



# *SAA*

# *final project*

2014044293 박시현  
2017012033 배수연  
2017012924 변성훈



③ 목적지까지 가이드

④ 다시 엘리베이터로 리턴

3F

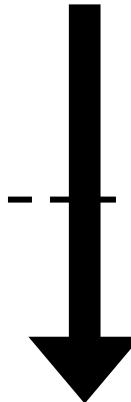
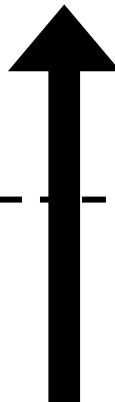
2F

1F

② 층수 계산

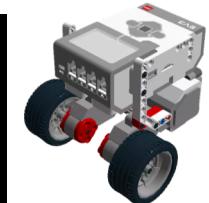
① 엘리베이터에 들어가서  
문을 바라봄

ELEVATOR



⑤ 층수 계산

⑥ 후진주차



# ① 엘리베이터에 들어가서 문을 바라봄

“

```
174     void  getElv()
175     {
176         resetMotorEncoder(lm);
177         int curval;
178         resetGyro(gs);
179         while(getMotorEncoder(lm) < 100) // go front
180         {
181             curval = getGyroDegrees(gs);
182             setMotorSpeed(lm, 30 - curval);
183             setMotorSpeed(rm, 30 + curval);
184         }
185
186         rightturn();
187         resetGyro(gs);
188         resetMotorEncoder(lm);
189
190         while(getMotorEncoder(lm) < 100) // go front
191         {
192             curval = getGyroDegrees(gs);
193             setMotorSpeed(lm, 30 - curval);
194             setMotorSpeed(rm, 30 + curval);
195         }
196
197         rightturn();
198         rightturn();|
```

## ② 층수 계산

“



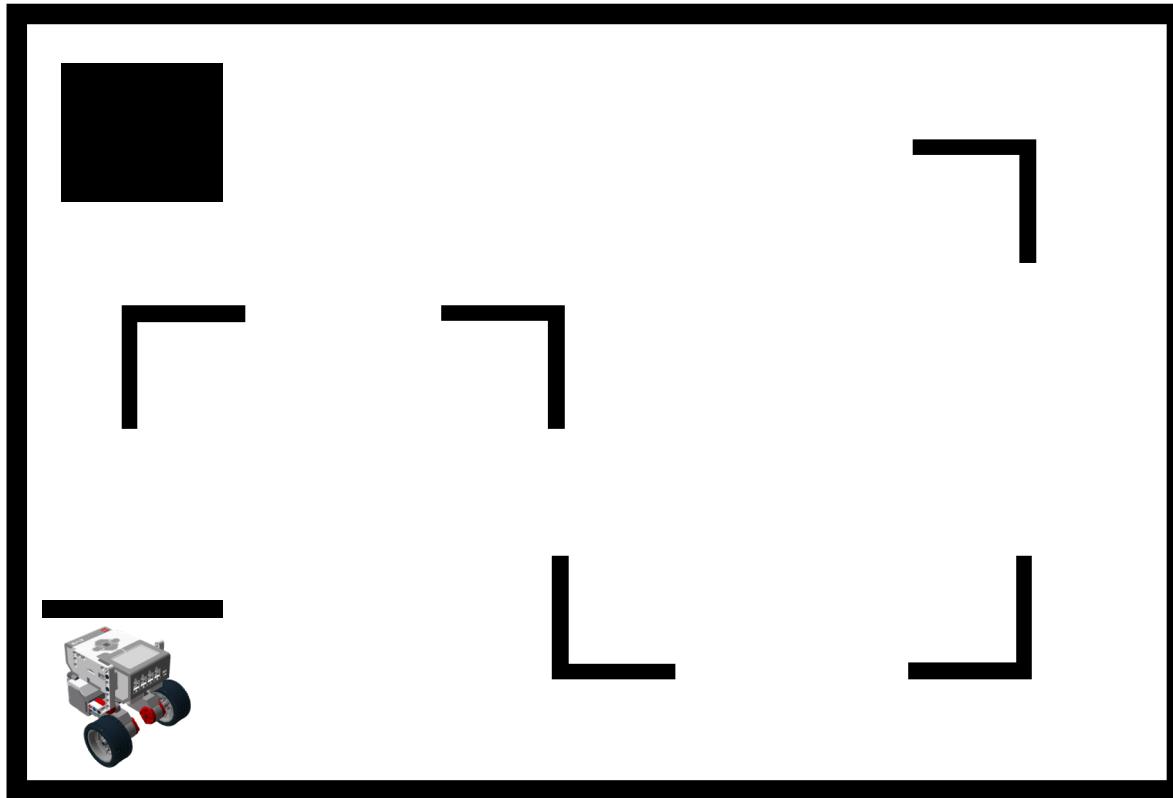
## ② 층수 계산

“

```
29 void moveFloor()
30 {
31     displayBigTextLine(1,"press button"); // erase point
32     while(getButtonPress(1) == 0) sleep(10);
33     eraseDisplay(); // erase point
34     startTask(doorState);
35     while(1){
36         if(openState == 0)
37         {
38             clearTimer(T1);
39             break;
40         }
41     }
42     int floorNow = 1; // start floor
43     int onetime = 1; // totaltime + operation once
44
45     while(1){
46         if(openState == 1 && onetime == 1)
47         {
48             if(179 < time100[T1] && time100[T1] < 188)
49                 floorNow++;
50             else if(200 < time100[T1] && time100[T1] < 210)
51                 floorNow = floorNow+2;
52             else if(230< time100[T1] && time100[T1] <250)
53                 floorNow = floorNow+3;
54             else if(260 < time100[T1] && time100[T1] < 275)
55                 floorNow = floorNow+4;
56
57             if ( floorNow == endfloor)
58             {
59                 displayBigTextLine(1,"arrive");
60                 displayBigTextLine(3,"%dF", floorNow);
61                 playSound(soundBeepBeep);
62                 break;
63             }
64
65         }
66     }
67
68     if ( floorNow == endfloor)
69     {
70         if ( floorNow == endfloor)
71         {
72             displayBigTextLine(1,"arrive");
73             displayBigTextLine(3,"%dF", floorNow);
74             playSound(soundBeepBeep);
75             break;
76         }
77     }
78
79     else if( endfloor < floorNow)
80     {
81         displayBigTextLine(1,"error");
82     }
83     else
84     {
85         displayBigTextLine(1,"wait");
86         displayBigTextLine(3,"%dF", floorNow);
87         while(1{
88             if(openState == 0)
89             {
90                 clearTimer(T1);
91                 onetime = 1;
92                 break;
93             }
94         }
95     }
96
97     stopTask(doorState);
98 }
```

- ③ 목적지까지 가이드
- ④ 다시 엘리베이터로 리턴

“



- ③ 목적지까지 가이드
- ④ 다시 엘리베이터로 리턴

“

Encoder값 우회전 Encoder값 좌회전 Encoder값 U턴

1000	2	800	1	1830	0
------	---	-----	---	------	---



Encoder값 좌회전 Encoder값 우회전 Encoder값

### ③ 목적지까지 가이드

### ④ 다시 엘리베이터로 리턴

```
12 void rightturn()
13 {
14     resetGyro(gs);
15     while(getGyroDegrees(gs) < 90)
16     {
17         setMotorSpeed(lm, 30);
18         setMotorSpeed(rm, -30);
19     }
20 }
21
22 void leftturn()
23 {
24     resetGyro(gs);
25     while(getGyroDegrees(gs) > -90)
26     {
27         setMotorSpeed(lm, -30);
28         setMotorSpeed(rm, 30);
29     }
30 }

32 void scanLine()
33 {
34     displayBigTextLine(1, "scan dark & push");
35     while(getButtonPress(1) == 0) sleep(10);
36     playTone(440, 20);
37     dark[0] = getColorReflected(cs1);
38     dark[1] = getColorReflected(cs2);
39     sleep(1000);

40     displayBigTextLine(1, "scan white & push");
41     resetMotorEncoder(lm);
42     setMotorSpeed(lm, 30);
43     setMotorSpeed(rm, 30);
44     sleep(1000);
45     playTone(440, 20);
46     white[0] = getColorReflected(cs1);
47     white[1] = getColorReflected(cs2);
48     sleep(1000);

49 //   for(int i = 0 ; i<2 ; i++) gray[i] = (dark[i]+white[i])/2;
50     bound = (dark[0]+dark[1]+white[0]+white[1])/4;
51
52 }
```

## ③ 목적지까지 가이드

## ④ 다시 엘리베이터로 리턴

```
54 task main()
55 {
56     scanLine();
57     int i = -1;
58     /*displayBigTextLine(1, "%d", dark[1]);
59     displayBigTextLine(3, "%d", white[1]);
60     displayBigTextLine(5, "%d", bound);*/
61     eraseDisplay();
62     resetGyro(gs);
63     int curval;
64     while(1){
65         sleep(1);
66         int left = getColorReflected(cs1);
67         int right = getColorReflected(cs2);
68
69         if(right < bound || left < bound)
70         {
71             setMotorSpeed(lm, 30);
72             setMotorSpeed(rm, 30);
73             sleep(100);
74
75             left = getColorReflected(cs1);
76             right = getColorReflected(cs2);
77
78             if(right < bound && left >= bound){
79                 i++;
80                 sleep(600); // go front little
81                 stack[i] = getMotorEncoder(lm); // modify point (change order)
82                 i++;
83                 stack[i] = 2; // right
84                 rightturn();
85                 resetGyro(gs);
86                 resetMotorEncoder(lm);
87             }
88         }
89     }
90 }
```

```
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124

    else if(left < bound && right >= bound) {
        i++;
        sleep(600); // go front little
        stack[i] = getMotorEncoder(lm); // modify point (change order)
        i++;
        stack[i] = 1; // left
        leftturn();
        resetGyro(gs);
        resetMotorEncoder(lm);
    }

    else if((left < bound) && (right < bound)){
        i++;
        stack[i] = getMotorEncoder(lm);
        setMotorSpeed(lm, 0);
        setMotorSpeed(rm, 0);
        sleep(3000);
        break;
    }
}
else
{
    curval = getGyroDegrees(gs);
    setMotorSpeed(lm, 30 - curval);
    setMotorSpeed(rm, 30 + curval);
}

for(int j = 0; j <= i; j++)
{
    //displayBigTextLine(1, "%d", i);
    displayBigTextLine((2*j)+1, "%d=%d", j, stack[j]);
}
//displayBigTextLine(1, "%d", i);
sleep(10000);
}
```

- ③ 목적지까지 가이드  
④ 다시 엘리베이터로 리턴



## ⑥ 후진주차

“

```
199 void outElv()
200 {
201     int curval;
202     resetMotorEncoder(lm);
203     resetGyro(gs);
204
205     while(getMotorEncoder(lm) < 100) // go front
206     {
207         curval = getGyroDegrees(gs);
208         setMotorSpeed(lm, 30 - curval);
209         setMotorSpeed(rm, 30 + curval);
210     }
211
212     rightturn();
213
214     int left = getColorReflected(cs1);
215     int right = getColorReflected(cs2);
216
217     while(left < bound && right < bound)
218     {
219         setMotorSpeed(lm, -10);
220         setMotorSpeed(rm, -10);
221         left = getColorReflected(cs1);
222         right = getColorReflected(cs2);
223     }
224
225     while(left <= gray || right <= gray)
226     {
227         if(left <= gray)
228         {
229             setMotorSpeed(lm, -10);
230             setMotorSpeed(rm, 0);
231         }
232         else
233         {
234             setMotorSpeed(lm, -10);
235             setMotorSpeed(rm, 0);
236         }
237
238         left = getColorReflected(cs1);
239         right = getColorReflected(cs2);
240     }
241
242     setMotorSpeed(lm, 0);
243     setMotorSpeed(rm, 0);
244     sleep(1000);
245 }
```

# 전체 코드

```
1 #pragma config(Motor, motorB, lm, PIDControl, encoder)
2 #pragma config(Motor, motorC, rm, PIDControl, encoder)
3 #pragma config(Sensor, S1, cs1, sensorEV3_Color)
4 #pragma config(Sensor, S2, cs2, sensorEV3_Color)
5 #pragma config(Sensor, S3, gs, sensorEV3_Gyro)
6 #pragma config(Sensor, S4, us, sensorEV3_Ultrasonic)
7
8 #define doorDistance 30
9
10 int endfloor = 5; // chage part
11 int stack[6] = {1000, 2, 800, 1, 1830, 0}; // change part
12 int bound = 50; // chage part
13 int openState = 0;
14 int i = 0;
15
16 task doorState()
17 {
18     while(1)
19     {
20         i = getUSDistance(us);
21         displayBigTextLine(7,"%d",i);
22         if (doorDistance < i)
23             openState = 1;
24         else
25             openState = 0;
26     }
27 }
```

```
29 void moveFloor()
30 {
31     displayBigTextLine(1,"press button"); // erase point
32     while(getButtonPress(1) == 0) sleep(10);
33     eraseDisplay(); // erase point
34     startTask(doorState);
35     while(1){
36         if(openState == 0)
37         {
38             clearTimer(T1);
39             break;
40         }
41     }
42     int floorNow = 1; // start floor
43     int onetime = 1; // totaltime + operation once
44
45     while(1){
46         if(openState == 1 && onetime == 1)
47         {
48             if(179 < time100[T1] && time100[T1] < 188)
49                 floorNow++;
50             else if(200 < time100[T1] && time100[T1] < 210)
51                 floorNow = floorNow+2;
52             else if(230< time100[T1] && time100[T1] <250)
53                 floorNow = floorNow+3;
54             else if(260 < time100[T1] && time100[T1] < 275)
55                 floorNow = floorNow+4;
56
57             if ( floorNow == endfloor)
58             {
59                 displayBigTextLine(1,"arrive");
60                 displayBigTextLine(3,"%dF", floorNow);
61                 playSound(soundBeepBeep);
62                 break;
63             }
64             else if( endfloor < floorNow)
65             {
66                 displayBigTextLine(1,"error");
67             }
68             else
69             {
70                 displayBigTextLine(1,"wait");
71                 displayBigTextLine(3,"%dF", floorNow);
72                 while(1{
73                     if(openState == 0)
74                     {
75                         clearTimer(T1);
76                         onetime = 1;
77                         break;
78                     }
79                 }
80             }
81         }
82     }
83 }
84 stopTask(doorState);
85 }
```

“

```
89 void rightturn()
90 {
91     resetGyro(gs);
92     while(getGyroDegrees(gs) < 90)
93     {
94         setMotorSpeed(lm, 30);
95         setMotorSpeed(rm, -30);
96     }
97 }
98 void leftturn()
99 {
100    resetGyro(gs);
101    while(getGyroDegrees(gs) > -90)
102    {
103        setMotorSpeed(lm, -30);
104        setMotorSpeed(rm, 30);
105    }
106 }
107 }
```

```
110 void guide()
111 {
112     int curval;
113     int i = -1;
114     resetGyro(gs);
115     while(1)
116     {
117         i++;
118         resetMotorEncoder(lm);
119         if(stack[i] == 0)
120             break;
121         else if(stack[i] == 2)
122         {
123             rightturn();
124             resetGyro(gs);
125         }
126         else if(stack[i] == 1)
127         {
128             leftturn();
129             resetGyro(gs);
130         }
131         else
132         {
```

```
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170 }
```

“

```
174 void getElv()
175 {
176     resetMotorEncoder(lm);
177     int curval;
178     resetGyro(gs);
179     while(getMotorEncoder(lm) < 100) // go front
180     {
181         curval = getGyroDegrees(gs);
182         setMotorSpeed(lm, 30 - curval);
183         setMotorSpeed(rm, 30 + curval);
184     }
185
186     rightturn();
187     resetGyro(gs);
188     resetMotorEncoder(lm);
189
190     while(getMotorEncoder(lm) < 100) // go front
191     {
192         curval = getGyroDegrees(gs);
193         setMotorSpeed(lm, 30 - curval);
194         setMotorSpeed(rm, 30 + curval);
195     }
196
197     rightturn();
198     rightturn();
199 }
```

```
201 void outElv()
202 {
203     int curval;
204     resetMotorEncoder(lm);
205     resetGyro(gs);
206
207     while(getMotorEncoder(lm) < 100) // go front
208     {
209         curval = getGyroDegrees(gs);
210         setMotorSpeed(lm, 30 - curval);
211         setMotorSpeed(rm, 30 + curval);
212     }
213
214     rightturn();
215
216     int left = getColorReflected(cs1);
217     int right = getColorReflected(cs2);
218
219     while(left < bound && right < bound)
220     {
221         setMotorSpeed(lm, -10);
222         setMotorSpeed(rm, -10);
223         left = getColorReflected(cs1);
224         right = getColorReflected(cs2);
225     }
226
227     while(left <= bound || right <= bound)
228     {
229         if(left <= bound)
230         {
231             setMotorSpeed(lm, -10);
232             setMotorSpeed(rm, 0);
233         }
234         else
235         {
236             setMotorSpeed(lm, -10);
237             setMotorSpeed(rm, 0);
238         }
239
240         left = getColorReflected(cs1);
241         right = getColorReflected(cs2);
242
243         setMotorSpeed(lm, 0);
244         setMotorSpeed(rm, 0);
245
246         sleep(1000);
247     }
```

“

```
250  task main()
251  {
252      getElv();
253      moveFloor();
254      guide();
255      moveFloor();
256      outElv();
257 }
```



*Thanks!*

Any questions?