

Arduino 기반 침입자 감지 시스템
설계 및 구현 보고서
(SAW)

2015 년 11 월

배 재 대 학 교
컴 퓨 터 공 학 과

목 차

1. 서론

1. 목적
2. Arduino
3. 사물인터넷

2. 본론

1. 구상도
2. 일정, 부품
3. 알고리즘
4. 회로도
5. 코드
6. 실행 화면
7. 결과물

3. 결론

참고문헌

부품 목록

1. 서론

1.1 목적

최근 인터넷, 무선인터넷 보급과 웨어러블 스마트 기기의 시장 성장으로 사용자에게 보다 많은 서비스를 제공한다. 이에 맞춰 IOT 기술의 발전은 동반 성장하게 되었다.

국내의 IOT 현황은 밝다고 할 수 있다. 방송통신위원회에서 2009년 10월 ‘사물 지능 통신기반 구축 기본계획’ 발표와 2011년 10월 7대 스마트 신산업 육성 전략에 사물인터넷이 포함되어 있다. 미래창조과학부에서도 2014년 5월 ‘사물 인터넷 기본 계획’을 시작으로 사물 인터넷 혁신센터 개소, 사물 인터넷 글로벌 협의체를 출범 했다.

이와 같은 시대의 흐름을 맞춰가기 위해 2학년 중 IOT와 Arduino에 관심이 많은 동기들과 학술제에서 Arduino를 기반으로 IOT를 구현하기로 하였다.

1.2 Arduino

Arduino는 오픈소스를 기반으로 한 단일 보드 마이크로 컨트롤러이다. Arduino는 다수의 스위치나 센서로부터 값을 받아드려 외부 전자 장치들을 통제함으로써 환경과 상호작용이 가능한 물건으로 만들 수 있다. 다양한 OS를 지원하고 다른 마이크로 컨트롤러와 다르게 컴파일된 펌웨어를 USB를 통해 쉽게 업로드 할 수 있다는 큰 장점이 있다. 통합 개발 환경 (IDE)를 제공하며 소프트웨어 개발과 실행코드 업로드도 제공한다. 개발 언어는 c++이다. Arduino가 인기를 끌면서 이를 비즈니스에 활용하는 기업도 늘고 있다. 레고는 자사의 장난감과 Arduino를 활용한 로봇교육 프로그램을 학생과 성인을 대상으로 북미 지역에서 운영하고 있다.

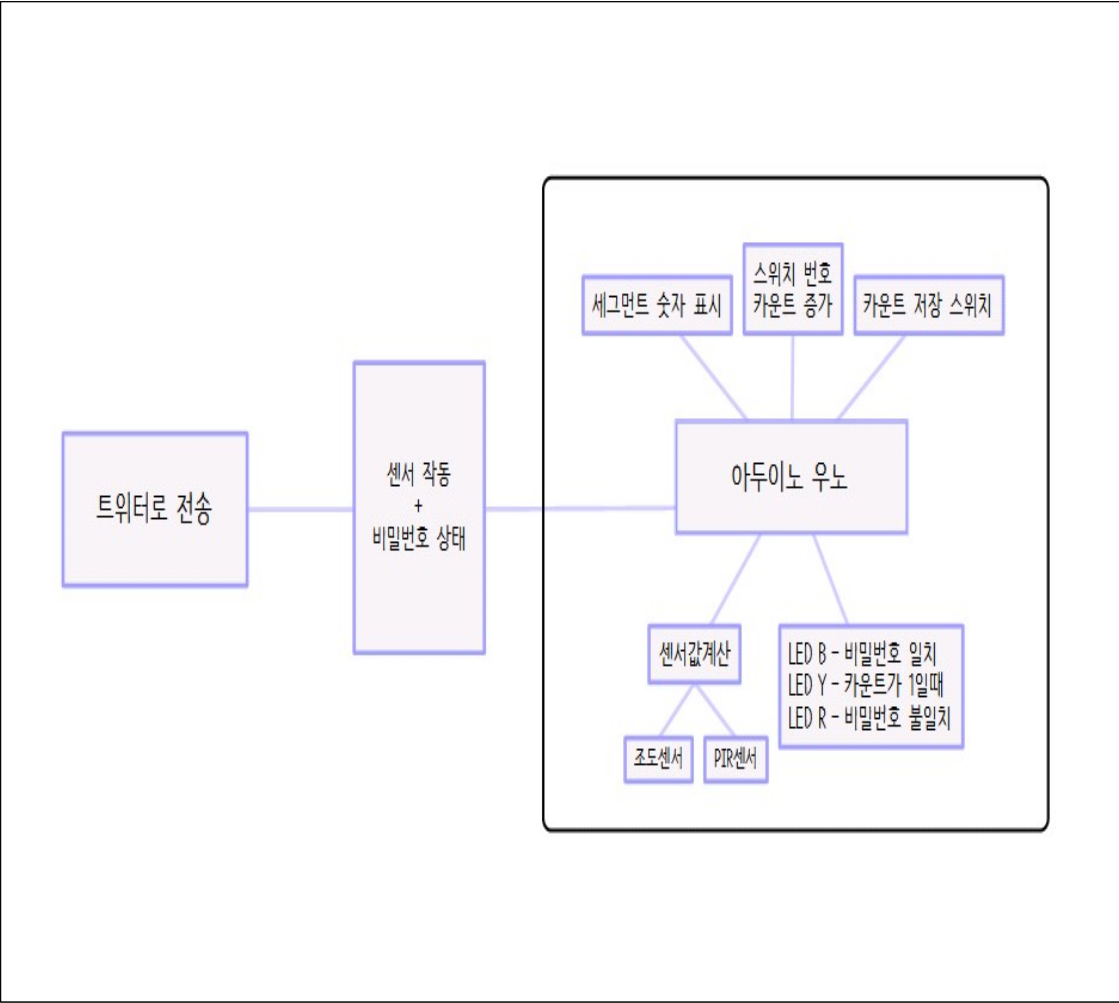
1.3 사물 인터넷

사물 인터넷(Internet of Things, IOT)은 각종 사물에 센서와 통신기능을 내장하여 인터넷에 연결하는 기술을 의미 한다. 여기서의 사물이란 가전제품, 모바일 장비, 웨어러블 컴퓨터 등 다양한 임베디드 시스템이 된다. 정보 기술 연구 가트너에 따르면 2009년까지 사물 인터넷을 사용하는 사물은 9억 개였으나 2020년까지 260억 개에 이를 것으로 예상된다. 또한 시스코 시스템즈 조사에 따르면 2013년부터 2022년까지 10년간 사물 인터넷이 14조 4천 달러의 경제적 가치를 창출할 것으로 기대한다.

사물 인터넷을 구축하기 위해서는 기술적인 환경 구축이 선행되어야 한다. 이는 크게 사물 신원 확인, 네트워크 구축, 사물에 감각 부여, 컨트롤 가능성으로 나눌 수 있다. 사물의 신원 확인은 각각의 객체는 다른 객체를 식별할 수 있게 해주는 신원이 필요하다. 이를 위해 IPv6체계가 필요하다. 네트워크 구축은 사물이 취합한 정보를 필요에 따라 다른 사물과 교환, 통합함으로써 새로운 정보를 만들 수 있어야 한다. 감각 부여는 사물에 미각, 후각, 촉각, 시각 등을 부여해 주변 환경의 변화를 측정 할 수 있도록 한다. 컨트롤 가능성은 임의적인 조작을 통해 사용자는 사물에게 행동을 지시 할 수 있어야 한다.

2. 본론

2.1 구상도



< 그림 1. 초기 구상도 >

Arduino와 IOT를 접목시키기 위해 구상 한 것은 침입자 감지 시스템이다. Arduino를 통해 비밀번호를 입력 받고 잠금 상태가 되었을 때 센서를 통해 동작이 인식 되면 트위터로 알람이 울린다.

1. 지정된 비밀번호 두 자리를 알맞게 입력한다.
2. 비밀번호가 맞으면 평소 상태로 집 안에서 움직여도 알람이 울리지 않는다.
3. 집 안에서 나가는 걸 가정, 비밀번호를 걸어 보안 상태로 만든다.
4. 보안 상태일시 집 안에 있는 PIR센서(동작인식센서)와 조도센서의 값을 종합해 침입자가 있는지 판단한다.
5. 있다고 판단될 시 지정된 트위터로 트윗을 작성한다.
6. 지정된 비밀번호를 다시 알맞게 입력하면 보안 상태가 해지되어 트위터가 더 이상 올라오지 않는다.
7. 2번 과정으로 돌아간다.

< 표 1. 초기 구상 실행순서 >

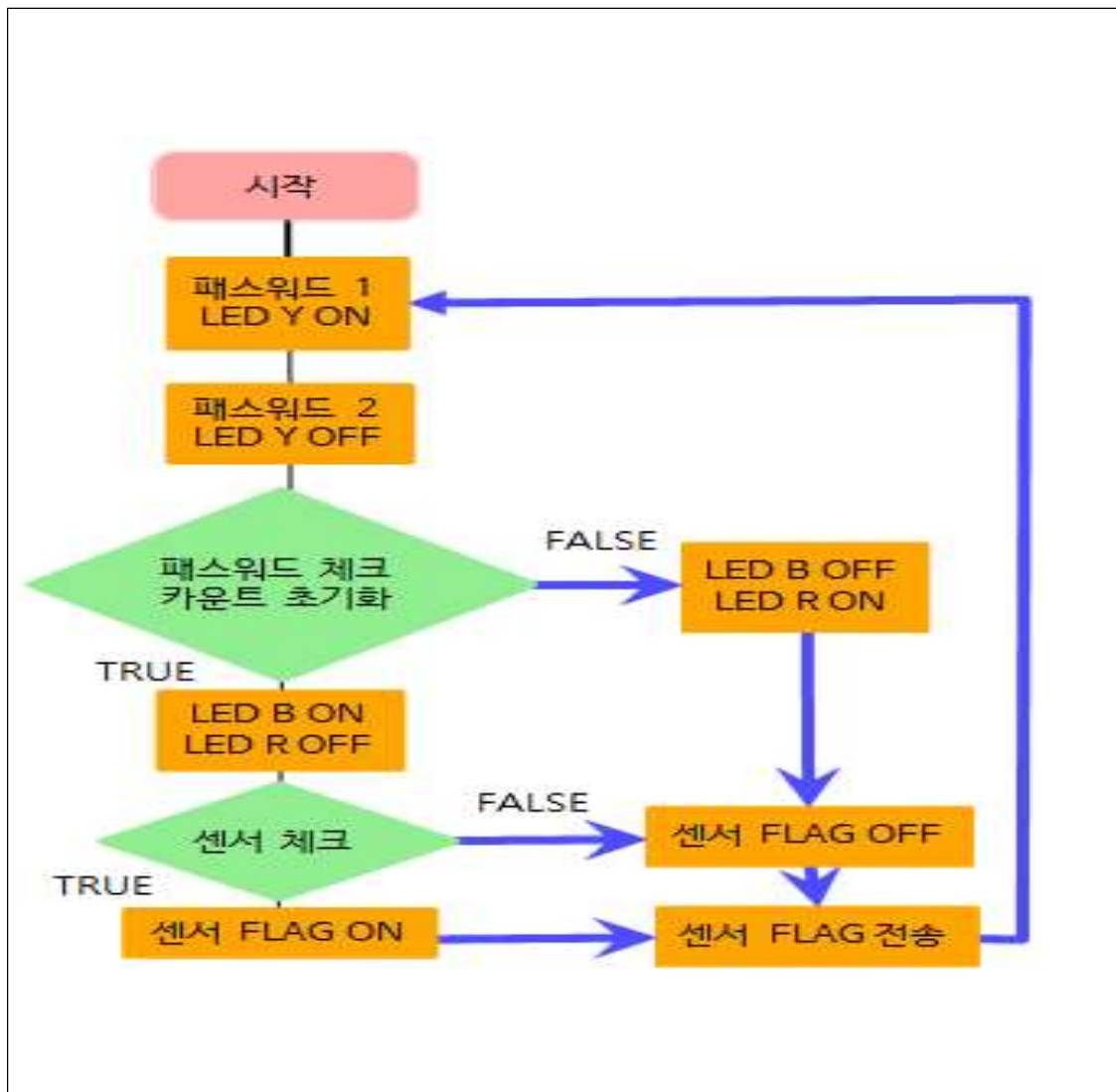
2.2 일정

주요 업무	일정 (주 단위)									
	1	2	3	4	5	6	7	8	9	10
주제 정하기										
구상도 제작										
Arduino 기초 공부										
역할분담										
부품 선정										
Arduino 심화학습(센서)										
Arduino 심화 학습(세그먼트)										
알고리즘 작성										
Arduino 심화 학습(트위터)										
회로도 작성										
Arduino 코딩										
집 모형 만들기										
점검										

< 표 2. 일정 간트 차트 >

일정은 위의 간트 차트를 통해 주 단위로 시간을 분배하였고, 필요한 작업을 분배하였다.

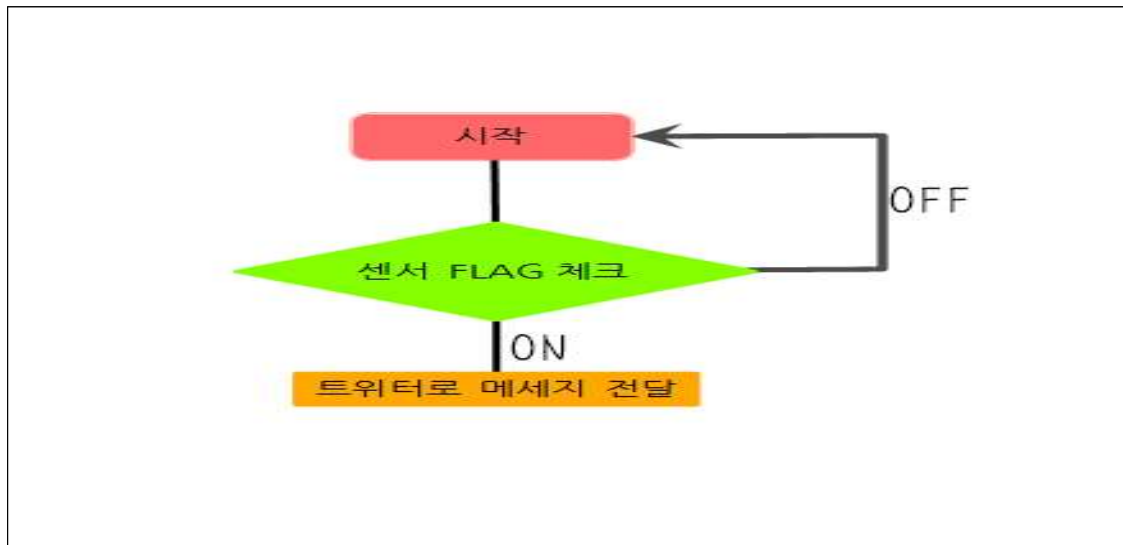
2.3 알고리즘



< 그림 2. 알고리즘 순서도 (1) >

그림 2 의 알고리즘은 비밀번호 체크와 세그먼트 숫자 카운트 그리고 센서 플래그 전송까지의 알고리즘이다.

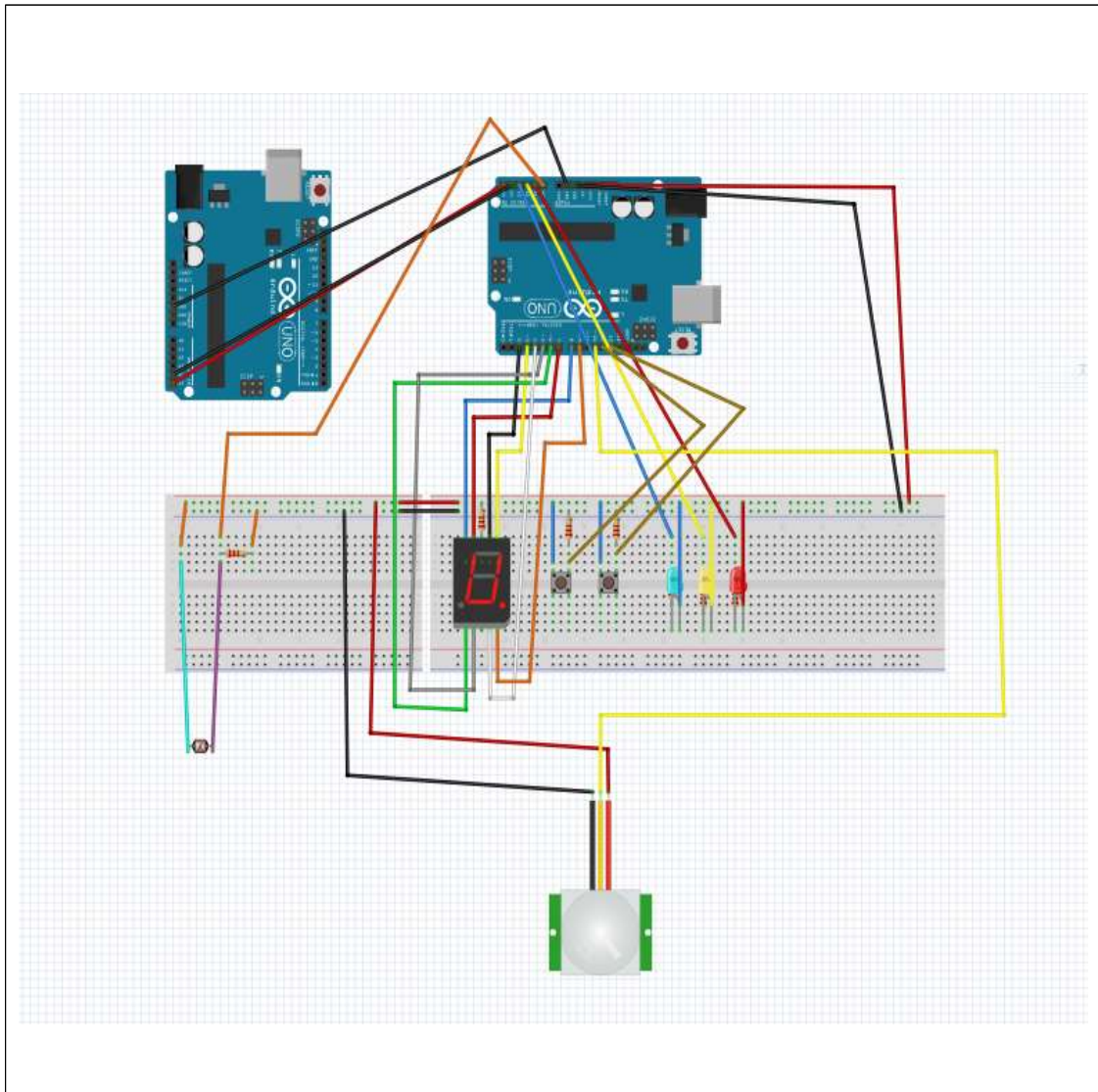
총 두 자리인 비밀번호 입력 중 첫 자리를 입력했을 때 LED Y가 켜지고 나머지 비밀번호 둘째 자리를 입력하면 LED Y가 꺼지면서 비밀번호가 맞는지 검사한다. 만약 비밀번호가 맞으면 LED B가 들어오면서 센서 플래그가 내려간다. 하지만 비밀번호가 맞지 않으면 LED R이 켜지면서 센서 플래그를 올린다. 현 상태의 센서 플래그를 전송한 뒤 비밀번호 입력 순서로 되돌아간다.



< 그림 3. 알고리즘 순서도 (2) >

그림 3 은 센서 플래그를 송신 받은 뒤 트위터로 전송할지 안할지의 알고리즘이다. 그림 2에서 센서 플래그를 송신 받고 플래그가 내려가 있는 상태로 넘어 왔을 경우, 트위터로 전송하지 않고 대기상태로 되돌아간다. 센서 플래그가 올라와 있는 상태로 넘어 왔을 경우는 트위터로 전송한다.

2.4 회로도



< 그림 4. 회로도 >

그림 4 는 위에 나온 구상도와 알고리즘을 회로도로 구현한 것이다. 처음 회로도를 구현 했을 때는 Arduino 하나로 구현 했지만, 와이파이 쉴드의 전력소모로 인해 세그먼트가 작동하지 않는 상태가 발생 하였다. 이를 해결 하기 위해 Arduino 두 개를 연동시키는 방법을 사용, 아날로그 I/O핀 A4, A5를 연결해 두 Arduino를 연동 시켰다. 연동시킨 두 Arduino를 각각 다른 일을 처리하고 필요한 값을 통신 하도록 설계 했고, 한쪽 Arduino에서 비밀번호와 센서 플래그를 종합적으로 처리하면 와이파이 쉴드가 부착된 Arduino로 값을 전달, 값을 판단해 트위터로 전송하도록 설계 하였다.

위의 Arduino중 센서와 세그먼트를 출력하는 Arduino를 sensor_arduino라고 하자 sensor_arduino의 아날로그 I/O핀 A0는 조도센서 값을 입력받는다. 나머지 A1, A2, A3의 아날로그 I/O핀으로 LED R, Y, B를 출력하고 A4, A5의 핀으로 다른 Arduino와 통신한다.

sensor_arduino의 디지털 I/O핀중 2번 핀부터 9번 핀 까지는 세그먼트의 출력을 담당하고, 12, 13번 핀으로 세그먼트의 숫자변경, 숫자의 저장하는 스위치의 입력 값을 받는다. 11번 핀으로는 PIR 센서의 값을 입력받는다.

나머지 하나의 Arduino는 와이파이 칩드를 통해 트위터에 전송함으로 twitter_arduino라고 명명했다. twitter_arduino의 아날로그 I/O 핀 A4, A5는 sensor_arduino와 통신하기 위한 핀이다.

2.5 코드

```
#include<Wire.h>                                //Arduino간 통신을 위한 헤더파일

const int sensor = A0;                          // 조도센서를 입력받는 A0핀
const int PIRSensor = 11;                       // PIR센서를 입력받는 11핀

const int ledR = A1;                            //LED R 출력 A1핀
const int ledY = A2;                            //LED Y 출력 A2핀
const int ledB = A3;                            //LED B 출력 A3핀

const int changeButton = 12;                    //세그먼트 숫자를 변경하는 스위치 입력 12핀
const int selectButton = 13;                   //출력된 숫자를 저장하는 스위치 입력 13핀

int selectNum[2] = {};                          //비밀번호 입력수를 저장하는 배열변수
int selectCount = 0;                            //몇번을 입력했는지 저장하는 카운트변수

int password[2] = {1,2};                       //저장되어 있는 비밀번호 1 2
bool unlocked = false;                         //비밀번호의 참 거짓을 판단하는 변수

int segPins[]={2,3,4,5,6,7,8,9};               //세그먼트 출력 2~9핀
byte digits[10][8]={                           //세그먼트 출력을 위한 다차원 배열
  {0,0,0,0,0,0,1,1},                          //세그먼트 제품의 특이점으로
  {1,0,0,1,1,1,1,1},                          //0 = HIGH이고 1 = LOW이다
  {0,0,1,0,0,1,0,1},
  {0,0,0,0,1,1,0,1},
  {1,0,0,1,1,0,0,1},
```

```

{0,1,0,0,1,0,0,1},
{0,1,0,0,0,0,0,1},
{0,0,0,1,1,0,1,1},
{0,0,0,0,0,0,0,1},
{0,0,0,0,1,0,0,1}
};

int countNum=0;                                //숫자 카운트를 저장하는 변수

int sensorVal=0;                                //센서의 값을 저장하는 변수

int benchmarkLight = 20;                        //조도센서의 변경값중 기준이 되는 수

void setup() {
  Serial.begin(9600);                            //시리얼 통신 시작
  Wire.begin();                                  //Arduino 간 통신 시작
  pinMode(changeButton, INPUT);                  //
  pinMode(selectButton, INPUT);                  //
  pinMode(ledR, OUTPUT);                         //      각 핀 번호의 input, output 선언
  pinMode(ledY, OUTPUT);                         //
  pinMode(ledB, OUTPUT);                         //
  pinMode(PIRSensor, INPUT);                     //
  for (int i = 0; i < 8; i++) pinMode(segPins[i], OUTPUT); //
}

void loop() {
  if(digitalRead(changeButton) == HIGH && digitalRead(selectButton) == LOW){
    countNum++;
    countNum %= 10;
    delay(250);
  }
  // 만약 changeButton가 들어왔을 때 countNum증가 표현은0~9까지이기 때문에 %10을
  // 추가함 빠르게 변하는 것을 방지하기 위해 delay를 줌
  else if(digitalRead(changeButton) == LOW && digitalRead(selectButton) == HIGH){
    if (selectCount<2){
      selectNum[selectCount] = countNum;
      selectCount++;
      countNum = 0;
      delay(400);
    }
  }
}

```

```

    }
    //만약 번호를 선택 하였을 때 비밀번호 2자리가 넘지 않게 입력되었을 경우만 생각, 카
    //운트된 수를 selectNum[selectCount] 에 저장하고 인덱스 증가
    for (int i = 0; i < 8; i++)    digitalWrite(segPins[i], digits[countNum][i]);
    //세그먼트 출력
    if (selectCount == 1)    digitalWrite(ledY, HIGH);
    //입력이 하나 들어올 경우 LED Y 켜
    else    digitalWrite(ledY, LOW);
    //아닐 경우 LED Y 끄
    if (selectCount == 2){        //비밀번호가 2자리 입력되었을 경우
        if ((selectNum[0] == password[0]) && (selectNum[1] == password[1]))
        unlocked = true; //만약 비밀번호가 맞게 입력되었을 경우 unlocked을 참으로 변경
        else unlocked = false;
    }
    //아닐 경우 거짓
    selectCount = 0;
}

sensorVal = analogRead(sensor);        //조도센서의 값을 받아 저장함
Serial.println(sensorVal);              //조도센서의 값을 시리얼 모니터에 출력

if (unlocked){                          //비밀번호가 풀렸을 경우
    digitalWrite(ledB, HIGH);
    digitalWrite(ledR, LOW);
} else{                                  //비밀번호가 걸렸을 경우
    digitalWrite(ledB, LOW);
    digitalWrite(ledR, HIGH);
}

Serial.println(sensorVal);              //조도센서의 값을 시리얼 모니터에 출력
if (!unlocked && (benchmarkLight > sensorVal) && digitalRead(PIRSensor) ==
HIGH){ //비밀번호가 걸렸으면서 조도센서 값이 기준치보다 작으면서 PIR 센서값이 들어
왔을때
    Wire.beginTransmission(4);          //4번 핀으로 통신
    Wire.write(1);                      //1을 보내줌
    Wire.endTransmission();
} else{
    Wire.beginTransmission(4);
    Wire.write(0);
    Wire.endTransmission();
}
}

```

< 표 3. sensor_arduino 소스 >

```

#include <Wire.h>           //Arduino간 통신을 위한 헤더파일
#include <Twitter.h>        //트위터를 처리하기 위한 헤더파일
#include <SPI.h>             //
#include <WiFi.h>           //와이파이 설드를 처리하기 위한 헤더파일

int locked = 0;

char ssid[] = "ollehEgg_419"; //연결할 무선 네트워크 이름을 입력해주세요.
char pass[] = "info10019";    //비밀번호를 입력해주세요.

Twitter twitter("3636103272-L5urY3iQPasf3vBVU9oE8Tvpypa7CnU8xnKeHzY");

char msg[] = "For someone came to my house ! WTF!!";
int num =0;
void setup() {
    Serial.begin(9600);
    Wire.begin(4);
    Wire.onReceive(receiveEvent);
/*시작*/

    delay(1000);
    WiFi.begin(ssid, pass);

    Serial.begin(9600);

    /*종료*/
}

void loop() {

    if(locked==1){
        delay(500);
        char tmp[10];
        sprintf(tmp, " %d",num);
        char str[100];
        strcpy(str, msg);
        strcat(str, tmp);
    }
}

```

```

Serial.println("connecting ..."); //와이파이 연결후 메세지 보내기 시작 전에 확인

if (twitter.post(str)) {          //메세지를 전송 확인

    int status = twitter.wait(&Serial); //메세지 보내기가 끝난 후 트위터로부터 상
태전달 받음
    if (status == 200) {           //200을 전달받으면
        Serial.println("OK.");     //성공된 것이므로 시리얼 모니터에 OK출력
    } else {                       //아닐 경우 실패 메세지와 전달받은 값 출력
        Serial.print("failed : code ");
        Serial.println(status);
    }
} else {                           //연결부터 실패 했을 경우
    Serial.println("connection failed.");
}
locked=0;
num++;
delay(10000);
}
}

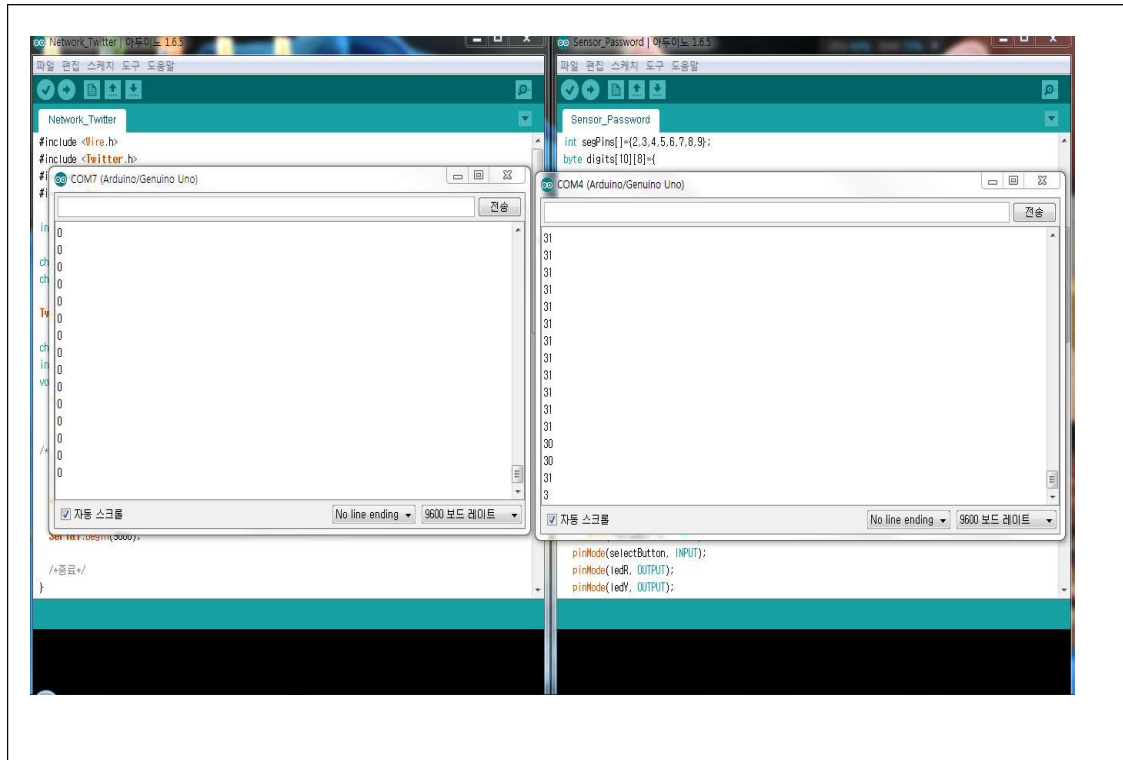
void receiveEvent(int howMany)
{
    locked = Wire.read();
    Serial.println(locked);
}

```

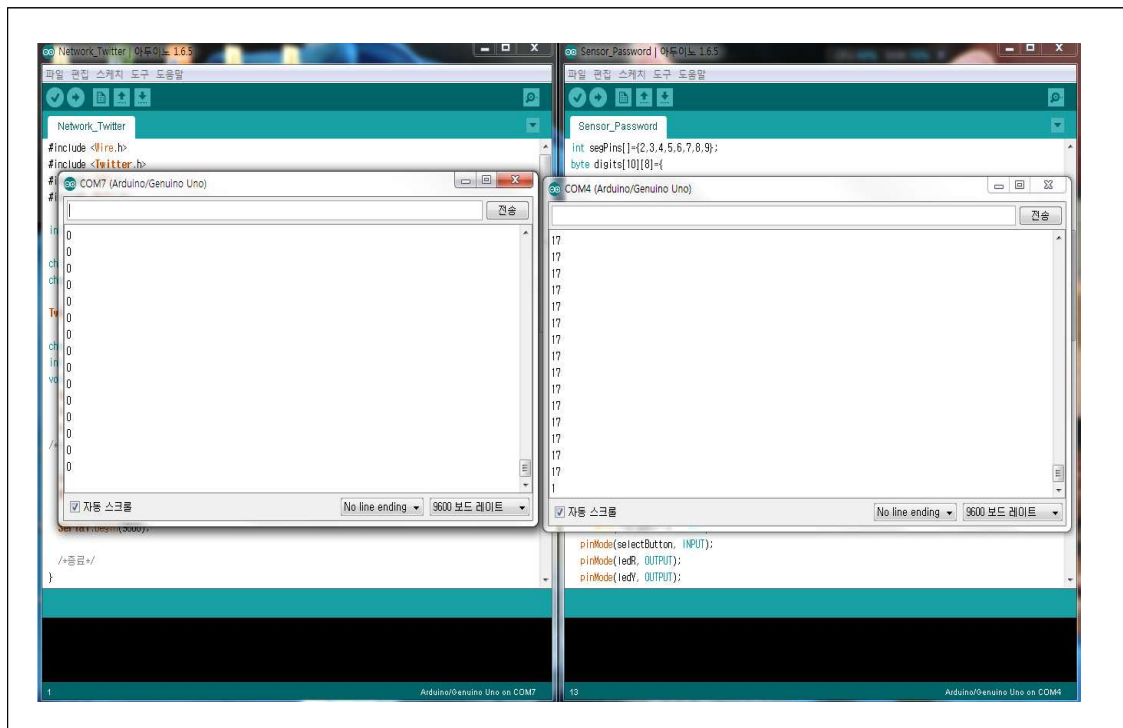
< 표 4. twitter_arduino 소스 >

코드에 대한 설명은 주석으로 대체함

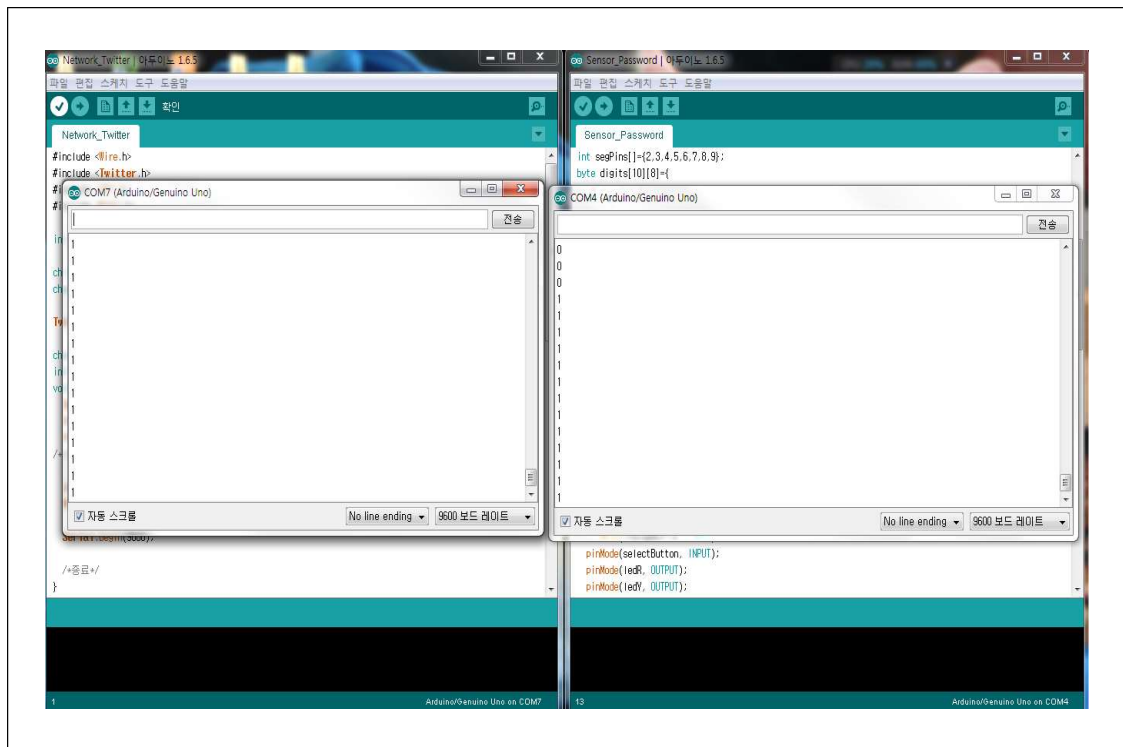
2.6 실행 화면



< 그림 5. 평소상태의 센서값과 판단하는 bool값 >

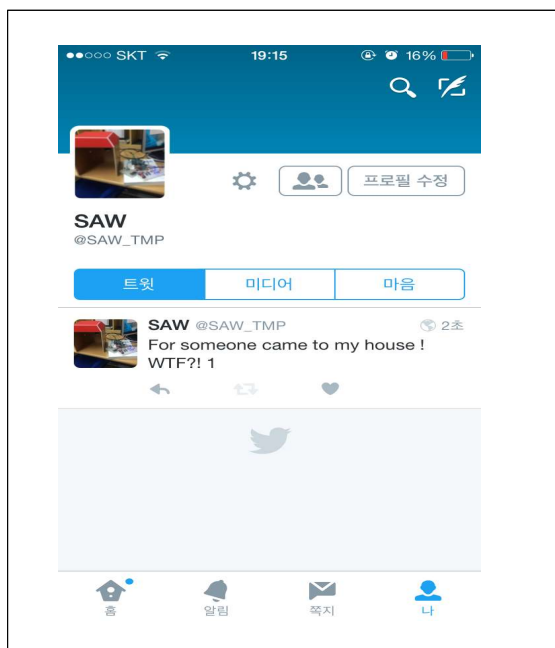


< 그림 6. 평소상태의 센서 변화값과 bool값 >

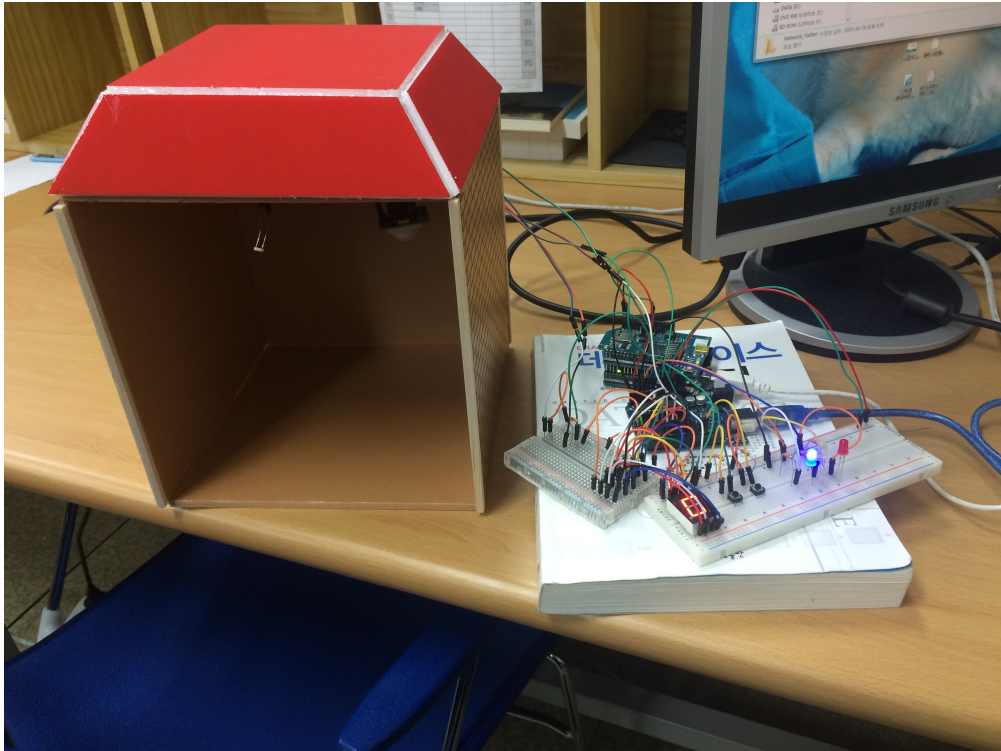


< 그림 7. 보안상태의 센서값 변동과 bool값의 변화 >

2.7 결과물



< 그림 8. 트위터에 올라온 알람 >



< 그림 9. 최종 완성본 >

3. 결론

처음 하는 팀 프로젝트로 개발 일정과 주제 선정보다 배우지 않은 Arduino의 개발에 어려움이 있었다. 하지만 Arduino 공부의 방향과 Arduino의 공식 커뮤니티를 통한 제공 라이브러리의 활용에 초점을 두어 프로젝트 개발에 차질 없이 진행 될 수 있었다. 팀원들 간의 의견 조율과 역할의 분담이 적절하게 이루어져 개발 진행 중 트러블이 생기지 않았다.

그 결과 2015년도 컴퓨터공학과 학술제 최우수상을 받을 수 있었다