

Objectif

Le but de ce TP est de réaliser un capteur de température/humidité sur la carte fournie. Les contraintes fonctionnelles sont les suivantes:

- Communiquer les valeurs sous un format lisible sur l'UART
- Au démarrage donner une valeur, puis au maximum une valeur toutes les 5 minutes, et au minimum toutes les heures
- Ne donner de valeur que si la température change d'un demi degré / l'humidité de 5%
- Le CPU doit être au repos autant que possible, on souhaite faire fonctionner la carte sur pile

Palier 1: démarrage de la carte

La première étape est de faire démarrer la carte sur un programme qui s'exécute uniquement en flash, et qui utilise des variables globales. Vous devez pour ça fournir les éléments suivants:

- un Makefile
- un linker script
- du code d'initialisation

Les règles obligatoires du Makefile sont les suivantes:

- all: compile les sources et génère de quoi charger la flash ou débbugger
- download: charge le fichier compilé en flash
- run: fait démarrer la carte
- debug: lance gdb avec le dernier binaire chargé en flash, pret à être débbugué

Pour charger la carte, vous utiliserez les carte discovery en mode SWD, et vous connecterez les fils comme indiqué sur l'image ci-dessous.

Le logiciel à utiliser est laissé à votre choix, vous pouvez utiliser les utilitaires de ST qui ont été porté [ici](#), ou bien openocd.

Palier 2: Boucle principale

Ce palier met en place la boucle principale du système. Maintenant que vous êtes capables de charger un programme en flash et de le débbugger, vous allez activer les interruptions et gérer l'activité du CPU.

Deux éléments importants dans un système embarqué sont la gestion du tick timer ainsi que du watchdog. Aussi pour ce palier, vous allez activer le SysTick et le watchdog. Le but est d'avoir un système qui tourne, et de pouvoir à tout moment savoir depuis combien de temps le système a démarré.

Il sera appréciable de ne pas faire de l'attente active hors des ISR.

Palier 3: UART

Pour ce palier, vous allez devoir implémenter un mini driver UART, qui se contentera de renvoyer tous les caractère qu'il reçoit. Comme pour le palier précédent, on préférera se réveiller sur un évènement plutôt que de faire de l'attente active.

Palier 4: Bootloader

Cette étape consiste à développer un bootloader qui sera installé en flash. Ce bootloader devra avoir les propriétés suivantes:

- initialiser la mémoire
- charger un binaire en flash

Vous devrez proposer une interface pour télécharger un binaire via l'UART, et l'écrire en flash.

Par ailleurs vous devrez rajouter la règle suivante à votre Makefile:

- bootloader: compile le bootloader et le charge en flash

Il sera donc nécessaire de fournir un linker script différent pour le bootloader et l'application.

Palier 5: Capteur T/H

Lors de cette dernière étape, vous allez devoir lire les données du capteur HTS221 pour en déduire la température et l'humidité. Il vous faudra pour cela faire un driver I2C. Vous pourrez utiliser le code de la bibliothèque stm32, et vous devrez lire avec attention le fonctionnement du port I2C du capteur dans la datasheet fournie. Dans un deuxième temps vous devrez faire une interpolation linéaire de la valeur relevée par rapport aux données de calibration, et gérer les contraintes fonctionnelles données en début de sujet.

Vous noterez que le port I2C utilisé par le HTS221 est le numéro 2.

Ressources:

Pour vous aider à développer, vous pouvez vous référer aux documents suivants:

- [Documentation Cortex M-3](#)
- [Documentation STM32151](#)