

## Activité : Création d'une mini-station météo connectée sur serveur Node JS

### Éléments de correction (document professeur)

- 1) **Compléter** le schéma de câblage du mini-projet en complétant le schéma du document réponse 1 donné en fin d'activité. **Voir DR1**
- 2) **Compléter** la chaîne d'information du mini-projet avec les blocs : « acquérir », « traiter », « communiquer » et « transmettre » en complétant la chaîne d'information du document réponse 2 donné en fin d'activité. **Voir DR2**
- 3) **Ajouter** sur la chaîne d'information précédemment réalisée le nom du matériel répondant à chaque fonction : acquérir : « capteur TM36 » ou « capteur LM35 » et « capteur LDR », traiter et communiquer : « carte Arduino Uno » et transmettre : « carte Raspberry Pi ». **Voir DR2**
- 4) **Ajouter** sur la chaîne d'information précédemment réalisée les flux d'informations : « tension analogique », « nombre numérique », « nombre numérique JSON ». **Faire valider** par le professeur. **Voir DR2**

### Partie prototypage :

- 5) **Acquérir** simplement avec une carte Arduino Uno les grandeurs de température et de luminosité issues des capteurs en vous aidant des activités faites précédemment.

**Voir DR3 et programme.ino**

**Traiter** les grandeurs en effectuant leurs mises à l'échelle afin d'obtenir leurs valeurs en Degré-Celsius et en Lux. **Voir DR3 et programme.ino**

**Afficher** les valeurs de température et de luminosité sur le moniteur série de votre PC.

**Voir DR3 et programme.ino**

- 6) **Transformer** les grandeurs de température et de luminosité affichées sur le moniteur série au format JSON en vous aidant du document ressource 1. **Afficher** les données au format JSON sur le moniteur série de votre PC. **Voir DR3 et programme.ino**

- 7) **Valider** la conformité des données JSON en copiant (« CTRL C ») les données JSON affichée par le moniteur série et en les collant (« CTRL V ») sur le site web : <https://jsonlint.com/>

Appuyer sur le bouton ValidateJ SON pour contrôler la conformité des données :

The image shows a screenshot of the JSONLint website. On the left, a text area contains a JSON object: 

```
{
  "TMP36": 17.38,
  "LDR": 416.50
}
```

. A red arrow points from this text area to a button labeled "Validate JSON". To the right of the button, the text "Le site web indique que les données sont conformes :" is displayed in red. Below this text, there is a box titled "Results" which contains a green bar with the text "valid JSON".

## Activité : Création d'une mini-station météo connectée sur serveur Node JS

8) **Connectez-vous** en local sur le serveur Node-Red d'une carte Raspberry Pi avec l'URL donnée par le professeur : « IpRaspberry Pi:1880 », par exemple : « 10.10.8.70:1880 ».

Suivre le [tutoriel guide\\_RaspberryPi](#) en fichier lien pour configurer correctement la Raspberry Pi. C'est le professeur qui configure toutes les Raspberry Pi préalablement. Les élèves ne font que programmer sous Node Red.

Je vous conseille de configurer une première Raspberry Pi à votre domicile sur votre réseau local (LAN). Votre routeur (« box ») servira également de passerelle pour que la mini-station météo soit accessible depuis le réseau mondial (WAN). Configurer une Raspberry Pi au lycée est source d'ennuis avec le réseau local et le proxy du lycée, vous gagnerez du temps à le faire chez vous.

Une fois toutes les configurations de base de la carte Raspberry Pi réalisées, vous pouvez utiliser le logiciel Win32DiskImager pour réaliser des copies de la carte microSD pour l'ensemble des Raspberry Pi de vos groupes d'élèves.

La seule modification de configuration à réaliser sera de changer l'adresse IP de chaque Raspberry Pi et du routeur afin d'être sur le réseau du lycée (voir [tutoriel guide\\_RaspberryPi](#)).

### A votre domicile :

interface eth0  
static ip\_address=192.168.0.70/24  
static\_routers=192.168.0.254  
static\_domain\_name\_servers=192.168.0.254

### Au lycée :

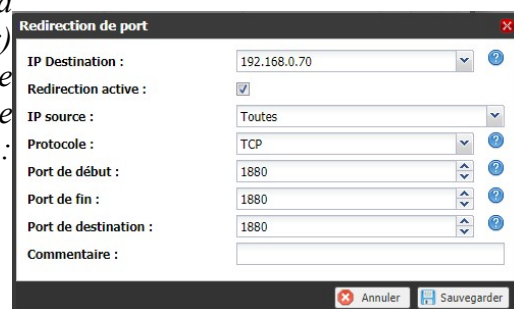
interface eth0  
static ip\_address=10.10.8.60/8  
static\_routers=10.128.1.62  
static\_domain\_name\_servers=10.128.1.62

9) **Réaliser** le programme Node-Red en vous aidant du document ressource 2 : programmation sous Node-Red. [Suivre le tutoriel présent dans le document activité\\_node\\_redélève.](#)

10) Pour une connexion en local : **Connecter** vous avec votre navigateur sur la page web du serveur Node-Red de la carte Raspberry Pi avec l'url donné par le professeur pour afficher la température et la luminosité : « IpRaspberry Pi:1880/ui » par exemple : « 10.10.8.70:1880/ui ».

*Pour une connexion depuis internet, il faut mettre la Raspberry Pi sur le réseau local de votre routeur (box) ADSL et réaliser une redirection de port comme lors de l'activité : « mise en réseau d'une caméra IP ». Ensuite connecter vous avec votre navigateur avec l'url : « IPexterne\_box:1880/ui »  
Par exemple : « 55.160.205.187:1880/ui ».*

**Faire valider** par le professeur.



Suivre le [tutoriel guide\\_RaspberryPi](#) en fichier lien.

## Activité : Création d'une mini-station météo connectée sur serveur Node JS

### Document Réponse DR1 : Schéma de câblage de la mini-station météorologique connectée

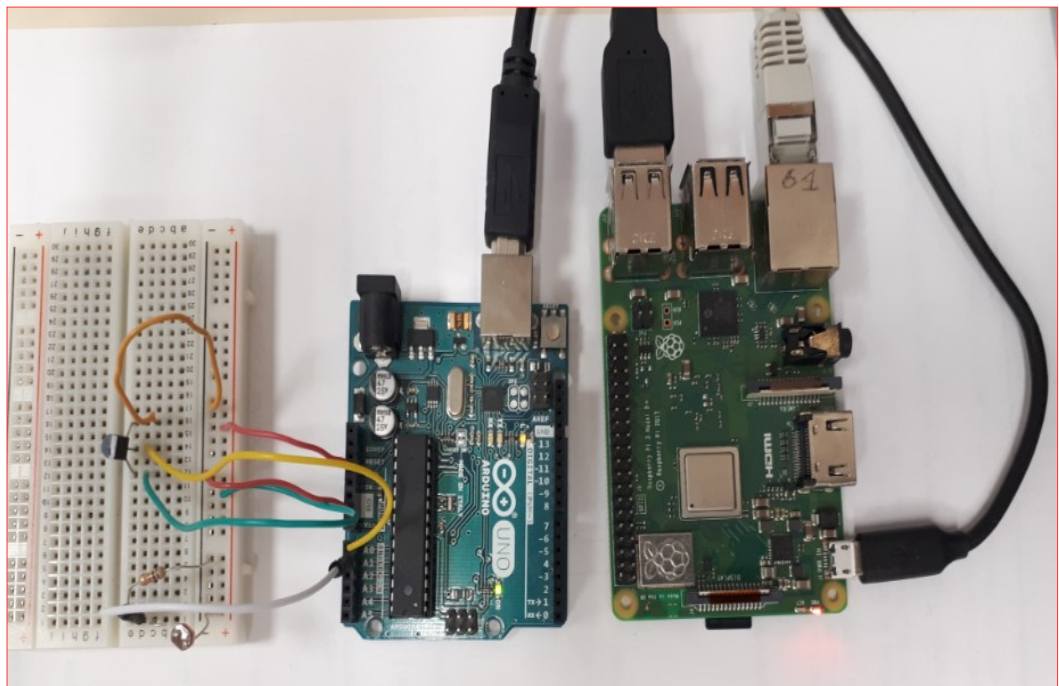
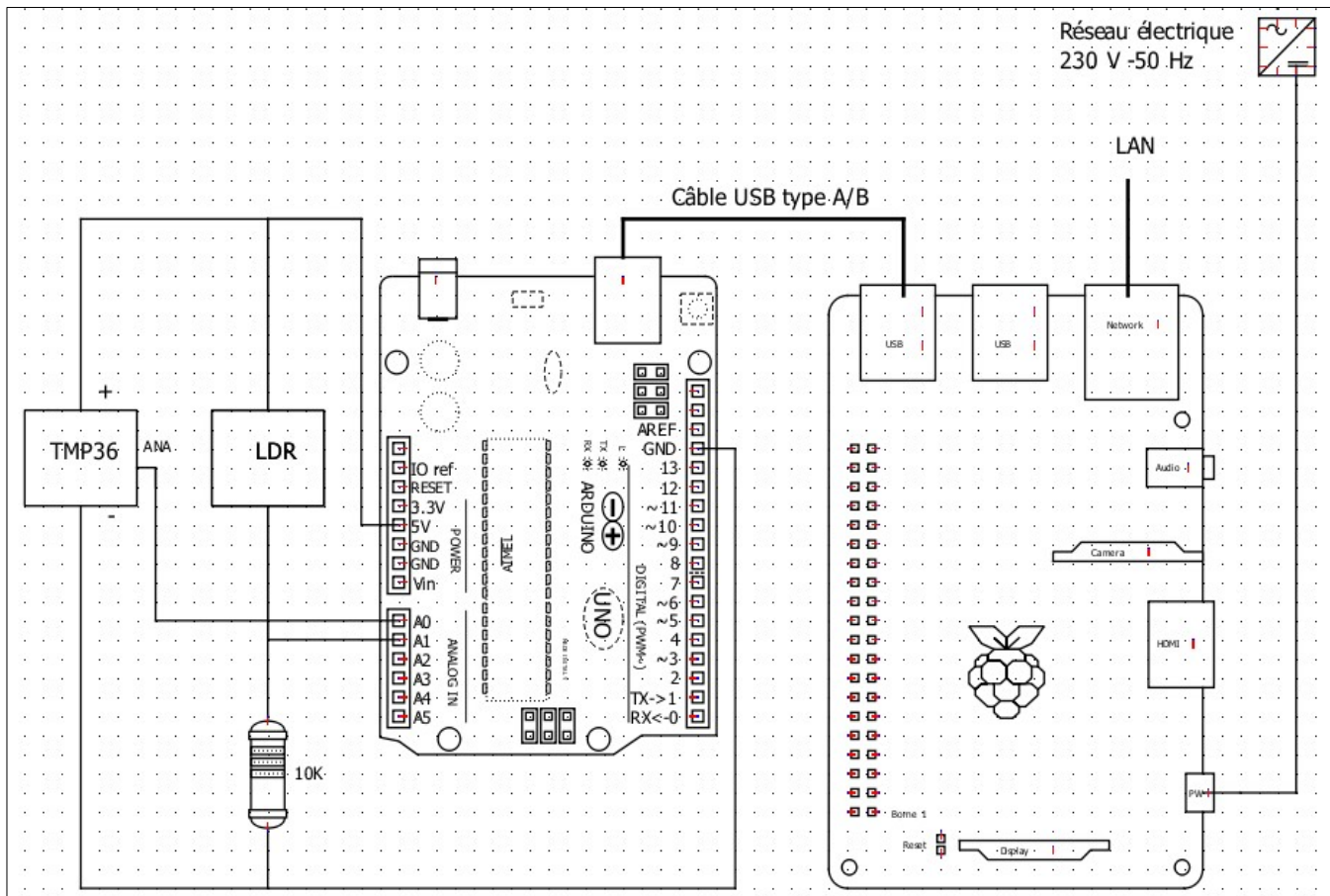
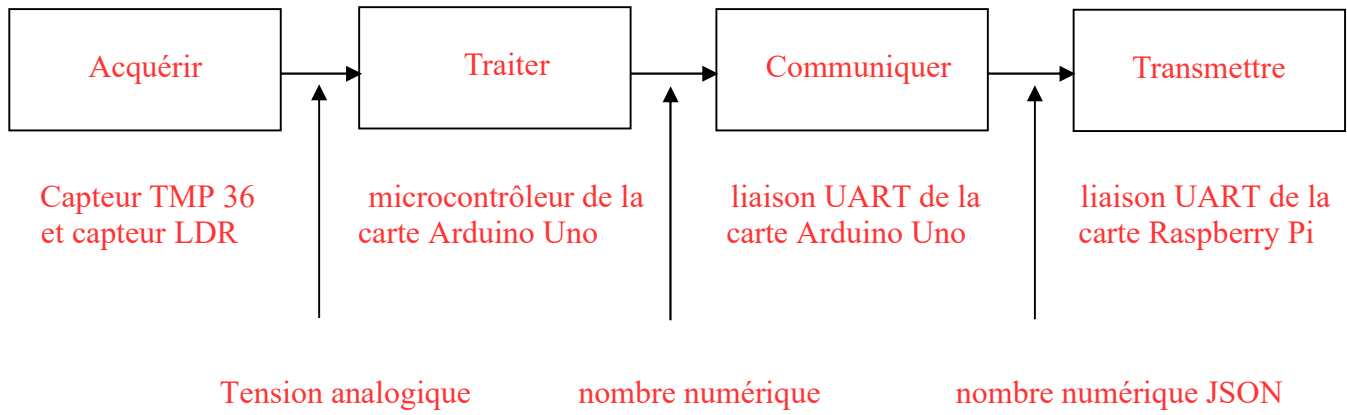


Photo du montage

## Activité : Création d'une mini-station météo connectée sur serveur Node JS

### Document Réponse DR2 : Chaîne d'information de la mini-station météorologique connectée



## Activité : Création d'une mini-station météo connectée sur serveur Node JS

### Document Ressource DR3 : Programmation sous Arduino

#### Programme Arduino avec les commentaires pour comprendre chaque instruction :

```

/*/////////////////////////////////////////////////////////////////
Programme : Mini-station météo sur Serveur Node-Red
Date : 27/02/2023
Nom du programmeur : PIERRON Benoit
Lycée : Jean Jaurès, 12400 Saint Affrique
////////////////////////////////////////////////////////////////*/

void setup()//Création de la fonction d'initialisation
{
  Serial.begin(115200);//Initialisation de la liaison UART (moniteur série) avec un débit de 115200 bauds
}

void loop()//Création de la fonction principale
{
  int a0 = analogRead(A0);/*Création de la variable (nombre entier) a0 qui va contenir la valeur du nombre
numérique compris entre 0 et 1023 qui est l'image de la tension analogique de la broche A0*/
  float voltage = (a0 / 1023.0) * 5.0; /*Création de la variable voltage (nombre décimal) qui calcul
la valeur de la tension analogique de la broche A0 en multipliant par le quantum du CAN*/
  float temp = (voltage - .5) * 100; /*Création de la variable temp (nombre décimal) qui calcul la valeur
de la température issue du capteur TMP36. On enlève 0.5 V pour l'Offset du capteur (500mV pour 0 degré)
et on multiplie par 100 car le capteur donne 10 mV par degré.*/
  int a1 = analogRead(A1);/*Création de la variable (nombre entier) a1 qui va contenir la valeur du nombre
numérique compris entre 0 et 1023 qui est l'image de la tension analogique de la broche A1*/
  float eclair = (a1*0.7); /*Création de la variable eclair (nombre décimal) qui calcul la valeur de la
température issue du capteur LDR. On multiplie par 0.7 pour simplifier la mise à l'échelle qui n'est pas
linéaire. Cela fonctionne pour des éclaircissement autour de 500 Lux. Avec un Luxmètre, le nombre numérique
est environ de 700 pour 500 Lux.*/
  //
  Serial.print("{\"TMP36\":"); /*On écrit dans le moniteur série {"TMP36": On écrit \" pour ne pas fermer
avec les guillemets mais dire au compilateur qu'on veut écrire une guillemet, on échappe avec : \
  Serial.print(temp); //On écrit dans le moniteur série la valeur de la variable temp
  Serial.print(",\"LDR\":"); //On écrit dans le moniteur série , "LDR":
  Serial.print(eclair); //On écrit dans le moniteur série la valeur de la variable eclair
  Serial.println("}"); //On écrit dans le moniteur série }
  delay(5000); //Pause de 5 secondes pour stabiliser l'affichage des valeurs sur la dashboard côté client
}

```

QUESTION 5

QUESTION 6

#### Valeurs affichée par le moniteur série :

```

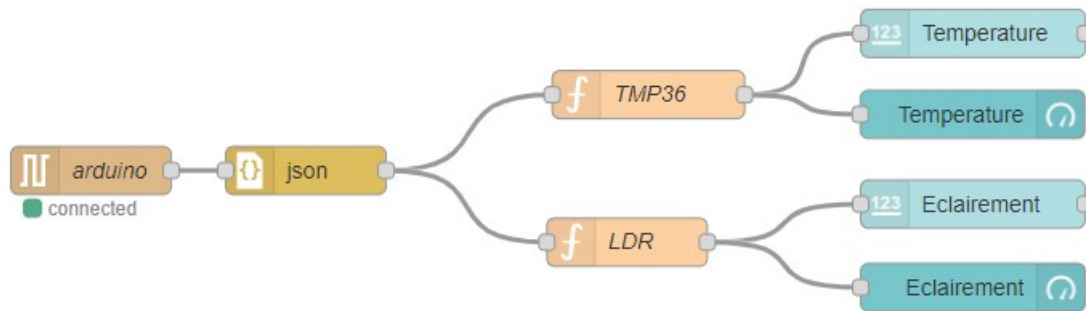
1 {
2   "TMP36": 17.38,
3   "LDR": 416.50
4 }

```

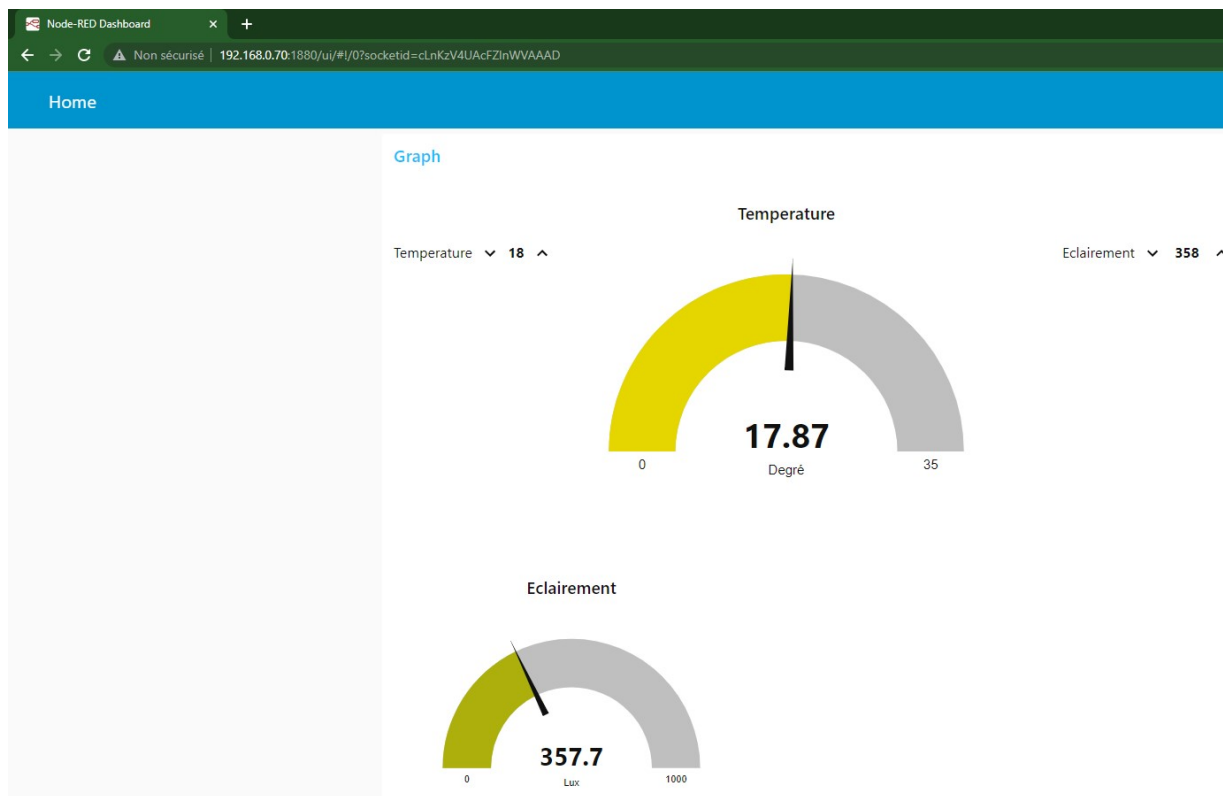


## Activité : Création d'une mini-station météo connectée sur serveur Node JS

Programme Node-Red complet :



Vue de la dashboard sur le serveur Node-Red accessible sur le port 1880/ui de la Raspberry Pi :



Pour toutes questions concernant l'activité, n'hésitez pas à me contacter :

[benoit.pierron@ac-toulouse.fr](mailto:benoit.pierron@ac-toulouse.fr)

ou directement par téléphone au lycée : 05.65.98.14.80