

# SYSTÈME TEMPS RÉEL ET MULTITÂCHE

Gestion d'écrans LCD en temps réel avec FreeRTOS

Fait par KABORE Aronn, SERE Akram, ZIDA Eben-Ezer et ZOUGOU  
Daniel

## 1. Le système est à la fois un système temps réel et multi-tâche

### Système Temps Réel

Un système temps réel est un système informatique qui doit répondre à des contraintes temporelles strictes. Dans notre cas,

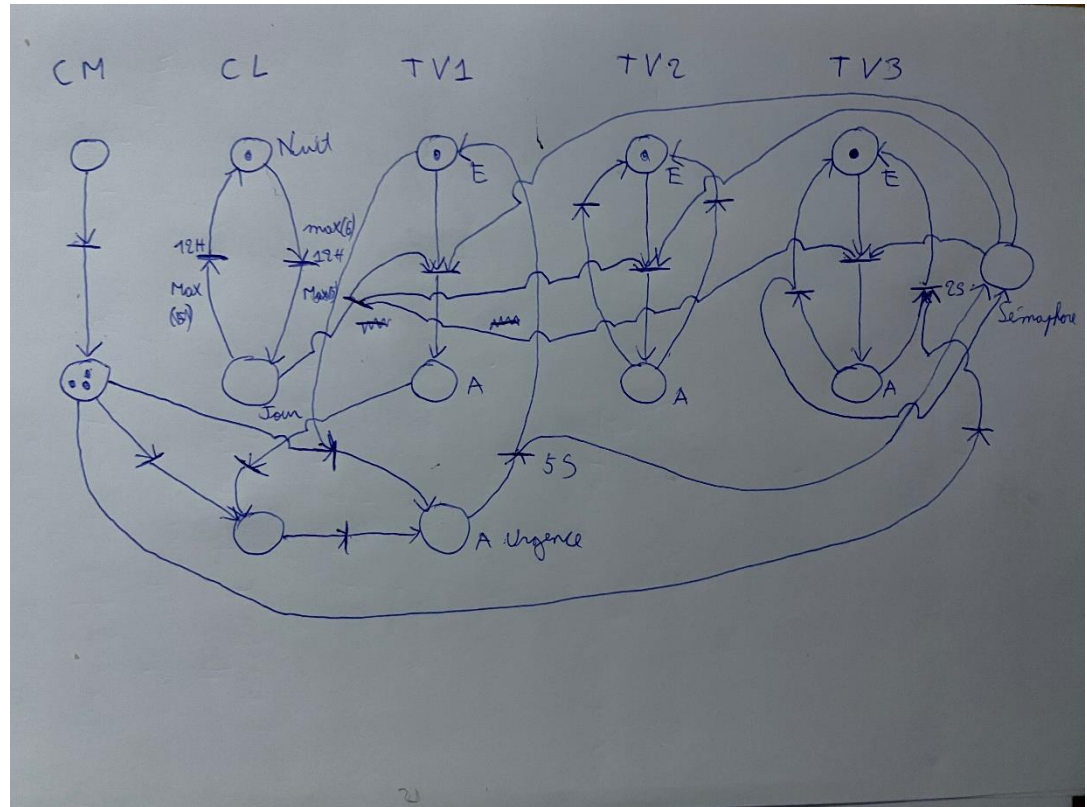
- Les télévisions doivent s'éteindre automatiquement après un délai maximum de deux heures.
- En cas de détection de mouvement, un message d'alerte doit être affiché immédiatement sur l'écran 1.
- Pendant la nuit, toutes les télévisions doivent s'éteindre automatiquement, sauf en cas d'affichage d'un message d'urgence.

### Système Multi-tâche

Un système multi-tâche est capable d'exécuter plusieurs tâches simultanément. Ici :

- Le système doit gérer l'état de trois télévisions indépendamment.
- Il doit surveiller constamment le capteur de luminosité pour déterminer si c'est le jour ou la nuit.
- Il doit aussi surveiller constamment le capteur de mouvement
- Il doit être capable de gérer les priorités des tâches, comme éteindre une télévision pour afficher un message d'urgence.

## 2. Réseau de pétri du fonctionnement du système



### 3. Les tâches et leurs propriétés

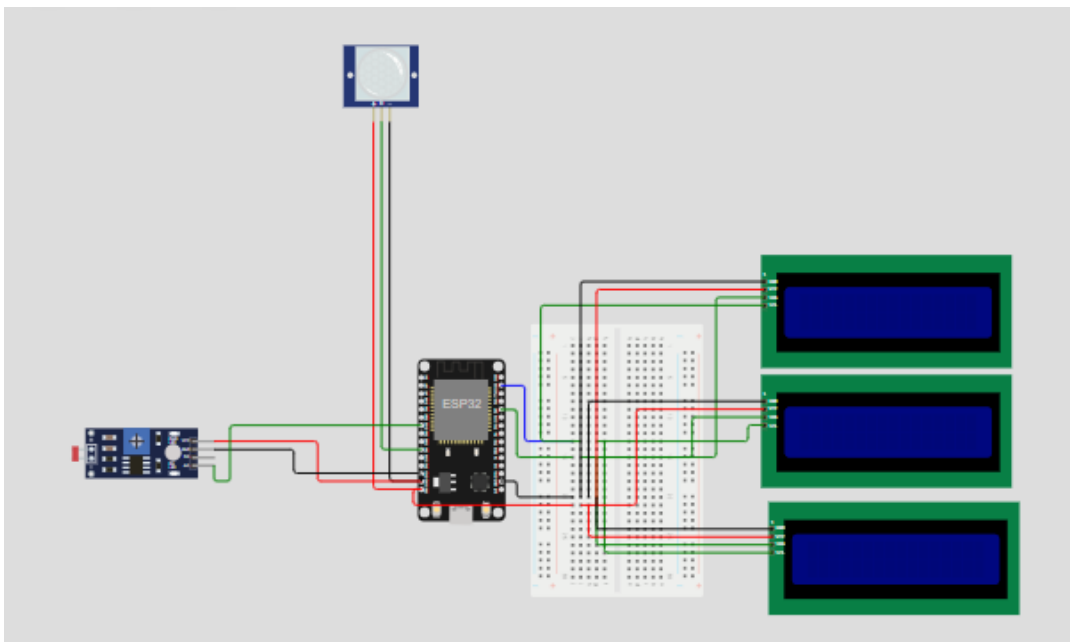
- **TaskLuminosity :**  
Lit la luminosité ambiante à partir du capteur.  
**Périodicité :** périodique  
**Priorités :** priorité normale
- **TaskTelevision :**  
Gère l'allumage des écrans LCD en fonction de la luminosité détectée.  
**Périodicité :** périodique  
**Priorités :** priorité basse  
**Délais :** 2 heures
- **TaskHandleMotion :**  
Gère l'affichage d'un message d'urgence en cas de détection de mouvement.  
**Périodicité :** non périodique, asynchrone

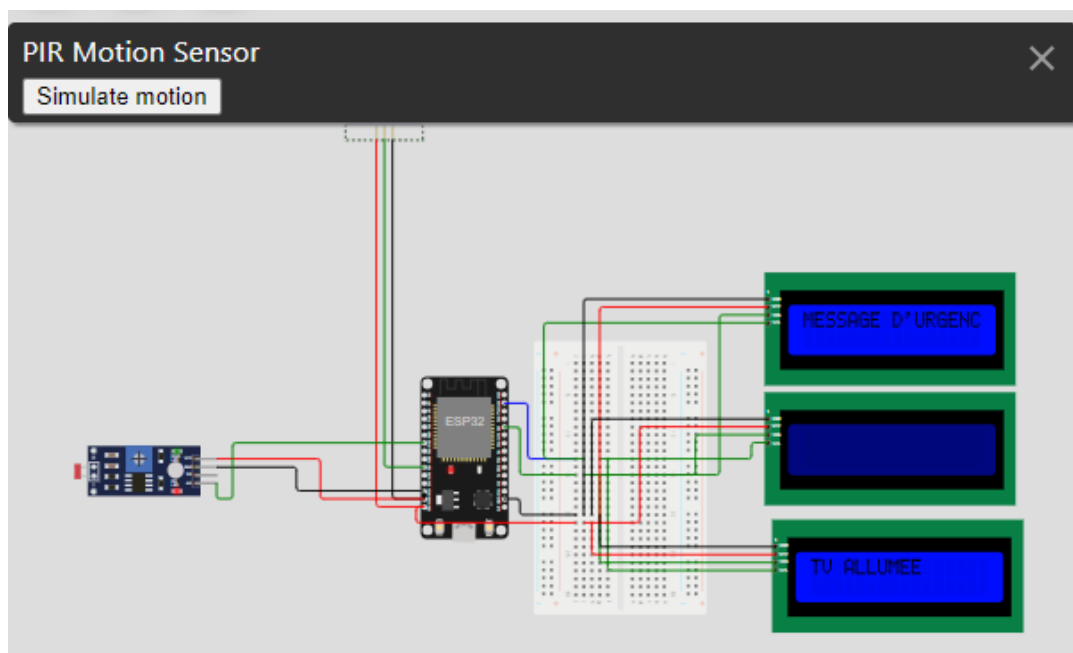
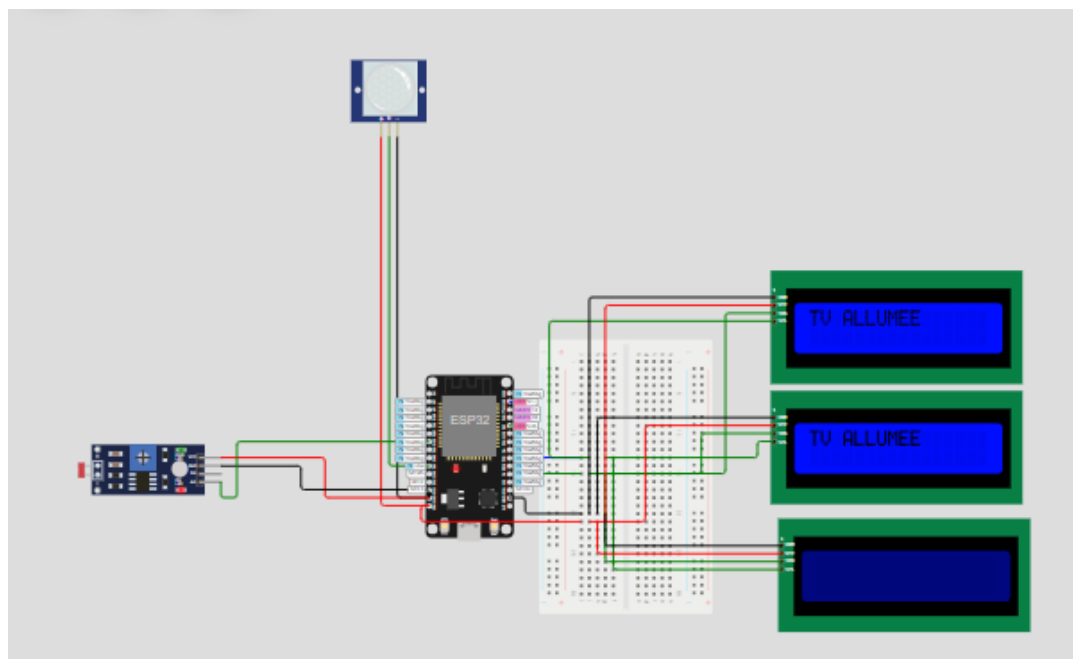
**Priorités** : priorité maximale

#### 4. Programmation et simulation

[Lien vers la simulation sur Wokwi](#)

#### 5. Captures montrant les différents états du système





## 6. Résumé du projet

Ce projet implémente un système temps réel utilisant des écrans LCD ainsi que des capteurs de luminosité et de mouvement. Dans un premier temps, le système gère l'affichage sur trois écrans LCD en fonction de la luminosité ambiante et ensuite détecte les mouvements pour activer un message d'urgence.

Un capteur de luminosité mesure l'intensité lumineuse. Si la luminosité dépasse le seuil défini, `SEUIL_LUMINOSITE`, le système considère qu'il fait jour et envoie cet événement à la queue. En revanche, si la luminosité est inférieure au seuil, le système éteint les écrans LCD.

En ce qui concerne les écrans LCD, trois sont utilisés pour afficher des informations. Ainsi, lorsque la luminosité indique qu'il fait jour, un message "TV ALLUMEE" est affiché sur les écrans pendant une durée définie par un timer. Après cette durée, les écrans s'éteignent automatiquement.

De plus, un capteur de mouvement détecte toute présence dans la zone. Dès qu'un mouvement est détecté, un message d'urgence est affiché sur l'écran 1.

Enfin, le projet utilise FreeRTOS pour gérer les différentes tâches telles que la lecture de la luminosité, la gestion des écrans et la détection de mouvement de manière concurrente. Pour ce faire, chaque tâche s'exécute sur un cœur spécifique du microcontrôleur.

## 7. Erreurs rencontrées

### Build failed!

```
sketch.ino:2:10: fatal error: FreeRTOS: No such file or directory
#include <FreeRTOS>
          ^~~~~~
compilation terminated.
Error during build: exit status 1
```

Cette erreur a été résolue en retirant l'inclusion explicite de la librairie FreeRTOS.