

과목	컴퓨터 시스템 입문
프로젝트 명	스마트 도어락 시스템
과제 기간	2017. 3. 26 ~ 2017. 6. 21 (약 3개월)
참여 인원	4명



팀 원	학과	학년	학번	E-mail
김윤기	디지털정보공학과	3	201300712	sospecial30@naver.com
박다혜	컴퓨터전자시스템공학부	3	201501202	skssskdy@naver.com
김민근	컴퓨터전자시스템공학부	3	201500410	sdfg1289@naver.com
김지수	컴퓨터전자시스템공학부	2	201600799	jisu9863@naver.com

## **목차**

### **I. 개요**

- 1) 개발 목적
- 2) 개발 목표

### **II. 개발 과정**

- 1) 전체 설계
- 2) 개발 일정
- 3) 부품 설명
- 4) 진행 과정

### **III. 세부 작품 내용**

- 1) 구상도
- 2) 소스코드
- 3) 회로도
- 4) 작동 과정 및 결과
- 5) 문제 해결 과정

### **IV. 결론**

- 1) 분석 및 고찰
- 2) 향후 계획

### **V. 참고자료**

## I. 개요

### 1) 개발 목적

맞벌이 가족이 증가함에 따라 집이 비거나 아이가 혼자서 집을 지키게 되는 경우가 많아졌고, 그 것을 노리고 주거침입을 시도하여 범행을 저지르려는 사람들 또한 늘어나고 있다. 하지만 현재 우리가 흔히 사용하는 기존 도어락의 기능으로는 낯선 사람이 비밀번호를 알게 되면 쉽게 출입이 가능하고, 범행 시간동안 집 주인은 그 것을 알 수 없어서 범행이 이미 일어난 뒤 그것을 해결해야하는 어려움 등이 발생한다. 이는 가정집뿐 만이 아니라 중요한 정보가 많은 사무실이나 연구실 등에서도 마찬가지일 것이다.

지금은 4차 산업혁명을 이끌어갈 사물인터넷(IoT)시대이다. 생소하기만 했던 이 단어가 어느새 우리 생활 속에 깊이 스며들어있다. 우리는 이 점점 발전하는 사물인터넷 기술을 접목하여 이러한 문제점을 보완하여 몇 가지 기능을 추가한 도어락을 만들어보고자 하였다.

### 2) 개발 목표

- 수업시간에 학습한 아두이노와 여러 모듈의 사용법을 활용하여 도어락을 구현해 본다.
- 카메라 모듈, SD카드 모듈을 이해하고 사용법을 익혀 도어락에 적용한다.
- Blynk라는 어플리케이션을 이용하여 원격으로 아두이노를 컨트롤할 수 있는 방안을 도색해 본다.
- 프로젝트를 진행하며 발생하는 여러 가지 문제점과 그에 대한 해결방안을 통한 주체적인 학습을 진행한다.

## II. 개발 과정

### 1) 전체 설계

- 키패드, 블루투스를 이용한 입력 제어
- passcheck 알고리즘, flag연산을 통한 출력의 방향 결정
- 비밀번호 판별 결과에 따른 LED, 부저, 서보모터, Serial출력
- TWI 통신을 통해 두 아두이노 우노 보드를 연결
- 카메라 모듈과 SD카드 리더기 모듈을 사용한 사진 촬영 및 저장
- Blynk 어플리케이션을 이용한 도어락 원격 제어 및 확인 시스템


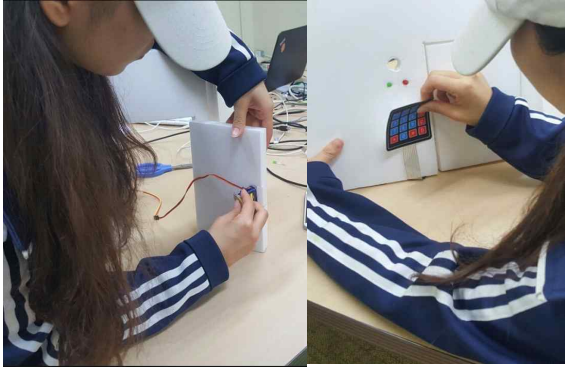
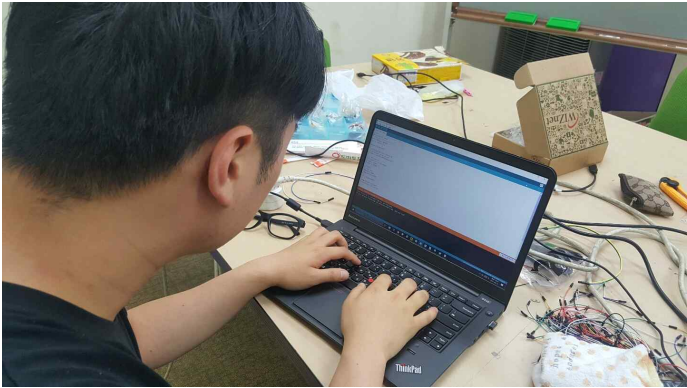
## 2) 개발 일정

날짜	활동	내용	역할 분담
3/26	아이디어 회의	스마트 도어락 시스템(인증 절차를 거쳐 등록된 스마트 폰으로 잠금 해제)	김윤기, 박다혜, 김민근, 김지수
3/26 ~ 4/24	각자 자료조사	안드로이드, 와이파이, 도어락	김윤기, 박다혜, 김민근, 김지수
4/24	아이디어 회의	아이디어 수정(+사진, 휴대폰으로 상태 전송), 역할 분배	김윤기, 박다혜, 김민근, 김지수
5/4	물품 구매	키패드 모듈	박다혜
5/9 ~ 5/10	블루투스 기능 있는 도어락 구현	서보모터, 거리감지센서,	박다혜
		부저, LED, 스위치	김지수
		키패드	김민근
		블루투스	김윤기
5/15	집 모형 제작	집 모형 만들고, 모듈 연결	김윤기, 박다혜, 김민근, 김지수
5/15	도어락 전체 구현	코드 일부 수정(부저)	김지수, 박다혜
		순서대로 도어락 구현	김윤기, 박다혜, 김민근, 김지수
5/17	도어락 전체 구현	코드 일부 수정 및 전체 구현	김윤기, 박다혜, 김민근, 김지수
5/30	자료 조사	클라우드 서비스(Ubidots)	박다혜
		Blynk	김윤기, 김민근
		카메라 모듈	김지수
6/7	자료조사, 물품구매	사진 서버 전송 방법 조사	박다혜, 김지수
		Blynk 사용	김윤기, 김민근
		카메라 모듈, SD카드 어댑터 구매	김지수, 박다혜
6/18 ~ 6/19	각자 파트 코드 작성	사진 촬영 및 사진 저장	김지수, 박다혜
		아두이노 TWI 통신	김지수
		Blynk	김윤기, 김민근
6/20 ~ 6/21	보고서작성, 코드 수정, 전체 동작 확인	코드 수정 및 전체 동작 확인	김윤기, 박다혜, 김민근, 김지수
		보고서 작성	김지수
6/21	보고서, 영상	보고서 작성, 동영상 촬영	김윤기, 박다혜, 김민근, 김지수
		동영상 편집	김지수

### 3) 부품 설명

부품 사진	이름 / 개수	부품 사진	이름 / 개수
	아두이노 우노 보드 / 2		아두이노 메가 + 와이파이 쉴드(wizfi 310) / 1
	서보 모터 / 1		거리 감지 센서 / 1
	키패드 모듈 / 1		블루투스 모듈 / 1
	부저 / 1		LED / 2
	스위치 모듈 / 1		저항 / 여러개
	TTL시리얼 JPEG 카메라 모듈 / 1		MicroSD 카드 리더기 모듈 / 1

#### 4) 진행 과정 모습

사진	내용
	<p>집 모형 제작</p>
	<p>만든 집 모형에 필요한 모듈 설치</p>
	<p>코드 구현 (키패드 등등, 카메라, Blynk)</p>
	<p>회로도 만들기</p>



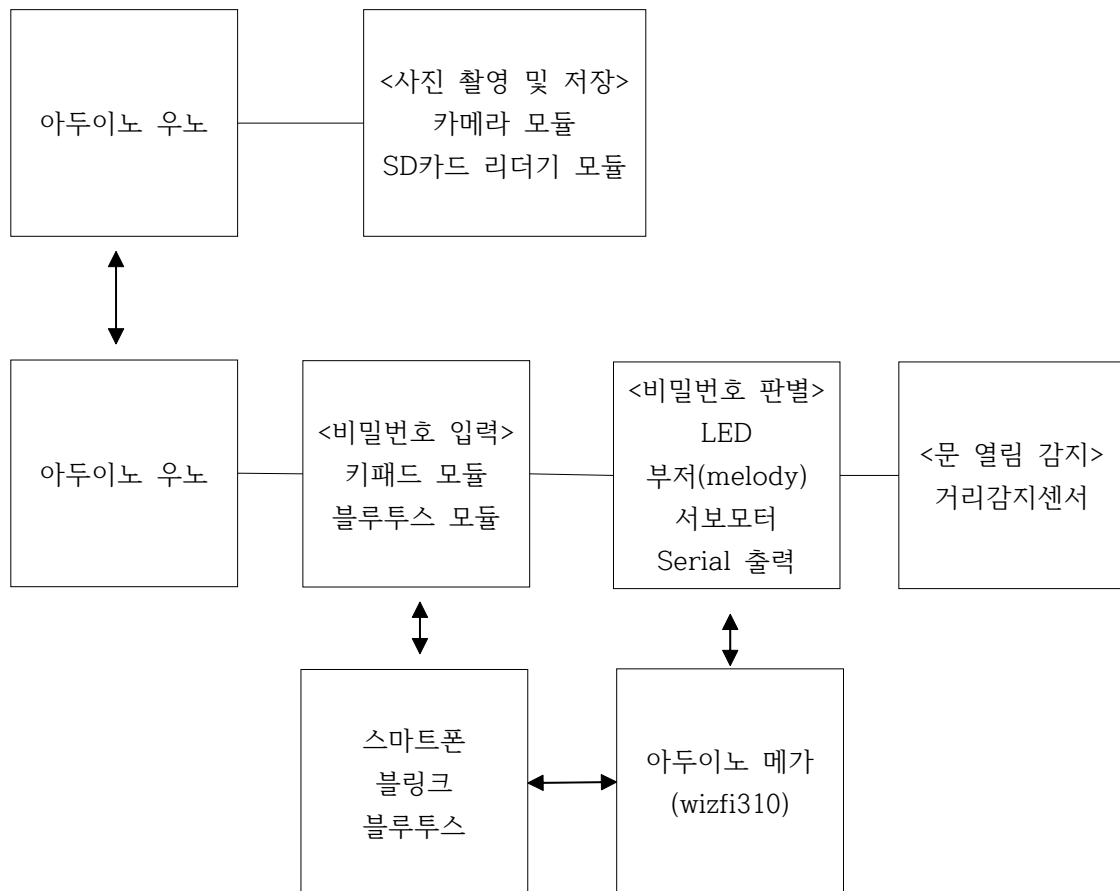
카메라 모듈을 좀 더 수월하게  
사용하기 위하여 납땜 작업



보드를 3개나 사용하여  
복잡해진 선들을 정리

### Ⅲ. 세부 작품 내용

#### 1) 구상도



#### 2) 소스코드 (주석)

<Keypad module - Arduino Uno 1>

```
#include <Wire.h> //아두이노끼리 TWI 통신하기위한 라이브러리
#include <Adafruit_VC0706.h> //camera 모듈 라이브러리
#include <SPI.h> //SD카드 모듈과 SPI 통신하기위한 라이브러리
#include <SD.h> //SD카드 모듈 라이브러리
```

```
#include <SoftwareSerial.h>
```

```
#define chipSelect 10
```

```
// Rx, Tx 핀 설정
```

```
// Using SoftwareSerial (Arduino 1.0+) or NewSoftSerial (Arduino 0023 & prior):
```

```
#if ARDUINO >= 100
```

```
// On Uno: camera TX connected to pin 2, camera RX to pin 3:
```



```

SoftwareSerial cameraconnection = SoftwareSerial(2, 3);
#else
NewSoftSerial cameraconnection = NewSoftSerial(2, 3);
#endif

//camera 객체 생성
Adafruit_VC0706 cam = Adafruit_VC0706(&cameraconnection);

int flag = 0; //receiveEvent 발생 확인 변수

void setup() {
  Wire.begin(4); //TWI통신 설정
  Wire.onReceive(receiveEvent);

  // When using hardware SPI, the SS pin MUST be set to an
  // output (even if not connected or used). If left as a
  // floating input w/SPI on, this can cause lockuppage.
  #if !defined(SOFTWARE_SPI)
  #if defined(__AVR_ATmega1280__) || defined(__AVR_ATmega2560__)
    if(chipSelect != 53) pinMode(53, OUTPUT); // SS on Mega
  #else
    if(chipSelect != 10) pinMode(10, OUTPUT); // SS on Uno, etc.
  #endif
  #endif

  Serial.begin(9600);
  Serial.println("Camera start");

  // see if the card is present and can be initialized:
  if (!SD.begin(chipSelect)) {
    Serial.println("Card failed, or not present");
    // don't do anything more:
    return;
  }

  // Try to locate the camera
  if (cam.begin()) {
    Serial.println("Camera Found:");
  } else {
    Serial.println("No camera found");
    return;
  }
}

```

```

}

// Print out the camera version information (optional)
char *reply = cam.getVersion();
if (reply == 0) {
    Serial.print("Failed to get version");
} else {
    Serial.println("-----");
    Serial.print(reply);
    Serial.println("-----");
}

// Set the picture size - you can choose one of 640x480, 320x240 or 160x120
// Remember that bigger pictures take longer to transmit!

//cam.setImageSize(VC0706_640x480);      // biggest
//cam.setImageSize(VC0706_320x240);      // medium
cam.setImageSize(VC0706_160x120);        // small
}

void loop() {
    delay(100);
    if(flag == 1){
        uint8_t imgsize = cam.getImageSize();
        Serial.print("Image size: ");
        if (imgsize == VC0706_640x480) Serial.println("640x480");
        if (imgsize == VC0706_320x240) Serial.println("320x240");
        if (imgsize == VC0706_160x120) Serial.println("160x120");
        Serial.println("Snap in 3 secs...");
        delay(3000);
        if (! cam.takePicture())
            Serial.println("Failed to snap!");
        else
            Serial.println("Picture taken!");

        // Create an image with the name IMAGExx.JPG
        char filename[13];
        strcpy(filename, "IMAGE00.JPG");
        for (int i = 0; i < 100; i++) {
            filename[5] = '0' + i/10;
            filename[6] = '0' + i%10;

```

```

    // create if does not exist, do not open existing, write, sync after write
    if (! SD.exists(filename)) {
        break;
    }
}
// Open the file for writing
File imgFile = SD.open(filename, FILE_WRITE);

// Get the size of the image (frame) taken
uint16_t jpglen = cam.frameLength();
Serial.print("Storing ");
Serial.print(jpglen, DEC);
Serial.print(" byte image.");

int32_t time = millis();
pinMode(8, OUTPUT);

// Read all the data up to # bytes!
byte wCount = 0; // For counting # of writes
while (jpglen > 0) {
    // read 32 bytes at a time;
    uint8_t *buffer;
    uint8_t bytesToRead = min(32, jpglen); // change 32 to 64 for a speedup but
may not work with all setups!
    buffer = cam.readPicture(bytesToRead);
    imgFile.write(buffer, bytesToRead);
    if(++wCount >= 64) { // Every 2K, give a little feedback so it doesn't appear
locked up
        Serial.print('.');
        wCount = 0;
    }
    //Serial.print("Read "); Serial.print(bytesToRead, DEC); Serial.println(" bytes");
    jpglen -= bytesToRead;
}
imgFile.close();

time = millis() - time;
Serial.println("done!");
Serial.print(time); Serial.println(" ms elapsed");
Serial.println("-----");

```

```

    }
    flag = 0;
}

void receiveEvent(int howMany){
    while(Wire.available()){
        char c = Wire.read();
        Serial.print(c);
    }
    flag = 1;
    Serial.println("");
}

```

### <Camera module - Arduino Uno 2>

```

#include <Wire.h> //아두이노끼리 TWI 통신하기위한 라이브러리
#include <Servo.h>
#include <SoftwareSerial.h>
#include "pitches.h" //부저음 라이브러리
#include <Keypad.h>

```

```

#define passwordLength 5

```

```

SoftwareSerial BTSerial(2, 3); //블루투스 (TX, RX) 핀 설정
Servo servo;

```

```

int servoPin = A3; //서보모터 핀
int distancePin = A0; //거리감지센서 핀
int buzzerPin = A2; //부저 핀
int greenLed = 4;
int redLed = 5;
int switchPin = A1; //스위치 핀

```

```

char password[passwordLength]={'1','2','3','4','#'};
int flag = 0; //입력된 비밀번호 상태 변수
char Key[100];
char Blue[100];
int i;

```

```

void passcheck(char c[]){
    int i;
    for(i=0;i<passwordLength;i++){

```

```

    if(c[i]!=password[i]){ //비밀번호가 틀렸을 때
        flag=-1;
        break;
    }
    else{ //맞았을 때
        flag=1;
    }
}
}
}

```

```

const byte numRows= 4; //number of rows on the keypad
const byte numCols= 4; //number of columns on the keypad

```

```

//keymap defines the key pressed according to the row and columns just as
appears on the keypad
char keymap[numRows][numCols]=
{
{'1', '2', '3', 'A'},
{'4', '5', '6', 'B'},
{'7', '8', '9', 'C'},
{'*', '0', '#', 'D'}
};

```

```

//Code that shows the the keypad connections to the arduino terminals
byte rowPins[numRows] = {13,12,11,10}; //Rows 0 to 3
byte colPins[numCols]= {9,8,7,6}; //Columns 0 to 3

```

```

//initializes an instance of the Keypad class
Keypad myKeypad= Keypad(makeKeymap(keymap), rowPins, colPins, numRows,
numCols);

```

```

void setup()
{
    Wire.begin();
    servo.attach(servoPin);
    Serial.begin(9600);
    BTSerial.begin(9600);
    Serial.println("Doorlock program!");
    servo.write(10); //서보모터 초기 각도 설정 -> 잠김상태
    pinMode(greenLed, OUTPUT);
}

```

```

pinMode(redLed, OUTPUT); // 빨간불
pinMode(switchPin, INPUT_PULLUP);
}

//If key is pressed, this key is stored in 'keypressed' variable
//If key is not equal to 'NO_KEY', then this key is printed out

void loop()
{
    int buttonState = digitalRead(switchPin); //스위치 눌리면 열림
    if(buttonState == 0){
        flag = 1;
    }
    char keypressed = myKeypad.getKey();

    Key[i]=keypressed; //입력된 키값을 배열에 저장
    if(Key[i]!='#'){ // #이 입력되면 비밀번호 체크
        passcheck(Key);
        Serial.println("quit");
        if(flag==1){
            Serial.println("Wrong password!");
        }
        i=0;
    }
    else{
        if (keypressed != NO_KEY){ // #이 아닌 다른 키값이 입력되면
            if(i==0){ //첫번째 입력이 들어오면 camera와 연결된 아두이노에
                Serial.println("Send TWI signal to camera");//신호를 보냄
                Wire.beginTransmission(4);
                Wire.write("picture");
                Wire.endTransmission();
            }
            Serial.print(keypressed);
            i++;
        }
    }
}

if (BTSerial.available()) { //블루투스 입력
    Blue[i]=BTSerial.read();
    if(Blue[i]!='#'){
        passcheck(Blue);
    }
}

```

```

        Serial.println("quit");
        if(flag== -1)
            Serial.println("Wrong password!");
        i=0;
    }
    else{
        Serial.write(Blue[i]);
        i++;
    }
}
if(flag == 1){ //비밀번호가 맞았을 때
    servo.write(100); // 각도 100도로 이동 ->잠금해제
    correctMelody();
    Serial.println("Door open");
    delay(3000);

    while(1){ //문이 닫힐때 까지 distance 체크
        int distanceValue = analogRead(distancePin);
        float distance = 12343.85 * pow(distanceValue, -1.15); //문과의 거리 측정 변수
        delay(1500);
        Serial.println(distance);
        if(distance < 9){ //문이 닫히면
            distanceValue = analogRead(distancePin);
            distance = 12343.85 * pow(distanceValue, -1.15);
            Serial.println(distance);
            delay(2000);
            if(distance < 9){
                servo.write(10); // 각도 10도로 이동
                closeMelody();
                flag = 0;
                break;
            }
        }
    }
}
else if(flag == -1){ //비밀번호가 틀렸을 때
    wrongMelody();
    flag = 0;
    Serial.println("Send TWI signal to Byink");
    Wire.beginTransmission(5);
    Wire.write("warning");
}

```

```

        Wire.endTransmission();
    }
}

//비번이 맞았을 경우 멜로디
int melody1[] = {NOTE_C1, NOTE_AS6, NOTE_D7, 0}; //멜로디 배열
int noteDurations1[] = {6, 6, 6, 6}; //음의 길이 배열(4 note, 8 note, 16 note, etc.)
//비번이 틀렸을 경우 멜로디
int melody2[] = {NOTE_A7, NOTE_F1, NOTE_A7, NOTE_F1, NOTE_A7, NOTE_F1};
int noteDurations2[] = {8, 4, 8, 4, 8, 4};
//문이 닫혔을 경우 멜로디
int melody3[] = {NOTE_D7, NOTE_AS6, NOTE_C1, 0}; //멜로디 배열
int noteDurations3[] = {6, 6, 6, 6}; //음의 길이 배열(4 note, 8 note, 16 note, etc.)

void correctMelody(){
    digitalWrite(greenLed, HIGH);
    for (int thisNote1 = 0; thisNote1 < 4; thisNote1++){
        int noteDuration1 = 1000 / noteDurations1[thisNote1]; //음의 길이 계산 (1초를
note type으로 나누어줌)
        tone(buzzerPin, melody1[thisNote1], noteDuration1);
        delay(noteDuration1*1.30); // 음 구분을 위해 delay시간 필요(음의 길이 + 30%)
        noTone(buzzerPin);
    }
    digitalWrite(greenLed, LOW);
}

void wrongMelody(){
    for (int thisNote2 = 0; thisNote2 < 6; thisNote2++){
        int noteDuration2 = 1000 / noteDurations2[thisNote2];
        digitalWrite(redLed, HIGH);
        tone(buzzerPin, melody2[thisNote2], noteDuration2);
        delay(noteDuration2*1.30);
        noTone(buzzerPin);
        digitalWrite(redLed, LOW);
    }
}

void closeMelody(){
    for (int thisNote3 = 0; thisNote3 < 4; thisNote3++){
        int noteDuration3 = 1000 / noteDurations3[thisNote3];

```



```

    tone(buzzerPin, melody3[thisNote3], noteDuration3);
    delay(noteDuration3*1.30);
    noTone(buzzerPin);
  }
}

```

### <Blynk - arduino mega>

```

/* Comment this out to disable prints and save space */
#define BLYNK_PRINT Serial
/* Defines the serial connected to WizFi310 */
#define SERIAL_WIFI Serial3

#include <Blynk_WizFi310.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "3d525b0bc3d84f9a86a42ea65be8dcac ";

// WiFi ssid와 password
char ssid[] = "mingeun";
char pass[] = "13579086";

void setup(){

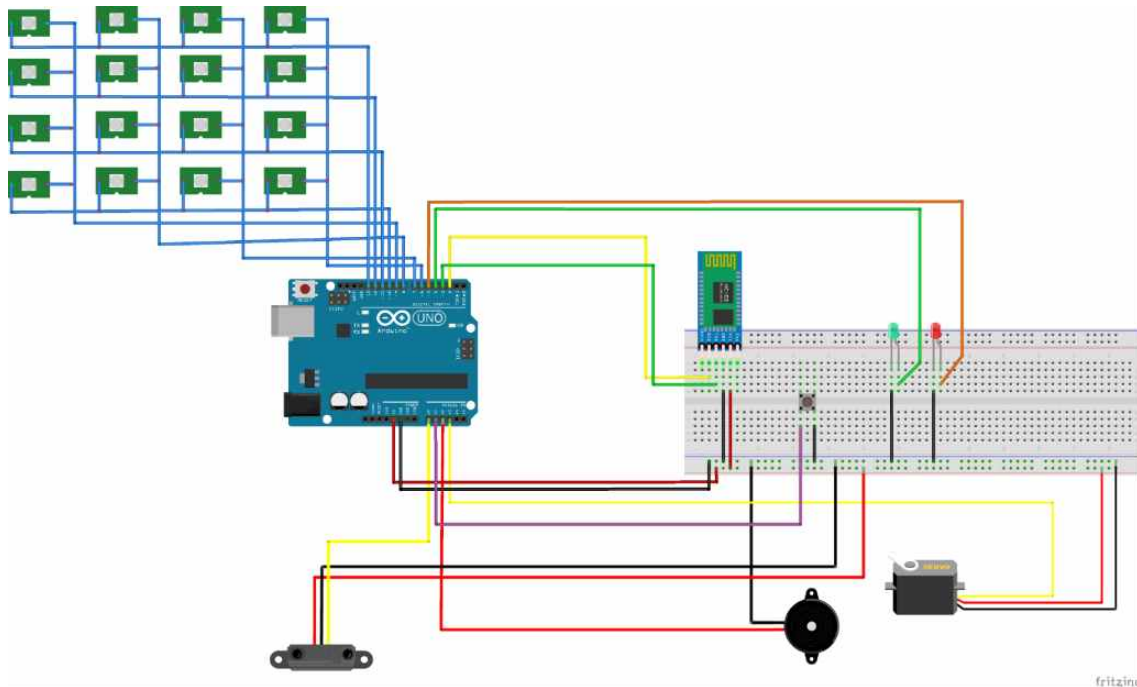
  Serial.begin(9600);
  SERIAL_WIFI.begin(115200);
  pinMode(5,INPUT); //keypad와 연결된 uno의 빨간 LED에서 받아오는 신호
  WiFi.init(&SERIAL_WIFI);
  Blynk.begin(auth, ssid, pass);
}

void loop()
{
  Blynk.run();
  int recieveSignal = digitalRead(5);
  if(recieveSignal == 1){ //신호가 들어오면 알람을 보냄
    Blynk.notify("틀린 비밀번호가 입력되었어요!");
  }
}

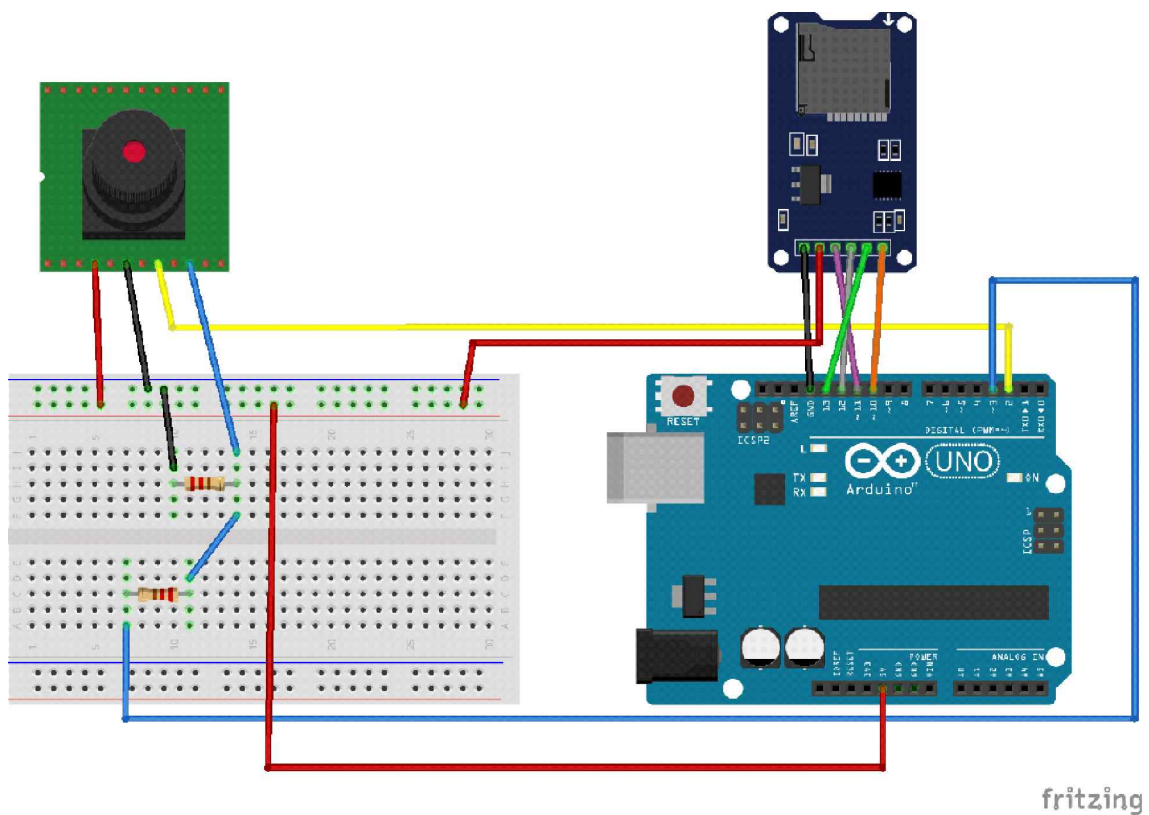
```

### 3) 회로도

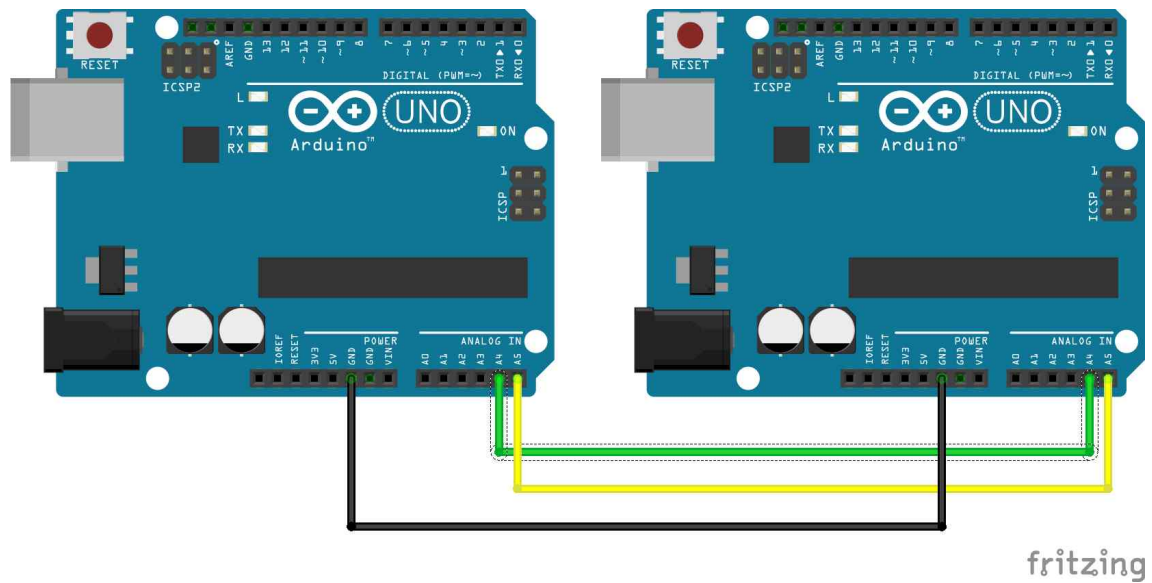
<Keypad module - Arduino Uno 1>



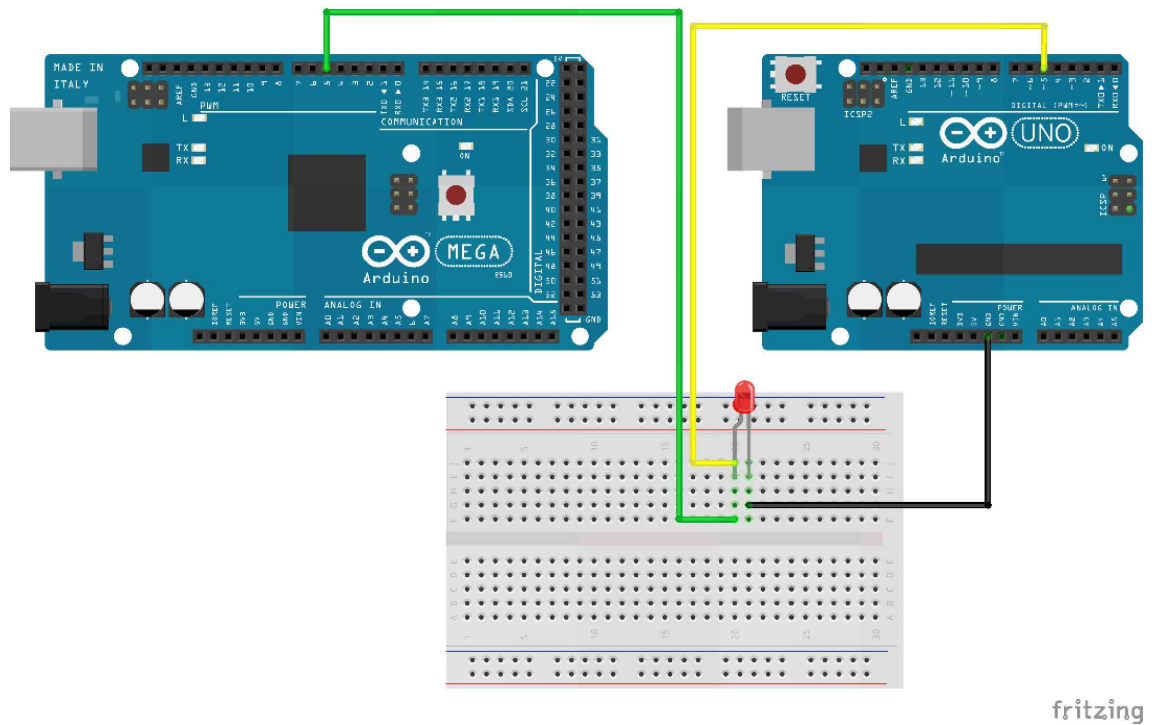
<Camera module - Arduino Uno 2>



<TWI 통신으로 Uno1 과 Uno2를 연결>



<LED 신호로 Uno와 Mega 연결>



#### 4) 작동 과정 및 결과

키패드로 비밀번호를 입력받는다. passcheck 함수를 호출하여 원래 비밀번호와 비교를 한다. 비밀번호가 일치한다면 flag=1이 되고 다음의 동작들을 하게 된다. 서보모터 동작(잠금장치 해제), correctMelody 부저 동작, 초록색 LED 점등, "door open"이라는 Serial.print, 거리 감지센서 동작 시작한다. 만약 비밀번호가 일치하지 않는다면 flag=-1이 되고 다음의 동작들을 하게 된다. closeMelody 부저 동작, 빨간색 LED 점등, "wrong password!"라는 Serial.print, flag=0으로 초기화한다. 그래서 다음 입력을 받을 준비를 한다.

잠금장치가 열리면 거리감지센서가 동작을 시작해 문이 닫힐 때까지 거리를 측정한다. distance<9 임이 두 번 확인되면(문이 완전히 닫힘을 확인하기 위해) 서보모터가 동작하여 잠금장치를 걸고 closeMelody 부저가 동작하고 flag=0으로 초기화를 한다.

도어락의 안쪽에는 스위치 모듈이 있는데 이것이 눌리면 flag=1로 바뀌면서 다음의 동작을 한다. 서보모터 동작(잠금장치 해제), correctMelody 부저 동작, 초록색 LED 점등, "door open"이라는 Serial.print, 거리감지센서 동작 시작한다.

LED 신호로 UNO와 Mega를 연결한다. 아두이노 메가는 와이파이 칩드가 부착된 wizfi310을 사용한다. 신호를 받으면 blynk.notify함수를 통해 미리 등록해 둔 스마트폰으로 푸시알림이 올린다.

#### 5) 문제 해결 과정

- 거리감지센서를 문 지방에 딱 붙도록 부착하였더니 제대로 거리를 감지하지 못해 문이 닫혔다는 것을 인식하지 못했다. 거리감지센서가 감지하는 거리를 시리얼모니터로 측정해보면서 일정 거리이상 가까워지면 실제 거리보다 훨씬 길게 측정된다는 것을 알게 되었고 문과 어느 정도 거리를 둔 곳에 거리감지센서를 부착하는 것으로 문제를 해결하였다.

- TWI 통신에서의 리시브이벤트 함수 안에 사진을 찍기 위한 코드를 넣었더니 사진이 찍혔다는 출력문 나왔지만 저장되는 사진의 바이트가 0 이었다. flag 변수를 이용해서 리시브이벤트 함수가 일어났을때만 loop문에서 사진이 찍히도록 하여 문제를 해결했다.

- 카메라모듈을 처음 받았을 때 도선이 전혀 연결되어있지 않은 상태였기 때문에 직접 납땜을 해서 도선을 연결하였다.

- 가장 문제였던 것은 카메라모듈의 연결문제였다. 분명히 작동하던 카메라가 갑자기 작동하지 않아서 카메라 모듈에 전류가 흐르는지 LED로 확인했고 납땜을 3번이나 다시 해봤다. 길게 연결된 점프선을 전부 바꿔보는 것도 2번이나 했다. 점프선을 바꿨을 때 제대로 동작하는 것을 확인했고 처음에는 전선들을 정리하다가 너무 많이 움직여서 안쪽에서 끊어진 것이 아닌가, 카메라 모듈이 점프선에 과부하를 주는 것이 아닌가하고 의심해봤지만 결론은 새로 업로드할때마다 카메라의 선들을 다시 연결해주어야 하는 것임을 알게되었다.

- Blynk를 이용해서 틀린 비밀번호가 입력되었을 때 핸드폰으로 알림을 받게 하기 위해서 keypad모듈과 연결되어있던 우노보드를 wizfi310과 연결된 메가 보드로 변경하였으나 keypad 라이브러리와 blynk 라이브러리가 알 수 없는 충돌을 일으켰다. 라이브러리 자체를 변경할수 없었으므로 우노보드 두 개와 메가보드 하나를 모두 TWI통신으로 연결하려 하였지만 blynk 코드안의 리시브이벤트함수가 제대로 작동하지 않았기 때문에 실패하였고 고민 끝에 비밀번호가 틀리면 동작하는 빨간 LED가 켜지는 신호를 메가에서는 입력신호로 받아와서 알림을 보내는 방법으로 문제를 해결하였다

## IV. 결론

### 1) 분석 및 고찰

기말 프로젝트를 시작하기 전, 마음이 맞는 사람들과 조를 이루었는데 아두이노 보드를 다루어 보았던 사람이 많아서 좋은 결과물을 산출해낼 자신이 있었다.

하지만 한편으로는 그만큼 대단한 작품을 만들어야 할 것 같아 걱정과 부담감이 느껴졌었다.

아두이노 보드를 이용하여 개발을 해보았던 경험이 있었지만 막상 기말 프로젝트를 시작하고 나니 새롭게 공부하여 알게 된 사실이 많았다.

아두이노 보드간 TWI통신, 카메라 모듈을 이용하여 사진을 찍고 SD카드에 저장하는 방법등을 공부할 수 있었다. 또 각 부품을 제어해서 작품이 원하는 동작을 수행하게끔 논리적으로 프로그래밍을 하는 실력도 키울 수 있었다.

기술적인 부분 말고도 조원들과의 팀워크의 중요성을 알 수 있었다. 3개월 가량을 함께 프로젝트를 진행하면서 조원들과 원활하게 소통하면서 문제를 해결하는 방법과 회의를 통해 각자의 역할을 알맞게 분담하는 방법을 배울 수 있었다.

실력있는 사람들끼리 모여도 조원들간의 소통이 잘 되지 않는다면 팀프로젝트는 진행되지 않을 것이기 때문에 각자의 기술적인 지식보다 팀워크가 더 중요한 것이라고 느꼈다.

프로젝트를 하면서 어려웠던 점이 있다면 카메라 모듈에 연결된 선들은 오랫동안 꽂혀있으면 제 기능을 하지 못하게 되어 수시로 선을 바꾸어 줘야하는 번거로움이 있었다.

작품을 완성하고 나서 작품이 제대로 동작한다는 것 보다는 조원들과 많은 시행착오를 함께 해결해가며 완성했다는 사실에 더 성취감을 느낄 수 있었다.

과제물을 제출하기 위해 만났지만 서로 많은 것을 배울 수 있어서 좋은 시간들이었던 것 같다.

### 2) 향후 계획

현재 만들어진 도어락 시스템은 사진을 찍으면 블랙박스처럼 나중에 일이 발생한 후 그 사진을 컴퓨터에 직접 연결하여 확인해야 한다는 한계가 있다. 이 점을 보완하여 한 가지 방법을 고안하게 되었다. 크게 다른 점은 사진을 서버로 전송하는 것의 유무이다.

새롭게 생각한 방법은 찍힌 사진을 서버로 보내고 현재 만들어진 도어락 시스템이 메시지를 보낼 때 전송된 사진이 있는 URL을 함께 보내는 것이다. 메시지를 받은 사람은 원한다면 URL을 클릭함으로써 사진을 확인할 수 있게 된다.

이 방법으로 누가 현재 집에 들어가려고 시도하는지 그 자리에서 바로 알 수 있게 되고, 만약 그게 범죄자라면 그 사람을 제 시간에 잡을 수 있게 될 것이다. 여기서 조금 더 생각해 보면, 사람들이 잘 때는 메시지가 와도 알기가 어렵기 때문에, 경보음을 울려서 알 수 있도록 할 수 있을 것이다. 일단 거리 감지 센서를 두 개 더 부착하여 몇 명이 들어오고 나갔는지 알 수 있게 한다. 그런 다음 그 날 들어오는 인원수를 설정하고, 모두 집에 들어오게 되면, 그 다음 비밀번호를 입력하는 순간부터는 경보음이 울리도록 하는 것이다.

후에 스마트 도어락 시스템을 보완하여 시중에 내놓아도 부족하지 않은 퀄리티를 가진 도어락을 완성시켜 보고자한다.

## V. 참고자료

<http://blog.naver.com/woals719/220934628518> - 아두이노 도어락 오픈  
<https://kocoafab.cc/fboard/590> - 아두이노 서버, 데이터 전송(사진)  
<http://cafe.naver.com/arduinoistory/38525> - 아두이노 사진 서버 전송  
<http://blog.naver.com/roboholic84/220483456256> - 아두이노와 카메라 모듈로 사진 촬영  
<http://blog.naver.com/yuyyulee/220548025758> - 와이파이 통신 - webserver  
<http://wiznetian.com/article/wizarduino-ubidots-%ed%81%b4%eb%9d%bc%ec%9a%b0%eb%93%9c-%ec%97%b0%eb%8f%99/> - 클라우드 서비스(Ubidots)  
<http://blog.daum.net/rockjy99> - 아두이노 통신  
<http://wiznetian.com/article/wizfi310-%ec%95%84%eb%91%90%ec%9d%b4%eb%85%b8-%eb%9d%bc%ec%9d%b4%eb%b8%8c%eb%9f%ac%eb%a6%ac%ec%97%90%ec%84%9c-%ec%a7%80%ec%9b%90%ed%95%98%ec%a7%80-%ec%95%8a%eb%8a%94-wizfi310-%eb%b8%b0%eb%8a%a5/> - 아두이노 쉘드 정보  
<http://cafe.naver.com/arduinoistory/32049> - 아두이노 통신  
<http://blog.naver.com/PostView.nhn?blogId=opusk&logNo=220986288719&redirect=Dlog&widgetTypeCall=true> - blynk 라이브러리 포팅  
<https://www.youtube.com/watch?v=12CUhNIBXGc> - 아두이노 카메라 프로토 타입  
<http://blog.naver.com/PostView.nhn?blogId=roboholic84&logNo=220821919602> - 아두이노로 ip카메라 만들기  
<http://blog.naver.com/roboholic84/220483456256> - 카메라 모듈 이용  
이 외 여러 블로그, 카페에서 회로도 참고.