

T.P. 5 : Programmation sous Contiki OS et déploiement d'un réseau

Rappel : Préparation de l'environnement

Si besoin : installation de l'environnement de développement Contiki OS / TelosB

- Étape 1 - Télécharger contiki-2.7.zip: <http://sourceforge.net/projects/contiki/files/>
 - extraire l'archive dans votre home : /home/student
 - installer avec « sudo apt-get install » les paquets suivants :
 - ant
 - gcc-msp430 (en cas de problème à la cross-compilation, installer également binutils-msp430 et msp430-libc)

Partie 1 : Cross-compilation et flashage des capteurs

L'objectif de cette première partie est de manipuler les outils de compilation.

- Exercice 1 – Compilation native de Contiki sur x86
 - aller dans le répertoire ~/contiki-2.7/examples/hello-world/
 - compiler avec la commande : make TARGET=native
 - Le programme affiche ce message sur l'écran et semble être bloqué. Contiki est bien en cours d'exécution mais n'a plus rien à faire. Pour terminer l'exécution, faire *ctrl-C*.
- Exercice 2 – Cross-compilation de Contiki sur TelosB - sky
 - aller dans le répertoire ~/contiki-2.7/examples/sky/ et cross-compiler les exemples avec la commande : make TARGET=sky
 - insérer un mote dans un port USB et le flasher avec la commande : sudo make blink.upload
 - observer le code source du programme blink (/platform/sky/apps).
 - vérifier que lors de chargement du code, une LED rouge proche du connecteur USB clignote bien : le programme blink s'exécute ensuite
 - compiler et charger le précédent programme hello-world sur le capteur TelosB : à quelle limitation des capteurs faites-vous face ?
 - il faut maintenant pouvoir récupérer l'impression « Hello world » : se connecter au mote inséré avec la commande : sudo make TARGET=sky login
 - appuyer sur le bouton « reset » pour voir l'impression
 - tester enfin l'exécution du programme test-button et observer son code source. Que fait-il ?
- Exercice 3 – Le shell Contiki
 - il est possible grâce à un shell interactif d'exécuter des commandes directement sur le capteur via la connexion USB
 - aller dans le répertoire ~/contiki-2.7/examples/sky-shell/
 - exécuter : sudo make sky-shell.upload ; puis s'y connecter avec : sudo make login
 - tester les commandes suivantes : help, sense | senseconv, power | powerconv, ps, blink 10, nodeid, repeat 2 2 { echo again } &, reboot

Partie 2 : Programmes communicants

L'objectif de cette seconde partie est de modifier des programmes communicants et de créer des mini-réseaux de capteurs.

- **Exercice 4** – Battle royale
 - aller dans le répertoire ~/contiki-2.7/examples/ipv6/simple-udp-rpl/
 - faire une copie puis modifier broadcast-example.c pour qu'il envoie votre prénom et qu'il affiche le message reçu
 - flasher le mote et s'y connecter avec « login » pour voir les messages circuler
- **Exercice 5** – Mesure et collecte de données physiques
 - aller dans le répertoire ~/contiki-2.7/examples/ipv6/rpl-collect/
 - lire attentivement le code des programmes udp-sender.c et udp-sink.c
 - noter l'adresse de votre mote au tableau et flasher un sender ; attendre qu'il soit détecté par le sink de la salle
 - en binôme, flasher un sink et un sender. Quelles sont les tailles des firmwares transférés ? Les comparer à la taille de la mémoire flash des capteurs utilisés (CM5000MSP).
 - sur le sink, vérifier la bonne collecte des données avec login et en essayant de les déchiffrer en vous aidant du code source
 - utiliser maintenant le programme « collect view » qui permet de visualiser les données reçues (commande « sudo ant run » dans ~/contiki-2.7/tools/collect-view/)
 - quel problème au niveau du routage RPL apparaît quand il y a plusieurs sink dans un réseau ?
 - commenter la ligne 1145 du fichier core/net/rpl/rpl-dag.c pour remédier au problème. Attention : tous les sinks doivent être redémarrés en même temps avec cette modification.

Partie 3 : COAP

L'objectif de cette dernière partie est d'interroger directement les motes depuis Internet grâce au protocole COAP.

- **Exercice 6** – REST / COAP tutorial
 - Installer le module Copper pour Firefox qui permet de gérer nativement le protocole COAP : <https://addons.mozilla.org/fr/firefox/addon/copper-270430/>
 - Suivre le tutorial REST/COAP à l'adresse suivante : http://anrg.usc.edu/contiki/index.php/REST_example_running_on_Cooja_and_Sky_motes

Partie 4 : Pour aller plus loin

S'il vous reste du temps, tester d'autres exemples de programmes ou de services s'exécutant sur les motes sky. Une bonne manière d'apprendre et de partir d'un exemple existant et de le modifier pour lui ajouter de nouvelles fonctionnalités de votre chef.

Pour approfondir d'autres aspects sur Contiki et son implantation des protocoles de l'IoT, des tutoriaux sont également disponibles à l'adresse suivante :

http://anrg.usc.edu/contiki/index.php/Contiki_tutorials