

TP2 – Programmation avec Contiki Manipulation d'actionneur et temporisateur (timer)

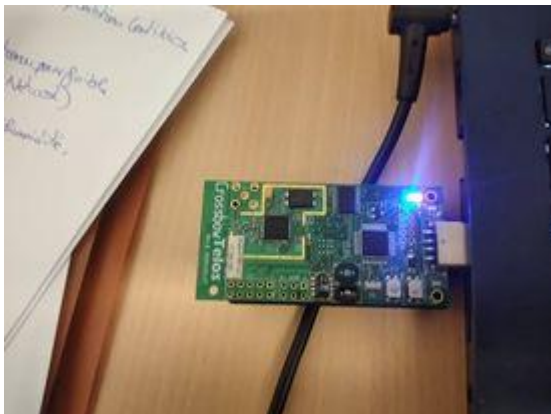
Partie 1 :

Développer une première application avec Contiki

#Créer un fichier programme nommé « hello-world.c »

```
#include "contiki.h"
PROCESS(hello_world_process, "Hello world process");
AUTOSTART_PROCESSES(&hello_world_process);
PROCESS_THREAD(hello_world_process, ev, data)
{
    PROCESS_BEGIN();
    printf("Hello, world!\n");
    PROCESS_END();
}
```

#Insérer le capteur Tmote Sky (TelosB) sur le port USB du PC



Exercice 1 : Commander la couleur des leds via le bouton user

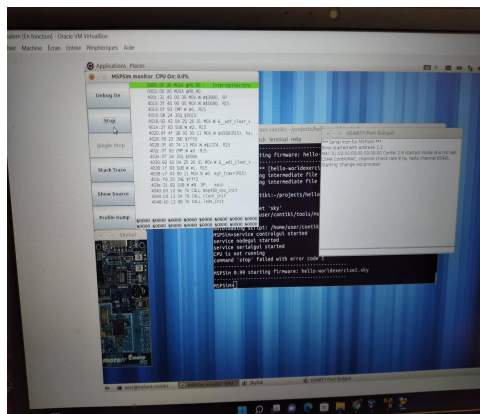
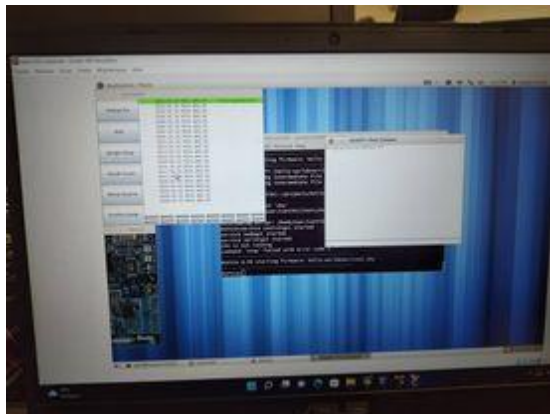
```
#include "stdbool.h"
#include "leds.h"
#include "button-sensor.h"
#include "contiki.h"
PROCESS(hello_world_process, "change led process");
AUTOSTART_PROCESSES(&hello_world_process);
PROCESS_THREAD(hello_world_process, ev, data)
{
    PROCESS_BEGIN();
    SENSORS_ACTIVATE(button_sensor);

    static bool b = true;
```

```

    leds_off(LEDS_RED);
    leds_off(LEDS_GREEN);
    leds_off(LEDS_BLUE);
    while(1){
        PROCESS_WAIT_EVENT();
        if (ev == sensors_event && data == &button_sensor) {
            printf("Button pressed!");
            if(b){
                printf("VERT! \n");
                leds_off(LEDS_RED);
                leds_on(LEDS_GREEN);
                b = !b;
            }else{
                printf("Rouge! \n");
                leds_on(LEDS_RED);
                leds_off(LEDS_GREEN);
                b = !b;
            }
        }
    }
    printf("end of programm");
    PROCESS_END();
}

```



Exercise 2 :

```

#include "contiki.h"
#include <stdio.h>
#include <button-sensor.h>
#include <leds.h>

PROCESS(count_process, "Count process");
AUTOSTART_PROCESSES(&count_process);
PROCESS_THREAD(count_process, ev, data)
{

```

```

PROCESS_BEGIN();
SENSORS_ACTIVATE(button_sensor);
printf("Counting. \n");
#Déclarer un Timer et une variable de type entier « int » pour le compteur :
static struct etimer et;
static int i;
#Initialiser le Timer
etimer_set(&et,CLOCK_SECOND) ;
#Déclarer la section event lorsque le Timer s'expire :
while(1){
    PROCESS_WAIT_EVENT();
    if (etimer_expired(&et)){
        printf("Compteur : %d !\n",i);
        if(i&1){
#led allumé
            leds_on(LEDS_RED);
        }
#led éteinte
        else{
            leds_off(LEDS_RED);
        }
#led allumé

        if(i&2){
            leds_on(LEDS_GREEN);
        }
#led éteinte
        else{
            leds_off(LEDS_GREEN);
        }
#led allumé
        if(i&3){
            leds_on(LEDS_BLUE);
        }
#led éteinte
        else{
            leds_off(LEDS_BLUE);
        }
#Incrémenter le compteur
        i += 1;

    }
#Réinitialiser le Timer
    etimer_set(&et,CLOCK_SECOND);
#Ajouter une fonction d'initialisation du compteur lorsque le bouton du capteur est pressé
    if (ev == sensors_event && data == &button_sensor) {
        i=0; }
}

```

```

PROCESS_END();
}

```

