

Prise en main pickit et cartes LED's CHAT

Dans ce TP nous allons prendre en main le cycle de développement sans boot-loader pour des processeurs PIC32MX. Pour votre plaisir, nous utiliserons des modules issus de l'exposition LED's CHAT en 2013 qui sont dotés de 31 LEDs RVB, des capteurs IR et des portes de communication en série. L'objectif de la série de TP est d'écrire un bootloader sécurisé pour ces cartes qui passe par les portes de série.

Pour démarrer, téléchargez l'archive BaseProjectSSE.tar.gz. Il contient un Makefile qui s'occupe des principales tâches que vous avez à effectuer : compiler et linker les sources, récupérer, au besoin, le compilateur et la bibliothèque Harmony de Microchip dans /tmp. Regardez notamment l'effet des commandes `make`, `make deploy`, et `make clean`.

Il est important de passer un peu de temps à lire la documentation disponible sur microprocesseurs, ainsi que sur Harmony. Tout est accessible à partir de la page Systèmes Embarqués.

I Triangles LED's CHAT

Ces triangles sont dotés d'un microcontrôleur PIC32MX795F512H, de 31 LED's RVB avec des puces drivers, des connecteurs pour la liaison entre eux pour la communication et l'alimentation, des capteurs infrarouges, et d'un connecteur de programmation.

Ce dernier est au milieu de la carte (une suite de 6 trous. Le trou le plus excentré de la carte correspond au Pickit à la marque (triangle blanc).

Les connecteurs disposent de 7 contacts, de gauche à droite : DIN, GND, 5V, 3.3V, 5V, GND, DOUT. Vous pouvez alimenter un triangle isolé à l'aide d'une source alimentée par USB, par exemple les cartes Pinguino. Cherchez sur celles-ci (documentation) les tensions GND, 3.3V, 5V. et reliez-les à la carte LED's CHAT par un connecteur.

A terme, nous utiliserons les cartes Pinguino pour communiquer avec les cartes LED's CHAT.

Une fois alimentée, le programme de la carte LED's CHAT se met en marche. Il dépend de la programmation si cela se voit.

Par un deuxième câble USB, utilisez le pickit pour déployer (`make deploy`) l'application fournie.

Assurez-vous avant de brancher le pickit :

- Que la carte LED's CHAT est alimentée.
- Que vous posez le pickit grâce à un *peigne* dans le bon sens sur le connecteur de programmation (triangle blanc vers l'extérieur!).

Puis, lancez la programmation tout en tenant le pickit avec une petite tension contre les contacts du connecteur (s'assurer du contact). En cas d'échec de programmation, réessayer (parfois, des faux contacts empêchent la bonne communication).

II Linker scripts, hexfiles, mapfiles et Cie

Regardez ce qui se passe lors du processus de compilation. Celui-ci utilise un script `.ld` par défaut. Comparez ses définitions avec les zones de mémoire décrites dans la documentation du PIC32MX.

Regardez également la sortie produite par `bin2hex` ainsi que le fichier `.map` dans le dossier `dist`. Il s'agit des comptes rendus des bouts de code placés par ici et par là dans la mémoire lors de la programmation.

Où sont placées les fonctions et variables du projet? Et dans quel ordre? Faites une copie du linker script et modifiez le Makefile pour utiliser la copie. Modifiez des emplacements mémoire et observez les effets sur les adresses générées. D'après vous, qu'est-ce que vous pouvez modifier, et qu'est-ce qui ne peut pas bouger?!