

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

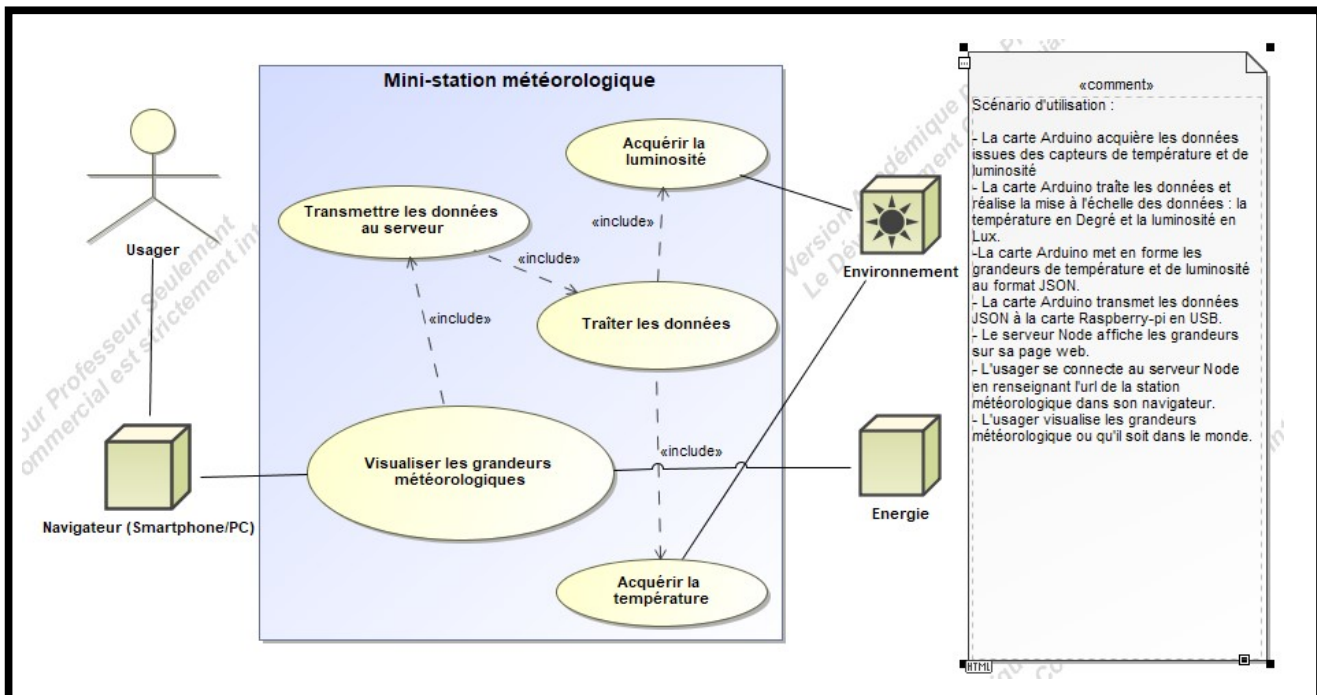
Objectif :

L'objectif de cette activité est de réaliser une mini-station météo connectée qui permet de connaître les conditions météorologiques de notre logement à tout moment et en tout lieu avec notre Smartphone.

Les grandeurs mesurées seront la température et la luminosité, par la suite il sera possible d'ajouter d'autres capteurs.

Une page web appelée « Dashboard » servira d'interface homme machine (IHM). Cette page web sera hébergée sur un serveur Node installé sur une carte Raspberry Pi disponible depuis le réseau Ethernet et Internet.

Ci-dessous le diagramme de cas d'utilisation et le scénario d'utilisation :



Matériel nécessaire pour la réalisation du mini-projet :

- Une carte Arduino Uno avec son alimentation alternative/continue.
- Une carte Raspberry Pi avec son alimentation et Node-Red installé sur l'OS de la carte microSD.
- Un capteur de température analogique type TM36 ou LM35.
- Un capteur de luminosité analogique type LDR avec une résistance de 10 K Ω , 1/4 W pour être montée en pont diviseur de tension.

Réaliser un compte rendu d'activité en répondant aux questions suivantes.

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

Partie conception :

- 1) **Compléter** le schéma de câblage du mini-projet en complétant le schéma du document réponse 1 donné en fin d'activité.
- 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.
- 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 ».
- 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.

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.

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

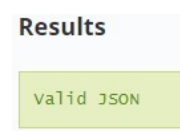
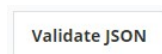
Afficher les valeurs de température et de luminosité sur le moniteur série de votre PC.

Faire valider par le professeur.

- 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.

- 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 ValidateJSON pour contrôler la conformité des données : **Results**



Faire valider par le professeur.

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

- 9) **Réaliser** le programme Node-Red en vous aidant du document ressource 2 : programmation sous Node-Red.

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

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.

Redirection de port

IP Destination : 192.168.0.70

Redirection active : ☒

IP source : Toutes

Protocole : TCP

Port de début : 1880

Port de fin : 1880

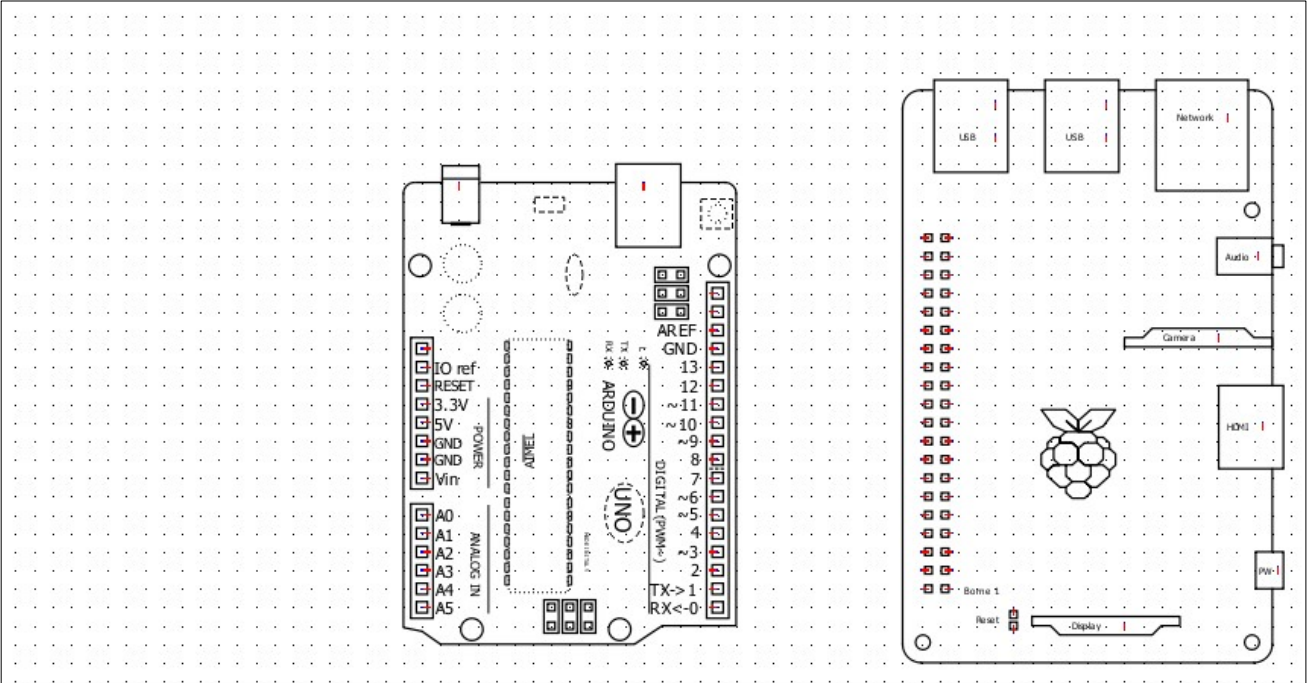
Port de destination : 1880

Commentaire :

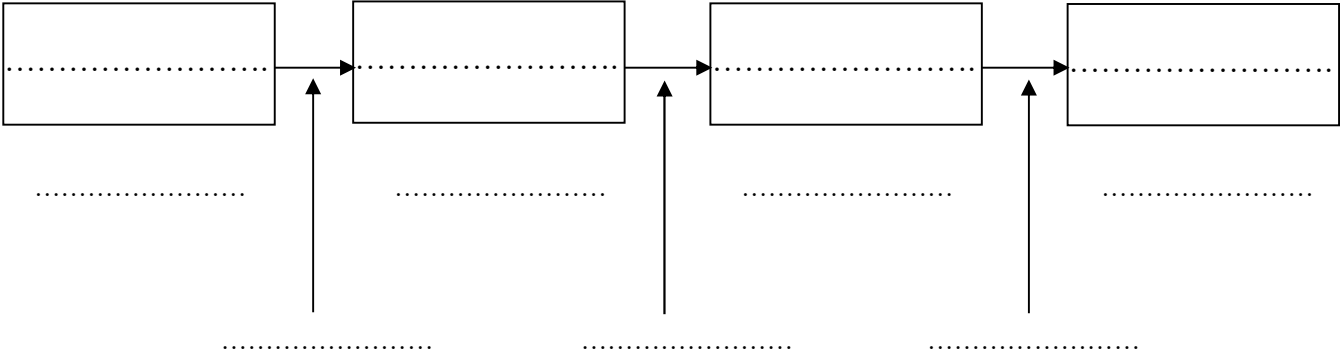
Annuler Sauvegarder

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

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



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 DR1 : Format de données JSON

Manipuler des données JSON :

Le **JavaScript Object Notation (JSON)** est un format standard utilisé pour représenter des données autrement dit, c'est une façon de stocker des informations structurées de façon semblable aux objets JavaScript. Il est habituellement utilisé pour structurer et transmettre des données sur des sites web (par exemple, envoyer des données depuis un serveur vers un client afin de les afficher sur une page web ou vice versa).

Exemple de syntaxe JSON:

```
{
  "espece" : "Chien",
  "race" : "Caniche",
  "couleur" : "Blanc",
  "age" : "6"
}
```

Données JSON envoyées par l'Arduino :

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

Comme vous le voyez, JSON est un format de données consistant en paires de nom/valeur (ou clé/valeur) ayant la forme de chaînes de caractères. Les nom et valeur sont séparés par deux points **:** et chaque paire est séparée de la suivante par une virgule.

Bien que trouvant sa source dans JavaScript, beaucoup de langages de programmation peuvent générer et lire le format JSON. Il est donc devenu très populaire pour le stockage, la lecture et le partage d'information dans les applications et services web.

Par conséquent, il n'y a pas de méthodes ou autres fonctionnalités dans JSON, il n'y a que du texte. Et c'est bien ainsi, car c'est ce qui en fait un format universel d'échange de données.

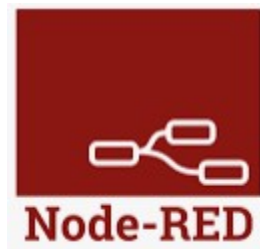
Activité : Création d'une mini-station météo connectée sur serveur Node JS

Document Ressource DR2 : Programmation sous Node-Red

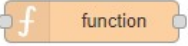
Présentation de Node-Red :

Node-Red est un projet Open Source basé sur Node.JS (JavaScript côté serveur) créé par IBM en 2013 afin de développer des applications d'objets connectés professionnelles. Il permet de faire du prototypage d'application très rapidement.

Node-Red est un outil de programmation graphique par assemblage de nodes (blocs fonctionnels) qui permet des liens entre les flux de données entrant et sortant. Il est notamment utilisé pour développer des applications de l'Internet des objets (IoT).

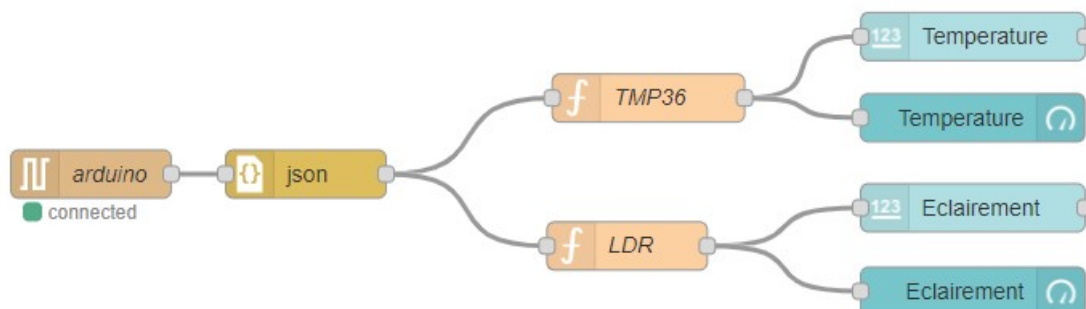


Il fournit un éditeur basé sur un navigateur qui facilite l'assemblage des nœuds constituant un *flow* qui sera déployé en un clic. Il existe une vaste gamme de nœuds dans la palette.

Il est possible de coder directement en JavaScript avec le Node function (fonction). 

Node-Red est désormais installé par défaut lors de l'installation sur le Raspberry Pi OS.

Programme complet de la mini-station météo que l'on va apprendre à développer :



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

Tutoriel de programmation de la mini-station météo :

I/ Connexion au port série de l'Arduino avec Node-Red :

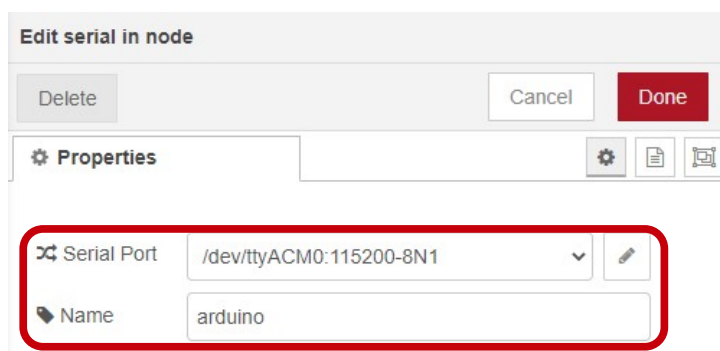
Déconnecter votre carte Arduino de votre PC et connecter la carte Arduino à la carte Raspberry Pi que vous indiquera le professeur.

Aller sur Node-Red avec le navigateur d'un PC connecté au réseau Ethernet du lycée en tapant l'adresse IP + port de la carte Raspberry Pi que vous indiquera le professeur, exemple : « 10.10.8.70:1880 ».

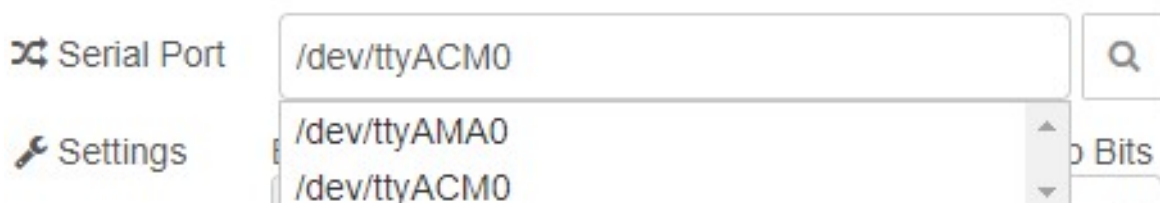
Glisser/Déposer le Node Serial disponible dans la palette Network sur la page blanche, ce Node permettra d'acquérir les données de l'Arduino. **Faire** un double clic sur le Node pour ouvrir le panneau de configuration.



Faire un double clic sur le Node pour ouvrir le panneau de configuration, **changer** son nom : « arduino ».

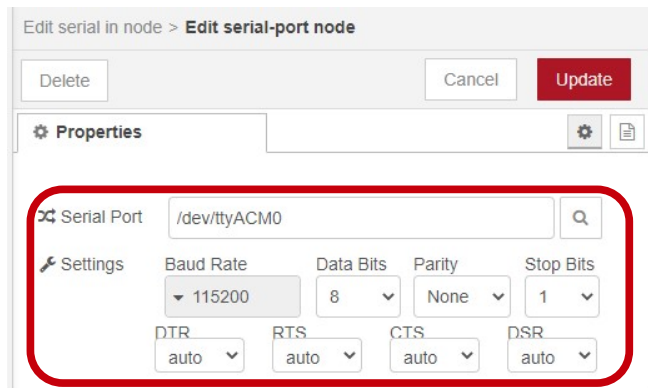


Cliquer sur le crayon pour ajouter une nouvelle connexion. **Utiliser** la loupe pour lister les ports COM. **Sélectionner** le port sur lequel est branchée la carte Arduino sur le Raspberry Pi : /dev/ttyACM0 par exemple.



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

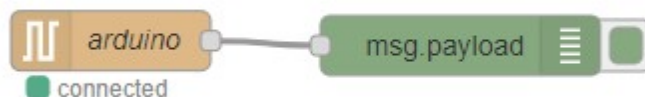
Configurer et enregistrer (Update) la liaison série comme ci-dessous :



Chercher le Node Debug.

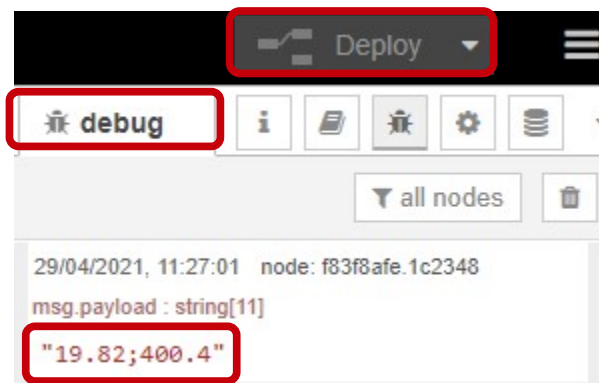


Relier le Node Serial au Node debug avec un fil entre les deux carrés. Votre premier flow est créé.



Déployer le flow en cliquant sur Deploy.

Ouvrir l'onglet debug pour visualiser les données qui arrivent de l'Arduino.



Faire valider par le professeur.

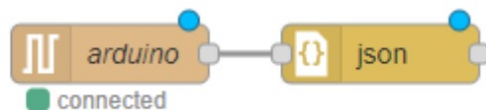
Activité : Création d'une mini-station météo connectée sur serveur Node JS

II/ Extraire les données JSON venant de l'Arduino avec Node-Red

Glisser/Déposer le Node JSON disponible dans la palette parser sur la page afin de pouvoir convertir la chaîne de caractère provenant de l'Arduino en objet JSON exploitable par Node-Red.



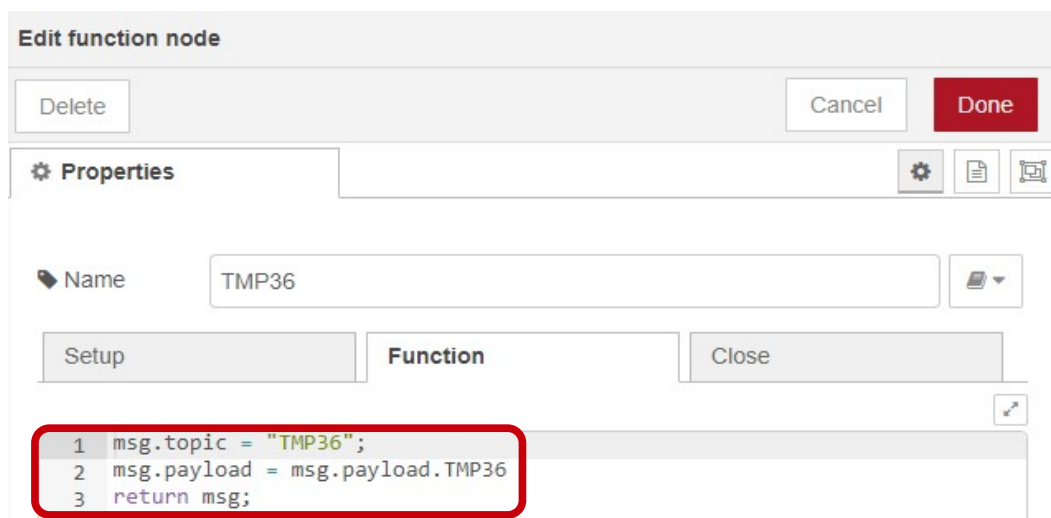
Supprimer le Node debug et **relier** le Node Serial au Node JSON.



Glisser/Déposer le Node function disponible dans la palette parser sur la page.

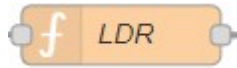


Faire un double clic sur le Node pour ouvrir le panneau de configuration et **écrire** le programme JavaScript ci-dessous. Node-Red transfère des messages (msg) au format JSON entre chaque Node. Les données se trouvent dans la clé payload. La première ligne extrait la mesure du capteur TMP36 et écrase le payload actuel. La fonction retourne le message (msg) actualisé. Il ne renvoie plus que la mesure du capteur.

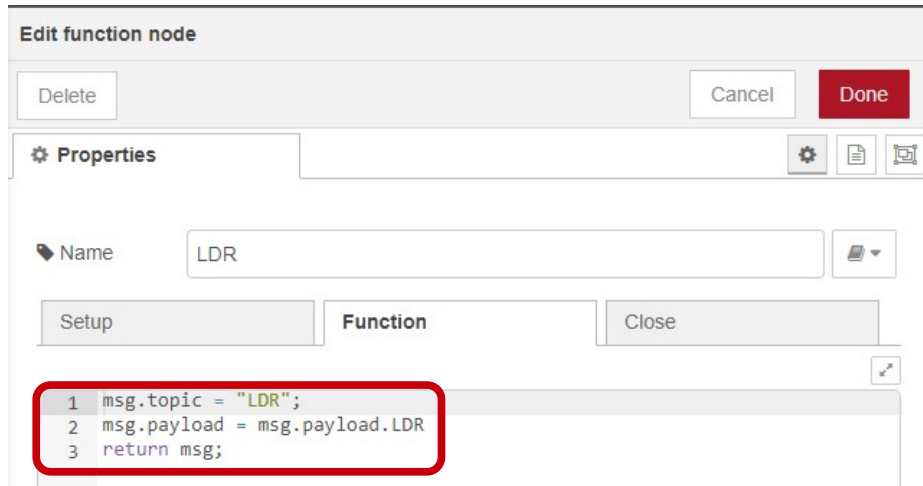


Activité : Création d'une mini-station météo connectée sur serveur Node JS

Glisser/Déposer le Node function disponible dans la palette parser sur la page.



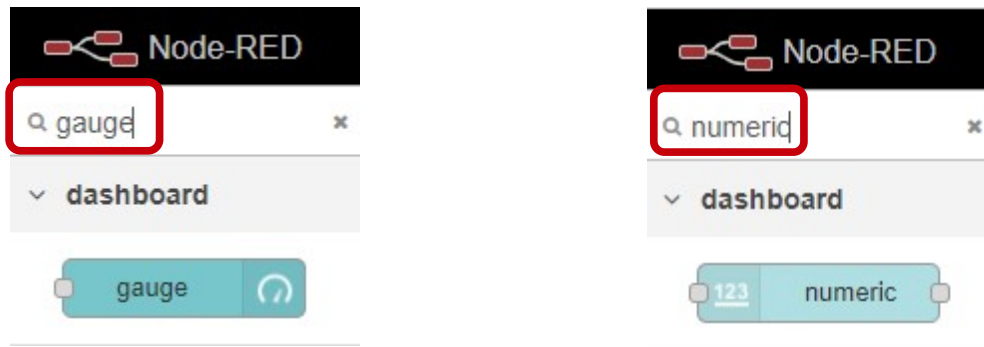
Faire un double clic sur le Node pour ouvrir le panneau de configuration et **écrire** le programme JavaScript ci-dessous. Node-Red transfère des messages (msg) au format JSON entre chaque Node. Les données se trouvent dans la clé payload. La première ligne extrait la mesure du capteur LDR et écrase le payload actuel.



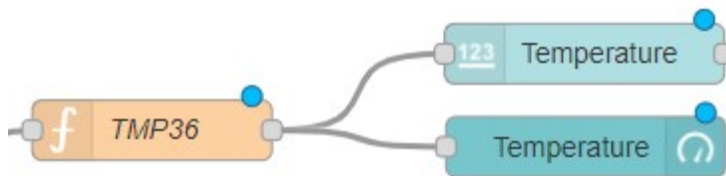
Activité : Création d'une mini-station météo connectée sur serveur Node JS

III/ Visualiser les mesures :

Glisser/Déposer les Node gauge et numeric disponible dans la palette dashboard sur la page.



Relier par des traits le Node function TMP36 aux Nodes gauge et numeric.



Faire un double clic sur le Node gauge pour ouvrir le panneau de configuration.

Configurer et enregistrer (Done) comme ci-dessous :

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

Réaliser la même chose pour la LDR.

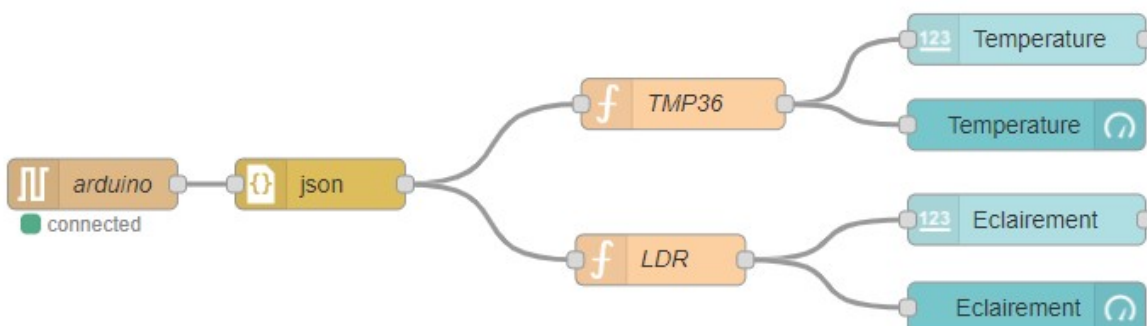


Edit gauge node
Delete
Cancel
Done

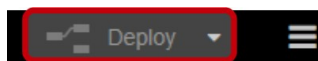
Properties

Group [Home] Graph
Size 11 x 8
Type Gauge
Label Temperature
Value format {{value}}
Units Degré
Range min 0 max 35
Colour gradient
Sectors 0 ... 15 ... 25 ... 35
Name

Vérifier que votre programme final ressemble bien à celui-ci :



Déployer le flow en cliquant sur Deploy :

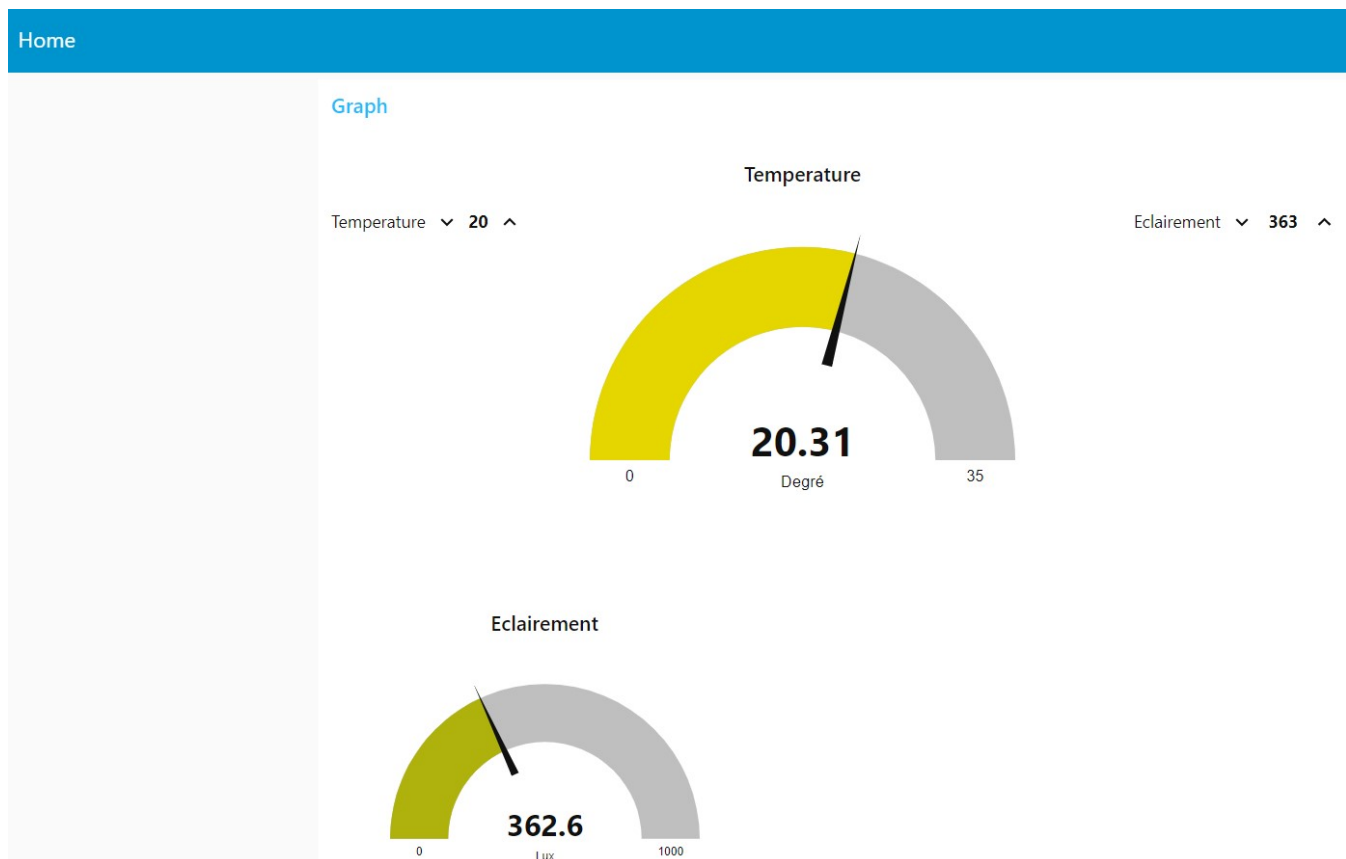


Activité : Création d'une mini-station météo connectée sur serveur Node JS

Pour une connexion en local : **Connectez**-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 : « IP Raspberry 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 : « IP externe _box :1880/ui » par exemple : « 55.160.205.187:1880/ui ».

Vérifier si votre page web (Dashboard) ressemble à celle ci-dessous :



Faire valider par le professeur.