Embedded Hardware and Operating Systems: Practice Peer-graded Assignment TinyOS -- Hands-on

Objective: To program a TinyOS application in practice

Materials: For the current assignment, the TinyOS is installed on a Contiki (3.0) system. This system is run using the latest VMWare player.

Report: Functionality of the Blink Application

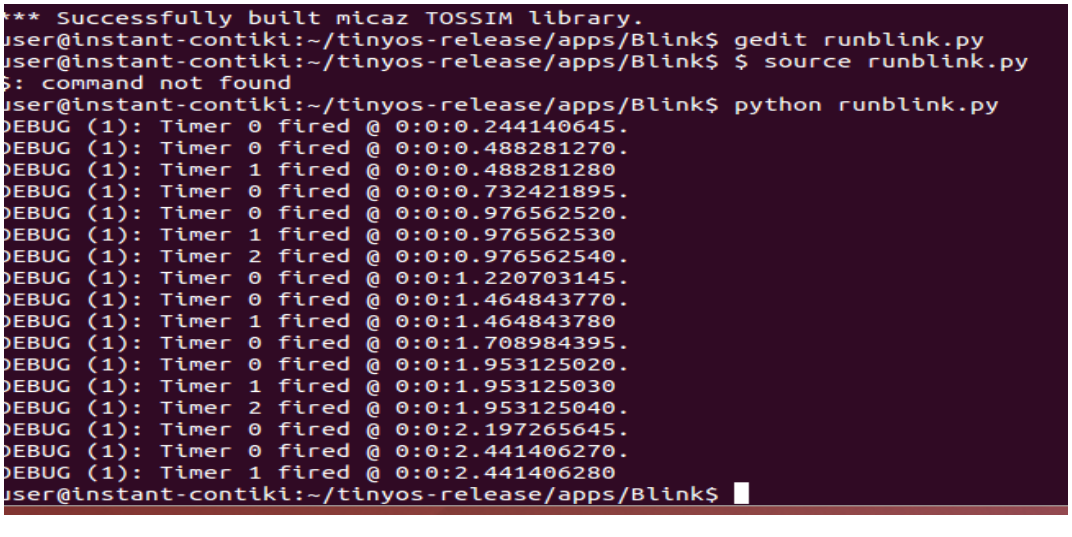
Introduction

This report provides an overview of the functionality of the Blink application, explains how it works, describes the priorities set in the example, and outlines the modifications made to the code.

Functionality of the Blink Application

The Blink application is a simple example designed to demonstrate basic functionality on TinyOS-compatible hardware. It blinks an LED on and off at a predefined interval. The application serves as a starting point for learning about programming embedded systems using TinyOS.

A. Task 1 In this task, simulated timers are used to blink LEDs in the TinyOS program. A “runblink.py” porgam is created in Python language. The program is compiled using the TinyOS compiler and the compiled environment is then run in the TOSSIM simulator. The output from the program is as shown in figure 1



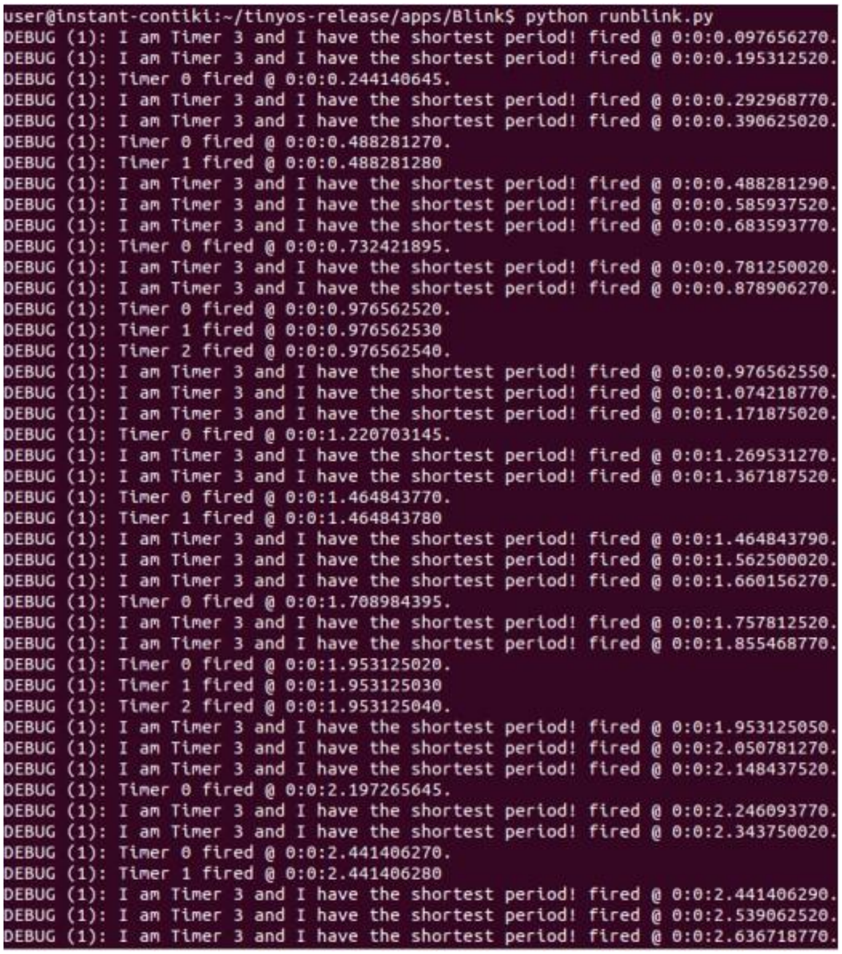
```
*** Successfully built micaz TOSSIM library.
user@instant-contiki:~/tinyos-release/apps/Blink$ gedit runblink.py
user@instant-contiki:~/tinyos-release/apps/Blink$ $ source runblink.py
$: command not found
user@instant-contiki:~/tinyos-release/apps/Blink$ python runblink.py
DEBUG (1): Timer 0 fired @ 0:0:0.244140645.
DEBUG (1): Timer 0 fired @ 0:0:0.488281270.
DEBUG (1): Timer 1 fired @ 0:0:0.488281280
DEBUG (1): Timer 0 fired @ 0:0:0.732421895.
DEBUG (1): Timer 0 fired @ 0:0:0.976562520.
DEBUG (1): Timer 1 fired @ 0:0:0.976562530
DEBUG (1): Timer 2 fired @ 0:0:0.976562540.
DEBUG (1): Timer 0 fired @ 0:0:1.220703145.
DEBUG (1): Timer 0 fired @ 0:0:1.464843770.
DEBUG (1): Timer 1 fired @ 0:0:1.464843780
DEBUG (1): Timer 0 fired @ 0:0:1.708984395.
DEBUG (1): Timer 0 fired @ 0:0:1.953125020.
DEBUG (1): Timer 1 fired @ 0:0:1.953125030
DEBUG (1): Timer 2 fired @ 0:0:1.953125040.
DEBUG (1): Timer 0 fired @ 0:0:2.197265645.
DEBUG (1): Timer 0 fired @ 0:0:2.441406270.
DEBUG (1): Timer 1 fired @ 0:0:2.441406280
user@instant-contiki:~/tinyos-release/apps/Blink$ 
```

B. Task 2:

- In Task 2, a timer called “Timer 3” is added to the Blink Application using the following line:

  components new TimerMilliC() as Timer3;

  - This timer is scheduled for every 100ms in the BlinkC.nc script using the following line: **call Timer3.startPeriodic(100);**
  - This timer is programmed to print out “I am Timer 3 and I have the shortest period!” using the following lines of code: **event void Timer3.fired() { dbg("BlinkC"**, **"I am Timer 3 and I have the shortest period! fired @ %s.\n", sim_time_string()); }**
  - Finally, the python script is edited to simulate for 2000 events instead of 100 just by replacing the input to the loop **: for i in range(0, 2000): t.runNextEvent()**



```
user@instant-contiki:~/tinyos-release/apps/Blink$ python runblink.py
DEBUG (1): I am Timer 3 and I have the shortest period! fired @ 0:0:0.097656270.
DEBUG (1): I am Timer 3 and I have the shortest period! fired @ 0:0:0.195312520.
DEBUG (1): Timer 0 fired @ 0:0:0.244140645.
DEBUG (1): I am Timer 3 and I have the shortest period! fired @ 0:0:0.292968770.
DEBUG (1): I am Timer 3 and I have the shortest period! fired @ 0:0:0.390625020.
DEBUG (1): Timer 0 fired @ 0:0:0.488281270.
DEBUG (1): Timer 1 fired @ 0:0:0.488281280
DEBUG (1): I am Timer 3 and I have the shortest period! fired @ 0:0:0.488281290.
DEBUG (1): I am Timer 3 and I have the shortest period! fired @ 0:0:0.585937520.
DEBUG (1): I am Timer 3 and I have the shortest period! fired @ 0:0:0.683593770.
DEBUG (1): Timer 0 fired @ 0:0:0.732421895.
DEBUG (1): I am Timer 3 and I have the shortest period! fired @ 0:0:0.781250020.
DEBUG (1): I am Timer 3 and I have the shortest period! fired @ 0:0:0.878906270.
DEBUG (1): Timer 0 fired @ 0:0:0.976562520.
DEBUG (1): Timer 1 fired @ 0:0:0.976562530
DEBUG (1): Timer 2 fired @ 0:0:0.976562540.
DEBUG (1): I am Timer 3 and I have the shortest period! fired @ 0:0:0.976562550.
DEBUG (1): I am Timer 3 and I have the shortest period! fired @ 0:0:1.074218770.
DEBUG (1): I am Timer 3 and I have the shortest period! fired @ 0:0:1.171875020.
DEBUG (1): Timer 0 fired @ 0:0:1.220703145.
DEBUG (1): I am Timer 3 and I have the shortest period! fired @ 0:0:1.269531270.
DEBUG (1): I am Timer 3 and I have the shortest period! fired @ 0:0:1.367187520.
DEBUG (1): Timer 0 fired @ 0:0:1.464843770.
DEBUG (1): Timer 1 fired @ 0:0:1.464843780
DEBUG (1): I am Timer 3 and I have the shortest period! fired @ 0:0:1.464843790.
DEBUG (1): I am Timer 3 and I have the shortest period! fired @ 0:0:1.562500020.
DEBUG (1): I am Timer 3 and I have the shortest period! fired @ 0:0:1.660156270.
DEBUG (1): Timer 0 fired @ 0:0:1.708984395.
DEBUG (1): I am Timer 3 and I have the shortest period! fired @ 0:0:1.757812520.
DEBUG (1): I am Timer 3 and I have the shortest period! fired @ 0:0:1.855468770.
DEBUG (1): Timer 0 fired @ 0:0:1.953125020.
DEBUG (1): Timer 1 fired @ 0:0:1.953125030
DEBUG (1): Timer 2 fired @ 0:0:1.953125040.
DEBUG (1): I am Timer 3 and I have the shortest period! fired @ 0:0:1.953125050.
DEBUG (1): I am Timer 3 and I have the shortest period! fired @ 0:0:2.050781270.
DEBUG (1): I am Timer 3 and I have the shortest period! fired @ 0:0:2.148437520.
DEBUG (1): Timer 0 fired @ 0:0:2.197265645.
DEBUG (1): I am Timer 3 and I have the shortest period! fired @ 0:0:2.246093770.
DEBUG (1): I am Timer 3 and I have the shortest period! fired @ 0:0:2.343750020.
DEBUG (1): Timer 0 fired @ 0:0:2.441406270.
DEBUG (1): Timer 1 fired @ 0:0:2.441406280
DEBUG (1): I am Timer 3 and I have the shortest period! fired @ 0:0:2.441406290.
DEBUG (1): I am Timer 3 and I have the shortest period! fired @ 0:0:2.539062520.
DEBUG (1): I am Timer 3 and I have the shortest period! fired @ 0:0:2.636718770.
```

In this assignment, timers were programmed in the TinyOS LED blinking application. As can be seen from the results of Task 1 and Task 2, the application was compiled and simulated successfully using the TOSSIM library via a python script.