Uppsala University

Information Technology

Wireless Communication and Built-in Systems

Contiki Lab - An Intrusion Detection System

Z1 User

---

*Student:*_____

*Email:*_____

*Date:*_____

January 25, 2021

Introduction

The goal of the assignment is to write your first program in Contiki-ng. You will start with an existing program: The clicker application that was introduced in the lecture. The clicker application counts clicks from wireless sensor node and visualizes the number of clicks on the base station LEDs. Your goal is to change the clicker application to a simple intrusion detection application. In this application, a node should send a packet to a base station whenever it is shaken. You will have one Zolertia Z1 nodes as client and one NRF52840dk board as base station to test your code.

Tips

Directory and folder is marked with *italic font*.
Linux terminal code is marked with `typewriter font`.
C code is marked with `green typerwriter font`.

Make yourself ready for the lab

Setup and test the development environment

You will be using the virtual machine to do this lab. The image has been provided to you in the lecture: Hello World! If you have any questions, please go back the Hello World! lecture and look for supporting information.

1. Open VM image with VirtualBox and login with the password: "user".
2. Connect the Z1 board and nRF51840 DK board to your computer. From the virtual box menu devices, select the two devices. For the USB devices to be recognized in the virtual box, you might need to add guest-additions.
3. To ensure the USB devices are recognized by the machine, open the terminal and execute `dmesg` or `lsusb` commands.

Preparing the source code

1. Download the example code for lab2 from studium Files in the folder *Labs*, file name is *lab2.tar.gz*.
2. Unzip the file with comamnd:
   `tar xvzf lab2.tar.gz`
3. Go the unziped folder *lab2*. Test the example code on both boards:
   Z1: `make TARGET=z1 clicker.upload`
   nRF52840 DK: `make TARGET=nrf52840 basestation.upload`
4. Remember to make a copy of the source code before you modify it.

Task 1

You are to write an intrusion detection program. There should be two types of nodes: A base station (nRF52840 DK) that will visualize alarms using its LEDs, and client nodes (Z1) that will send an alarm packet whenever they are being shaken. The client nodes should use accelerometer data to detect movement. A client node should send alarms when it is shaken and the base station node should visualize using its LEDs that an alarm is received. If the base station does not receive an alarm for some time, it should turn off its LEDs again. Here are some hints:

1. An example program showing how to use the Z1's accelerometer with Contiki-ng is available in *lab2* example folder. Check out the file in *lab2* directory *lab2/test-adxl345.c*. (Notice that the example uses some of the accelerometer's features that you do not need for this task.) First flash the example code into Z1 mote `make TARGET=z1 test-adxl345.upload` then use `make TARGET=z1 login` to check the accelerometer reading via serial port log.
2. It is enough to check the acceleration along one axis. A sampling rate of 100Hz, i.e., reading the accelerometer 100 times a second, is also sufficient for the simple intrusion detection system.
3. If you have multiple nodes attached and want to upload a program only to one of the nodes, specifiy the `TARGET=nrf51840 or TARGET=z1`.

4. Notice that you can use functions such as `PROCESS_WAIT_EVENT()` only within a process, and not within other functions.

5. To achieve **Turn off LEDs when base station does not receive an alarm for some time**, try to write another `PROCESS_THREAD` and name it `led_process`. After the last alarm 10 seconds if there is not any new incoming alarm, turn off all LEDs.

6. The function `process_poll(&process_name);` allows you to request the process called *proccess_name* to be run next. You can use this function, e.g., to trigger a process to be run after you have received a packet. In the corresponding process you might use process yield or wait functions such as `PROCESS_YIELD_UNTIL` or `PROCESS_WAIT_EVENT_UNTIL`:

```
PROCESS_THREAD(led_process, ev, data) {
PROCESS_BEGIN();
PROCESS_YIELD_UNTIL(ev == PROCESS_EVENT_POLL);
PROCESS_END();
}
```

Task 2

Now you should extend the intrusion detection system built in Task 1 to handle more than one sensor. Now we will only raise the alarm if BOTH accelerometer and button detect an event within a short time period, such as 10s.

Using the LEDs, the base station should now visualize which sensor is triggered on client node. Use one LED for each of the sensor, e.g. LED1 indicates accelerometer and LED2 represents button (as in task 1 the LEDs should turn of if not alarm events for a short time period, such as 10s). Furthermore, if report both events the third LED should light up, indicating an alarm.

The demonstration first step is shaking client node and pressing button independently, then the second step is to shake client node and then press the button immediately to show that the base station turn on and off LEDs accordingly.