

```

1  // ***** 0. Documentation Section *****
2  // TableTrafficLight.c for Lab 10
3  // Runs on LM4F120/TM4C123
4  // Index implementation of a Moore finite state machine to operate a traffic light.
5  // east/west red light connected to PB5
6  // east/west yellow light connected to PB4
7  // east/west green light connected to PB3
8  // north/south facing red light connected to PB2
9  // north/south facing yellow light connected to PB1
10 // north/south facing green light connected to PB0
11 // pedestrian detector connected to PE2 (1=pedestrian present)
12 // north/south car detector connected to PE1 (1=car present)
13 // east/west car detector connected to PE0 (1=car present)
14 // "walk" light connected to PF3 (built-in green LED)
15 // "don't walk" light connected to PF1 (built-in red LED)
16
17 // ***** 1. Pre-processor Directives Section *****
18 #include "TEaS.h"
19 #include "tm4c123gh6pm.h"
20
21 // ***** 2. Global Declarations Section *****
22 struct state{
23     unsigned long out; // output signals or cars
24     unsigned long walk; // outputs signals for pedestrian
25     unsigned long time; // Delay 10ms
26     unsigned long Next[9]; //switches
27 };
28 typedef const struct state SType;
29
30 #define GoW 0 // go toward west
31 #define WW 1 // wait west
32 #define GoS 2 // go toward south
33 #define WS 3 // wait south
34 #define GoPed 4 // pedestrian "walk"
35 #define RUN1 5 // RUN1,RUN2,RUN3 hurry up toggling signal
36 #define RUN2 6
37 #define RUN3 7
38 #define stopPed 8 // pedestraings "do not walk"
39 //Initalize FSM
40 SType FSM[8] ={
41
42     {0x21, 0x02, 50, {GoW, WW, GoW, WW, WW, WW, WW, WW}},
43     {0x22, 0x02, 35, {GoS, GoS, GoS, GoS, GoPed, GoS, GoPed, GoS}},
44     {0x0C, 0x02, 50, {GoS, GoS, WS, WS, WS, WS, WS, WS}},
45     {0x14, 0x02, 35, {GoW, GoW, GoW, GoW, GoPed, GoPed, GoW, GoPed}},
46     {0x24, 0x08, 50, {GoPed, RUN1, RUN1, RUN1, GoPed, RUN1, RUN1, RUN1}},
47     {0x24, 0x02, 15, {RUN2, RUN2, RUN2, RUN2, RUN2, RUN2, RUN2, RUN2}},
48     {0x24, 0x00, 15, {RUN3, RUN3, RUN3, RUN3, RUN3, RUN3, RUN3, RUN3}},
49     {0x24, 0x00, 15, {GoW, GoS, GoW, GoW, GoPed, GoS, GoW, GoW}}
50 };
51
52 unsigned long S; // current state
53 unsigned long input; // sensor inputs
54
55 // FUNCTION PROTOTYPES: Each subroutine defined
56 void DisableInterrupts(void); // Disable interrupts
57 void EnableInterrupts(void); // Enable interrupts
58
59 // initlize PLL
60
61 void SysTick_Init(void); // initlize timer
62 void PortB_Init(void); // initlize port B
63 void PortE_Init(void); // initlize port E
64 void PortF_Init(void); // initlize port F
65 void SysTick_Wait(unsigned long delay);
66 void SysTick_wait10ms(unsigned long delay); // delay for 10 ms
67 void Init_All(void); // initlizer for all functions
68
69 // ***** 3. Subroutines Section *****
70 int main(void){
71     TEaS_Init(SW_PIN_PE210, LED_PIN_PB543210,ScopeOff); // activate grader and set system clock to
72     80 MHz
73     EnableInterrupts();

```

```

73     Init_All(); // Activate all functions
74     S = GoW;
75     while(1){
76         GPIO_PORTB_DATA_R = FSM[S].out;
77         GPIO_PORTF_DATA_R = FSM[S].walk;
78         SysTick_wait10ms(FSM[S].time);
79         input = GPIO_PORTE_DATA_R;
80         S = FSM[S].Next[input];
81     }
82 }
83
84
85 void Init_All(void) {
86     SysTick_Init();
87     //Phase locked loop
88     //PLL_Init();
89     PortB_Init();
90     PortE_Init();
91     PortF_Init();
92 }
93
94 // PORT B INITIALIZATION
95 void PortB_Init(void){
96     volatile unsigned long delay;
97     SYSCTL_RCGC2_R |= 0x02;
98     delay = SYSCTL_RCGC2_R;
99     GPIO_PORTB_LOCK_R |= 0x4C4F434B;
100    GPIO_PORTB_CR_R |= 0x3F;
101    GPIO_PORTB_DIR_R |= 0x3F;
102    GPIO_PORTB_AFSEL_R &= ~ (0x3F);
103    GPIO_PORTB_DEN_R |= 0x3F;
104    GPIO_PORTB_AMSEL_R &= ~ (0x3F);
105    GPIO_PORTB_PCTL_R &= ~ (0x00FFFFFF);
106 }
107 // PORT E INITIALIZATION
108 void PortE_Init(void){
109     volatile unsigned long delay;
110     SYSCTL_RCGC2_R |= 0x10;
111     delay = SYSCTL_RCGC2_R;
112     GPIO_PORTE_LOCK_R |= 0x4C4F434B;
113     GPIO_PORTE_CR_R |= 0x07;
114     GPIO_PORTE_DIR_R &= ~ (0x07);
115     GPIO_PORTE_AFSEL_R &= ~ (0x07);
116     GPIO_PORTE_DEN_R |= 0x07;
117     GPIO_PORTE_AMSEL_R &= ~ (0x07);
118     GPIO_PORTE_PCTL_R &= ~ (0x00000FFF);
119 }
120 // PORT F INITIALIZATION
121 void PortF_Init(void){
122     volatile unsigned long delay;
123     SYSCTL_RCGC2_R |= 0x00000020;
124     delay = SYSCTL_RCGC2_R;
125     GPIO_PORTF_LOCK_R |= 0x4C4F434B;
126     GPIO_PORTF_CR_R |= 0x0A;
127     GPIO_PORTF_DIR_R |= (0x0A);
128     GPIO_PORTF_AFSEL_R &= ~ (0x0A);
129     GPIO_PORTF_DEN_R |= 0x0A;
130     GPIO_PORTF_AMSEL_R &= ~ (0x0A);
131     GPIO_PORTF_PCTL_R &= ~ (0x0000F0F0);
132 }
133 // SYSTICK INITIALIZATION
134 void SysTick_Init(void){
135     NVIC_ST_CTRL_R = 0;
136     NVIC_ST_CTRL_R = 0x00000005;
137 }
138
139 // DELAY 10ms function
140 void SysTick_Wait (unsigned long delay){
141     NVIC_ST_RELOAD_R = delay-1 ;
142     NVIC_ST_CURRENT_R = 0;
143     while ((NVIC_ST_CTRL_R&0x00010000) == 0){
144     }
145 }

```

```
146
147 // Delay function
148 void SysTick_wait10ms(unsigned long delay){
149     unsigned long i;
150     for(i =0; i<delay; i++){
151         SysTick_Wait(800000); // 10ms delay function 800000*12.5ns=10ms
152     }
153 }
154
```