

Rapport TP Contiki /Cooja

On peut visualiser la liste des capteurs, obtenue avec la commande `sense | senseconv`

- light 1
- light 2
- temperature
- relative humidity
- rssi
- Voltage

Programing

Observations de la première simulation sur Cooja :
Lorsqu'on clique pour la première fois : toutes les leds s'allument.
Lors d'un second clique, les leds s'éteignent.

Dans le code, la partie correspondante correspond à :

- en fonction de la variable active :
- si not active, on appelle la fonction `SENSOR_ACTIVATE(light_sensor)`

- si active, c'est alors la fonction `SENSOR_DEACTIVATE` qui est appelée.

De plus, entre deux événements (2 clics) ,on peut voir que le processus est mis en attente, par l'appel de la fonction `PROCESS_WAIT_EVENT_UNTIL`

Quand l'événement est capté, il re-donne la main au processus, et le code correspondant peut alors continuer de s'exécute.

On peut agir sur `LEDS_RED` uniquement en utilisant `leds_toogle(LEDS_red)`, qui a pour effet de basculer d'un état à l'autre la led voulue.

Une autre manière de faire est d'utiliser les deux méthodes `leds_on` et `leds_off` (en fonction de la variable active), qui précisent dans quel état spécifique l'on souhaite mettre la LED.

Activation de la LED RED pendant 5 secondes (voir commentaires) :

```
static struct etimer et;  
SENSORS_ACTIVATE(light_sensor);  
etimer_set(&et, 5*CLOCK_SECOND); // Definition de la durée du timer de 5 secondes  
while(1) {  
  
    PROCESS_WAIT_EVENT_UNTIL(ev == sensors_event && data == &button_sensor);  
  
    leds_on(LED_RED);  
    etimer_reset(&et); // Mise à zéro du timer une fois que la Led est allumée  
    PROCESS_WAIT_EVENT_UNTIL(etimer_expired(&et)); //Attente de la fin du timer  
    leds_off(LED_RED);  
  
    ...  
}
```

5 Clignotement avec intervalles 1secondes éteinte/ Allumée :

On peut effectuer 5 clignotement grace à une boucle for, qui réinitialise un timer d'une seconde entre chaque basculement de cette manière :

```
static struct etimer et;  
etimer_set(&et, CLOCK_SECOND);  
... // Déclenchement de l'évenement de clique  
for(i=0; i<5; i++){  
    leds_on(LED_RED);  
    etimer_reset(&et);  
    PROCESS_WAIT_EVENT_UNTIL(etimer_expired(&et));  
    leds_off(LED_RED);  
    etimer_reset(&et);  
    PROCESS_WAIT_EVENT_UNTIL(etimer_expired(&et));  
}  
...
```

Communication entre capteurs

On peut voir que la valeur qui définit la puissance de radio se situe dans l'appel à la fonction `cc2420_set_txpower(txpower);`

La variable `txpower` est défini comme ça:
`txpower = CC2420_TXPOWER_MAX;`

Pour réduire cette valeur, on peut changer la valeur de `CC2420_TXPOWER_MAX`, qui est défini dans le fichier situé dans `/dev/core/cc2420.h`
Par défaut, elle vaut 127.

On remarque également que quand 2 noeuds sont trop éloignés, la réception ne peut pas se faire (led Verte qui s'arrête de clignoter)