Internet of Things (IOT) – lab 2

Objective:

(1) Demonstrate the LED blinking at Arduino UNO
(2) Establish communication from Arduino Due in UDOO (a coordinator hub) to Arduino UNO

Arduino Due in UDOO

The information provided from page 1 to page 3 is mostly the same as stated in lab 1. This may save you the effort of referring to lab 1.

Insert the SD card into UDOO (Figure 2). Download from the Internet Arduino IDE 1.5.4, Arduino IDE patch files and serial driver (Figure 1). Install the downloaded software. Copy the patch files (bossac and cygwin1.dll) into …\hardware\tools directory of Arduino IDE, for example c:\Program File(x86)\Arduino\hardware\tools .

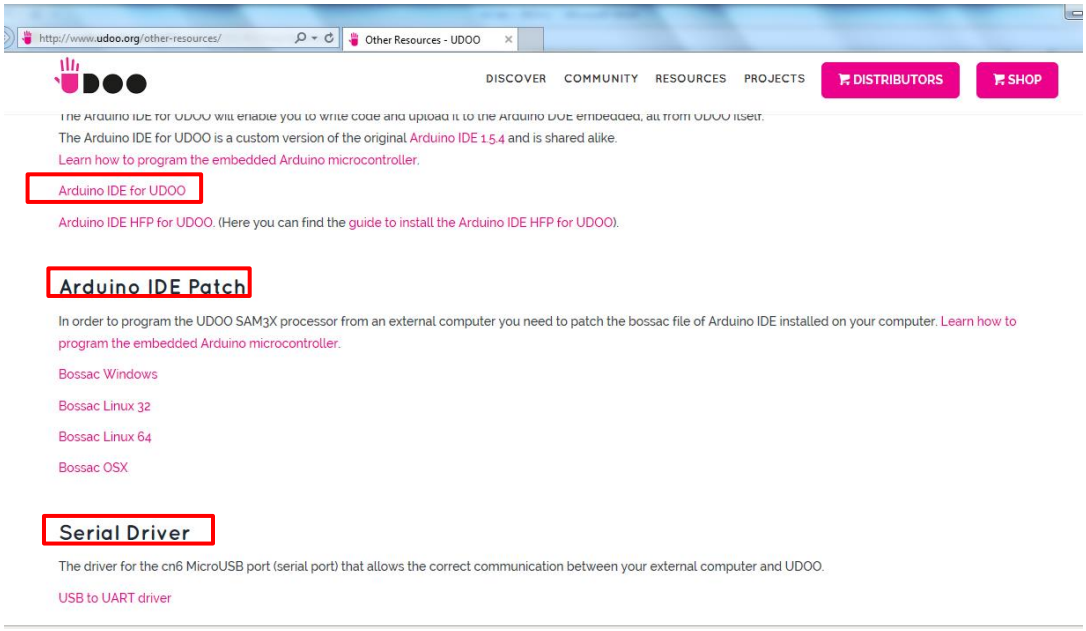

Figure 1: Arduino IDE, patch files and serial driver

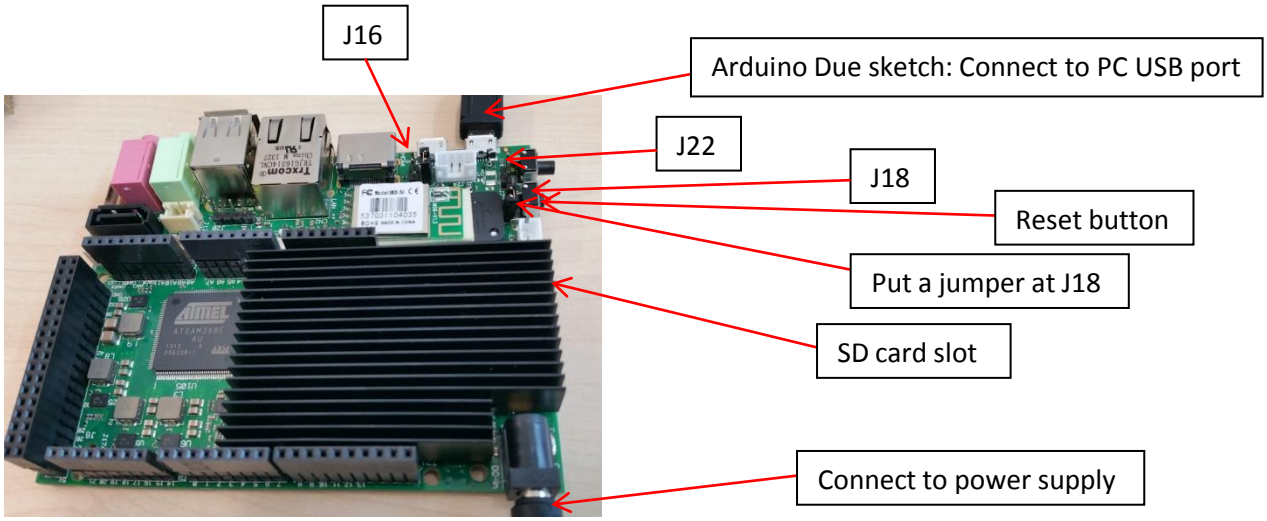

Figure 2: UDOO

Download a serial monitor software (Tera Term for Windows, serial tools for OS X, minicom for LINUX) from the Internet and installed the software. Connect the UDOO board to the PC USB port and power supply (Figure 2). Put a jumper at J18 as shown in Figure 2. As an example, start the Tera Term software. Select Serial and a relevant COM port (for example COM18). At Tera Term, select Setup -> serial port … -> baud rate 115200. Press the reset button at J18. Messages like Figure 3 will appear.

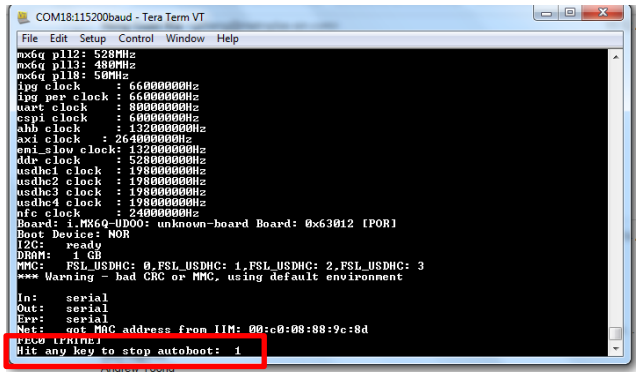

Figure 3: To stop autoboot

When the message, “Hit any key to stop autoboot:”, quickly hit any key from keyboard like b key or z key to stop the kernel boot.

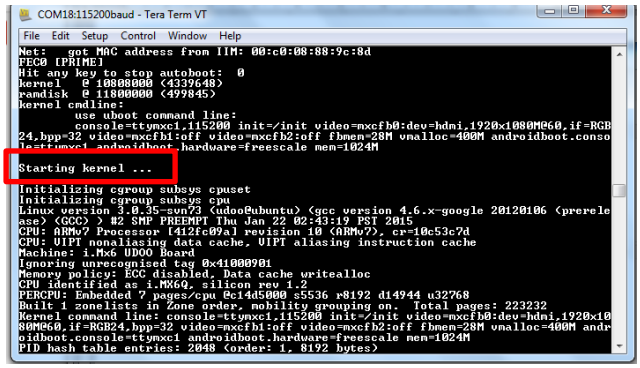

Figure 4: Starting kernel

If user is too slow, the kernel will start to boot (red rectangular box in Figure 4). Then the user needs to redo again. A successful stop will see output like Figure 5 below. Notice the prompt in the red rectangular box.

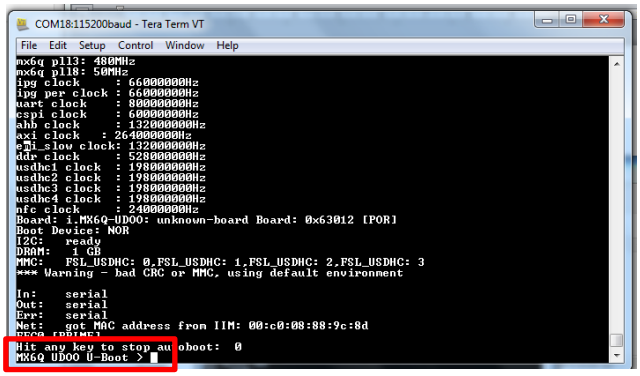

Figure 5: Kernel boot stopped

Close the serial monitor, for example Tera Term. Remove J18 jumper. Put the jumper J22 for one second and remove it. This is to erase the previous sketch. Do the same for jumper J16. This is to reset the SAM3X8E processor. This processor is used for Arduino Due sketch programming.

At Arduino IDE, select Tools -> Board -> Arduino Due (Programming Port). At Arduino IDE, select Tools -> Port -> select relevant port (for example COM18). At Arduino IDE, compile and upload the lab1_Due file which contains the arduino sketch. If successful, messages like in Figure 6 will be displayed.

Figure 6: Successully compile and upload sketch

Connect the XBee shield on top of the UDOO (Figure 7). Notice that there is a XBee module. A red light is seen when there is a communication with another XBee module.

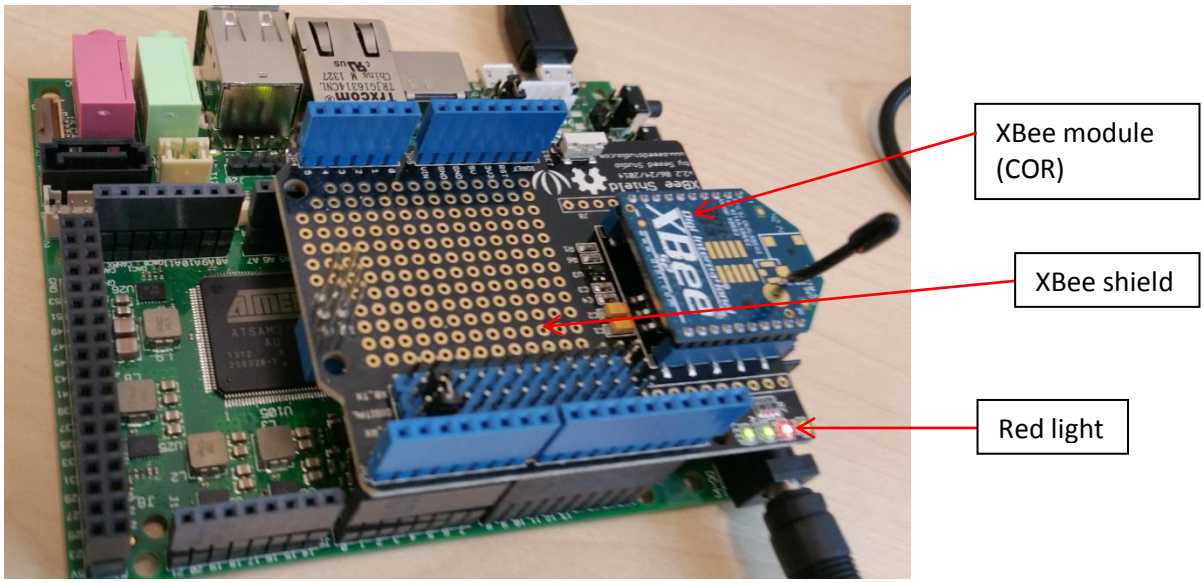

Figure 7: XBee shield on top of UDOO

Arduino UNO configuration

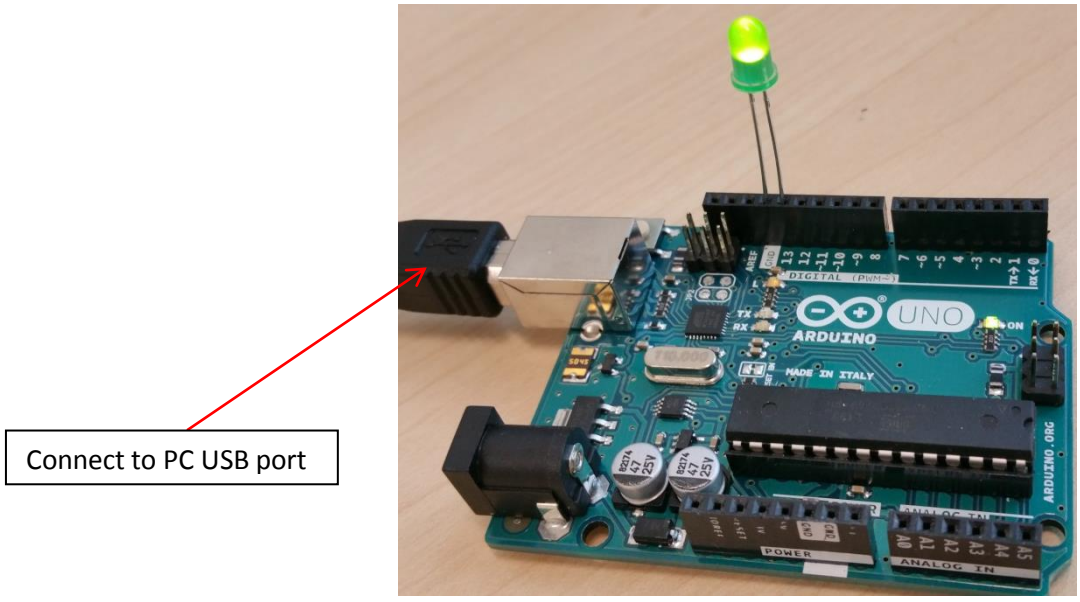

Figure 8: Arduino UNO

Connect the Arduino UNO circuit board to the USB port of a PC (Figure 8). This is to provide power supply and communication with the PC. Connect to power bank only provides power supply to Aduino UNO. Insert a LED (Figure 8) at Grd (shorter lead) and pin 13 (longer lead).

At Arduino IDE, select Tools -> Board -> Arduino Uno. At Arduino IDE, select Tools -> Port -> select relevant port (for example COM13). At Arduino IDE, compile and upload your file. The LED in Figure 8 should blink every 100 milliseconds.

Now, at Arduino IDE, compile and upload another file, lab1__1UNO file.

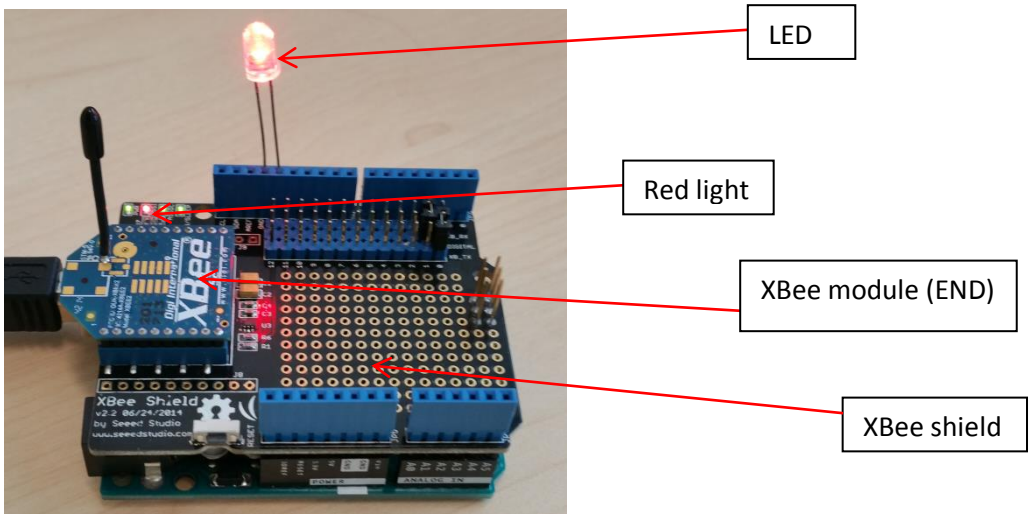

Figure 9: XBee shield above Arduino UNO

Connect the XBee shield on top of the Arduino UNO as shown in Figure 9. Notice that there is a XBee module. Insert a LED (Figure 9) at Grd (shorter lead) and pin 13 (longer lead).

A red light is seen when there is a communication with another XBee module.

The UDOO and Arduino UNO communication set-up should look like Figure 10.

A successful connection with the Arduino Due in UDOO will show messages like HLHLHLH … (Figure 11) displayed in the serial monitor screen and the LED light will blink every 10 milliseconds.

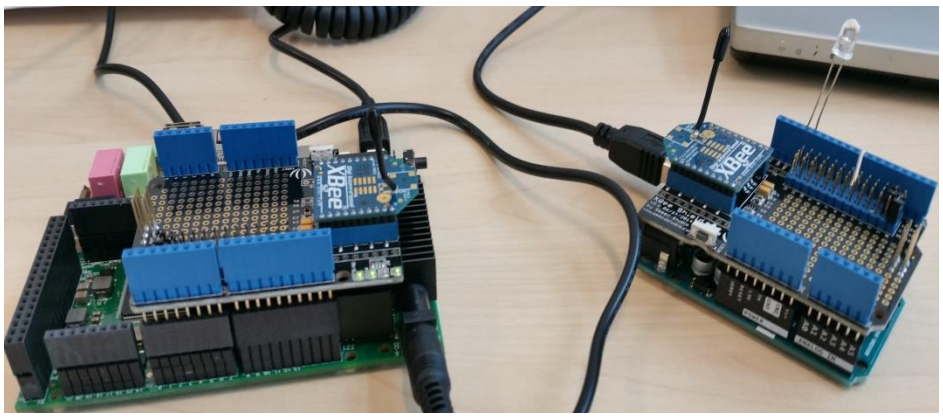

Figure 10: UDOO and Arduino UNO communication setup

Figure 11: Messages in serial monitor at Arduino UNO

After you have finished, get a check off from one of the instructors or TA's by system demonstration.