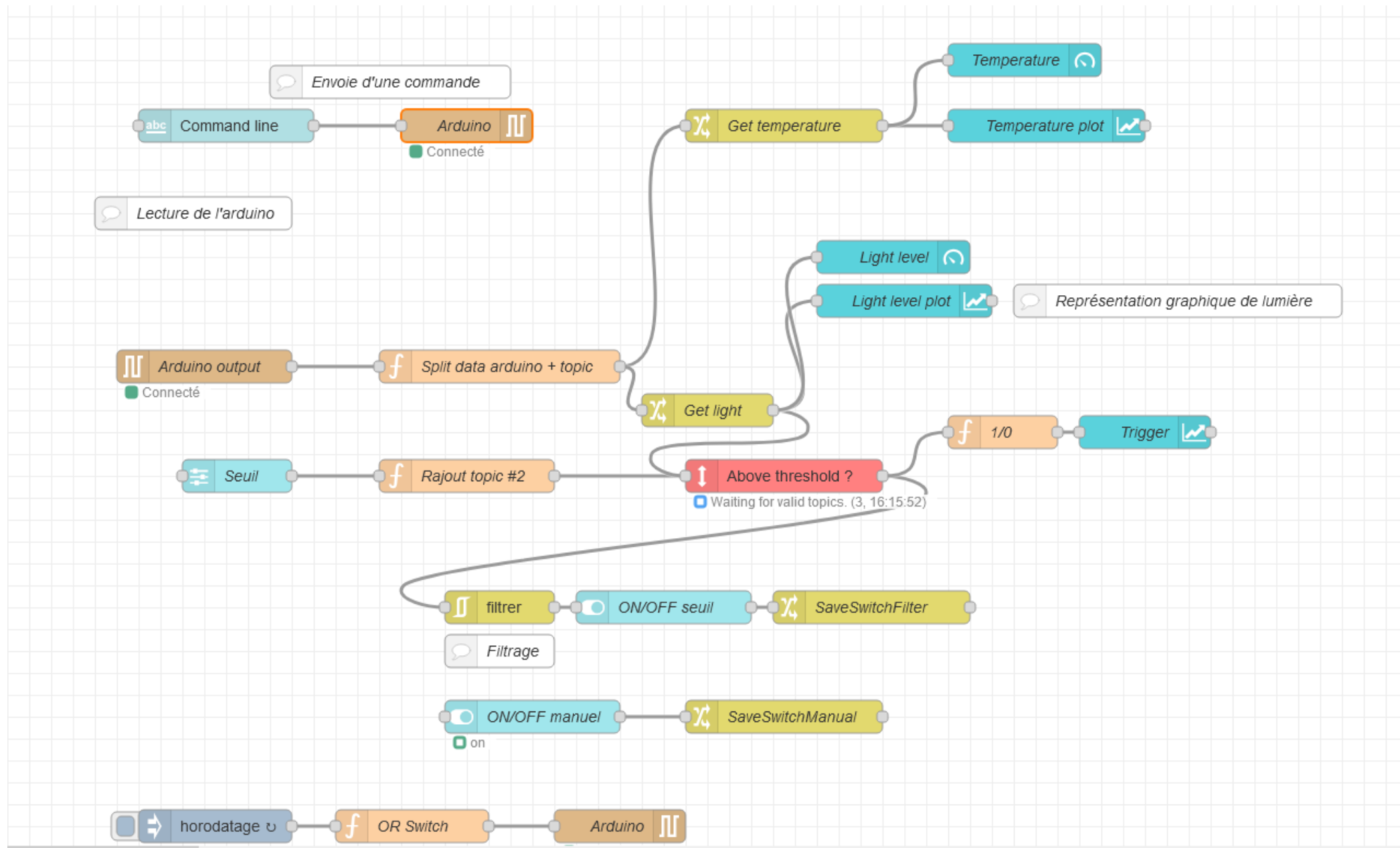
- Dans un terminal taper : *node-red*
- Dans un navigateur rentrer <http://127.0.0.1:1880>

```
barrillon@marmaille:~$ node-red
4 Dec 09:46:28 - [info]

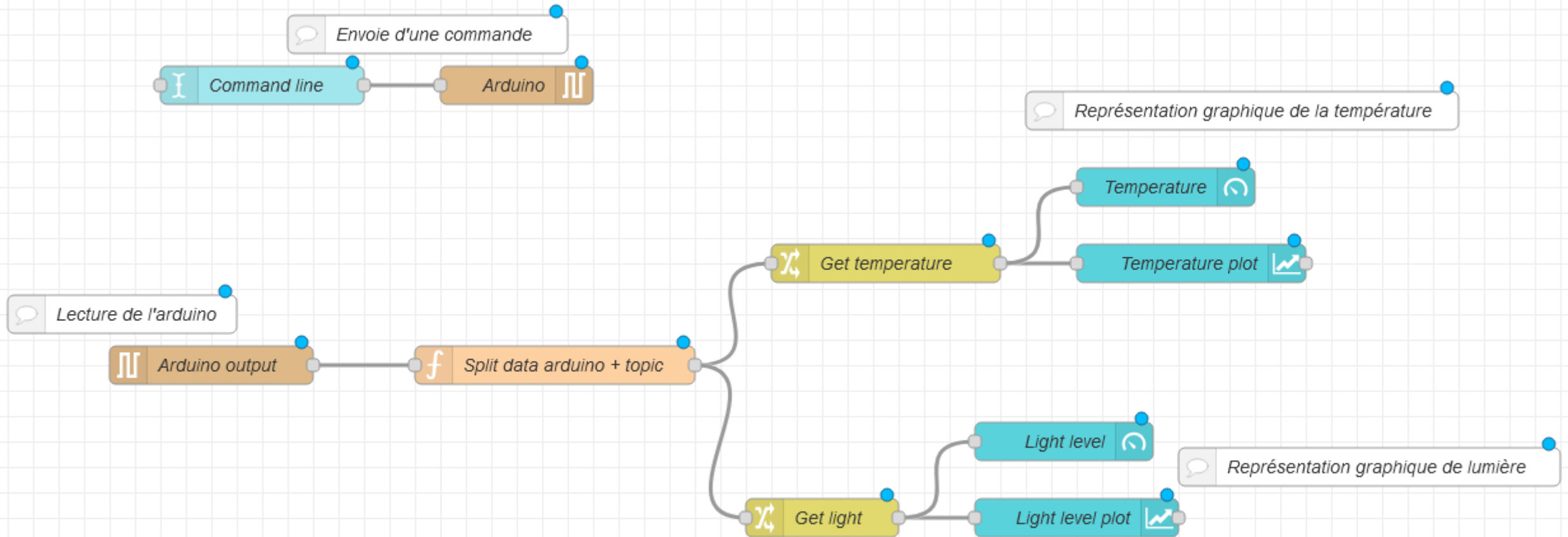
Bienvenue sur Node-RED
=====

4 Dec 09:46:28 - [info] Node-RED version: v3.1.1
4 Dec 09:46:28 - [info] Node.js version: v20.10.0
4 Dec 09:46:28 - [info] Linux 6.2.0-37-generic x64 LE
4 Dec 09:46:28 - [info] Chargement des noeuds de la palette
4 Dec 09:46:29 - [info] Fichier de paramètres : /home/barrillon/.node-red/settings.js
```

Code final espéré



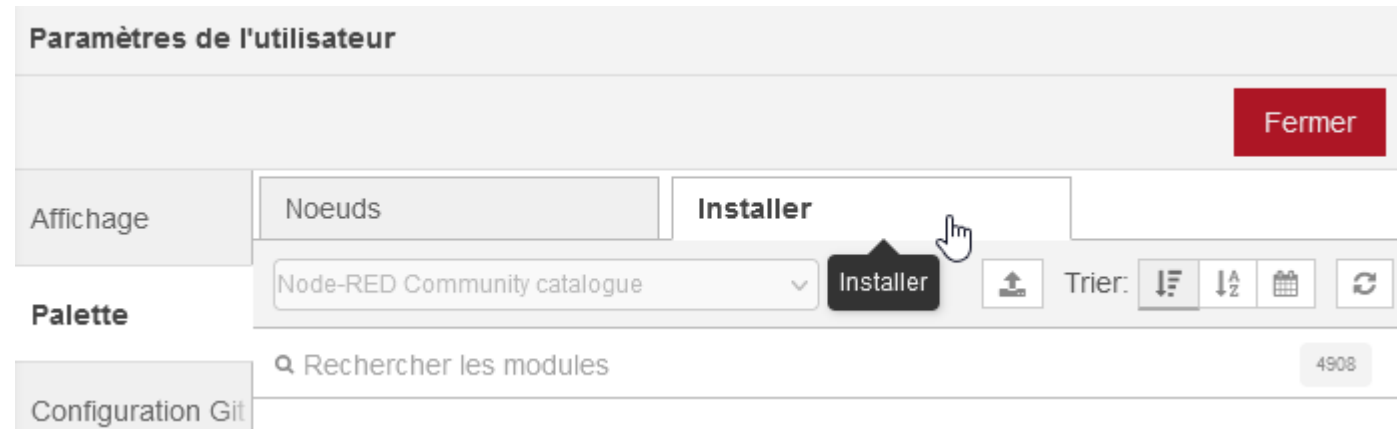
Mais on va commencer par ça !



Palette et installation de nœuds



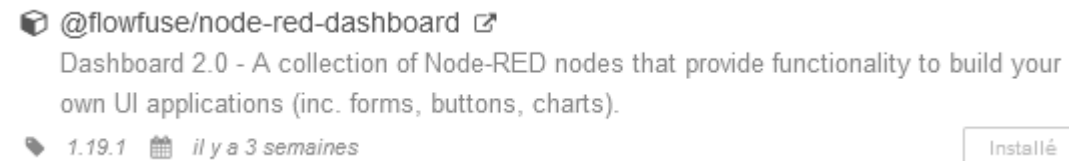
Certains nœuds font partie de bibliothèques à installer via la palette.



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**

Normalement vous avez tous sauf le dernier




Noëuds utilisés

 change Changer des paramètres

 serial in Lien série IN ou OUT

 serial out

 function Fonction codée en js

 filter Filtre

 ComparatorUI timate Comparaison (valeur > seuil)

Affichage sur dashboard

 text abc Texte

 gauge Jauge

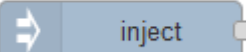
 chart Graphique

 slider Glissière (input)

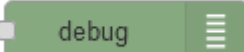
 abc text input Texte (input)

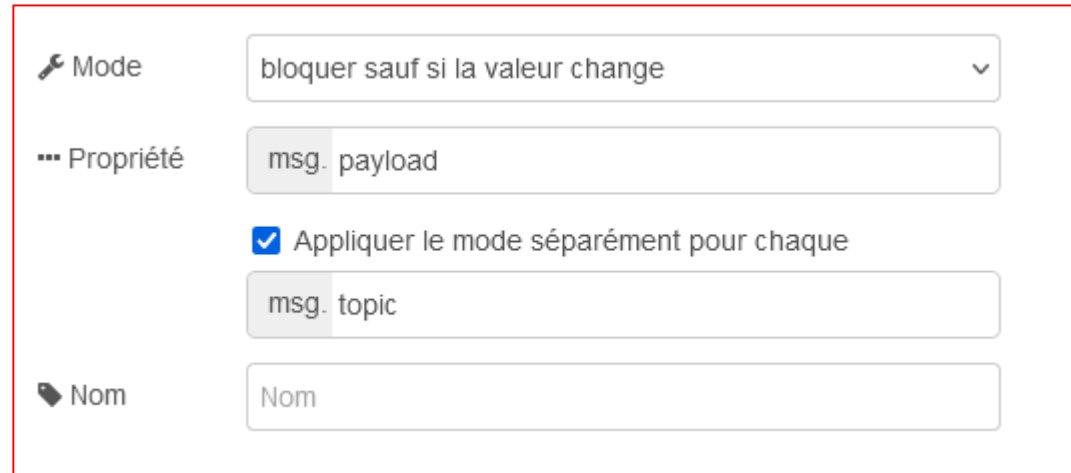
 switch Interrupteur

Injection

 inject

Debug output

 debug



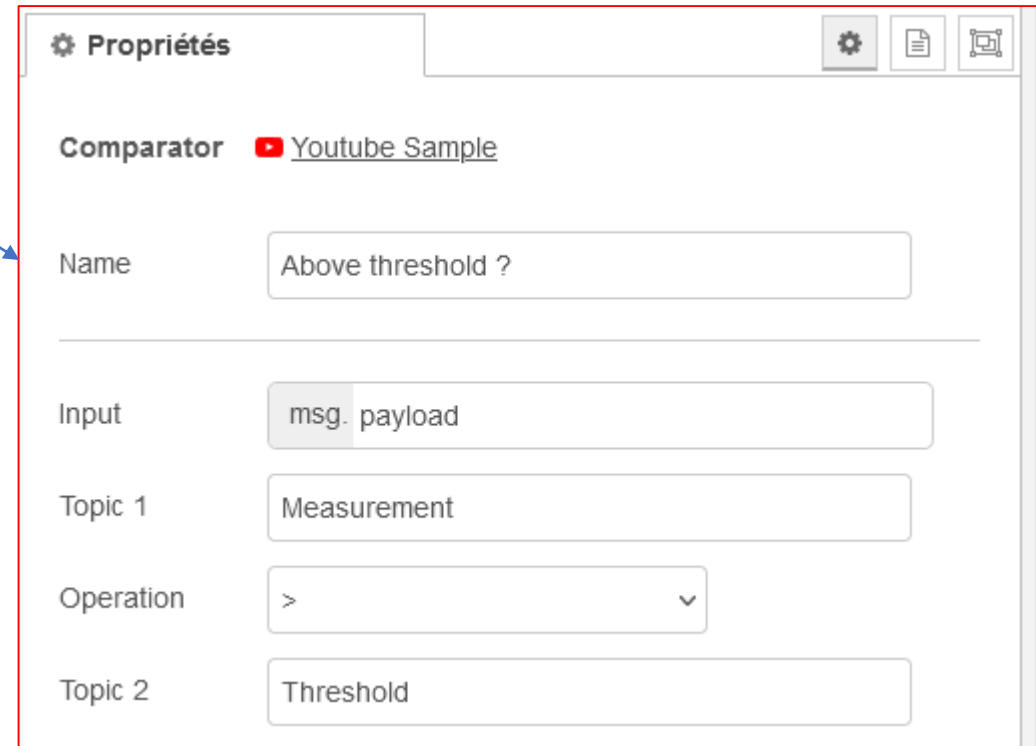
Mode: bloquer sauf si la valeur change

Propriété: msg. payload

Appliquer le mode séparément pour chaque

Propriété: msg. topic

Nom: Nom



Propriétés

Comparator: Youtube Sample

Name: Above threshold ?

Input: msg. payload

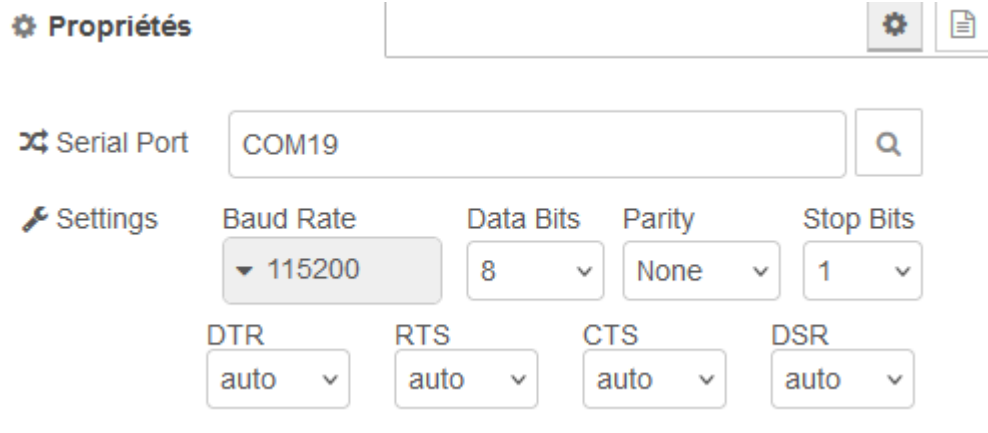
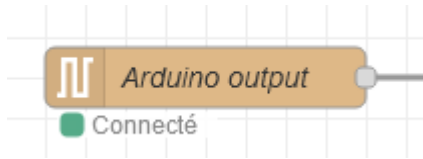
Topic 1: Measurement

Operation: >

Topic 2: Threshold

Lecture de l'arduino et affichage

Configuration sur NodeRed du lien série OUT

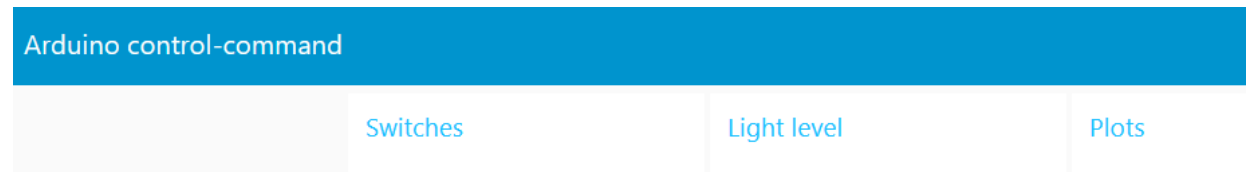
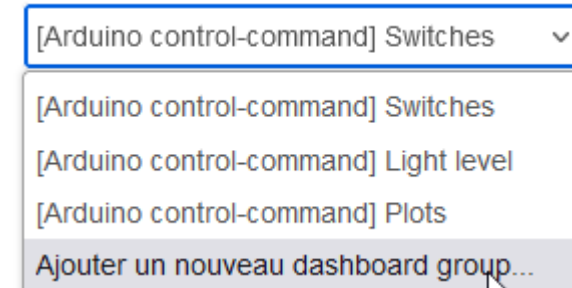


Sélectionner le bon serial port et configurer le lien.

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

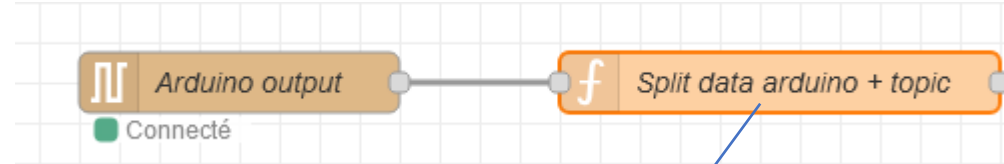


Récupérer les données transmises par l'Arduino

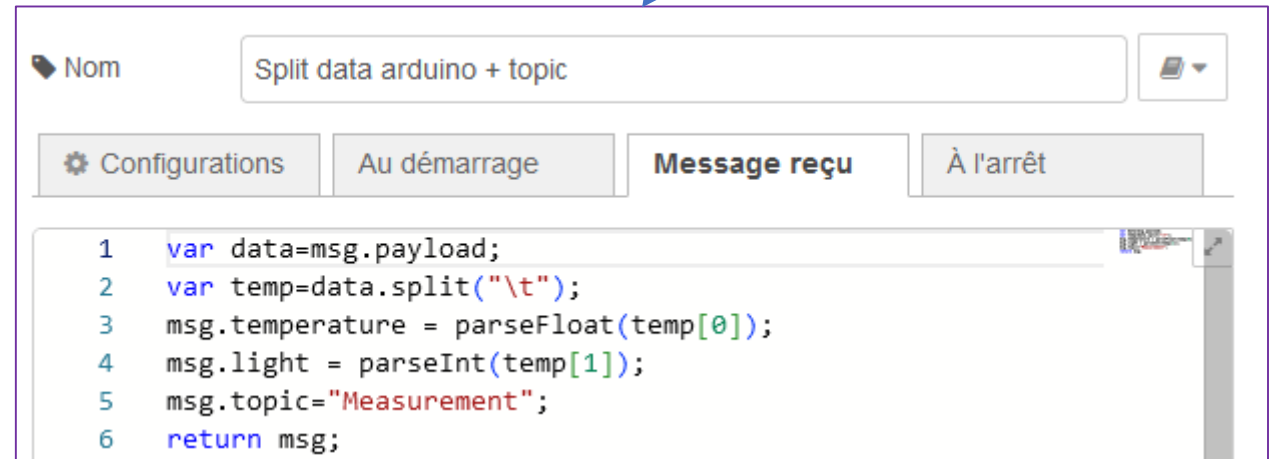
Debug permet d'afficher le msg.payload en sortie d'un block



The screenshot shows a Node-RED flow with two blocks: 'Arduino output' (orange) and 'debug 1' (green). The 'Arduino output' block is connected to the 'debug 1' block. Below the flow, the debug console is open, showing a message with the payload '22.18-27.00'. The console also shows the date and time of the message: '03/12/2024 16:20:09' and the node name: 'noeud: debug 1'.



The screenshot shows a Node-RED flow with two blocks: 'Arduino output' (orange) and 'Split data arduino + topic' (orange). The 'Arduino output' block is connected to the 'Split data arduino + topic' block.



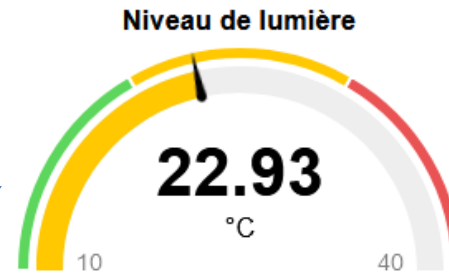
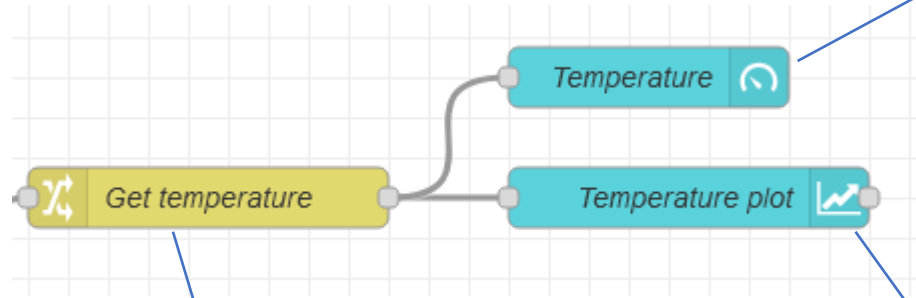
The screenshot shows the configuration panel for the 'Split data arduino + topic' block. The panel has a 'Nom' field with the value 'Split data arduino + topic'. Below the name field, there are three tabs: 'Configurations', 'Au démarrage', and 'Message reçu'. The 'Message reçu' tab is selected, and it shows the following JavaScript code:

```
1 var data=msg.payload;
2 var temp=data.split("\t");
3 msg.temperature = parseFloat(temp[0]);
4 msg.light = parseInt(temp[1]);
5 msg.topic="Measurement";
6 return msg;
```

Fonction javascript qui permet de séparer les valeurs de température et d'intensité lumineuse et leur donner un nom.

+ rajout du topic, on verra plus loin à quoi ça sert...

Affichage de la température



A configurer

Modifier le noeud gauge

Supprimer Annuler Terminer

Propriétés

- Name: Temperature
- Group: [TP Node-Red] Température et lumière
- Size: 3 x 3
- Type: Half Gauge
- Style: Needle
- Limits: Range min: 10 max: 40
- Segments: 10 (green), 20 (yellow), 30 (red)

Nom: Get temperature

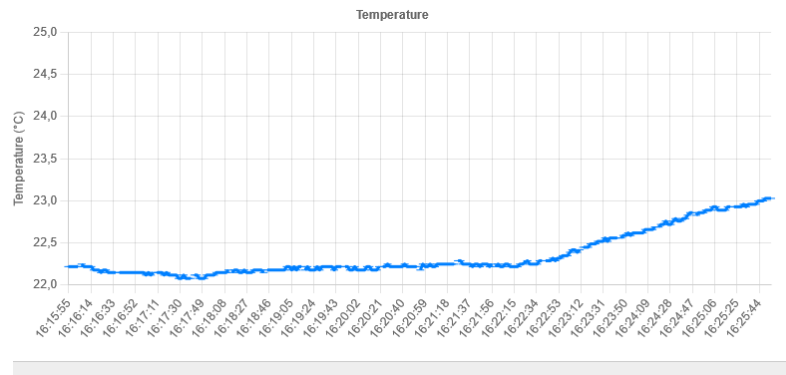
Règles

Définir sur la valeur

- msg.payload
- msg.temperature

Copie profonde de la valeur

A configurer



Propriétés

Nom: Temperature plot

Group: [TP Node-Red] Plots

Size: 6 x 6

I Label: Temperature

Class: Optional CSS class name(s)

Type: Line Show Legend

Action: Append

Point Style: Shape Line Radius (px) 4

X-Axis Type: Timescale

Format: Automatic

X-Axis Label:

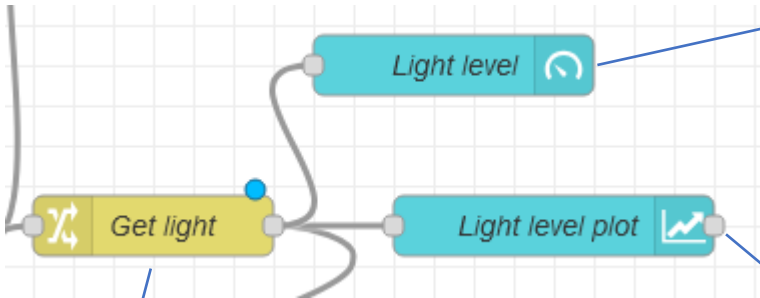
X-Axis Limit: last 10 Minutes OR 1000 points

Y-Axis Label: Temperature (°C)

Y-Axis: min: 22 max: 25

Affichage niveau de lumière

Similaire à l'affichage de la température

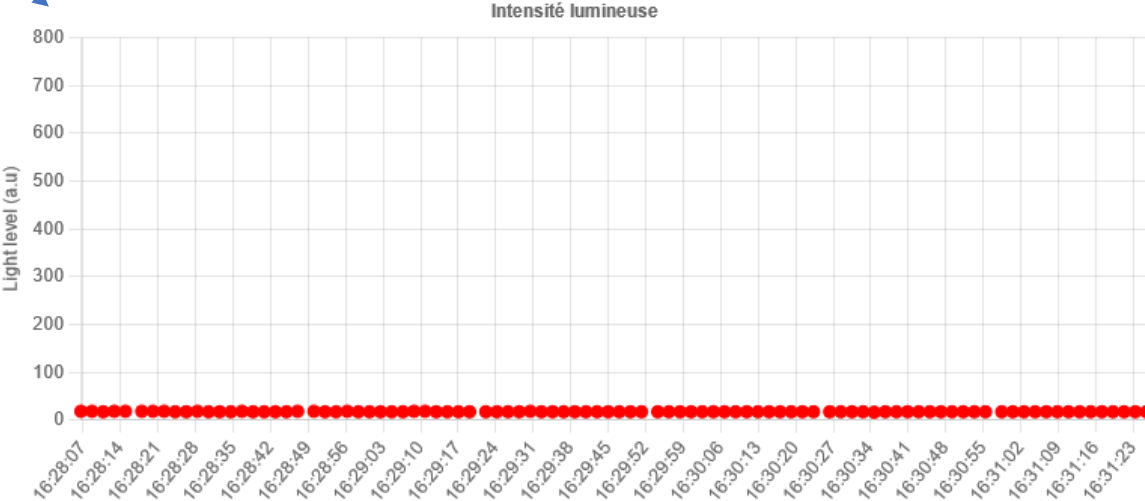


A configurer aussi !

Nom: Get light

Règles:

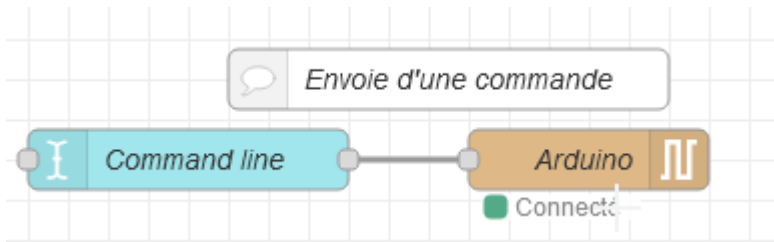
- Définir: msg. payload
- sur la valeur: msg. light
- Copie profonde de la valeur



Envoie d'une commande à l'arduino

Lien avec le ON/OFF codé dans l'arduino:

- Simple texte en input



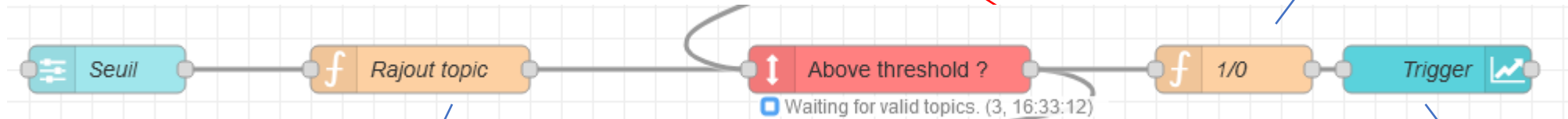
On va plus loin

- Détection d'un niveau de lumière supérieur à un seuil réglé par l'opérateur

Utilisation du topic pour comparer 2 payloads

Input	msg. payload
Topic 1	Measurement
Operation	>
Topic 2	Threshold

Réglage du seuil



Nom 1/0

Configurations Au démarrage M

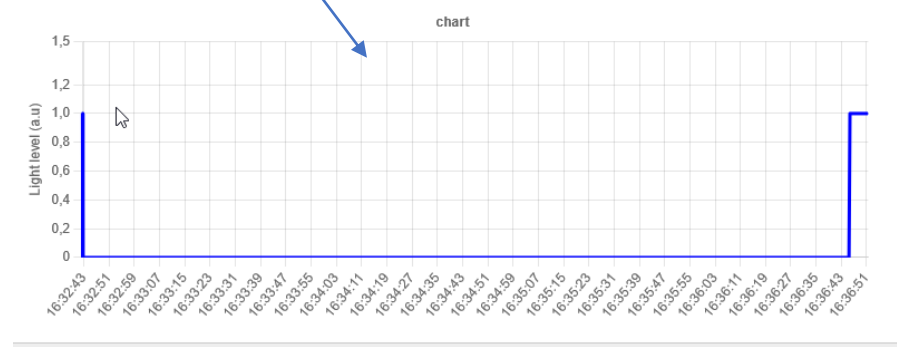
```
1 var Etat = msg.payload ;
2 if (Etat === true )
3 {
4   msg.payload= 1;
5 }
6 else if (Etat === false )
7 {
8   msg.payload= 0;
9 }
10 else
11 {
12   msg.payload = "";
13 }
14 return msg;
15
```

True → 1
False → 0

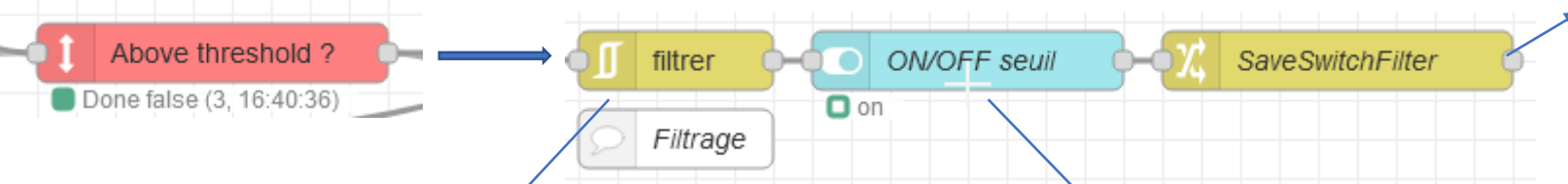
Nom Rajout topic

Configurations Au démarrage

```
1 msg.topic="Threshold";
2 return msg;
3
```



On va plus loin



Nom: SaveSwitchFilter

Règles

Définir ▼

▼ flow. switchFilter

sur la valeur ▼ msg. payload

Copie profonde de la valeur

Sauvegarde dans une variable

ON/OFF seuil

Affichage est ce que le seuil déclenche ou pas le ventil.

Propriétés

Mode: bloquer sauf si la valeur change

Propriété: msg. payload

Appliquer le mode séparément pour chaque

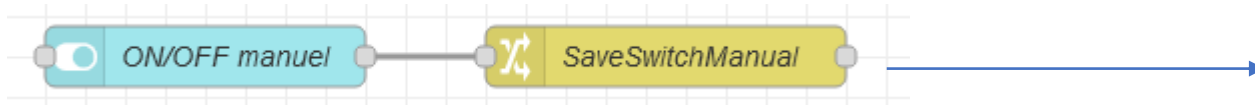
msg. topic

Nom: Nom

Fait suivre msg.payload que si la valeur change.

On va plus loin

ON/OFF du ventilateur manuel



Affichage sur dashboard

Propriétés

Nom: SaveSwitchManual

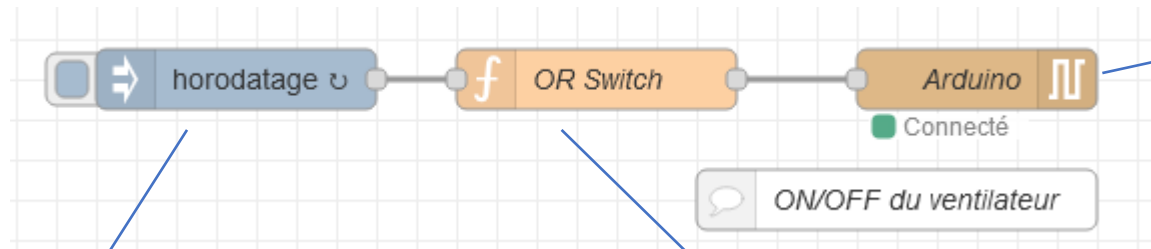
Règles

Définir ▼ flow. switchManual

sur la valeur ▼ msg. payload Copie profonde de la valeur

Sauvegarde dans une variable

On va plus loin



Envoi de la commande à l'Arduino

Répéter

intervalle secondes

Action répétée toutes les 2 secondes

Propriétés

Nom: OR Switch

Configurations: Au démarrage | Message reçu | À l'arrêt

```
1 var switchFilter = flow.get("switchFilter") || false;
2 var switchManual = flow.get("switchManual") || false;
3 var Etat = switchFilter || switchManual;
4 if (Etat === true )
5 {
6   msg.payload= "ON";
7 }
8 else if (Etat === false )
9 {
10  msg.payload= "OFF";
11 }
12 else
13 {
14  msg.payload = "";
15 }
16 return msg;
17
```

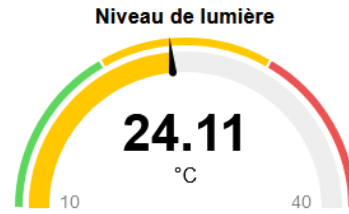
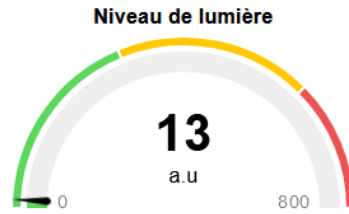
Utilisation des 2 variables pour faire un « ou » des deux.

Dashboard

☰ TP Node-Red

🏠 TP Node-Red

Température et lumière



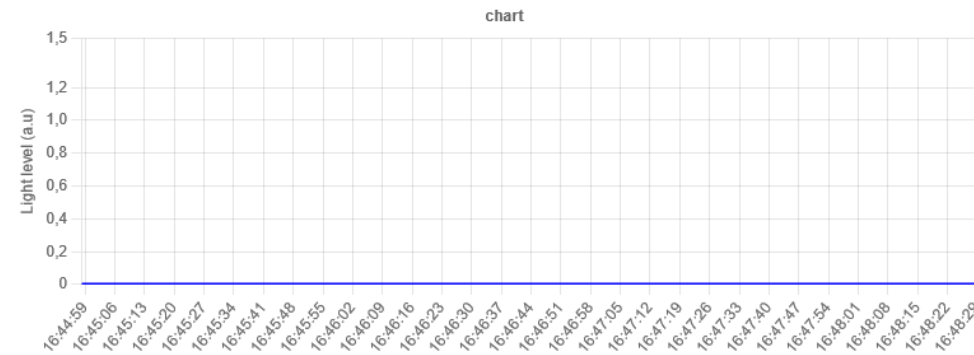
ON/OFF manuel



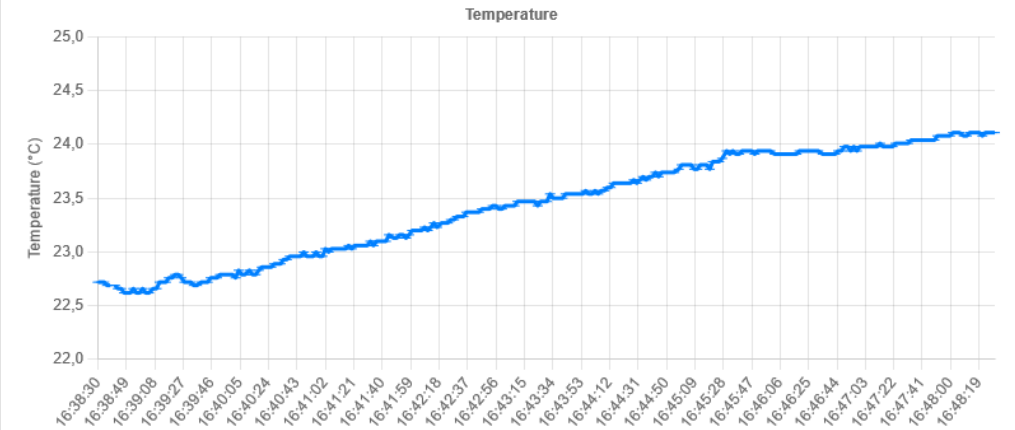
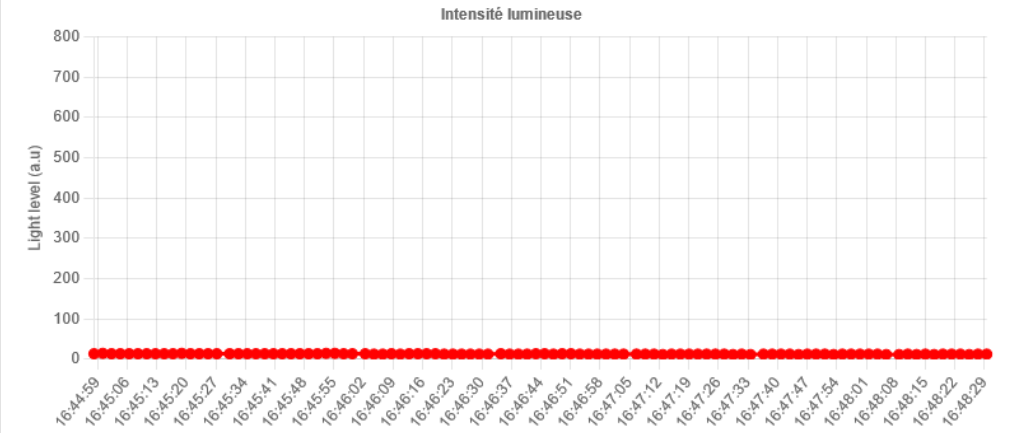
Seuil (a.u)



ON/OFF seuil



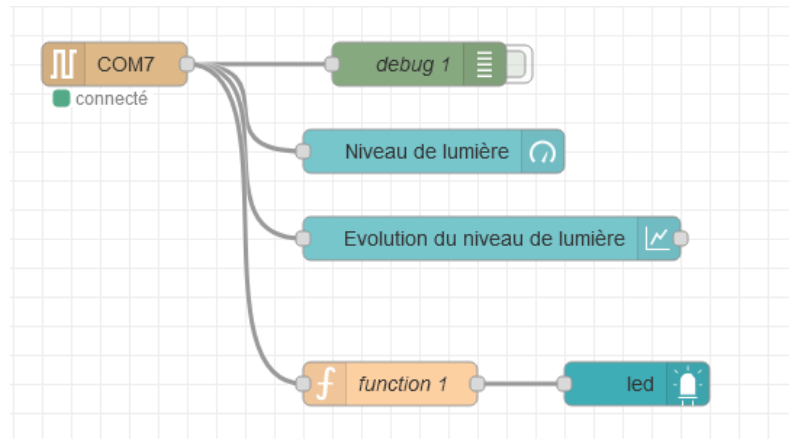
Plots





- Comment faire pour avoir une LED allumée sur le dashboard quand le seuil est dépassé ?
- Avez-vous trouvé comment faire un graphique plus grand que 5 de côté ?

Rajout d'une fonction pour détecter un niveau trop élevé



Modifier le noeud fonction

Supprimer Annuler Terminer

Propriétés

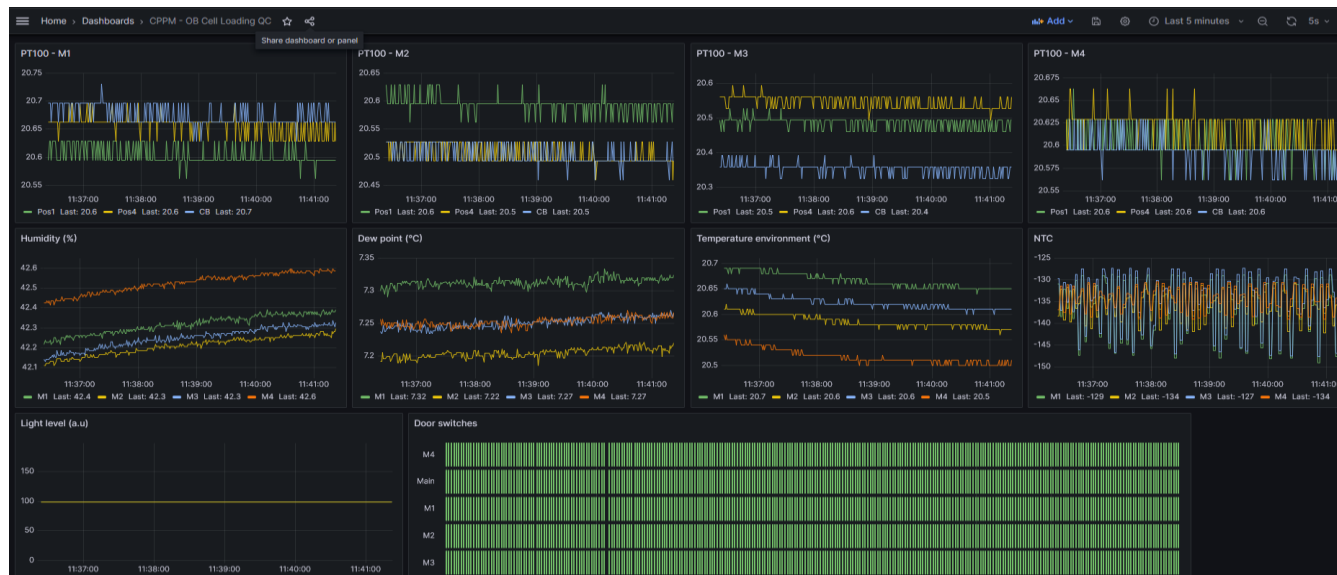
Nom: fonction 1

Configurations Au démarrage Message reçu À l'arrêt

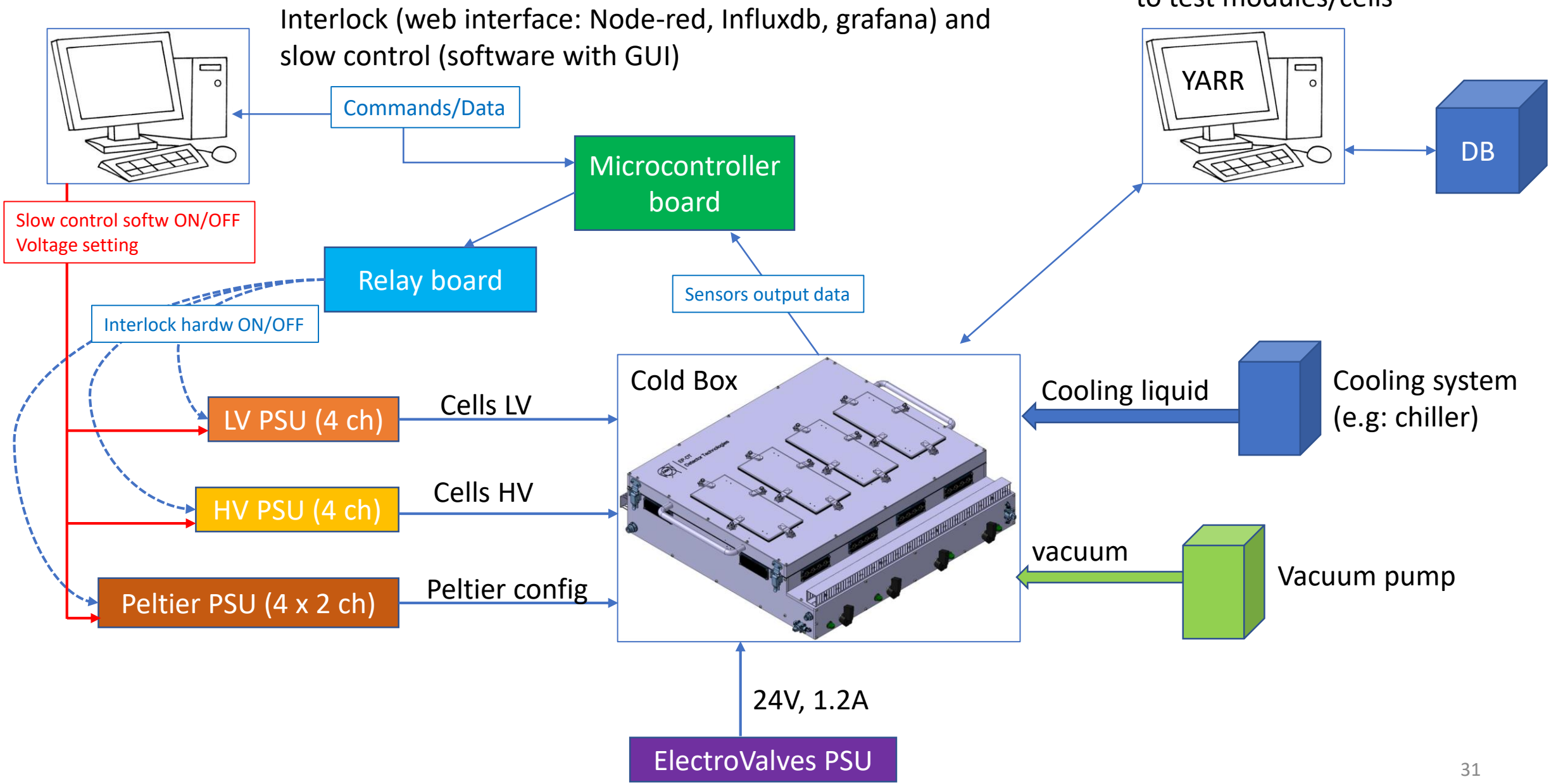
```
1 var luz = msg.payload;
2 if (luz > 300)
3 {
4     msg.payload = true;
5 }
6 else
7 {
8     msg.payload = false;
9 }
10 return msg;
```

EXEMPLE D'APPLICATION AU CPPM

Systeme de test Modules/Cells loaded



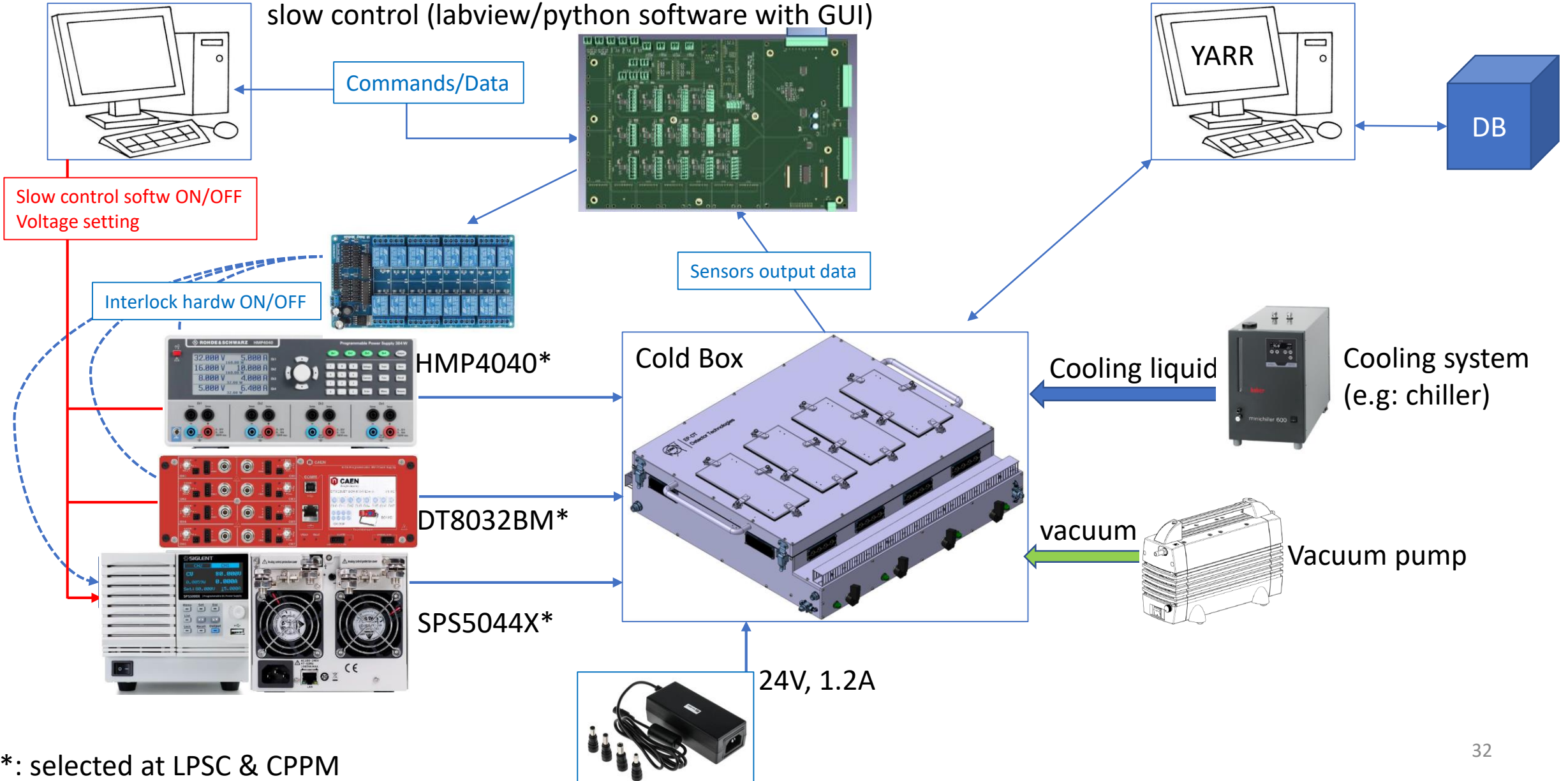
Système de test Modules/Cells loaded



Système de test Modules/Cells loaded

Interlock (web interface: Node-red, Influxdb, grafana) and slow control (labview/python software with GUI)

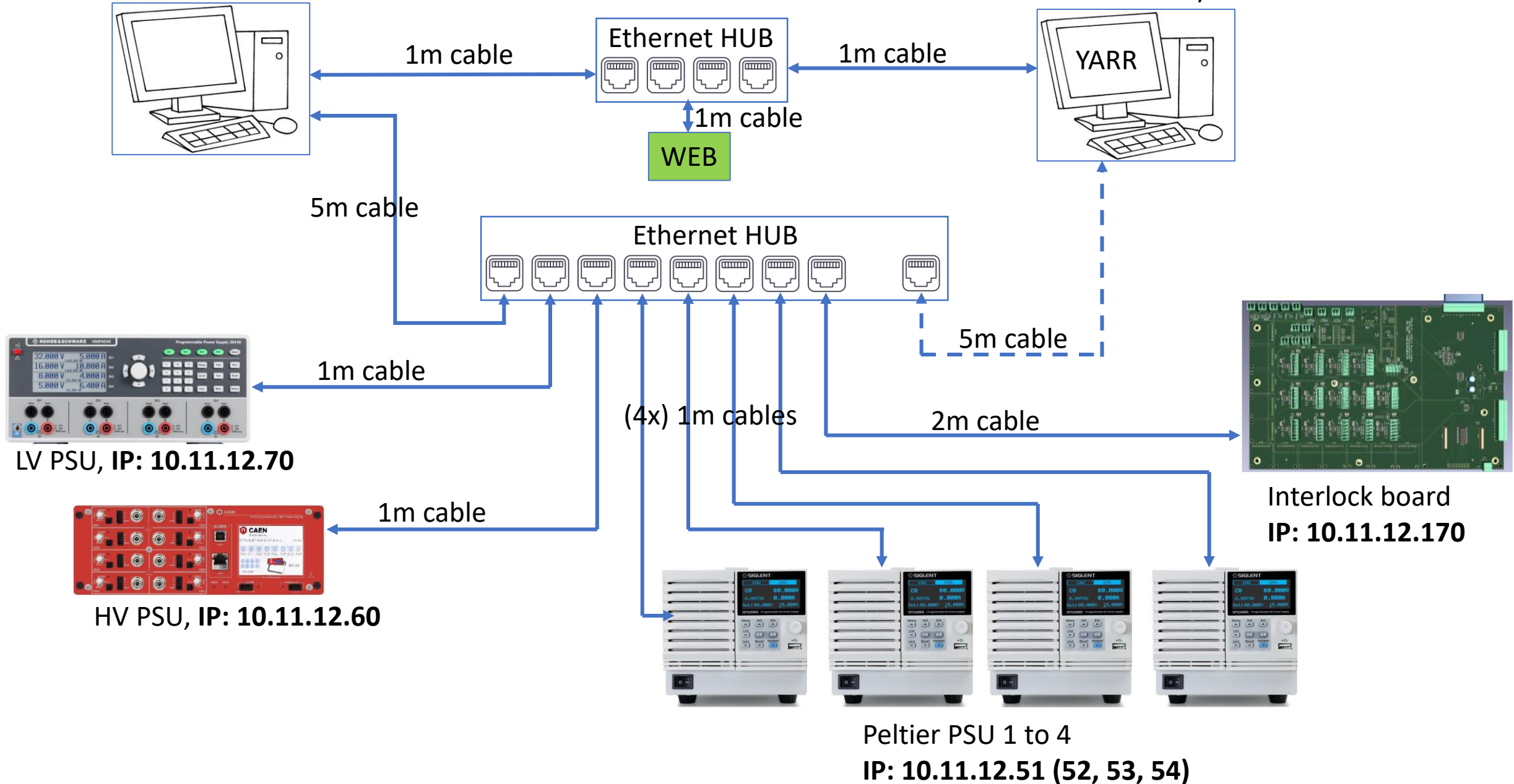
YARR PC running scripts to test modules/cells



IP addressing of loaded cells test set-up

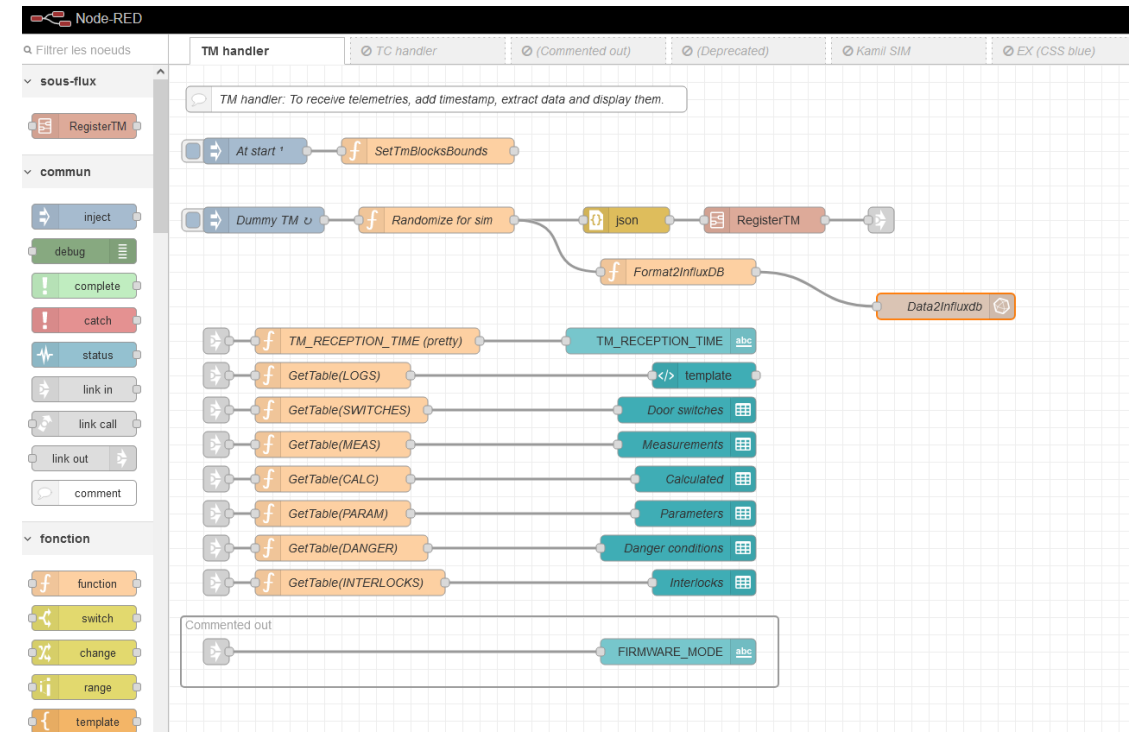
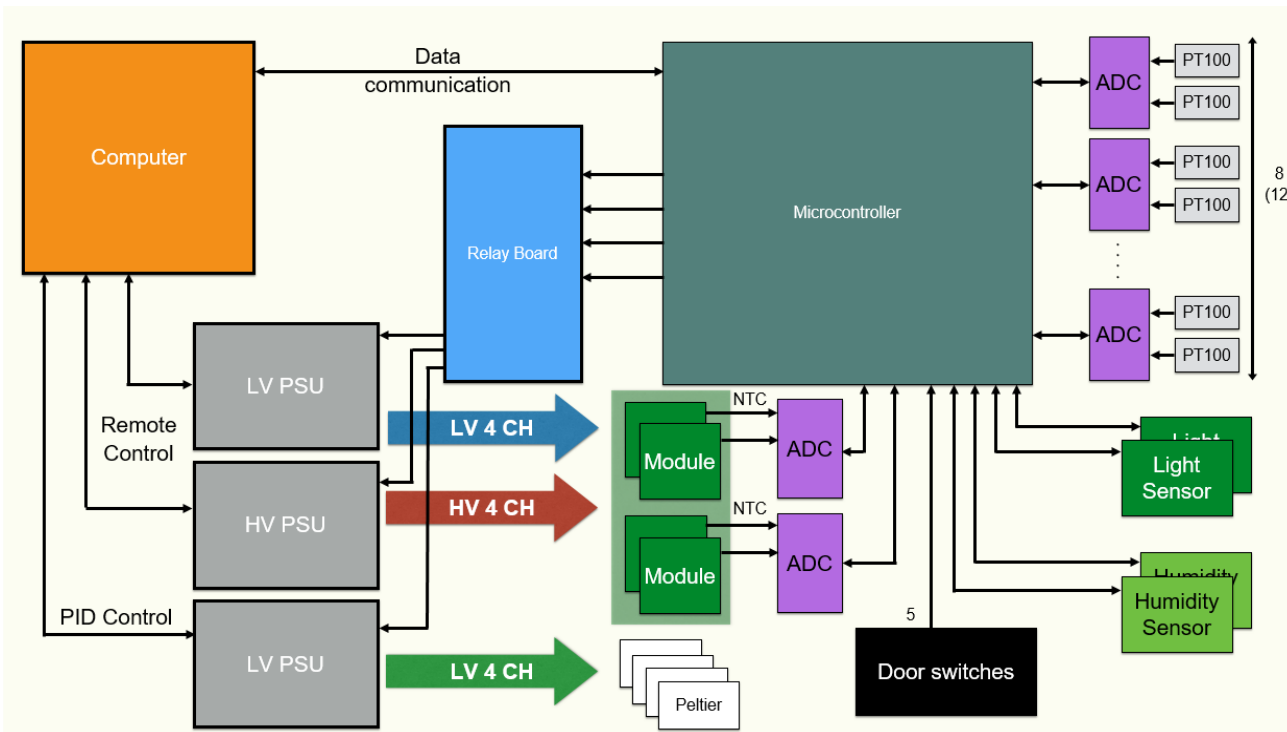
Interlock & slow control PC
IP: 10.11.12.13 – Mask: 255.255.255.0

YARR PC running scripts
to test modules/cells



Pour itk

- Le code gère les données et logs fournis par le microcontrôleur
- Il intègre leur sauvegarde sur Influxdb



Partie interlock

PT100	Va...	NTC / Light	Va...	TM_RECEPTION_TIME	RESET LOCKED-IN INTERLOCKS				
M1_PT100_11_T	20.427	M1_NTC_T	-136.6...	05/02/2024 16:41:42 1707147702007 1					
M1_PT100_14_T	20.46	M2_NTC_T	-136.0...	Dangers (actual)	Statds	Interlocks (lock-in)	Statds	Controlled devices	Statds
M2_PT100_11_T	20.46	M3_NTC_T	-134.3...	1_LIGHT_DETECTED		1_LIGHT_DETECTED		M1_PEL_OFF	
M2_PT100_14_T	20.359	M4_NTC_T	-133.82	2_DOOR_OPENED		2_DOOR_OPENED		M2_PEL_OFF	
M3_PT100_11_T	20.359	M12_LIGHT_ADC	0	3_M1_PT100_TOO_HOT		3_M1_PT100_TOO_HOT		M3_PEL_OFF	
M3_PT100_14_T	20.393	M34_LIGHT_ADC	0	3_M2_PT100_TOO_HOT		3_M2_PT100_TOO_HOT		M4_PEL_OFF	
M4_PT100_11_T	20.46			3_M3_PT100_TOO_HOT		3_M3_PT100_TOO_HOT		M1_HV_OFF	
M4_PT100_14_T	20.46			3_M4_PT100_TOO_HOT		3_M4_PT100_TOO_HOT		M2_HV_OFF	
				4_M1_NTC_SUPER_HOT		4_M1_NTC_SUPER_HOT		M3_HV_OFF	
				4_M2_NTC_SUPER_HOT		4_M2_NTC_SUPER_HOT		M4_HV_OFF	
				4_M3_NTC_SUPER_HOT		4_M3_NTC_SUPER_HOT		M1_LV_OFF	
				4_M4_NTC_SUPER_HOT		4_M4_NTC_SUPER_HOT		M2_LV_OFF	
				5_DEWPOINT_REACHED		5_DEWPOINT_REACHED		M3_LV_OFF	
				6_LOW_BOARD_VOLTAGE		6_LOW_BOARD_VOLTAGE		M4_LV_OFF	
				7_WATCHDOG_RESTART		7_WATCHDOG_RESTART			
				8_PC_COM_LOST		8_PC_COM_LOST			
				9_NOT_ENOUGH_DRY_AIR		9_NOT_ENOUGH_DRY_AIR			
				10_NOT_ENOUGH_VACUUM		10_NOT_ENOUGH_VACUUM			
Ambient	Va...	Dewpoints	Va...						
M1_ENV_T	20.44	M1_ENV_DEWT	9.358						
M1_ENV_RH	49.293	M2_ENV_DEWT	9.277						
M2_ENV_T	20.4	M3_ENV_DEWT	9.294						
M2_ENV_RH	49.145	M4_ENV_DEWT	9.213						
M3_ENV_T	20.47								
M3_ENV_RH	48.979								
M4_ENV_T	20.36								
M4_ENV_RH	49.053								
Door switches	St...								
DSWITCH_1_OPEN	Closed								
DSWITCH_2_OPEN	Closed								
DSWITCH_3_OPEN	Closed								
DSWITCH_4_OPEN	Closed								
DSWITCH_5_OPEN	Closed								
				Teensy Notifications (Latest first)					
				Time		Message			
				05/02/2024 16:39:28		Legal TC received. Resetting all interlocks.			
				05/02/2024 16:38:32		Starting. Setup complete. Now protecting. Custom parameters reset to default.			
				05/02/2024 16:38:32		Starting. Setup complete. Now protecting. Custom parameters reset to default.			
				05/02/2024 16:38:32		Starting. Setup complete. Now protecting. Custom parameters reset to default.			

Partie slow control Pilotage des 3 types d'alimentation

The interface displays four modules (Module 1 to Module 4) and a common section. Each module has a 'SOM OK' indicator, 'FORCE LISTEN', 'POWER ON', and 'POWER OFF' buttons. Below these are 'Enable channel 1 (TOP)' and 'Enable channel 2 (BOTTOM)' toggle switches. The 'Temp. regulation' section for each module includes:

- Dew point: 6.8 °C (Module 1), 6.7 °C (Module 2), 6.9 °C (Module 3), 6.9 °C (Module 4)
- Module: 25.0 °C
- Coolblock: 20.7 °C
- PID regulation enable: toggle switch
- Use the Coolblock (PT100) like T° reference: dropdown menu
- T° Target: slider set to 20
- Use 12°C above dew point like set point: toggle switch
- PT100 (cooling block temp.): 0 °C

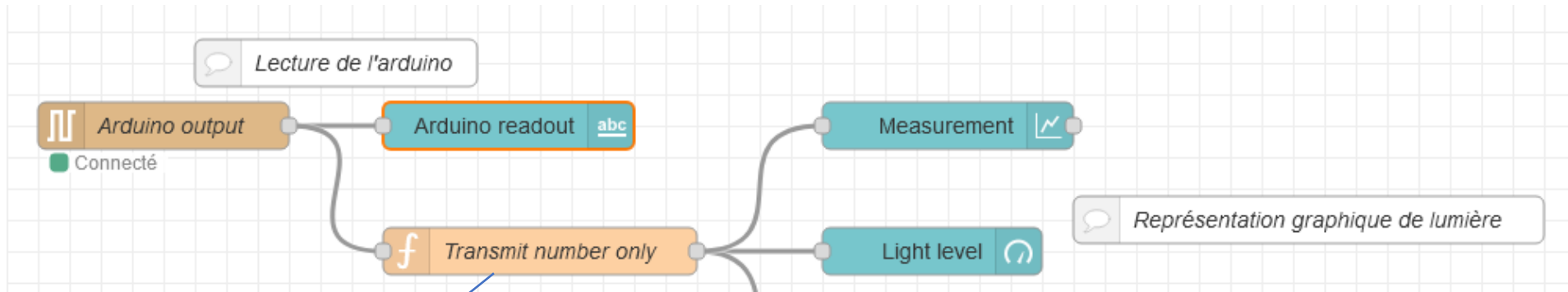
The 'Common' section includes:

- 'SOM OK' indicator, 'FORCE LISTEN', and 'CLEAR ALARM' buttons.
- Ramp Up and Ramp Down sliders.
- Ramp on Power Down toggle switch.
- Four PSU channels (Module 1 to 4) with status indicators (On, FP Interlocked, Ramping Up, Ramping Down, KILL (qcxlock), FP Disabled, Local Ctrl, Over cur.) and analog gauges for Voltage Measured and Current Measured.
- Each channel has 'Voltage' and 'Current' dropdown menus with 'SET' buttons.
- 'ON' and 'OFF' buttons for each channel.

BACK-UP

Affichage niveau de lumière

Supprimer les NaN via une fonction en javascript



Modifier le noeud fonction

Supprimer Annuler Terminer

Propriétés

Nom Transmit number only

Configurations Au démarrage Message reçu À l'arrêt

```
1 var output = parseFloat(msg.payload) ;
2 if (!Number.isNaN(output))
3 {
4     msg.payload= output;
5     msg.topic="Measurement"
6 }
7 else
8 {
9     msg.payload = "";
10 }
11 return msg;
12
```

```
var output = parseFloat(msg.payload) ;
if (!Number.isNaN(output))
{
    msg.payload= output;
    msg.topic="Measurement"
}
else
{
    msg.payload = "";
}
return msg;
```