

Descriptif de la formation

L'engouement pour les cibles embarquées programmables n'est plus à démontrer et nous pouvons voir dans le monde de l'Enseignement Supérieur et de la Recherche de plus en plus l'utilisation de ces systèmes pour la réalisation d'instruments scientifiques. Il existe aujourd'hui des kits faible coût-consommation, facile à prendre en main mais souffrant de certaines lacunes au niveau des performances notamment ils pour des applications complexes. Or, aujourd'hui il est bien souvent primordial de développer une instrumentation adaptée temps-réel, multitâche et utilisant les mêmes architectures et protocoles que les machines virtuelles utilisant des OS plus classiques. La grande diversité des kits et outils d'environnement de programmation sur le marché permet un grand degré de liberté pour choisir le matériel et le langage afin d'aboutir à des temps d'exécution et de calcul proche d'un code écrit en langage bas niveau.

Cette formation a pour but de faire connaître différentes cibles embarquées programmables en utilisant différents langages et outils d'environnement de programmation dédiés à un échantillon de kits issus du commerce et de tenter de démystifier leurs prises en main.

Cette ANF est constituée d'un module de 4h.

Descriptif des cibles/modules/ kits/IDE (sous windows) présentés

Arduino / Seeeduino

Arduino/Seeeduino sont des circuits imprimés (matériel libre) sur lesquels on retrouve un microcontrôleur programmable (type Atmel AVR : ATmega328 ou ATmega2560 pour les versions récentes) et une interface entrée/sortie simple pour analyser et produire des signaux électriques, contrôler des appareils domestiques, piloter des modules, etc. Le microcontrôleur est préprogrammé avec un bootloader de façon à éviter l'utilisation d'un programmeur dédié. Les modules sont programmés à travers une connexion série RS232-USB. L'Arduino/Seeeduino possède plusieurs entrées/sorties numériques, dont certaines peuvent produire des signaux PWM et entrées analogiques. Le logiciel de programmation est une application Java, libre et multiplateformes servant d'éditeur de code et de compilateur, et pouvant transférer le firmware via une liaison série RS232, Bluetooth ou USB. Le langage de programmation utilisé est le C/C++, compilé avec avr-g++ et lié à la bibliothèque de développement Arduino. Nous présenterons l'IDE Arduino à travers un ou deux exemples.

CF2 Persistor

CF2 Persistor est un ensemble ordinateur électronique miniature et système de logiciels conçus pour réaliser des acquisitions de données faible puissance (<250mW) et des applications de contrôle. Programmable en langage C à travers l'IDE CodeWarrior Development Studio (pour Palm OS), constitué d'un système d'exploitation PicoDOS, d'une bibliothèque riche de pilotes matériels et de fonctions spécifiques, pour la gestion de stockage sur carte CompactFlash (format FAT32), le CF2 Persistor permet le développement de projets personnalisés simples à

des projets de collecte de données très complexes. Il réduit le temps de développement et les risques (il est très fiable) en proposant un environnement de développement mature, de qualité industrielle, ainsi qu'un riche ensemble de fonctionnalités pour le contrôle des périphériques I/O. Le CF2 Persistor dispose d'un Motorola 68332 en tant que contrôleur principal. Les ressources comprennent un UART, un bus SPI, un PIT, une horloge système réglable (jusqu'à 16MHz), etc. Nous évoquerons un ou deux exemples dans une démonstration.

[EK-LM3S8962 \(Texas Instruments\)](#)

Le kit d'évaluation Stellaris LM3S8962 fournit une plate-forme d'évaluation compact et polyvalent pour Ethernet + CAN sur ARM Stellaris® microcontrôleurs-M3 Cortex™. Chaque carte possède une Debug Interface In-Circuit (ICDI) qui fournit des fonctionnalités de débogage matériel non seulement pour les périphériques Stellaris à bord, mais aussi pour toute carte cible basée sur microcontrôleur Stellaris. Les kits d'évaluation contiennent tous les câbles, les logiciels et la documentation nécessaires pour développer et exécuter facilement et rapidement des applications pour microcontrôleurs Stellaris. Une carte d'évaluation comprend un Stellaris LM3S8962 (ARM), un contrôleur Ethernet 10/100 intégré, un écran OLED graphique avec une résolution de 128 pixels x 96, une LED utilisateur, des commutateurs de navigation (boutons-poussoirs), un haut-parleur magnétique, etc. La carte peut être programmée en langage C (compilateur Keil) ou directement en LabVIEW en utilisant le module LabVIEW for embedded (ARM) que nous verrons brièvement dans cette formation.

Développement sous ANDROID Ice Cream Sandwich (version 4.x.x)

[SMARTPHONE / TABLETTE Android seuls](#)

Prenons la mesure des choses, il y aurait aujourd'hui plus de 400 millions d'appareils sous Android. Le développement d'applications sur cet OS devient une solution pour de nombreux dispositifs. Il s'appuie généralement sur 3 points : les langages Java, XML et l'environnement de développement SDK (libre) qui fournit l'API de programmation, l'émulateur, la documentation et des exemples. Nous aborderons ici les bases pour la mise en place de cet environnement SDK, intégré à Eclipse et nous présenterons une ou deux applications utilisant un Smartphone avec l'un des ses périphériques (accéléromètre, pression, etc.).

[SMARTPHONE / TABLETTE Android + module externe](#)

Il est aujourd'hui possible de contrôler et de gérer des dispositifs externes à l'aide d'appareil exploitant l'OS Android. Ces dispositifs ne se limitent pas qu'aux protocoles et capteurs déjà présents dans ces appareils mais utilisent également d'autres périphériques pouvant s'y connecter. L'avantage est de pouvoir profiter de toute la puissance, des ressources et des protocoles d'un appareil sous Android afin de contrôler et de communiquer avec des équipements (ou capteurs) externes. L'inconvénient est le développement multiple d'une application sous Android, d'un firmware et de cartes électroniques pour l'interfaçage entre l'appareil Android et l'équipement externe. Nous présenterons ici quelques exemples simples.