Parker Authier & Ben Kepner
Prof. Foster
CE 420-03L Lab 3
Fall Semester 10/31/201

Parker Authier & Ben Kepner
CE 420-03L Lab 3
Fall Semester 10/31/2019

Objectives:

The objective of this assignment was to use C to write a program to simulate a simple traffic light, demonstrating use of I/O capabilities of the PIC32 without the use of interrupts.

Hardware:

For this lab we used the colored LEDs and push buttons on the I/O shield.

Source Code:

main.c

```
/* ************************************************************************** */

/* ************************************************************************** */

#include <p32xxxx.h>
#include "TLights.h"

// configuration bit settings, Fcy=80MHz, Fpb=40MHz
#pragma config POSCMOD=XT, FNOSC=PRIPLL
#pragma config FPLLIDIV=DIV_2, FPLLMUL=MUL_20, FPLLODIV=DIV_1
#pragma config FPBDIV=DIV_2, FWDTEN=OFF, CP=OFF, BWP=OFF


//global variable
unsigned char mode = MODE1;        //system operating mode initialized to MODE1

int main(void) {
        unsigned char i;
        unsigned char new_mode;
        initTLights();
        initButtons();

//      //If you want to test the timer you can uncomment the following while loop
//      while(1) {
//      testTimer();
//      }
//

        mode = MODE1;

   //implement an infinite while loop that reads the input buttons and sets the operating mode
   //of the traffic light based on the inputs

        while (1)
        {
        mode = readButtons(mode);

        //state machine to handle modes
        switch(mode)
        {
```

```
        case MODE1:
                mode1TLights();
                break;
        case MODE2:
                mode2TLights();
                break;
        case MODE3:
                mode3TLights();
                mode = MODE1;
                break;
        }
        }


}
```

TLights.c

```
/* ************************************************************************** */
/**
  @Company
        Kettering University

  @File Name
        TLights.c

  @Summary
        Implements the functions necessary for the traffic light control

  @Description

  @Author
        Girma Tewolde
  @Last modification
        10/15/2019 @ 6:24 PM
 */
/* ************************************************************************** */

#include <p32xxxx.h>
#include "TLights.h"
```

```
//For testing timer function, toggle RED and GREEN LEDs on one of the light sets
void testTimer(void) {
        enableTLights();
        outputTLights(1, RED);
        msDelay(1000);
        outputTLights(1, GREEN);
        msDelay(1000);
}

//initialize the pushbutton switches (configure port direction)
void initButtons(void) {
        //configure the button pins as inputs
   S1_dir = IN;                 //S1
   S2_dir = IN;                                                 //S2

}

//read the switches inputs
//Input: the current operating mode of the lights is provided as input parameter
//output: the new operating mode of the traffic lights returned
//make sure you handle key bounce issues with software delays for debouncing
//disable the LEDs (turn off their power) while button is held down
unsigned char readButtons(unsigned char mode) {
        //the switches are active low
   //check if S2 is pressed ... to enter MODE3

   if(S2 == 0)
        {
        mode = MODE3;
        }


   //check if S1 is pressed to toggle between MODE1 and MODE2


   else if(S1 == 0)
        {
        if(mode == MODE1)
        mode = MODE2;
        else
        mode = MODE1;
        }

        // disable light and wait for button to be released
```

```
   disableTLights();
   while((S1 == 0) || (S2 == 0));
   enableTLights();

        return mode;
}

//initialize the traffic lights
//configure the I/O pin directions that control the decoder devices that in turn
//drive the LEDs
//Also configure the I/O pin direction for the LED power pin which is controlled
//by the MOSFET switch
void initTLights(void) {
        //configure decoder pins as outputs
        //Dec1 pins as outputs

   Dec1bit0_dir = OUT;
        Dec1bit1_dir = OUT;

        //Dec2 pins as outputs

        Dec2bit0_dir = OUT;
        Dec2bit1_dir = OUT;
        //configure LED enable pin as output
        LEDs_EN_dir = OUT;

}

//enable the lights by activating the MOSFET switch (active low signal))
void enableTLights(void) {
        //Enable LEDs ... controlled by active low MOSFET
        LEDs_EN = 0;

}

//disable the lights by deactivating the MOSFET switch (active low signal))
void disableTLights(void) {
        //Disable LEDs ... controlled by active low MOSFET
        LEDs_EN = 1;

}

//control lights
//lightNum -- specifies which light set to control (1 or 2)
//color -- specifies what color to turn on (RED, YELLOW, GREEN, WHITE)
```

```
void outputTLights(unsigned char lightNum, unsigned char color)
{
	if(lightNum == 0)
	disableTLights();
	if (lightNum == 1) {
	//control the light that is controlled by Dec1
	enableTLights();
	Dec1bit0 = color >> 0;
	Dec1bit1 = color >> 1;
	}
	if (lightNum == 2) {
	//control the light that is controlled by Dec2
	enableTLights();
	Dec2bit0 = color >> 0;
	Dec2bit1 = color >> 1;
	}
}

//runs the lights in operating mode MODE1
void mode1TLights(void) {

	enableTLights();
	outputTLights(1, GREEN);
	outputTLights(2,RED);
	msDelay(4000);
	outputTLights(1,YELLOW);
	msDelay(1000);
	outputTLights(1,RED);
	outputTLights(2,GREEN);
	msDelay(4000);
	outputTLights(2,YELLOW);
	msDelay(1000);
}

//runs the lights in operating mode MODE2
void mode2TLights(void) {

	enableTLights();
	outputTLights(1,RED);
	outputTLights(2,RED);
	msDelay(500);

	outputTLights(0,RED);
	msDelay(500);
```

```
	enableTLights();
	outputTLights(1,RED);
	outputTLights(2,RED);
	msDelay(500);

	outputTLights(0,RED);

	msDelay(500);

}

//runs the lights in operating mode MODE3
void mode3TLights(void)
{

	mode1TLights();
	outputTLights(1,RED);
	outputTLights(2,RED);
	msDelay(1000);
	multiplexColors(RED, WHITE, RED, WHITE, 5000);
	for(int i = 0; i < 6; i++)
	{
	outputTLights(1,RED);
	outputTLights(2,RED);
	msDelay(500);
	multiplexColors(RED, WHITE, RED, WHITE, 500);
	}

}

//generate time delay for the specified amount of milliseconds
void msDelay(unsigned int ms) {
	// Convert ms microseconds into how many clock ticks it will take
	unsigned int ticks = ms * (ONE_SEC_TICKS/1000);
//	ms *= TIMER_FREQ / 1000000 ; // Core Timer updates every 2 ticks

	_CP0_SET_COUNT(0); // Set Core Timer count to 0

	while (ticks > _CP0_GET_COUNT()); // Wait until Core Timer count reaches the
number we calculated earlier
}

//run two colors on the same light for a given time
void multiplexColors(unsigned char color1_1, unsigned char color1_2,
		unsigned char color2_1, unsigned char color2_2, unsigned int ms)
```

```
{
        //ensure LEDs are enabled
        enableTLights();

        // Convert ms microseconds into how many clock ticks it will take
        unsigned int ticks = ms * (ONE_SEC_TICKS/1000);

        _CP0_SET_COUNT(0); // Set Core Timer count to 0

        // Wait until Core Timer count reaches the number we calculated earlier
        while (ticks > _CP0_GET_COUNT())
        {
        outputTLights(1, color1_1);
        outputTLights(1, color1_2);
        outputTLights(2, color2_1);
        outputTLights(2, color2_2);
        }


}
```

Disassembly listing

```
Disassembly Listing for TLight
Generated From:
/home/ben/Sync/school/CE420/lab03/TLight.X/dist/default/debug/TLight.X.debug.elf
Oct 31, 2019 4:29:09 PM

---  /home/ben/Sync/school/CE420/lab03/TLight.X/main.c
------------------------------------------------
1:              /* ************************************************************************** */
2:
3:              /* ************************************************************************** */
4:
5:              #include <p32xxxx.h>
6:              #include "TLights.h"
7:
8:              // configuration bit settings, Fcy=80MHz, Fpb=40MHz
9:              #pragma config POSCMOD=XT, FNOSC=PRIPLL
10:             #pragma config FPLLIDIV=DIV_2, FPLLMUL=MUL_20, FPLLODIV=DIV_1
11:             #pragma config FPBDIV=DIV_2, FWDTEN=OFF, CP=OFF, BWP=OFF
12:
13:
14:             //global variable
```

```
15:                unsigned char mode = MODE1;        //system operating mode initialized to
MODE1
16:
17:                int main(void) {
9D000814  27BDFFE8   ADDIU SP, SP, -24
9D000818  AFBF0014   SW RA, 20(SP)
9D00081C  AFBE0010   SW FP, 16(SP)
9D000820  03A0F021   ADDU FP, SP, ZERO
18:                unsigned char i;
19:                unsigned char new_mode;
20:                initTLights();
9D000824  0F400060   JAL initTLights
9D000828  00000000   NOP
21:                initButtons();
9D00082C  0F40001A   JAL initButtons
9D000830  00000000   NOP
22:
23:                //         //If you want to test the timer you can uncomment the following while
loop
24:                //         while(1) {
25:                //         testTimer();
26:                //         }
27:                //
28:
29:                mode = MODE1;
9D000834  24020001   ADDIU V0, ZERO, 1
9D000838  A3828010   SB V0, -32752(GP)
30:
31:                            //implement an infinite while loop that reads the input buttons and sets
the operating mode
32:                            //of the traffic light based on the inputs
33:
34:                while (1)
35:                {
36:                            mode = readButtons(mode);
9D00083C  93828010   LBU V0, -32752(GP)
9D000840  00402021   ADDU A0, V0, ZERO
9D000844  0F40002C   JAL readButtons
9D000848  00000000   NOP
9D00084C  A3828010   SB V0, -32752(GP)
37:
38:                            //state machine to handle modes
39:                            switch(mode)
9D000850  93828010   LBU V0, -32752(GP)
9D000854  24030002   ADDIU V1, ZERO, 2
```

```
9D000858  1043000D   BEQ V0, V1, 0x9D000890
9D00085C  00000000   NOP
9D000860  24030003   ADDIU V1, ZERO, 3
9D000864  1043000E   BEQ V0, V1, 0x9D0008A0
9D000868  00000000   NOP
9D00086C  24030001   ADDIU V1, ZERO, 1
9D000870  10430003   BEQ V0, V1, 0x9D000880
9D000874  00000000   NOP
40:                         {
41:                         case MODE1:
42:                         mode1TLights();
9D000880  0F4000D4   JAL mode1TLights
9D000884  00000000   NOP
43:                         break;
9D000888  0B40022D   J 0x9D0008B4
9D00088C  00000000   NOP
44:                         case MODE2:
45:                         mode2TLights();
9D000890  0F400104   JAL mode2TLights
9D000894  00000000   NOP
46:                         break;
9D000898  0B40022D   J 0x9D0008B4
9D00089C  00000000   NOP
47:                         case MODE3:
48:                         mode3TLights();
9D0008A0  0F400136   JAL mode3TLights
9D0008A4  00000000   NOP
49:                         mode = MODE1;
9D0008A8  24020001   ADDIU V0, ZERO, 1
9D0008AC  A3828010   SB V0, -32752(GP)
50:                         break;
9D0008B0  00000000   NOP
51:                         }
52:                 }
9D000878  0B40020F   J 0x9D00083C
9D00087C  00000000   NOP
9D0008B4  0B40020F   J 0x9D00083C
9D0008B8  00000000   NOP
53:
54:
55:                 }
56:
---  /home/ben/Sync/school/CE420/lab03/TLight.X/TLights.c
--------------------------------------------
1:                  /* ************************************************************************** */
```

```
2:                  /**
3:                  @Company
4:                  Kettering University
5:
6:                  @File Name
7:                  TLights.c
8:
9:                  @Summary
10:                 Implements the functions necessary for the traffic light control
11:
12:                 @Description
13:
14:                 @Author
15:                 Girma Tewolde
16:                 @Last modification
17:                 10/15/2019 @ 6:24 PM
18:                 */
19:                 /* ************************************************************************** */
20:
21:                 #include <p32xxxx.h>
22:                 #include "TLights.h"
23:
24:                 //For testing timer function, toggle RED and GREEN LEDs on one of the light
sets
25:                 void testTimer(void) {
9D000000  27BDFFE8    ADDIU SP, SP, -24
9D000004  AFBF0014    SW RA, 20(SP)
9D000008  AFBE0010    SW FP, 16(SP)
9D00000C  03A0F021    ADDU FP, SP, ZERO
26:                     enableTLights();
9D000010  0F40007C    JAL enableTLights
9D000014  00000000    NOP
27:                     outputTLights(1, RED);
9D000018  24040001    ADDIU A0, ZERO, 1
9D00001C  00002821    ADDU A1, ZERO, ZERO
9D000020  0F400095    JAL outputTLights
9D000024  00000000    NOP
28:                     msDelay(1000);
9D000028  240403E8    ADDIU A0, ZERO, 1000
9D00002C  0F400172    JAL msDelay
9D000030  00000000    NOP
29:                     outputTLights(1, GREEN);
9D000034  24040001    ADDIU A0, ZERO, 1
9D000038  24050002    ADDIU A1, ZERO, 2
9D00003C  0F400095    JAL outputTLights
```

```
9D000040  00000000   NOP
30:               msDelay(1000);
9D000044  240403E8   ADDIU A0, ZERO, 1000
9D000048  0F400172   JAL msDelay
9D00004C  00000000   NOP
31:               }
9D000050  03C0E821   ADDU SP, FP, ZERO
9D000054  8FBF0014   LW RA, 20(SP)
9D000058  8FBE0010   LW FP, 16(SP)
9D00005C  27BD0018   ADDIU SP, SP, 24
9D000060  03E00008   JR RA
9D000064  00000000   NOP
32:
33:               //initialize the pushbutton switches (configure port direction)
34:               void initButtons(void) {
9D000068  27BDFFF8   ADDIU SP, SP, -8
9D00006C  AFBE0004   SW FP, 4(SP)
9D000070  03A0F021   ADDU FP, SP, ZERO
35:               //configure the button pins as inputs
36:                           S1_dir = IN;                        //S1
9D000074  3C03BF88   LUI V1, -16504
9D000078  946260C0   LHU V0, 24768(V1)
9D00007C  24040001   ADDIU A0, ZERO, 1
9D000080  7C824204   INS V0, A0, 8, 1
9D000084  A46260C0   SH V0, 24768(V1)
37:                           S2_dir = IN;                                                  //S2
9D000088  3C03BF88   LUI V1, -16504
9D00008C  946260C0   LHU V0, 24768(V1)
9D000090  24040001   ADDIU A0, ZERO, 1
9D000094  7C820004   INS V0, A0, 0, 1
9D000098  A46260C0   SH V0, 24768(V1)
38:
39:               }
9D00009C  03C0E821   ADDU SP, FP, ZERO
9D0000A0  8FBE0004   LW FP, 4(SP)
9D0000A4  27BD0008   ADDIU SP, SP, 8
9D0000A8  03E00008   JR RA
9D0000AC  00000000   NOP
40:
41:               //read the switches inputs
42:               //Input: the current operating mode of the lights is provided as input parameter
43:               //output: the new operating mode of the traffic lights returned
44:               //make sure you handle key bounce issues with software delays for
debouncing
45:               //disable the LEDs (turn off their power) while button is held down
```

```
46:             unsigned char readButtons(unsigned char mode) {
9D0000B0  27BDFFE8   ADDIU SP, SP, -24
9D0000B4  AFBF0014   SW RA, 20(SP)
9D0000B8  AFBE0010   SW FP, 16(SP)
9D0000BC  03A0F021   ADDU FP, SP, ZERO
9D0000C0  00801021   ADDU V0, A0, ZERO
9D0000C4  A3C20018   SB V0, 24(FP)
47:             //the switches are active low
48:                     //check if S2 is pressed ... to enter MODE3
49:
50:                     if(S2 == 0)
9D0000C8  3C02BF88   LUI V0, -16504
9D0000CC  8C4260D0   LW V0, 24784(V0)
9D0000D0  30420001   ANDI V0, V0, 1
9D0000D4  14400005   BNE V0, ZERO, 0x9D0000EC
9D0000D8  00000000   NOP
51:             {
52:                     mode = MODE3;
9D0000DC  24020003   ADDIU V0, ZERO, 3
9D0000E0  A3C20018   SB V0, 24(FP)
9D0000E4  0B40004A   J 0x9D000128
9D0000E8  00000000   NOP
53:             }
54:
55:
56:                     //check if S1 is pressed to toggle between MODE1 and MODE2
57:
58:
59:                     else if(S1 == 0)
9D0000EC  3C02BF88   LUI V0, -16504
9D0000F0  8C4260D0   LW V0, 24784(V0)
9D0000F4  30420100   ANDI V0, V0, 256
9D0000F8  1440000B   BNE V0, ZERO, 0x9D000128
9D0000FC  00000000   NOP
60:             {
61:                     if(mode == MODE1)
9D000100  93C30018   LBU V1, 24(FP)
9D000104  24020001   ADDIU V0, ZERO, 1
9D000108  14620005   BNE V1, V0, 0x9D000120
9D00010C  00000000   NOP
62:                     mode = MODE2;
9D000110  24020002   ADDIU V0, ZERO, 2
9D000114  A3C20018   SB V0, 24(FP)
9D000118  0B40004A   J 0x9D000128
9D00011C  00000000   NOP
```

```
63:                         else
64:                         mode = MODE1;
9D000120  24020001   ADDIU V0, ZERO, 1
9D000124  A3C20018   SB V0, 24(FP)
65:             }
66:
67:             // disable light and wait for button to be released
68:                         disableTLights();
9D000128  0F400088   JAL disableTLights
9D00012C  00000000   NOP
69:                         while((S1 == 0) || (S2 == 0));
9D000130  00000000   NOP
9D000134  3C02BF88   LUI V0, -16504
9D000138  8C4260D0   LW V0, 24784(V0)
9D00013C  30420100   ANDI V0, V0, 256
9D000140  1040FFFC   BEQ V0, ZERO, 0x9D000134
9D000144  00000000   NOP
9D000148  3C02BF88   LUI V0, -16504
9D00014C  8C4260D0   LW V0, 24784(V0)
9D000150  30420001   ANDI V0, V0, 1
9D000154  1040FFF7   BEQ V0, ZERO, 0x9D000134
9D000158  00000000   NOP
70:                         enableTLights();
9D00015C  0F40007C   JAL enableTLights
9D000160  00000000   NOP
71:
72:             return mode;
9D000164  93C20018   LBU V0, 24(FP)
73:             }
9D000168  03C0E821   ADDU SP, FP, ZERO
9D00016C  8FBF0014   LW RA, 20(SP)
9D000170  8FBE0010   LW FP, 16(SP)
9D000174  27BD0018   ADDIU SP, SP, 24
9D000178  03E00008   JR RA
9D00017C  00000000   NOP
74:
75:             //initialize the traffic lights
76:             //configure the I/O pin directions that control the decoder devices that in turn
77:             //drive the LEDs
78:             //Also configure the I/O pin direction for the LED power pin which is controlled
79:             //by the MOSFET switch
80:             void initTLights(void) {
9D000180  27BDFFF8   ADDIU SP, SP, -8
9D000184  AFBE0004   SW FP, 4(SP)
9D000188  03A0F021   ADDU FP, SP, ZERO
```

```
81:                 //configure decoder pins as outputs
82:                 //Dec1 pins as outputs
83:
84:                             Dec1bit0_dir = OUT;
9D00018C  3C03BF88   LUI V1, -16504
9D000190  946260C0   LHU V0, 24768(V1)
9D000194  7C021084   INS V0, ZERO, 2, 1
9D000198  A46260C0   SH V0, 24768(V1)
85:                 Dec1bit1_dir = OUT;
9D00019C  3C03BF88   LUI V1, -16504
9D0001A0  946260C0   LHU V0, 24768(V1)
9D0001A4  7C024A44   INS V0, ZERO, 9, 1
9D0001A8  A46260C0   SH V0, 24768(V1)
86:
87:                 //Dec2 pins as outputs
88:
89:                 Dec2bit0_dir = OUT;
9D0001AC  3C03BF88   LUI V1, -16504
9D0001B0  90626140   LBU V0, 24896(V1)
9D0001B4  7C020844   INS V0, ZERO, 1, 1
9D0001B8  A0626140   SB V0, 24896(V1)
90:                 Dec2bit1_dir = OUT;
9D0001BC  3C03BF88   LUI V1, -16504
9D0001C0  946260C0   LHU V0, 24768(V1)
9D0001C4  7C020844   INS V0, ZERO, 1, 1
9D0001C8  A46260C0   SH V0, 24768(V1)
91:                 //configure LED enable pin as output
92:                 LEDs_EN_dir = OUT;
9D0001CC  3C03BF88   LUI V1, -16504
9D0001D0  946260C0   LHU V0, 24768(V1)
9D0001D4  7C022104   INS V0, ZERO, 4, 1
9D0001D8  A46260C0   SH V0, 24768(V1)
93:
94:                 }
9D0001DC  03C0E821   ADDU SP, FP, ZERO
9D0001E0  8FBE0004   LW FP, 4(SP)
9D0001E4  27BD0008   ADDIU SP, SP, 8
9D0001E8  03E00008   JR RA
9D0001EC  00000000   NOP
95:
96:                 //enable the lights by activating the MOSFET switch (active low signal))
97:                 void enableTLights(void) {
9D0001F0  27BDFFF8   ADDIU SP, SP, -8
9D0001F4  AFBE0004   SW FP, 4(SP)
9D0001F8  03A0F021   ADDU FP, SP, ZERO
```

```
98:               //Enable LEDs ... controlled by active low MOSFET
99:               LEDs_EN = 0;
9D0001FC  3C03BF88   LUI V1, -16504
9D000200  946260D0   LHU V0, 24784(V1)
9D000204  7C022104   INS V0, ZERO, 4, 1
9D000208  A46260D0   SH V0, 24784(V1)
100:
101:              }
9D00020C  03C0E821   ADDU SP, FP, ZERO
9D000210  8FBE0004   LW FP, 4(SP)
9D000214  27BD0008   ADDIU SP, SP, 8
9D000218  03E00008   JR RA
9D00021C  00000000   NOP
102:
103:              //disable the lights by deactivating the MOSFET switch (active low signal))
104:              void disableTLights(void) {
9D000220  27BDFFF8   ADDIU SP, SP, -8
9D000224  AFBE0004   SW FP, 4(SP)
9D000228  03A0F021   ADDU FP, SP, ZERO
105:                        //Disable LEDs ... controlled by active low MOSFET
106:                        LEDs_EN = 1;
9D00022C  3C03BF88   LUI V1, -16504
9D000230  946260D0   LHU V0, 24784(V1)
9D000234  24040001   ADDIU A0, ZERO, 1
9D000238  7C822104   INS V0, A0, 4, 1
9D00023C  A46260D0   SH V0, 24784(V1)
107:
108:              }
9D000240  03C0E821   ADDU SP, FP, ZERO
9D000244  8FBE0004   LW FP, 4(SP)
9D000248  27BD0008   ADDIU SP, SP, 8
9D00024C  03E00008   JR RA
9D000250  00000000   NOP
109:
110:              //control lights
111:              //lightNum -- specifies which light set to control (1 or 2)
112:              //color -- specifies what color to turn on (RED, YELLOW, GREEN, WHITE)
113:              void outputTLights(unsigned char lightNum, unsigned char color)
114:              {
9D000254  27BDFFE8   ADDIU SP, SP, -24
9D000258  AFBF0014   SW RA, 20(SP)
9D00025C  AFBE0010   SW FP, 16(SP)
9D000260  03A0F021   ADDU FP, SP, ZERO
9D000264  00801821   ADDU V1, A0, ZERO
9D000268  00A01021   ADDU V0, A1, ZERO
```

```
9D00026C  A3C30018   SB V1, 24(FP)
9D000270  A3C2001C   SB V0, 28(FP)
115:                 if(lightNum == 0)
9D000274  93C20018   LBU V0, 24(FP)
9D000278  14400003   BNE V0, ZERO, .LVL16
9D00027C  00000000   NOP
116:                 disableTLights();
9D000280  0F400088   JAL disableTLights
9D000284  00000000   NOP
117:                 if (lightNum == 1) {
9D000288  93C30018   LBU V1, 24(FP)
9D00028C  24020001   ADDIU V0, ZERO, 1
9D000290  14620013   BNE V1, V0, 0x9D0002E0
9D000294  00000000   NOP
118:                 //control the light that is controlled by Dec1
119:                 enableTLights();
9D000298  0F40007C   JAL enableTLights
9D00029C  00000000   NOP
120:                 Dec1bit0 = color >> 0;
9D0002A0  93C2001C   LBU V0, 28(FP)
9D0002A4  30420001   ANDI V0, V0, 1
9D0002A8  304400FF   ANDI A0, V0, 255
9D0002AC  3C03BF88   LUI V1, -16504
9D0002B0  946260D0   LHU V0, 24784(V1)
9D0002B4  7C821084   INS V0, A0, 2, 1
9D0002B8  A46260D0   SH V0, 24784(V1)
121:                 Dec1bit1 = color >> 1;
9D0002BC  93C2001C   LBU V0, 28(FP)
9D0002C0  00021042   SRL V0, V0, 1
9D0002C4  304200FF   ANDI V0, V0, 255
9D0002C8  30420001   ANDI V0, V0, 1
9D0002CC  304400FF   ANDI A0, V0, 255
9D0002D0  3C03BF88   LUI V1, -16504
9D0002D4  946260D0   LHU V0, 24784(V1)
9D0002D8  7C824A44   INS V0, A0, 9, 1
9D0002DC  A46260D0   SH V0, 24784(V1)
122:                 }
123:                 if (lightNum == 2) {
9D0002E0  93C30018   LBU V1, 24(FP)
9D0002E4  24020002   ADDIU V0, ZERO, 2
9D0002E8  14620013   BNE V1, V0, 0x9D000338
9D0002EC  00000000   NOP
124:                 //control the light that is controlled by Dec2
125:                 enableTLights();
9D0002F0  0F40007C   JAL enableTLights
```

```
9D0002F4  00000000   NOP
126:                        Dec2bit0 = color >> 0;
9D0002F8  93C2001C   LBU V0, 28(FP)
9D0002FC  30420001   ANDI V0, V0, 1
9D000300  304400FF   ANDI A0, V0, 255
9D000304  3C03BF88   LUI V1, -16504
9D000308  90626150   LBU V0, 24912(V1)
9D00030C  7C820844   INS V0, A0, 1, 1
9D000310  A0626150   SB V0, 24912(V1)
127:                        Dec2bit1 = color >> 1;
9D000314  93C2001C   LBU V0, 28(FP)
9D000318  00021042   SRL V0, V0, 1
9D00031C  304200FF   ANDI V0, V0, 255
9D000320  30420001   ANDI V0, V0, 1
9D000324  304400FF   ANDI A0, V0, 255
9D000328  3C03BF88   LUI V1, -16504
9D00032C  946260D0   LHU V0, 24784(V1)
9D000330  7C820844   INS V0, A0, 1, 1
9D000334  A46260D0   SH V0, 24784(V1)
128:                        }
129:              }
9D000338  03C0E821   ADDU SP, FP, ZERO
9D00033C  8FBF0014   LW RA, 20(SP)
9D000340  8FBE0010   LW FP, 16(SP)
9D000344  27BD0018   ADDIU SP, SP, 24
9D000348  03E00008   JR RA
9D00034C  00000000   NOP
130:
131:              //runs the lights in operating mode MODE1
132:              void mode1TLights(void) {
9D000350  27BDFFE8   ADDIU SP, SP, -24
9D000354  AFBF0014   SW RA, 20(SP)
9D000358  AFBE0010   SW FP, 16(SP)
9D00035C  03A0F021   ADDU FP, SP, ZERO
133:
134:                        enableTLights();
9D000360  0F40007C   JAL enableTLights
9D000364  00000000   NOP
135:                        outputTLights(1, GREEN);
9D000368  24040001   ADDIU A0, ZERO, 1
9D00036C  24050002   ADDIU A1, ZERO, 2
9D000370  0F400095   JAL outputTLights
9D000374  00000000   NOP
136:                        outputTLights(2,RED);
9D000378  24040002   ADDIU A0, ZERO, 2
```

```
9D00037C  00002821   ADDU A1, ZERO, ZERO
9D000380  0F400095   JAL outputTLights
9D000384  00000000   NOP
137:                 msDelay(4000);
9D000388  24040FA0   ADDIU A0, ZERO, 4000
9D00038C  0F400172   JAL msDelay
9D000390  00000000   NOP
138:                 outputTLights(1,YELLOW);
9D000394  24040001   ADDIU A0, ZERO, 1
9D000398  24050001   ADDIU A1, ZERO, 1
9D00039C  0F400095   JAL outputTLights
9D0003A0  00000000   NOP
139:                 msDelay(1000);
9D0003A4  240403E8   ADDIU A0, ZERO, 1000
9D0003A8  0F400172   JAL msDelay
9D0003AC  00000000   NOP
140:                 outputTLights(1,RED);
9D0003B0  24040001   ADDIU A0, ZERO, 1
9D0003B4  00002821   ADDU A1, ZERO, ZERO
9D0003B8  0F400095   JAL outputTLights
9D0003BC  00000000   NOP
141:                 outputTLights(2,GREEN);
9D0003C0  24040002   ADDIU A0, ZERO, 2
9D0003C4  24050002   ADDIU A1, ZERO, 2
9D0003C8  0F400095   JAL outputTLights
9D0003CC  00000000   NOP
142:                 msDelay(4000);
9D0003D0  24040FA0   ADDIU A0, ZERO, 4000
9D0003D4  0F400172   JAL msDelay
9D0003D8  00000000   NOP
143:                 outputTLights(2,YELLOW);
9D0003DC  24040002   ADDIU A0, ZERO, 2
9D0003E0  24050001   ADDIU A1, ZERO, 1
9D0003E4  0F400095   JAL outputTLights
9D0003E8  00000000   NOP
144:                 msDelay(1000);
9D0003EC  240403E8   ADDIU A0, ZERO, 1000
9D0003F0  0F400172   JAL msDelay
9D0003F4  00000000   NOP
145:             }
9D0003F8  03C0E821   ADDU SP, FP, ZERO
9D0003FC  8FBF0014   LW RA, 20(SP)
9D000400  8FBE0010   LW FP, 16(SP)
9D000404  27BD0018   ADDIU SP, SP, 24
9D000408  03E00008   JR RA
```

```
9D00040C  00000000   NOP
146:
147:              //runs the lights in operating mode MODE2
148:              void mode2TLights(void) {
9D000410  27BDFFE8   ADDIU SP, SP, -24
9D000414  AFBF0014   SW RA, 20(SP)
9D000418  AFBE0010   SW FP, 16(SP)
9D00041C  03A0F021   ADDU FP, SP, ZERO
149:
150:                        enableTLights();
9D000420  0F40007C   JAL enableTLights
9D000424  00000000   NOP
151:                        outputTLights(1,RED);
9D000428  24040001   ADDIU A0, ZERO, 1
9D00042C  00002821   ADDU A1, ZERO, ZERO
9D000430  0F400095   JAL outputTLights
9D000434  00000000   NOP
152:                        outputTLights(2,RED);
9D000438  24040002   ADDIU A0, ZERO, 2
9D00043C  00002821   ADDU A1, ZERO, ZERO
9D000440  0F400095   JAL outputTLights
9D000444  00000000   NOP
153:                        msDelay(500);
9D000448  240401F4   ADDIU A0, ZERO, 500
9D00044C  0F400172   JAL msDelay
9D000450  00000000   NOP
154:
155:                        outputTLights(0,RED);
9D000454  00002021   ADDU A0, ZERO, ZERO
9D000458  00002821   ADDU A1, ZERO, ZERO
9D00045C  0F400095   JAL outputTLights
9D000460  00000000   NOP
156:                        msDelay(500);
9D000464  240401F4   ADDIU A0, ZERO, 500
9D000468  0F400172   JAL msDelay
9D00046C  00000000   NOP
157:
158:                        enableTLights();
9D000470  0F40007C   JAL enableTLights
9D000474  00000000   NOP
159:                        outputTLights(1,RED);
9D000478  24040001   ADDIU A0, ZERO, 1
9D00047C  00002821   ADDU A1, ZERO, ZERO
9D000480  0F400095   JAL outputTLights
9D000484  00000000   NOP
```

```
160:                        outputTLights(2,RED);
9D000488  24040002   ADDIU A0, ZERO, 2
9D00048C  00002821   ADDU A1, ZERO, ZERO
9D000490  0F400095   JAL outputTLights
9D000494  00000000   NOP
161:                        msDelay(500);
9D000498  240401F4   ADDIU A0, ZERO, 500
9D00049C  0F400172   JAL msDelay
9D0004A0  00000000   NOP
162:
163:                        outputTLights(0,RED);
9D0004A4  00002021   ADDU A0, ZERO, ZERO
9D0004A8  00002821   ADDU A1, ZERO, ZERO
9D0004AC  0F400095   JAL outputTLights
9D0004B0  00000000   NOP
164:
165:                        msDelay(500);
9D0004B4  240401F4   ADDIU A0, ZERO, 500
9D0004B8  0F400172   JAL msDelay
9D0004BC  00000000   NOP
166:
167:            }
9D0004C0  03C0E821   ADDU SP, FP, ZERO
9D0004C4  8FBF0014   LW RA, 20(SP)
9D0004C8  8FBE0010   LW FP, 16(SP)
9D0004CC  27BD0018   ADDIU SP, SP, 24
9D0004D0  03E00008   JR RA
9D0004D4  00000000   NOP
168:
169:            //runs the lights in operating mode MODE3
170:            void mode3TLights(void)
171:            {
9D0004D8  27BDFFD8   ADDIU SP, SP, -40
9D0004DC  AFBF0024   SW RA, 36(SP)
9D0004E0  AFBE0020   SW FP, 32(SP)
9D0004E4  03A0F021   ADDU FP, SP, ZERO
172:
173:                        mode1TLights();
9D0004E8  0F4000D4   JAL mode1TLights
9D0004EC  00000000   NOP
174:                        outputTLights(1,RED);
9D0004F0  24040001   ADDIU A0, ZERO, 1
9D0004F4  00002821   ADDU A1, ZERO, ZERO
9D0004F8  0F400095   JAL outputTLights
9D0004FC  00000000   NOP
```

```
175:                    outputTLights(2,RED);
9D000500  24040002   ADDIU A0, ZERO, 2
9D000504  00002821   ADDU A1, ZERO, ZERO
9D000508  0F400095   JAL outputTLights
9D00050C  00000000   NOP
176:                    msDelay(1000);
9D000510  240403E8   ADDIU A0, ZERO, 1000
9D000514  0F400172   JAL msDelay
9D000518  00000000   NOP
177:                    multiplexColors(RED, WHITE, RED, WHITE, 5000);
9D00051C  24021388   ADDIU V0, ZERO, 5000
9D000520  AFA20010   SW V0, 16(SP)
9D000524  00002021   ADDU A0, ZERO, ZERO
9D000528  24050003   ADDIU A1, ZERO, 3
9D00052C  00003021   ADDU A2, ZERO, ZERO
9D000530  24070003   ADDIU A3, ZERO, 3
9D000534  0F400191   JAL multiplexColors
9D000538  00000000   NOP
178:                    for(int i = 0; i < 6; i++)
9D00053C  AFC00018   SW ZERO, 24(FP)
9D000540  0B400168   J 0x9D0005A0
9D000544  00000000   NOP
9D000594  8FC20018   LW V0, 24(FP)
9D000598  24420001   ADDIU V0, V0, 1
9D00059C  AFC20018   SW V0, 24(FP)
9D0005A0  8FC20018   LW V0, 24(FP)
9D0005A4  28420006   SLTI V0, V0, 6
9D0005A8  1440FFE7   BNE V0, ZERO, 0x9D000548
9D0005AC  00000000   NOP
179:                    {
180:                    outputTLights(1,RED);
9D000548  24040001   ADDIU A0, ZERO, 1
9D00054C  00002821   ADDU A1, ZERO, ZERO
9D000550  0F400095   JAL outputTLights
9D000554  00000000   NOP
181:                    outputTLights(2,RED);
9D000558  24040002   ADDIU A0, ZERO, 2
9D00055C  00002821   ADDU A1, ZERO, ZERO
9D000560  0F400095   JAL outputTLights
9D000564  00000000   NOP
182:                    msDelay(500);
9D000568  240401F4   ADDIU A0, ZERO, 500
9D00056C  0F400172   JAL msDelay
9D000570  00000000   NOP
183:                    multiplexColors(RED, WHITE, RED, WHITE, 500);
```

```
9D000574  240201F4   ADDIU V0, ZERO, 500
9D000578  AFA20010   SW V0, 16(SP)
9D00057C  00002021   ADDU A0, ZERO, ZERO
9D000580  24050003   ADDIU A1, ZERO, 3
9D000584  00003021   ADDU A2, ZERO, ZERO
9D000588  24070003   ADDIU A3, ZERO, 3
9D00058C  0F400191   JAL multiplexColors
9D000590  00000000   NOP
184:                        }
185:
186:              }
9D0005B0  03C0E821   ADDU SP, FP, ZERO
9D0005B4  8FBF0024   LW RA, 36(SP)
9D0005B8  8FBE0020   LW FP, 32(SP)
9D0005BC  27BD0028   ADDIU SP, SP, 40
9D0005C0  03E00008   JR RA
9D0005C4  00000000   NOP
187:
188:              //generate time delay for the specified amount of milliseconds
189:              void msDelay(unsigned int ms) {
9D0005C8  27BDFFF0   ADDIU SP, SP, -16
9D0005CC  AFBF000C   SW RA, 12(SP)
9D0005D0  AFBE0008   SW FP, 8(SP)
9D0005D4  03A0F021   ADDU FP, SP, ZERO
9D0005D8  AFC40010   SW A0, 16(FP)
190:                        // Convert ms microseconds into how many clock ticks it will take
191:                        unsigned int ticks = ms * (ONE_SEC_TICKS/1000);
9D0005DC  8FC40010   LW A0, 16(FP)
9D0005E0  00801821   ADDU V1, A0, ZERO
9D0005E4  00031080   SLL V0, V1, 2
9D0005E8  00401821   ADDU V1, V0, ZERO
9D0005EC  00031140   SLL V0, V1, 5
9D0005F0  00431023   SUBU V0, V0, V1
9D0005F4  00441021   ADDU V0, V0, A0
9D0005F8  00021880   SLL V1, V0, 2
9D0005FC  00431021   ADDU V0, V0, V1
9D000600  00021180   SLL V0, V0, 6
9D000604  AFC20000   SW V0, 0(FP)
192:              //        ms *= TIMER_FREQ / 1000000 ; // Core Timer updates every 2 ticks
193:
194:                        _CP0_SET_COUNT(0); // Set Core Timer count to 0
9D000608  00001021   ADDU V0, ZERO, ZERO
9D00060C  40824800   MTC0 V0, Count
9D000610  000000C0   EHB
195:
```

```
196:                    while (ticks > _CP0_GET_COUNT()); // Wait until Core Timer count
reaches the number we calculated earlier
9D000614  00000000   NOP
9D000618  40034800   MFC0 V1, Count
9D00061C  8FC20000   LW V0, 0(FP)
9D000620  0062102B   SLTU V0, V1, V0
9D000624  1440FFFC   BNE V0, ZERO, 0x9D000618
9D000628  00000000   NOP
197:            }
9D00062C  03C0E821   ADDU SP, FP, ZERO
9D000630  8FBF000C   LW RA, 12(SP)
9D000634  8FBE0008   LW FP, 8(SP)
9D000638  27BD0010   ADDIU SP, SP, 16
9D00063C  03E00008   JR RA
9D000640  00000000   NOP
198:
199:            //run two colors on the same light for a given time
200:            void multiplexColors(unsigned char color1_1, unsigned char color1_2,
201:                                 unsigned char color2_1, unsigned char color2_2, unsigned int
ms)
202:            {
9D000644  27BDFFE0   ADDIU SP, SP, -32
9D000648  AFBF001C   SW RA, 28(SP)
9D00064C  AFBE0018   SW FP, 24(SP)
9D000650  03A0F021   ADDU FP, SP, ZERO
9D000654  00804021   ADDU T0, A0, ZERO
9D000658  00A02021   ADDU A0, A1, ZERO
9D00065C  00C01821   ADDU V1, A2, ZERO
9D000660  00E01021   ADDU V0, A3, ZERO
9D000664  A3C80020   SB T0, 32(FP)
9D000668  A3C40024   SB A0, 36(FP)
9D00066C  A3C30028   SB V1, 40(FP)
9D000670  A3C2002C   SB V0, 44(FP)
203:                    //ensure LEDs are enabled
204:                    enableTLights();
9D000674  0F40007C   JAL enableTLights
9D000678  00000000   NOP
205:
206:                    // Convert ms microseconds into how many clock ticks it will take
207:                    unsigned int ticks = ms * (ONE_SEC_TICKS/1000);
9D00067C  8FC40030   LW A0, 48(FP)
9D000680  00801821   ADDU V1, A0, ZERO
9D000684  00031080   SLL V0, V1, 2
9D000688  00401821   ADDU V1, V0, ZERO
9D00068C  00031140   SLL V0, V1, 5
```

```
9D000690  00431023   SUBU V0, V0, V1
9D000694  00441021   ADDU V0, V0, A0
9D000698  00021880   SLL V1, V0, 2
9D00069C  00431021   ADDU V0, V0, V1
9D0006A0  00021180   SLL V0, V0, 6
9D0006A4  AFC20010   SW V0, 16(FP)
208:
209:                 _CP0_SET_COUNT(0); // Set Core Timer count to 0
9D0006A8  00001021   ADDU V0, ZERO, ZERO
9D0006AC  40824800   MTC0 V0, Count
9D0006B0  000000C0   EHB
210:
211:                 // Wait until Core Timer count reaches the number we calculated earlier
212:                 while (ticks > _CP0_GET_COUNT())
9D0006B4  0B4001C3   J .LVL67
9D0006B8  00000000   NOP
9D00070C  40034800   MFC0 V1, Count
9D000710  8FC20010   LW V0, 16(FP)
9D000714  0062102B   SLTU V0, V1, V0
9D000718  1440FFE8   BNE V0, ZERO, 0x9D0006BC
9D00071C  00000000   NOP
213:                 {
214:                 outputTLights(1, color1_1);
9D0006BC  93C20020   LBU V0, 32(FP)
9D0006C0  24040001   ADDIU A0, ZERO, 1
9D0006C4  00402821   ADDU A1, V0, ZERO
9D0006C8  0F400095   JAL outputTLights
9D0006CC  00000000   NOP
215:                 outputTLights(1, color1_2);
9D0006D0  93C20024   LBU V0, 36(FP)
9D0006D4  24040001   ADDIU A0, ZERO, 1
9D0006D8  00402821   ADDU A1, V0, ZERO
9D0006DC  0F400095   JAL outputTLights
9D0006E0  00000000   NOP
216:                 outputTLights(2, color2_1);
9D0006E4  93C20028   LBU V0, 40(FP)
9D0006E8  24040002   ADDIU A0, ZERO, 2
9D0006EC  00402821   ADDU A1, V0, ZERO
9D0006F0  0F400095   JAL outputTLights
9D0006F4  00000000   NOP
217:                 outputTLights(2, color2_2);
9D0006F8  93C2002C   LBU V0, 44(FP)
9D0006FC  24040002   ADDIU A0, ZERO, 2
9D000700  00402821   ADDU A1, V0, ZERO
9D000704  0F400095   JAL outputTLights
```

```
9D000708  00000000   NOP
218:                        }
219:
220:
221:              }
9D000720  03C0E821   ADDU SP, FP, ZERO
9D000724  8FBF001C   LW RA, 28(SP)
9D000728  8FBE0018   LW FP, 24(SP)
9D00072C  27BD0020   ADDIU SP, SP, 32
9D000730  03E00008   JR RA
9D000734  00000000   NOP
```

Conclusions:

Without interrupts, the user experience pushing buttons is lacking.