

ECEN301 Embedded Systems Lab 8

Cross compiler IDE and GNU debug

Daniel Eisen 300447549

October 16, 2020

1 Objectives

2 Methodology

Appendix

```
1  /*
2  * bbb_gpio_test.cpp
3  *
4  Simple On-board LED flashing program — written by Derek Molloy
5  for the ee402 module
6
7  This program uses USR LED 0 and can be executed in three ways:
8  makeLED on
9  makeLED off
10 makeLED flash (flash at 100ms intervals — on 50ms/off 50ms)
11 makeLED status (get the trigger status)
12 */
13
14 #include <iostream>
15 #include <fstream>
16 #include <string>
17 #include <cstdio>
18 #include <unistd.h>
19 using namespace std;
20
21 #define LED0_PATH "/sys/class/leds/beaglebone:green:usr0"
22 char LED_PATH[100] = "/sys/class/leds/beaglebone:green:usr1";
23
24 void removeTrigger()
25 {
26     // remove the trigger from the LED
27     std::fstream fs;
28     for (int i = 0; i < 4; ++i)
29     {
30         sprintf(LED_PATH, "/sys/class/leds/beaglebone:green:usr%d/trigger", i);
31         fs.open(LED_PATH, std::fstream::out);
32         fs << "none";
33     }
34     fs.close();
35 }
36
37 void clearLEDS()
38 {
39     std::fstream fs;
40     fs.open("/sys/class/leds/beaglebone:green:usr0/brightness",
41            std::fstream::out);
42     fs << "0";
43     fs.close();
44
45     fs.open("/sys/class/leds/beaglebone:green:usr1/brightness",
46            std::fstream::out);
47     fs << "0";
48     fs.close();
49
50     fs.open("/sys/class/leds/beaglebone:green:usr2/brightness",
51            std::fstream::out);
52     fs << "0";
53     fs.close();
54
55     fs.open("/sys/class/leds/beaglebone:green:usr3/brightness",
56            std::fstream::out);
57     fs << "0";
58     fs.close();
59 }
60
61 int main(int argc, char* argv[])
62 {
63     if (argc != 2)
64     {
65         cout << "Usage is makeLED and one of: on, off, flash or status" << endl;
66         cout << "e.g. makeLED flash" << endl;
67     }
68
69     string cmd(argv[1]);
70     std::fstream fs;
71     cout << "Starting the LED flash program" << endl;
72     cout << "The LED Path is: " << LED0_PATH << endl;
73
74     // select whether it is on, off or flash
75     if (cmd == "on")
76     {
77         removeTrigger();
78         fs.open(LED0_PATH "/brightness", std::fstream::out);
79         fs << "1";
80         fs.close();
81     }
82     else if (cmd == "off")
83     {
84         removeTrigger();
85         fs.open(LED0_PATH "/brightness", std::fstream::out);
86         fs << "0";
87         fs.close();
88     }
89     else if (cmd == "flash")
90     {
91         fs.open(LED0_PATH "/trigger", std::fstream::out);
92         fs << "timer";
93         fs.close();
94         fs.open(LED0_PATH "/delay_on", std::fstream::out);
95         fs << "50";
96         fs.close();
97         fs.open(LED0_PATH "/delay_off", std::fstream::out);
98         fs << "50";
99         fs.close();
100    }
101
102    else if (cmd == "cylon")
103    {
104        removeTrigger();
105        std::fstream fs;
106        while (1)
107        {
108            for (int i = 1; i < 4; ++i)
109            {
110                clearLEDS();
111                sprintf(LED_PATH,
112                       "/sys/class/leds/beaglebone:green:usr%d/brightness", i);
113                fs.open(LED_PATH, std::fstream::out);
114                fs << "1";
115                fs.close();
116
117                for (int d = 0; d < 10000000; ++d)
118                    ;
119
120                for (int i = 2; i > -1; --i)
121                {
122                    clearLEDS();
123                    sprintf(LED_PATH,
124                           "/sys/class/leds/beaglebone:green:usr%d/brightness", i);
125                    fs.open(LED_PATH, std::fstream::out);
126                    fs << "1";
127                    fs.close();
128
129                    for (int d = 0; d < 10000000; ++d)
130                        ;
131                }
132            }
133        }
134
135    }
136    else if (cmd == "status")
137    {
138        // display the current trigger details
139        fs.open(LED0_PATH "/trigger", std::fstream::in);
140        string line;
141        while (getline(fs, line))
142            cout << line;
143        fs.close();
144    }
145    else
146    {
147        cout << "Invalid command" << endl;
148    }
149    cout << "Finished the LED flash program" << endl;
150    return 0;
151 }
```