

E302C Gestion des périphériques

Travail : Gestion de périphériques avec la BeagleBone Black

Dans le cadre de l'activité *E302C Gestion des périphériques*, vous devez réaliser un **travail en trinôme**. Il s'agit d'un travail de conception et d'implémentation d'une librairie de gestion de périphériques pour la BeagleBone Black, codée en Python sur un Linux embarqué. Au minimum, votre librairie gèrera les pins GPIO, un premier périphérique simple tel qu'un capteur ultrasonique ou de proximité et un second périphérique plus complexe exploitant une communication I²C.

1 Contenu du travail

Le but du travail consiste à développer une librairie Python de contrôle de périphériques branchés sur la BeagleBone Black. Votre travail comportera plusieurs étapes :

1. Définir l'architecture de votre librairie, c'est-à-dire choisir comment vous allez représenter les éléments qu'il faudra contrôler (module avec fonctions, objets représentant les pins GPIO, etc.).
2. Implémenter la gestion des pins GPIO dans votre librairie et faire un programme d'exemple l'utilisant pour faire clignoter une LED.
3. Sur base du code gérant les pins, implémenter la gestion d'un périphérique simple qui se branche directement sur des pins GPIO (capteur ultrasonique, bouton, détecteur de proximité, etc.) et faire un programme d'exemple l'utilisant.
4. Implémenter la gestion du protocole I²C afin de contrôler l'un des périphériques proposés à la section 1.1 et faire un programme d'exemple l'utilisant.

Dans tous les cas, on n'attend pas de vous que vous gériez toutes les possibilités offertes par le périphérique que vous devez contrôler. Mieux vaut n'implémenter que le support d'une fonction, mais que ce dernier soit bien fait et démontrable. Vous ne pouvez évidemment pas utiliser une librairie déjà existante, le but du travail étant de créer sa propre librairie.

1.1 Périphériques I²C

Voici la liste des périphériques I²C parmi lesquels votre trinôme doit en choisir un :

- PCF8591 AD/DA converter
- ADXL345 Digital accelerometer
- BMP280 Barometric pressure, temperature and altitude sensor
- MPU6050 Accelerometer and gyroscope
- PN532 NFC/RFID controller
- SSD1306 OLED graphic display
- LCM1602 LCD display
- STM8S105C4 Line follower
- TCS34725 RGB color sensor
- CAP1188 8-key capacitive touch sensor breakout
- TCA9548A multiplexer
- MB85RC256V FRAM breakout
- VL6180X Time of flight distance ranging sensor
- TTP229-LSF 8-key touch pad detector IC

2 Échéance et livrable

Votre travail est à remettre pour le **vendredi 31 mars 2017 à 18h30** au plus tard, aucun retard ne sera toléré. Envoyez votre code, ou un lien vers votre dépôt GitHub, ainsi qu'une brève documentation de ce dernier (pouvant être dans le code source) et un fichier README expliquant comment reproduire les trois exemples que vous aurez implémentés, ainsi que toute autre information pertinente pour la compréhension de votre librairie.

3 Évaluation

Le projet sera évalué sur un total de **20 points** dont voici la répartition :

1. **La librairie** (5 points)
 - Le choix d'architecture est pertinent, adapté et rend la librairie pratique à utiliser (2 points)
 - La qualité du code est suffisante, pas de duplication (2 points)
 - La librairie est facilement extensible (1 point)
2. **Base** (4 points)
 - La manipulation des pins GPIO fonctionne correctement (1 point)
 - Le programme d'exemple de clignotement de la LED fonctionne correctement (1 point)
 - La fonction de gestion du périphérique simple fonctionne correctement (1 point)
 - Le programme d'exemple du périphérique simple fonctionne correctement (1 point)
3. **Avancé** (7 points)
 - Les fonctions d'utilisation du périphérique complexe sont correctes (3 points)
 - La compréhension du fonctionnement de la communication I²C est bonne (2 points)
 - L'ajout du support d'un nouveau périphérique I²C serait aisé (1 point)
 - Le programme d'exemple du périphérique complexe fonctionne correctement (1 point)
4. **Documentation** (4 points)
 - La librairie est suffisamment commentée, les fonctions sont spécifiées (2 points)
 - Le fichier README contient le descriptif des montages des exemples (1 point)
 - Le langage Python et ses possibilités sont bien utilisées (1 point)