
CPE Lyon - 4IRC - S7 - Année 2022/23

Architecture et Protocoles Réseaux pour l'IoT

TP 1 - Prise en main d'un micro-contrôleur

Ce TP a pour but deux objectifs :

1. Vous faire découvrir la programmation d'un micro-contrôleur.

Analyse et préliminaires

Exercice 1. Étude du système

La première étape lorsqu'on veut développer un système embarqué est de sélectionner le matériel qui sera utilisé. Cela passe par la définition des besoins, puis l'étude des solutions existantes.

Dans notre cas, le besoin est relativement simple : la découverte de la programmation sur micro-contrôleur. Pour ce qui est du choix du système, il a été fait pour vous, ce qui est bien souvent le pire des cas possible, car vous n'avez pas été consultés en amont, et devez donc vous débrouiller avec ce qui vous est imposé.

Dans le cas de ce TP, quatre choix étaient possibles.

- Utiliser des cartes C8051F02x de SiliconLabs.
- Utiliser des cartes Arduino Uno.
- Utiliser des cartes STM32 de ST Micro-electronics.
- Utiliser des cartes Micro :bit

Question 1. Rechercher les caractéristiques des diverses cartes en question et les micro-contrôleurs utilisés par chacune d'entre elles.

Le choix c'est porté sur la quatrième solution, car le micro-contrôleur utilisé par ces cartes est beaucoup plus adapté et répandu pour le développement de systèmes embarqués que ceux des deux premières solutions, qui utilisent des micro-contrôleurs 8 bits, coûteux, et peu puissants.

Les deux autres solutions utilisent des micro-contrôleurs de même architecture, mais notre choix s'est portée vers les cartes Micro :bit.

Exercice 2. Documentations techniques

La deuxième étape lorsqu'on veut écrire du code pour un système embarqué, c'est de trouver la documentation technique du micro-contrôleur.

L'accès aux documentations techniques est un des critères principaux dans le choix des composants pour un projet, aussi bien pour les concepteurs que pour les utilisateurs.

Si vous avez réalisé la conception du circuit, vous aurez normalement déjà eu besoin de ces documentations, dans le cas contraire, c'est le moment de les récupérer. Pour le module Micro :bit, il y a plusieurs documentations techniques nécessaires pour utiliser la totalité des fonctionnalités du module.

Pour récupérer les documentations techniques, vous avez le choix entre le site du fabricant et celui du distributeur chez qui vous avez achetés les composants. Pour la majorité des composants on utilise le site du distributeur (possible uniquement s'il fournit les bonnes documentations), mais pour les micro-contrôleurs il est préférable de consulter le site du fabricant car il fournit aussi les "errata" (notes d'informations contenant les corrections sur la documentation) et le plus souvent des notes d'application qui indiquent comment utiliser le composant pour telle ou telle application (avec parfois des exemples de code).

Question 1. Rechercher les différentes documentations techniques pour la carte Micro :bit et de ses composants. Est-ce que le site du distributeur (BBC) propose des documentations plus complètes que ceux des fabricants des composants ?

Outils de programmation

La troisième étape ne concerne toujours pas le code que vous voulez écrire.

Pour écrire du code vous n'avez pas besoin de la "cible" (target en anglais) qui est le matériel sur lequel vous allez exécuter la version compilée du code, mais uniquement d'un poste de développement, couramment appelé hôte (host en anglais).

Cependant, la cible en question n'a que faire des fichiers sources qui se trouvent sur votre poste de développement, et il vous faudra un certain nombre d'outils pour passer des fichiers source (quelque soit le langage) au code exécutable par votre micro-contrôleur.

C'est aussi un point très important dans le choix d'un système embarqué ou d'un micro-contrôleur :

Question 2. Quels sont les outils dont j'aurais besoin pour passer de mon code source à un système fonctionnant avec la carte Micro-bit ?

Environnement de travail

Pour la suite de TPs micro-contrôleurs, vous allez adapter votre environnement de travail à vos préférences :

- **Attention :** La carte micro :bit peut être programmée en plusieurs langages, on vous conseille d'utiliser C++ (moins gourmand en ressources et plus rapide à exécuter). Plus d'information sur : <https://lancaster-university.github.io/microbit-docs/>
- Éditeur de texte
- OS : Linux native ou VM

Matériel : un port USB (et le PC qui va avec)

Pour ce qui est de la liaison de la carte à votre ordinateur, le port USB sera le choix de prédilection, cette carte peut être programmée par Bluetooth mais ça sera moins performant.

Test à réaliser

Vous allez tester l'environnement de développement avec quelques exercices d'accès aux différents capteurs intégrés. Vous pouvez trouver du code source de départ sur : <https://git.mob-dev.fr/Schoumi/microbit-samples>

Exercice 3. Test du module

Commencez par vérifier que votre module fonctionne (rien de plus frustrant que d'essayer de programmer un micro-contrôleur qui ne réponds pas).

Exercice 4. Interaction avec les LEDs embarquées

Tester l'écran des LEDs embarquées en affichant un message qui va changer une fois les boutons intégrés sont pressés.

Exercice 5. Capteur de température

Tester l'accès au capteur de température et affichez la valeur et une alerte si la température dépasse un seuil de 30 degrés.

Exercice 6. Capteur accéléromètre

Le capteur mesure les données des mouvements en trois axes (x,y,z), faites une application qui affiche le nombre de fois que le capteur est incliné à sa gauche

Exercice 7. Capteur boussole

Créer une application boussole en utilisant le capteur magnétomètre intégré est affichant le Nord dans la matrice de LEDs