CECS 346 - Spring 2021

Project 1 - Traffic Light Controller (Advanced)

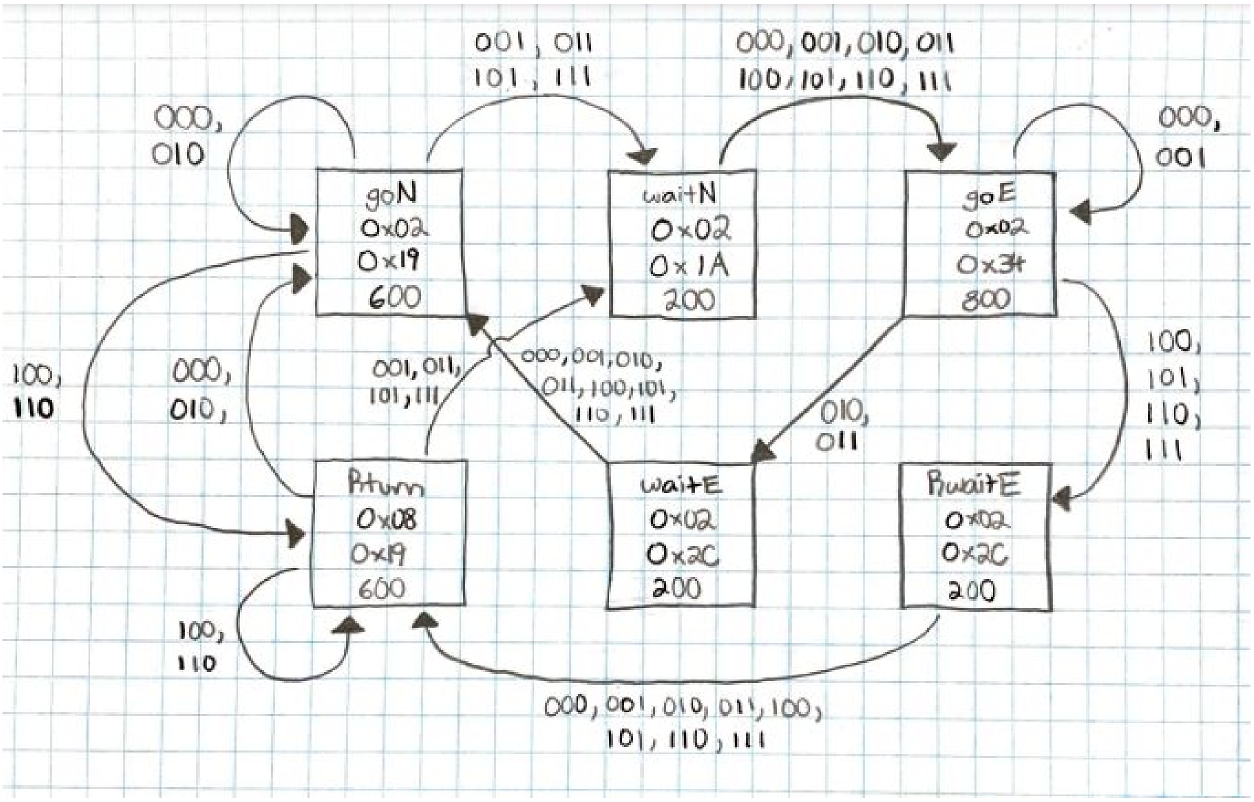
Nicholas Bishop, Darren Cobian

State Diagrams

State Table

State	Input							
	000	**001**	**010**	**011**	**100**	**101**	**110**	**111**
goN (0x02, 0x19, 600)	goN	waitN	goN	waitN	Rturn	waitN	Rturn	waitN
waitN (0x02, 0x1A, 200)	goE	goE	goE	goE	goE	goE	goE	goE
goE (0x02, 0x34, 800)	goE	goE	waitE	waitE	RwaitE	RwaitE	RwaitE	RwaitE
waitE (0x02, 0x2C, 200)	goN	goN	goN	goN	goN	goN	goN	goN
Rturn (0x08, 0x19, 600)	goN	waitN	goN	waitN	Rturn	waitN	Rturn	waitN
RwaitE (0x02, 0x2C, 200	Rturn	Rturn	Rturn	Rturn	Rturn	Rturn	Rturn	Rturn

State Graph


goN
0x02
0x19
600
waitN
0x02
0x1A
200
goE
0x02
0x34
800
Rturn
0x08
0x19
600
waitE
0x02
0x2C
200
RwaitE
0x02
0x2C
200
000,
010
001, 011
101, 111
000, 001, 010, 011
100, 101, 110, 111
000,
001
100,
110
000,
010,
001, 011,
101, 111
000, 001, 010,
011, 100, 101,
110, 111
010,
011
100,
101,
110,
111
100,
110
000, 001, 010, 011, 100,
101, 110, 111

Microprocessor + Project 1 Implementation on Breadboard

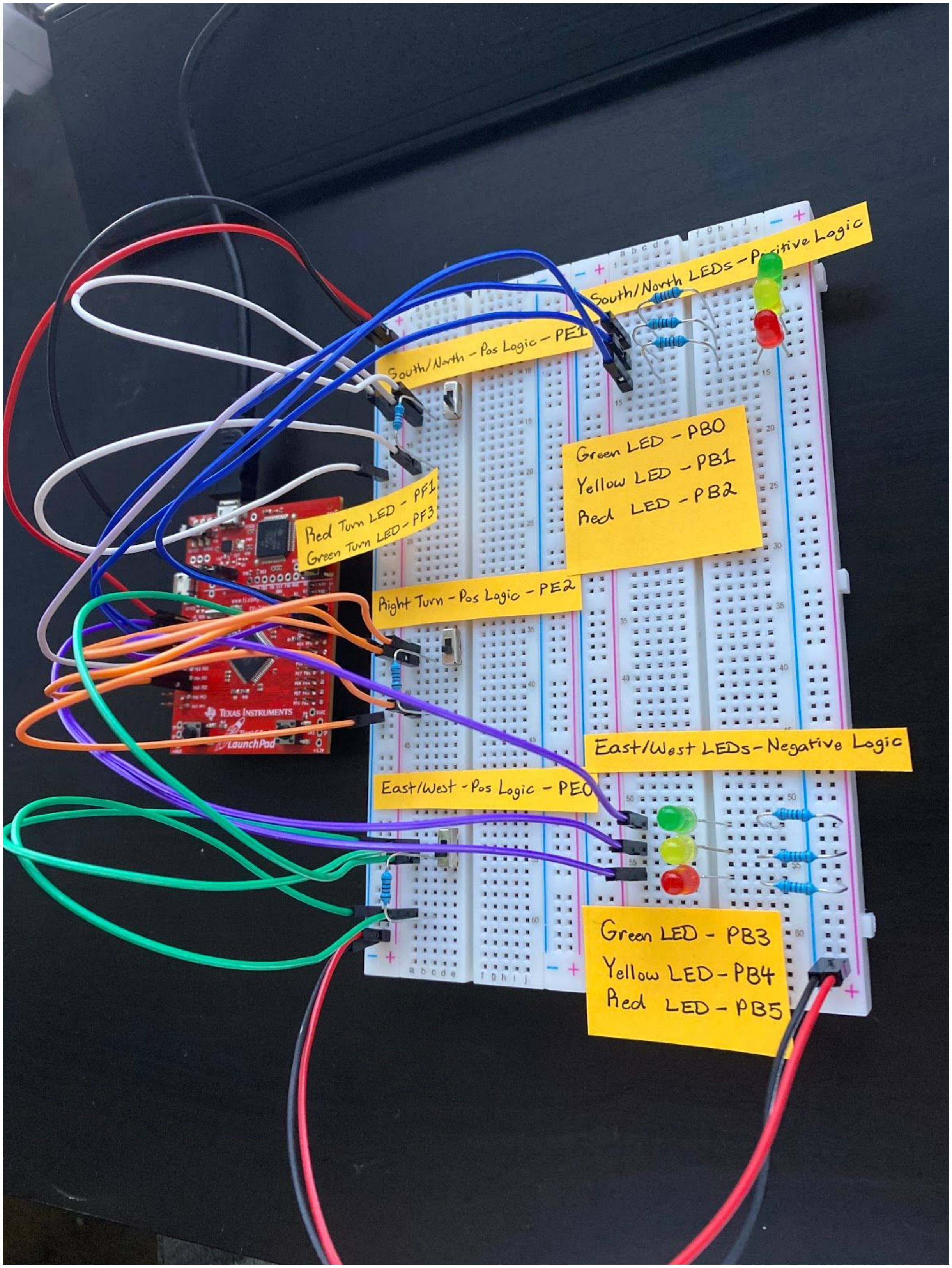
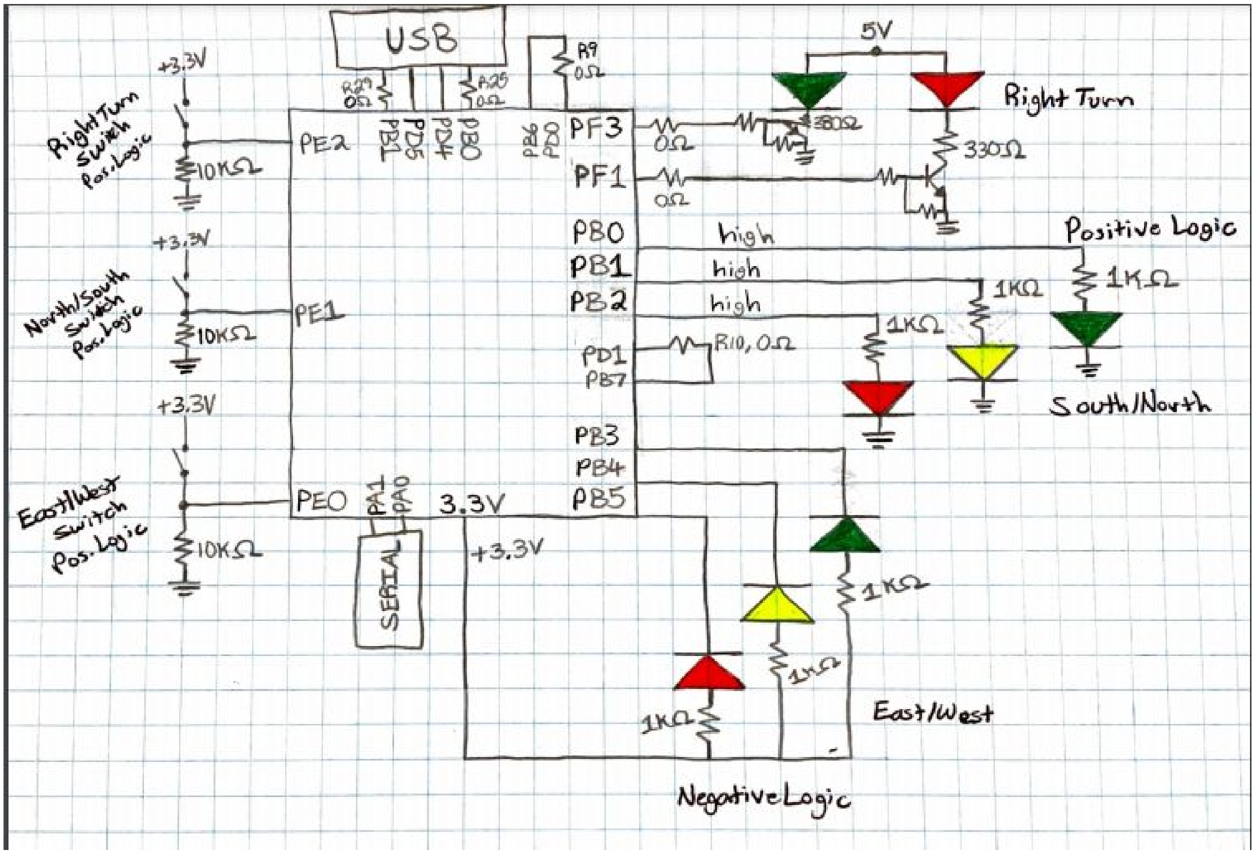
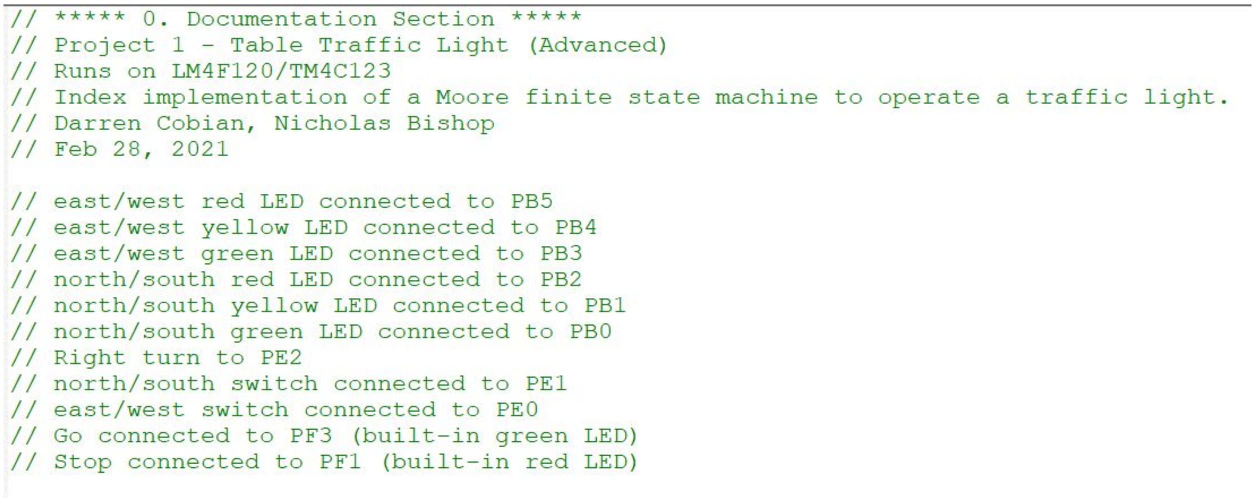


Diagram of Hardware System

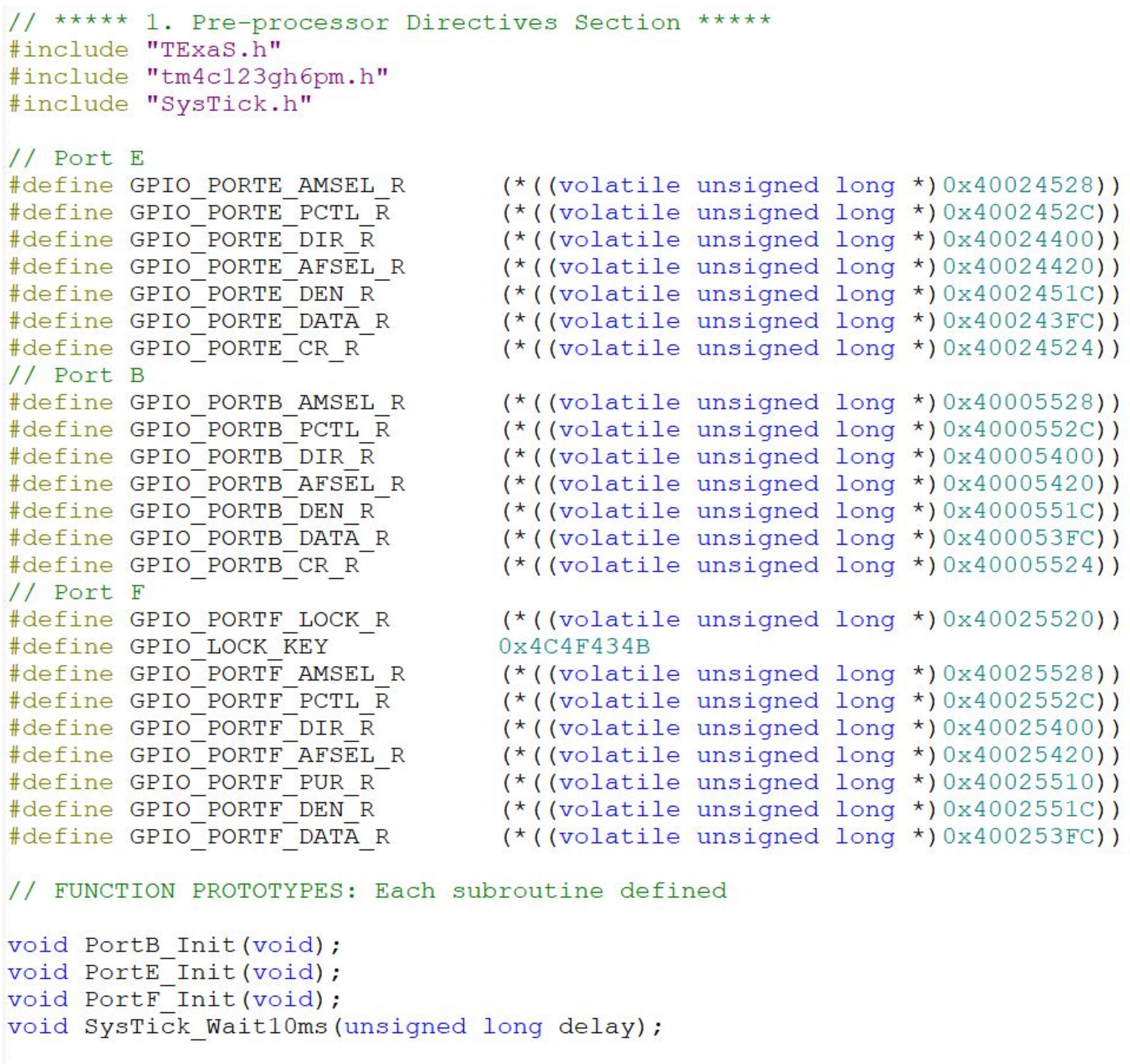


Software Source Code

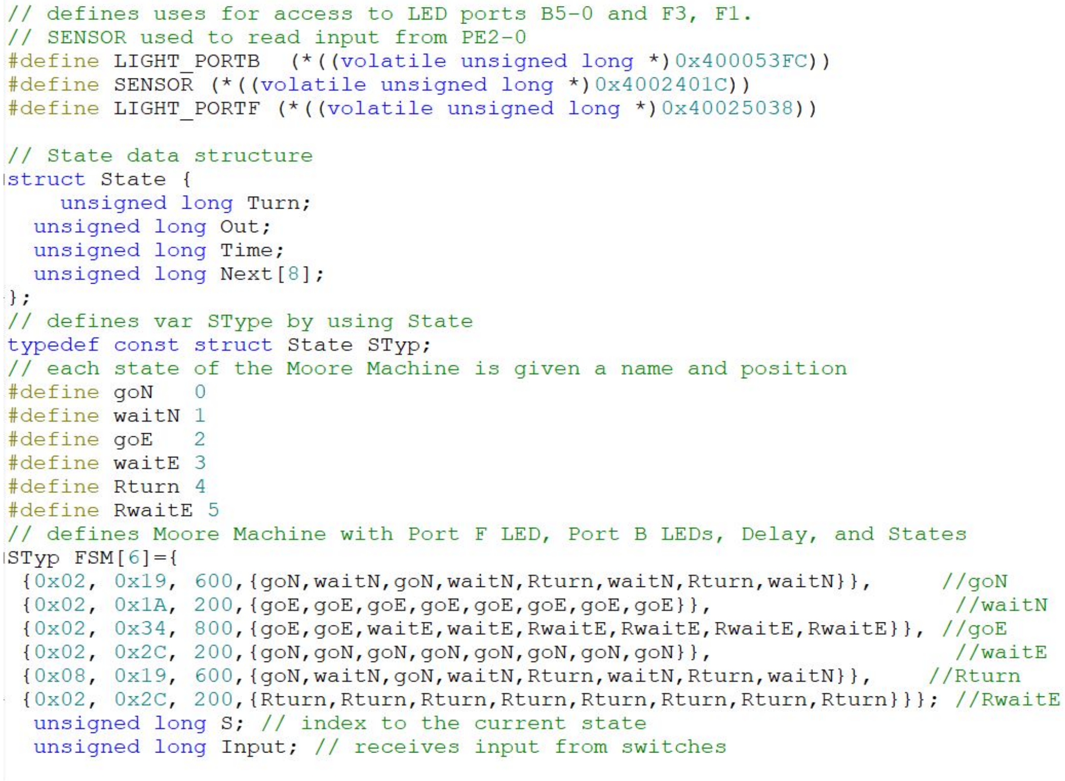
Project1.c



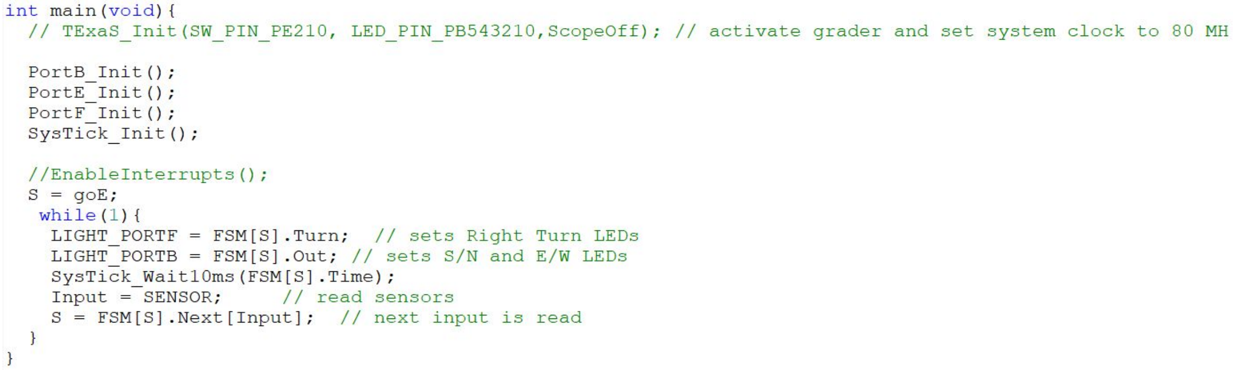
```
// ***** 0. Documentation Section *****
// Project 1 - Table Traffic Light (Advanced)
// Runs on LM4F120/TM4C123
// Index implementation of a Moore finite state machine to operate a traffic light.
// Darren Cobian, Nicholas Bishop
// Feb 28, 2021

// east/west red LED connected to PB5
// east/west yellow LED connected to PB4
// east/west green LED connected to PB3
// north/south red LED connected to PB2
// north/south yellow LED connected to PB1
// north/south green LED connected to PB0
// Right turn to PE2
// north/south switch connected to PE1
// east/west switch connected to PE0
// Go connected to PF3 (built-in green LED)
// Stop connected to PF1 (built-in red LED)
```



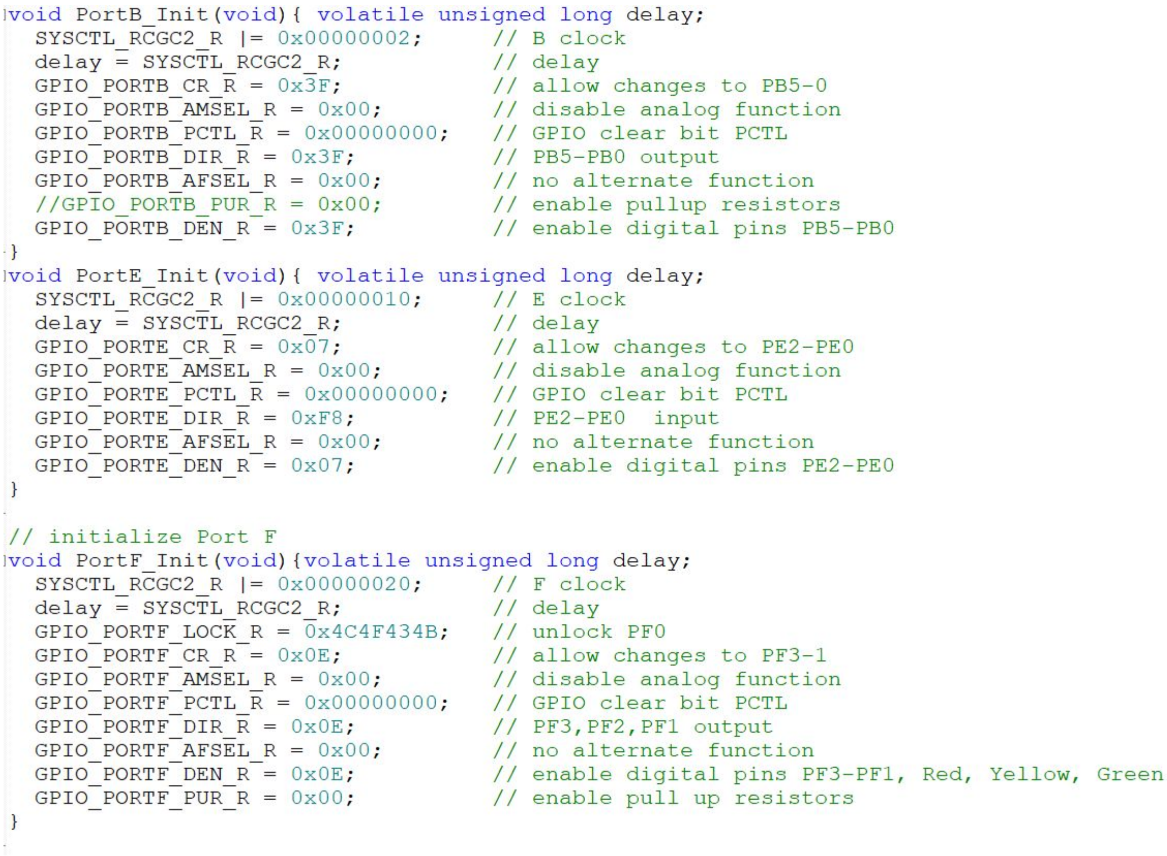
```c
// ***** 1. Pre-processor Directives Section *****
#include "TExaS.h"
#include "tm4c123gh6pm.h"
#include "SysTick.h"

// Port E
#define GPIO_PORTE_AMSEL_R      (*((volatile unsigned long *)0x40024528))
#define GPIO_PORTE_PCTL_R       (*((volatile unsigned long *)0x4002452C))
#define GPIO_PORTE_DIR_R        (*((volatile unsigned long *)0x40024400))
#define GPIO_PORTE_AFSEL_R      (*((volatile unsigned long *)0x40024420))
#define GPIO_PORTE_DEN_R        (*((volatile unsigned long *)0x4002451C))
#define GPIO_PORTE_DATA_R       (*((volatile unsigned long *)0x400243FC))
#define GPIO_PORTE_CR_R         (*((volatile unsigned long *)0x40024524))
// Port B
#define GPIO_PORTB_AMSEL_R      (*((volatile unsigned long *)0x40005528))
#define GPIO_PORTB_PCTL_R       (*((volatile unsigned long *)0x4000552C))
#define GPIO_PORTB_DIR_R        (*((volatile unsigned long *)0x40005400))
#define GPIO_PORTB_AFSEL_R      (*((volatile unsigned long *)0x40005420))
#define GPIO_PORTB_DEN_R        (*((volatile unsigned long *)0x4000551C))
#define GPIO_PORTB_DATA_R       (*((volatile unsigned long *)0x400053FC))
#define GPIO_PORTB_CR_R         (*((volatile unsigned long *)0x40005524))
// Port F
#define GPIO_PORTF_LOCK_R       (*((volatile unsigned long *)0x40025520))
#define GPIO_LOCK_KEY           0x4C4F434B
#define GPIO_PORTF_AMSEL_R      (*((volatile unsigned long *)0x40025528))
#define GPIO_PORTF_PCTL_R       (*((volatile unsigned long *)0x4002552C))
#define GPIO_PORTF_DIR_R        (*((volatile unsigned long *)0x40025400))
#define GPIO_PORTF_AFSEL_R      (*((volatile unsigned long *)0x40025420))
#define GPIO_PORTF_PUR_R        (*((volatile unsigned long *)0x40025510))
#define GPIO_PORTF_DEN_R        (*((volatile unsigned long *)0x4002551C))
#define GPIO_PORTF_DATA_R       (*((volatile unsigned long *)0x400253FC))

// FUNCTION PROTOTYPES: Each subroutine defined

void PortB_Init(void);
void PortE_Init(void);
void PortF_Init(void);
void SysTick_Wait10ms(unsigned long delay);
```



```c
// defines uses for access to LED ports B5-0 and F3, F1.
// SENSOR used to read input from PE2-0
#define LIGHT_PORTB  (*((volatile unsigned long *)0x400053FC))
#define SENSOR (*((volatile unsigned long *)0x4002401C))
#define LIGHT_PORTF (*((volatile unsigned long *)0x40025038))

// State data structure
struct State {
    unsigned long Turn;
  unsigned long Out;
  unsigned long Time;
  unsigned long Next[8];
};
// defines var SType by using State
typedef const struct State STyp;
// each state of the Moore Machine is given a name and position
#define goN   0
#define waitN 1
#define goE   2
#define waitE 3
#define Rturn 4
#define RwaitE 5
// defines Moore Machine with Port F LED, Port B LEDs, Delay, and States
STyp FSM[6]={
 {0x02, 0x19, 600,{goN,waitN,goN,waitN,Rturn,waitN,Rturn,waitN}},     //goN
 {0x02, 0x1A, 200,{goE,goE,goE,goE,goE,goE,goE,goE}},                  //waitN
 {0x02, 0x34, 800,{goE,goE,waitE,waitE,RwaitE,RwaitE,RwaitE,RwaitE}}, //goE
 {0x02, 0x2C, 200,{goN,goN,goN,goN,goN,goN,goN,goN}},                  //waitE
 {0x08, 0x19, 600,{goN,waitN,goN,waitN,Rturn,waitN,Rturn,waitN}},    //Rturn
 {0x02, 0x2C, 200,{Rturn,Rturn,Rturn,Rturn,Rturn,Rturn,Rturn,Rturn}}}; //RwaitE
  unsigned long S; // index to the current state
  unsigned long Input; // receives input from switches
```



```c
int main(void){
  // TExaS_Init(SW_PIN_PE210, LED_PIN_PB543210,ScopeOff); // activate grader and set system clock to 80 MH

  PortB_Init();
  PortE_Init();
  PortF_Init();
  SysTick_Init();

  //EnableInterrupts();
  S = goE;
   while(1){
    LIGHT_PORTF = FSM[S].Turn;  // sets Right Turn LEDs
    LIGHT_PORTB = FSM[S].Out; // sets S/N and E/W LEDs
    SysTick_Wait10ms(FSM[S].Time);
    Input = SENSOR;     // read sensors
    S = FSM[S].Next[Input];  // next input is read
  }
}
```



```c
void PortB_Init(void){ volatile unsigned long delay;
  SYSCTL_RCGC2_R |= 0x00000002;     // B clock
  delay = SYSCTL_RCGC2_R;           // delay
  GPIO_PORTB_CR_R = 0x3F;           // allow changes to PB5-0
  GPIO_PORTB_AMSEL_R = 0x00;        // disable analog function
  GPIO_PORTB_PCTL_R = 0x00000000;   // GPIO clear bit PCTL
  GPIO_PORTB_DIR_R = 0x3F;          // PB5-PB0 output
  GPIO_PORTB_AFSEL_R = 0x00;        // no alternate function
  //GPIO_PORTB_PUR_R = 0x00;        // enable pullup resistors
  GPIO_PORTB_DEN_R = 0x3F;          // enable digital pins PB5-PB0
}
void PortE_Init(void){ volatile unsigned long delay;
  SYSCTL_RCGC2_R |= 0x00000010;     // E clock
  delay = SYSCTL_RCGC2_R;           // delay
  GPIO_PORTE_CR_R = 0x07;           // allow changes to PE2-PE0
  GPIO_PORTE_AMSEL_R = 0x00;        // disable analog function
  GPIO_PORTE_PCTL_R = 0x00000000;   // GPIO clear bit PCTL
  GPIO_PORTE_DIR_R = 0xF8;          // PE2-PE0  input
  GPIO_PORTE_AFSEL_R = 0x00;        // no alternate function
  GPIO_PORTE_DEN_R = 0x07;          // enable digital pins PE2-PE0
}

// initialize Port F
void PortF_Init(void){volatile unsigned long delay;
  SYSCTL_RCGC2_R |= 0x00000020;     // F clock
  delay = SYSCTL_RCGC2_R;           // delay
  GPIO_PORTF_LOCK_R = 0x4C4F434B;   // unlock PF0
  GPIO_PORTF_CR_R = 0x0E;           // allow changes to PF3-1
  GPIO_PORTF_AMSEL_R = 0x00;        // disable analog function
  GPIO_PORTF_PCTL_R = 0x00000000;   // GPIO clear bit PCTL
  GPIO_PORTF_DIR_R = 0x0E;          // PF3,PF2,PF1 output
  GPIO_PORTF_AFSEL_R = 0x00;        // no alternate function
  GPIO_PORTF_DEN_R = 0x0E;          // enable digital pins PF3-PF1, Red, Yellow, Green
  GPIO_PORTF_PUR_R = 0x00;          // enable pull up resistors
}
```