# PROJET 2 – SNIOT : Maison connectée avec ZephyrOs et ESP32

## Introduction

L’objectif est de découvrir le système d’exploitation embarqué temps réel Zephyr. Vous allez découvrir l’environnement de développement en réalisant un projet de maison connectée. Vous allez apprendre à :

* Réaliser une description matérielle du système dans le Device Tree
* Utiliser les fonctionnalités d’un système temps réel (Programmation Concurrente, Thread, Sémaphore, Mutex …)
* Utiliser le système de driver
* Communiquer des données sans fil
* Créer une API en python

Les rendus attendus sont :

* Le report GitHub avec l’historique du travail
* Le rapport de conception rempli
* Une démonstration du programme

Un répertoire github contient les fichiers importants pour la réalisation du projet : <https://github.com/Vivoulia/sniot-zephyr.git>

Durée du projet : 3 séances

## Cahier des charges

|  |  |  |
| --- | --- | --- |
| **Fonction** | **Nom** | **Description** |
| F1 | Acquisition de la température et de l’humidité | Acquisition de la température et de l’humidité toutes les 10s avec le capteur XHT11 |
| F2 | Acquisition de la quantité de « vapeur » | Lecture analogique de la quantité de vapeur avec le Steam Sensor toutes les 10s |
| F3 | Allumage LED démarrage | Allumage de la LED Orange au démarrage du système |
| F4 | Système d’alarme d’intrusion | L’appui sur le bouton gauche place la maison en mode « Alarme ». Si un intru est détecté avec le capteur PIR un signal sonore à une fréquence de 1000Hz sonne tant que l’intru n’est pas parti. |
| F5 | Affichage LCD | Un message d’accueil s’affiche au lancement. Un message s’affiche en mode Alarme. Un message s’affiche quand un intru est détecté. |
| F6 | Envoi des données par Wifi | La température et l’humidité sont envoyées par Wifi à un serveur. |
| F7 (bonus) | Transmission NFC | Utiliser la puce NFC pour rediriger l’utilisateur vers un site internet par exemple |

## Rapport de projet

### Préliminaires

1. Décrire avec vos mots à quoi sert le mot-clé volatile et quand l’utiliser en programmation embarqué ?

Le mot-clé "volatile" en programmation est utilisé pour indiquer au compilateur que la valeur d'une variable peut changer à tout moment.

Il est principalement utilisé en programmation embarquée, où les variables peuvent être modifiées par des interruptions matérielles ou d'autres processus en dehors du contrôle direct du programme principal.

Lorsque vous déclarez une variable comme "volatile" dans un code embarqué, cela signifie que le compilateur ne doit pas optimiser ou mettre en cache cette variable. Cela garantit que le programme lit toujours la valeur actuelle de la variable à partir de la mémoire.

1. Quel outil pouvez-vous utiliser pour protéger une variable partagée entre deux threads ?

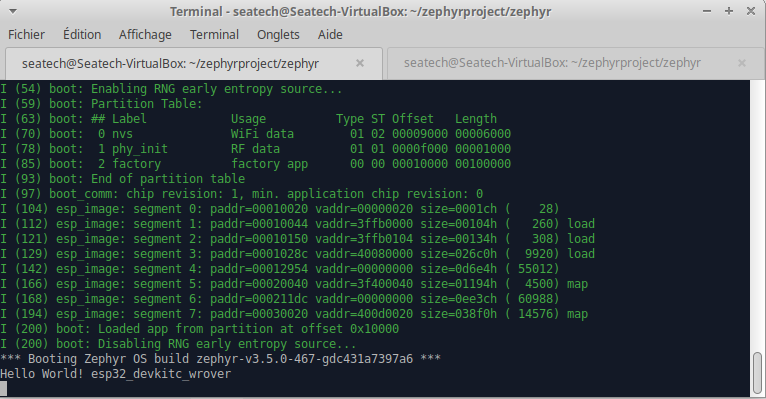
Voici les outils :

- Mutex (Verrous) : mécanisme de synchronisation qui garantit qu'un seul thread à la fois peut accéder à la ressource partagée.

- Sémaphores : pour implémenter différentes stratégies de contrôle d'accès, comme l'exclusion mutuelle.

En utilisant des mutex ou des sémaphores, vous pouvez éviter les conditions de concurrence (race conditions) et garantir que les threads accèdent à la variable partagée de manière coordonnée et sécurisée.

3. Flash Esp32



### Allumage de la LED de démarrage

La LED orange est sur le GPIO 12. Ajouter la description de la LED dans la DeviceTree grâce à un fichier **d’overlay**. Récupérer le GPIO avec la macro prévue à cet effet dans le code. Initialiser le GPIO à l’état haut au démarrage.

### Affichage LCD

Votre collègue Bruno a développé un code pour le driver de l’afficheur I²C LCD 1602. Cependant il y avait un afterwork hier soir et vous le soupçonnez d'avoir bâclé le travail. Le driver contient probablement des erreurs faites attention !

Le driver est disponible sur le répertoire github du projet. Intégrer le driver à votre code en corrigeant les potentielles erreurs.

### Acquisition de la température et de l’humidité

Le capteur de température et d’humidité utilisé est le DTH11. Il fonctionne avec un bus de donnée sur une seule ligne GPIO. Un protocole bien spécifique est défini dans la datahseet du capteur. Nous n’allons heureusement pas implémenter ce protocole car Il existe déjà un driver Zephyr.

Ajouter dans l’overlay de votre projet un GPIO pour le capteur avec la syntaxe Figure 1 puis récupérer le *device* dans votre code.

En utilisant l’interface sensor de Zephyr et les fonctions sensor\_sample\_fetch, sensor\_channel\_get et sensor\_value\_to\_double ainsi que la documentation <https://docs.zephyrproject.org/latest/hardware/peripherals/sensor.html>, récupérer et afficher une fois toutes les 10 secondes les valeurs de température et d’humidité.



Figure 1: Syntaxe pour le pin GPIO dans l'overlay de votre projet

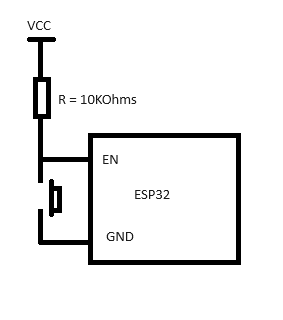
### Acquisition de l’humidité analogique

En utilisant l’exemple [exemple adc Zephyr](https://docs.zephyrproject.org/latest/samples/drivers/adc/README.html) implémenter la lecture analogique de l’humidité avec le « steam sensor ».

### Détection de l’appui bouton

1. Dessiner le schéma électronique classique permettant de relier un bouton à un microcontrôleur.

Voici comment vous pouvez relier un bouton à un microcontrôleur :



Matériel nécessaire :

* + Un microcontrôleur
  + Un bouton-poussoir
  + Une résistance : une résistance de pull-up ( 10 kOhms).
  + Fils de connexion.

Ce schéma permet d'utiliser le bouton-poussoir pour détecter quand il est enfoncé. Lorsque le bouton est enfoncé, il établit une connexion entre la broche du microcontrôleur et la masse (GND), ce qui déclenche un niveau logique bas. Lorsque le bouton est relâché, la résistance de pull-up ramène le niveau logique à haut (VCC)

1. A quel phénomène faut-il être vigilant lorsqu’on cherche à détecter l’appui bouton dans le programme ?

Lorsque vous cherchez à détecter l'appui d'un bouton dans un programme, il faut être vigilant vis-à-vis du phénomène de rebond mécanique . Le rebond mécanique est un problème courant lors de l'utilisation de boutons-poussoirs ou d'interrupteurs mécaniques. Il se produit lorsque le bouton est enfoncé ou relâché, et pendant ce processus, les contacts mécaniques à l'intérieur du bouton font généralement des contacts et des interruptions très rapides et instables.

1. Quand vous utilisez une variable dans une interruption quel mot-clé doit être utilisé pour déclarer cette variable ?

Lorsque vous utilisez une variable dans une interruption en programmation, il est généralement recommandé d'utiliser le mot-clé "volatile".

Le mot-clé "volatile" informe le compilateur que la valeur de la variable peut être modifiée en dehors du flux de contrôle normal du programme, notamment par une interruption matérielle. Cela empêche le compilateur d'effectuer des optimisations qui pourraient rendre les accès à la variable incohérents.

Deux boutons sont présents sur le système sur les pins 16 et 27. Ajouter les dans l’overlay de votre projet. N’oublier pas d’ajouter un alias. Récupérer ensuite le pin gpio dans le programme. Initialiser le GPIO et configurer une interruption. Écrire un code permettant de détecter les appuis bouton. Vérifier le bon fonctionnement de ce code dans le moniteur.

Nous allons maintenant rajouter une fonctionnalité d’affichage LCD reliée à l’appui bouton. Un code propre et bien conçu ne doit pas contenir de longues procédures dans les interruptions. La fonctionnalité d’affichage sera donc placée dans un processus autre que la fonction d’interruption.

### Buzzer

Le buzzer est relié au pin 25. Ajouter le GPIO dans l’overlay de votre projet et initialisé le dans votre code. Le buzzer ne contient pas d’oscillateur intégré. Vous devez donc programmer une PWM permettant de générer le son à la fréquence demandée. Créer un thread permettant de faire osciller le buzzer. Afin d’éviter d’avoir mal à la tête à la fin de la journée ajouter une configuration permettant de désactiver le buzzer. Relier l’activation du buzzer avec le bouton. Tester en affichant un message sur l’écran LCD.

### Capteur de présence et système d’alarme

Maintenant que le buzzer fonctionne il reste à ajouter le capteur de présence qui est sur le pin 14. Ajouter le GPIO correspondant au projet. Pour utiliser le capteur, il suffit de lire la valeur renvoyée par celui-ci :

* 0 : Pas de présence
* 1 : Présence détectée

Vérifier le bon fonctionnement du capteur puis réaliser le système d’alarme complet. Merci de ne pas utiliser le buzzer pendant la phase de test mais seulement des print ou l’écran LCD.

Réaliser l’ensemble du cahier des charges à l’exception de la partie Wifi.