

Node red - Chronomètre

1 Présentation

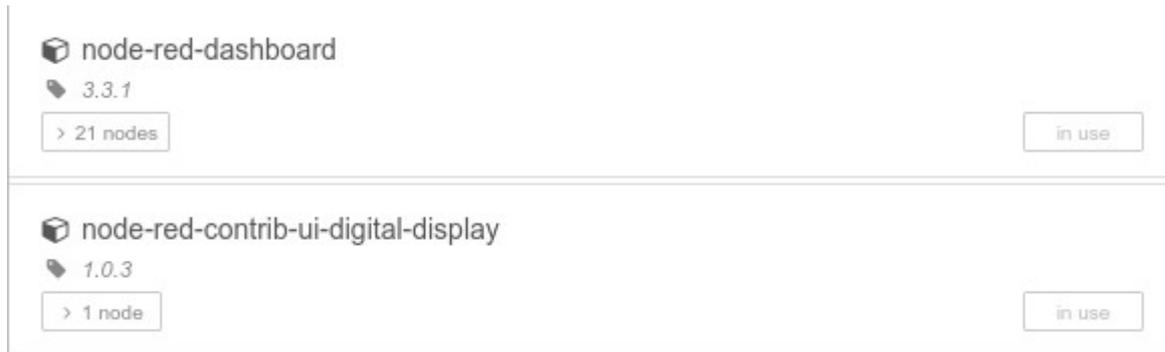
Pour votre projet, on vous demande de réaliser un chronomètre. Cette fiche montre une méthode utilisant le serveur Node-red.

La partie « **travail à faire** » résume la méthode de travail pour arriver à un système fonctionnel.

La partie « **Chronomètre , le code** » donne toute les explications du fonctionnement du code

2 Configuration du serveur Node-red

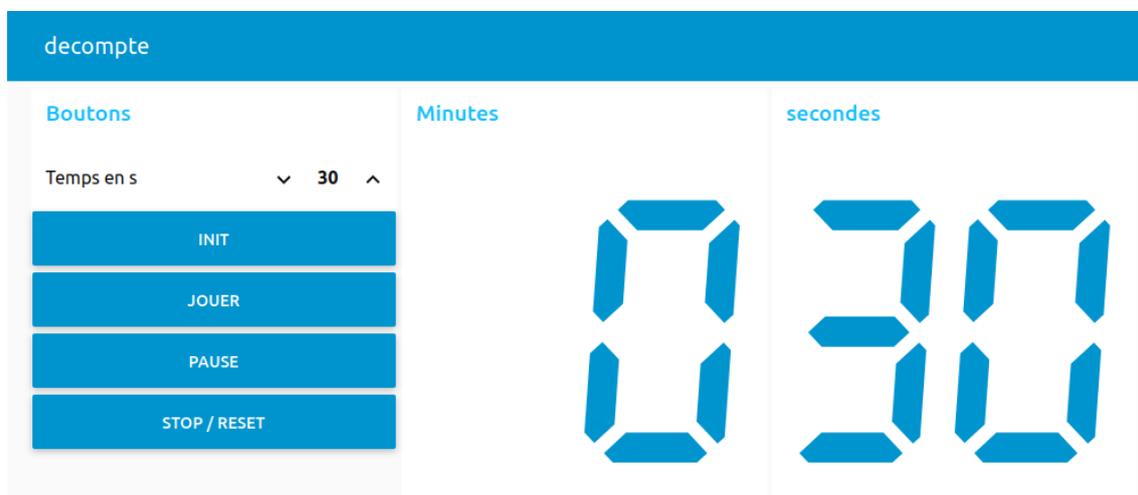
2.1.1 A droite, dans Manage Palette, installer celles-ci :



3 Travail à faire

- Dans Node-red, importer le fichier **chrono_eleve.json**. C'est un compteur basique.
- Dans un navigateur, faire **AdresseIP_serveur:1880/ui** pour obtenir l'interface utilisateur du chrono
- Faire une liste de ce qui fonctionne et de ce qui ne fonctionne pas
- Sur le schéma **chrono.png**, écrire le rôle de chaque nœuds pour comprendre l'organisation du code.
- A partir du paragraphe 'Améliorer la fonction « Compter les secondes en fonction des boutons »':
 - Rajouter les boutons manquants
 - Compléter la fonction « **Compter les secondes en fonction des boutons** ».
- Rajouter d'autre fonctions si besoin dans votre système.

4 Chronomètre complet vue dans le dashboard

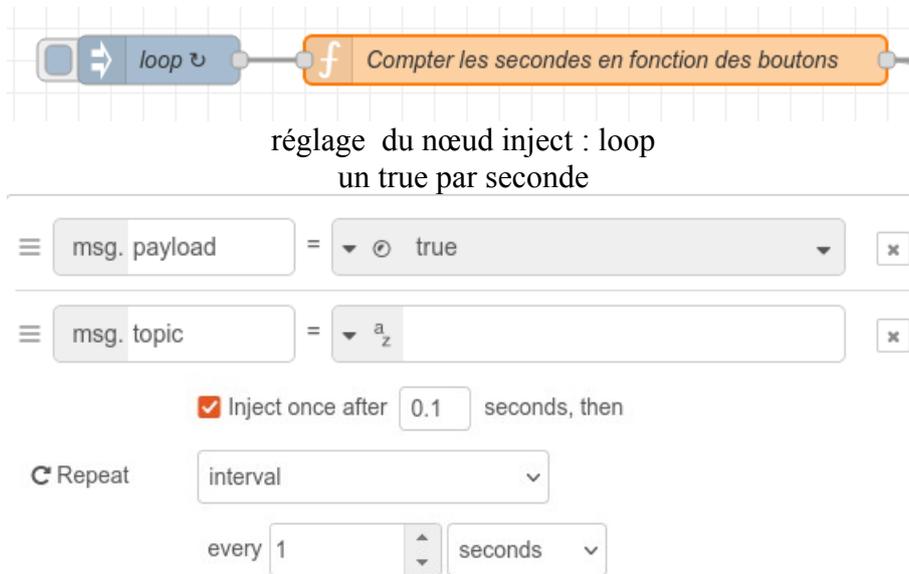


5 Chronomètre , le code

Le fonctionnement est séparé en :

- Une entrée numérique pour définir la durée du jeux
- Un setup, état du jeux au départ en Pause
- Une série de boutons pour définir l'état du jeux : Init (du temps), Jouer, Pause, Reset (du temps)
- Une fonction qui va décrémenter le chrono toutes les secondes
- Une fonction pour passer des secondes à mn et s
- Afficheurs minutes et secondes

5.1 Une fonction qui va décrémenter le chrono toutes les secondes



réglage du nœud inject : loop
un true par seconde

msg. payload = true

msg. topic = a_z

Inject once after 0.1 seconds, then

Repeat interval every 1 seconds

5.2 Les variables :

x1 = global.get("variable1") pour lire une variable1 visible dans tout le code
global.set("variable1", "1234"); pour mettre une variable1 à 1234

5.3 Fonction de base « Compter les secondes en fonction des boutons » :

```
var test = msg.payload;
var etat = global.get("etatJeu");
var tempsRest = global.get("tempsRestant");

if (test && etat=="Jouer"){
  tempsRest--;
  global.set("tempsRestant", tempsRest);
  msg.payload = tempsRest;
  return msg;
}
```

test => test reçoit un vrai toute les secondes
si un vrai et etat du jeu est vrai
décrémenter le chronomètre
enregistrer le temps restant
envoyer le temps restant dans msg.payload, la sortie de la fonction

5.4 Une entrée numérique pour définir la durée du jeux



Réglage de l'entrée numérique, Temps en s

→ Range min max step

Réglage du nœud change « set global.tempsJeu »

Rules

Set
to the value Deep copy value

5.5 Une série de boutons pour définir l'état du jeux : Init (du temps), Jouer, Pause, Reset (du temps)



Le bouton Jouer va envoyer le mot : Jouer lors d'un clic.

When clicked, send:

Payload
Topic

Un nœud « change » prend mot précédent et le met dans la variable etatJeu

Rules

Set
to the value Deep copy value

5.6 secondes vers mn:ss



```
var msg1 = {};  
var msg2 = {};  
var secondes;  
var minutes ;  
secondes = msg.payload;  
//quotient = (a - a % b) / b;  
minutes = (msg.payload - msg.payload % 60) /60;
```

```

msg1.payload= minutes;
//remainder = a % b
msg2.payload = msg.payload%60;
return [ msg1,msg2];

```

5.7 Améliorer la fonction « Compter les secondes en fonction des boutons »

5.7.1 Rajouter les boutons manquants

5.7.2 On donne ce code qui gère le bouton « Jouer »

```

var test = msg.payload;
var etat = global.get("etatJeu");
var tempsRest = global.get("tempsRestant");

if (test && etat=="Jouer"){
tempsRest--;
global.set("tempsRestant", tempsRest);
msg.payload = tempsRest;
return msg;
}

```

5.7.3 Compléter les lignes en points « //..... » pour gérer les autres boutons

```

var test = msg.payload;
var etat = global.get("etatJeu");
var tempsRest = global.get("tempsRestant");

if (tempsRest <= 1 && etat != "Pause") {
//gérer le cas où le temps devient négatif
//mettre le jeu en pause
.....
global.set("etatJeu", etat);

}
if (etat == "Init"){
//définir le temps de jeu
tempsRest = global.get("tempsJeu");
global.set("tempsRestant", tempsRest);
//jeu en pause en attendant le start
.....
global.set("etatJeu", etat);
//envoyer la valeur pour affichage
msg.payload = tempsRest;
return msg;
}
if (test && etat=="Jouer"){
//chronometre actif
//décrémenter les secondes
tempsRest--;
global.set("tempsRestant", tempsRest);
msg.payload = tempsRest;
return msg;
}

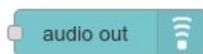
```

```

else if (test && etat == "Stop") {
//Arrêt du jeu et mise à 0 du temps, tempsRest
.....
global.set("tempsRestant", tempsRest);
//Mettre jeux en pause pour ne pas envoyer des 0 indéfiniment
.....;
global.set("etatJeux", etat);
//pour affichage du temps à 0
msg.payload = tempsRest;
return msg;
}
else if (test && etat == "Pause") {
//ne rien faire
}

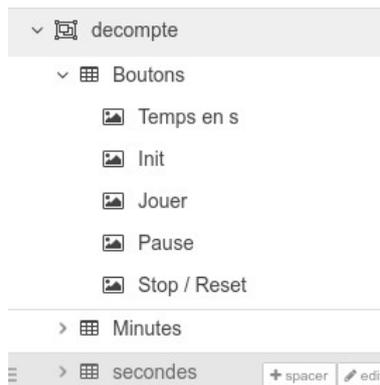
```

5.8 Améliorer avec un son



Utiliser un nœud « audio out »
Déclencher un son lorsque le chronometre arrive à 00.

5.9 Améliorer l’affichage



A droite, afficher le dashboard :
Les blocs s’affichent dan l’ordre de ce menu. On peut les glisser-déplacer pour changer l’affichage.