Project 2: Simple Smart House

CECS 346 Aaron Mai 016651243

Introduction:

In this lab we are to build a simplified smart house with the following automatic control. The smart house will simulate a garage door and have three onboard LEDS to indicate the door status open/close/moving with blue,green, and flashing red respectively. The door will be powered by a full step drive stepper motor/IR sensor and will implement systick modules as well as interrupts.

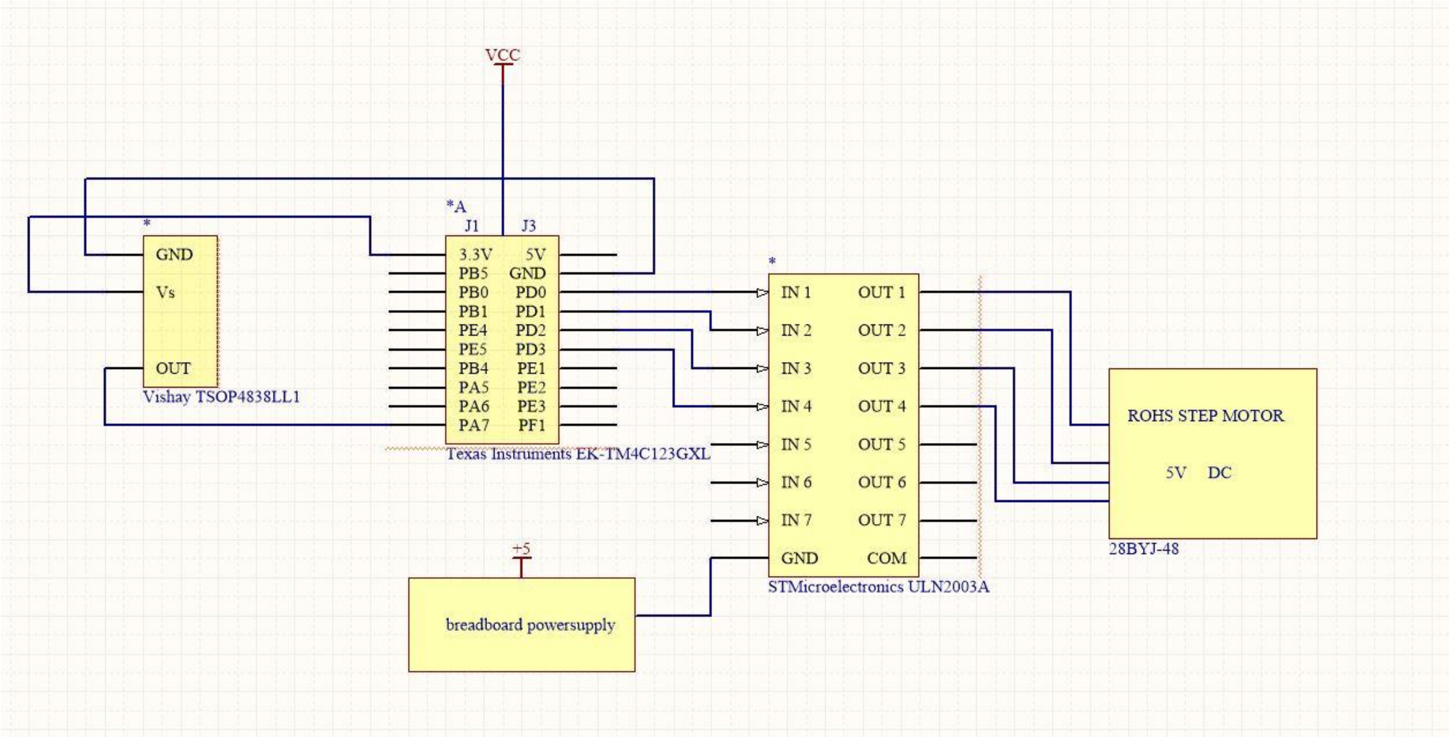
Operation - Link to Video Description:

https://photos.app.goo.gl/W6fBD6sLq8gpQuJQ8

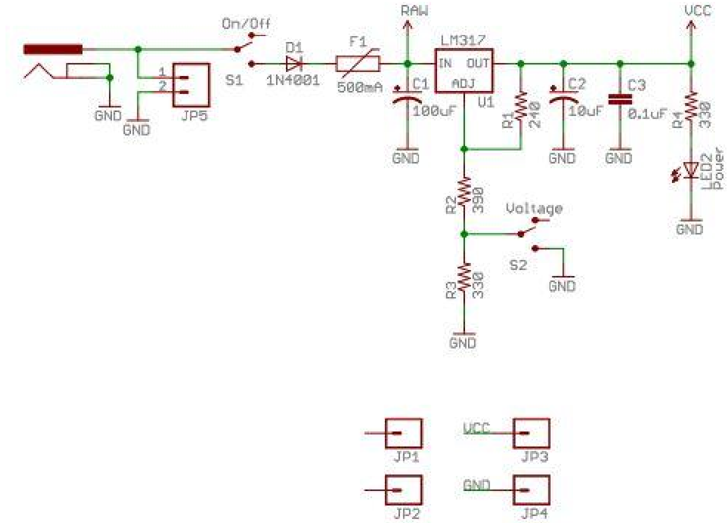
Theory

In this project the automatic garage door is very applicable to the real world. Newer garage doors use sensors to detect incoming or leaving cars and open or close them automatically for convenience's sake. Within our design we used LEDS to indicate the door status open/close/moving with blue,green, and flashing red respectively. We also used interrupts and systick timers to determine the operation performed when an input is received either through the sensor or on the onboard switches.

Hardware Design: Circuit Diagram

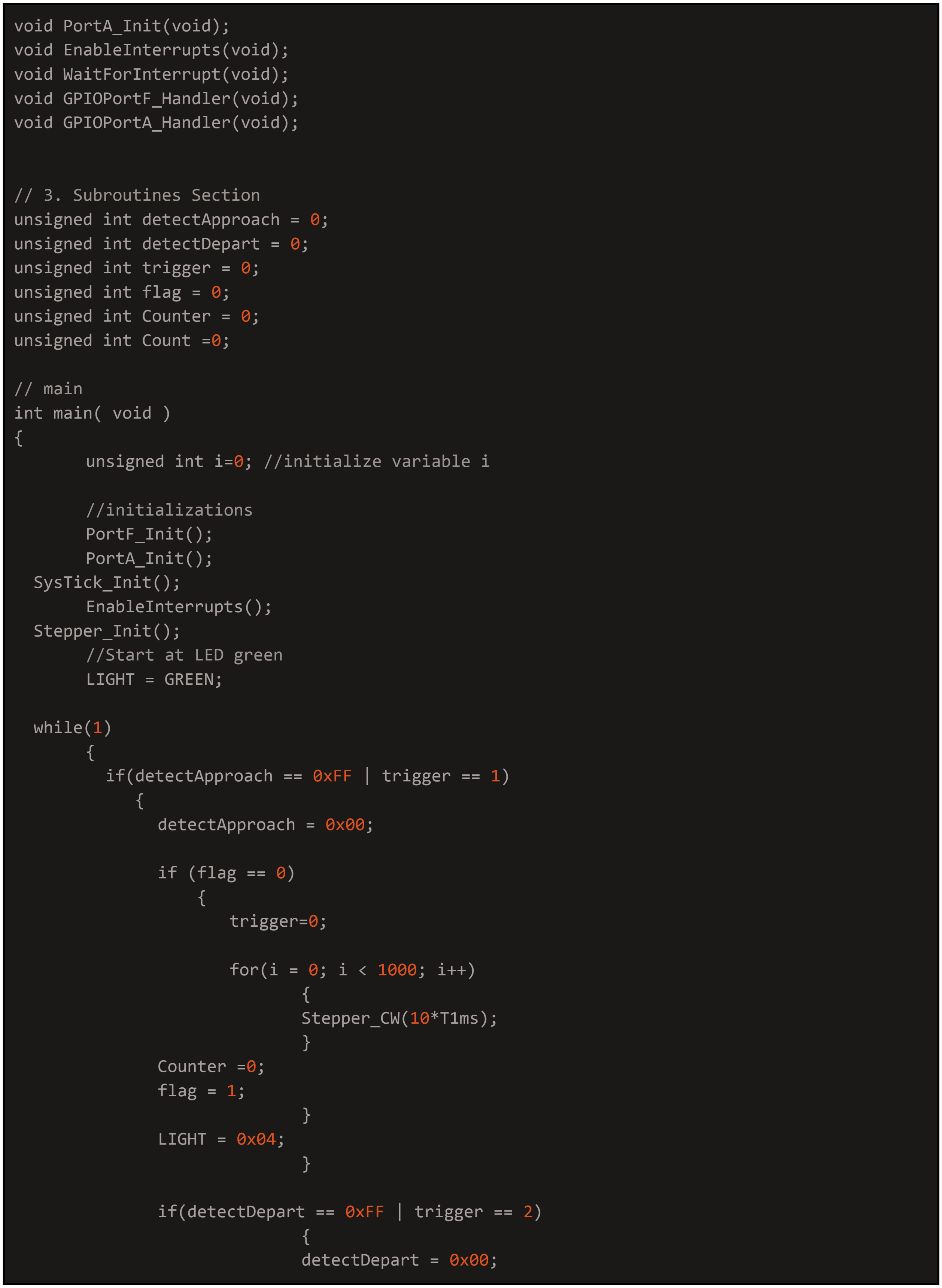


Breadboard power supply circuit diagram

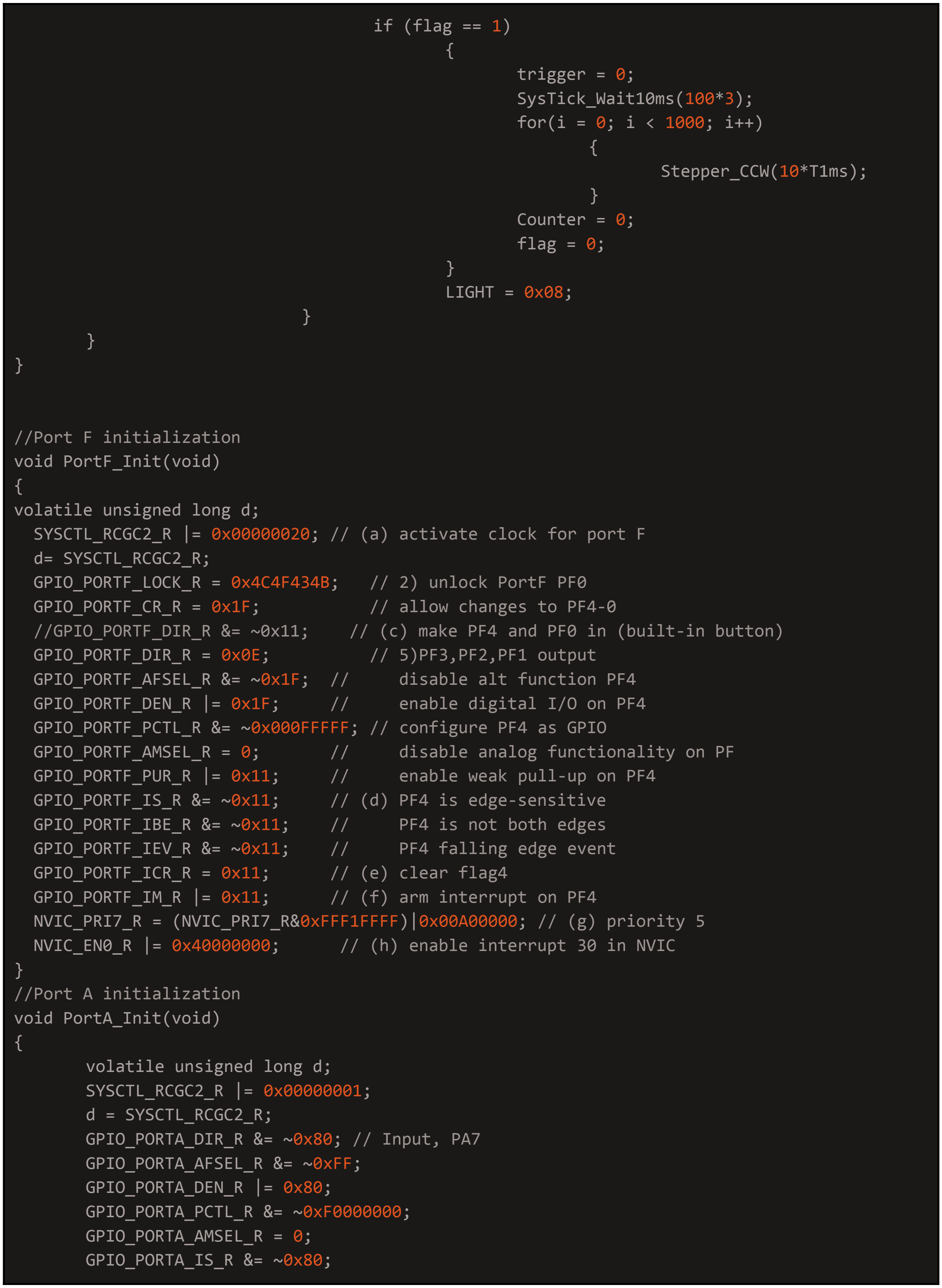


Source Code

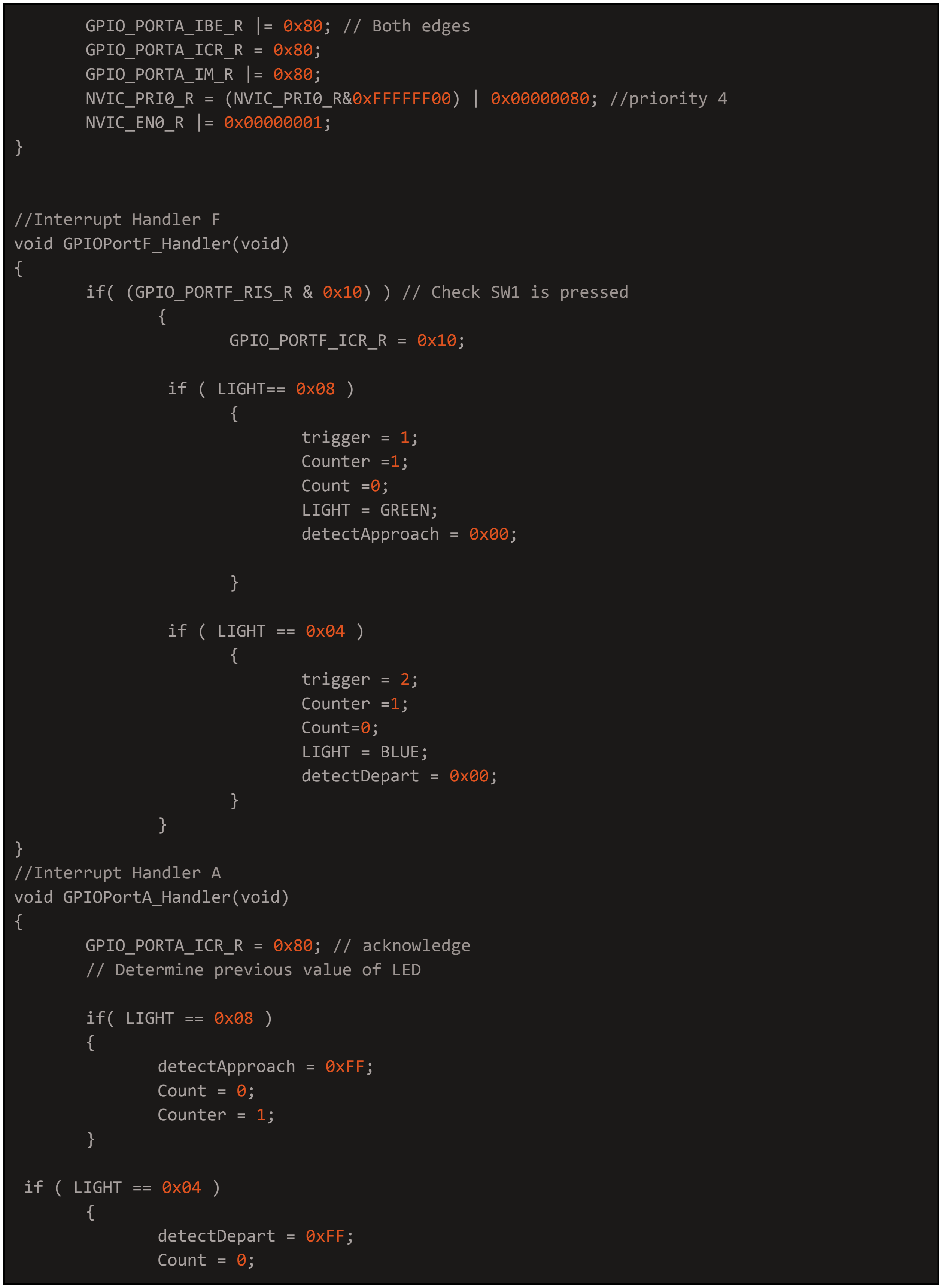
```
// Documentation
// CECS346 Project 2 - A Simple Smart Home
// Description: In this lab we will be building on our previous IR sensor lab.
//In this lab we will be building an embedded design where we construct a simplified smart
house.
//The smart house will simulate a garage door and have three onboard LEDS to indicate the
door status
//open/close/opening+closing with blue,green, and flashing red respectively.
// The door will be powered by a ROHS full step drive stepper motor and will implement
systick modules as well as interrupts.

// I will be using ports F(onboard LEDS + sw1) and A(IR sensor input)
// Student Name: Aaron Mai 016651243

//Input + Output:
// PA7 - IR sensor input
// PF3/PF2/PF3 - LED output
// PF0 - Garage door open/close switch
// PD3 connected to driver for stepper motor coil A
// PD2 connected to driver for stepper motor coil A'
// PD1 connected to driver for stepper motor coil B
// PD0 connected to driver for stepper motor coil B'

//Preprocessor Directives
#include <stdint.h>
#include "stepper.h"
#include "SysTick.h"
#include "tm4c123gh6pm.h"

//// 1. Preprocessor Directives Section
// Constant declarations to access port registers using
// symbolic names instead of addresses
#define GPIO_PORTF_DATA_R       (*((volatile unsigned long *)0x400253FC))
#define LIGHT                (*((volatile unsigned long *)0x40025038))

//define constants + aliases
#define RED 0x02;
#define BLUE   0x04;
#define GREEN 0x08;
#define T1ms 16000


// 2. Declarations Section
//   Function Prototypes
void PortF_Init(void);
```



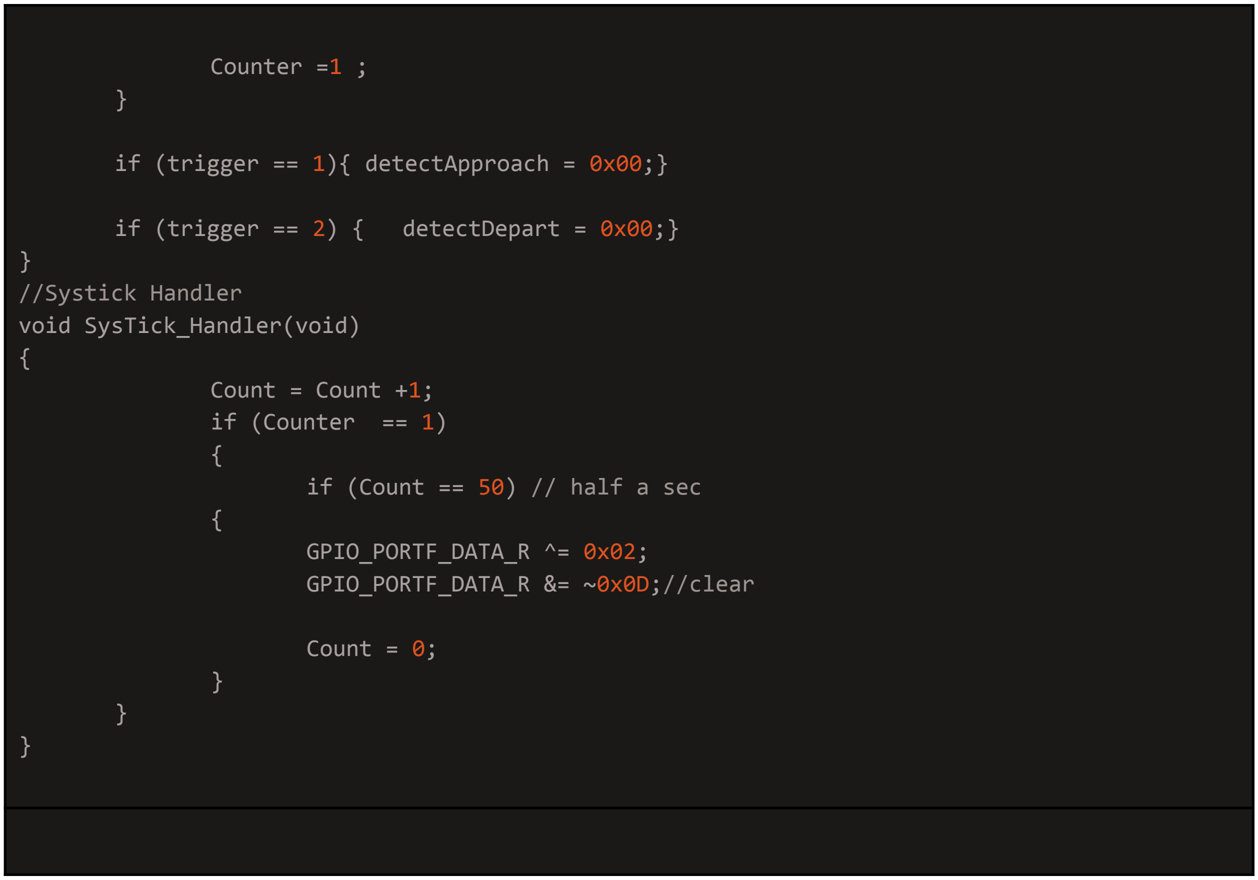
```
void PortA_Init(void);
void EnableInterrupts(void);
void WaitForInterrupt(void);
void GPIOPortF_Handler(void);
void GPIOPortA_Handler(void);


// 3. Subroutines Section
unsigned int detectApproach = 0;
unsigned int detectDepart = 0;
unsigned int trigger = 0;
unsigned int flag = 0;
unsigned int Counter = 0;
unsigned int Count =0;

// main
int main( void )
{
       unsigned int i=0; //initialize variable i

       //initializations
       PortF_Init();
       PortA_Init();
  SysTick_Init();
       EnableInterrupts();
  Stepper_Init();
       //Start at LED green
       LIGHT = GREEN;

  while(1)
       {
         if(detectApproach == 0xFF | trigger == 1)
            {
               detectApproach = 0x00;

               if (flag == 0)
                   {
                      trigger=0;

                      for(i = 0; i < 1000; i++)
                             {
                             Stepper_CW(10*T1ms);
                             }
               Counter =0;
               flag = 1;
                             }
               LIGHT = 0x04;
                             }

               if(detectDepart == 0xFF | trigger == 2)
                             {
                             detectDepart = 0x00;
```



```
                                        if (flag == 1)
                                                {
                                                        trigger = 0;
                                                        SysTick_Wait10ms(100*3);
                                                        for(i = 0; i < 1000; i++)
                                                                {
                                                                        Stepper_CCW(10*T1ms);
                                                                }
                                                        Counter = 0;
                                                        flag = 0;
                                                }
                                                LIGHT = 0x08;
                                }
        }
}


//Port F initialization
void PortF_Init(void)
{
volatile unsigned long d;
  SYSCTL_RCGC2_R |= 0x00000020; // (a) activate clock for port F
  d= SYSCTL_RCGC2_R;
  GPIO_PORTF_LOCK_R = 0x4C4F434B;   // 2) unlock PortF PF0
  GPIO_PORTF_CR_R = 0x1F;           // allow changes to PF4-0
  //GPIO_PORTF_DIR_R &= ~0x11;    // (c) make PF4 and PF0 in (built-in button)
  GPIO_PORTF_DIR_R = 0x0E;          // 5)PF3,PF2,PF1 output
  GPIO_PORTF_AFSEL_R &= ~0x1F;  //     disable alt function PF4
  GPIO_PORTF_DEN_R |= 0x1F;     //     enable digital I/O on PF4
  GPIO_PORTF_PCTL_R &= ~0x000FFFFF; // configure PF4 as GPIO
  GPIO_PORTF_AMSEL_R = 0;       //     disable analog functionality on PF
  GPIO_PORTF_PUR_R |= 0x11;     //     enable weak pull-up on PF4
  GPIO_PORTF_IS_R &= ~0x11;     // (d) PF4 is edge-sensitive
  GPIO_PORTF_IBE_R &= ~0x11;    //     PF4 is not both edges
  GPIO_PORTF_IEV_R &= ~0x11;    //     PF4 falling edge event
  GPIO_PORTF_ICR_R = 0x11;      // (e) clear flag4
  GPIO_PORTF_IM_R |= 0x11;      // (f) arm interrupt on PF4
  NVIC_PRI7_R = (NVIC_PRI7_R&0xFFF1FFFF)|0x00A00000; // (g) priority 5
  NVIC_EN0_R |= 0x40000000;      // (h) enable interrupt 30 in NVIC
}
//Port A initialization
void PortA_Init(void)
{
        volatile unsigned long d;
        SYSCTL_RCGC2_R |= 0x00000001;
        d = SYSCTL_RCGC2_R;
        GPIO_PORTA_DIR_R &= ~0x80; // Input, PA7
        GPIO_PORTA_AFSEL_R &= ~0xFF;
        GPIO_PORTA_DEN_R |= 0x80;
        GPIO_PORTA_PCTL_R &= ~0xF0000000;
        GPIO_PORTA_AMSEL_R = 0;
        GPIO_PORTA_IS_R &= ~0x80;
```



```
        GPIO_PORTA_IBE_R |= 0x80; // Both edges
        GPIO_PORTA_ICR_R = 0x80;
        GPIO_PORTA_IM_R |= 0x80;
        NVIC_PRI0_R = (NVIC_PRI0_R&0xFFFFFF00) | 0x00000080; //priority 4
        NVIC_EN0_R |= 0x00000001;
}


//Interrupt Handler F
void GPIOPortF_Handler(void)
{
        if( (GPIO_PORTF_RIS_R & 0x10) ) // Check SW1 is pressed
                {
                        GPIO_PORTF_ICR_R = 0x10;

                 if ( LIGHT== 0x08 )
                        {
                                trigger = 1;
                                Counter =1;
                                Count =0;
                                LIGHT = GREEN;
                                detectApproach = 0x00;

                        }

                 if ( LIGHT == 0x04 )
                        {
                                trigger = 2;
                                Counter =1;
                                Count=0;
                                LIGHT = BLUE;
                                detectDepart = 0x00;
                        }
                }
}
//Interrupt Handler A
void GPIOPortA_Handler(void)
{
        GPIO_PORTA_ICR_R = 0x80; // acknowledge
        // Determine previous value of LED

        if( LIGHT == 0x08 )
        {
                detectApproach = 0xFF;
                Count = 0;
                Counter = 1;
        }

 if ( LIGHT == 0x04 )
        {
                detectDepart = 0xFF;
                Count = 0;
```



```
                Counter =1 ;
        }

        if (trigger == 1){ detectApproach = 0x00;}

        if (trigger == 2) {   detectDepart = 0x00;}
}
//Systick Handler
void SysTick_Handler(void)
{
                Count = Count +1;
                if (Counter  == 1)
                {
                        if (Count == 50) // half a sec
                {
                        GPIO_PORTF_DATA_R ^= 0x02;
                        GPIO_PORTF_DATA_R &= ~0x0D;//clear

                        Count = 0;
                }
        }
}
```

Conclusion

To conclude the project has been a culmination of everything we learned this semester. We used things such as using interrupts to systick to interfacing and implementing stepper motors and onboard LEDS. The most problems I had were implementing interrupts but after reviewing the interrupt lecture slides I eventually got it. I learned alot doing this project and am super thankful that professor Min He gave me a board when my original TM4C123 board burnt out.