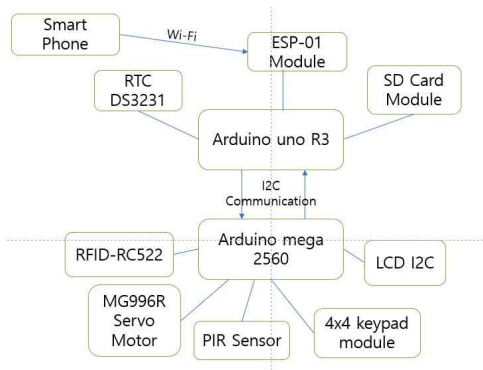


## 프로젝트 기술서

프로젝트 명	아두이노와 앱인벤터를 이용한 스마트 도어락		개발기간	2022.04 ~ 2022.12
구현기술	Arduino		개발 장소	순천향대학교
	C, C++			
	App Inventor		투입인원	2명
환경	개발 환경	OS	DataBase	
	Arduino IDE App Inventor	Windows 10 & 11	TinyDB (App Inventor 내에서 자체제공 됨)	
Project시연 Github 링크	<a href="https://github.com/companykim/graduation">https://github.com/companykim/graduation</a>			
프로젝트 소개	<b>[프로젝트의 시작]</b>  <b>1. 사전조사</b> 도어락 장치는 많은 가정집에서 사용할 정도로 보편화 되어 있고, 지금도 다양한 형태로 개발되고 있다. 이제는 IOT 기술이 발달함에 따라 이를 접목시킨 제품들이 개발되고 있지만 아직은 대중화되기 어려운 측면들이 있다.  <b>2. 프로젝트 개발방향</b> 이 프로젝트는 아두이노라는 교육용 소프트웨어를 활용해 3가지의 방식을 사용하여 문을 개폐할 수 있는 스마트 도어락을 개발하고자 함.  1) 키패드와 카드키를 사용 이 두가지 방식은 기존 도어락에 있는 방식들을 그대로 차용하여 알고리즘을 만들고 개발함.  2) 스마트폰 앱을 사용 앱인벤터로 간단히 앱을 제작하고, 아두이노와 스마트폰을 Wi-Fi로 연동하고 앱에서 비밀번호를 입력해서 문을 열고 닫는 알고리즘을 만듦.			
	<b>[프로젝트의 주요기능]</b>  <b>1. 키패드로 비밀번호를 입력해서 일치하면 문이 열리고 5초가 지난 이후 자동으로 문이 닫힘.</b>  <b>2. 카드를 RFID 모듈에 대면 문이 열리고 5초가 지난 이후 자동으로 문이 닫힘.</b>  <b>3. 비밀번호를 입력할 어플을 만듦. 비밀번호는 관리자 비밀번호를 입력하고 새로운 비밀번호를 입력하면 앱 인벤터 자체에 있는 DB에 저장되어 비밀번호를 변경 가능함.</b>			

- 아두이노 Wi-Fi 모듈(ESP-8266)을 통해 아두이노와 스마트폰을 연동
- 스마트폰 앱에서 비밀번호를 입력해서 일치하면 문이 열리고 5초가 지난 이후 자동으로 문이 닫힘.
- 문이 열렸을 때, 움직임이 감지될 경우 5초 정도의 딜레이가 발생하고 그 이후 자동으로 문이 닫힘.
- 키패드에서 # 버튼을 누르면 LCD화면에 랜덤 비밀번호가 생성되고 그것을 입력해야 열리도록 함.
- 문이 열리면 SD카드에 기록이 남도록 함.

[구성 및 흐름]



[프로젝트 구현 결과]

- 키패드로 비밀번호를 입력하고 일치하면 문이 열리고 5초 이후 문이 닫히는 알고리즘 구현.  
키패드에서 A를 입력하면 키패드 기능이 활성화되고, 비밀번호를 입력하고 일치하다면 문이 열린다. 비밀번호는 #버튼을 누르면 LCD 화면에 랜덤 비밀번호가 생성된다. 비밀번호는 랜덤 비밀번호를 생성한 뒤에 입력해야 정상적으로 문이 열린다.



## [Slave\_final.ino]

```
#include <Keypad.h>
const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};

byte rowPins[ROWS] = {A8, A9, A10, A11}; //R1 R2 R3 R4
byte colPins[COLS] = {A12, A13, A14, A15}; // C1 C2 C3 C4

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);
String input = "";
String PW = "A0203";
int keyCount;
```

(...생략...)

```
**** keypad starting ****/
randomSeed(analogRead(1));
char key = keypad.getKey();

if (key == 'A') {
  z = 1;
  delay(300);
}

val = digitalRead(48);
```

(...생략...)

```
*** keypad part ***/
if (z == 1)
{
  if (key)
  {
    if (key == '*')
    {
      if (input == PW)
      {
        Wire.beginTransmission(Slave);
        Wire.write(4);
        Wire.endTransmission();

        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("open the door");
        input = "";
        keyCount = 0;
        PW = " ";
        servo.write(0);
        delay(5000);

        val3 = digitalRead(48);
        for (int i = 0; i < 50; i++)
        {
          if (val3 == HIGH)
          {
            lcd.setCursor(0, 1);
            lcd.print("movement..");
            servo.write(0);
            delay(100);
          }
        }
      }
    }
  }
}
```

(...생략...)

```

        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("close the door");
        servo.write(160);
        delay(3000);
        lcd.clear();
        lcd.print("ready...");
        z = 0;
    }
    else // wrong password
    {
        PW = " ";
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("wrong password");
        input = "";
        keyCount = 0;
    }
}

else if (key == '#')
{
    int val = random(0, 9999);
    String ranpw = String(val);
    PW = 'A'+ranpw;
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("new password: ");
    lcd.setCursor(0, 1);
    lcd.print(ranpw);
}
else
{
    input += key;
}
delay(100);
}
}

```

## 2. 카드를 RFID 모듈에 대면 문이 열리고 5초 뒤에 문이 닫히는 알고리즘

RFID 모듈에 댄 카드가 지정해둔 카드키와 일치하다면 문이 열린다.



[Slave\_final.ino]

```

/****RFID setting****/
#include <SPI.h>
#include <MFRC522.h>
#define RST_PIN 49
#define SS_PIN 53
MFRC522 mfrc(SS_PIN, RST_PIN);

```

(...생략...)

```

/***** RFID part *****/
if (z == 0) {
  if ( !mfrC.PICC_IsNewCardPresent() || !mfrC.PICC_ReadCardSerial() ) {
    return;
  }
  String content = "";
  byte letter;
  for(byte i = 0; i < mfrC.uid.size; i++) {
    content.concat(String(mfrC.uid.uidByte[i] < 0x10 ? " 0" : " "));
    content.concat(String(mfrC.uid.uidByte[i], HEX));
  }
  Serial.println();

  content.toUpperCase();

  if(content.substring(1) == "63 D8 99 06")
  {
    Wire.beginTransmission(Slave);
    Wire.write(3);
    Wire.endTransmission();

    servo.write(0);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("open the door");
    delay(5000);
  }
}

```

(...생략...)

```

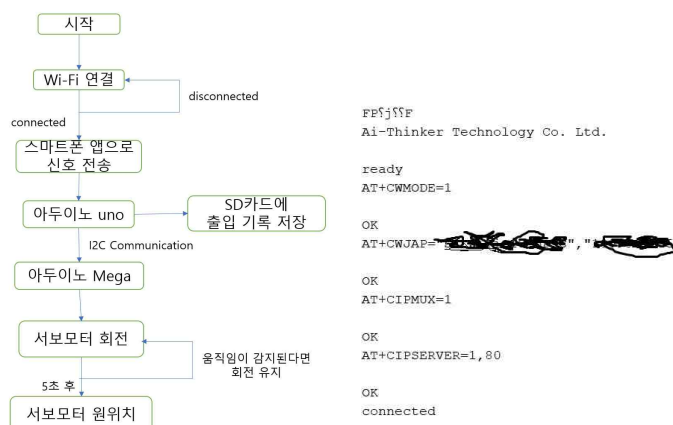
val2 = digitalRead(40);
for (int i = 0; i < 50; i++)
{
  if (val2 == HIGH)
  {
    lcd.setCursor(0, 1);
    lcd.print("movement..");
    servo.write(0);
    delay(100);
  }

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("close the door");
  servo.write(160);
  delay(3000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("ready...");
}
else {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("wrong card");
}
}
}

```

### 3. Wi-Fi 모듈로 아두이노와 휴대폰을 연동하고, 앱에서 비밀번호를 입력하고 그것이 일치하면 문이 열리고 5초가 지나면 자동으로 문이 닫히는 알고리즘

아두이노에서 ESP8266이라는 모듈을 사용하여 아두이노에서 Wi-Fi를 연결할 수 있도록 하고, 스마트폰 앱에서 동일한 Wi-Fi 환경일 때, 입력한 비밀번호가 일치하면 문이 열리도록 함. Wi-Fi 주소는 주소와 비밀번호를 IDE에 입력하고 업로딩을 할 때 Serial Monitor에서 연결되었음을 확인할 수 있다.



## [Master\_final.ino]

```
/*ESP-01 setting*/
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2, 3); //RX TX
#define DEBUG true
```

(...생략...)

```
/*wifi connecting */
sendData("AT+RST\r\n",2000,DEBUG);
sendData("AT+CMODE=1\r\n",1000,DEBUG);
sendData("AT+CWJAP=\"SK_WiFiGIGA5D9C\", \"1809064185\"\r\n", 5000, DEBUG);
delay(1000);
sendData("AT+CWMUX=1\r\n",1000,DEBUG);
sendData("AT+CIPSERVER=1,80\r\n",1000,DEBUG);

Serial.println("connected");
```

(...생략...)

```
/*ESP-01 loop*/
int pinNumber;
if(mySerial.available()) //
{
    if(mySerial.find("+IPD,"))
    {
        delay(1000);
        int connectionID = (mySerial.read()-48);
        mySerial.find("pin=");

        pinNumber = (mySerial.read()-48)*10;
        pinNumber += (mySerial.read()-48);

        if (pinNumber == 13)
        {
            /*SD(wifi)*/
            File dataFile1 = SD.open("DB.txt", FILE_WRITE);
            if (dataFile1) {
                dt1 = clock.getDateTime();
                dataFile1.print(dt1.year); dataFile1.print("-");
                dataFile1.print(dt1.month); dataFile1.print("-");
                dataFile1.print(dt1.day); dataFile1.print(" ");
                dataFile1.print(dt1.hour); dataFile1.print(":");
                dataFile1.print(dt1.minute); dataFile1.print(":");
                dataFile1.print(dt1.second);
                dataFile1.println(" door used by smartphone");
                dataFile1.close();
            }
        }
    }
}
```

(...생략...)

```
/*open at wifi*/
Wire.beginTransmission(Slave);
Wire.write(1);
Wire.endTransmission();
delay(5000);

/*close at wifi*/
Wire.beginTransmission(Slave);
Wire.write(2);
Wire.endTransmission();
delay(1000);

String closeCommand = "AT+CIPCLOSE=";
closeCommand += connectionID;
closeCommand += "\r\n";
sendData(closeCommand, 1000, DEBUG);
}
}
```

(...생략...)

```
/*ESP-01의 정보를 설정하기 위한 함수*/
String sendData(String command, const int timeout, boolean debug) {
    String response = "";
    mySerial.print(command);
    long int time = millis();
    while( (time+timeout) > millis()) {
        while(mySerial.available()) {
            char c = mySerial.read();
            response+=c;
        }
    }
    if(debug) {
        Serial.print(response);
    }
    return response;
}
```

#### [slave\_final.ino]

```
/****** Wifi part *****/
Wire.onReceive(record);

if (rec[0] == 1) {
  servo.write(0);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("open the door");
  delay(5000);
}

if (rec[0] == 2) {
  if (val == HIGH) {
    servo.write(0);
    lcd.setCursor(0, 1);
    lcd.print("movement..");
    delay(3000);
  }

  else {
    rec[0] = 3;
    servo.write(180);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("close the door");
    delay(3000);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("ready...");
  }
}
```

#### 4. 문이 열렸을 때마다 SD카드에 출입 기록이 저장되도록 함.

블랙박스처럼 문이 열렸을 때, SD 카드에 열렸을 당시의 날짜와 시간, 어떤 방식으로 열렸는지가 TXT 파일로 생성됨. 문장은 (날짜) door used by (...) 형식으로 출력됨.

```
2022-11-1 0:7:25 door used by smartphone
2022-11-1 0:22:3 door used by smartphone
2022-11-1 15:14:0 door used by smartphone
2022-11-1 15:17:29 door used by RFID
2022-11-1 15:17:38 door used by RFID
2022-11-1 15:19:18 door used by smartphone
2022-11-1 15:20:10 door used by smartphone
2022-11-1 15:20:30 door used by smartphone
2022-11-1 15:20:48 door used by smartphone
2022-11-1 15:22:39 door used by smartphone
2022-11-1 15:22:55 door used by smartphone
2022-11-1 15:23:14 door used by smartphone
2022-11-1 15:23:23 door used by smartphone
2022-11-2 16:37:29 door used by smartphone
2022-11-2 16:40:33 door used by smartphone
2022-11-2 16:40:48 door used by smartphone
2022-11-2 16:41:11 door used by smartphone
2022-11-2 16:42:15 door used by smartphone
2022-11-2 16:49:51 door used by smartphone
2022-11-2 16:50:3 door used by smartphone
```

#### [Master\_final.ino]

```
/***I2C setting***/
#include <Wire.h>
#define Slave 0x04
byte rec[128];
```

(...생략...)

```
/***SD setting***/
#include <SD.h>
#include <SPI.h>

/***DS3231 setting***/
#include <DS3231.h>
DS3231 clock;
RTCDateTime dt1;
RTCDateTime dt2;
RTCDateTime dt3;

void setup() {
  Wire.begin(Slave);
  SPI.begin();
  Serial.begin(9600);
  clock.begin();
  mySerial.begin(9600);
```

(...생략...)

```

/****SD setup****/
Serial.println("Initializing SD card...");

if (!SD.begin(10)) {
  Serial.println("Card failed");
  while(1);
}
Serial.println("Card initialized");
Serial.println();

/***** wifi connecting *****/
sendData("AT+RST\r\n",2000,DEBUG);
sendData("AT+CWMODE=1\r\n",1000,DEBUG);
sendData("AT+CWJAP=\"SK_WiFiGIGA5D9C\", \"1809064185\"\r\n", 5000, DEBUG);
delay(1000);
sendData("AT+CWMUX=1\r\n",1000,DEBUG);
sendData("AT+CIPSERVER=1,80\r\n",1000,DEBUG);

Serial.println("connected");

/****DS3231 setup****/
clock.setDateTime(__DATE__, __TIME__);
}

```

(...생략...)

```

/*****SD(wifi)*****/
File dataFile1 = SD.open("DB.txt", FILE_WRITE);
if (dataFile1) {
  dt1 = clock.getDateTime();
  dataFile1.print(dt1.year); dataFile1.print("-");
  dataFile1.print(dt1.month); dataFile1.print("-");
  dataFile1.print(dt1.day); dataFile1.print(" ");
  dataFile1.print(dt1.hour); dataFile1.print(":");
  dataFile1.print(dt1.minute); dataFile1.print(":");
  dataFile1.print(dt1.second);
  dataFile1.println(" door used by smartphone");
  dataFile1.close();
}

```

(...생략...)

```

Wire.onReceive(record);
if (rec[0] == 3) {
  /*****SD(RFID)*****/
  File dataFile2 = SD.open("DB.txt", FILE_WRITE);
  if (dataFile2) {
    dt2 = clock.getDateTime();
    dataFile2.print(dt2.year); dataFile2.print("-");
    dataFile2.print(dt2.month); dataFile2.print("-");
    dataFile2.print(dt2.day); dataFile2.print(" ");
    dataFile2.print(dt2.hour); dataFile2.print(":");
    dataFile2.print(dt2.minute); dataFile2.print(":");
    dataFile2.print(dt2.second);
    dataFile2.println(" door used by RFID");
    dataFile2.close();
  }
  rec[0] = 10;
}

```

(...생략...)

```

if (rec[0] == 4) {
  /*****SD(Keypad)*****/
  File dataFile3 = SD.open("DB.txt", FILE_WRITE);
  if (dataFile3) {
    dt3 = clock.getDateTime();
    dataFile3.print(dt3.year); dataFile3.print("-");
    dataFile3.print(dt3.month); dataFile3.print("-");
    dataFile3.print(dt3.day); dataFile3.print(" ");
    dataFile3.print(dt3.hour); dataFile3.print(":");
    dataFile3.print(dt3.minute); dataFile3.print(":");
    dataFile3.print(dt3.second);
    dataFile3.println(" door used by keypad");
    dataFile3.close();
  }
  rec[0] = 20;
}

```

[Slave\_final.ino]

```

Wire.beginTransmission(Slave);
Wire.write(3);
Wire.endTransmission();

```

## 프로젝트 결과분석

앱인벤터에서 OTP 방식을 사용해 몇분 뒤에 자동 폐기되는 임시 비밀번호를 문자 메시지로 보내고 그 비밀번호



호를 앱에서 입력하게끔 하고 싶었으나 문자를 보내는 시간이 너무 오래 걸려서 하지 못했던 것이 굉장히 아쉽다.

또, 스마트폰에서 바뀐 비밀번호를 아두이노의 키패드에서도 사용할 수 있게 개발한다면 비밀번호가 혼동되지 않는 효과가 생길 것이라 생각한다.