

# 감시 시스템

12조

20161511 유종인

20161523 하태성

---

# Contents

1    **사용한 센서**

2    **감시 시스템 작동 순서**

4    **부가적인 기능**

3    **회로도 및 회로 구현**

4    **아두이노 코드**

5    **결과[시연영상]**

# 사용한 센서

---

LCD  
[wk-02]

LED  
[wk-03]

KEY PAD  
[wk-09]

조도센서  
[wk-10]

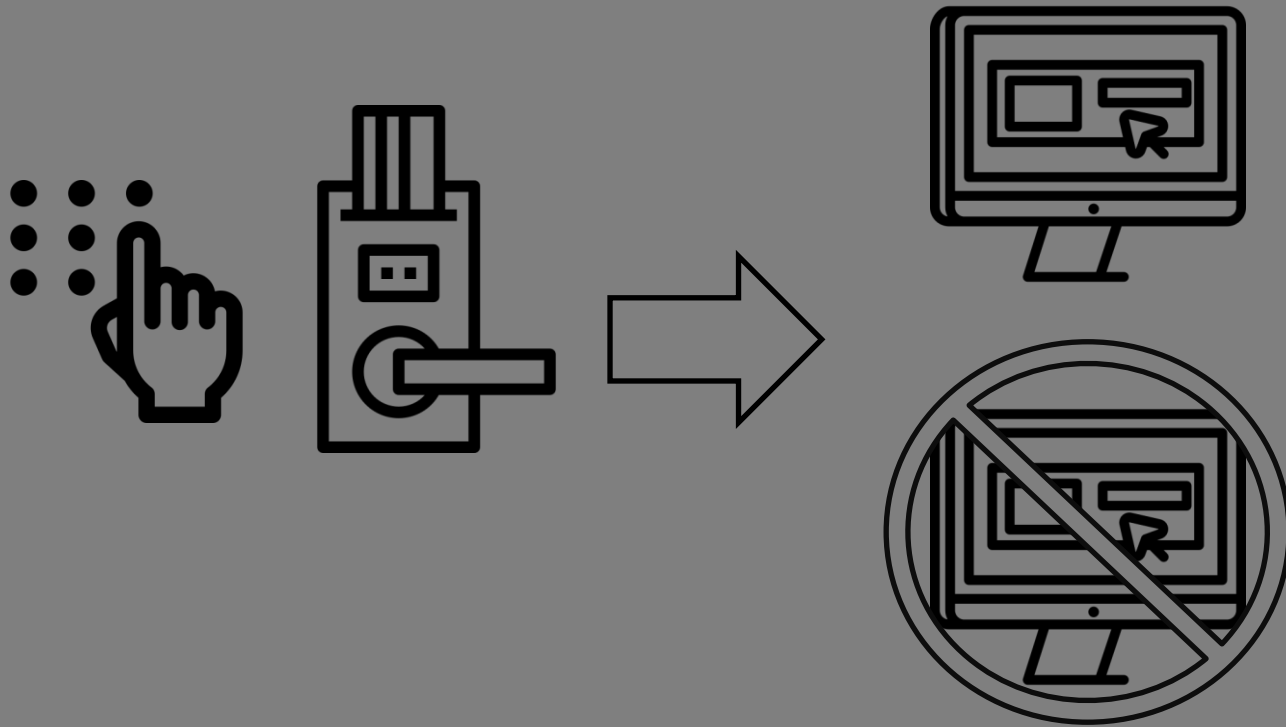
MFRC522  
[wk-13]

초음파  
센서  
[wk-13]

아두이노  
메가

# 감시 시스템 작동 순서

Step 1



비밀번호 및 카드 키 입력을 통해 모니터링의 시작/종료를 결정한다.

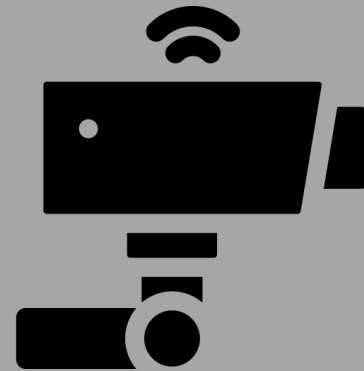
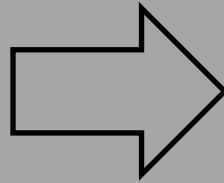
# 감시 시스템 작동 순서

---

Step 2



Monitoring Start!



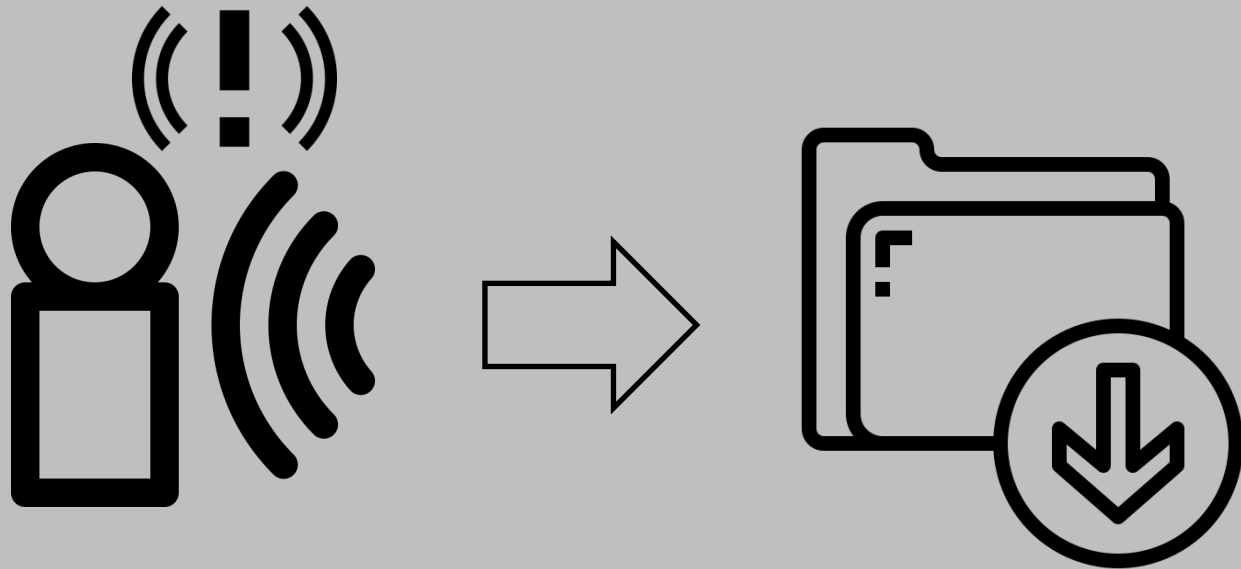
Sensor Start!

모니터링이 시작되면 초음파 센서와 조도센서가 작동을 시작한다.

# 감시 시스템 작동 순서

---

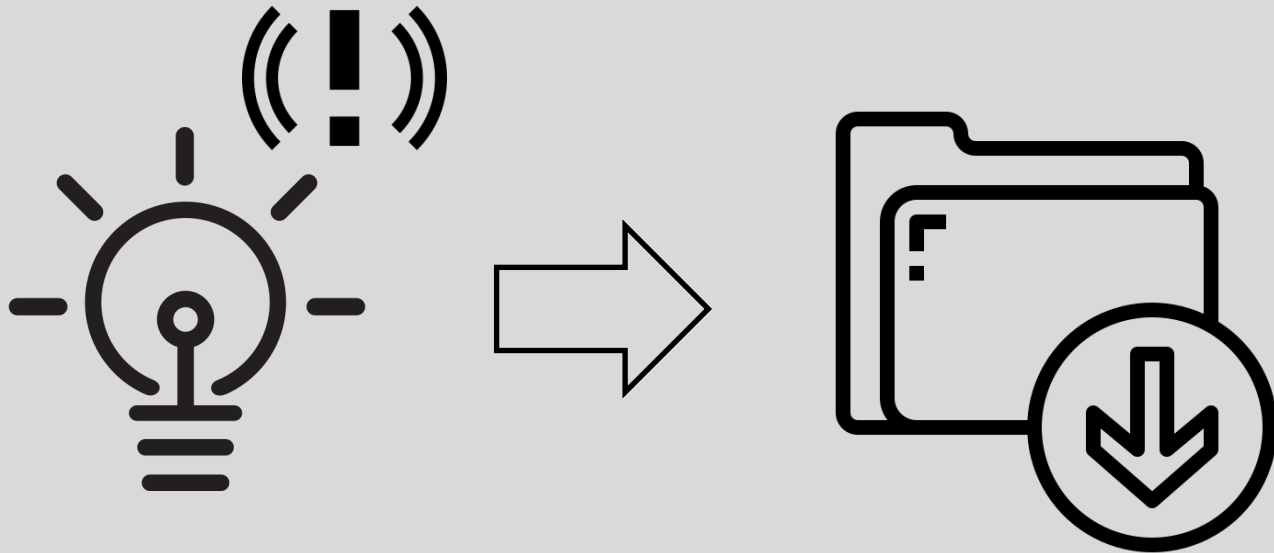
Step 3



초음파센서에 누군가 감지되었을 때 감지된 횟수를 저장한다.

# 감시 시스템 작동 순서

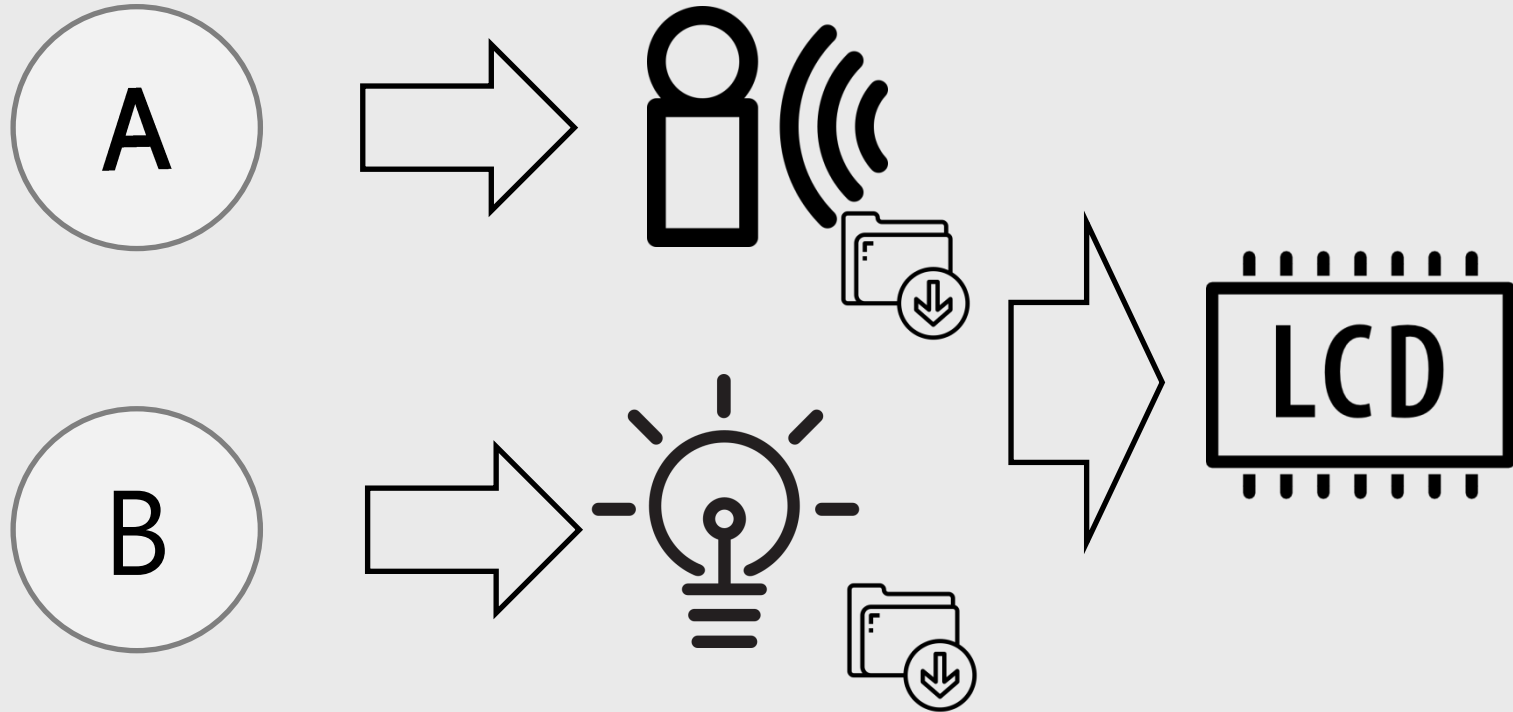
Step 4



조도 센서에 일정 이상의 밝기가 감지되면 감지된 횟수를 저장한다.

# 감시 시스템 작동 순서

Step 5

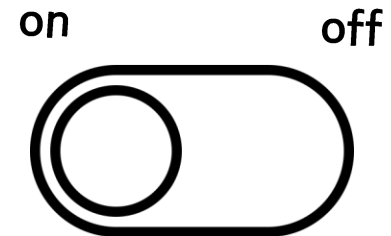
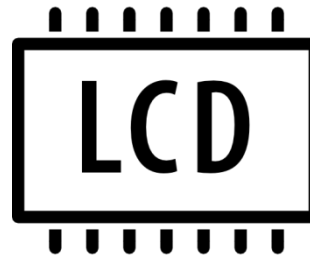
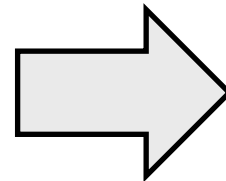
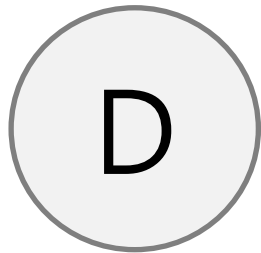
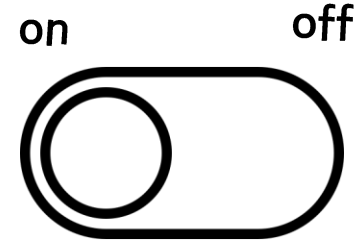
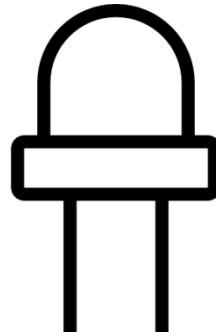
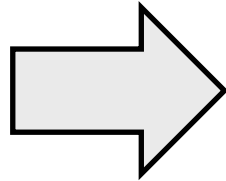
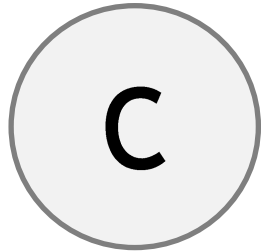


A버튼을 누르면 초음파센서가 감지한 횟수,  
B버튼을 누르면 조도센서가 감지한 횟수를 LCD에 출력한다.



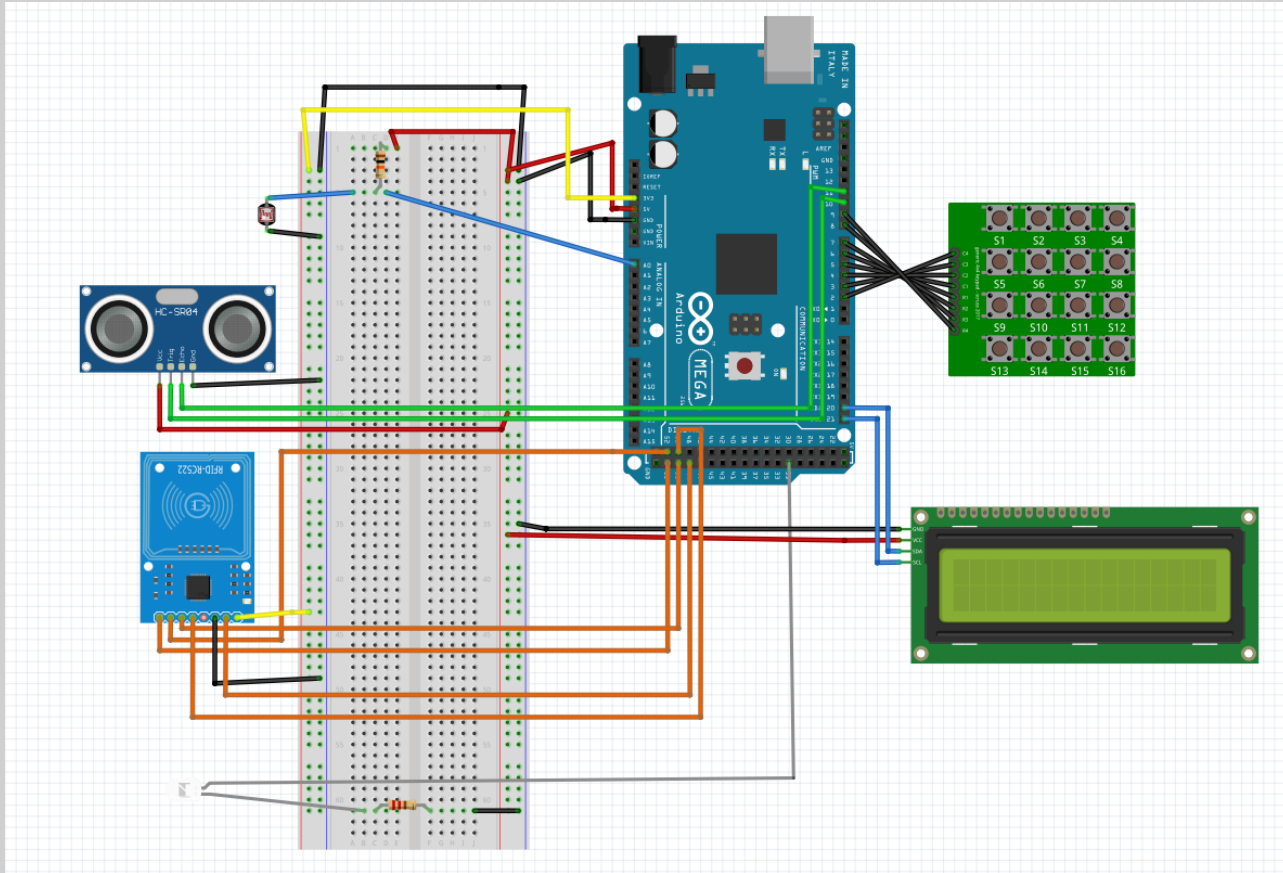
# 부가적인 기능

add



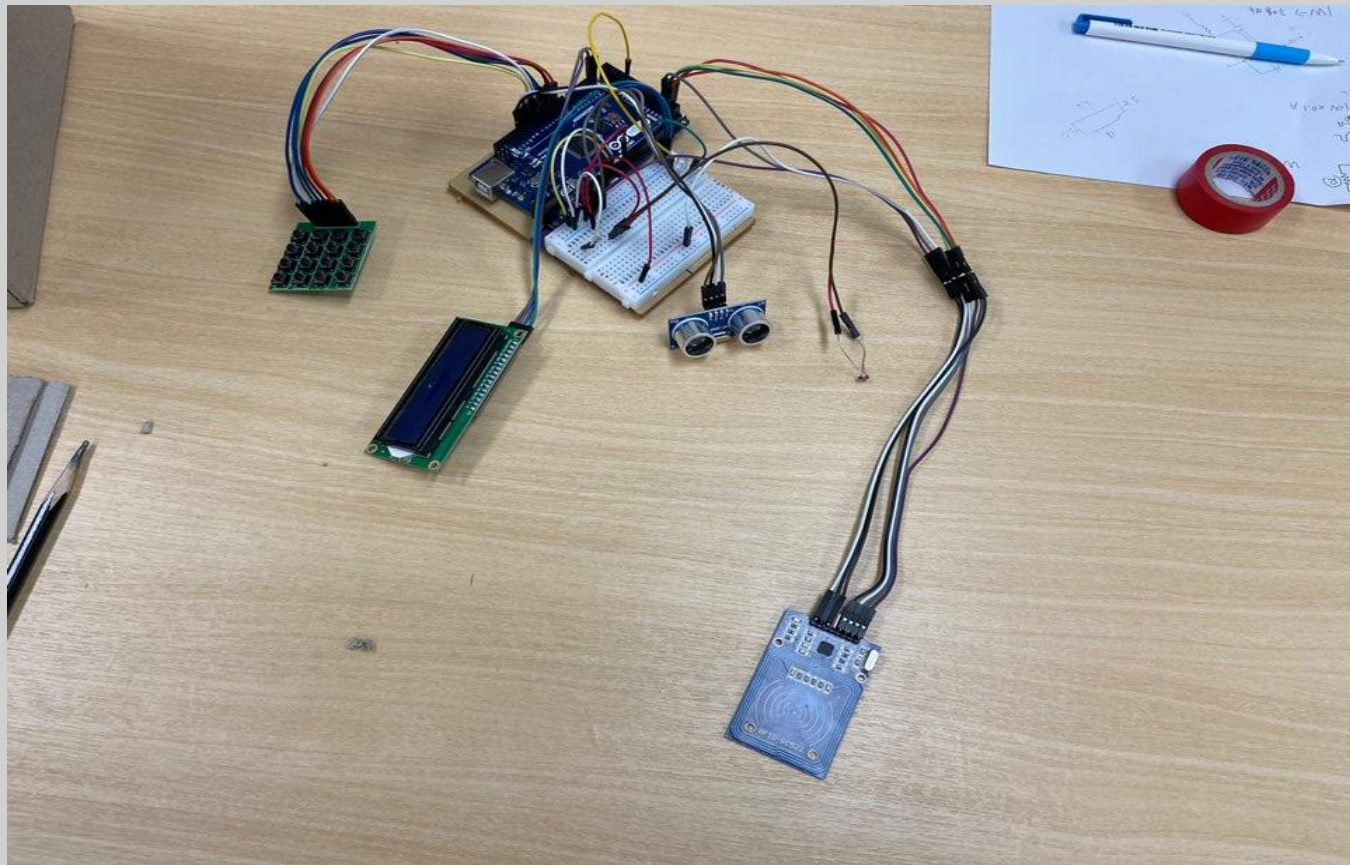
C버튼을 누르면 LED를 점등/소등 할 수 있고  
D버튼을 누르면 LCD의 Backlight를 점등/소등 할 수 있다.

# 회로도 및 회로구현



# 회로도 및 회로구현

---



# 아두이노 코드

```
1 #include <SPI.h>
2 #include <MFRC522.h>
3 #include <Keypad.h>
4 #include <Wire.h>
5 #include <LiquidCrystal_I2C.h>
6
7 LiquidCrystal_I2C lcd(0x27,16,2);
8 const byte ROWS = 4;    // 행(rows) 개수
9 const byte COLS = 4;    // 열(columns) 개수
10 char keys[ROWS][COLS] = {
11   {'1','2','3','A'},
12   {'4','5','6','B'},
13   {'7','8','9','C'},
14   {'*','0','#','D'}
15 };
16 #define SS_PIN 53
17 #define RST_PIN 49
18 const int trigPin = 10;
19 const int echoPin = 11;
20 const int CdSPin = 0;
21 const int led = 31;
22
23 int wrong = 0;
24 int wrong2 = 0;
25 int counter = 0;
26 int counter2 = 0;
27 int turnOn = 0;
28 int detected = 0;
29 int pause = 0;
30 byte rowPins[ROWS] = {6,7,8,9};    // R1, R2, R3, R4 단자가 연결된 아두이노 핀 번호
31 byte colPins[COLS] = {5,4,3,2};    // C1, C2, C3, C4 단자가 연결된 아두이노 핀 번호
32 Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
33
```

각 모듈들의  
핀 번호

센서의  
시작/종료를  
확인하기 위한  
변수

Keypad의  
Keymap생성

# 아두이노 코드

```
MFRC522 mfrc522(SS_PIN, RST_PIN); // Instance of the class
byte password[4]={50,52,54,56};
byte password2[4]={0,0,0,0};
byte rfid[8] = {0,0,0,0,0,0,0,0};
byte rfid2[8] = {244,189,86,211};
int rfidIsOk = 0;
int count = 0;
int switchOk = 0;
int back = 0;
long duration;
int distance;
boolean Lock = 0;
boolean ledCheck = 0;
void setup() {
    Serial.begin(9600);
    lcd.init(); // LCD 설정
    lcd.clear(); // LCD를 모두 지운다.
    lcd.backlight(); // 백라이트를 켜다.
    lcd.setCursor(0,0);
    lcd.print("press password");
    delay(3000);
    lcd.setCursor(0,1);
    lcd.print("or put Cardkey ");

    pinMode(led,OUTPUT);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);

    SPI.begin();
    mfrc522.PCD_Init();
    Serial.println("RFID reading UID");
}
```

비밀번호  
카드 키 설정

비밀번호와  
카드 키가  
입력되었을 때  
판단을 위한 변수

LCD를 설정하고  
첫 화면에 띄울  
문장 출력

LED와 초음파센서  
설정

# 아두이노 코드

```
void loop() {  
  
  if ( mfrc522.PICC_IsNewCardPresent())  
  {  
    if ( mfrc522.PICC_ReadCardSerial())  
    {  
      Serial.print("Tag UID:");  
      for (byte i = 0; i < mfrc522.uid.size; i++) {  
        Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");  
        Serial.print(mfrc522.uid.uidByte[i], HEX);  
      }  
      for(byte i = 0; i<8; i++)  
      {  
        rfid[i]=0;  
      }  
      for(byte i = 0; i<mfrc522.uid.size; i++)  
      {  
        rfid[i] = mfrc522.uid.uidByte[i];  
      }  
  
      rfidIsOk=0;  
      for(byte i = 0; i<mfrc522.uid.size; i++)  
      {  
        if(rfid[i] == rfid2[i])  
        {  
          rfidIsOk++;  
        }  
        else  
        {  
          wrong++;  
        }  
      }  
    }  
  }  
}
```

카드 키를  
입력했을 때

Serial번호를  
rfid에 저장 후

rfid2와 비교

# 아두이노 코드

```
if(wrong>0){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Wrong Cardkey");
    delay(3000);
    lcd.clear();
    lcd.print("press password");
    lcd.setCursor(0,1);
    lcd.print("or put Cardkey");
    wrong=0;
}
if(rfidIsOk==4){
    if(Lock==0)
    {
        Lock=1;
        lcd.clear();
        lcd.backlight();
        lcd.setCursor(0,0);
        lcd.print("Start Monitoring");
        delay(3000);
        lcd.clear();
    }
    else
    {
        Lock=0;
        lcd.clear();
        lcd.backlight();
        lcd.setCursor(0,0);
        lcd.print("Stop Monitoring");
        delay(5000);
        lcd.clear();
        lcd.print("press password");
        lcd.setCursor(0,1);
        lcd.print("or put Cardkey");
    }
}
```

카드 키의  
Serial번호가 틀리면  
"wrong Cardkey"  
3초 후 첫 화면 출력

맞으면 센서를  
시작하고  
"Start Monitoring"  
출력

다시 한번 입력하면  
"Stop Monitoring"  
5초 후 첫 화면 출력

# 아두이노 코드

```
char key = keypad.getKey();
| if(key) {
    pause = 1;
    lcd.backlight();
    Serial.println(key);
    password2[count]=key;
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(key);
    count++;
}
if(count==4)
{
    switchOk=0;
    for(int i =0; i<4; i++)
    {
        if(password2[i] == password[i])
        {
            switchOk++;
            pause = 0;
        }
        else
        {
            wrong2++;
            pause = 0;
        }
    }
}
```

Key 입력 4자리를  
받으면 비밀번호와  
비교하고

count를  
0으로 초기화

pause는 센서작동  
도중 비밀번호  
입력 시 delay  
때문에 작성



# 아두이노 코드

```
if(key=='A')
{
    lcd.backlight();
    lcd.setCursor(0,0);
    lcd.print("detected:");
    lcd.setCursor(0,1);
    lcd.print(detected);
    lcd.print("times");
    count =0;
    delay(3000);
    lcd.clear();
    if(Lock==0)
    {
        lcd.print("press password");
        lcd.setCursor(0,1);
        lcd.print("or put Cardkey");
    }
}
```

A버튼을 눌렀을 때

```
if(key=='B')
{
    lcd.backlight();
    lcd.setCursor(0,0);
    lcd.print("Turned on:");
    lcd.setCursor(0,1);
    lcd.print(turnOn);
    lcd.print("times");
    delay(3000);
    lcd.clear();
    if(Lock==0)
    {
        lcd.print("press password");
        lcd.setCursor(0,1);
        lcd.print("or put Cardkey");
    }
    count = 0;
}
```

B버튼을 눌렀을 때

# 아두이노 코드

---

```
if(Lock==1)
{
    if(pause==0)
    {
        digitalWrite(trigPin, LOW);
        delayMicroseconds(2);
        digitalWrite(trigPin, HIGH);
        delayMicroseconds(10);
        digitalWrite(trigPin, LOW);

        duration = pulseIn(echoPin, HIGH);

        distance= duration*0.034/2;

        lcd.setCursor(0,1);
        Serial.print("Distance: ");
        Serial.println(distance);
        delay(500);
        if(distance<100)
        {
            counter++;
            detected = counter;
        }
    }
}
else
{
    counter = 0;
}
```

```
if(Lock==1)
{
    if(pause==0)
    {
        lcd.setCursor(0,1);
        lcd.clear();

        int illuminance; // 현재의 밝기. 0~100%
        int adcValue; // 실제 센서로부터 읽은 값 (0~1023)

        // Cds cell을 통하여 입력되는 전압을 읽는다.
        adcValue = analogRead(CdSPin);
        // 아날로그 입력 값을 0~100의 범위로 변경한다.
        illuminance = map(adcValue, 0, 1023, 100, 0);
        Serial.print("illuminance: ");
        Serial.println(illuminance);
        if(illuminance>50)
        {
            counter2++;
            turnOn=counter2;
        }
    }
}
else
{
    counter2 = 0;
}
```

# 완성품

---



# 결과 [시연영상]

---



**THANK YOU**