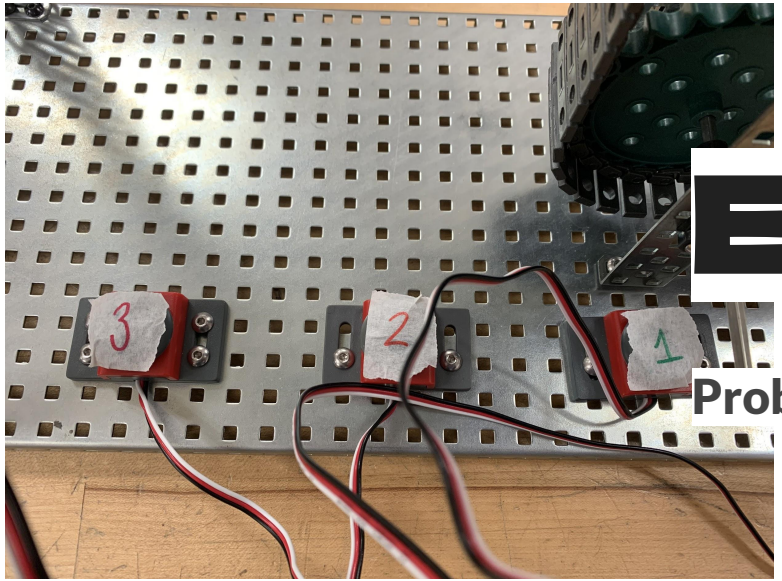


Elevator

Problem 5 on 3.1.7. Machine Control Design

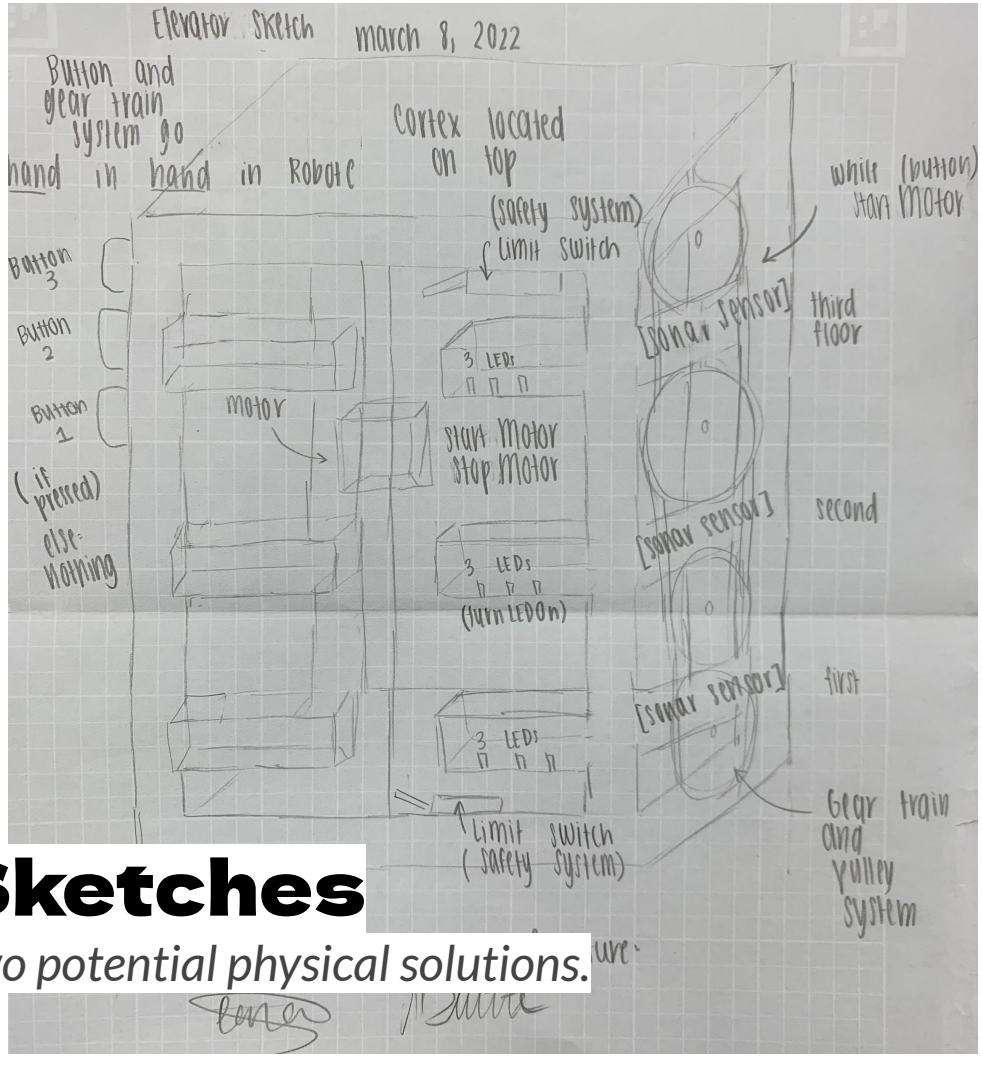
Overview

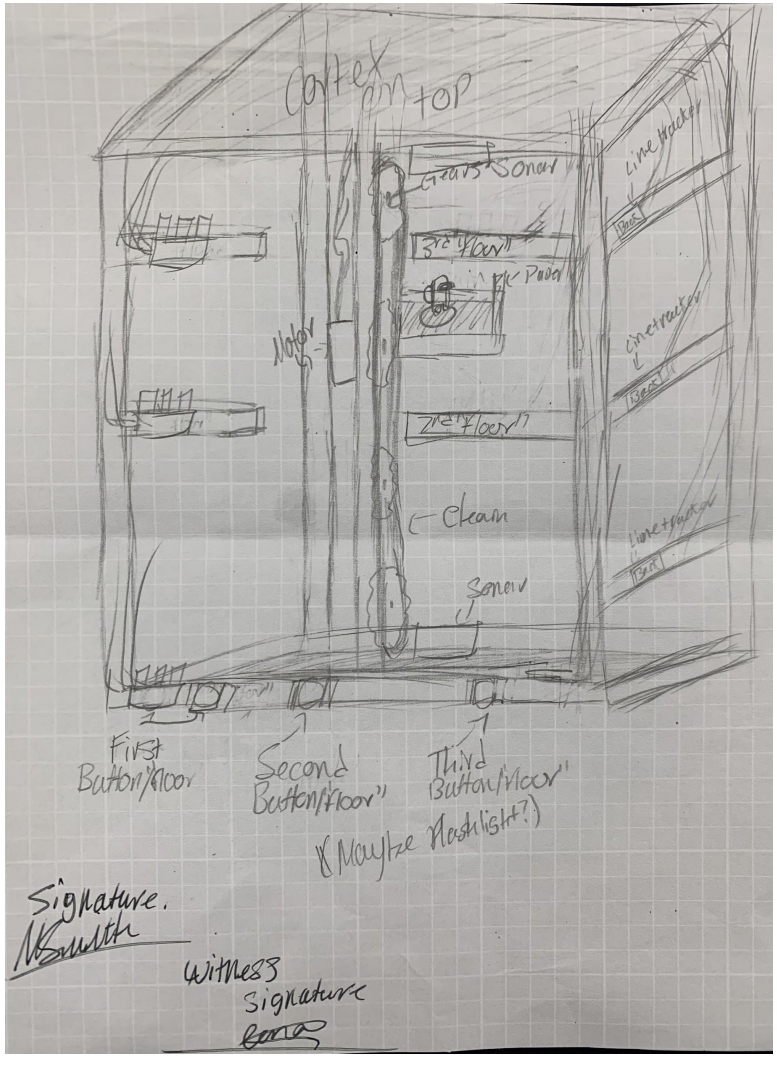
The elevator required the following:

- 3 floors: ground, second and third floors.
  - 3 LED lights on *each floor*, making for a total of 9.
  - 3 call buttons calling to each floor.
    - Bump switches.
- Built-in safety mechanism after a “user-determined period of nonuse”.
  - Timer in the code; T1.
- Way to register the floors; sensors *must* be used.
  - Determined the use of sprockets and chains and line followers.

Sketches

Two potential physical solutions.





```
task main() {
  ClearTimer(T1); // Servos use a timer to determine detination
  while(1==1) {
      if(time10[T1] == 1000) {
        turnLEDOff(green);
        turnLEDOff(red);
        turnLEDOff(yellow);
        setServo(rightServo, 115);
        setServo(leftServo, -115);
        turnLEDOn(green);
    }

    if(SensorValue(limitSwitch1)==1) { // Registering floors
      ClearTimer(T1);
      turnLEDOff(green);
      turnLEDOff(red);
      turnLEDOff(yellow);
      setServo(rightServo, 115);
      setServo(leftServo, -115);
      turnLEDOn(green);
    }

    if(SensorValue(bumpSwitch1)==1) { // Determening destination, the floor the servo will end up on.
      ClearTimer(T1);
      turnLEDOff(green);
      turnLEDOff(red);
      turnLEDOff(yellow);
      setServo(rightServo, 115);
      setServo(leftServo, -115);
      turnLEDOn(green);
    }
```

Sketches

Potential program solutions.

#1 used servos as a backup in case the line followers didn't work. Scrapped due to inefficiency.

```
/* first color floor is white, second is blue, and third is black */
task main() {
  while (1 == 1) {
    clearTimer(T1);
    LED();
    //Safety();

    /* first floor */
    if (SensorValue(lineFollower) > 2950 && SensorValue(lineFollower) < 3100 && SensorValue(bumpSwitchThree) == 1) {
      startMotor(rightMotor, 127);
      /* untilLight(3100, lineFollower);
      stopMotor(rightMotor); */
    }
    if (SensorValue(lineFollower) > 2950 && SensorValue(lineFollower) < 3100 && SensorValue(bumpSwitch) == 1) {
      startMotor(rightMotor, 127);
      /* untilLight(3100, lineFollower);
      stopMotor(rightMotor); */
    }
    if (SensorValue(lineFollower) > 2950 && SensorValue(lineFollower) < 3100 && SensorValue(bumpSwitchTwo) == 1) {
      stopMotor(rightMotor);
    }
```

Sketches

Potential program solutions.

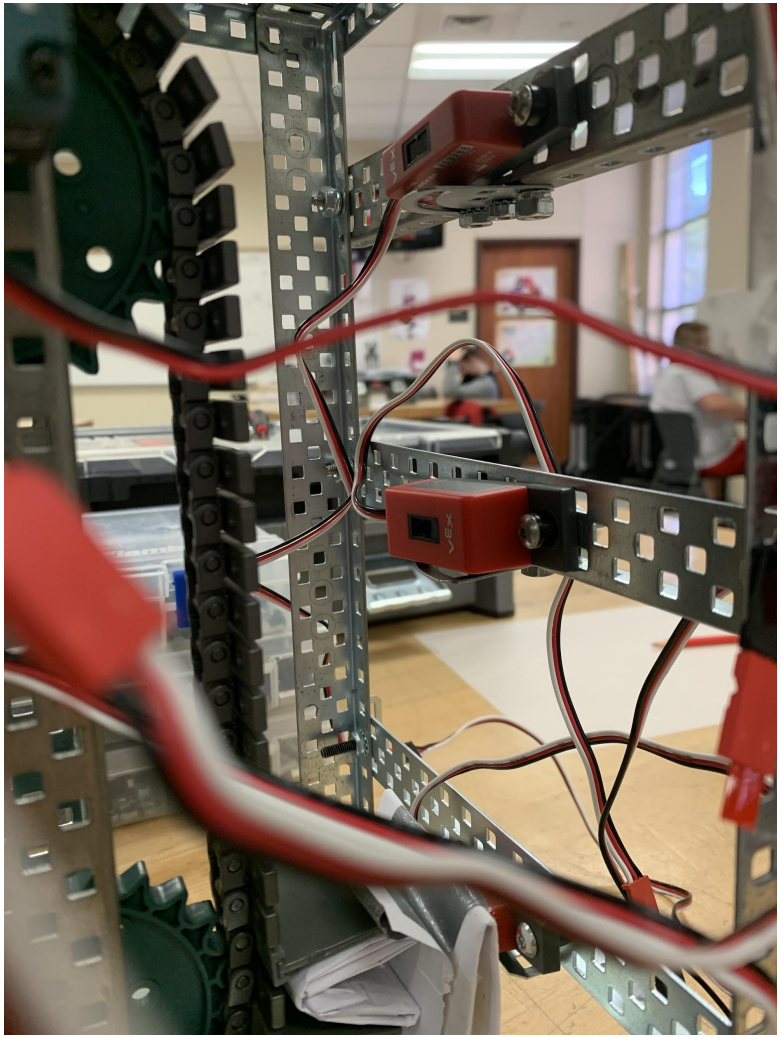
#2 planned to use colored paper for the line followers. Wasn't accurate enough and later remodified.

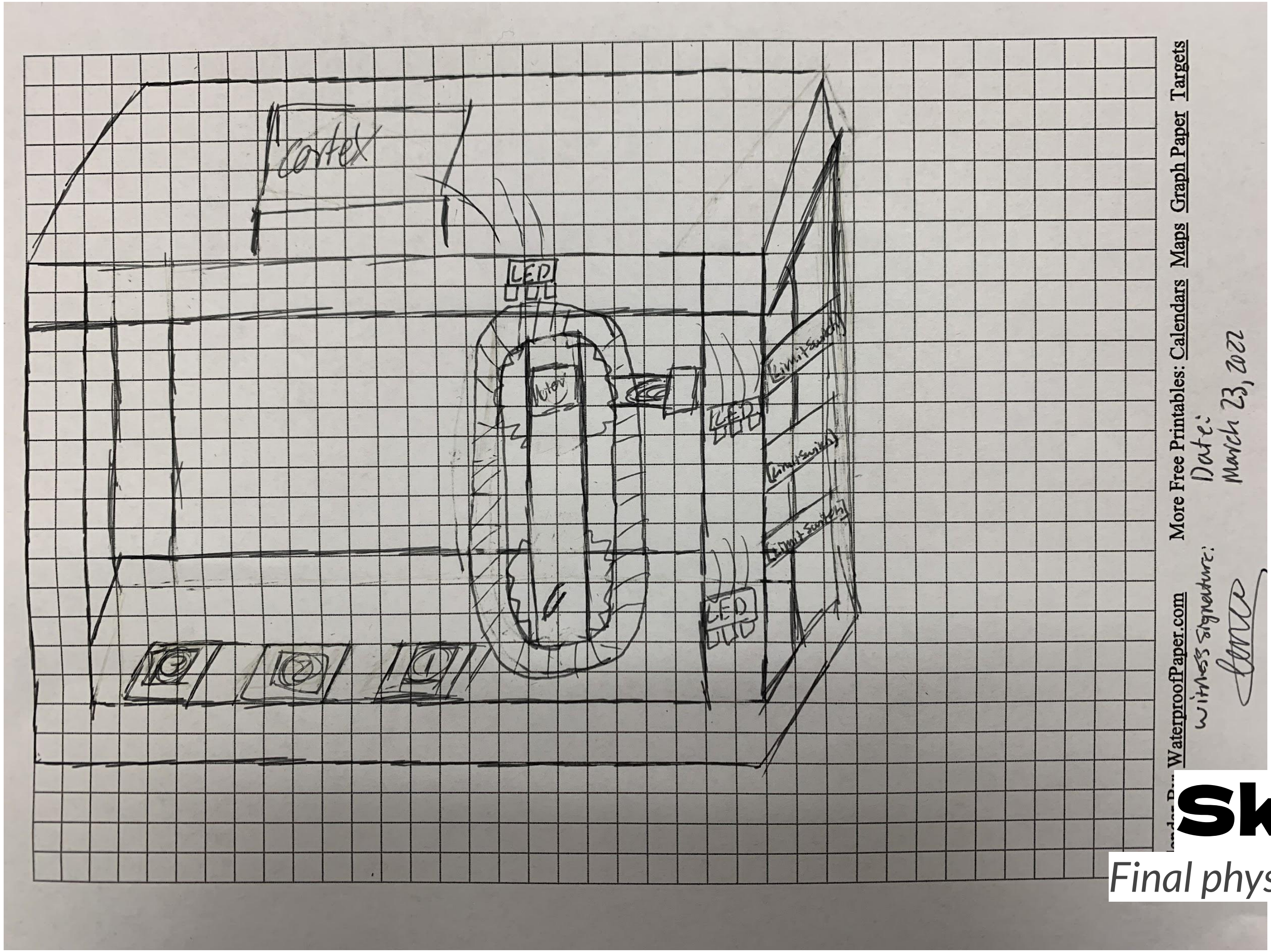
Building

Using the requirements and restrictions from the overview, we ended with this design:

Changes made throughout the building process include:

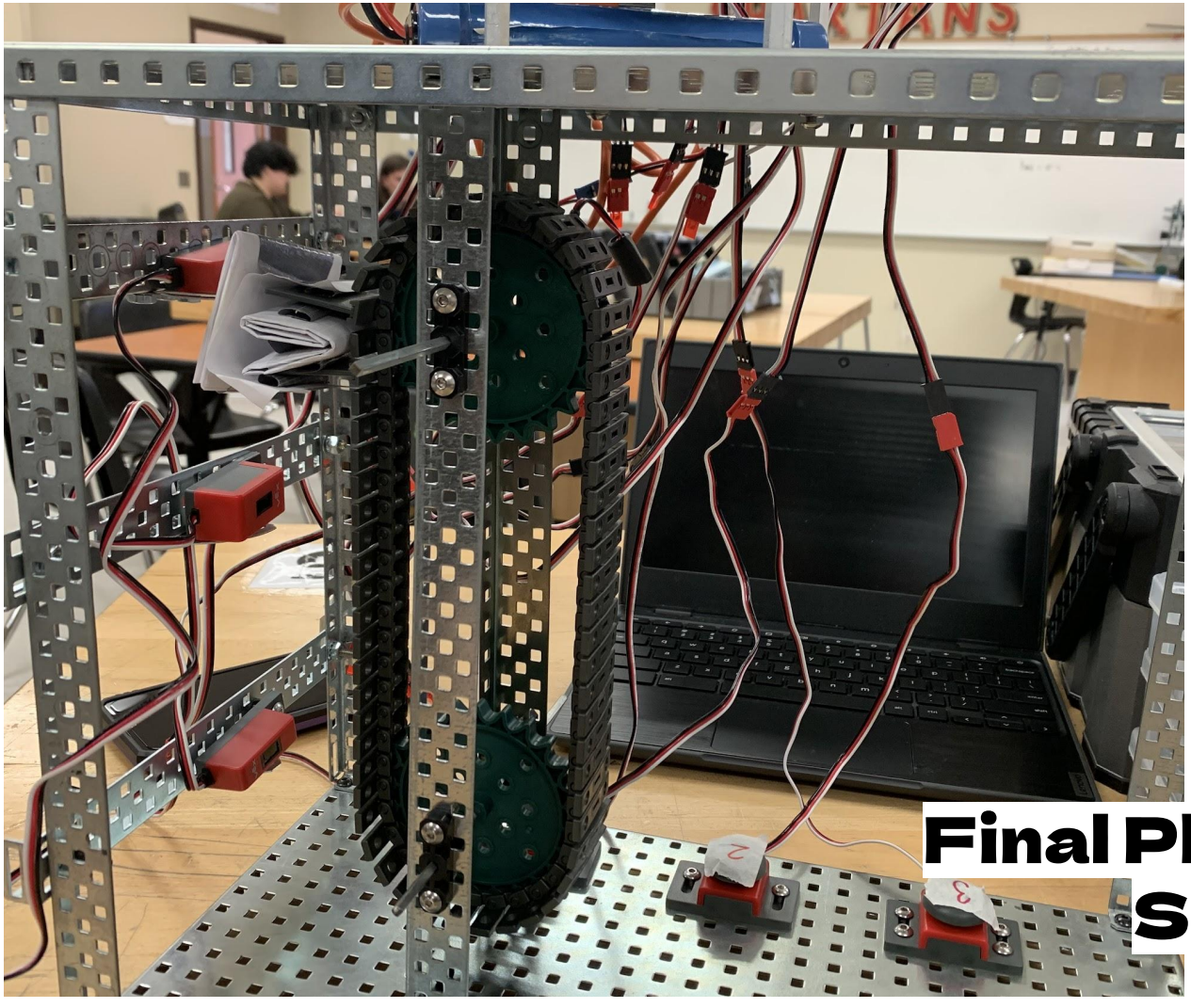
- Each floor having one line follower attached instead of just one on the elevator.
- Switching the position of the sprocket and chain to be closer to the line followers.





Sketch

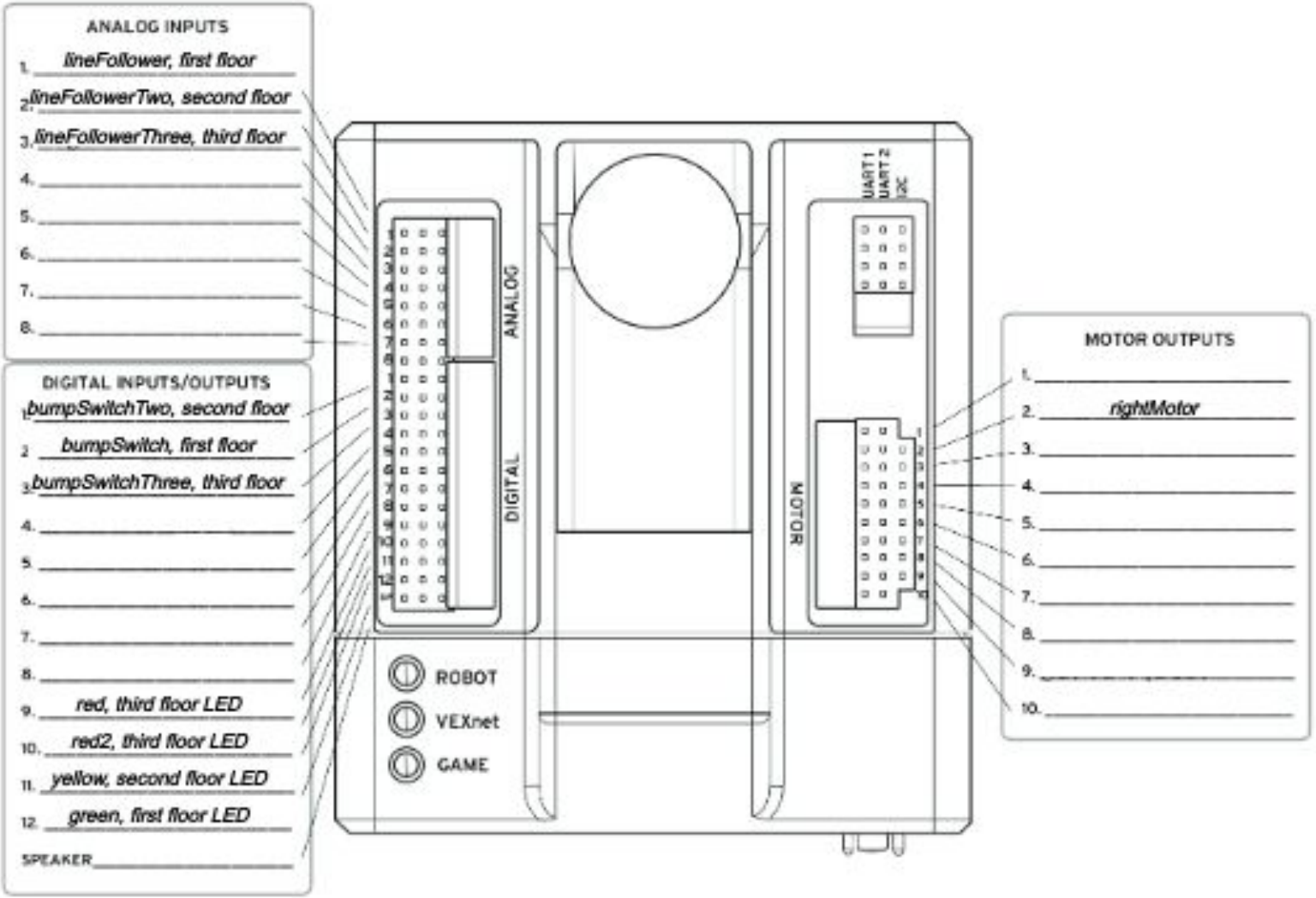
Final physical sketch.



Final Physical Solution

Wire Set-up

```
1  #pragma config(Sensor, in1,    lineFollower,   sensorLineFollower)
2  #pragma config(Sensor, in2,    lineFollowerTwo, sensorLineFollower)
3  #pragma config(Sensor, in3,    lineFollowerThree, sensorLineFollower)
4  #pragma config(Sensor, dgtl1,  bumpSwitchTwo,  sensorTouch)
5  #pragma config(Sensor, dgtl2,  bumpSwitch,     sensorTouch)
6  #pragma config(Sensor, dgtl3,  bumpSwitchThree, sensorTouch)
7  #pragma config(Sensor, dgtl9,  red2,           sensorLEDtoVCC)
8  #pragma config(Sensor, dgtl10, red,            sensorLEDtoVCC)
9  #pragma config(Sensor, dgtl11, yellow,         sensorLEDtoVCC)
10 #pragma config(Sensor, dgtl12, green,          sensorLEDtoVCC)
11 #pragma config(Motor,  port2,           rightMotor,    tmotorVex269_MC29, openLoop)
```


ANALOG INPUTS
1. lineFollower, first floor
2. lineFollowerTwo, second floor
3. lineFollowerThree, third floor
4.
5.
6.
7.
8.
DIGITAL INPUTS/OUTPUTS
1. bumpSwitchTwo, second floor
2. bumpSwitch, first floor
3. bumpSwitchThree, third floor
4.
5.
6.
7.
8.
9. red, third floor LED
10. red2, third floor LED
11. yellow, second floor LED
12. green, first floor LED
SPEAKER
ANALOG
DIGITAL
UART 1
UART 2
I2C
MOTOR
ROBOT
VEXnet
GAME
MOTOR OUTPUTS
1.
2. rightMotor
3.
4.
5.
6.
7.
8.
9.
10.

```
17    // FLOOR ONE
18    if (SensorValue(lineFollowerTwo) < 200 && SensorValue(bumpSwitch) == 1) {
19      turnLEDOff(red);
20      turnLEDOff(red2);
21      turnLEDOff(yellow);
22      turnLEDOn(green);
23      startMotor(rightMotor, 17);
24      untilLight(300, lineFollower);
25      stopMotor(rightMotor);
26      clearTimer(T1);
27    }
28    if (SensorValue(lineFollowerThree) < 1500 && SensorValue(bumpSwitch) == 1) {
29      turnLEDOff(red);
30      turnLEDOff(red2);
31      turnLEDOff(yellow);
32      turnLEDOn(green);
33      startMotor(rightMotor, 17);
34      untilLight(300, lineFollower);
35      stopMotor(rightMotor);
36      clearTimer(T1);
37    }
38    if (SensorValue(lineFollower) < 200 && SensorValue(bumpSwitch) == 1) {
39      turnLEDOff(red);
40      turnLEDOff(red2);
41      turnLEDOff(yellow);
42      turnLEDOn(green);
43      stopMotor(rightMotor);
44    }
```

Sketch

Final program sketch.

Final Program

// *If* the **line follower** detects the elevator's value and **bump switch** are pressed of the same floor,

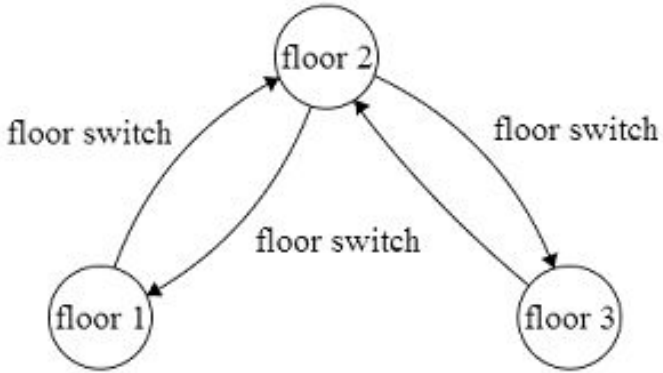
*< 200 on first and second, < 1500 on third.

// all led lights will be off except the one for the determined floor,

// and the motor moves to it's designated floor *until* the **line follower** on said floor detects the elevator's *value.

*300 on first and second, 1,000 on third.

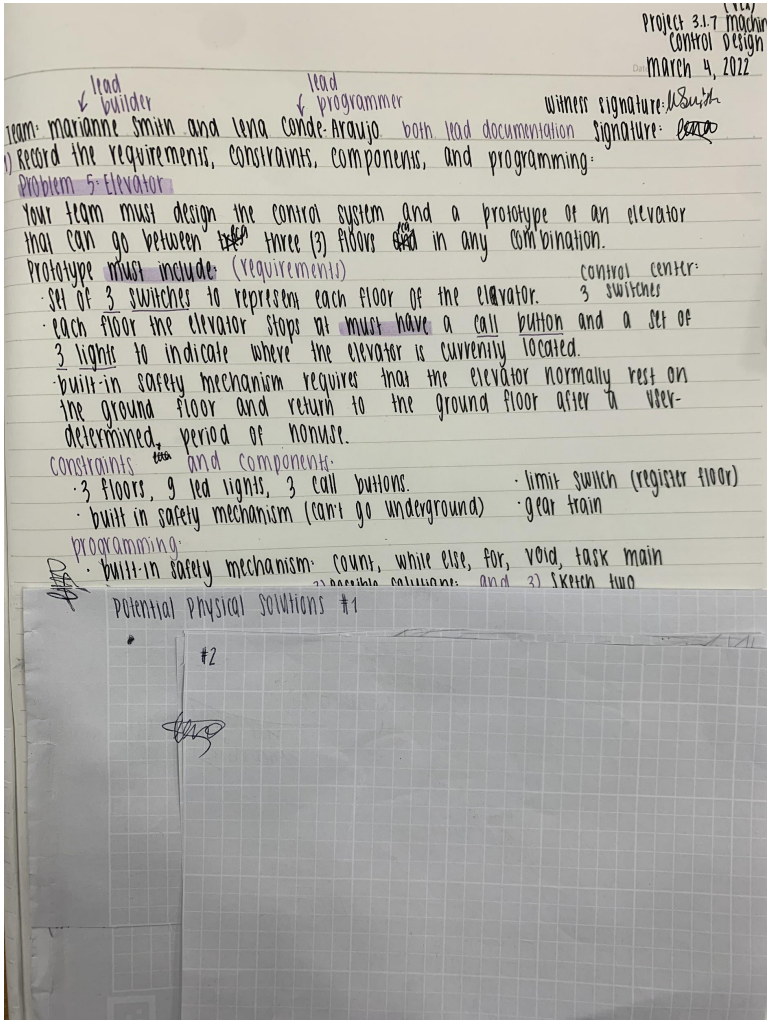
// After user use, clear the timer for the safety mechanism.



Safety Mechanism

- Supported by using the command *ClearTimer(T1)*; right before the while loop continuously runs the code and anytime a bump switch is pressed.
- Returns the elevator to the first floor after a 10 second period of non-use.

```
105 void Safety() {
106   if(time1[T1] == 10000) {
107     startMotor(rightMotor, 17);
108     untilLight(300, lineFollower);
109     stopMotor(rightMotor);
110   }
111 }
```



Project 3.1.7 Machine Control Design
March 4, 2022

lead builder — lead programmer

Team: Marianne Smith and Lena Conde-Araujo both lead documentation

Witness signature: MSmith

Signature:

1) Record the requirements, constraints, components, and programming:

Problem 5: Elevator

Your team must design the control system and a prototype of an elevator that can go between three (3) floors in any combination.

Prototype must include: (requirements)

- Set of 3 switches to represent each floor of the elevator. control center: 3 switches
- Each floor the elevator stops at must have a call button and a set of 3 lights to indicate where the elevator is currently located.
- 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.

Constraints and components:

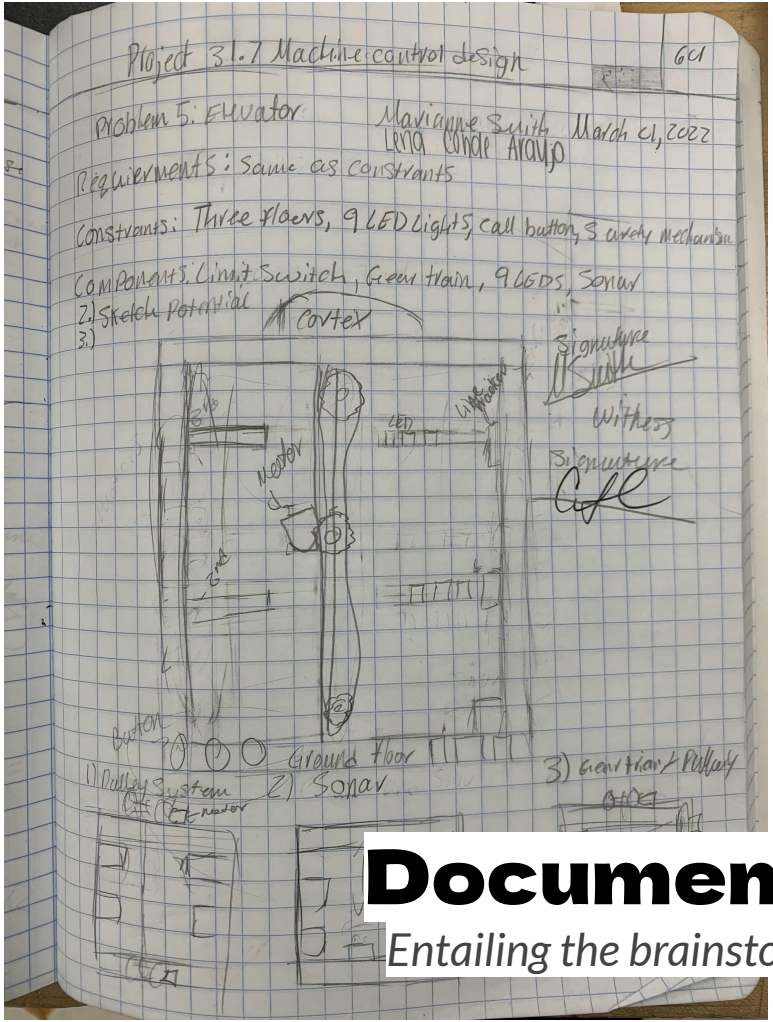
- 3 floors, 9 led lights, 3 call buttons.
- built in safety mechanism (can't go underground)
- limit switch (register floor)
- gear train

programming:

- built-in safety mechanism: count, while else, for, void, task main

Potential Physical Solutions #1

#2



Project 3.1.7 Machine control design 6C1

Problem 5: Elevator Marianne Smith Lena Conde Araujo March 01, 2022

Requirements: Same as constraints

Constraints: Three floors, 9 LED Lights, call button, 3 switch, mechanism

Components: Limit Switch, Gear train, 9 LEDs, Sonar

2.) Stretch Potential

3.)

Cortex

Signature Witness Signature

Motor

LED

Light sensor

Button

Ground floor

1) Pulley System 2) Sonar

3) Gear train + Pulley

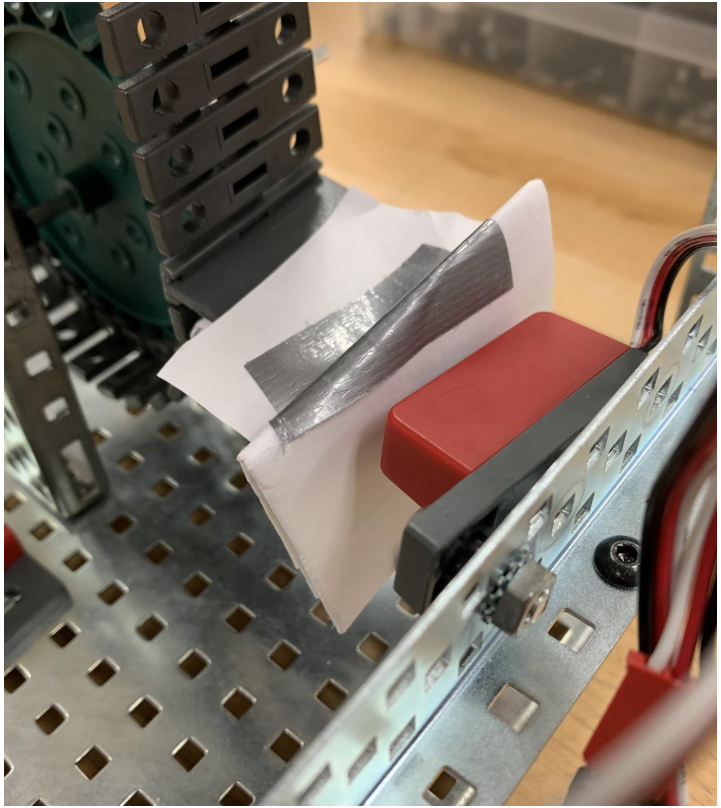
Documentation

Entailing the brainstorming process.

Conclusion Q's

What was the most difficult part of the problem?

The most difficult part of the problem was deciding on a sensor to include in our prototype. The elevator can use many different sensors or methods to get from one floor to another, ranging from being servos to ultrasonic range finders. We had to go through a difficult process of deciding the most efficient way to move our elevator, considering factors such as accuracy.



Conclusion Q’s

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

1. Safety mechanism: using an improved sonar sensor for increased accuracy.
2. Improved method of movement. Sprocket and chain resembles that of older elevators, would've preferred to go down a more modern method.