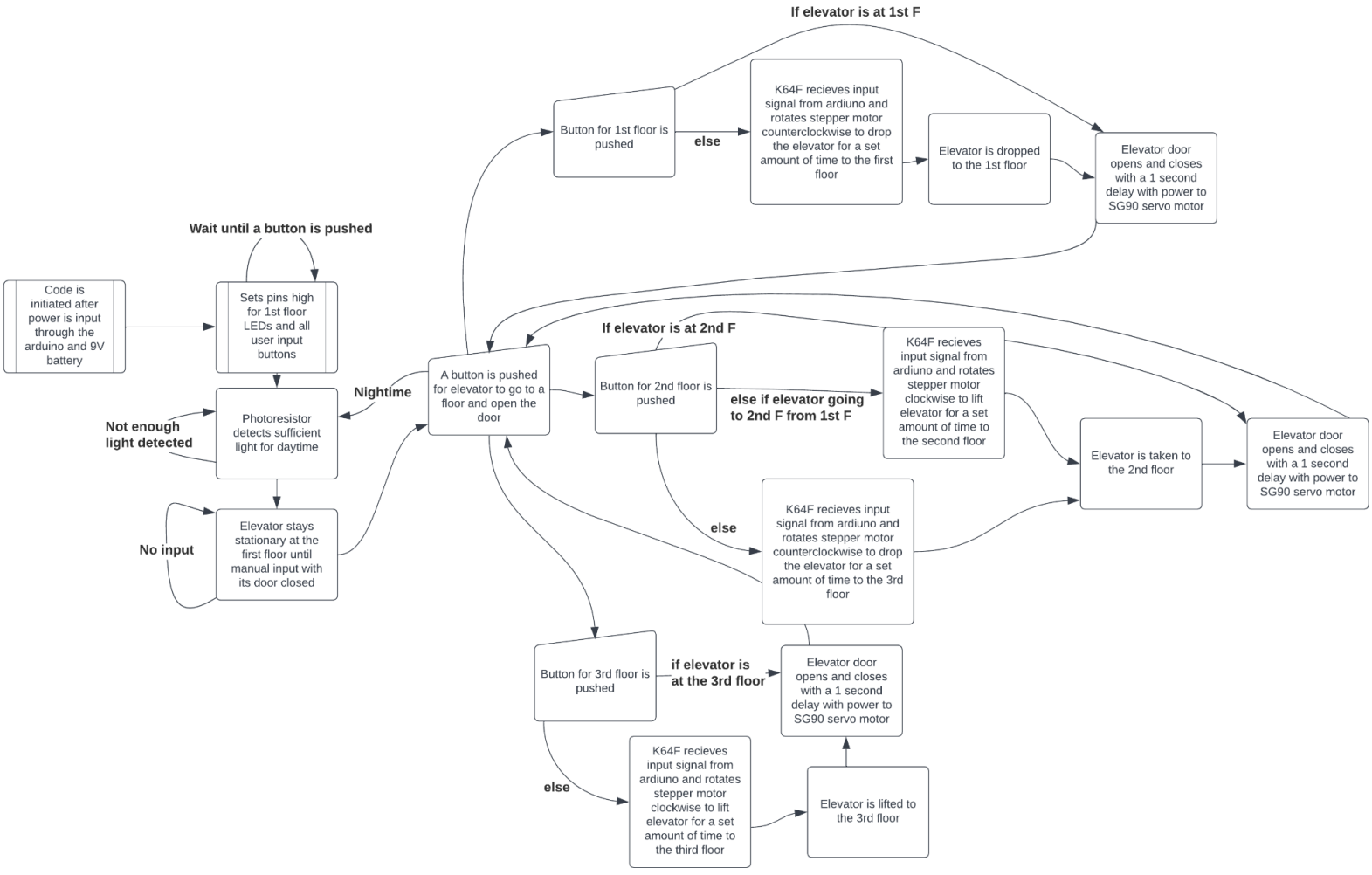
EE128 - Mini Project Report
3-Story Elevator with Light Sense
Kenghao Cai and Antonio Garcia
May 31, 2022

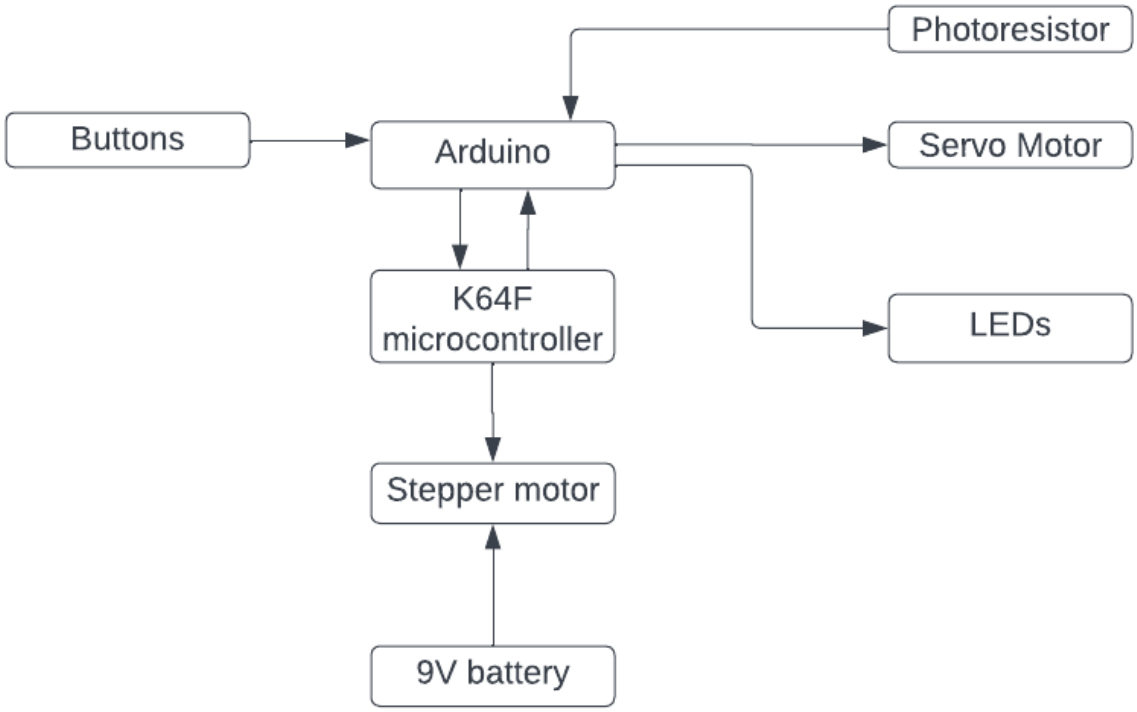
Project Description

The project that we both decided on creating was an elevator which would go up and down with user input. Our goal was to create a functioning elevator with the help of two motors, one to open the door and the other to lift the elevator itself with functioning LEDs for every floor to distinguish where the elevator is currently positioned.. We made requirements for the project as a minimum basis and guideline to start our creation process. Requirements were to have both motors capable of their job which is opening the elevator door for the SG90 servo motor, and lifting the elevator for the stepper motor.

Youtube Demo:
EE128 Mini-project Elevator

System Design





Implementation Details

The code below is for the current floor for which the LEDs are turned on, for instance if the current floor of the elevator is 3 all 3 lines of LEDs will be set to HIGH.

```
if (currfloor == 3) {
  digitalWrite(led3, HIGH);
  digitalWrite(led2, HIGH);
  digitalWrite(led1, HIGH);
}
else if (currfloor == 2) {
  digitalWrite(led2, HIGH);
  digitalWrite(led1, HIGH);
  digitalWrite(led3, LOW);
}
else if (currfloor == 1) {
  digitalWrite(led1, HIGH);
  digitalWrite(led2, LOW);
  digitalWrite(led3, LOW);
}
```

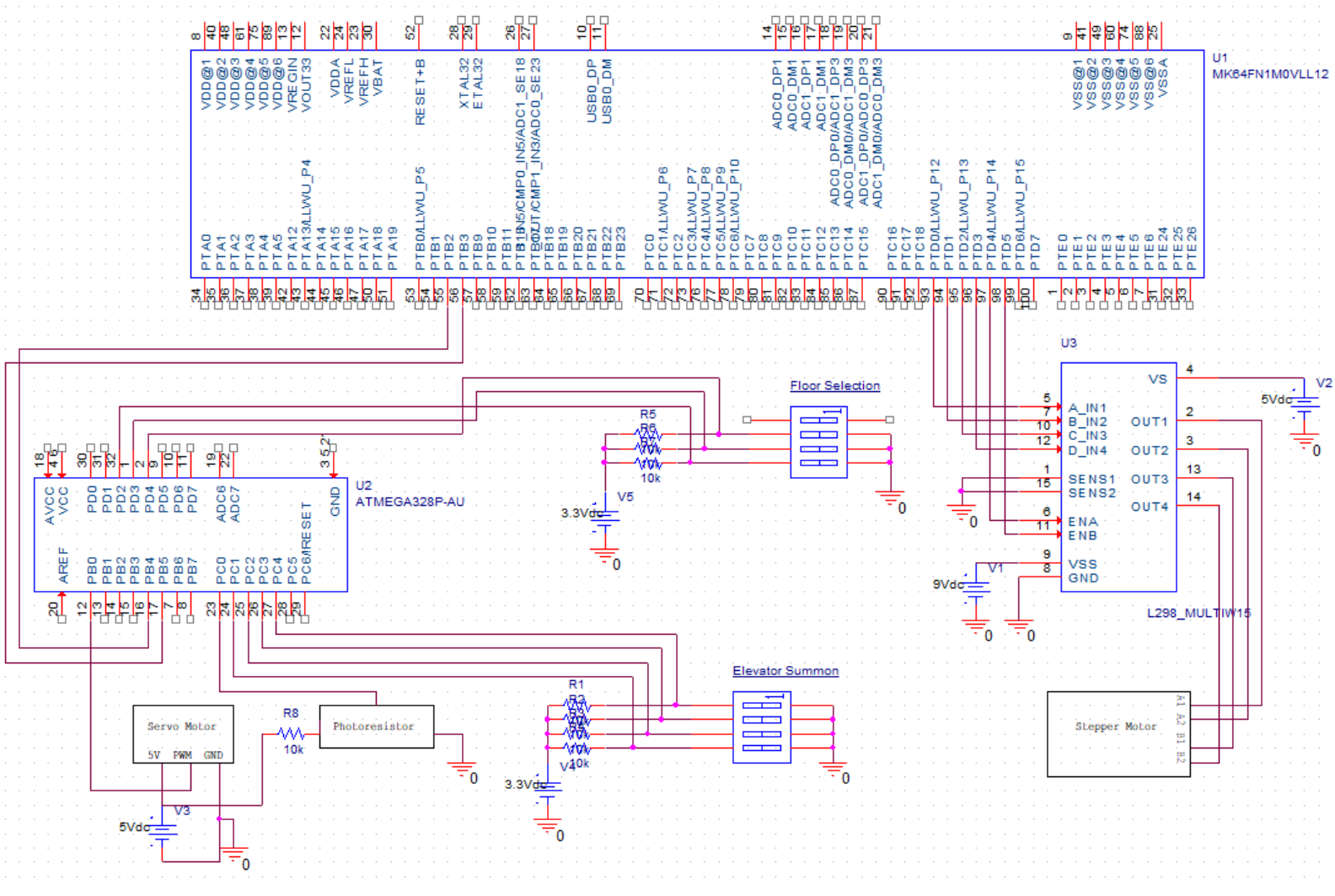
The code below is for the servo motor to rotate from 130 degrees (closed position) to 75 degrees (open position) in order to open the elevator door for 1 second before closing it, 75 → 130degrees

```
if (doorsignal == true){     //to open
    for(angle = 130; angle > 75; angle--) {
      servo.write(angle);
      delay(30);
```

```
    }
    delay(1000);   // auto close after 1 sec
    for(angle = 75; angle < 130; angle++){
      servo.write(angle);
      delay(30);
    }
    doorsignal = false;
}
else {
  servo.write(angle);
}
```

The code below is a section of the chunk of code which lets the Arduino interface with the K64F by setting pins 12 (moving the elevator down; CCW motor movement) and 13 (moving the elevator up; CW motor movement) on the Arduino to HIGH and LOW to move the stepper motor in the appropriate direction, then signaling the servo motor to open and close the door.

```
if (currfloor - targetfloor < 0) {
    while (targetfloor - currfloor > 0) {
      digitalWrite(elevatorUp, HIGH);
      delay(2900);
      digitalWrite(elevatorUp, LOW);
      currfloor++;
    }
    doorsignal = true;
}
```



Testing/Evaluation

The test environment required to utilize the elevator would be any room with adequate lighting since the photoresistor will need to pick up sufficient light in order to allow the entire system to work. The system requires a 9V input which can't be supplied by the K64F or Arduino which led us to utilize a generic 9V battery for functionality. There were various test scenarios done during the creation of the project, but mostly two specifically aimed at. Those two were if the elevator was in a bright or dark room, since the photoresistor has to determine based on the preset value to allow full elevator functionality; it was detrimental that the elevator only worked when the photoresistor determined it was "day time". Every component was working as intended, allowing the elevator to work in coordination with the floor buttons, inside buttons, and floor LEDs displaying the current position of the elevator.

Discussions

There were many challenges in the process of making the project work, some challenges were overcome by simply buying supplies from a hardware store, for instance Home Depot. A large challenge occurred in the later part of the project which was the actual elevator itself not being able to be lifted by the stepper motor because all the components added to the elevator made it too heavy for the motor to lift. The servo motor was attached underneath the inside of the elevator floor, Figure 3 shows the between portion where the servo was stored underneath, and connected to a hard plastic cutout which was connected to a cardboard door held in place with paper, cardboard and more glue. Attaching the strings to the elevator added much more weight since the elevator had to be centered with the string holding it up we decided to connect the string to 4 strings connected to the inside of the elevator

Limitations were hardware related. Since the stepper motor was unable to handle the elevator's weight we had to compromise and have the actual elevator be outside and be used as a demonstration of how it would work in full functionality. Another limitation was in fact time since the project was done over the course of two weeks the elevator could have been much more complex and designed better with a longer period of time.

Some possible improvements which can be done upon the system include having a larger motor which is capable of withstanding the weight of the elevator and its components. Soldering the wires which were connected on the inside of the elevator shaft, Fig 2, since the line for the floor 2 LEDs became weak and loose which led to the second and third floor not knowing the location of the elevator.

Conclusion

The elevator project was completed by both members in the lab group, Kenghao Cai and Antonio Garcia, the completed design can be seen in Figure 4 and in the Youtube demo link on the first page.. During the entirety of the project the work was split between both members fairly during the time spent working on the project. Both members worked evenly on the source code, pre planning the model, Home Depot shopping, the physical elevator model creation, wiring, and report.

Appendix

Arduino code

```
#include <Servo.h>

Servo servo;
int angle = 130;
bool doorsignal = false;
int currfloor = 1;
int targetfloor = 1;
const int elevatorUp = 13;
const int elevatorDown = 12;

int upSignal = 0;
int downSignal = 0;
const int floor1summon = 15;      //A1
const int floor2summonup = 16;    //A2
const int floor2summondown = 17;  //A3
const int floor3summon = 18;      //A4

int floor1signal, gofloor1signal = 0;
int floor2signalup, gofloor2signal = 0;
int floor2signaldown = 0;
int floor3signal, gofloor3signal = 0;

const int gofloor1 = 2;
const int gofloor2 = 3;
const int gofloor3 = 4;

const int led1 = 5;
const int led2 = 6;
const int led3 = 7;

int ADCvalue = 0;

void setup() {
  Serial.begin(9600);

  servo.attach(8);
  servo.write(angle);

  pinMode(elevatorUp, OUTPUT);
  pinMode(elevatorDown, OUTPUT);
```

```
  pinMode(floor1summon, INPUT);
  pinMode(floor2summonup, INPUT);
  pinMode(floor2summondown, INPUT);
  pinMode(floor3summon, INPUT);

  pinMode(gofloor1, INPUT);
  pinMode(gofloor2, INPUT);
  pinMode(gofloor3, INPUT);

  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
}

void loop() {
  ADCvalue = analogRead(A0);

if (ADCvalue > 600) {
//********** Floor level display **********
  if (currfloor == 3) {
    digitalWrite(led3, HIGH);
    digitalWrite(led2, HIGH);
    digitalWrite(led1, HIGH);
  }
  else if (currfloor == 2) {
    digitalWrite(led2, HIGH);
    digitalWrite(led1, HIGH);
    digitalWrite(led3, LOW);
  }
  else if (currfloor == 1) {
    digitalWrite(led1, HIGH);
    digitalWrite(led2, LOW);
    digitalWrite(led3, LOW);
  }


//************* OPEN ELEVATOR DOOR *************
  if (doorsignal == true){     //to open
      for(angle = 130; angle > 75; angle--) {
        servo.write(angle);
        delay(30);
      }
      delay(1000);   // auto close after 1 sec
      for(angle = 75; angle < 130; angle++){
        servo.write(angle);
        delay(30);
      }
      doorsignal = false;
  }
  else {
    servo.write(angle);
  }
//*************    END   *************

//********* Elevator Summon *********
```

```
  floor1signal = digitalRead(floor1summon);
  floor2signalup = digitalRead(floor2summonup);
  floor2signaldown = digitalRead(floor2summondown);
  floor3signal = digitalRead(floor3summon);

  gofloor1signal = digitalRead(gofloor1);
  gofloor2signal = digitalRead(gofloor2);
  gofloor3signal = digitalRead(gofloor3);
  delay(100);

  if (floor1signal == LOW || gofloor1signal == LOW) {
    targetfloor = 1;
  }
  else if (floor2signaldown == LOW || gofloor2signal == LOW) {
    targetfloor = 2;
  }
  else if (floor2signalup == LOW || gofloor2signal == LOW) {
    targetfloor = 2;
  }
  else if (floor3signal == LOW || gofloor3signal == LOW) {
    targetfloor = 3;
  }
  if (currfloor - targetfloor < 0) {
      while (targetfloor - currfloor > 0) {
        digitalWrite(elevatorUp, HIGH);
        delay(2900);
        digitalWrite(elevatorUp, LOW);
        currfloor++;
      }
      doorsignal = true;
  }
  else if (currfloor - targetfloor > 0) {
    while ((currfloor - targetfloor) > 0) {
      digitalWrite(elevatorDown, HIGH);
      delay(2900);
      digitalWrite(elevatorDown, LOW);
      currfloor--;
    }
    doorsignal = true;
  }
  else if (currfloor == targetfloor && (floor1signal == LOW || floor2signalup == LOW
|| floor2signaldown == LOW || floor3signal == LOW)) {
    doorsignal = true;
  }
  else {
    doorsignal = false;
  }
//*************    END   *************
}
else {
  delay(100);
}
}
```

K64F code

```
#include "fsl_device_registers.h"

uint32_t ROT_CW = 0, ROT_CCW = 0;
static int i = 0;

int main(void)
{
	SIM_SCGC5 |= SIM_SCGC5_PORTB_MASK; /*Enable Port B Clock Gate Control*/
	SIM_SCGC5 |= SIM_SCGC5_PORTD_MASK; /*Enable Port D Clock Gate Control*/

	PORTB_GPCLR = 0x000C0100;	//GPIO for Port B
	PORTD_GPCLR = 0x003F0100;	//GPIO for Port D

	//Configure Port D Pins 0-5 for Output;
	GPIOD_PDDR |= (1 << 0);
	GPIOD_PDDR |= (1 << 1);
	GPIOD_PDDR |= (1 << 2);
	GPIOD_PDDR |= (1 << 3);
	GPIOD_PDDR |= (1 << 4);
	GPIOD_PDDR |= (1 << 5);

	//Set PB[3:2] for input;
	GPIOB_PDDR |= (0 << 2);
	GPIOB_PDDR |= (0 << 3);

	unsigned int delay = 9000;

	while(1) {
		ROT_CW = GPIOB_PDIR & 0x04;
		ROT_CCW = GPIOB_PDIR & 0x08;

		if (ROT_CW && !ROT_CCW) {
			GPIOD_PDOR = 0x36;
			for(i = 0; i < delay; i++);
			GPIOD_PDOR = 0x35;
			for(i = 0; i < delay; i++);
			GPIOD_PDOR = 0x39;
			for(i = 0; i < delay; i++);
			GPIOD_PDOR = 0x3A;
			for(i = 0; i < delay; i++);
		}
		else if (!ROT_CW && ROT_CCW) {
			GPIOD_PDOR = 0x3A;
			for(i = 0; i < delay; i++);
			GPIOD_PDOR = 0x39;
			for(i = 0; i < delay; i++);
			GPIOD_PDOR = 0x35;
			for(i = 0; i < delay; i++);
			GPIOD_PDOR = 0x36;
			for(i = 0; i < delay; i++);
		}
	}
	return 0;
}
```

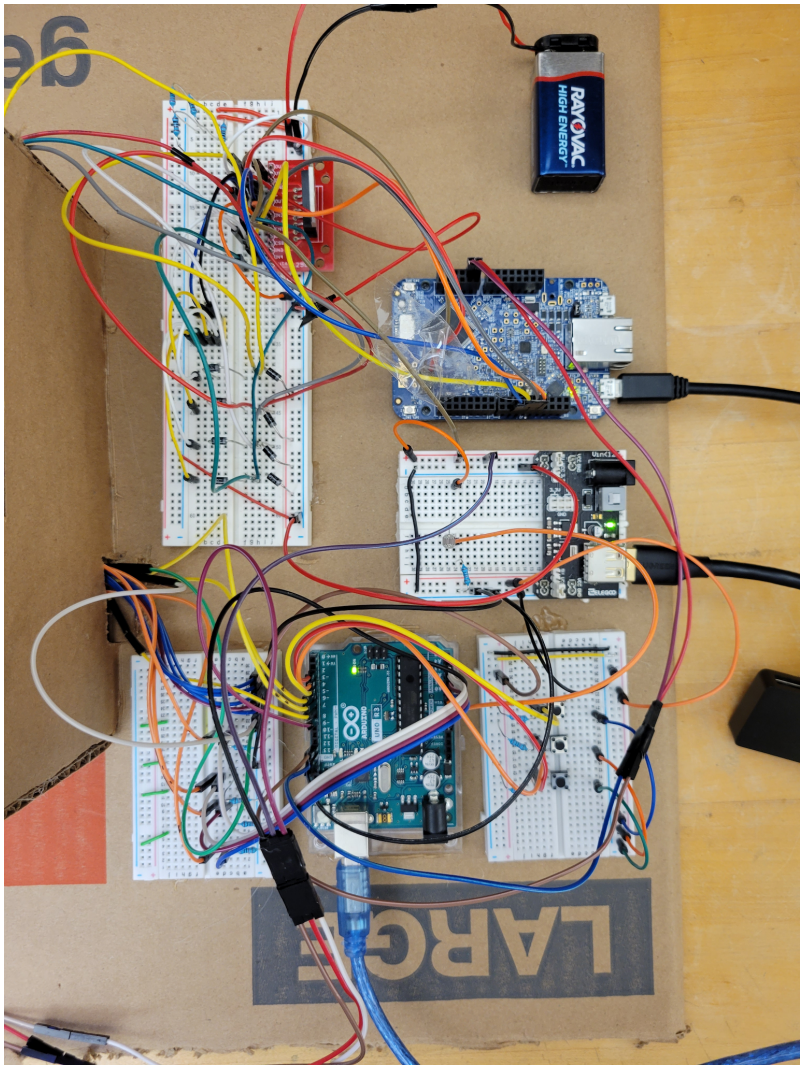


Figure 1: Wiring layout for the Arduino and K64F

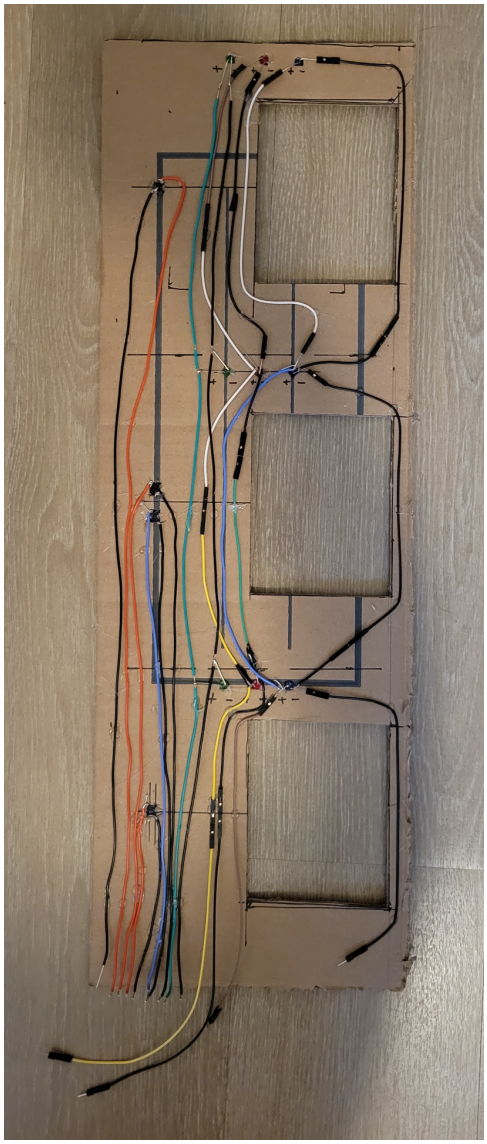


Figure 2: The inside of the elevator shaft comprising of the LED wiring along the wall

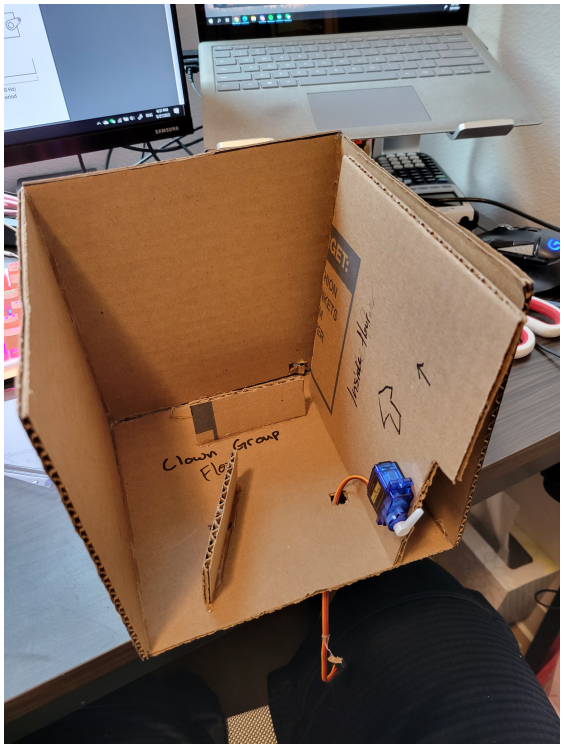


Figure 3: The inside of the elevator with the SG90 Servo motor in place to operate the door

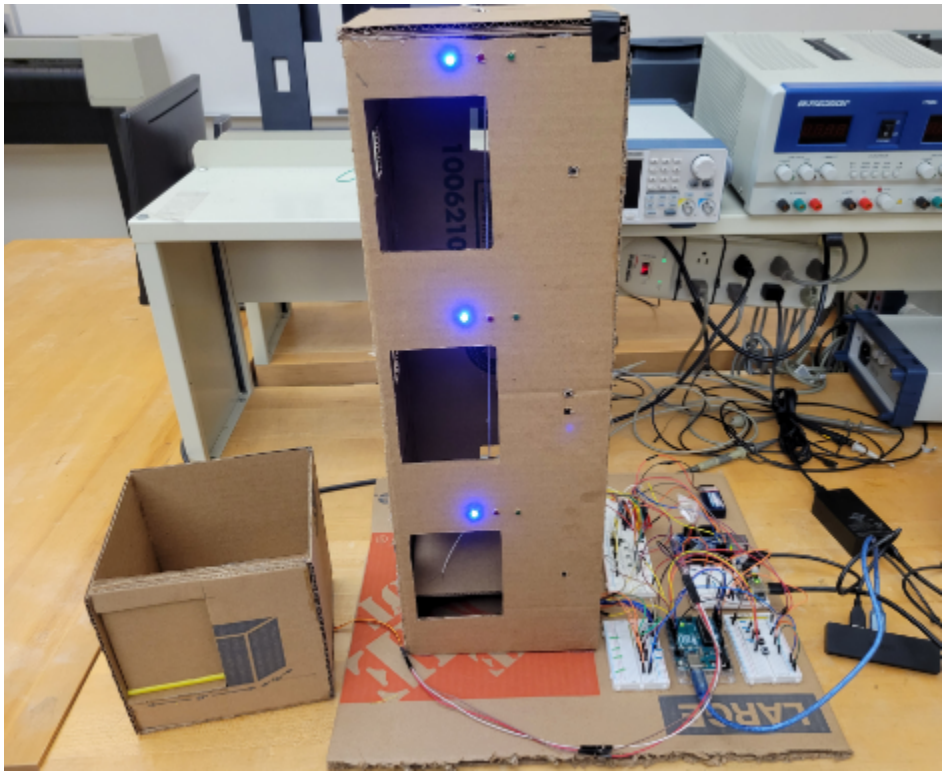


Figure 4: Completed Project