CS 501 – Internet of Things

Assignment 1

I. Introduction

In the first assignment, we are going to prepare our working environment to be able to run, develop, modify and debug software for wireless sensors. During this assignment you are going to compile and run our first program on sensor hardware. As tradition demands we are going to run a simple Hello World! program on a sensor node that outputs “Hello World”. In this Session we have the following goals:

- See and understand Contiki C-programming code,
- Compile the code,
- Deploy code on a sensor node.

In fact, Contiki provides a fully configured development environment that runs as a virtual machine. In this session we install and configure the free VMware Player, run the Contiki Image on it and connect sensor nodes to the Contiki environment to be able to deploy programs.

To run Contiki on your computer, you need to do the following:

1. Go to Applications - Accessories - Terminal
2. On terminal window, type "vmplayer"; Wait for Virtual Machine window to open up
3. Select "Remind Me Later" on additional pop-up window
4. Double click "Open a Virtual Machine" on main window
5. Select "pencil like" icon on top left of window to open up the "Location Box", into which, type in "/usr/share/InstantContiki2.6"
6. Double click on file with extension ".vmx"
7. Click "Play Virtual Machine" at the bottom of main window
8. Select "I copied it" on pop-up window and wait for few seconds
9. On next pop-up window, select "Remind Me Later"
10. On the login screen, type username/password: contiki/user.

II. Prerequisites

- VMware Workstation Player:
  Get it from https://www.vmware.com/products/workstation-player.html

- InstantContiki 2.6 (not 2.6.1)

  Get it from https://sourceforge.net/projects/contiki/files/Instant%20Contiki/Instant%20Contiki%202.6/

III. Installation

If you do not have Contiki installed, you may follow the instruction before to install Contiki. You can download the software following the above link. First install the VMware Workstation Player according to its installation instructions.

Extract the Instant Contiki Files to your hard drive and Open the vmx File in the VMware Player.

If you have problems installing the player and the software have a look at this tutorial explaining the installation process step-by-step (http://www.contiki-os.org).

IV. Hello World! Code Example

```
1  #include "contiki.h"
2
3  #include <stdio.h> /* For printf() */
4  /*---------------------------------------------------------*/
5  PROCESS(hello_world_process, "Hello world process");
6  AUTOSTART_PROCESSES(&hello_world_process);
7  /*---------------------------------------------------------*/
8  PROCESS THREAD(hello_world_process, ev, data)
9  {
10   PROCESS BEGIN();
11
12   printf("Hello, world\n");
13   PROCESS_END();
14
15 }
16 /*---------------------------------------------------------*/
```

In the above listing we see our first program. To understand the code, we go through it line by line.

In L.1 we include the Contiki header files that include the Contiki OS into the compiled program and allow to access the scheduling and abstraction APIs of Contiki.

Contiki follows Protothreads (see also https://en.wikipedia.org/wiki/Protothreads), a concurrent programming model with a low-overhead.

L.3 includes the standard input/output library needed to write to the standard output.

L.5 defines the processes to be executed and included during runtime. In this case we include only one process: the *Hello World process*. It is possible and usually the case to define several concurrent processes, for example one for processing and collecting data, one for transmitting or receiving data.

L.6 tells the operating system which process to start on startup of the sensor node. L.8-L.10 are the head of a function and L.14 defines the end of the process.

L.12 is where the magic happens and the String gets to the console via the *printf* command. Most of the contiki commands rely on standard-C.

V. The Make Files

In order to run the code on the sensor node we need make files telling the compiler how to compile the programs for which specific platform. In this case we need two make files, one for telling the compiler where the contiki headers can be found, and one defining the platform the code has to be compiled for.

The following is the make file: “Makefile”

```
1 CONTIKI_PROJECT = hello-world
2 all: $(CONTIKI_PROJECT)
3
4 CONTIKI = ../..
5 include $(CONTIKI)/Makefile.include
```

The following is the make target: “Makefile.target” (optional)

```
1 TARGET=native
```

Note that if you do not have “Makefile.target”, you may provide the target in the command line during compilation.

VI. Exercise

You may create a directory “A1” under “/home/user/contiki-2.6/examples/” to put your source code. Then, change your working directory to “/home/user/contiki-2.6/examples/A1” and copy the following files into the directory:

- hello-world.c
- Makefile
- Makefile.target (optional)

Then use the following command to compile and run the code.

```
$ make hello-world
$ make hello-world.upload
$ make login
$ ./hello-world.native
```

Run the commands from above to (1) compile the program, (2) upload the program to the sensor node connected, (3) login into the console of the connected sensor node, and (4) to run your program.

After the successful login to the node, restart it by pressing the red reset button on the mote and see the *Hello World* program starting.

Note that if your working directory does not have “Makefile.target”, you may use the following commands:

```
$ make TARGET=native hello-world
$ make TARGET=native hello-world.upload
$ make TARGET=native login
$ ./hello-world.native
```

In addition, you have to extend the code to generate ten random numbers between 0 and 1, display them, and compute their sum. For that purpose, you have to include the file

#include <random.h> /* For random_rand() */

VII. Submission

You have to submit the file HelloWorld.c, and the pdf file that contains the different stages to run your program and the screenshots of its execution.

Good luck :)