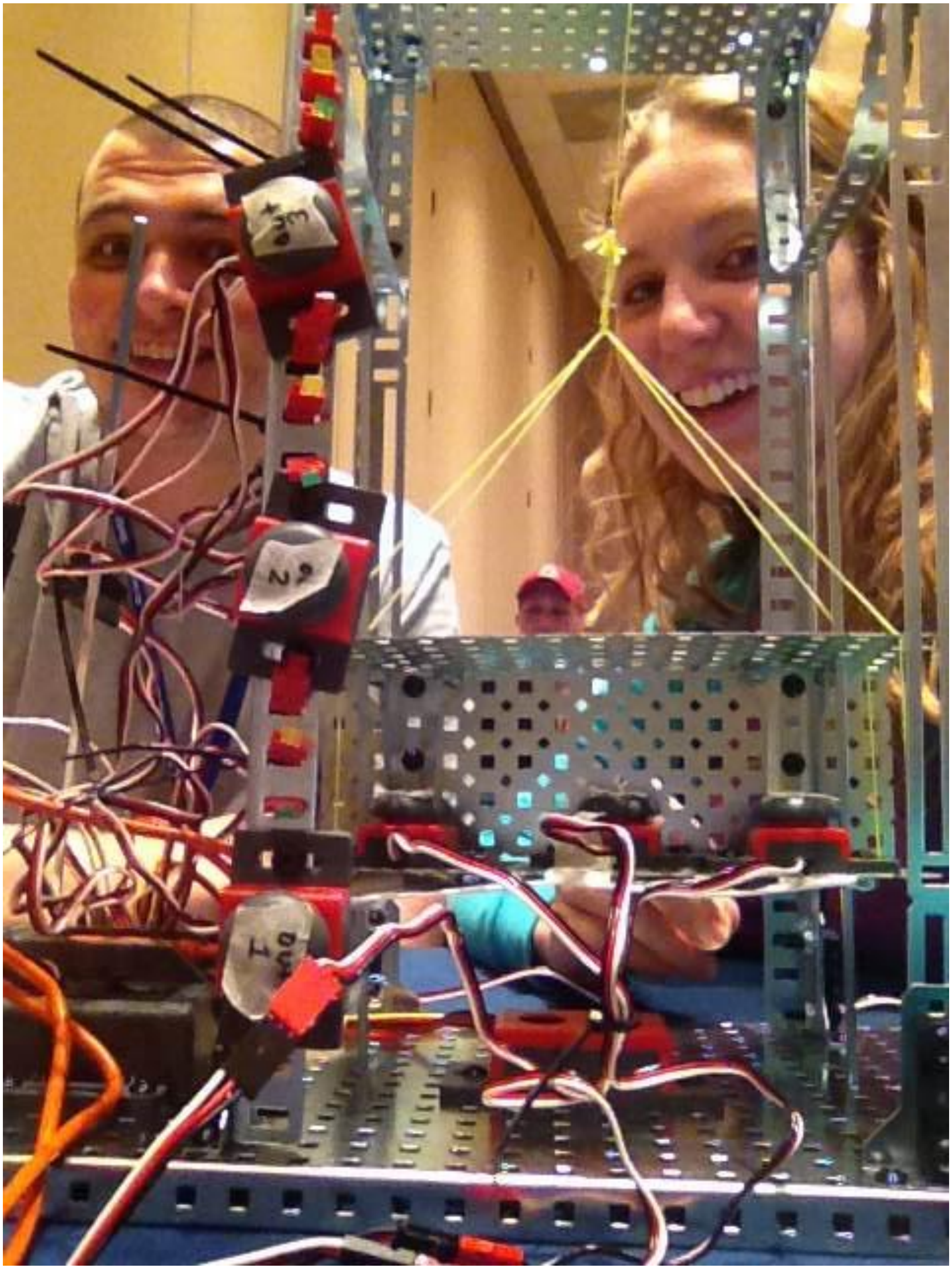
Machine Control Design
1/14/14
POE Training
Kristin and Brooks

Table of Contents

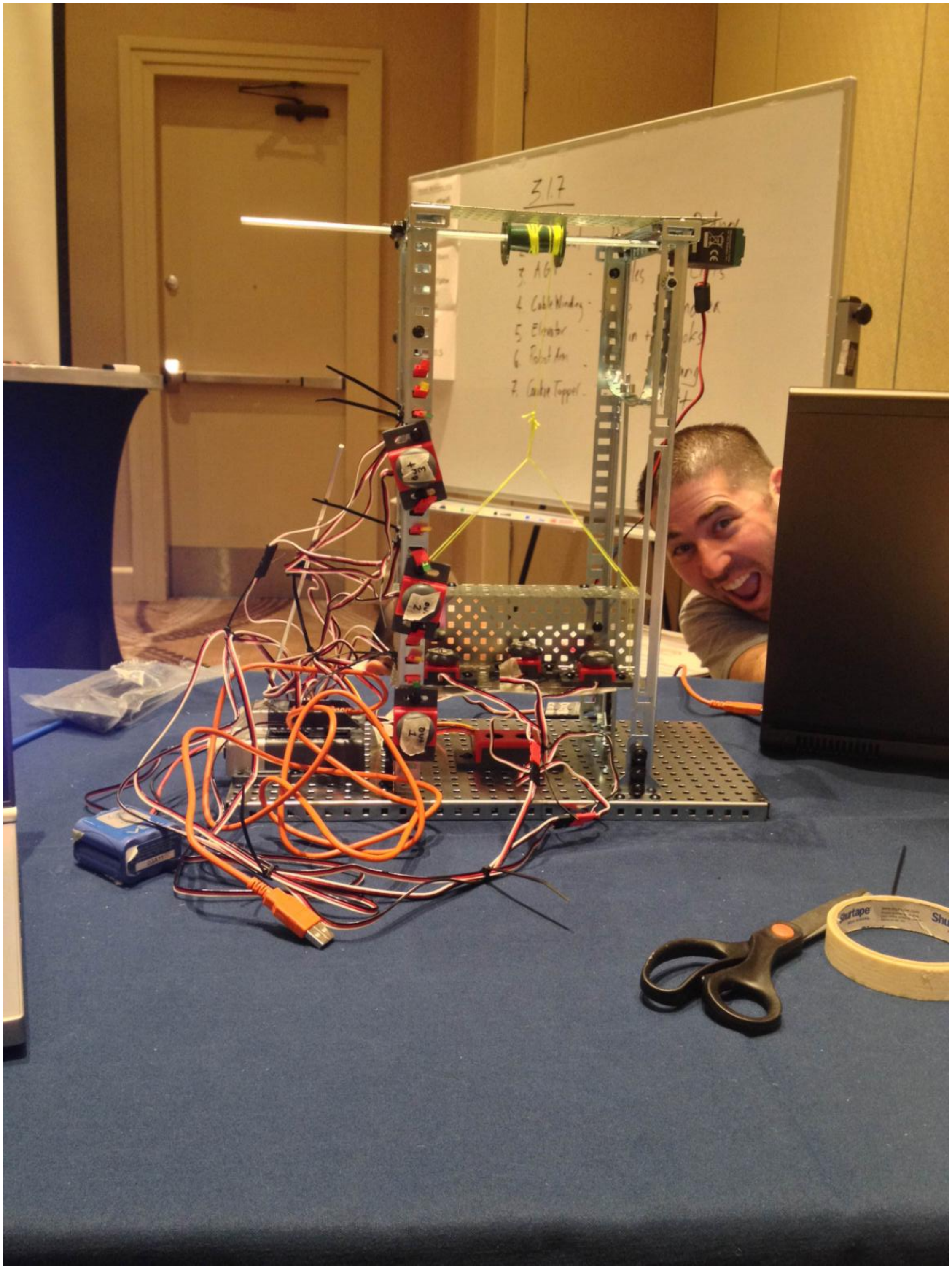
Problem

Problem 5: Elevator *(Hardware Level 4 Software Level 4)*
A company would like to begin producing residential elevators. Your team must design the control system and a prototype of an elevator that can go between three floors in any combination. The prototype must include a set of three switches to represent each floor of the elevator. Each floor the elevator stops at must have a call button and a set of three lights to indicate where the elevator is currently located. A built-in safety mechanism requires that the elevator normally rest on the ground floor and return to the ground floor after a user-determined period of nonuse.

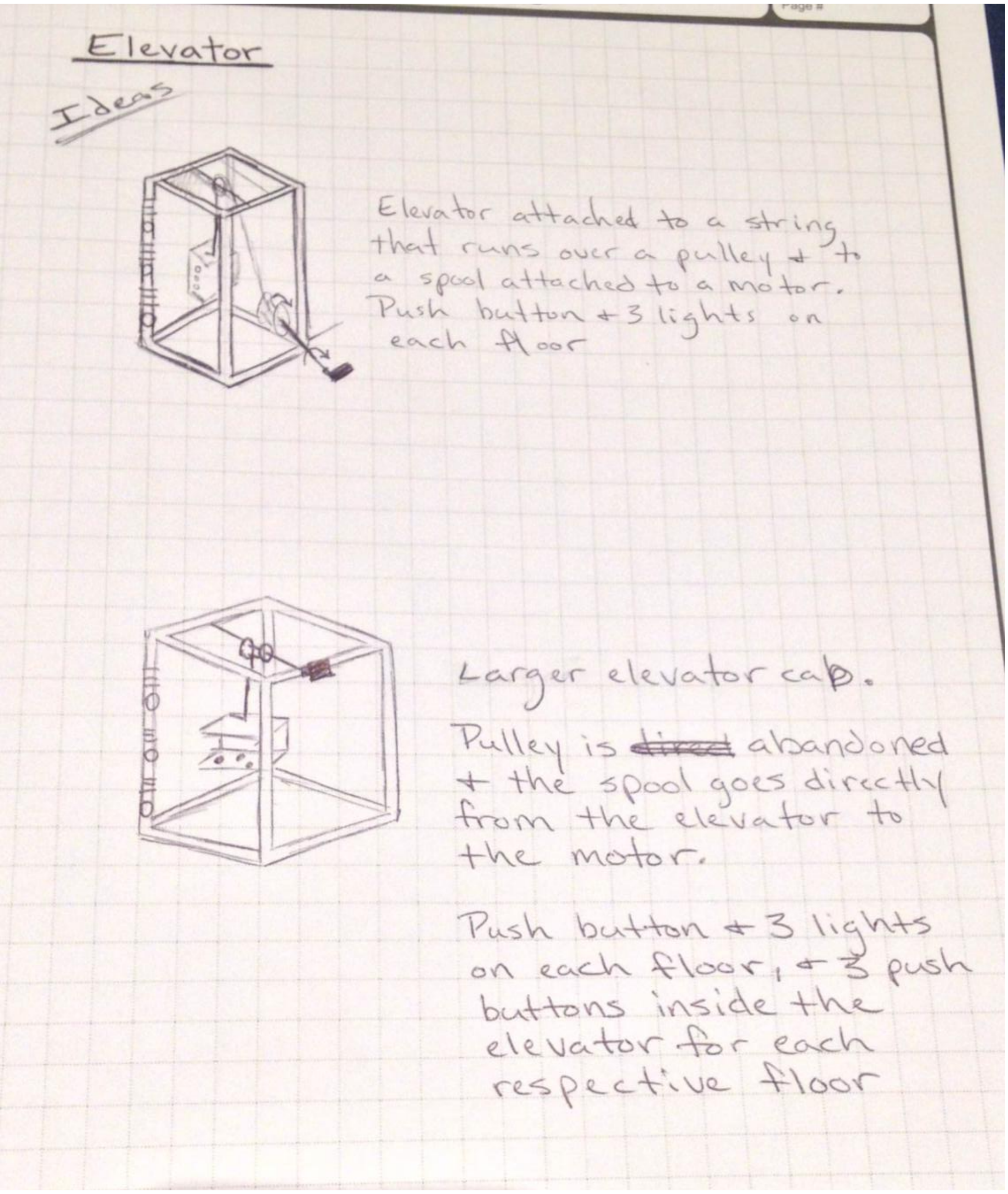
Picture 1 of the Final Solution



Picture 2 of the Final Solution



Two Potential Sketches



Elevator

Ideas

Elevator attached to a string that runs over a pulley + to a spool attached to a motor. Push button + 3 lights on each floor

Larger elevator cab.

Pulley is ~~direct~~ abandoned + the spool goes directly from the elevator to the motor.

Push button + 3 lights on each floor, + 3 push buttons inside the elevator for each respective floor

Two Potential Program Sketches

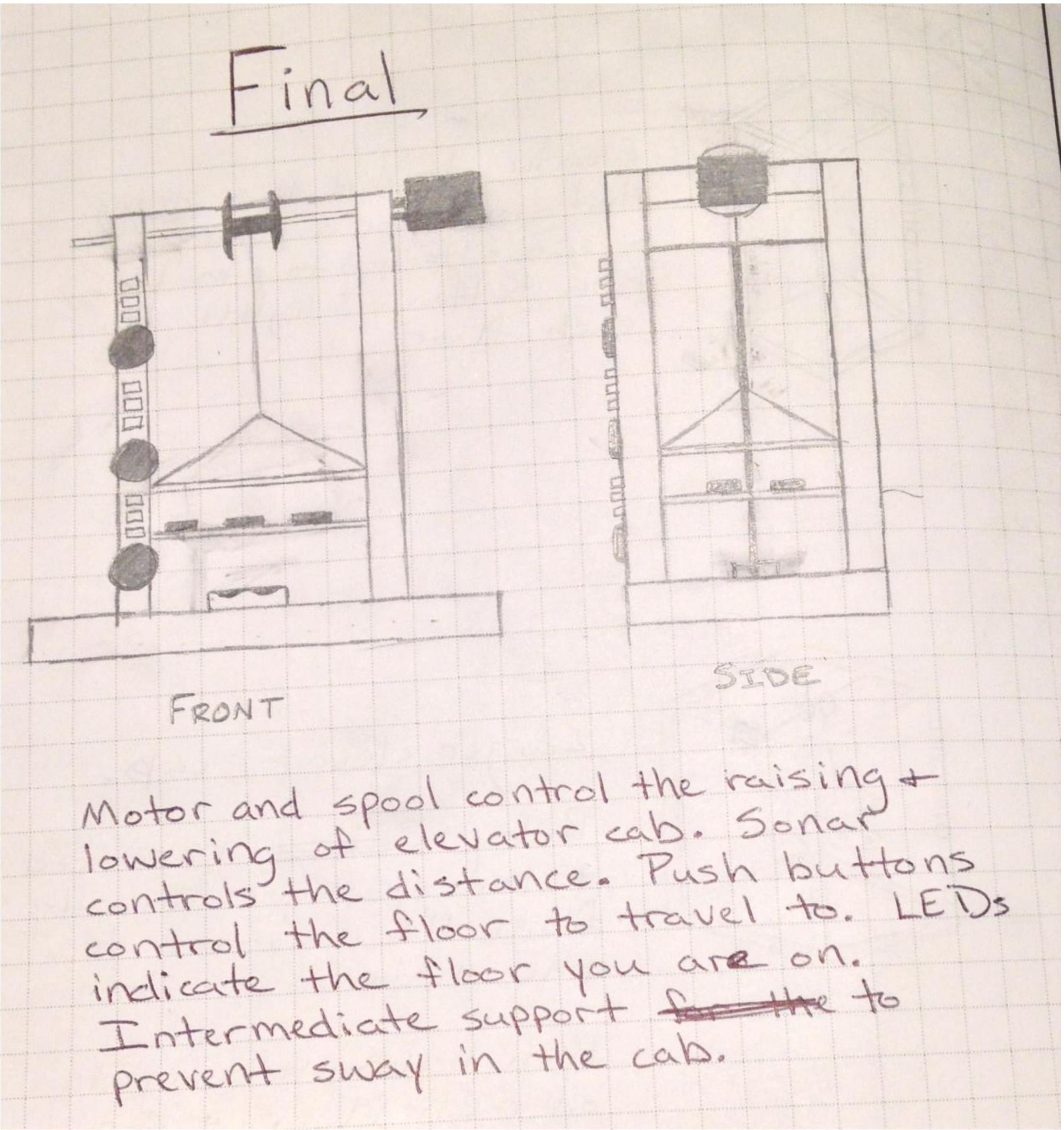
1.

```
constantly wait at floor 1
if inside button is pushed or if outside button is pushed, go to floor, turn LEDs on once at floor
Use ultrasonic to sense floor height
```

2.

```
Clear timer
Turn off LEDs
Move to floor 1
Always
Check if timer is at 5 seconds
If it is, move to floor 1
Check for button pushes
If a button is pushed, call a function MoveToFloor(in floorNum)
 In MoveToFloor(in floorNum)
Turn LEDS off
If sonar distance is above floor distance, move down until sonar reads appropriate distance
Opposite for sonar distance below floor level
Stop motor once at floor
Turn on LEDs for floor
Reset timer
```

Final Physical Sketches



Motor and spool control the raising + lowering of elevator cab. Sonar controls the distance. Push buttons control the floor to travel to. LEDs indicate the floor you are on. Intermediate support ~~for the~~ to prevent sway in the cab.

Final Program

```
#pragma config(Sensor, dgtl1,  G,           sensorLEDtoVCC)
#pragma config(Sensor, dgtl2,  Y,           sensorLEDtoVCC)
#pragma config(Sensor, dgtl3,  R,           sensorLEDtoVCC)
#pragma config(Sensor, dgtl4,  sonar,       sensorSONAR_cm)
#pragma config(Sensor, dgtl7,  button_in_1,    sensorTouch)
#pragma config(Sensor, dgtl8,  button_in_2,    sensorTouch)
#pragma config(Sensor, dgtl9,  button_in_3,    sensorTouch)
#pragma config(Sensor, dgtl10, button_out_1,   sensorTouch)
#pragma config(Sensor, dgtl11, button_out_2,   sensorTouch)
#pragma config(Sensor, dgtl12, button_out_3,   sensorTouch)
#pragma config(Motor,  port2,          motor1,        tmotorVex393, openLoop)
//*!!Code automatically generated by 'ROBOTC' configuration wizard               !!*//

/*
Project Title: Activity 3.1.7 Machine Control Design
Team Members: Kristin and Brooks
Date Started: 1/13/2014
Date Completed: 1/14/2014
Section: POE Training

Task Description: Elevator

Pseudocode:
constantly wait at floor 1
if inside button is pushed or if outside button is pushed, go to floor, turn LEDs on once at floor
Use ultrasonic to sense floor height
*/

void MoveToFloor(int floorNum);                 // move elevator to designated floor
void turnLedsOff ();                            // turn all LEDs off
void turnOnLed(int ledNum);                     // turn on particular LED

task main()
{
        ClearTimer(T1);                         // reset timer
        turnLedsOff();                          // turn all LEDs off - they are on by default

        if (SensorValue[sonar] != 5)            // if elevator is not already at floor 1
                MoveToFloor(1);                 // move to floor 1

        while(1)
        {
                if (time100[T1] > 50)           // check if motor hasn't moved in 5 seconds
                        MoveToFloor(1);         // if it hasn't moved in 5 seconds, move to floor 1
                if (SensorValue[button_in_1] == 1 || SensorValue[button_out_1] == 1)
                                                // if inside or outside button 1 is hit
                        MoveToFloor(1);         // move to floor 1
                if (SensorValue[button_in_2] == 1 || SensorValue[button_out_2] == 1)
                                                // if inside or outside button 2 is hit
                        MoveToFloor(2);         // move to floor 2
                if (SensorValue[button_in_3] == 1 || SensorValue[button_out_3] == 1)
                                                // if inside or outside button 3 is hit
                        MoveToFloor(3);         // move to floor 3
        }
}

void MoveToFloor(int floorNum)
{
        turnLedsOff();                          // turn all LEDS off

        if(SensorValue[sonar] > floorNum*5)                     // if elevator is higher than desired floor
                while (SensorValue[sonar] != floorNum*5)
                                                // while sonar distance is not at 5 times floor number
                startMotor(motor1, -40);        // move elevator down
        else                                    // else - elevator is lower than desired floor
                while (SensorValue[sonar] != floorNum*5)
                                                // while sonar distance is not at 5 times floor number
                startMotor(motor1, 40);         // move elevator up

        stopMotor(motor1);                      // stop motor

        turnOnLed(floorNum);                    // turn on appropriate LEDs

        ClearTimer(T1);                         // reset timer since elevator moved
}

/* turnLedsOff function turn all LEDS off */
void turnLedsOff ()
{
        turnLEDOff(G);
        turnLEDOff(Y);
        turnLEDOff(R);
}

/* turnOnLed function turns on a specific floor's LEDs*/
void turnOnLed(int ledNum)
{
        switch (ledNum)                                 // ledNum is floor number
        {
        case 1:                                         // floor 1
                {
                        turnLEDOn(G);                   // turn on green LEDs
                }
                break;
        case 2:                                 // floor 2
            {
                turnLEDOn(Y);                   // turn on yellow LEDs
            }
            break;
        case 3:                                 // floor 3
            {
                turnLEDOn(R);                   // turn on red LEDs
            }
            break;
        }
    }
```

Conclusion Questions

1. What was the most difficult part of the problem?

   The most difficult part of the problem was getting the LEDs hardwired properly. We also had a tough time stopping the motor in the right place since it tends to have a bit of momentum pulling it above or below the correct floor position.

2. List and describe two features that were not part of the design problem that could be added to improve your design.

   1. We could have used a gear system with a track to move the elevator to get a more accurate floor distance.
   2. We could have used an encoder to also help get a more accurate floor distance.