

# Réseau IoT Chenillard KNX

Théo Delagarde - Corentin Grosos  
24 mai 2022



# Sommaire

1. Introduction

2. Interface Web  
& chenillard

3. Architecture &  
communications

4. Synchronisation

5. Conclusion



# Introduction

**Objectif :** Faire communiquer une maquette knx et une interface web par le biais d'une API Rest.

**Langages de programmation choisis :**

Client : HTML / CSS / JS

Server : Javascript

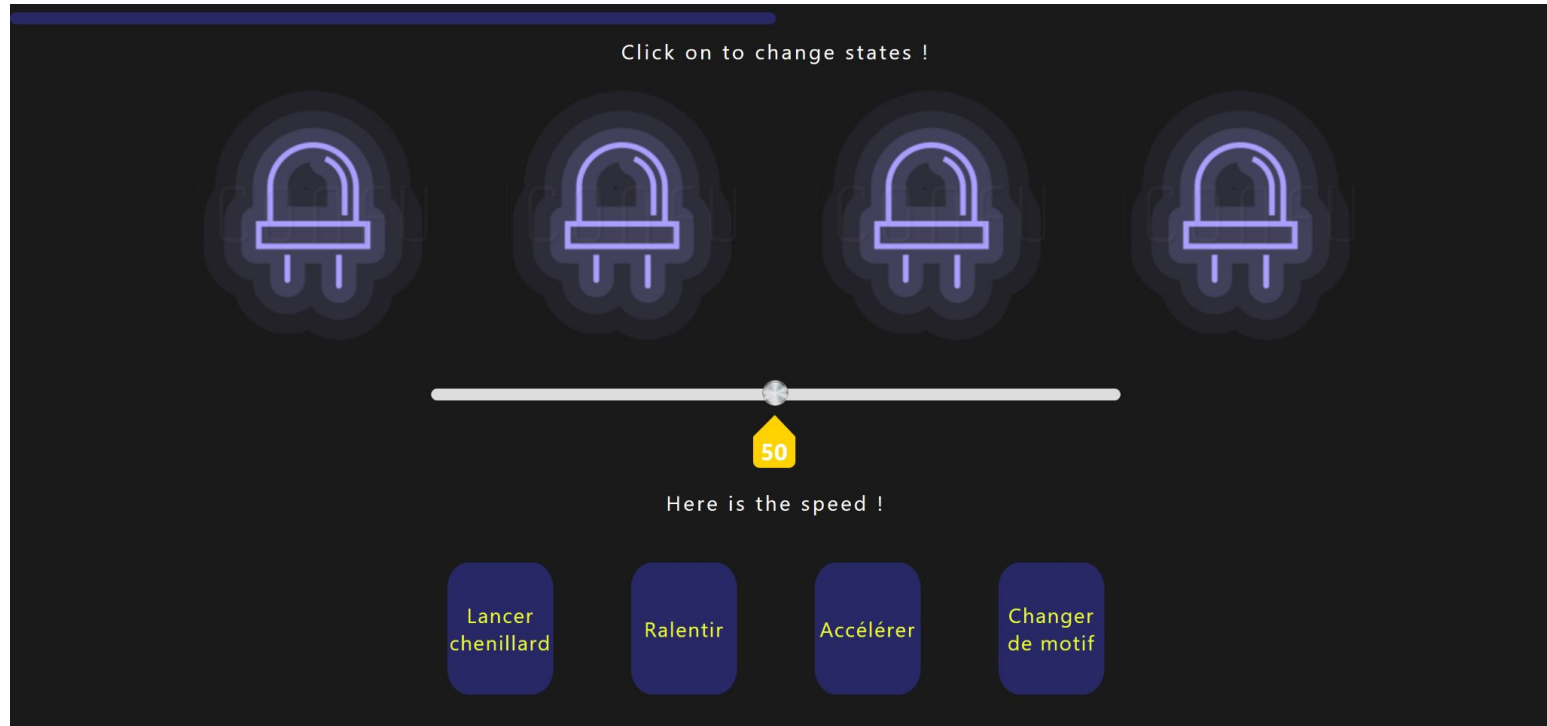
**Organisation du projet :**

Interface Web/API

Connexion KNX

Communication client / serveur / KNX

# Interface Web



# Interface Web

Lancer  
chenillard

## Gestion du chenillard

Actif quand chenille\_On == true

Appel de la fonction récursive chenilleMOTIFS()

Ralentir

## Gestion de la vitesse définie par

$\text{ActualSpeed} = \text{minSpeed} + \text{IntervalSpeed} + \text{IntSpeed}$

Accélérer

Avec minSpeed = 300, Interval Speed = 80 et

IntSpeed défini entre 0 et 10 selon les boutons Ralentir et Accélérer

$\Rightarrow 300 \text{ ms} < \text{ActualSpeed} < 1100 \text{ ms}$

Changer  
de motif

## Gestion du motif parmi 6 implémentés

traitement si  
(chenille\_On == true)



# Interface Web

## Fonctionnement du chenillard :

Fonction récursive appelée toutes les 300 - 1100 millisecondes

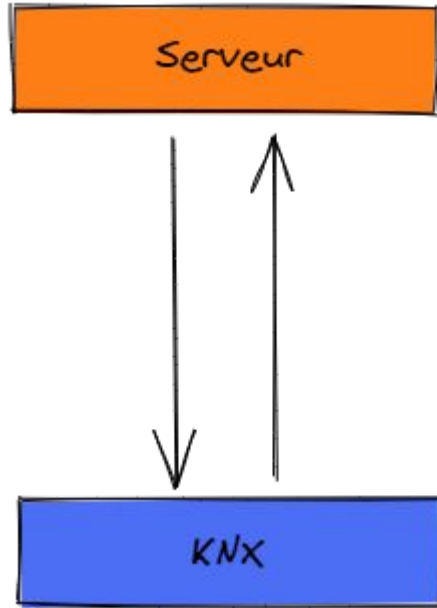
- Chenillard Simple
- Chenillard inversé
- Chenillard B2B
- Allumage aléatoire
- Allumage en quinconce
- Clignotement



```
function chenilleMOTIFS(){
  if(chenille_On == true){
    sleep(actualSpeed).then(() => {
      switch(decideMotif()){
        case "chenillardSimple":
          /* traitement */
          /* modification des leds */
          /* calcul de l'indice pour le prochain appel */
          return chenilleMOTIFS();
        case "chenillardInverse":
          /* traitement */
          /* modification des leds */
          /* calcul de l'indice pour le prochain appel */
          return chenilleMOTIFS();
        case "chenillardBackToBack":
          ...
        case "allumageAleatoire":
          ...
        case "parityQuinconce":
          ...
        case "everyLEDsOn":
          ...
      }
    })
  }
}
```



# Communication Serveur - KNX



## Librairie KNX pour JS

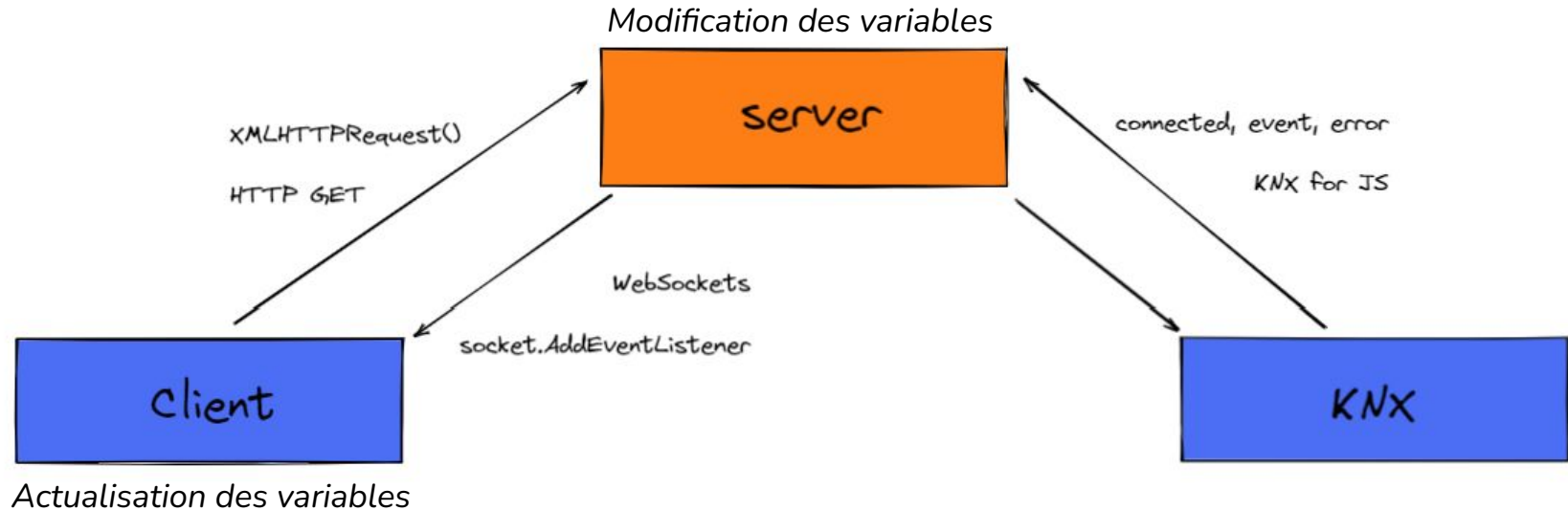
- Renseignement de l'adresse IP de la maquette
- Handlers contenant 3 évènements :
  - ◆ connected
  - ◆ event
  - ◆ error

## Librairie de déconnexion

async-exit-hook

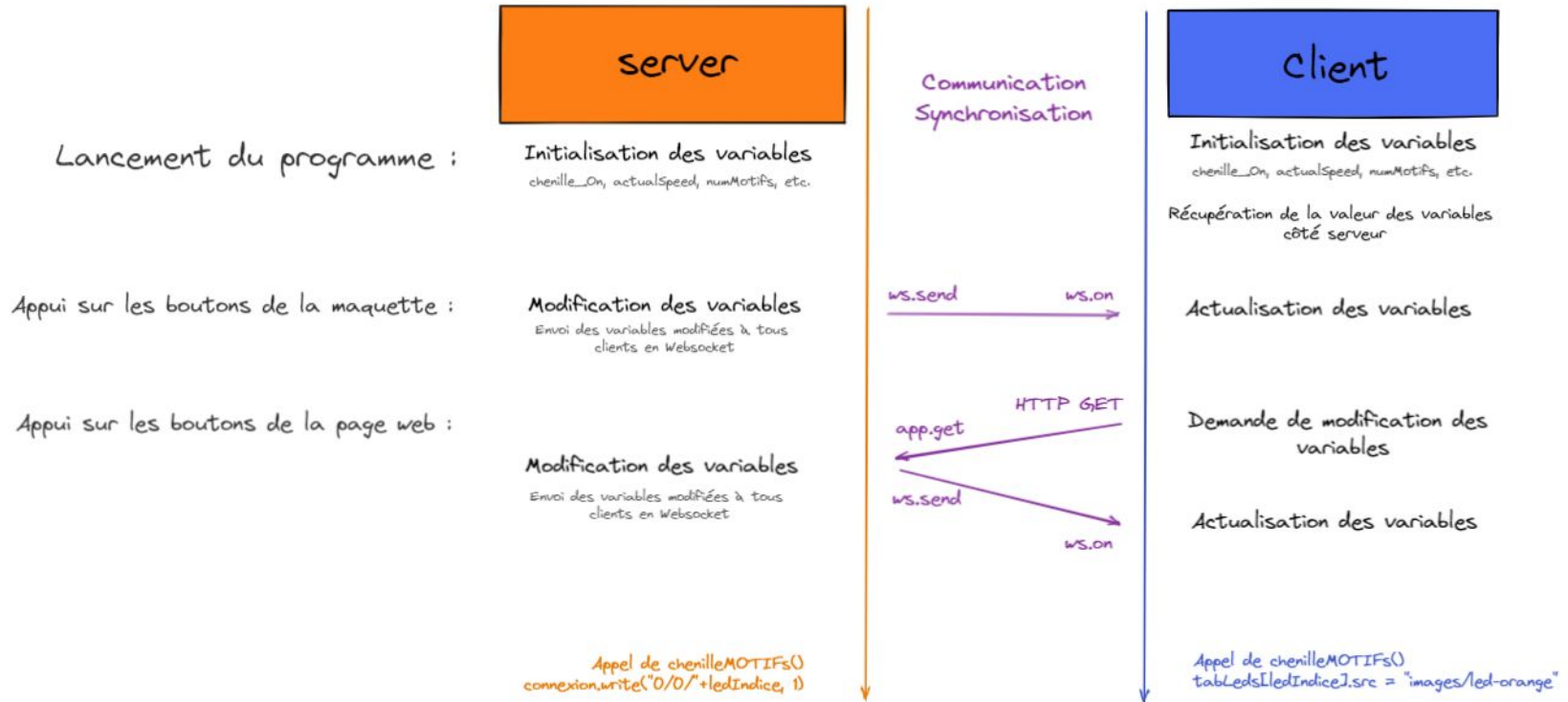


# Architecture





# Synchronisation client / serveur





# Conclusion

Développement API Rest

Pas d'utilisation de framework

Implémentation simple du KNX

Améliorations possibles

Démonstration