

Application du C++ au domaine des objets connectés

Année 2017-2018

Ce bureau d'étude est supporté par la société INTEL à travers le don de cartes Edison/Galiléo et de kit de capteurs/actionneurs.

ATTENTION : Il vous est demandé de ne pas mettre à jour le firmware des cartes ni de changer le mot de passe du compte « root »

1. Objectifs

Ce Bureau d'étude a plusieurs objectifs :

- Réfléchir à une conception orientée objet
- Utiliser le langage C++
- Développer une bibliothèque modulaire, extensible pour s'interfacer avec des capteurs et des actionneurs
- Imaginer un problème à résoudre ou un nouveau service que vous pouvez créer dans le monde des objets connectés et de l'Internet des Objets
- Développer une application attractive pour utiliser votre bibliothèque
- Faire le lien avec la journée Internet des Objets organisée à l'INSA en avril/mai
- Pour les meilleures réalisations, servir de vitrine au DGEI

2. Déroulement

Ce BE se déroule sur 5 séances de 2h45. Lors de la première séance, Il vous sera confié un kit IoT comprenant une carte edison, un ensemble de capteurs et actionneurs ainsi que tous les câbles nécessaires. Ces équipements seront sous votre responsabilité et vous pourrez les garder avec vous durant toute la durée du projet.

Si vos idées nécessitent des extensions en termes de capteurs ou d'actionneurs, n'hésitez pas à le faire remonter au plus tôt à vos enseignants, nous essayerons de vous fournir le matériel nécessaire, si possible entre la première et deuxième séance.

Après la dernière séance de BE de votre groupe lors de la séance d'examen vous devrait rendre :

- votre réalisation en état de fonctionnement pour évaluation
- l'ensemble de vos fichiers source dans une archive
- un rapport décrivant votre conception (cas d'utilisation, scénarios, diagramme de classe, diagramme de séquence, etc), le déroulement du projet, comment faire marcher votre système, toute analyse intéressante sur un maximum de 6 pages.
- Une photo de votre réalisation (à envoyer à monteil@insa-toulouse.fr)
- Une vidéo de 2minutes maximum mettant en valeur votre projet à destination des journées portes ouvertes de l'INSA (à mettre sur un support de diffusion de vidéo si possible et à envoyer à monteil@insa-toulouse.fr)

- La notation se fera par vos enseignants de TP sur la base de cette grille qui sera affinée.

Projet fonctionnant	5
Respect des principes de l'objet	2
Respect du cahier des charges en terme d'utilisation du C++	4
Programme propre et commenté	2
Qualité et quantité du projet	2
Idée intéressante en terme d'application	2
Rapport clair et intéressant	3
Non respect des échéances	-5

Pour respecter le cahier des charges en terme d'utilisation du C++, vous devrez utiliser dans votre programme, les concepts suivant du C++ :

- Création de plusieurs classes
- Utilisation du mécanisme d'héritage
- Redéfinition d'opérateur
- Utilisation de la STL
- Utilisation des exceptions

3. Contenu des séances

Ce découpage est indicatif vous pouvez prendre de l'avance.

- Première séance

Cette séance sera dédiée à la prise en main de la carte Edison et de ses capteurs actionneurs ainsi que le début de la réflexion sur une bibliothèque objet. Nous ne programmerons pas en utilisant la bibliothèque arduino mais directement Gpio (General Purpose IO interface) pour attaquer la bibliothèque libmraa.

i. Découverte du matériel

- Carte edison : des informations diverses se trouvent ici : <http://arduino.cc/en/ArduinoCertified/IntelEdison>
- La carte d'entrée/sortie : l'ensemble des informations sont disponibles à l'adresse suivante : http://www.seeedstudio.com/wiki/Grove_-_Starter_Kit_Plus

ii. Connexion

1. Connexion via le wifi

Les cartes sont configurées pour être accessible via le réseau wifi IOT au sein du DGEI. Vous trouverez les adresses des cartes en annexe. Vous pouvez vous connecter via ssh en utilisant le compte root, le mot de passe est insainsa.

2. Connexion via un câble série

Vous pouvez aussi vous connecter via le port uart de la carte :

- Ouvrir un terminal sur le PC
- Connecter le port Uart (prise jack sur la galileo) à une prise USB de votre PC
- Connecter le câble réseau

- Brancher la led sur la prise D2
- Alimenter la carte galiléo
- Regarder sur votre PC le port utilisé par la carte en utilisant la commande :
dmesg qui vous donne les derniers messages du système vous verrez le port usb à utiliser.
Remplacer XX dans la commande ci-dessous par le port trouvé
- Taper dans le terminal de votre PC la commande :
screen /dev/ttyUSBXX 115200

Vous avez maintenant une console ouverte sur la carte, si elle est encore en train de démarrer vous devez voir les messages de démarrage sinon vous voyez le prompt de login

iii. Programmation

- Vous trouverez à cette adresse une description des fonctions utilisées ci-dessous :
http://iotdk.intel.com/docs/master/mraa/gpio_8h.html
- Utiliser l'éditeur vi par exemple pour écrire le programme suivant permettant de faire clignoter la led branchée précédemment :

```
#include <mraa/aio.h>
#include <mraa/gpio.h>

int main(){
    mraa_gpio_context m_gpio;

    // initialisation
    mraa_init();
    m_gpio = mraa_gpio_init(2);
    mraa_gpio_dir(m_gpio, MRAA_GPIO_OUT);
    // allumage et extinction
    mraa_gpio_write(m_gpio, 1);
    sleep(2);
    mraa_gpio_write(m_gpio, 0);
    sleep(2);
    mraa_gpio_write(m_gpio, 1);
    // fin
    mraa_gpio_close(m_gpio);

    return(0);
}
```

- Compiler votre programme sur la carte avec la commande suivante :
g++ monprog.C -lmraa -Wall

puis exécuter le. Vous trouverez d'autres exemples plus complexes sur moodle dans le cours de C++ (écran lcd, servo moteur).

iv. Tester l'ensemble des capteurs/actionneurs via des petits programmes

v. Arrêt de la carte

Vous pouvez arrêter proprement la carte avec la commande « halt » ou bien « reboot » si vous voulez la redémarrer.

- **Deuxième séance**

L'objectif de cette séance est de mettre en place une bibliothèque en C++ pour manipuler l'ensemble des senseurs et actionneurs de manière aisée et standard. Réfléchissez aux attributs communs, aux informations complémentaires intéressantes, aux méthodes communes et différentes. Ceci doit vous amener à proposer une architecture de classes avec des classes héritées.

Dans le rapport, vous devrez justifier ce choix et commentez l'arborescence de classe que vous avez créé. Ecrivez et testez votre code.

- **Troisième séance**

Dans cette séance, vous allez finir de coder votre bibliothèque et commencez à réfléchir à une application à même de tirer parti des senseurs et actionneurs.

- i. Vous définirez les use cases montrant comment votre application va fonctionner, vous pouvez fournir quelques scénarios.
- ii. Réfléchissez ensuite aux principales classes, attributs et méthodes nécessaires pour construire votre application. Ces 2 points alimenteront aussi votre rapport.
- iii. Valider votre approche avec l'encadrant de TP
- iv. Commencez à coder votre application

- **Quatrième séance / cinquième séance**

Finaliser votre application et votre rapport.

4. Remarques complémentaires :

- Le noyau linux configuré sur votre carte permet aussi un accès réseau. Vous pouvez récupérer l'adresse de la carte avec la commande ifconfig
- Vous pouvez utiliser des commandes réseaux si votre réseau hôte le permet : ssh (pour avoir un autre terminal en vous connectant), scp (pour copier des fichiers), etc
- Si possible mettez la vidéo de 2 minutes expliquant et montrant votre projet sur un support largement accessible de type youtube et utilisable par le DGEI et INTEL.

Soyez inventif !!

Annexe adresse IP des cartes :

Nom	IP	Nom	IP
Edison 1	10.105.1.23	Edison 15	10.105.1.44
Edison 2	10.105.1.24	Edison 16	10.105.1.45
Edison 3	10.105.1.25	Edison 17	10.105.1.46
Edison 4	10.105.1.26	Edison 18	10.105.1.47

Edison 5	10.105.1.27	Edison 19	10.105.1.48
Edison 6	10.105.1.28	Edison 20	10.105.1.49
Edison 7	10.105.1.29	Edison 21	10.105.1.50
Edison 8	10.105.1.30	Edison 22	10.105.1.51
Edison 9	10.105.1.31	Edison 23	10.105.1.52
Edison 10	10.105.1.32	Edison 24	10.105.1.53
Edison 11	10.105.1.33	Edison 25	10.105.1.54
Edison 12	10.105.1.34	Edison 26	10.105.1.55
Edison 13	10.105.1.35	Edison 27	10.105.1.56
Edison 14	10.105.1.36	Edison 28	10.105.1.57
		Edison 29	10.105.1.58
Module Wifi 1	10.105.1.60	Module Wifi 7	10.105.1.66
Module Wifi 2	10.105.1.61	Module Wifi 8	10.105.1.67
Module Wifi 3	10.105.1.62	Module Wifi 9	10.105.1.68
Module Wifi 4	10.105.1.63	Module Wifi 10	10.105.1.69
Module Wifi 5	10.105.1.64	Module Wifi 11	10.105.1.70
Module Wifi 6	10.105.1.65	Module Wifi 12	10.105.1.71
Module Wifi 7	10.105.1.66	Module Wifi 13	10.105.1.72
Module Wifi 8	10.105.1.67	Module Wifi 7	10.105.1.66