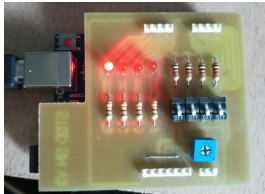


## 1 Présentation de l'outil Node-red

Node red est un outil visuel qui va nous permettre d'obtenir des pages web pour les capteurs et actionneurs reliés à une carte arduino.

Les codes sont séparés dans plusieurs blocs ce qui va organiser plus facilement notre code.



Arduino et potentiomètre



Codage d'une jauge



Page web avec la position du potentiomètre

La liaison entre la carte arduino et Node-red sera la liaison série.  
 Le serveur Node-red est installé sur une carte Raspberry

### 1.1 Déroulement du TP :

- Installation de l'IDE Arduino
- Rappel, reprise des codes Arduino déjà fait dans l'année
- Vérification que ces codes envoient ou reçoivent des données sur le port série
- Découverte de Node-red seul
- Mise en place de Node-red avec Arduino

**Faire un compte rendu de TP avec des imprimés écran**

## 2 Installer des logiciels en mode graphique

### 2.1 Capture d'écran

L'installation d'une application en mode graphique s'effectuent depuis le menu "Framboise/Preferences/Add Remove Software" en saisissant son nom dans la zone de texte de la boîte de dialogue.

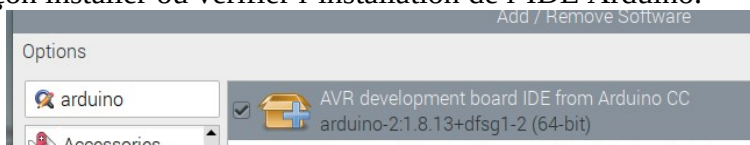
Installer ou vérifier l'installation de xfce4-screenshooter.



Tester ce logiciel. Expliquer à quoi il sert. Dans quel dossier avez vous sauvegardé les fichiers.

### 2.2 IDE Arduino

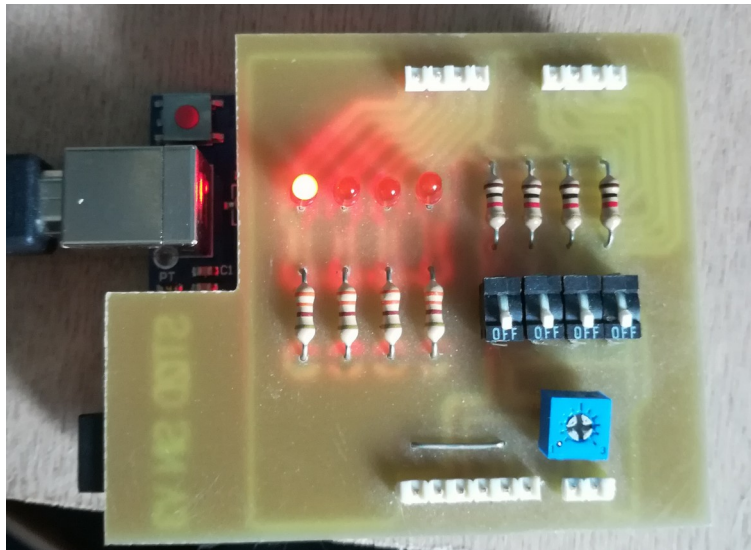
De la même façon installer ou vérifier l'installation de l'IDE Arduino.



Si besoin, aller dans préférences et mettre en français. Relancer Arduino.

### 3 Utilisation du shield de TP

Brancher le shield sur une carte arduino et un cordon usb vers la carte Raspberry.



|                  |             |             |             |             |           |           |            |            |      |
|------------------|-------------|-------------|-------------|-------------|-----------|-----------|------------|------------|------|
| Numéro de broche | 2           | 3           | 4           | 5           | 8         | 9         | 10         | 11         | A0   |
| composant relié  | inter<br>I2 | inter<br>I3 | inter<br>I4 | inter<br>I5 | led<br>D8 | led<br>D9 | led<br>D10 | led<br>D11 | Pot. |

Ouvrir le classique « Blink » des exemples. Téléverser. La led de la carte Arduino doit clignoter.  
Rajouter la ligne

```
#define LED_BUILTIN 8
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000);
}
```

Tester, la led D8 doit clignoter.  
Tester les autres leds.

### 4 Lecture de la position du potentiomètre sur le port série

Téléverser ce programme dans l'arduino

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

void loop() {
  //écriture port série
  int pot1 = analogRead(A0);
  Serial.print("Port A0 = ");
  Serial.print(pot1);
  Serial.print(";");
  Serial.print("\n");
  delay(1000);
}
```

Dans le moniteur série, visualiser les valeurs obtenues.

## 5 Commande d'une led par la liaison série

Téléverser ce code dans la carte Arduino

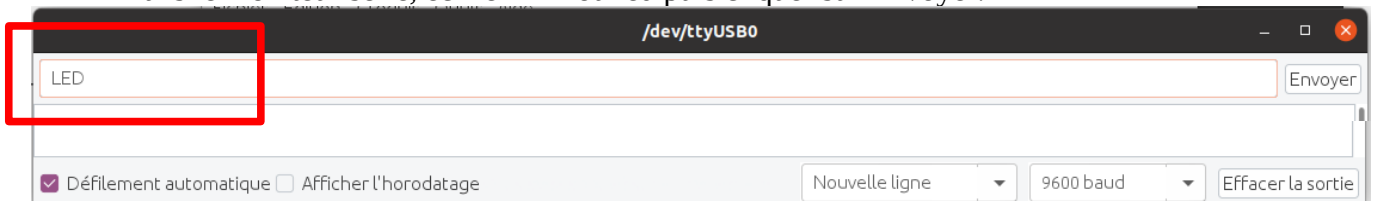
```
#define LED 8

String recu;

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

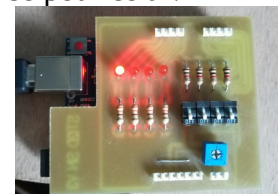
void loop() {
  //lecture port série
  if(Serial.available()>0)
  {
    recu = Serial.readStringUntil('\n');
    // Ordre d'allumer la LED
    if(recu=="LED") digitalWrite(LED,HIGH);
    else
    // Ordre d'éteindre la LED
    if(recu == "led") digitalWrite(LED,LOW);
  }
}
```

Dans le moniteur série, écrire LED ou led puis cliquer sur Envoyer.



Décrire le fonctionnement obtenu. Expliquer en faisant l'algorithme de ce programme.

Sur le schéma ci-dessous, rajouter la liaison série. Quel cordon est utilisé pour cela ?



**Liaison série par le port USB**


## 6 Utilisation de node red

Node red est un outil visuel qui va nous permettre d'obtenir des pages web pour les capteurs et actionneurs reliés à une carte arduino.

Les codes sont séparés dans plusieurs blocs ce qui va organiser plus facilement notre code.

### 6.1 Accès à votre serveur Node-red

Node red est installé en tant que serveur dans la carte Raspberry.

Dans Framboise, Programmation, vérifier que vous avez l'icône  Node RED.

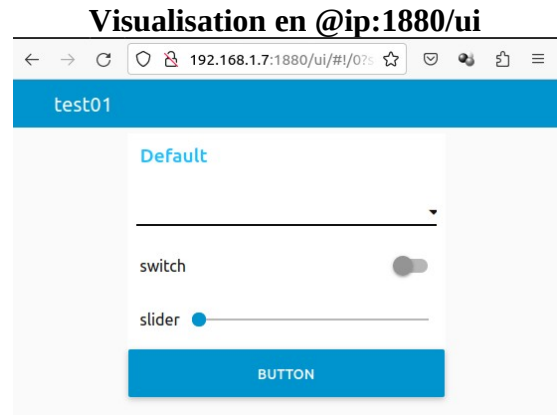
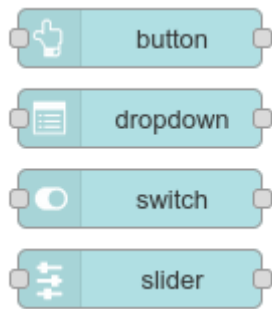
Cliquer dessus, décrire l'écran obtenu.

Vous êtes dans le menu d'édition d'un flow.

## 6.2 Utilisation du dashboard, création d'une page web

Exemple de résultat :

### Programmation en @ip:1880



### 6.2.1 A droite, dans Manage Palette, installer celle-ci :



A gauche, vous devez obtenir une vingtaine de blocs qui pourront être visibles sur une page web.

Cet icône montre qu'il y a un réglage à faire :



Après chaque modification, il faut cliquer sur  pour valider.

### 6.2.2 Premier flow

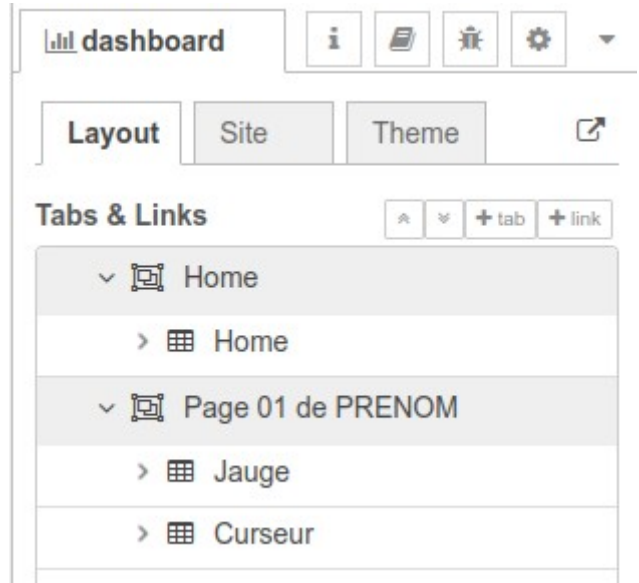
Recopier le flow suivant :



A gauche, afficher le menu dashboard

Avec +tab, créer une page nommée « page 1 de NOM.

Dans ce tab, avec +group, ajouter les groupes Jauge et Curseur.



Double cliquer sur les objets et leurs affecter leur tab et leur groupe



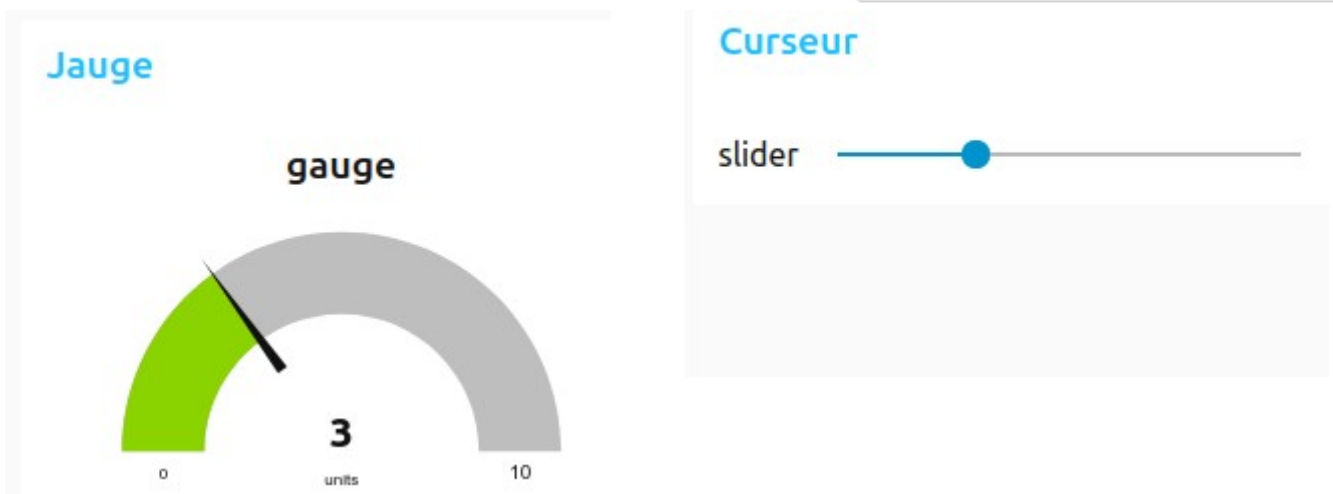
Il faut créer une page web : tab = « Page 01 de votre NOM »

## Page 01 de PRENOM

Double cliquer sur un objet et lui donner un group

Group

[Page 01 de PRENOM] Curseur

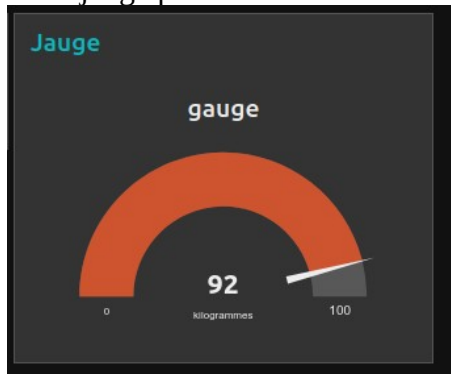


Vous devez obtenir ce résultat. Si vous bouger le curseur, la jauge suit. Dans le menu d'édition, la position du slider est reporté, ici 3.



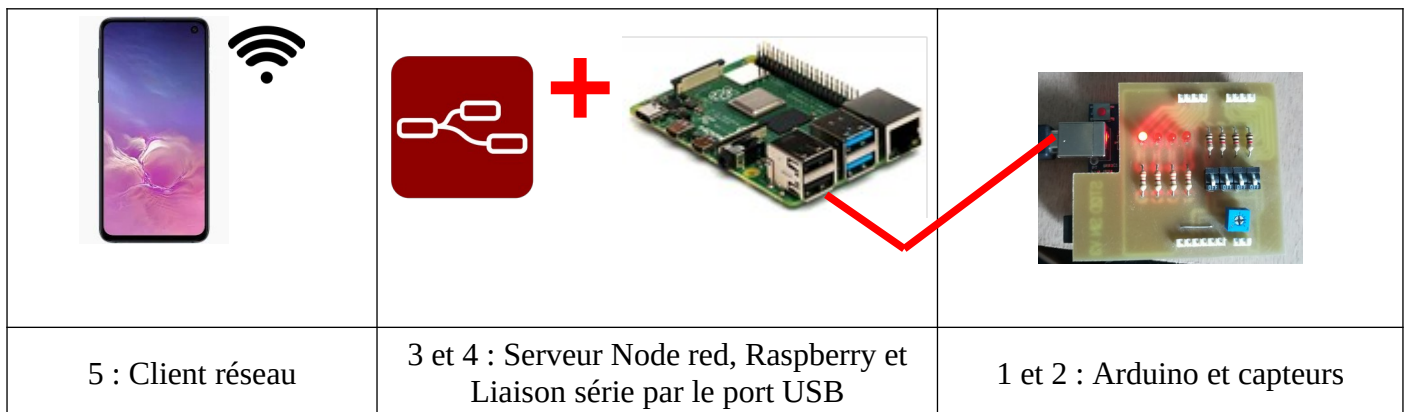
Si vous avez du temps, explorer le menu dashboard => Theme pour changer le style de la page.

Changer l'unité et l'échelle de la jauge pour obtenir ceci :



## 7 Node red et Arduino

### 7.1 Organisation et fonctionnement de la maquette.



1. Les capteurs sont placés sur la carte arduino.
2. Le code de la carte arduino est le code habituel. Les valeurs des capteurs sont envoyés sur le port série par `Serial.print(valeur du capteur)`
3. La carte raspberry contient un serveur Node red
4. Node red communique avec la carte arduino par la liaison série
5. Le client, par le réseau peut accéder aux valeurs des capteurs à `adresseIP_carteRaspberry:1880/ui`

### 7.2 Affichage de la position du potentiomètre

#### 7.2.1 Téléverser ce programme dans la carte Arduino

Vérifier le résultat de 0 à 1023 dans le moniteur série

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

void loop() {
  int pot1 = analogRead(A0);
  //écriture port série
  Serial.print("Port A0 = ");
  Serial.print(pot1);
  Serial.print(";");
  Serial.print("\n");
  delay(1000);
}
```

```
Port A0 = 598;
Port A0 = 599;
Port A0 = 598;
Port A0 = 598;
Port A0 = 599;
Port A0 = 598;
Port A0 = 598;
Port A0 = 599;
Port A0 = 598;
Port A0 = 599;
Port A0 = 598;
Port A0 = 598;
Port A0 = 598;
```

#### 7.2.2 Flow node red

Supprimer ou désactiver les flows déjà existants. Seul celui-ci doit être « enable ».

Tracer ce flow.



Configurer le serial\_in ainsi et déployer

Rapidement, le point connected montre que Node red voit la carte arduino.

**Properties**

Serial Port: /dev/ttyUSB0

Settings: Baud Rate: 9600, Data Bits: 8, Parity: None, Stop Bits: 1

DTR: auto, RTS: auto, CTS: auto, DSR: auto

Input: Optionally wait for a start character of [ ], then Split input on the character [ \n ] and deliver ASCII strings

Output: Add character to output messages [ \n ]

Request: Default response timeout: 10000 ms



En cliquant à droite sur debug, on peut voir les données arrivant de la carte arduino

**debug**

all nodes | all

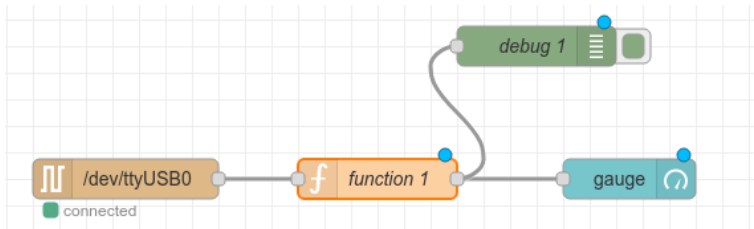
```
09/03/2023 09:40:15 node: debug 1
msg.payload : string[15]
▶ "Port A0 = 598;\n"

09/03/2023 09:40:16 node: debug 1
msg.payload : string[15]
▶ "Port A0 = 598;\n"
```

### 7.2.3 Le flow complet

Compléter le flow

A droite, menu dashboard :  
+tab : page capteurs de PRENOM  
+group potentiomètre



**dashboard**

Layout | Site | Theme

tabs & links: +tab +link

- > Home
- > Page 01 de PRENOM
- > page capteurs de PRENOM
  - > Potentiomètre


### 7.2.4 Mise en forme pour affichage

Dans le menu debug, on voit que le message est Port A0 = 598 ; \n.

Pour utiliser ce message, une fonction « prépare gauge » va en extraire la valeur, ici 598.

Préparer les objets.

▼ fonction



Name: Prépare Jauge

Setup On Start **On Message**

```
1 var o;  
2 var msg1 = {};  
3 o = msg.payload.split(" ");  
4 msg1.payload = o[3];  
5 o = msg1.payload.split(";");  
6 msg1.payload = o[0];  
7 return msg1;
```



Edit gauge node

Delete Cancel Done

Properties

Group: [page capteurs de PRENOM] Potentiomé

Range: min 0 max 1023



Valider par

Dans debug on peut vérifier que seul le 598 est gardé

@IP:1880/ui

debug


all nodes

```
"598"  
09/03/2023 10:19:24 node: debug 1  
msg.payload : string[3]  
"598"
```

page capteurs de PRENOM

Potentiomètre

gauge



0 598 units 1023

Changer la position du potentiomètre, la page doit s'actualiser en direct.

## 7.2.5 Sauvegarder votre travail

A droite, faire Export, current flow => Download

Export nodes

Export: selected nodes **current flow** all flows

Clipboard

Local

Export nodes

JSON

Flow 2

- /dev/ttyUSB0
- debug 1
- Prépare Jauge
- gauge

Global Configuration Nodes

- /dev/ttyUSB0:9600-8N1
- [page capteurs de PRENOM] Potentiomètre
- page capteurs de PRENOM

Un fichier flows.json est placé dans le dossier Téléchargement du poste actuel.



Vous pourrez réutiliser ce fichier avec la commande Import.

## 7.3 Allumer des leds depuis le dashboard

### 7.3.1 Reprendre et téléverser ce code dans la carte Arduino

```
#define LED 8



String recu;

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

void loop() {
  //écriture port série
  int pot1 = analogRead(A0);
  Serial.print("Port A0 = ");
  Serial.print(pot1);
  Serial.print(";");
  Serial.print("\n");
  delay(1000);
  //lecture port série
  if(Serial.available()>0)
  {
    recu = Serial.readStringUntil('\n');
    // Ordre d'allumer la LED
    if(recu=="LED") digitalWrite(LED,HIGH);
    else
    // Ordre d'éteindre la LED
    if(recu == "led") digitalWrite(LED,LOW);
  }
}
```

Ne pas effacer le flow précédent, le mettre sur disable.  
Tracer un nouveau flow



| Placer le switch dans cette page et ce groupe (à créer)                            | Configurer le switch pour qu'il envoie un texte lorsqu'il est activé                 |
|--|--|
|  |  |

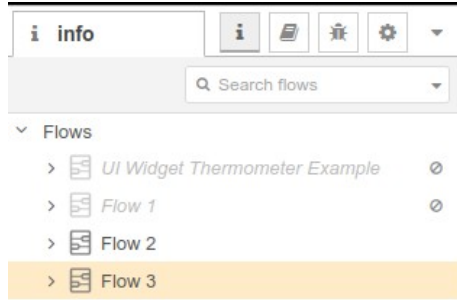
Deployer et vérifier.



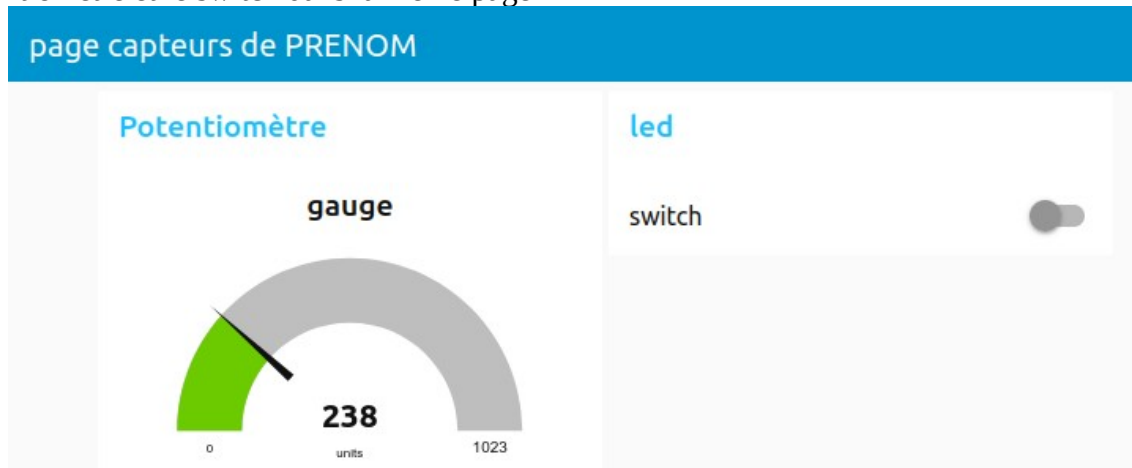
La led led8 doit s'allumer et s'éteindre avec le switch !

### 7.3.2 Rassembler les deux flows précédents

Mettre les deux flows sur enable. Ici Flow2 et Flow3.



Comme ces deux flows utilise le même tab dans le dashboard, une fois déployé on obtient le potentiomètre et le switch dans la même page



### 7.3.3 Commande des autres leds

A partir de ce code :

```
if(Serial.available(>0)
{
  recu = Serial.readStringUntil('\n');
  // Ordre d'allumer la LED
  if(recu=="LED") digitalWrite(LED,HIGH);
  else
  // Ordre d'éteindre la LED
  if(recu == "led") digitalWrite(LED,LOW);
}
```

Le nouveau code commence par

```
#define LED8 8
#define LED9 9
#define LED10 10
#define LED11 11
```

Rajouter des **else if** pour pouvoir commander les quatres leds.

Téléverser dans Arduino

Modifier le dashboard

Faire valider par le professeur.

### 7.4 Ajouter une image

Configurer le fichier **settings.js** pour définir où stocker les images ici /home/felix (voir en annexe)

Redémarrer node-red

mettre une image dans ce dossier /home/felix



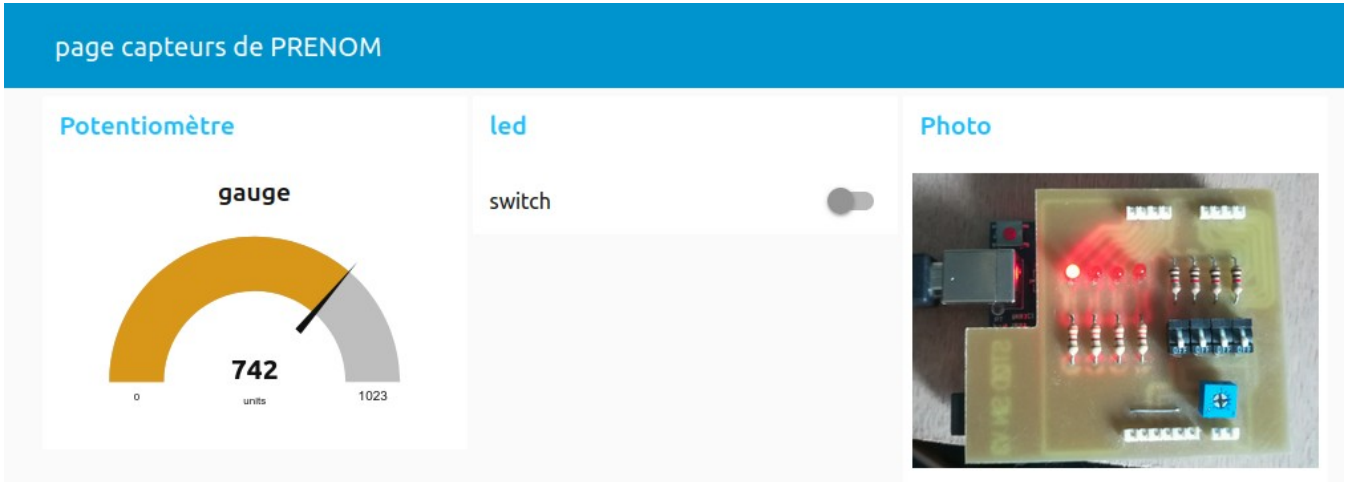
Dans le dashboard, placer un objet

Template

Mettre le code html dans le template

```
1 <img src='/shield.jpg'>
```

Résultat



### 7.5 Ajouter du texte



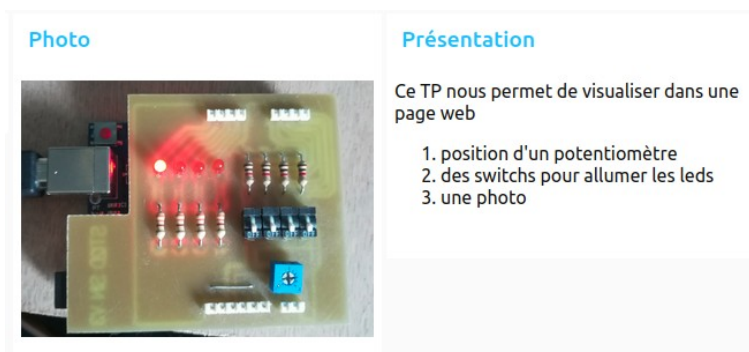
Dans le dashboard, placer un objet

Mettre le code html dans le template

Ce TP nous permet de visualiser dans une page web

```
<ol>
<li>position d'un potentiomètre</li>
<li>des switchs pour allumer les leds</li>
<li>une photo</li>
</ol>
```

Résultat



## 8 Annexe, configurer settings.js pour les images

Redémarrer node-red par snap restart node-red

Lire les logs par

```
tail -n 30 /var/log/syslog
```

Cette ligne nous indique où trouver le fichier settings.js

```
Settings file : /var/snap/node-red/871/settings.js
```

Ouvrir le fichier

```
sudo nano /var/snap/node-red/871/settings.js
```

Avec CTRL W, rechercher cette ligne et ajouter la nouvelle. Ne pas oublier la virgule.

```
//httpStatic: '/home/nol/node-red-static/', //single static source  
  httpStatic: '/home/felix/',
```

Redémarrer node-red

Cette ligne des logs nous indique que la modification est prise en compte

```
9 Mar 14:13:01 - [info] HTTP Static : /home/felix > /
```