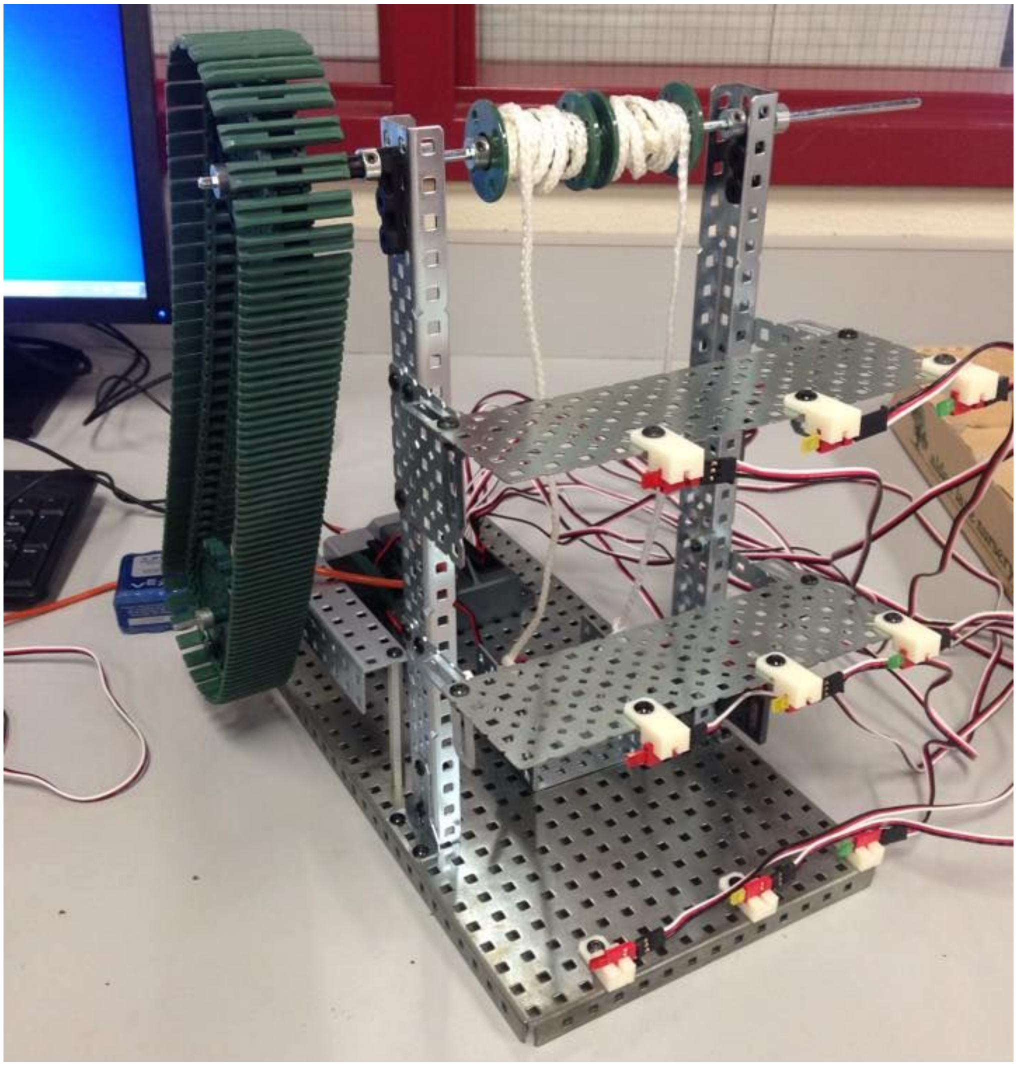


3.1.7 Machine Control Design - Elevator

Principles of Engineering 3˚

Rhea Manocha

Kristie Chan

3 May 2016

Table of Contents

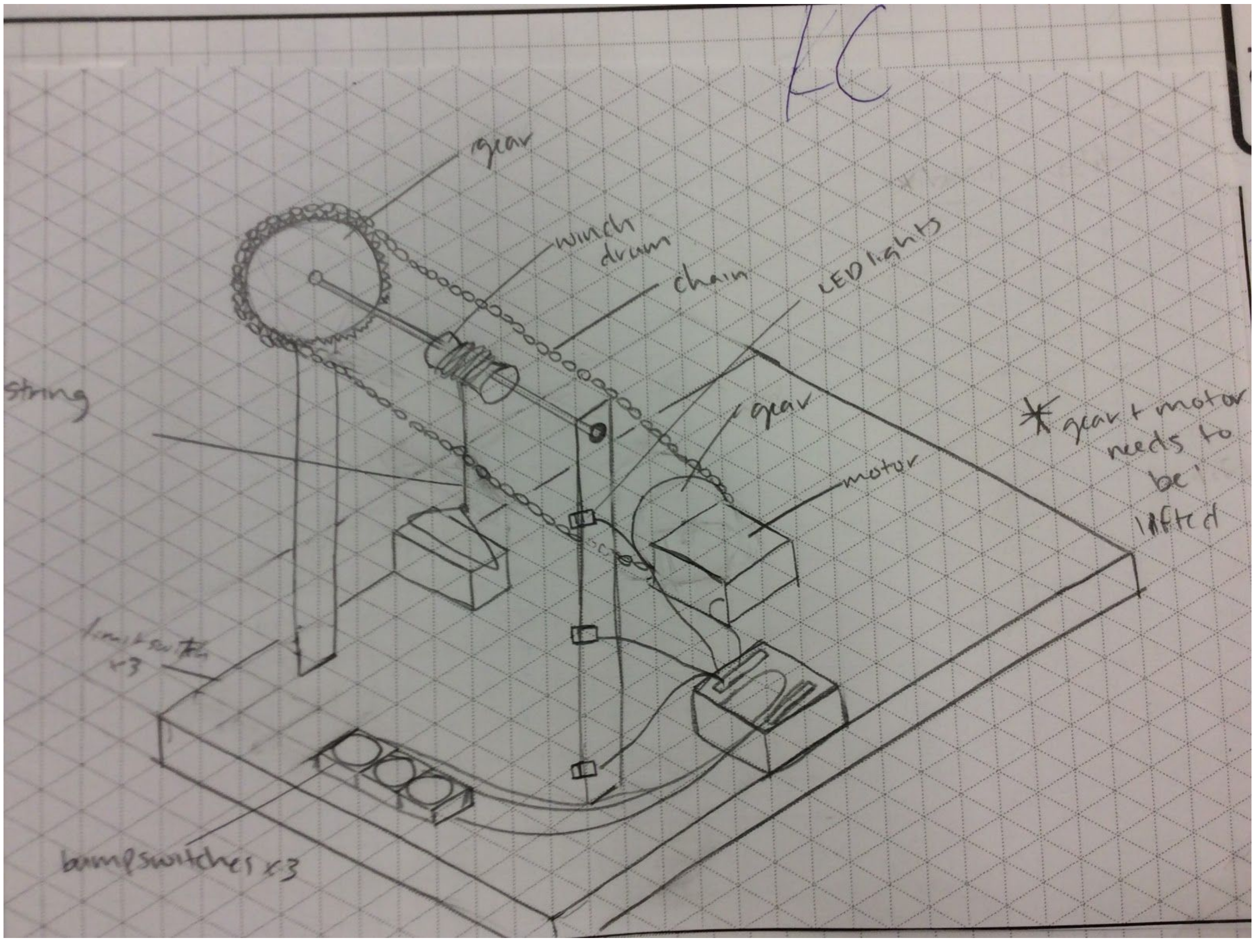
Design Statement	2
Initial Physical Sketches	2
Initial Flowcharts	3
Initial Program Sketches	5
Final Physical Sketch	8
Final Flowcharts	8
Final Program Sketch	9
Conclusion Questions	11

Design Statement

Problem 5: Elevator (Hardware Level 4 Software Level 4)

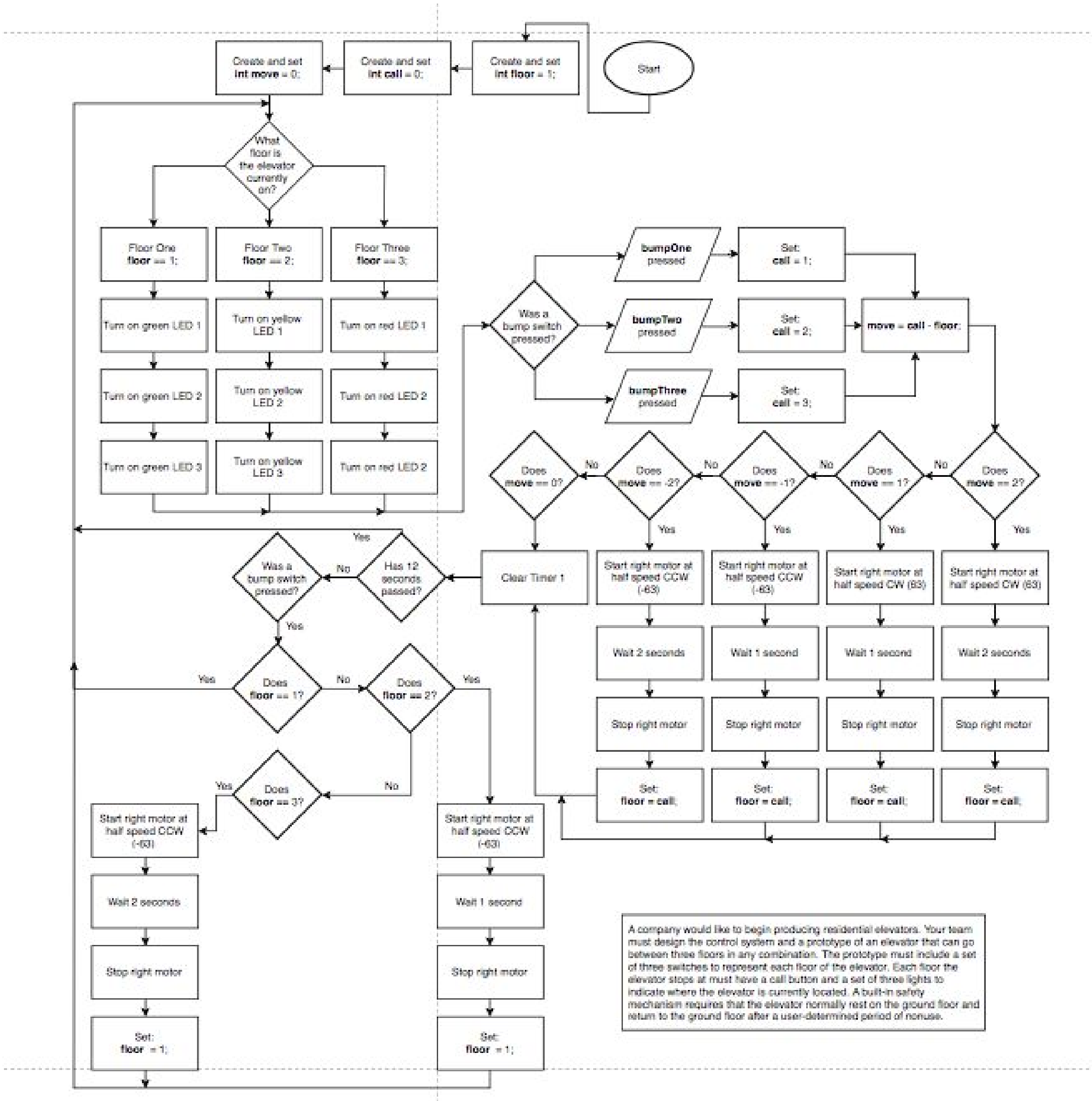
A company would like to begin producing residential elevators. Your team must design the control system and a prototype of an elevator that can go between three floors in any combination. The prototype must include a set of three switches to represent each floor of the elevator. Each floor the elevator 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.

Initial Physical Sketches



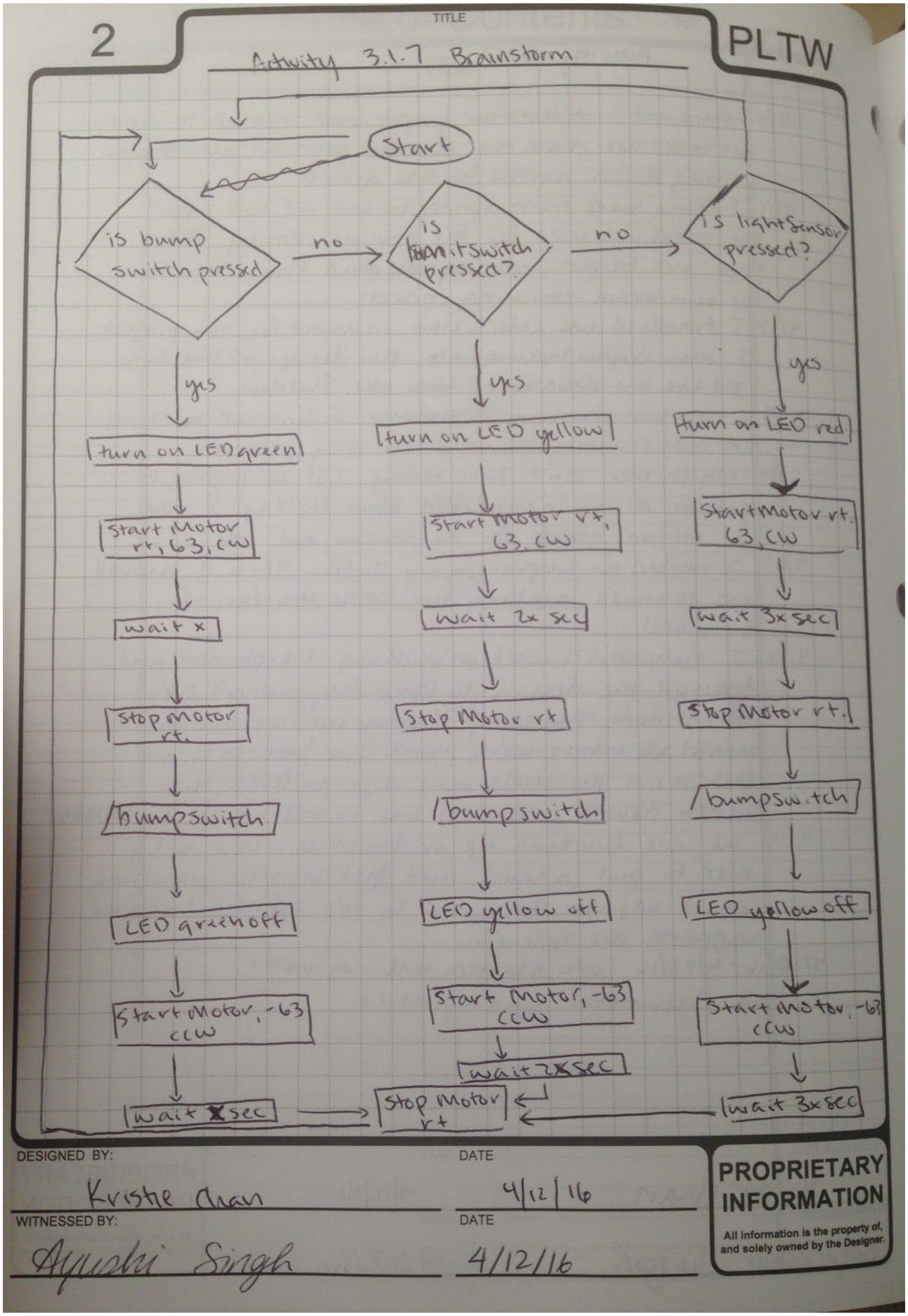
Initial Flowcharts

Design #1



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.

Design #2



2

TITLE Activity 3.1.7 Brainstorm

PLTW

DESIGNED BY: Kristie Chan

DATE 4/12/16

WITNESSED BY: Ayushi Singh

DATE 4/12/16

PROPRIETARY INFORMATION

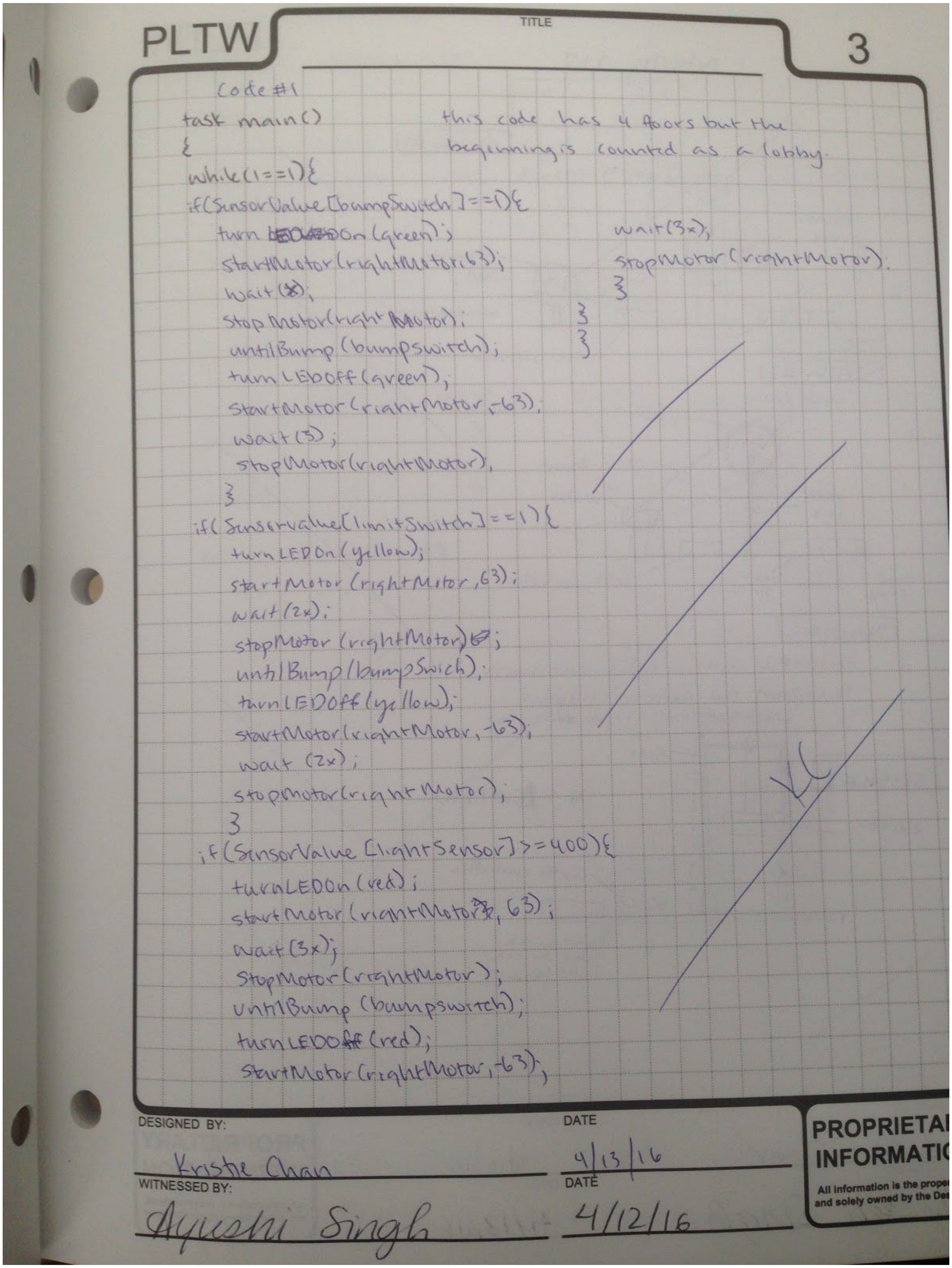
All information is the property of, and solely owned by the Designer.

Initial Program Sketches

Design #1

```
void floorLight(){ //Setting floor variable/lights to track what floor the elevator is currently on
        if(floor == 1) { //If elevator is on floor 1
                turnLEDOn(greenOne); //Then turn on the green light on floor one
                turnLEDOn(greenTwo); //Then turn on the green light on floor two
                turnLEDOn(greenThree); //Then turn on the green light on floor three
        }
        else if(floor == 2) {
                turnLEDOn(yellowOne); //Then turn on the yellow light on floor one
                turnLEDOn(yellowTwo); //Then turn on the yellow light on floor two
                turnLEDOn(yellowThree); //Then turn on the yellow light on floor three
        }
        else if(floor == 3) {
                turnLEDOn(redOne); //Then turn on the red light on floor one
                turnLEDOn(redTwo); //Then turn on the red light on floor two
                turnLEDOn(redThree); //Then turn on the red light on floor three
        }
}
int floor = 1; //Variable to track what floor elevator is currently on
int call = 0; //Variable to track what floor elevator is called to
int move = 0; //Variable to calculate how much elevator needs to move
while (1 == 1){ //Loop forever

        floorLight(); //Setting lights to track what floor the elevator is currently on

        //Bump switch input to call elevator to a floor
        if(SensorValue[bumpOne] == 1) { //If the first floor bump switch is pressed
                call = 1; //Sets call to equal the floor it was called to
        }
        else if(SensorValue[bumpTwo] == 2) { //If the two floor bump switch is pressed
                call = 2; //Sets call to equal the floor it was called to
        }
        else if(SensorValue[bumpThree] == 1) { //If the third floor bump switch is pressed
                call = 3; //Sets call to equal the floor it was called to
        }
        move = call - floor; //Calculate how much elevator needs to move

        //Based on the value of move, elevator will move to floor it was called to
        if(move == 2){ //If the elevator is going 2 floors up
                startMotor(rightMotor, 63); //Start the right motor at half speed CW
                wait(2); //Wait 2 seconds
                stopMotor(rightMotor); //Stop motor after 2 seconds
        }
        else if(move == 1){
                startMotor(rightMotor, 63); //Start the right motor at half speed CW
                wait(1); //Wait 1 second
        stopMotor(rightMotor); //Stop motor after 1 second
    }
    else if(move == -1){
        startMotor(rightMotor, -63); //Start the right motor at half speed CCW
        wait(1); //Wait 1 second
        stopMotor(rightMotor); //Stop motor after 1 second
    }
    else if(move == -2){
        startMotor(rightMotor, -63); //Start the right motor at half speed CCW
        wait(2); //Wait 2 seconds
        stopMotor(rightMotor); //Stop motor after 2 seconds
    }
    else if(move == 0){
        //Don’t move
    }
    floor = call; //Set floor to the floor it was called to and is now on
    floorLight(); //Setting lights to track what floor the elevator is currently on

    //Timer to track if elapsed time (12 seconds) has passed since elevator has moved, then will move it back to first floor
    ClearTimer(T1);
    while(Timer[T1] <= 12000) { //While the timer is less than or equal to 12 seconds
        if(Timer[T1] >= 12000 && SensorValue[bumpOne] && 0 && SensorValue[bumpTwo] == 0 && SensorValue[bumpThree] == 0){ //If 12 seconds has passed and a bump switch hasn’t been pressed
            if(floor == 2){ //If elevator is on the second floor
                startMotor(rightMotor, -63); //Start the right motor at half speed CCW
                wait(1); //Wait 1 second
                stopMotor(rightMotor); //Stop motor after 1 second
            }
            else if(floor == 3){ //If elevator is on the third floor
                startMotor(rightMotor, -63); //Start the right motor at half speed CCW
                wait(2); //Wait 2 seconds
                stopMotor(rightMotor); //Stop motor after 1 second
            }
            floor = 1; //Set floor for first floor
        }
    }
    ClearTimer(T1); //Clear timer again so it doesn’t keep running; not necessary
```

Design #2



PLTW

TITLE

3

Code #1

this code has 4 floors but the beginning is counted as a lobby.

```
task main()
{
while(1==1){
if(SensorValue[bumpSwitch]==1){
   turnLEDOn(green);
   startMotor(rightMotor,63);
   wait();
   stopMotor(rightMotor);
   untilBump(bumpSwitch);
   turnLEDOff(green);
   startMotor(rightMotor,-63);
   wait(3);
   stopMotor(rightMotor);
   }
if(Sensorvalue[limitSwitch]==1){
   turnLEDOn(yellow);
   startMotor(rightMotor,63);
   wait(2x);
   stopMotor(rightMotor);
   untilBump(bumpSwich);
   turnLEDOff(yellow);
   startMotor(rightMotor,-63);
   wait(2x);
   stopMotor(rightMotor);
   }
if(SensorValue[lightSensor]>=400){
   turnLEDOn(red);
   startMotor(rightMotor,63);
   wait(3x);
   StopMotor(rightMotor);
   untilBump(bumpswitch);
   turnLEDOff(red);
   startMotor(rightMotor,-63);
   wait(3x);
   stopMotor(rightMotor);
   }
  }
 }
```

KC

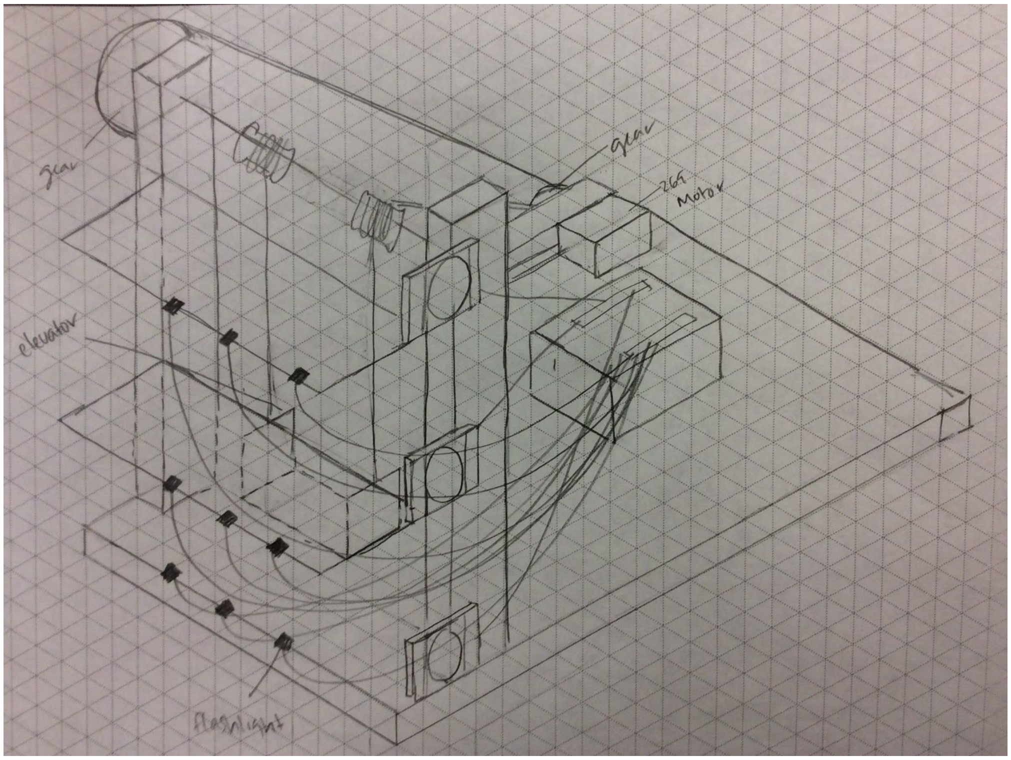
DESIGNED BY: Kristie Chan DATE 4/13/16

WITNESSED BY: Ayushi Singh DATE 4/12/16

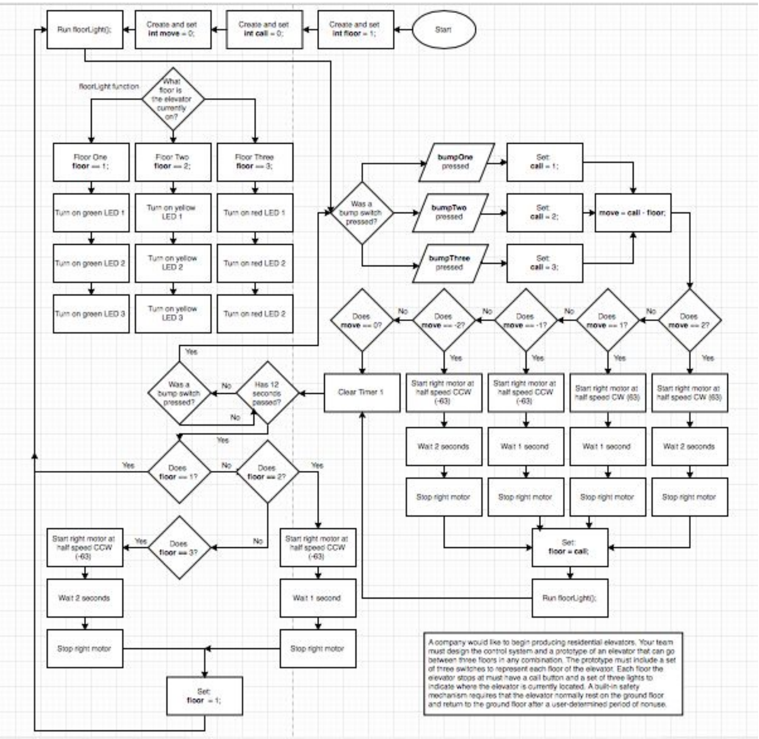
PROPRIETA
INFORMATIO

All information is the prope
and solely owned by the De

Final Physical Sketch



Final Flowchart



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.

Final Program Sketch

```
int flooor = 1; //Variable to track what floor elevator is currently on
int call = 0; //Variable to track what floor elevator is called to
int move = 0; //Variable to calculate how much elevator needs to move

void floorLight(){ //Setting floor variable/lights to track what floor the elevator is currently on
        if(flooor == 1) { //If elevator is on floor 1
                turnLEDOn(greenOne); //Then turn on the green light on floor one
                turnLEDOn(greenTwo); //Then turn on the green light on floor two
                turnLEDOn(greenThree); //Then turn on the green light on floor three
        }
        else if(flooor == 2) {
                turnLEDOn(yellowOne); //Then turn on the yellow light on floor one
                turnLEDOn(yellowTwo); //Then turn on the yellow light on floor two
                turnLEDOn(yellowThree); //Then turn on the yellow light on floor three
        }
        else if(flooor == 3) {
                turnLEDOn(redOne); //Then turn on the red light on floor one
                turnLEDOn(redTwo); //Then turn on the red light on floor two
                turnLEDOn(redThree); //Then turn on the red light on floor three
        }
}
task main()
{
        while (1 == 1){ //Loop forever

                floorLight(); //Setting lights to track what floor the elevator is currently on

                //Bump switch input to call elevator to a floor
                if(SensorValue[bumpOne] == 1) { //If the first floor bump switch is pressed
                        call = 1; //Sets call to equal the floor it was called to
                }
                else if(SensorValue[bumpTwo] == 1) { //If the two floor bump switch is pressed
                        call = 2; //Sets call to equal the floor it was called to
                }
                else if(SensorValue[bumpThree] == 1) { //If the third floor bump switch is pressed
                        call = 3; //Sets call to equal the floor it was called to
                }

                move = call - flooor; //Calculate how much elevator needs to move

                //Based on the value of move, elevator will move to floor it was called to
                if(move == 2){ //If the elevator is going 2 floors up
                        startMotor(rightMotor, -20); //Start the right motor at 20 CCW
                        wait(5); //Wait 2 seconds
                        stopMotor(rightMotor); //Stop motor after 2 seconds
                }
                else if(move == 1){
                        startMotor(rightMotor, -20); //Start the right motor at 20 CCW
                        wait(2); //Wait 1 second
                        stopMotor(rightMotor); //Stop motor after 1 second
                }
                else if(move == -1){
                        startMotor(rightMotor, 20); //Start the right motor at 20 CW
                        wait(2); //Wait 1 second
                        stopMotor(rightMotor); //Stop motor after 1 second
                }
                else if(move == -2){
                        startMotor(rightMotor, 20); //Start the right motor at 20 CW
                        wait(5); //Wait 2 seconds
                        stopMotor(rightMotor); //Stop motor after 2 seconds
                }
                else if(move == 0){
                        //Don't move
                }
                flooor = call; //Set floor to the floor it was called to and is now on
                floorLight(); //Setting lights to track what floor the elevator is currently on

                //Timer to track if elapsed time (12 seconds) has passed since elevator has moved, then
will move it back to first floor
                ClearTimer(T1);
                while(Time1[T1] <= 12000) { //While the timer is less than or equal to 12 seconds
                        if(Time1[T1] >= 12000 && SensorValue[bumpOne] && 0 &&
SensorValue[bumpTwo] == 0 && SensorValue[bumpThree] == 0){ //If 12 seconds has passed and a bump
switch hasn't been pressed
                if(flooor == 2){ //If elevator is on the second floor
                    startMotor(rightMotor, 20); //Start the right motor at 20 CW
                    wait(2); //Wait 1 second
                    stopMotor(rightMotor); //Stop motor after 1 second
                }
                else if(flooor == 3){ //If elevator is on the third floor
                    startMotor(rightMotor, 20); //Start the right motor at 20 CW
                    wait(5); //Wait 2 seconds
                    stopMotor(rightMotor); //Stop motor after 1 second
                }
                flooor = 1; //Set floor for first floor
                        }
                }
                ClearTimer(T1); //Clear timer again so it doesn't keep running; not necessary

}
}
```

Conclusion Questions

1. What was the most difficult part of the problem?
   The most difficult part of the problem was coding, because we encountered many problems that we didn’t anticipate. For example, we hadn’t thought about turning the LED off after it moved from floor to floor.
2. List and describe two features that were not part of the design problem that could be added to improve your design.
   We should have made the elevator more stable because the winch drums weren't in line with each other so it made the ropes uneven. Another change would be limiting the amount of wires used to attached the LED lights to the VEX Cortex just to make the design look nicer and less disorganized