

Part 1

For this part, we toggled the LEDs on and off on the STK-600 board in a back-and-forth sort of manner. The effect essentially resembled the LEDs on the car from the TV series *Knight Rider*. This part was written in C.

The code loaded the board's registers, set up the proper ports, output the proper values to said ports, created an LED pattern, created a delay for the light shift, then iterated through the process of lighting up one LED after another to give the back-and-forth effect.

```
1  /*
2  * Exercise_1.c
3  *
4  * Created: 2/6 2:27:46
5  * Author : Student
6  */
7
8  #include <avr/io.h>
9  #define F_CPU 8000000UL
10 #include <util/delay.h>
11
12 int main(void)
13 {
14     //Declare Variables
15     int led = 1;
16     int flag = 1;
17
18     //Set DDRB so LEDs are out
19     DDRB = 0xff;
20
21     //Create do-while loop
22     do {
23
24         //Set the output equal to ~led
25         PORTB = ~led;
26
27         //Detect shift direction
28         if (flag){
29             led <<= 1;    //Shift left once
30
31             //Detect if reached 256
32             if (led==256){
33                 flag = 0;
34                 led=led/4;    //Grab first two bits of led to restart at "10"
35             }
36         }
37         else{
38             led >>= 1;    //Shift right once
39
40             //Detect if reached 1
41             if (led == 1){
42                 flag = 1;
43             }
44         }
45         _delay_ms(200);
46     } while (1);
47
48     return 0;
49 }
50
```

Figure 1: Code from Part 1

Part 2

For this part of the lab, we used the STK-600 board to mimic a set of traffic lights at an intersection. The lights properly cycled through the red, yellow, and green lights to direct traffic from "2" directions, and flashed the yellow lights 5 times when necessary. This part was also written in C

Again, this code loaded the board's registers, created two different LED patterns, created the proper delays for both patterns (as well as the 5 blink repeat for the yellow lights), set up the ports, and iterated through the entire process.

```
1  /*
2   * Second_Attempt_Exercise_2.c
3   *
4   * Created: 2/6 4:23:40
5   * Author : Student
6   */
7
8
9  #define F_CPU 8000000UL
10 #include <avr/io.h>
11 #include <avr/pgmspace.h>
12 #include <util/delay.h>
13
14 //Declare variables and constants
15 const uint8_t tab[] PROGMEM = {0x0c,0x14,0x04,0x21,0x22,0x20};
16 int led = 0x0c;
17 int i,j;
18
19
```

Figure 2: First Snippet of Part 2

The main function for the code is on the next page.

```

18
19
20 int main(void)
21 {
22     //Declare output ports and LED assignments
23     DDRB = 0xff;
24     PORTB = ~led;
25
26     //Begin while loop
27     while (1) {
28
29         //Initial read of array values
30         led = pgm_read_byte(&tab[0]);
31         PORTB = ~led;
32
33         //delay 4 seconds
34         _delay_ms(4000);
35
36         //Loop through light changes 5 times
37         for(i = 0; i < 5; i++){
38
39             //Loop to read second and third light values tab[1-2] and display
40             for (j = 0; j < 2; j++){
41                 led = pgm_read_byte(&tab[j+1]);
42                 PORTB = ~led;
43
44                 //slight delay between each light transition
45                 _delay_ms(400);
46             }
47         }
48
49         //Read the 4th light values (tab[3]) and display
50         led = pgm_read_byte(&tab[3]);
51         PORTB = ~led;
52
53         //4 second delay
54         _delay_ms(4000);
55
56         //Loop through second set of light changes 5 times
57         for (i = 0; i < 5; i++){
58
59             //Loop through 5th & 6th light values
60             for (j = 0; j < 2; j++){
61                 led = pgm_read_byte(&tab[j+4]);
62                 PORTB = ~led;
63
64                 //Slight delay between each light transition
65                 _delay_ms(400);
66             }
67         }
68
69         return 0;
70     }
71 }
72
73

```

Figure 3: Second Snippet from Part 2