

Xiaobin Zhuang - 300519184

ECEN301 Lab 8

05/10/2020

## Objectives

The purpose of this lab session is to explore how to set up a debugging environment running on the Linux system.

## Methodology

We set up develop environment by followed the steps in the lab manual. There are few things need to mention in this lab report.

In step 1, when we set Cross compiler path, the file name was not CCS901 as shown in the manual, it was CCS910 in my computer. This was same when we were doing step 5 and 12.

In step 8, we accessed the IP address: 192.168.7.2. It showed the interface as shown in figure 1.

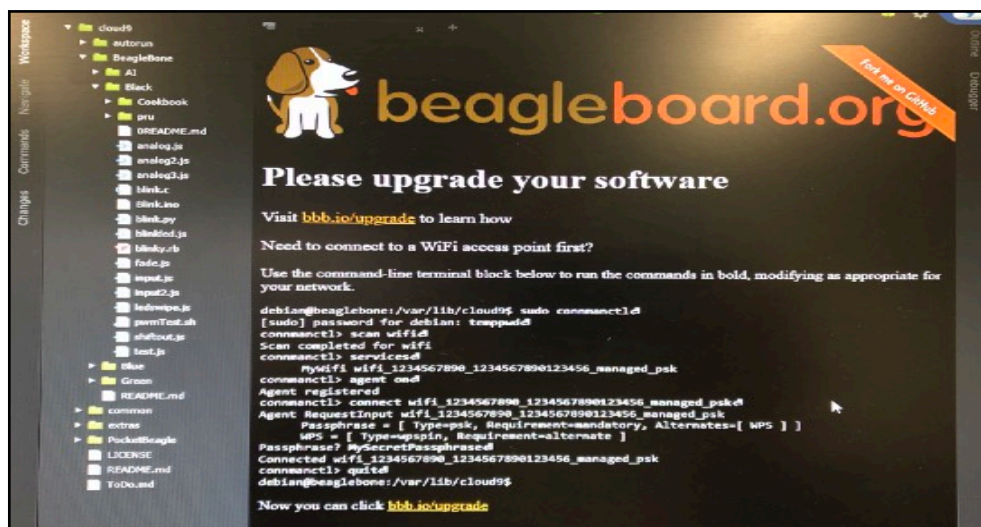


fig. 1

Then, we launched Putty, select SSH, enter same IP address and login. It displayed the information as shown in figure 2. After this done, we just left the Putty window. And we did not need to enter the password.

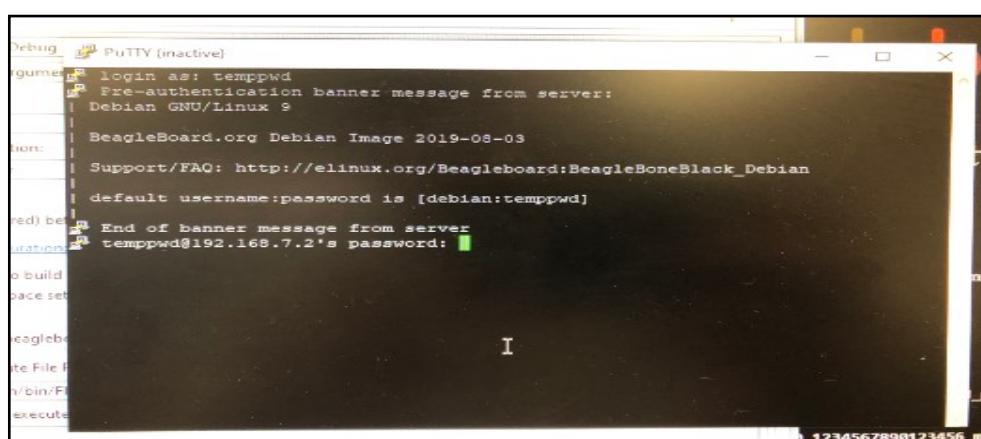


fig. 2

After everything has been set up, we tried different arguments: “flash”, “on” and “off”. The first LED would flash, on and off.

In the final step, we would make a “Cylon eye”. The code shown in the figure below.

```

19 #define LED0_PATH "/sys/class/leds/beaglebone:green:usr0"
20 #define LED1_PATH "/sys/class/leds/beaglebone:green:usr1"
21 #define LED2_PATH "/sys/class/leds/beaglebone:green:usr2"
22 #define LED3_PATH "/sys/class/leds/beaglebone:green:usr3"
23 #define DELAY 1000000
24 int i;
25
26 void removeTrigger(){
27     // remove the trigger from the LED
28     std::fstream fs;
29     //LED0
30     fs.open( LED0_PATH "/trigger", std::fstream::out);
31     fs << "none";
32     fs.close();
33     //LED1
34     fs.open( LED1_PATH "/trigger", std::fstream::out);
35     fs << "none";
36     fs.close();
37     //LED2
38     fs.open( LED2_PATH "/trigger", std::fstream::out);
39     fs << "none";
40     fs.close();
41     //LED3
42     fs.open( LED3_PATH "/trigger", std::fstream::out);
43     fs << "none";
44     fs.close();
45 }
46
47 void allLEDOff(){
48     std::fstream fs;
49     fs.open (LED0_PATH "/brightness", std::fstream::out);
50     fs << "0";
51     fs.close();
52
53     fs.open (LED1_PATH "/brightness", std::fstream::out);
54     fs << "0";
55     fs.close();
56
57     fs.open (LED2_PATH "/brightness", std::fstream::out);
58     fs << "0";
59     fs.close();
60
61     fs.open (LED3_PATH "/brightness", std::fstream::out);
62     fs << "0";
63     fs.close();
64 }
65
66 void led0_on(){
67     std::fstream fs;
68     fs.open (LED0_PATH "/brightness", std::fstream::out);
69     fs << "1";
70     fs.close();
71 }
72
73 void led1_on(){
74     std::fstream fs;
75     fs.open (LED1_PATH "/brightness", std::fstream::out);
76     fs << "1";
77     fs.close();
78 }
79
80 void led2_on(){
81     std::fstream fs;
82     fs.open (LED2_PATH "/brightness", std::fstream::out);
83     fs << "1";
84     fs.close();
85 }
86
87 void led3_on(){
88     std::fstream fs;
89     fs.open (LED3_PATH "/brightness", std::fstream::out);
90     fs << "1";
91     fs.close();
92 }
93
94 void led0_off(){
95     std::fstream fs;
96     fs.open (LED0_PATH "/brightness", std::fstream::out);
97     fs << "0";
98     fs.close();
99 }
100
101 void led1_off(){
102     std::fstream fs;
103     fs.open (LED1_PATH "/brightness", std::fstream::out);
104     fs << "0";
105     fs.close();
106 }
107
108 void led2_off(){
109     std::fstream fs;
110     fs.open (LED2_PATH "/brightness", std::fstream::out);
111     fs << "0";
112     fs.close();
113 }
114
115 void led3_off(){
116     std::fstream fs;
117     fs.open (LED3_PATH "/brightness", std::fstream::out);
118     fs << "0";
119     fs.close();
120 }
121

```

fig. 3

At the beginning, we defined 4 LEDs path to use in “Cylon eye” program. Then define two functions: removeTrigger() and allLEDOff(). removeTrigger(): remove the trigger of 4 LEDs. allLEDOff(): turn off all LEDs.

```

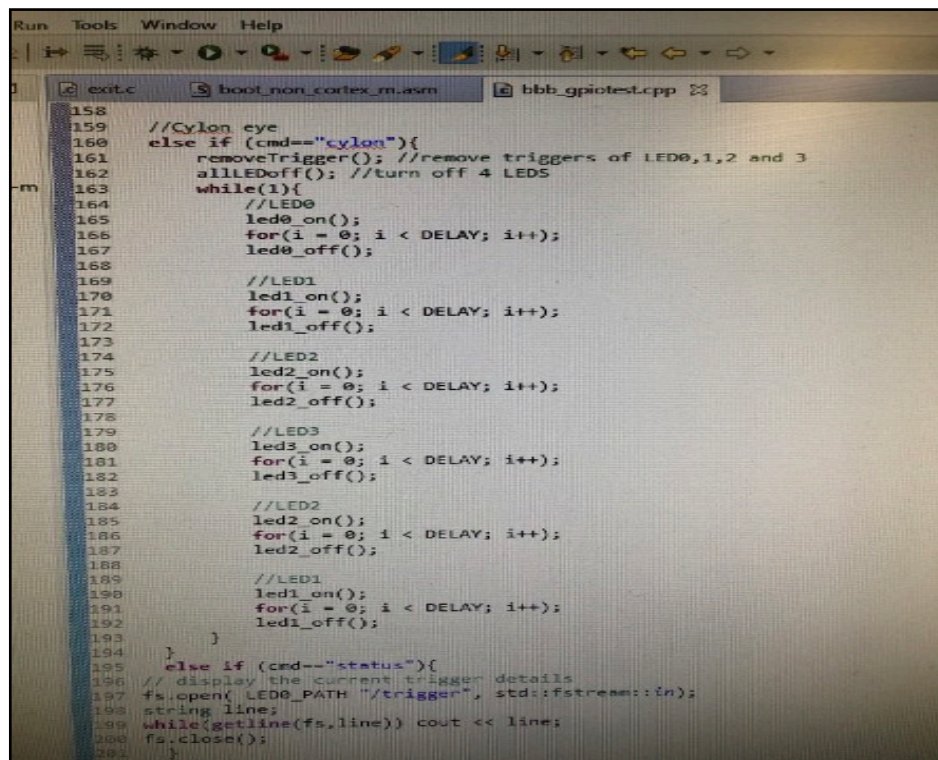
66 void led0_on(){
67     std::fstream fs;
68     fs.open (LED0_PATH "/brightness", std::fstream::out);
69     fs << "1";
70     fs.close();
71 }
72
73 void led1_on(){
74     std::fstream fs;
75     fs.open (LED1_PATH "/brightness", std::fstream::out);
76     fs << "1";
77     fs.close();
78 }
79
80 void led2_on(){
81     std::fstream fs;
82     fs.open (LED2_PATH "/brightness", std::fstream::out);
83     fs << "1";
84     fs.close();
85 }
86
87 void led3_on(){
88     std::fstream fs;
89     fs.open (LED3_PATH "/brightness", std::fstream::out);
90     fs << "1";
91     fs.close();
92 }
93
94 void led0_off(){
95     std::fstream fs;
96     fs.open (LED0_PATH "/brightness", std::fstream::out);
97     fs << "0";
98     fs.close();
99 }
100
101 void led1_off(){
102     std::fstream fs;
103     fs.open (LED1_PATH "/brightness", std::fstream::out);
104     fs << "0";
105     fs.close();
106 }
107
108 void led2_off(){
109     std::fstream fs;
110     fs.open (LED2_PATH "/brightness", std::fstream::out);
111     fs << "0";
112     fs.close();
113 }
114
115 void led3_off(){
116     std::fstream fs;
117     fs.open (LED3_PATH "/brightness", std::fstream::out);
118     fs << "0";
119     fs.close();
120 }
121

```

fig. 4

Then, we defined a bunch of function to control LED on and off.





```
158 //Cylon eye
159 else if (cmd=="cylon"){
160     removeTrigger(); //remove triggers of LED0,1,2 and 3
161     allLEDOff(); //turn off 4 LEDs
162     while(1){
163         //LED0
164         led0_on();
165         for(i = 0; i < DELAY; i++);
166         led0_off();
167
168         //LED1
169         led1_on();
170         for(i = 0; i < DELAY; i++);
171         led1_off();
172
173         //LED2
174         led2_on();
175         for(i = 0; i < DELAY; i++);
176         led2_off();
177
178         //LED3
179         led3_on();
180         for(i = 0; i < DELAY; i++);
181         led3_off();
182
183         //LED2
184         led2_on();
185         for(i = 0; i < DELAY; i++);
186         led2_off();
187
188         //LED1
189         led1_on();
190         for(i = 0; i < DELAY; i++);
191         led1_off();
192     }
193 }
194
195 else if (cmd=="status"){
196     // display the current trigger details
197     fs.open( LED0_PATH "/trigger", std::fstream::in);
198     string line;
199     while(getline(fs,line)) cout << line;
200     fs.close();
201 }
```

fig. 5

The main body of “Cylon eye” code shown in figure 5. First, removing all of the trigger to stop LED flashing. Next, turning off all of LEDs. This is to prevent the result to be affected by last command. Last, set a while loop, for example, turn LED0 on, after a delay, turn it off, then turn next LED on and then turn off again. Repeating these with the sequence: LED 0, 1, 2, 3, 2, and 1.