CS4222/CS5422

PROGRAMMING ASSIGNMENT #2

AY 2020-2021 / SEM 2

DUE: Feb 22 (Monday) 23:59

1. **This is an individual assignment.**
2. **Total Marks: 50.**
3. **This assignment carries 5% weightage to your final grade.**
4. **There is a 10% penalty per-day for late submission.**
5. **For any clarification on this assignment, post your queries through email the TAs.**

1. OVERVIEW

In this assignment, you will learn how to use the timers to schedule events. The overall task is to build a "light-controller buzzer" using the timers, sensors and buzzer.

2. Timer

Executions in Contiki is event-driven and use of timers are common. The Contiki system provides a set of timer libraries. In this assignment, you will look at using the *etimer* and *rtimer*.

A timer uses the data structure `struct timer`. Three function calls are used to control a timer.

- `timer_set()` is used to initialize and starts the expiration time.
- `timer_reset()` is used to restart the timer from previous expire time.
- `timer_restart()` is used to restart the timer from current time.

One difference between etimer and rtimer is the time resolution. etimer's clock resolution depends on the number of clock ticks per second (CLOCK_SECOND), while rtimer uses RTIMER_SECOND.

Another difference is that programming style. Etimer uses a more "sequential" model while rtimer uses callback.

Sample code for etimer

```
PROCESS_THREAD(example_process, ev, data)
{
   ….
   PROCESS_BEGIN();

   etimer_set(&timer_etimer, CLOCK_SECOND);  /* Delay 1 second */

   while(1) {
     PROCESS_WAIT_EVENT_UNTIL(etimer_expired(&timer_etimer));
     etimer_reset(&timer_etimer);
```

```
   }
   PROCESS_END();
}
```

Sample code for rtimer

```
PROCESS_THREAD(process_rtimer, ev, data)
{
  PROCESS_BEGIN();
  init_opt_reading();

  while(1) {
    rtimer_set(&timer_rtimer, RTIMER_NOW() + RTIMER_SECOND, 0,
               do_rtimer_timeout, NULL);
    PROCESS_YIELD();
  }

  PROCESS_END();
}
```

Take a look at the documentation at https://github.com/contiki-ng/contiki-ng/wiki/Documentation:-Timers for more information.

3. Programs Given

You will be given two programs (available on luminus workbin).

- etimer-buzzer.c: a sample program that shows how to use the etimer and the buzzer.
- rtimer-lightSensor.c: a sample program that shows how to read from the light sensor.

Make sure you can run both the C program and observe the output generated by the printf statements. Your Makefile should include names of the new programs.

```
CONTIKI_PROJECT = etimer-buzzer rtimer-lightSensor
```

4. Tasks

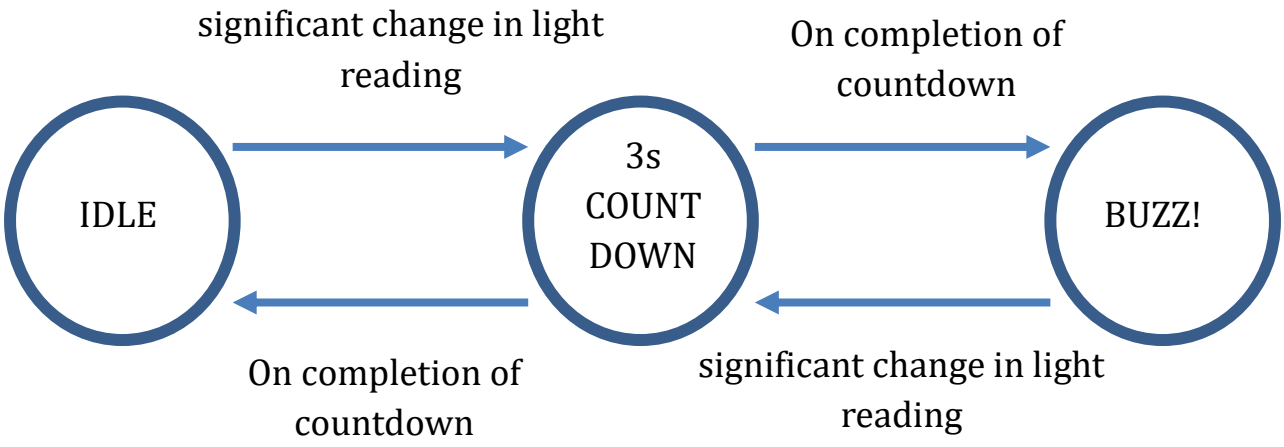
4.1 Clock Resolution

From the output of etimer-buzzer.c, note down the value of CLOCK_SECOND. Find out how many clock ticks corresponds to 1s in real time.

From the output of rtimer-lightSensor.c. note down the value of RTIMER_SECOND. Find out how many clock ticks corresponds to 1s in real time.

4.2 Code Implementation

Write the code to implement the state transition diagram given below.



- Significant change is defined by a change of more than **300** lux in the light intensity.
- Do not sample the light sensor at a rate higher than **4 Hertz** as the driver does not work well at higher sampling rate.
- You can use the buzzer code in the etimer-buzzer.c file to implement the BUZZ! state.

5. Submission

Please submit a single zip file ("**CS4222-Assignment2 - YourStudentNumber.zip**") to folder "Assignment2 Submission" on luminus by the due date. If you submit multiple times, only the latest submission will be evaluated.

Your submission should include the following:

1. (10%) For following instructions and naming convention given.
2. (30%) A text file (**readme.txt**) that contains the following:
    - Value of CLOCK_SECOND
    - Number of clock ticks per second in 1s (real time) using etimer.
    - Value of RTIMER_SECOND
    - Number of clock ticks per second in 1s (real time) using rtimer.
    - Any instruction on how to run your program (buzz.c) if needed.
3. (60%) A program called **buzz.c** that implements the state transition diagram given in Section 4.2. Grading will take into account the robustness and ease of use of the **light-controlled buzzer.**