창의SW기초설계

2020 Touchless Elevator

9조 최종발표

스마트기기공학 17011781 박현우
스마트기기공학 17011805 김건우
스마트기기공학 17011817 신도현
스마트기기공학 17011824 구범준

INDEX

Touchless Elevator

01

Review

Motivation


잠깐만!
엘리베이터 버튼을 누른 후,
반드시 손소독제를 사용하거나
흐르는 물에 손을 씻어주세요!
광진구 재난안전대책본부


14
13
12
11
10
9
B1

프로젝트의 목표

1. 기존 엘리베이터의 버튼 PUSH 방식을 변경하여 버튼을 신체적 접촉 없이 작동시킨다.
2. 모터를 이용해 버튼과 통신 가능한 엘리베이터를 구현한다.

Touchless Elevator는 왜 필요한가?

뉴스홈 | 최신기사

현대엘리베이터, 승강기 업계 첫 'IDEA 디자인 어워드

승고시간 | 2018-10-02 10:07

(서울=연합뉴스) 이승관 기자 = 현대엘리베이터[01780]는 미국 산업디자인협회(IDSA)가 주관하는 '2018 IDEA(International Design Excellence Awards)'에서 본상인 파이널리스트(Finalist)를 받았다고 2일 밝혔다.

레드닷 디자인 어워드, iF 디자인 어워드와 함께 '세계 3대 디자인 어워드'로 꼽히는 IDEA에서 엘리베이터 기업이 수상한 것은 이번이 처음이라고 회사 측은 강조했다.

현대엘리베이터가 출품한 '유니버설 멀티 바'는 기존 호출 버튼과 함께 성인 무릎 높이에 있는 '무릎 버튼', 센서 감지 구역에 발을 대면 엘리베이터를 호출하는 '터치리스 풋 버튼' 등을 함께 적용했다.

물건을 든 사용자나 키가 작은 어린이, 휠체어 이용자 등을 배려한 것으로, 엘리베이터 상하 이동 방향을 알려주는 홀랜턴도 입체형으로 디자인해 사용 편의성을 높였다.

회사 관계자는 "2017년 업계 최초로 iF 디자인 어워드 금상을 받은 데 이어 이번 수상으로 디자인 경쟁력을 또다시 인정받았다"면서 "엘리베이터의 본질이 이동을 편리하게 하는 수단이자 다양한 사람이 이용하는 공공재라는 두가지 측면을 감안한 디자인"이라고 말했다.

미국 IDEA 디자인 어워드에서 본상을 받은 현대엘리베이터의 '유니버설 멀티 바'. [현대엘리베이터 제공=연합뉴스]

[이슈톡] "손 대기 찝찝해서"...발로 움직이는 엘리베이터까지 등장

입력 2020-05-21 06:53 | 수정 2020-05-21 09:54

엘리베이터 버튼 안 눌러도 OK... 日기업들, '터치리스' 제품 개발중

5,814 읽음 · 2020.04.03

한경BUSINESS

[포스트 코로나 유망 비즈니스 22] "얼굴이 출입증" 손 안 대는 '터치리스' 뜬다

기사입력 2020.05.19. 오전 10:14 최종수정 2020.05.19. 오전 11:22

[커버스토리 = 포스트 코로나 유망 비즈니스 22선]
19. 터치리스

02

구현 과정

버튼 박스

엘리베이터 모형

버튼 - 도트 매트릭스 연결

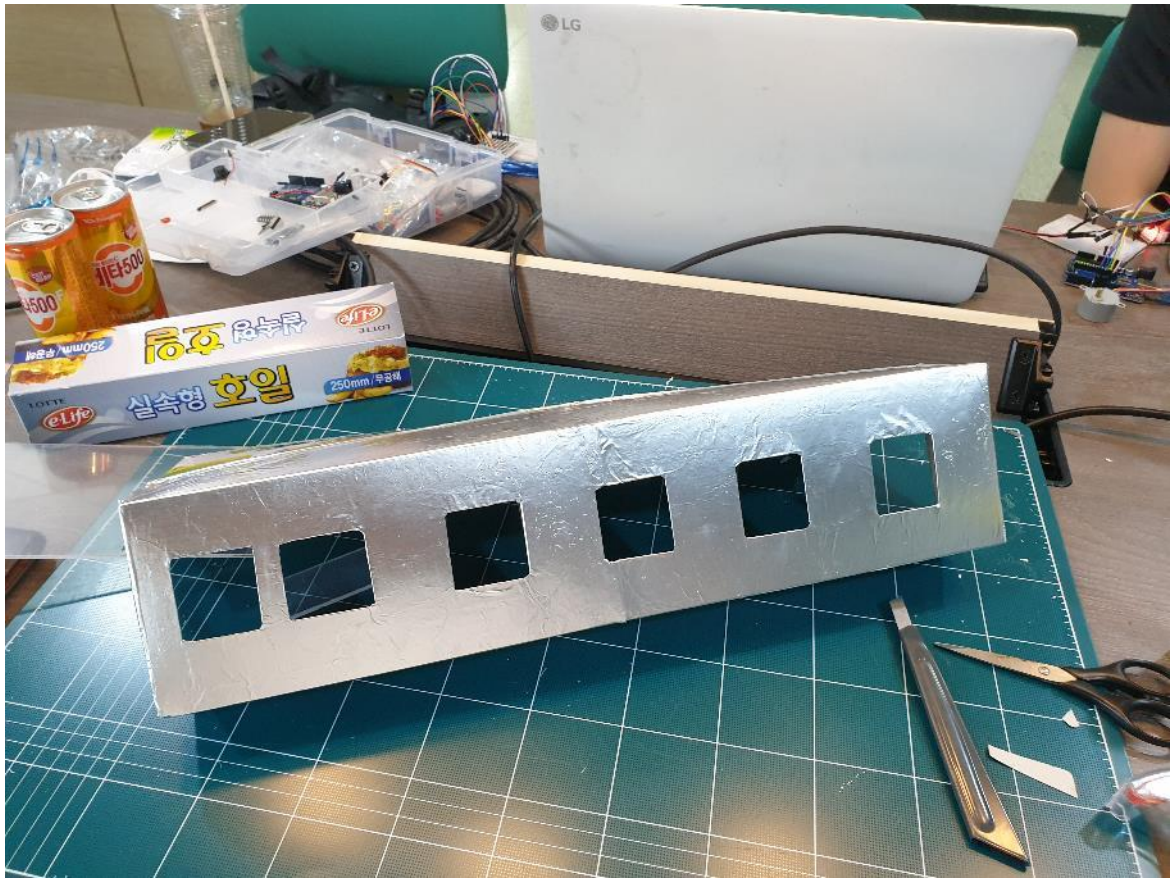
버튼 - 엘리베이터 연결

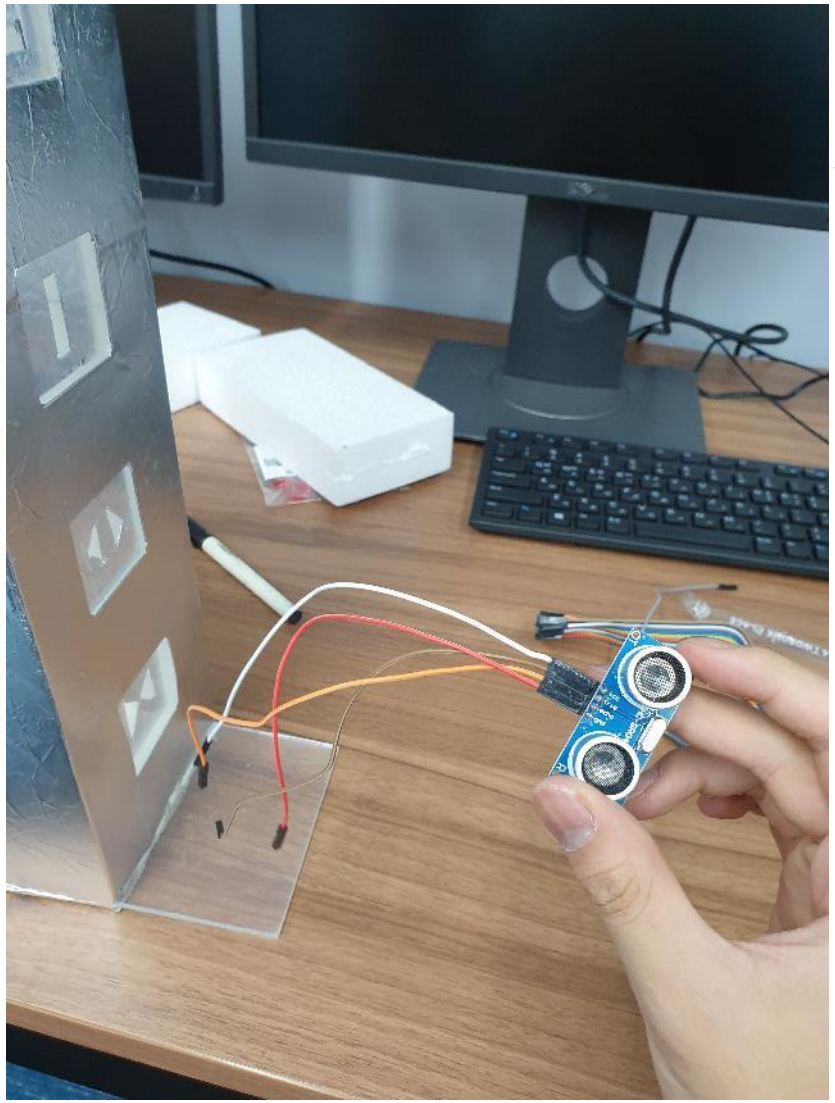
How to make?

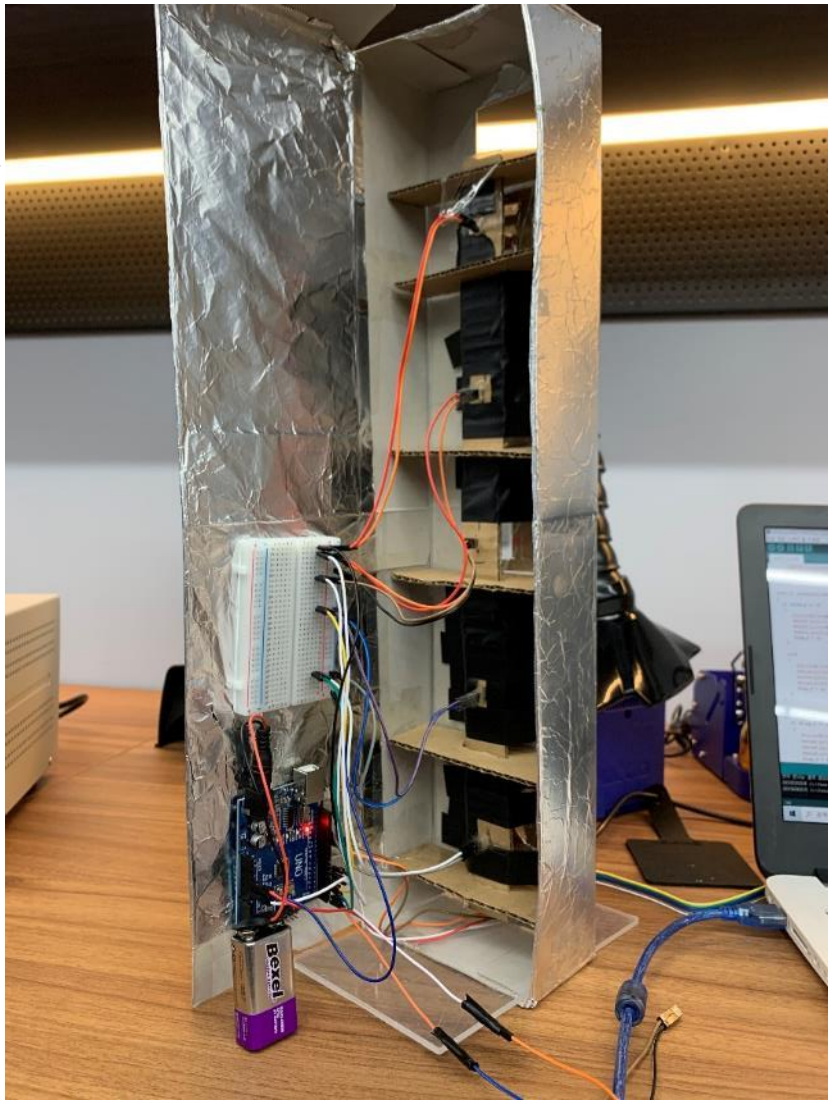
Button Box

버튼 박스 제작 과정

버튼 박스 제작 과정






Bexel

제작 과정에서의 조건 설정하기

1. 취소 버튼의 구현과 방법 ?
   : 동일한 버튼이 2번 인식되면 LED가 OFF
   하지만, 이동 중 2번 눌러 취소하고자 하더라도 해당 층으로 이동하는 것으로 전제

2. 여러 개의 버튼을 동시에 누른다면?
   : 손끝 ~ 초음파 센서 간 거리를 측정하기 때문에, 동시 누르는 경우는 없다고 전제
   (명령을 차례대로 입력해야 함)

초음파 센서로부터의 정량적 거리

초음파 센서 ~ 각 버튼까지의 정량적 거리

닫힘 : 3.0cm~8.0cm
열림 : 11.0cm~16.0cm
1층 : 19.0cm~24.0cm
2층 : 27.0cm~32.0cm
3층 : 35.0cm~40.0cm

그 외 : 측정 결과 반영 X

버튼 박스 주요 코드

```
34 void loop() {
35
36   int duration, distance;
37   digitalWrite(TRIGGER_PIN, HIGH);
38   delay(1000);
39   digitalWrite(TRIGGER_PIN, LOW);
40   duration = pulseIn(ECHO_PIN, HIGH);
41
42   distance = duration / 58;
43   if (distance >= 4000 || distance <= 0)
44   {
45     Serial.println("Out of range");  //check
46   }
47
48   else
49   led_operation(distance);
50 }
```

버튼 박스의 주요 코드

```
65 void led_operation(int distance)
66 {
67    if (distance>=3 && distance<=8) //닫힘버튼
68    {
69      if (flag_close == 0) //닫힐 경우
70      {
71         digitalWrite(LEDCLOSE_PIN, HIGH);
72         Serial.println("닫힘버튼 입력됨");
73         Serial.print(distance);
74         Serial.println(" cm");
75         flag_close = 1;
76         Serial.println("");
77      }
78
79      else // flag_close = 1 경우
80      {
81         digitalWrite(LEDCLOSE_PIN, LOW);
82         Serial.println("닫힘버튼 취소됨");
83         Serial.print(distance);
84         Serial.println(" cm");
85         flag_close = 0;
86      }
87   }
```

☞ 열림(닫힘) 버튼의 코드 경우,
flag_open(close) 변수 선언

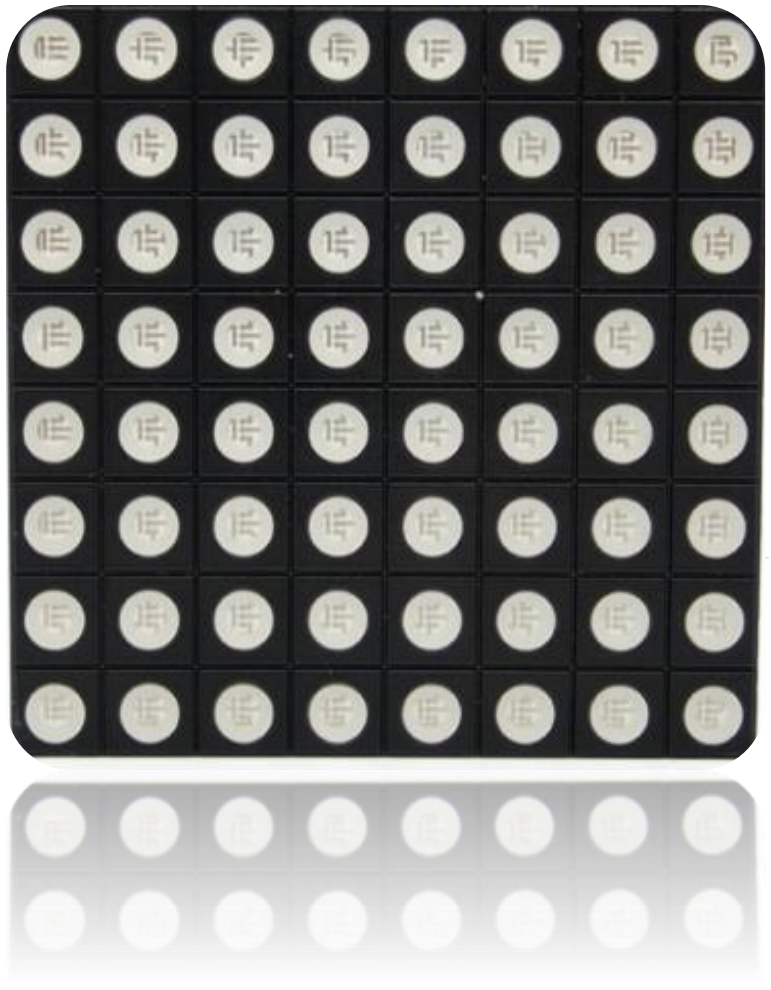
1, 2 3층 각각의 flag_[] 변수 이용
flag = 0 : LED OFF → ON
flag = 1 : LED ON → OFF

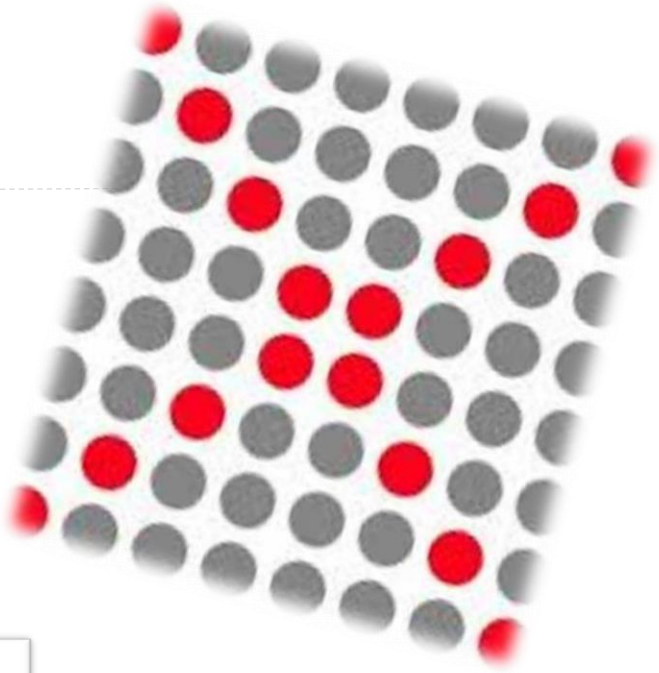
버튼 박스의 주요 코드

```
97  else if (distance>=18.5 && distance<=22.5) //1층버튼
98  {
99    if (flag_1 == 0)
100    {
101       Serial.write('x'); //다른 아두이노에 통신
102       digitalWrite(LED1_PIN, HIGH);
103       Serial.println();
104   //    Serial.println("1층 입력됨");
105   //    Serial.print(distance);
106   //    Serial.println(" cm");
107       flag_1 = 1;
108    }
109
110    else
111    {
112       Serial.write('x'); //다른 아두이노에 통신
113       digitalWrite(LED1_PIN, LOW);
114     //  Serial.println("1층 취소됨");
115     //  Serial.print(distance);
116     //  Serial.println(" cm");
117       flag_1 = 0;
118    }
119 }
```

다른 아두이노에 통신하게 위한
시리얼 모니터 출력 (1층: x, 2층: y,3층: z)
☞ UART통신 이용(추후 설명)

Dot Matrix





Dot Matrix ☞ 입력 층 수 표시

☞ 입력 받은 층을 인식하여

해당 층으로 이동중이라는 것을

도트 매트릭스에 표시

1층으로 이동중

2층으로 이동중

3층으로 이동중

도트 매트릭스의 주요 코드

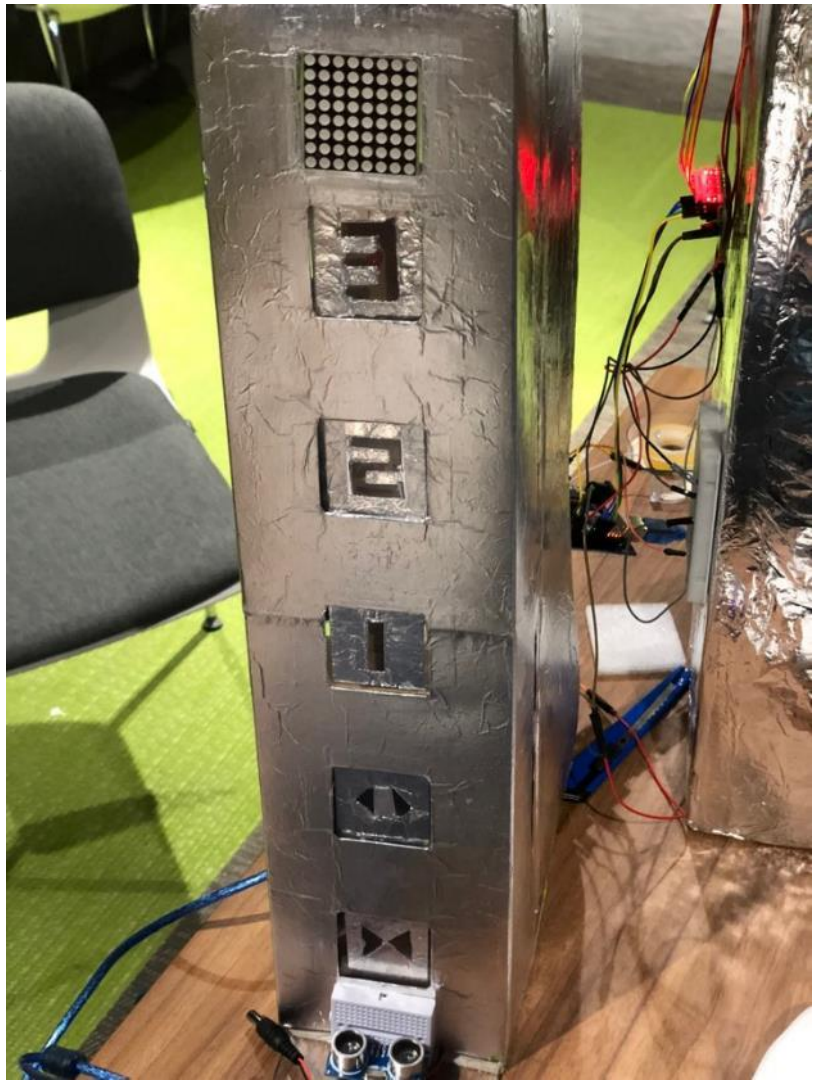
```
110  else if (distance>=19 && distance<=24) //1층버튼
111  {
112    if (flag_1 == 0)
113     {
114        Serial.write('x'); //다른 아두이노에 통신
115        digitalWrite(LED1_PIN, HIGH);
116        Dot_Matrix_UNO (0, 1);
117        Serial.println();
118    //    Serial.println("1층 입력됨");
119    //    Serial.print(distance);
120    //    Serial.println(" cm");
121        flag_1 = 1;
122     }
```

void Dot_Matrix_UNO(dot1, dot2)
함수로 **전송**하여 조건 판단 및 실행
Ex) (0,0) : nothing(X)
(0,1) : 1층
(1,0) : 2층
(1,1) : 3층

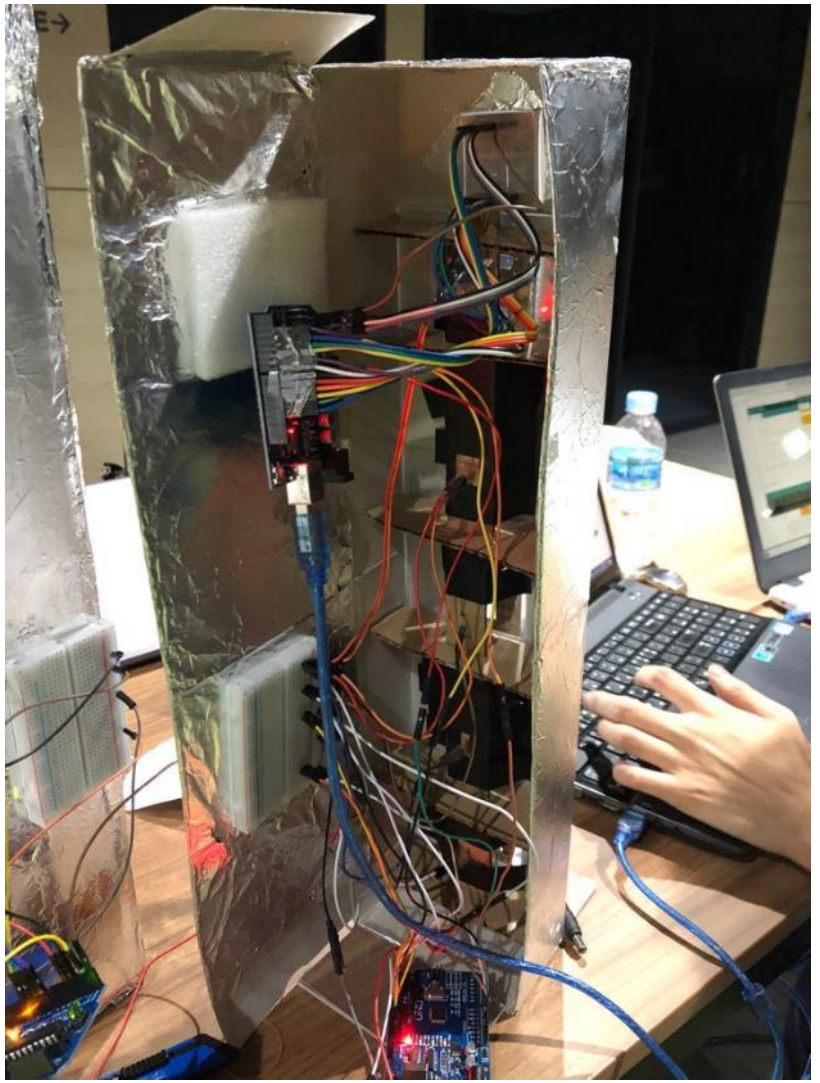
도트 매트릭스의 주요 코드

```
191 void Dot_Matrix_UNO (int dot1, int dot2)
192 {
193   if(tmpdot1 != dot1 || tmpdot2 != dot2)
194   {
195   tmpdot1 = dot1;
196   tmpdot2 = dot2;
197
198   digitalWrite(DOTPIN_1, dot1);
199   digitalWrite(DOTPIN_2, dot2);
200
201   Serial.print(dot1);
202   Serial.print(dot2);
203   Serial.println(" = 전송된 이진수");
204   }
205
206   else
207   {
208     Serial.println("이미 전송된 데이터");
209     Serial.print(dot1);
210     Serial.println(dot2);
211
212   }
213 }
```

버튼 박스 최종 모습



앞면

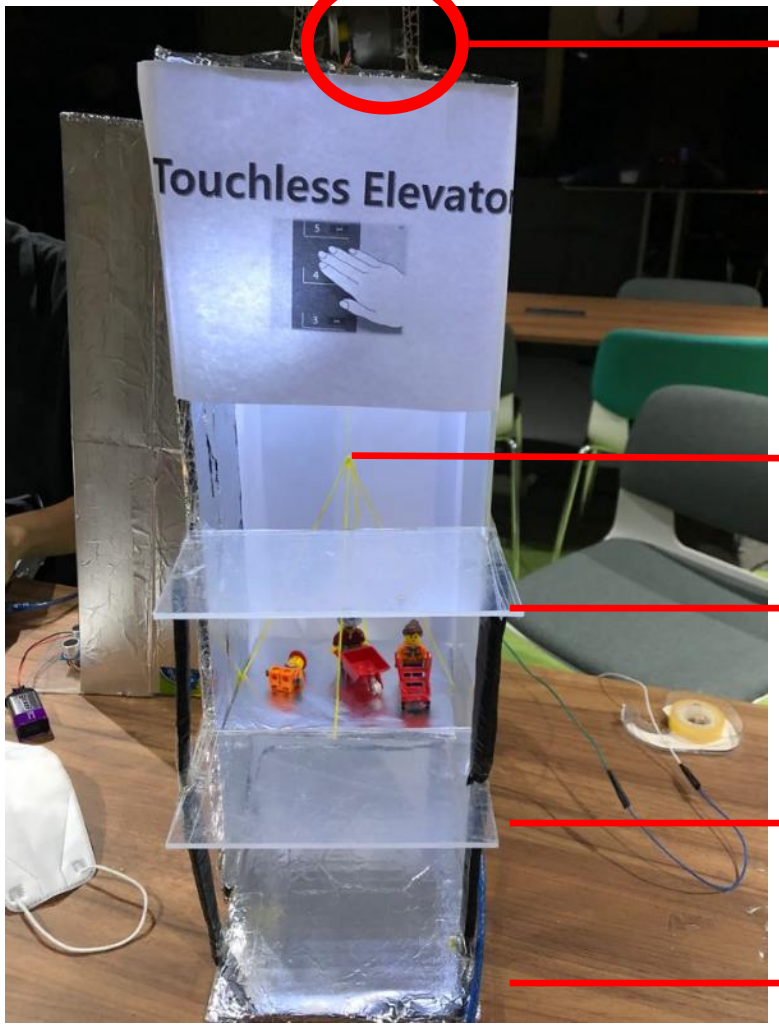


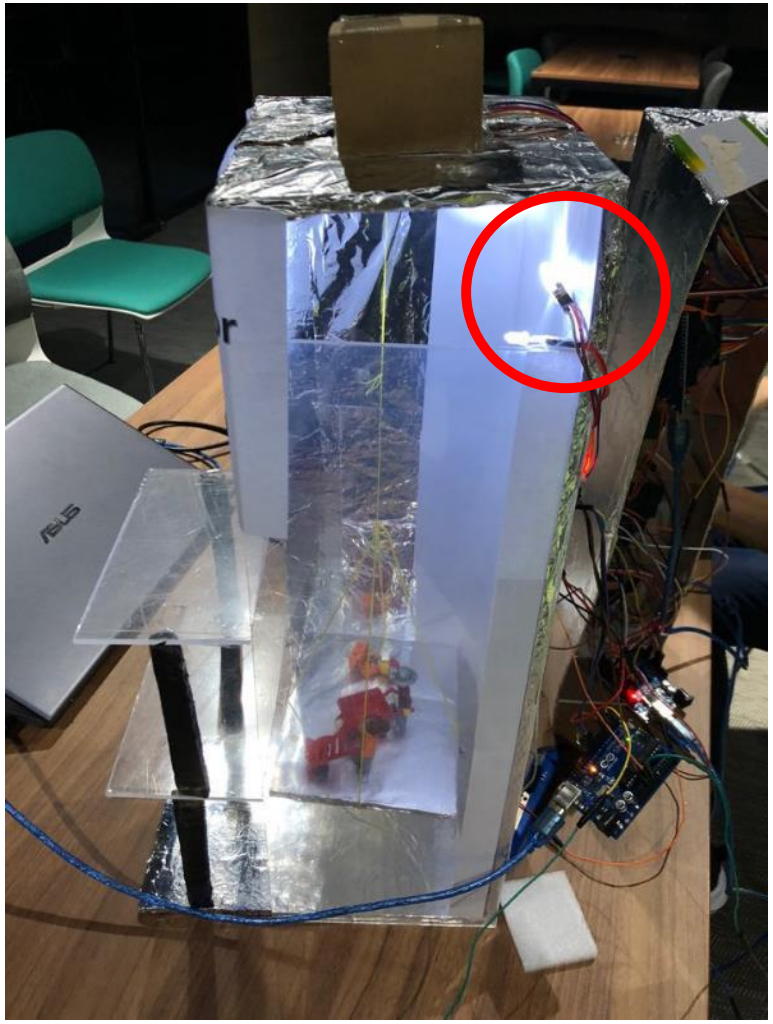
뒷면

엘리베이터 구현과 작동 원리

스텝 모터를 이용

엘리베이터 승강기 최종 모습





백색 LED 설치
☞ 내부를 환하게

엘리베이터 승강기 작동 코드

```
#define IN1  8
#define IN2  9
#define IN3  10
#define IN4  11
#define LED1_PIN 3
#define LED2_PIN 4
#define LED3_PIN 5

int             Steps = 0;
int             steps_left;
boolean         Direction = true;
unsigned long   last_time;
unsigned long   currentMillis ;
long            time;

int tmp; //입력층수
int ele_floor = 1; //현재 엘리베이터가 위치한 층수

void setup() {
  Serial.begin(9600);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(LED1_PIN, OUTPUT); //엘리베이터 조명
  pinMode(LED2_PIN, OUTPUT); //엘리베이터 조명
  pinMode(LED3_PIN, OUTPUT); //엘리베이터 조명
}
```

현재 층수와 입력한 층수정보로
엘리베이터 작동

엘리베이터 승강기 작동 코드

```
30 void loop() {
31
32   digitalWrite(LED1_PIN, HIGH); //엘리베이터 내부 조명 켜기
33   digitalWrite(LED2_PIN, HIGH); //엘리베이터 내부 조명 켜기
34   digitalWrite(LED3_PIN, HIGH); //엘리베이터 내부 조명 켜기
35
36   if (Serial.available()){ //UART통신으로 받아들인 정보가 있으면
37   char data = Serial.read(); //data변수 선언해서 읽어들인 정보를 data에 저장
```

엘리베이터 승강기 작동 코드

```
87  if (data== 'y') //받아들인 data가 y(2층)면 2층 버튼이 입력됐다고 판단함
88  {
89    tmp=2; //입력 층수 변수에 2대입
90    Serial.println(data); //시리얼 모니터에 받아들인 문자 출력
91
92    if (tmp> ele_floor) //입력한 층수가 현재 위치한 엘리베이터보다 높은 층수이면
93    {
94      steps_left = 4096*5*(tmp - ele_floor); //일정 스텝 수만큼 스텝모터를 돌림
95
96      while(steps_left>0)
97      {
98        Direction = true; //모터의 방향은 실이 감기는 방향(즉, 올라가는 방향)
99        currentMillis = micros();
100       if(currentMillis-last_time>=1000)
101       {
102         stepper(1);
103         time=time+micros()-last_time;
104         last_time=micros();
105         steps_left--;
106       }
107     } // while
108
109     ele_floor = tmp; //엘리베이터가 다 작동하면 ele_floor 변수(현재 엘리베이터 층수)를 도착한 층 정보로 초기화
110
111   }
```

엘리베이터 승강기 작동 코드

```
112     else if (tmp < ele_floor) //입력한 층수보다 엘리베이터가 더 높은 층에 있는 경우
113     {
114       steps_left = 4096*5*(ele_floor - tmp); //일정 스텝 수만큼 스텝모터를 돌림
115       while(steps_left>0)
116       {
117         Direction = false; //모터의 방향은 실이 풀리는 방향(즉, 내려가는 방향)
118         currentMillis = micros();
119         if(currentMillis-last_time>=1000)
120         {
121           stepper(1);
122           time=time+micros()-last_time;
123           last_time=micros();
124           steps_left--;
125         }
126       } // while
127
128       ele_floor = tmp; //엘리베이터가 다 작동하면 ele_floor 변수(현재 엘리베이터 층수)를 도착한 층 정보로 초기화
129
130     }
131   }
```

엘리베이터 승강기 작동 코드

```
180 void stepper(int xw) //스텝 당 4개 핀 제어 신호 출력 값 지정하는 함수
181 {
182  for (int x=0;x<xw;x++)
183  {
184    switch(Steps)
185    {
186      case 0:  runStep(LOW, LOW, LOW, HIGH);   break;
187      case 1:  runStep(LOW, LOW, HIGH, HIGH);   break;
188      case 2:  runStep(LOW, LOW, HIGH, LOW);   break;
189      case 3:  runStep(LOW, HIGH, HIGH, LOW);   break;
190      case 4:  runStep(LOW, HIGH, LOW, LOW);   break;
191      case 5:  runStep(HIGH, HIGH, LOW, LOW);   break;
192      case 6:  runStep(HIGH, LOW, LOW, LOW);   break;
193      case 7:  runStep(HIGH, LOW, LOW, HIGH);   break;
194      default:  runStep(LOW, LOW, LOW, LOW);   break;
195    }
196    SetDirection();
197   }
198
199 }
```

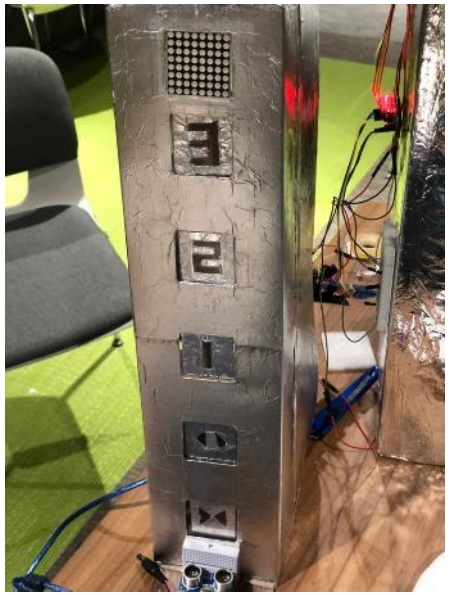
엘리베이터 승강기 작동 코드

```
200 void runStep(int value1, int value2, int value3, int value4) //4개 핀 제어 신호 출력
201 {
202  digitalWrite(IN1, value1);
203  digitalWrite(IN2, value2);
204  digitalWrite(IN3, value3);
205  digitalWrite(IN4, value4);
206 }
207
208 void SetDirection() //방향 전환 및 스텝 변수 초기화
209 {
210  if(Direction==1)  { Steps++; }
211  if(Direction==0) { Steps--; }
212  if(Steps>7) { Steps=0; }
213  if(Steps<0) { Steps=7; }
214 }
```

버튼-엘리베이터의 연결

버튼 박스와 엘리베이터 승강기와의 연결 방법 : UART통신방법 사용

송신부 RX



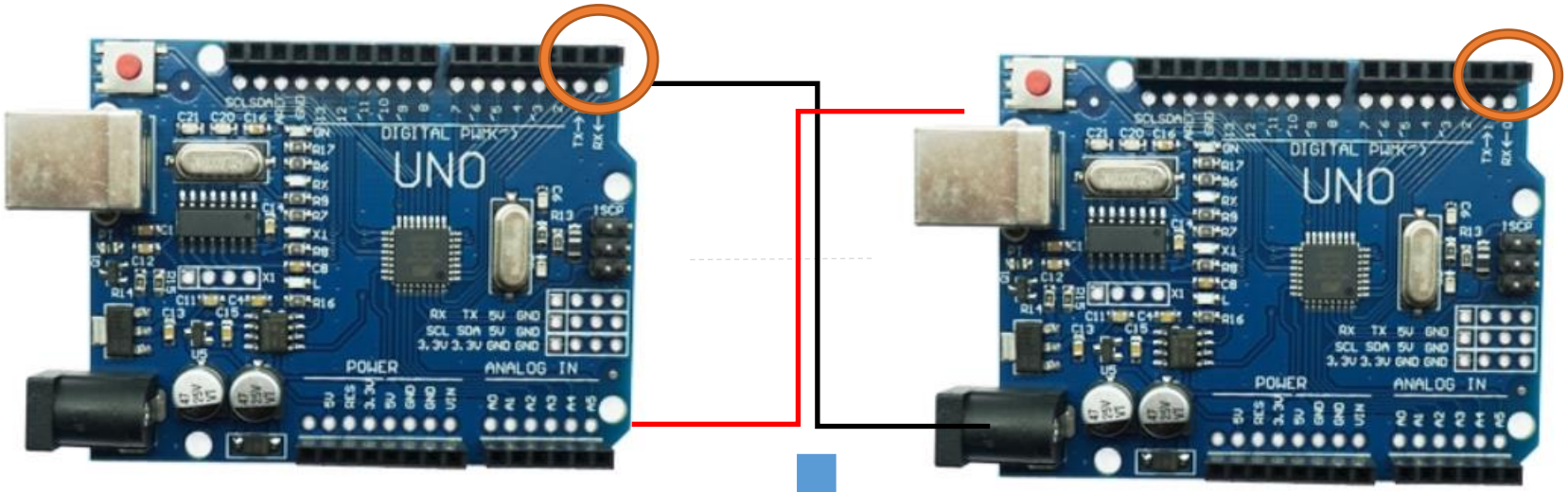
수신부 TX

UART통신

UART(Universal Asynchronous Receiver Transmitter)
- 2개의 아두이노 보드를 연결하여 동시에 통신하는 방법

조건 :
3개 선의 상호 연결 : RX 신호선 + TX 신호선
+ GND(전압 레벨을 맞추기 위함)

UART통신

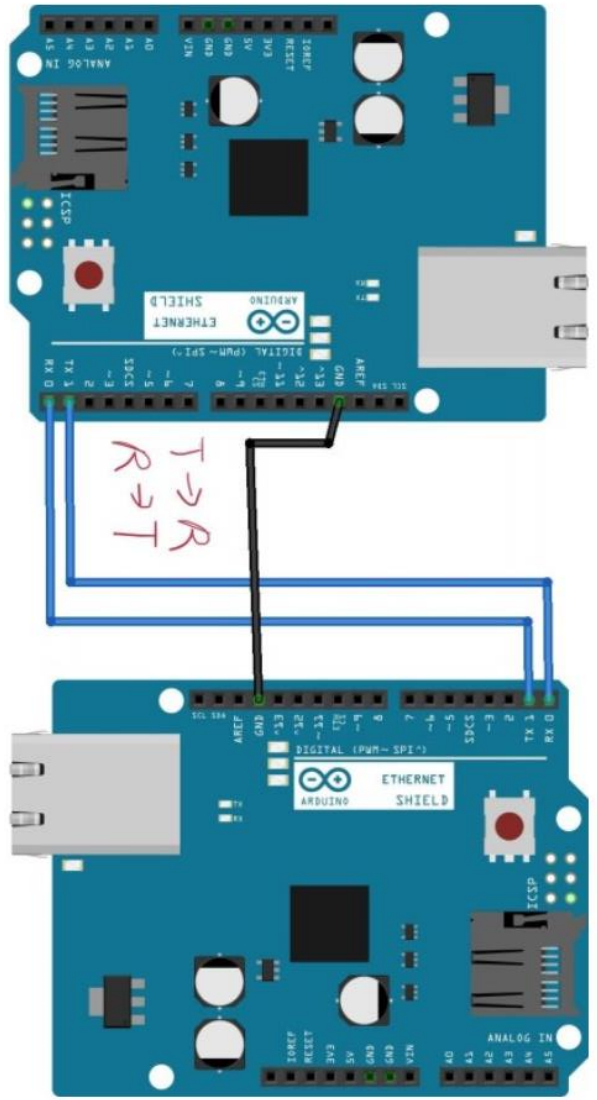


1번 아두이노

2번 아두이노

RX와 TX선 교차
+ GND 연결

☞ 데이터 전송 가능



UART 코드 구현

1 층의 정보가 이동되는 예시

```
else if (distance>=19 && distance<=24) //1층버튼
{
  if (flag_1 == 0)
   {
      Serial.write('x'); //다른 아두이노에 통신
      digitalWrite(LED1_PIN, HIGH);
      Dot_Matrix_UNO (0, 1);
      Serial.println();
  //    Serial.println("1층 입력됨");
  //    Serial.print(distance);
  //    Serial.println(" cm");
      flag_1 = 1;
   }
```

송신부 RX에서 1층에 해당되는 정보 ‘x’를
수신부 TX로 넘겨줌

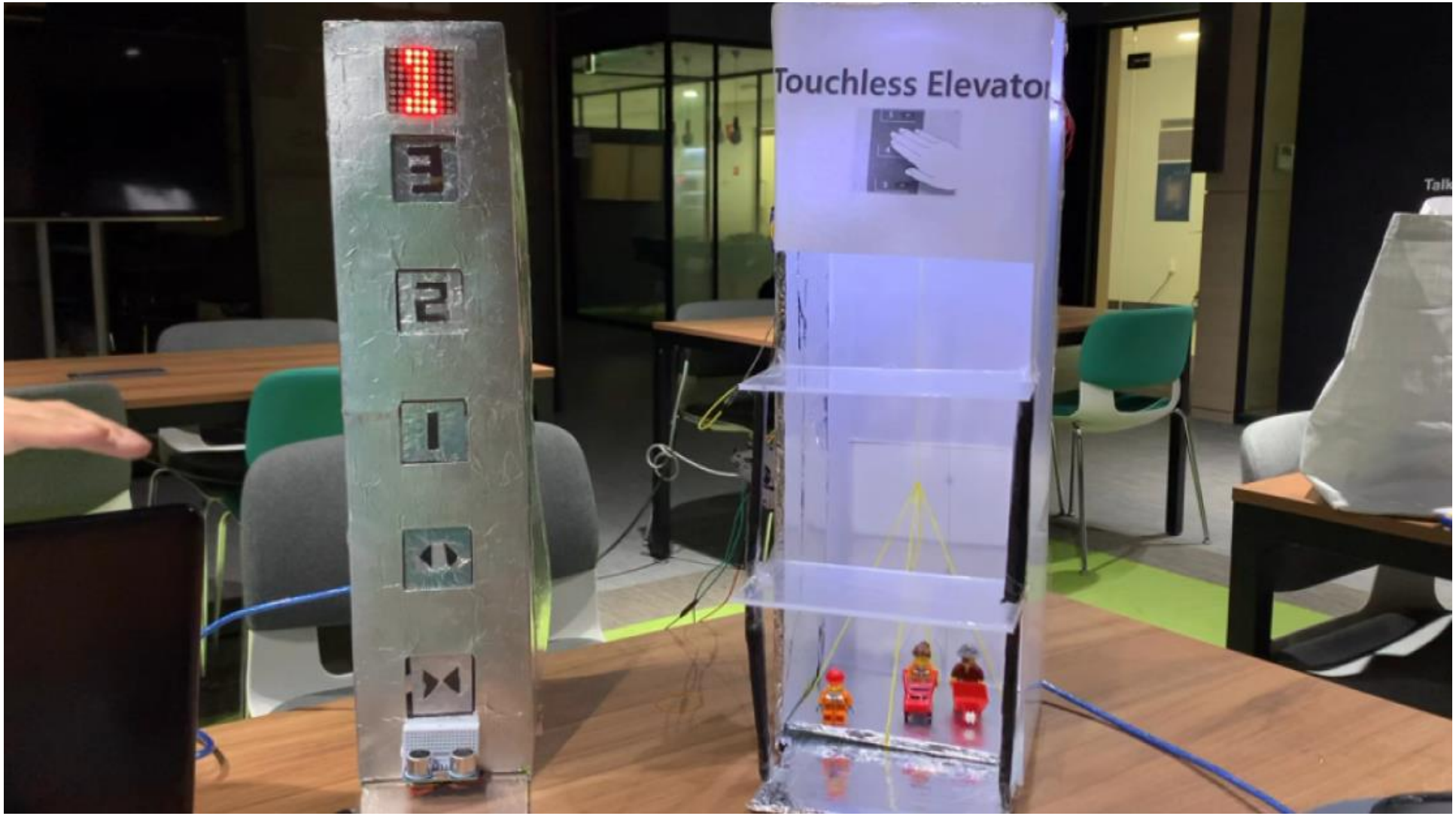
```
if (data=='x') //받아들인 data가 x(1층)면 1층 버튼이 입력됐다고 판단함
{
  tmp=1; //입력 층수 변수에 1대입
  Serial.println(data); //시리얼 모니터에 받아들인 문자 출력

  if (tmp < ele_floor) //입력한 층수보다 엘리베이터가 높은 층수에 있는 경우
  {
    steps_left = 4096*5 ; //일정 스텝 수만큼 스텝모터를 돌림
    Direction = false; //모터의 방향은 실이 풀리는 방향(즉, 내려가는 방향)
    while(steps_left>0)
    {
      currentMillis = micros();
      if(currentMillis-last_time>=1000)
       {
         stepper(1);
         time=time+micros()-last_time;
         last_time=micros();
         steps_left--;
       }
    } // while
    if (ele_floor - tmp == 2) //현재 엘리베이터 위치와 입력한 엘리베이터 층수간의 차이가 2면
    {
      steps_left = 4096*5; //앞서 돌린 스텝 수만큼 스텝모터를 돌림
      Direction = false; //모터의 방향은 실이 풀리는 방향(즉, 내려가는 방향)
      while(steps_left>0)
      {
        currentMillis = micros();
        if(currentMillis-last_time>=1000)
        {
          stepper(1);
          time=time+micros()-last time;
```

엘리베이터 1층 버튼 입력 정보 (x)를 받아서
스텝모터를 해당 층으로 이동하는 방향으로 회전
☞ 스텝모터가 회전함에 따라 실의 길이를 조절해
승강기가 해당 층으로 이동함

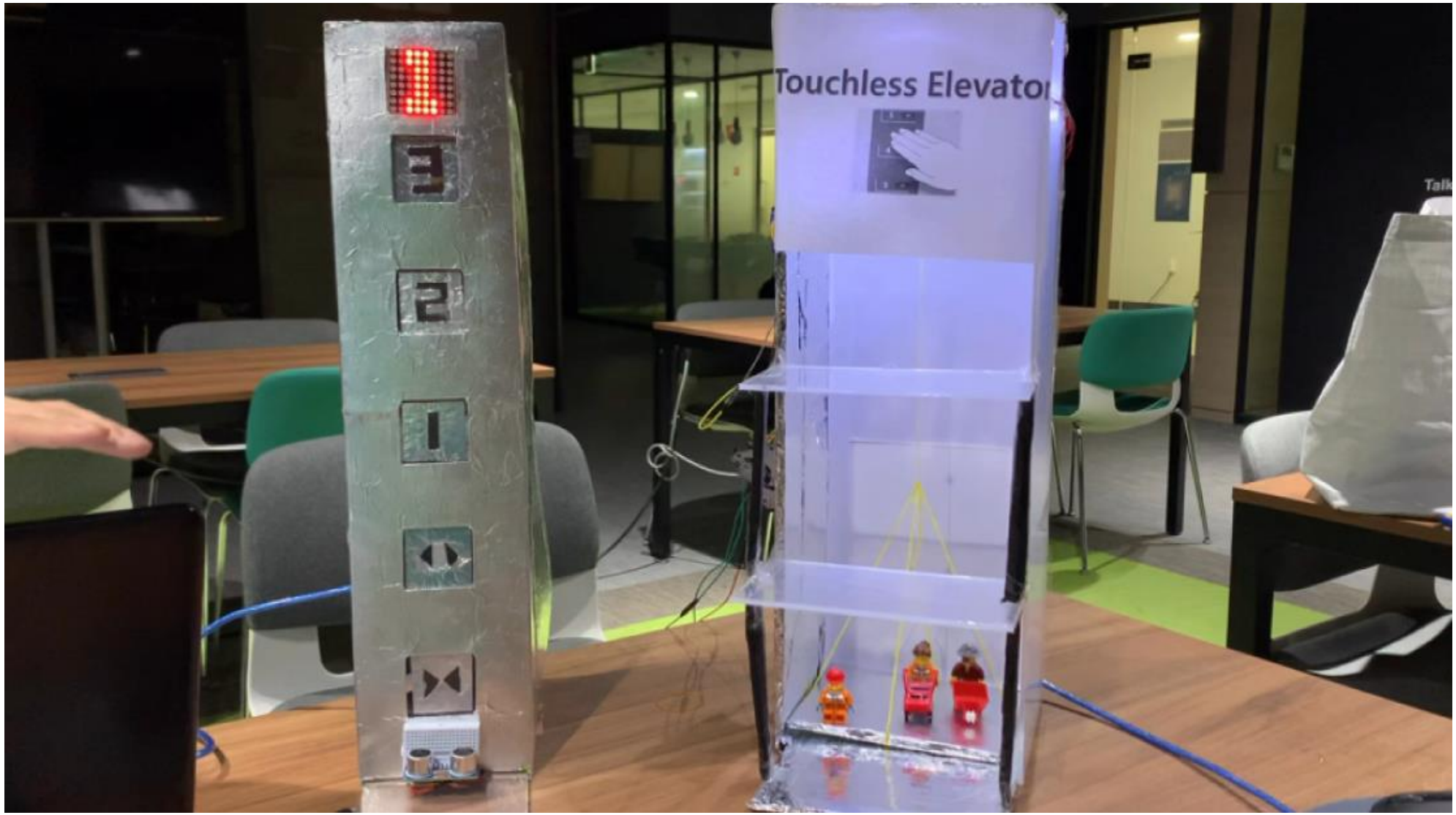
구현 영상

1→2층
(x 2배속)



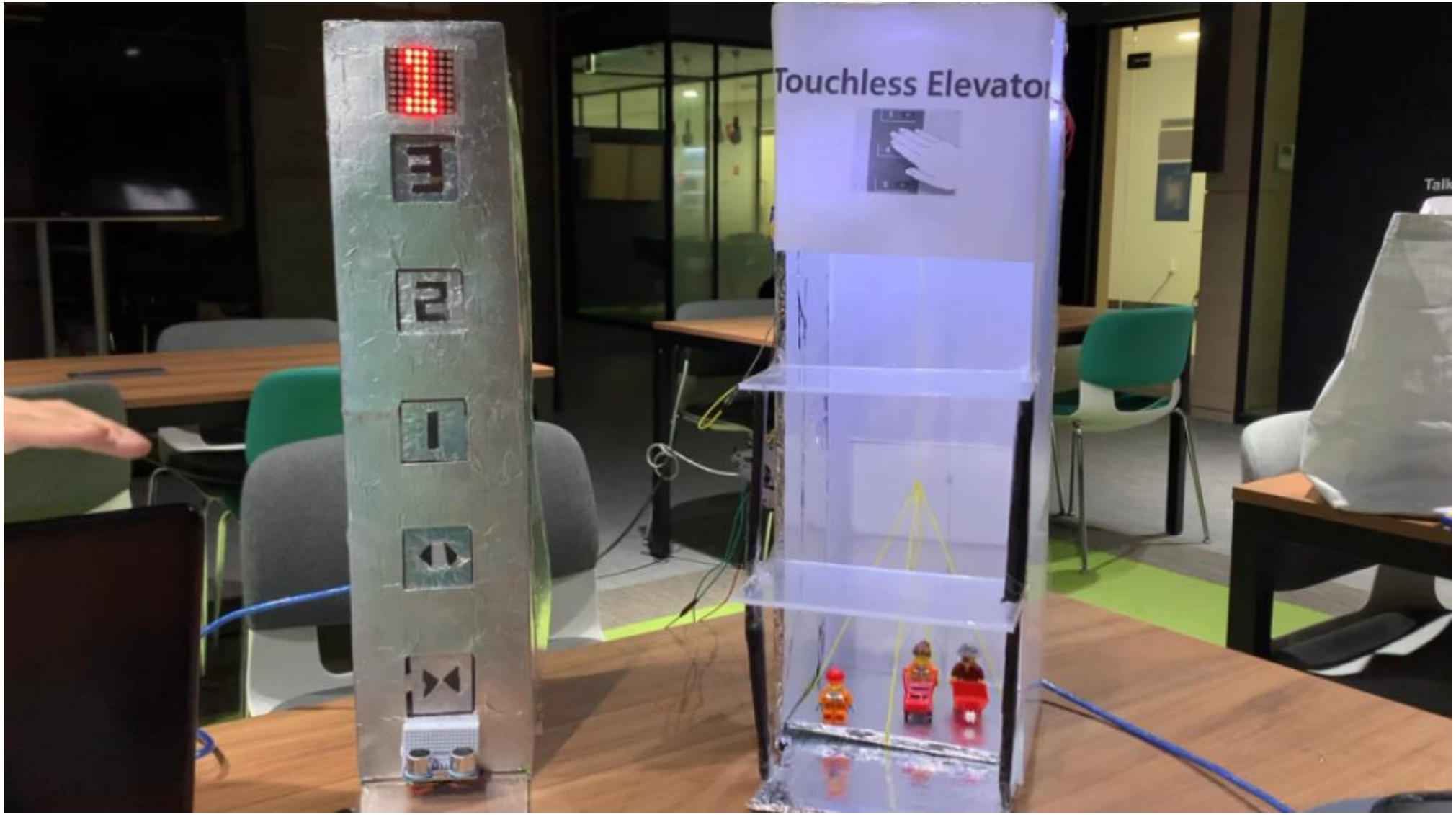
구현 영상

2→3층
(x 2배속)



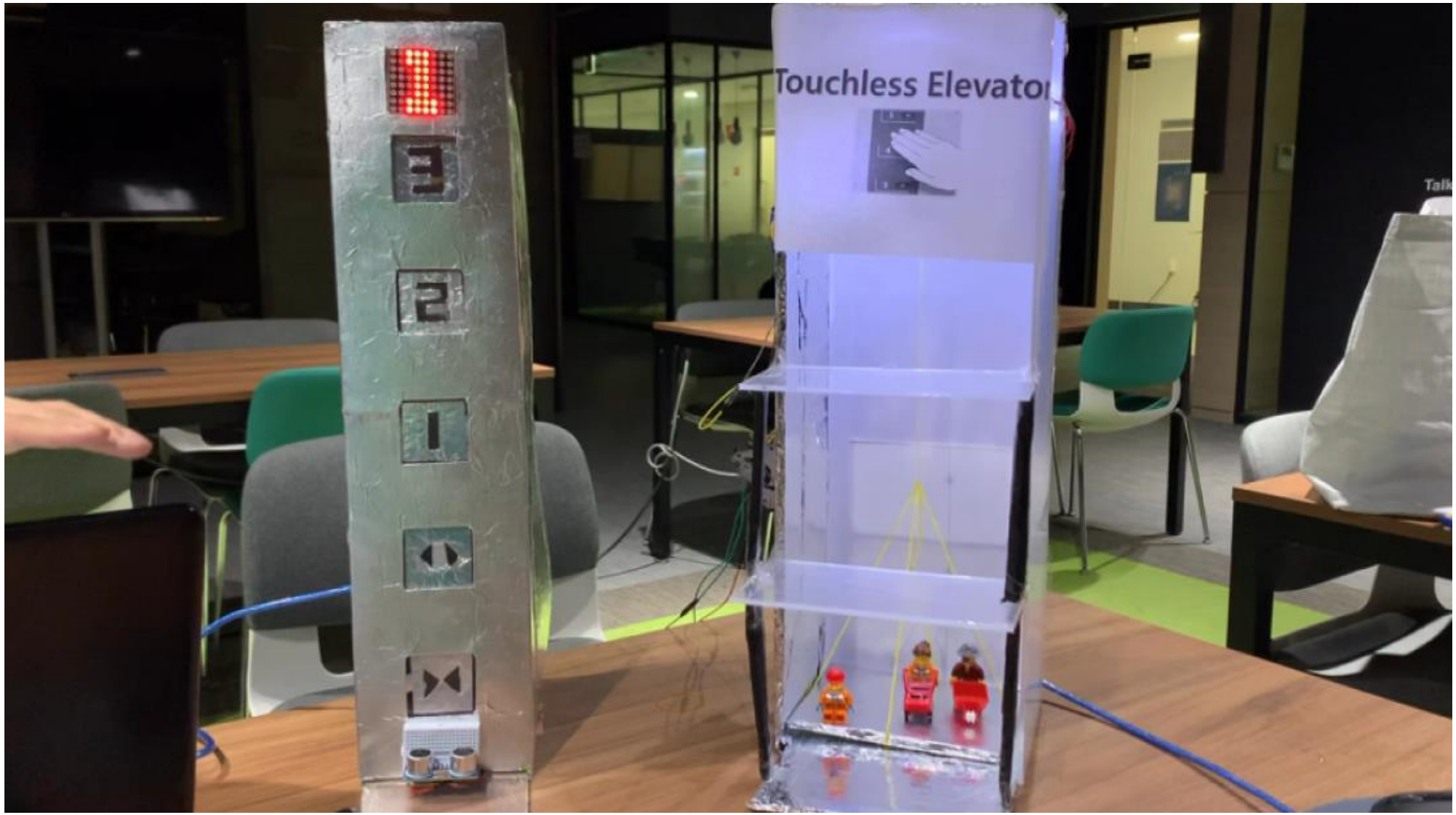
구현 영상

3→ 1→ 2층
(x 3배속)



구현 영상

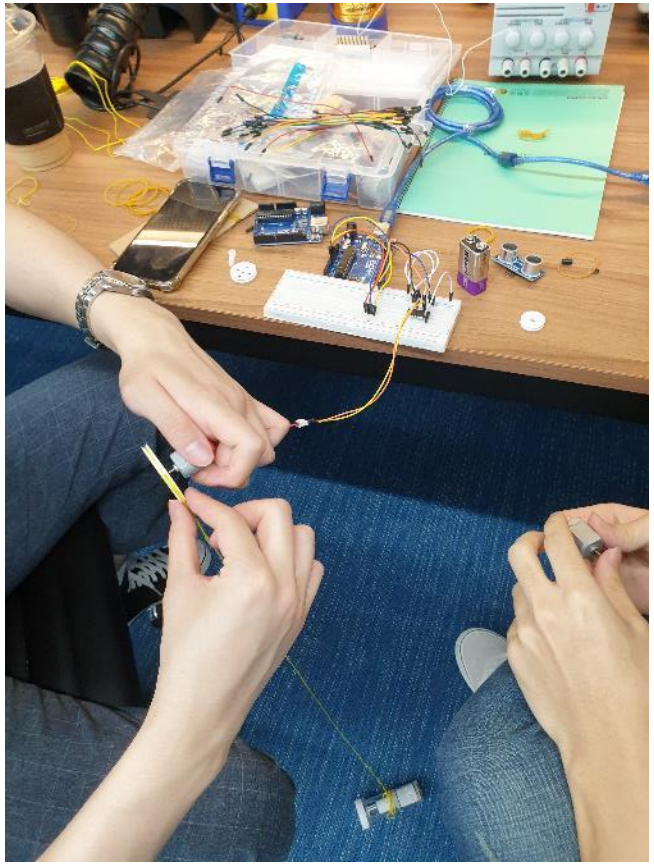
1→ 3→ 1층
(x 4배속)



03

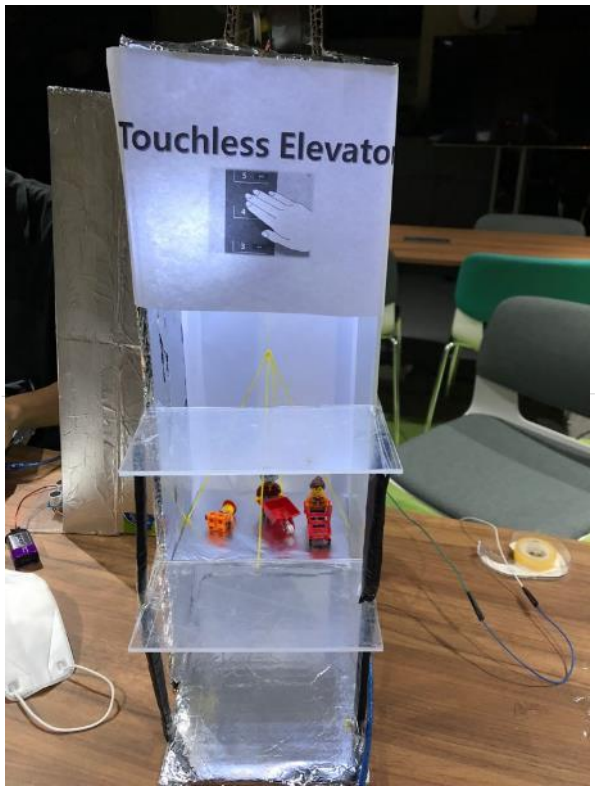
시행착오

Problem 1 :
DC모터로 승강기의 오르내림을 테스트 중,
DC모터가 예상보다 약해 **가벼운 물체조차 들어올리지 못하는 문제**가 발생

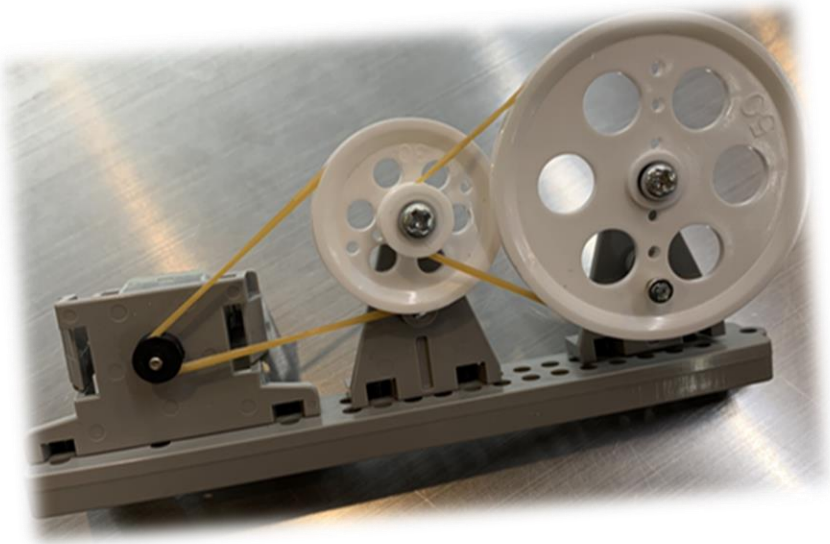


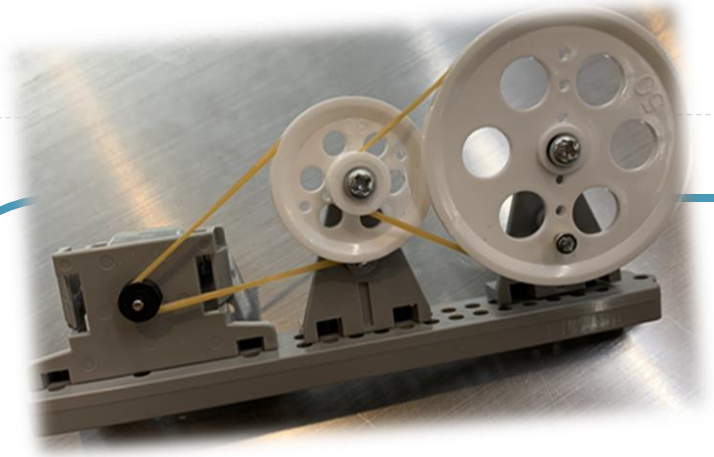
해결 :
여러 시도 끝에, DC모터보다
비교적 무거운 물체를 안정적으로 들어올릴 수 있는 **스텝모터**로 교체하기로 결정

변경 :
스텝모터가 엘리베이터의 문을 담당할 서보모터의 무게를 감당하기에 무리가 있다고 판단하여 **목표를 변경**
→ 엘리베이터 문 구현 계획을 철회



Problem 2 :
엘리베이터 문 미구현





스텝모터와 DC모터가 생각보다 많은 무게를 들어올리지 못함

-> 실제로 실험 결과 DC모터는 약 10cm*10cm*10cm 정도의 박스의 무게를 견디지 못함
스텝모터의 경우에도 이와 마찬가지

-> 승강기의 무게를 최소화하여 승강기의 바닥면만 구현
도르레를 추가 하였다면 승강기의 작동이 불가능 할 것이라 판단

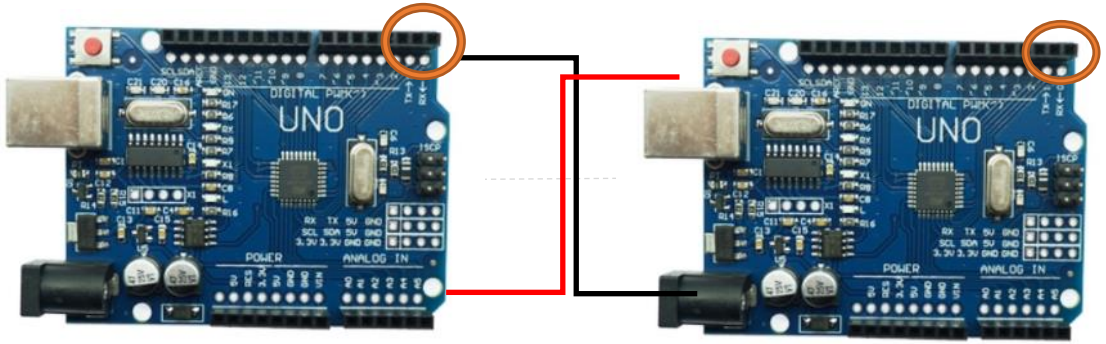
-> 문 구현 계획 철회

Problem 3 :
터치리스 엘리베이터 구현은
3개의 UNO보드를 연결한 형태

Dot Matrix 를 구성한 UNO보드에는
대부분의 pin이 사용되었고
남은 4개의 핀만을 이용하여
다른 UNO보드에서 **데이터를 전송받기** 위한
새로운 방법을 고안해 내야 했다.

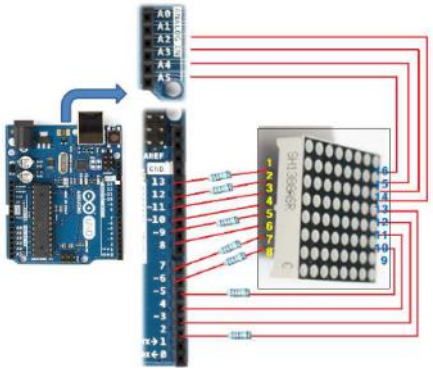
1번과 2번 아두이노가 이미 UART통신을
사용하므로, **UART 방법은 제외**해야 했다.

RX와 TX선 교차+ GND 연결



1번 아두이노
(Step Moter)

2번 아두이노
(Button Box)



3번 아두이노
(Dot Matrix)

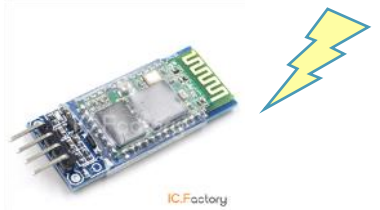
고안1 : 블루투스 모듈?

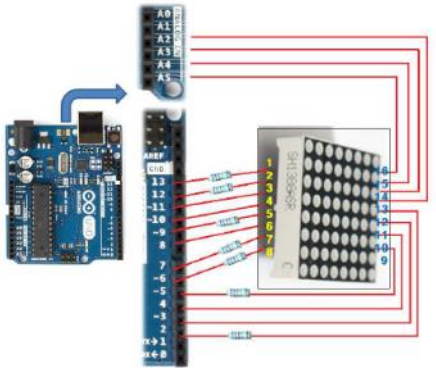
Dot Matrix를 구성한 UNO와
버튼박스 UNO간에 **블루투스 모듈을 이용**해
데이터를 전송받는 것을 고안

BUT

1. UART통신 중 다른 UNO 코드신호와의 충돌 가능성
2. Dot Matrix 와 블루투스 모듈을 함께 사용함으로써 발생하는 전력부족
3. 보유한 블루투스 모듈(HC-06)은 슬레이브 기능만 존재 – 연결불가

→ 위 3가지 기술적인 문제로 블루투스 철회



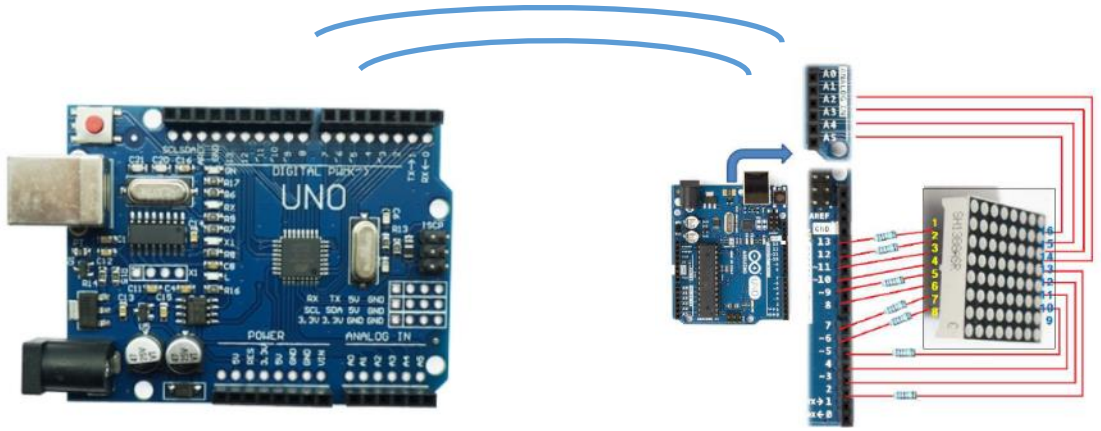


3번 UNO
(Dot Matrix)

2번 UNO
(Button Box)

고안2 : 2진법 신호 전송 !
2번 UNO와 3번 UNO간에 필요한 통신은 단순히 신호만 전송되면 충분한 **단방향 통신**이므로 M-M 선 두개를 연결해 **이진법으로 신호를 주고받는 것을 고안함**

2번 UNO에서
초음파 센서에 의해 목표 층이 입력되면,
↓
3번 UNO으로 2진법으로 신호가 전달되고,
↓
3번 UNO는 적절한 신호를 Dot Matrix에 출력



2번 UNO
(Button Box)

(0,0) : nothing(X)
1층 인식: (0,1) 전송
2층 인식: (1,0) 전송
3층 인식: (1,1) 전송

3번 UNO
(Dot Matrix)

(0,0) 입력: 출력하지 않음
(0,1) 입력됨 : '1' 출력
(1,0) 입력됨 : '2' 출력
(1,1) 입력됨 : '3' 출력

04

마무리

팀원의 역할

- 구범준: Touchless 엘리베이터의 승강기 제작 및 엘리베이터 작동 상태 점검
- 김건우: Touchless 엘리베이터의 버튼 박스 코드 제작과 전체적인 디자인 검토
- 박현우: Touchless 엘리베이터의 승강기 코드 제작과 전반적인 코드 점검
- 신도현: PPT / 영상 제작 및 Touchless 엘리베이터 검토 및 보완 담당

Thank you for watching!