

# Smart Door Lock



201802523 황주완  
201902126 한성주

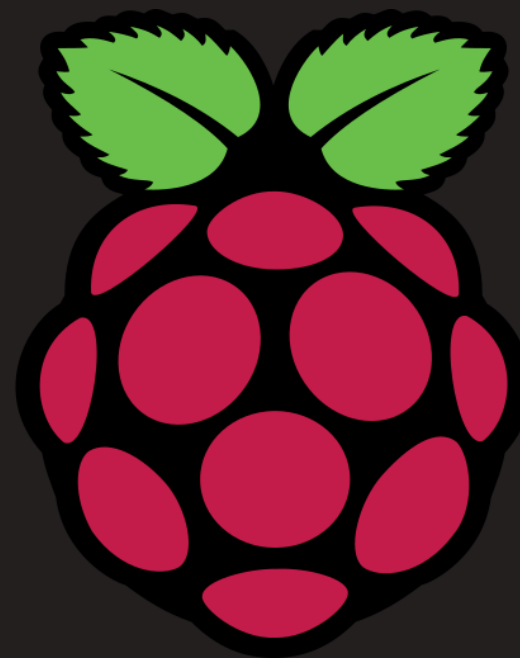
# 목차

1. 프로젝트 설명

2. 회로 구성

3. 코드 구현

4. 시연 영상





## 프로젝트 설명

### -> Smart Door Lock

#### 1) 부품

: I2C LCD, 피에로 부저, 서보 모터, 4\*4 키패드, 라즈베리파이

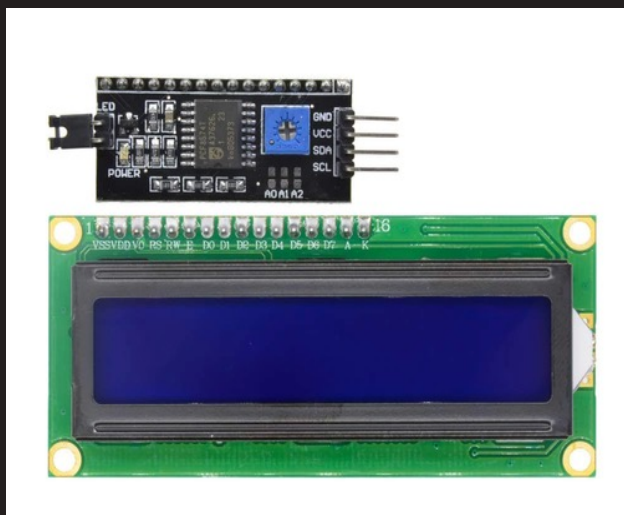
#### 2) 기능 소개

- 키패드를 통한 비밀번호 입력
- 비밀번호 입력 시 LCD에 표시(입력 창, 맞음/틀림)
- 비밀번호가 맞았을 때 정답 알림음이 울리고 모터가 180도 회전
- 비밀번호가 틀렸을 때 경보음이 울림

## ✓ 회로 구성

### 1) 부품

- I2C LCD



- 피에로 부저



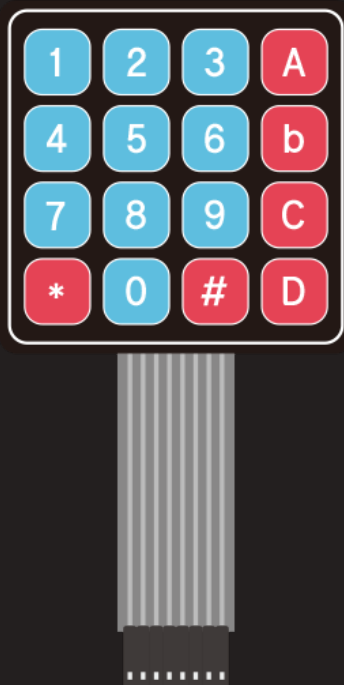
- 서보 모터



## ✓ 회로 구성

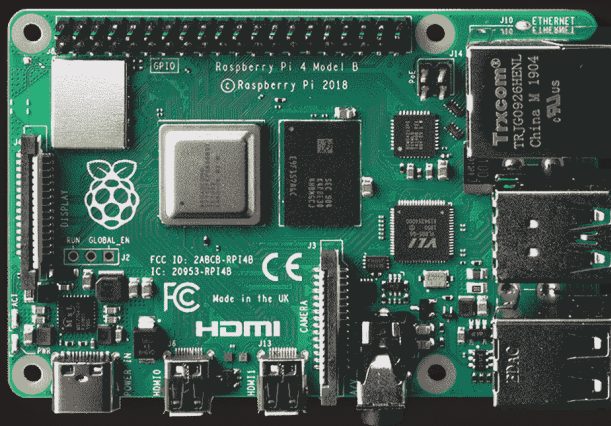
### 1) 부품

- 4\*4 키패드

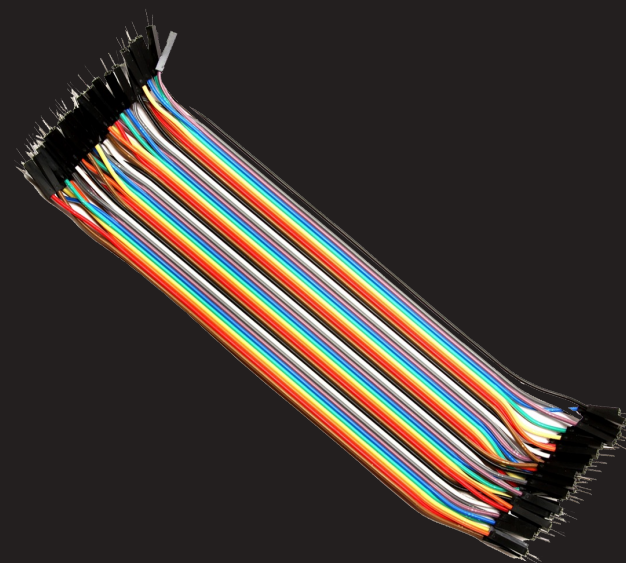


ATO Planet

- 라즈베리 파이 4



- 점퍼 케이블





## 회로 구성

### - I2C LCD

LCD	GPIO
GND	GND
VCC	5V 전원
SDA	GPIO 2 (SDA)
SCL	GPIO 3 (SCL)

### - 피에로 부저

Buzzer	GPIO
Buzzer(+)	GPIO 17
Buzzer(-)	GND

### - 서보 모터

Servo	GPIO
Signal	GPIO 18
VCC	5V 전원
GND	GND

### - 4\*4 키패드

Key Pad	GPIO
행 핀 (C1)	GPIO 5
행 핀 (C2)	GPIO 6
행 핀 (C3)	GPIO 13
행 핀 (C4)	GPIO 19
열 핀 (R1)	GPIO 12
열 핀 (R2)	GPIO 16
열 핀 (R3)	GPIO 20
열 핀 (R4)	GPIO 21



## 코드 구현



```
1 # 라이브러리 파일
2 import I2C_LCD_driver
3 import RPi.GPIO as GPIO
4 from time import sleep
5
6 # 키패드 행 핀 설정
7 C1 = 5
8 C2 = 6
9 C3 = 13
10 C4 = 19
11
12 # 키패드 열 핀 설정
13 R1 = 12
14 R2 = 16
15 R3 = 20
16 R4 = 21
17
18 # 부저, 서보모터 핀
19 buzzer = 17
20 servoPin = 18
```



## 코드 구현

```
22 relayState = True
23
24 # lcd객체 생성
25 lcd = I2C_LCD_driver.lcd()
26
27 # 시작 텍스트
28 lcd.lcd_display_string("System loading",1,1)
29 for a in range (0,16):
30     lcd.lcd_display_string(".",2,a)
31     sleep(0.1)
32 lcd.lcd_clear()
33
34 # 키가 눌리지 않았을 때는 -1
35 keypadPressed = -1
36
37 # 비밀번호
38 secretCode = "1111"
39 input = ""
40 GPIO.setwarnings(False)
41
42 SERVO_MAX_DUTY=12
43 SERVO_MIN_DUTY=3
```





## 코드 구현



```
45 GPIO.setmode(GPIO.BCM)
46 GPIO.setup(servoPin, GPIO.OUT)
47 GPIO.setup(buzzer,GPIO.OUT)
48
49 # 행 핀은 output
50 GPIO.setup(C1, GPIO.OUT)
51 GPIO.setup(C2, GPIO.OUT)
52 GPIO.setup(C3, GPIO.OUT)
53 GPIO.setup(C4, GPIO.OUT)
54
55 # 열 핀은 input
56 GPIO.setup(R1, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
57 GPIO.setup(R2, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
58 GPIO.setup(R3, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
59 GPIO.setup(R4, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
60
61 #서브 모터 핀 설정
62 servo=GPIO.PWM(servoPin, 50)
63 servo.start(0)
```



## 코드 구현

```
65 def keypadCallback(channel):
66     global keypadPressed
67     # 키패드가 눌러있지 않은 상태에서만 keypadPressed 값을 변경
68     if keypadPressed == -1:
69         keypadPressed = channel
70
71 # 상승 엣지 감지
72 GPIO.add_event_detect(R1, GPIO.RISING, callback=keypadCallback)
73 GPIO.add_event_detect(R2, GPIO.RISING, callback=keypadCallback)
74 GPIO.add_event_detect(R3, GPIO.RISING, callback=keypadCallback)
75 GPIO.add_event_detect(R4, GPIO.RISING, callback=keypadCallback)
76
77 # 열에 구체적 상태 부여
78 def setAllRows(state):
79     GPIO.output(C1, state)
80     GPIO.output(C2, state)
81     GPIO.output(C3, state)
82     GPIO.output(C4, state)
```



## 코드 구현



```
84 # PIN 확인 또는 초기화
85 def commands():
86     global relayState
87     global input
88     pressed = False
89     GPIO.output(C1, GPIO.HIGH)
90
91     # 비밀번호 입력 초기화
92     if (GPIO.input(R1) == 1):
93         print("Input reset!");
94         lcd lcd_clear()
95         lcd lcd_display_string("Clear",1,5)
96         sleep(1)
97         pressed = True
98     GPIO.output(C1, GPIO.HIGH)
```



## 코드 구현

```
100     # 비밀번호 check
101     if (not pressed and GPIO.input(R2) == 1):
102         # 비밀번호가 맞는 경우 lcd에 표시하고 버저가 한번 울림
103         if input == secretCode:
104             print("Code correct!")
105             lcd lcd_clear()
106             lcd lcd_display_string("Successful",1,3)
107             servo.ChangeDutyCycle(3.0)
108             sleep(2)
109             servo.ChangeDutyCycle(12.5)
110             sleep(2)
111             servo.ChangeDutyCycle(3.0)
112             sleep(2)
113
114
115         if relayState:
116             GPIO.output(buzzer,GPIO.HIGH)
117             sleep(0.3)
118             GPIO.output(buzzer,GPIO.LOW)
119             sleep(1)
120             relayState = False
121
122         elif relayState == False:
123             GPIO.output(buzzer,GPIO.HIGH)
124             sleep(0.3)
125             GPIO.output(buzzer,GPIO.LOW)
126             sleep(1)
127             relayState = True
```



## 코드 구현

```
129         # 비밀번호가 틀렸을 경우 lcd에 표시하고 버저가 3번 울림
130     else:
131         print("Incorrect code!")
132         lcd lcd_clear()
133         lcd lcd_display_string("Wrong PIN!",1,3)
134         GPIO.output(buzzer,GPIO.HIGH)
135         sleep(0.3)
136         GPIO.output(buzzer,GPIO.LOW)
137         sleep(0.3)
138         GPIO.output(buzzer,GPIO.HIGH)
139         sleep(0.3)
140         GPIO.output(buzzer,GPIO.LOW)
141         sleep(0.3)
142         GPIO.output(buzzer,GPIO.HIGH)
143         sleep(0.3)
144         GPIO.output(buzzer,GPIO.LOW)
145     pressed = True
146     GPIO.output(C1, GPIO.LOW)
147
148     if pressed:
149         input = ""
150     return pressed
```



## 코드 구현

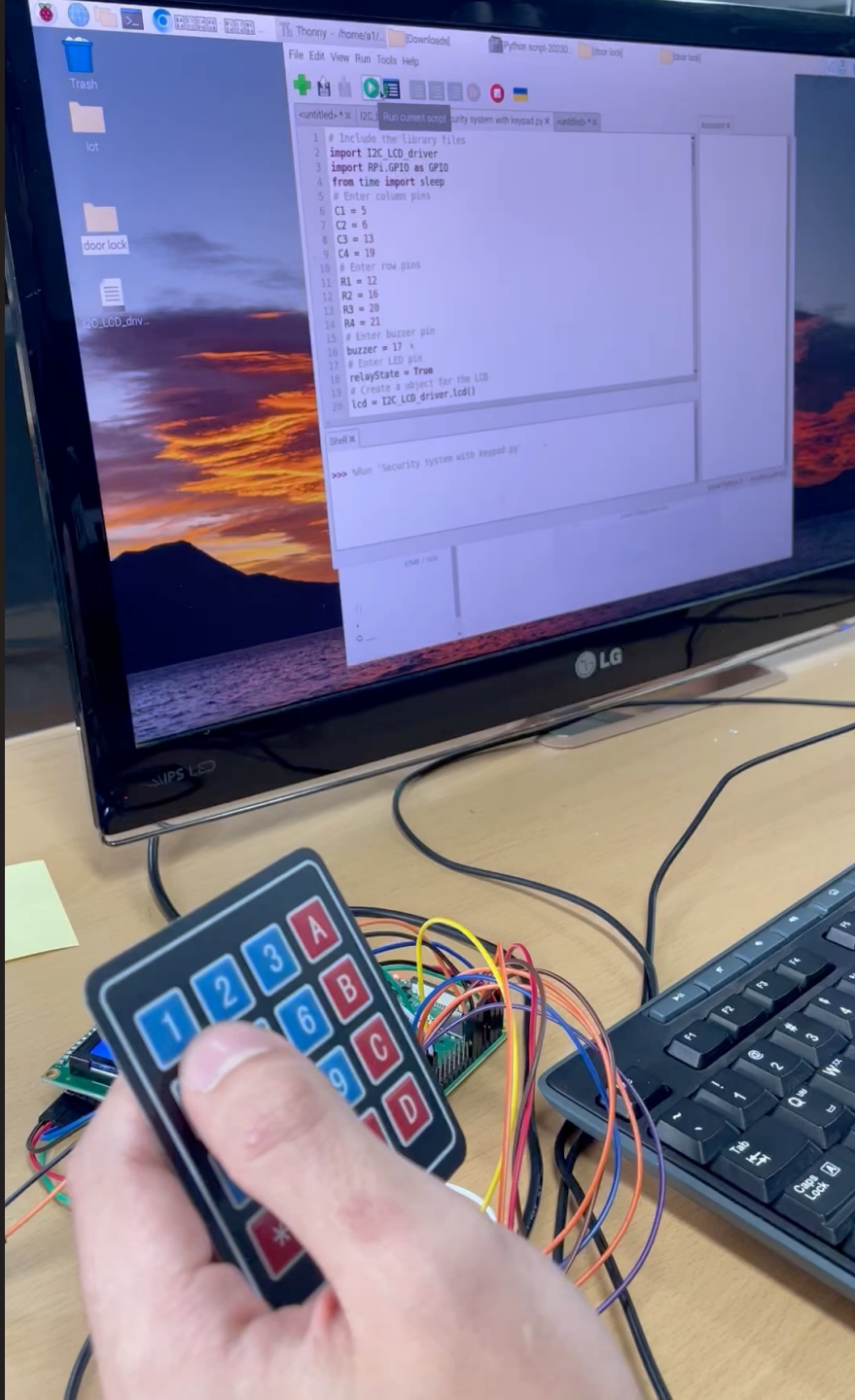


```
152 # 열을 읽고 값을 추가
153 def read(column, characters):
154     global input
155     GPIO.output(column, GPIO.HIGH)
156     if(GPIO.input(R1) == 1):
157         input = input + characters[0]
158         print(input)
159         lcd lcd_display_string(str(input),2,0)
160     if(GPIO.input(R2) == 1):
161         input = input + characters[1]
162         print(input)
163         lcd lcd_display_string(str(input),2,0)
164     if(GPIO.input(R3) == 1):
165         input = input + characters[2]
166         print(input)
167         lcd lcd_display_string(str(input),2,0)
168     if(GPIO.input(R4) == 1):
169         input = input + characters[3]
170         print(input)
171         lcd lcd_display_string(str(input),2,0)
172     GPIO.output(column, GPIO.LOW)
```



## 코드 구현

```
174 try:
175     while True:
176         lcd lcd_display_string("Enter your PIN:",1,0)
177
178         # 이전에 버튼이 눌렀다면 사용자가
179         # 버튼에서 손을 뗐는지 확인
180         if keypadPressed != -1:
181             setAllRows(GPIO.HIGH)
182             if GPIO.input(keypadPressed) == 0:
183                 keypadPressed = -1
184             else:
185                 sleep(0.1)
186         # 그렇지 않으면 input을 읽음
187         else:
188             if not commands():
189                 read(C1, ["D","C","B","A"])
190                 read(C2, ["#","9","6","3"])
191                 read(C3, ["0","8","5","2"])
192                 read(C4, ["*","7","4","1"])
193                 sleep(0.1)
194             else:
195                 sleep(0.1)
196
197 except KeyboardInterrupt:
198     print("Stopped!")
199
200 servo.stop()
201 GPIO.cleanup()
```



## ✓ 시연 영상

- 키패드의 숫자가 제대로 입력되고 있는지 shell 창을 통해 확인할 수 있다.



## ✓ 시연 영상

### Check Point !

- 키패드의 입력이 제대로 되고 있는가
- 키패드의 입력한 값이 LCD에 표시되는가
- 정답이 틀렸을 때 LCD에 틀렸다는 문구가 나오고 피어로 부저에서 경고음이 울리는가
- 키패드 입력 중 D버튼을 누르면 입력한 숫자가 초기화 되는가
- 정답이 맞았을 때 LCD에 정답이라는 문구가 나오고 피어로 부저와 서보 모터가 제대로 작동하는가



