CECS 346 Spring 2018 Final Project

Autonomous Stepper Robot & Smart Home

Seungjun Jeon Jonic Mecija

May 18, 2018

This project implements the use of an obstacle avoidance sensor and stepper motor to create an autonomous stepper robot and smart home.

Introduction

In this project we implemented two IR sensors and three stepper motors to create an autonomous stepper robot and a smart garage that can be integrated to work together. The robot has two modes which are activated by the on board buttons. The first mode goes forward, turns left, and detects for the garage door so it can wait for the garage to open. The second mode goes backward out of the garage, turns right, and goes forward again. The robot is powered by a power 4 pack of double A batteries that is connected to the ARM board. The garage also has two modes in which we open and close the garage depending if the sensor is triggered. With the use of edge triggered interrupts implemented in the IR sensor and the buttons, we control how the robot and garage behaves. The overall goal of this project is to utilize the interrupts and stepper motor to simulate an automatic garage and autonomous car.

Operation

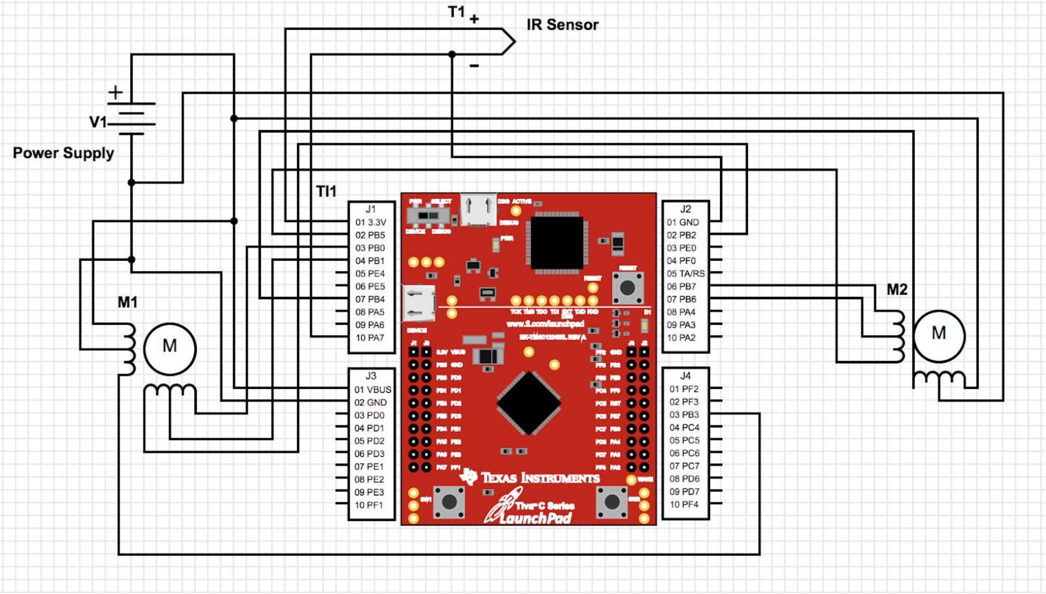
The car has two modes using the two onboard buttons. Pressing the first button will make the car go forward 720 degrees, then the car will turn left 90 degrees then proceeds to move forward until it detects an obstacle and finally the car will move 360 degrees forwards and stop. Pressing the second button will make the car go backwards 720 degrees then turn right 90 degrees and move forward 360 degrees and stop.The IR sensors will detect an obstacle in the way and if it does, the green LED will be on and the garage door will open. As it opens, the LED will flash red and change to blue. If the IR sensors detect an obstacle leaving then the blue LED will be on and the garage door will close. As it closes, the red LEDs will flash and change to green. If there is no obstacle and an onboard button is pressed, based on whether the garage door is open or closed, the garage door will open if it was initially closed or close if it was initially open. The door will open if the car gets close enough to the IR sensor, triggering the interrupt to occur. Then the car will move forward and stop. Then the button on the home is pressed to open the garage door and the second button to make the car go backward is pushed. Then the IR sensor is triggered because the car gets close enough and closes the garage door. Then the car rotates to the right and moves forward.

Theory

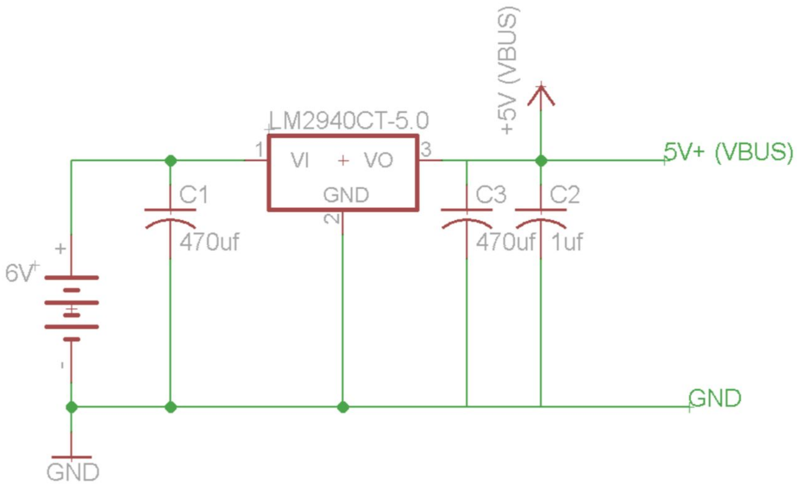
This project applies to real world applications because many people use automatic garage door opener for their homes. Garage doors use sensors to detect incoming obstacles or leaving obstacles to determine whether to open the garage door or to close the garage door. The LEDs are used to show whether the garage door is opening or closing. The LED flashing is an indication that the garage is opening or closing. Using interrupts and systick timer, we determine the operation it performs when an input is received. Using two systick timers in our project, one for the motors, and another for systick interrupt for the flashing LEDs, we can use control the rate of the flashing and how fast the motor will move. For this project, we used 50 ms for the flashing and 10ms for the motor speed.To make the car go forward, the right motor needs to rotate clockwise while the left motor needs to rotate counter clockwise. To make the car turn to the left, both motors needs to rotate clockwise and to make the car turn right, both motors needs to rotate counter clockwise. To make the car go backwards, the right motor needs to rotate counter clockwise and the left motor needs to move clockwise. The car only stops after its operations are complete or when the IR sensor senses an obstacle causing an edge-interrupt to stop the car.

Hardware

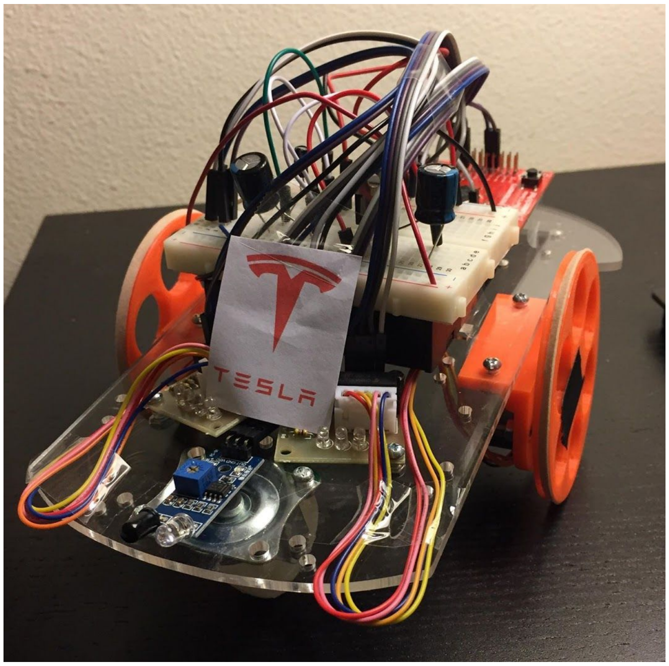
Microcontroller Connection Schematic:



Power Supply Schematic:



Stepper Robot Picture:



Video Demo Link:

https://www.youtube.com/watch?v=ACqIHwcikMs

Software

Car:

Stepper_Init(): Initializes Port B for the stepper motor.

Main(): The module that contains all the logic that will be implemented

GPIO_PortF_Handler(): If sw1 is pressed, sw1 becomes a 1 and sw2 becomes 0. If sw2 is pressed, sw2 becomes a 1 and sw1 is a 0

GPIO_PortA_Handler(): If the IR Sensors detects an obstacle, the variable that allows the car to keep moving becomes a 0 and a stop variable becomes a 1 to stop the car and once the IR Sensor detects no obstacle then the car will move forward 360 degrees then stop.

Stepper_Forward(delay): The car will move forward by 0.18 degrees

Stepper_Left(delay): The car will turn left once by 0.18 degrees

Stepper_Right(delay): The car will turn right once by 0.18 degrees

Stepper_Back(delay): The car will go backwards once by 0.18 degrees

Garage:

PortF_Init(), PortA_Init(), SysTick_Init(): Initializing the ports and systick.

GPIOPORTF_Handler(): The ISR for port f, this is used for when the onboard button is pressed.

GPIOPORTA_Handler(): The ISR for port A, this is used when the IR sensor detects an obstacle the LED will change colors.

SysTick_Handler(): The ISR for systick. This is used for the red flashing LEDs.

Stepper_CW(): The motor will move 0.18 degrees once clockwise.

Stepper_CCW(): The motor will move 0.18 degrees once counterclockwise.

Conclusion

Overall this project was a great learning experience in terms of learning how to work with the IR sensor, interrupts, and stepper motors. The part of the project that we spent the most time on was testing the timing between opening the garage for the car to enter and the amount of time for the car to wait. We overcame this problem by constantly testing out different timings for the car and garage speed. The use of previous labs and example code also benefited the project. We were able to base the project on previous knowledge learned from before. After testing out and confirming the code works on the board and stepper robot in relation to the garage, it was satisfying to finally get it working. Assignments with applicable knowledge and tangible project parts are the most interesting and fun. This project was an example of just that.

Car Code:

```
// FUNCTION DECLARATIONS
void PortF_Init(void);
void PortA_Init(void);
void EnableInterrupts(void);
void WaitForInterrupt(void);
#define T1ms 3000

// VARIABLE DECLARATIONS
unsigned int flag = 0;
unsigned int last = 0;
unsigned int stop = 0;
unsigned int sw1 = 0;
unsigned int sw2 = 0;
unsigned int go = 1;
unsigned int i = 0;

// MAIN
int main( void )
{
        // PORT INITILIZATIONS
        PortF_Init();
        PortA_Init();
        EnableInterrupts();
  Stepper_Init();

        // SUPER LOOP
        while ( 1 )
        {
                // ********* SWITCH ONE PROCESSES ********* //
                if ( sw1 == 1 )
                        {
                                if( flag == 1 )
                                        {
                                                // GO FORWARD 720 DEGREES
                                                for ( i = 0; i < 4000; i++ )
                                                        {
                                                                Stepper_Forward(10*T1ms);
                                                        }
                                                        SysTick_Wait10ms(5);

                                                // TURN RIGHT 90 DEGREES
                                                for ( i = 0; i < 900; i++ )
```

```
                    {
                        Stepper_Left(10*T1ms);
                    }
                    SysTick_Wait10ms(5);

                flag = 0;
            }

        // GO FORWARD
        if ( go == 1 )
            {
                Stepper_Forward(10*T1ms);

                // STOP IF OBJECT DETECTED
                if( stop == 1 )
                    {
                        SysTick_Wait10ms(5);
                        go = 0;
                    }
            }
        // IF OBJECT IS REMOVED
        if ( last == 1 )
            {
                SysTick_Wait10ms(100*15);
                // MOVE FORWARD 360 DEGREES
                for ( i = 0; i < 2500; i++ )
                    {
                        Stepper_Forward(10*T1ms);
                    }

                // STOP
                // RESET FLAGS
                SysTick_Wait10ms(5);
                last = 0;
                sw1 = 0;
            }
    }

// ********  SWITCH TWO PROCESSES ********* //
if ( sw2 == 1 )
    {
        // REVERSE 360 DEGREES
        for( i = 0; i < 2500; i++ )
            {
                Stepper_Back(10*T1ms);
```

```
                    }
                SysTick_Wait10ms(5);

                // TURN RIGHT 90 DEGREES
                for( i = 0; i < 900; i++ )
                    {
                        Stepper_Right(10*T1ms);
                    }
                SysTick_Wait10ms(5);

                // MOVE FORWARD 720 DEGREES
                for( i = 0; i < 4000; i++ )
                    {
                        Stepper_Forward(10*T1ms);
                    }

                // STOP
                SysTick_Wait10ms(5);
                sw2 = 0;
            }
    }
}

// SWITCH INTERRUPT
void GPIOPortF_Handler(void)
{
    // IF SW 1 IS PRESSED TRIGGER FLAGS
    if( (GPIO_PORTF_RIS_R & 0x10) )
        {
            sw1 = 1;
            flag = 1;
            sw2 = 0;
         GPIO_PORTF_ICR_R = 0x10;
        }

    // IF SW 2 IS PRESSED TRIGGER FLAGS
    if( (GPIO_PORTF_RIS_R & 0x01) )
        {
            sw2 = 1;
            sw1 = 0;
            flag = 0;
     GPIO_PORTF_ICR_R = 0x01;
        }
}
```

```
// IR SENSOR INTERRUPT
void GPIOPortA_Handler(void)
{
	// PIN 7 PORT A INPUT
	GPIO_PORTA_ICR_R = 0x80;

	// OBJECT IS APPROACHING
	if ( go == 1 )
		{
			stop = 1;
			last = 0;
		}
	// OBJECT IS DEPARTING
	if ( go == 0 )
		{
			go = 0;
			last = 1;
			stop = 0;
		}
}

// PORT INITILIZATIONS
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  // Unlock at
beginning, broke code
	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 funct on 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&0xFF00FFFF)|0x00A00000; // (g) priority 5
  NVIC_EN0_R |= 0x40000000;      // (h) enable interrupt 30 in NVIC
}
```

```
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;
}
```

Smart Home Code:

```
int main( void )
{  unsigned int i=0;
    //TExaS_Init(SW_PIN_PF40,LED_PIN_PF321); // this initializes the TExaS grader lab 2

    PortF_Init();
    PortA_Init();
 SysTick_Init(); // WANT PERIOD OF 1/2 FOR 3 SEC, x12, HALF PERIOD HIGH/LOW
    EnableInterrupts();
 Stepper_Init();

    LIGHT = GREEN;

 while(1)
        {


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



                    if ( flag == 0)
                        {
```

```
                                                  trigger=0;



                                                  for(i=0;i<5000;i++)
                                                         {


                                                                Stepper_CW(40000);


                                                                       // output every 10ms
                                                         }
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<4300;i++)
                                                         {


                                                                Stepper_CCW(60000);



                                                         }
Counter =0;

                                            // output every 10ms

                                    flag = 0;}                                       LIGHT =
0x08;
```

```
                }
    }
}

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;
            }
        }
}

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;}
}
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;
                }
        }
}
```