ECEN301 Embedded Systems Lab 7
Introduction to Embedded Linux

Daniel Eisen 300447549

October 14, 2020

1 Objectives

2 Methodology

Appendix

```c
/** Simple On-board LED flashing program - written in C by Derek Molloy
*    simple functional struture for the Exploring BeagleBone book
*
*    This program uses USR LED 3 and can be executed in three ways:
*         makeLED on
*         makeLED off
*         makeLED flash  (flash at 100ms intervals - on 50ms/off 50ms)
*         makeLED status (get the trigger status)
*
* Written by Derek Molloy for the book "Exploring BeagleBone: Tools and
* Techniques for Building with Embedded Linux" by John Wiley & Sons, 2014
* ISBN 9781118935125. Please see the file README.md in the repository root
* directory for copyright and GNU GPLv3 license information.            */

#include<stdio.h>
#include<stdlib.h>
#include<string.h>

#define LED0_PATH "/sys/class/leds/beaglebone:green:usr0"
#define LED1_PATH "/sys/class/leds/beaglebone:green:usr1"
#define LED2_PATH "/sys/class/leds/beaglebone:green:usr2"
#define LED3_PATH "/sys/class/leds/beaglebone:green:usr3"

void writeLED(char path[], char filename[], char value[]);
void removeTrigger();

int main(int argc, char* argv[])
{
    int c;
    char value[4];
    while(1){
        c = getchar();
        getchar();

        for (int i = 3; i >= 0; --i){
            sprintf(value, "%c",(c & (1 << i)) ? '1' : '0');
            removeTrigger();
            if (i==3) {
                writeLED(LED3_PATH, "/brightness", value);
            }else if (i==2) {
                writeLED(LED2_PATH, "/brightness", value);
            }else if (i==1){
                writeLED(LED1_PATH, "/brightness", value);
            }else if (i==0) {
                writeLED(LED0_PATH, "/brightness", value);
            }else {
                printf("default \n");
            }
        }
    }
    return 0;
}

void writeLED(char path[], char filename[], char value[])
{
    FILE* fp;   // create a file pointer fp
    char  fullFileName[100];  // to store the path and filename
    sprintf(fullFileName,"%s%s",path, filename); // write path and filename
    fp = fopen(fullFileName, "w+"); // open file for writing
    fprintf(fp, "%s", value);  // send the value to the file
    fclose(fp);  // close the file using the file pointer
}

void removeTrigger()
{
    writeLED(LED0_PATH,"/trigger", "none");
    writeLED(LED1_PATH,"/trigger", "none");
    writeLED(LED2_PATH,"/trigger", "none");
    writeLED(LED3_PATH,"/trigger", "none");
}
```