



# 고급네트워킹 과제2

담당교수 / 최정열 교수님  
이름(학번) / 박원용(20160919)  
정은지(20170864)

# - 목차 -

1. 프로젝트 개요
  - 1.1 주제 선정 배경
  - 1.2 프로젝트 목표
  - 1.3 유사 소프트웨어 분석
2. 시스템 구성도
  - 2.1 시스템 구성도
  - 2.2 아두이노 회로도
  - 2.3 흐름도
3. 앱의 주요 기능 및 화면
  - 3.1 화면 구성
  - 3.2 주요 기능
    - 3.2.1 CCTV 확인 기능
    - 3.2.2 CCTV 켜기/끄기 기능
    - 3.2.3 신고하기 기능
    - 3.2.4 알림 켜기/끄기 기능
    - 3.2.5 움직임 감지 알림 기능
4. 코딩 보고서
  - 4.1 안드로이드
    - 4.1.1 MainActivity.java
    - 4.1.2 activity\_main.xml
  - 4.2 아두이노
    - 4.2.1 cam\_IOT.ino
  - 4.3 라즈베리파이
    - 4.3.1 Motion 실행
    - 4.3.2 Mosquitto 실행
5. 참고자료

## - 그림목차 -

- [그림 1] LG U+ 펫케어 맘카
- [그림 2] ADT캡스 뷰가드미니 2.0
- [그림 3] 시스템 구성도
- [그림 4] 아두이노 회로도
- [그림 5] 흐름도 1
- [그림 6] 흐름도 2
- [그림 7] 앱의 메인화면 구성
- [그림 8] 신고 버튼 클릭 화면
- [그림 9] 움직임 알림 화면
- [그림 10] Motion 실행 화면
- [그림 11] Motion 스트리밍 서버 화면
- [그림 12] Mosquitto 실행 화면

## - 표목차 -

- [표 1] 유사 소프트웨어
- [표 2] MainActivity.java
- [표 3] activity\_main.xml
- [표 4] cam\_IOT.ino

# 1. 프로젝트 개요

## 1.1. 주제 선정 배경



1인 가구가 증가함에 따라 개인 CCTV의 설치가 늘어나고 있습니다. 이에 따라 CCTV를 활용한 홈 IOT 서비스가 활발하게 시행되고 있는데, 이러한 IOT CCTV에 인체 감지센서를 활용하여 보안에 중점을 둔 프로젝트를 구상하게 되었습니다.

## 1.2. 프로젝트 목표

최근 여성을 대상으로 한 성범죄나 스토킹 등의 문제가 나날이 늘어나고 있습니다. 또한, 보안이 취약한 구식 빌라에서의 도난사고도 꽤 빈번하게 일어나고 있기에 간편하게 집에 설치할 수 있는 홈 CCTV가 각광받고 있는 추세입니다.

이러한 트렌드에 맞춰 인체 감지센서와 카메라를 활용한 방법용 CAM IOT를 구현해 보는 것이 이번 프로젝트의 목표입니다.

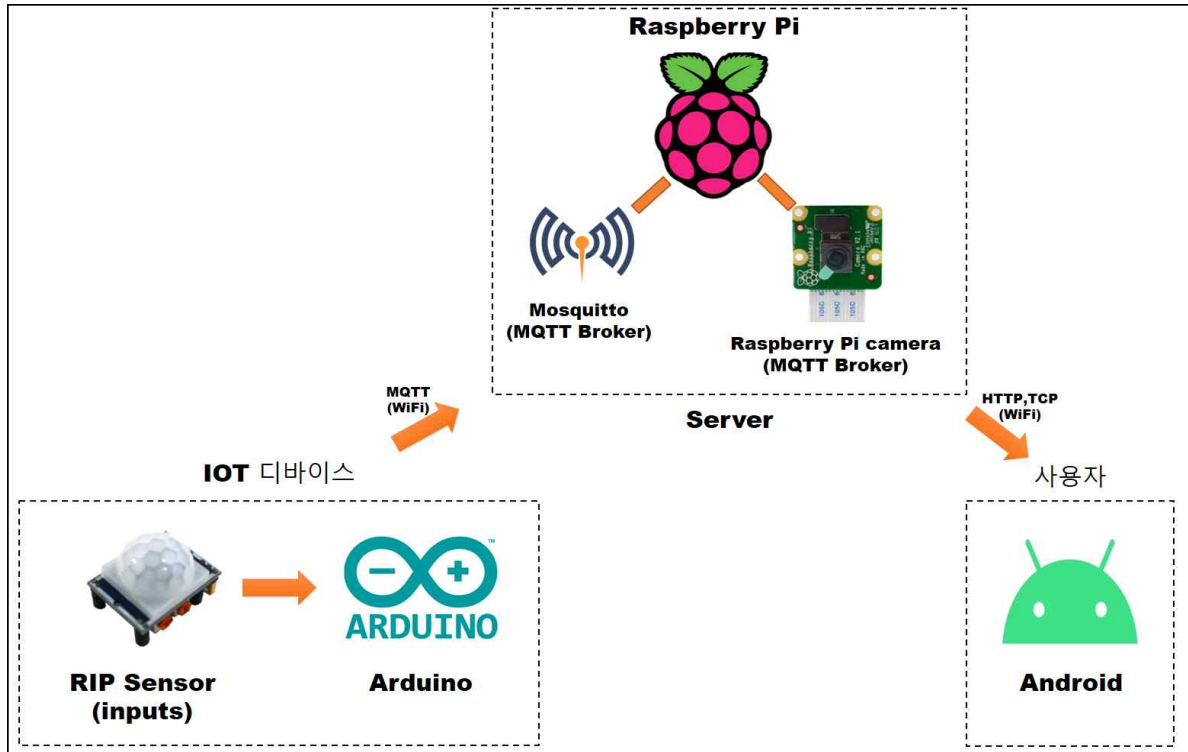
## 1.3. 유사 소프트웨어 분석

LG U+ 펫케어	ADT캡스 뷰가드미니 2.0
 <p><b>맘카</b> 실시간 촬영영상을 확인하며, 반려동물에게 목소리를 들려줄 수 있어요.</p> <p>[그림 1] LG U+ 펫케어 맘카</p>	<p>주요기능</p> <div>  <p>✓ <b>홈 CCTV 기능</b> 집 안 상황을 실시간으로 확인 가능합니다. 혼자 있는 자녀, 반려견 등 가족들의 모습을 확인할 수 있습니다.</p> </div> <div>  <p>✓ <b>양방향 대화 기능</b> 뷰가드미니를 통해 음성 대화가 가능합니다. 외출 시 집 안의 가족과 대화를 할 수 있어 편리합니다.</p> </div> <div>  <p>✓ <b>모션 카메라</b> 상하좌우 카메라 회전 기능으로 집 안 구석구석 원하는 화각의 모습을 확인할 수 있습니다.</p> </div> <div>  <p>✓ <b>프라이버시 보호</b> 카메라 렌즈 숨김 기능으로 영상송출 및 녹화 중단이 가능합니다.</p> </div> <p>[그림 2] ADT캡스 뷰가드미니 2.0</p>
<ul style="list-style-type: none"> <li>• 외출 중에도 반려동물의 모습을 실시간으로 확인할 수 있다.</li> <li>• 움직임을 감지해 반려동물의 행동패턴을 분석하고, 위험한 곳에 접근하지 않도록 도와준다.</li> </ul>	<ul style="list-style-type: none"> <li>• 보안 전문 회사인 ADT의 보안 서비스에 홈 CCTV 기능이 더해져, 보안에 이상이 있을 경우 보안요원이 출동하는 서비스를 제공한다.</li> <li>• 실제 사용자 사례 :             <ol style="list-style-type: none"> <li>1. 공동주택 현관에 비밀번호를 입력하고 출입해도 외부인 출입이 꽤 많아서 보안이 걱정되었는데 외출, 여행시에도 집 보안에 안심할 수 있는 공동주택 거주자</li> <li>2. 직장에 있는 시간 동안 뷰가드를 통해 아이들과 대화를 나눌 수 있게 된 워킹맘</li> </ol> </li> </ul>

[표 1] 유사 소프트웨어

## 2. 시스템 구성도

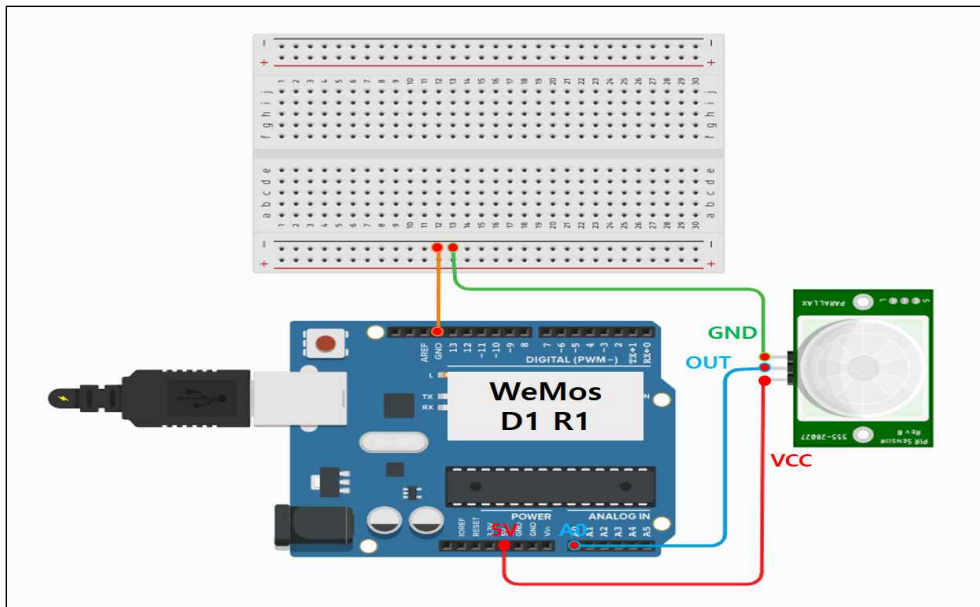
### 2.1 시스템 구성도



[그림 3] 시스템 구성도

- 시스템은 크게 IOT 디바이스, Server, 사용자로 구성되어 있습니다.
- IOT 디바이스의 경우 움직임을 감지하는 RIP 센서와 WiFi 기능이 있는 아두이노인 WeMos D1 R1 보드로 구성되어 있습니다.
- 서버의 경우 라즈베리파이와 MQTT 서버의 역할을 하는 MQTT Broker인 Mosquitto, 화면을 전송해줄 라즈베리파이 카메라로 구성되어 있습니다.
- 사용자는 카메라의 CCTV 화면을 볼 수 있고, 제어가 가능하도록 안드로이드로 구성했습니다.

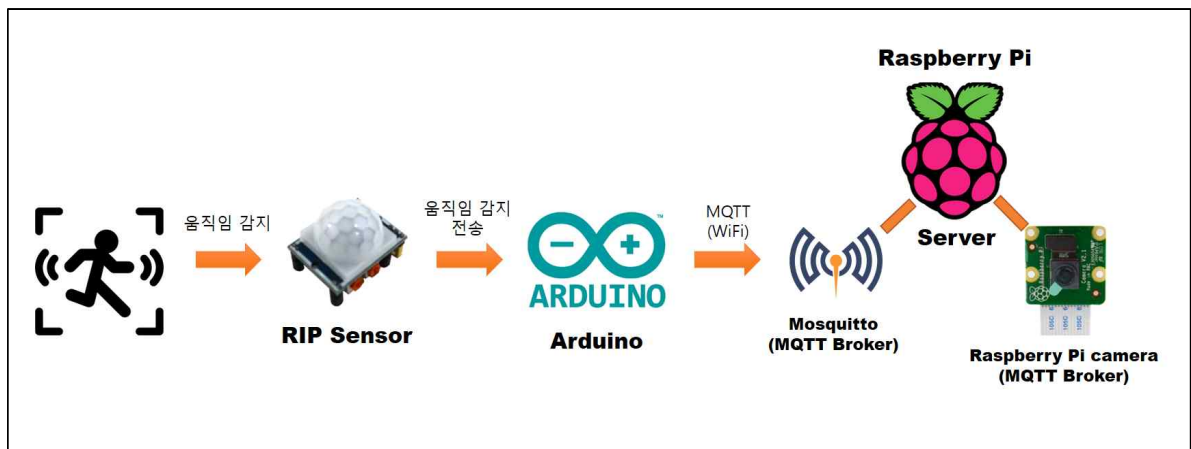
## 2.2 아두이노 회로도



[그림 4] 아두이노 회로도

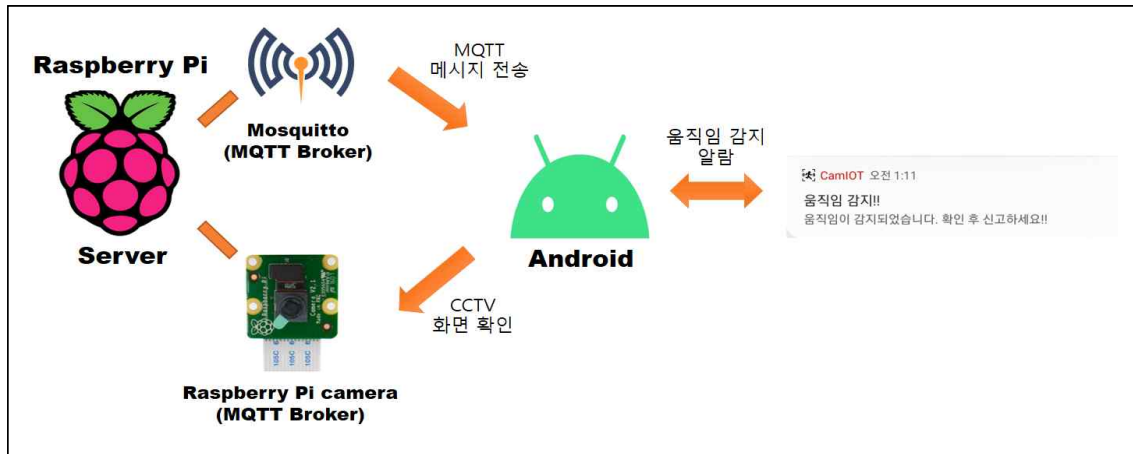
- 아두이노 회로의 구성으로는 WeMos D1 R1과 인체 감지센서, 그리고 보드판으로 이루어져 있습니다.
- WeMos D1 R1의 5V는 인체감지센서의 VCC와 연결하고, WeMos D1 R1의 A0는 인체감지센서의 OUT과 연결하고, 인체감지센서의 GND는 보드판의 13번째 -와 연결하고, 보드판의 14번째 -와 WeMos R1 D1의 GND와 연결합니다.
- 회로를 구성함으로써 인체 감지센서가 움직임을 감지하면, OUT으로 WeMos D1 R1의 A0로 전달해줘 감지가 가능하게 됩니다.

## 2.3 흐름도



[그림 5] 흐름도 1

- [그림 5]과 같이 RIP 센서에서 움직임이 감지되면 아두이노로 움직임 감지 신호를 보내고, 아두이노는 MQTT 프로토콜로 MQTT Broker에게 토픽을 발행합니다.

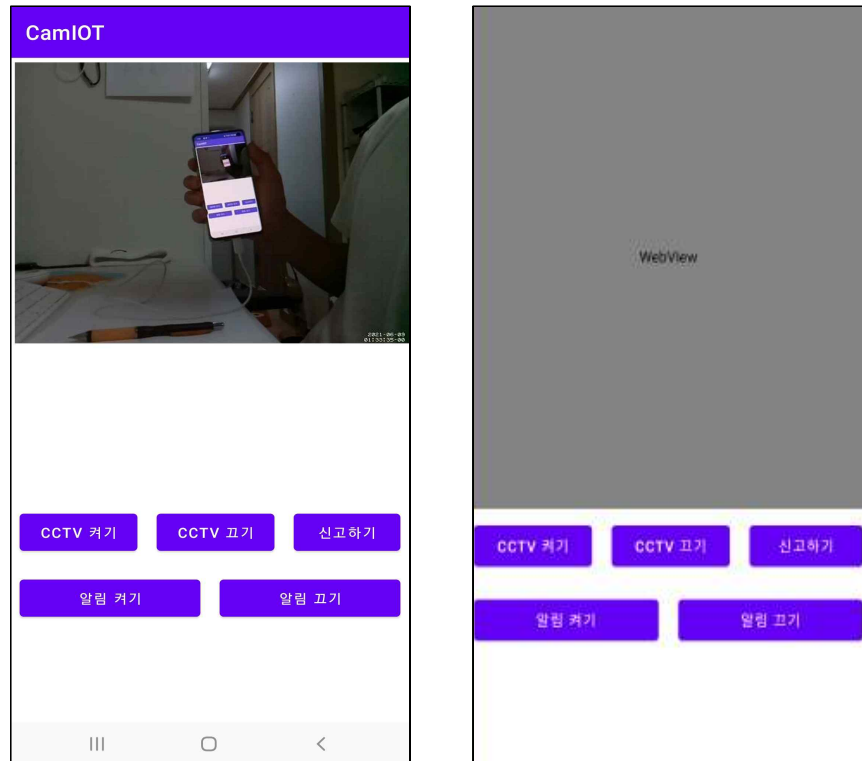


[그림 6] 흐름도 2

- [그림 5]에서 MQTT Broker로 전송된 토픽은 안드로이드가 구독하고 있어 MQTT Broker는 안드로이드에게 MQTT 메시지를 전송합니다.
- 전송받은 안드로이드는 상단바에 움직임이 감지되었다는 알림을 보여줍니다.
- 사용자는 알림을 확인하고 안드로이드 앱을 켜 CCTV 화면을 확인할 수 있습니다. 또한, 화면을 확인한 사용자는 안드로이드 앱 내의 기능들을 사용할 수 있습니다.

## 3. 화면 구성 및 기능

### 3.1 화면 구성



[그림 7] 앱의 메인화면 구성

앱의 메인화면의 UI는 크게 CCTV 화면을 보여주는 WebView와 CCTV 켜기 버튼, CCTV 끄기 버튼, 신고하기 버튼, 알림 켜기 버튼, 알림 끄기 버튼으로 구성되어 있습니다.

### 3.2 주요 기능

#### 3.2.1 CCTV 확인 기능

사용자는 앱을 접속해 WebView를 통해 라즈베리 카메라의 motion의 CCTV 화면을 확인 할 수 있습니다.

#### 3.2.2 CCTV 켜기/끄기 기능

사용자는 CCTV 화면을 키고 싶다면 CCTV 켜기 버튼을 클릭하여 CCTV 화면을 켤 수 있고, CCTV 화면을 끄고 싶다면 CCTV 끄기 버튼을 클릭하여 CCTV 화면을 끌 수 있습니다.



### 3.2.3 신고하기 기능



[그림 8] 신고 버튼 클릭 화면

- 사용자는 CCTV 화면을 확인해 침입자가 있음을 확인하면 신고하기 버튼을 클릭하여 신고할 수 있습니다.
- 신고하기 버튼을 클릭한다면 전화 다이얼로그를 통해 자동으로 112에 전화를 바로 할 수 있도록 구성하였습니다.

### 3.2.4 알림 켜기/끄기 기능

- 사용자는 알림을 키고 싶다면 알림 켜기 버튼을, 알림을 끄고 싶다면 알림 끄기 버튼을 클릭할 수 있습니다.
- 사용자가 집 내부에 있는 경우 알림 끄기 버튼을 클릭해 집에 있다는 것을 표시하여 알림이 안 오도록 할 수 있고, 사용자가 외출한 경우 알림 켜기 버튼을 통해 침입자를 방지할 수 있습니다.

### 3.2.5 움직임 감지 알림 기능



[그림 9] 움직임 알림 화면

- 사용자는 움직임 감지 기능을 통해 실시간으로 집의 움직임을 파악할 수 있습니다.
- 사용자가 외출한 집에서 외출한 경우 집에서 움직임이 발생했을 때 사용자에게 알림을 통해 움직임을 감지했다는 신호를 보낼 수 있습니다.

## 4. 코딩 보고서

### 4.1 안드로이드

#### 4.1.1 MainActivity.java

소스코드의 설명은 주석을 통해 적었습니다.

[표 2] MainActivity.java

```
package com.example.camiot;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;
import androidx.core.app.NotificationManagerCompat;

import android.app.AlertDialog;
import android.app.Notification;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Color;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.widget.Button;
import android.widget.Toast;

import org.eclipse.paho.client.mqttv3.IMqttDeliveryToken;
import org.eclipse.paho.client.mqttv3.MqttCallback;
import org.eclipse.paho.client.mqttv3.MqttClient;
import org.eclipse.paho.client.mqttv3.MqttException;
import org.eclipse.paho.client.mqttv3.MqttMessage;

public class MainActivity extends AppCompatActivity {

    private WebView cctvweb; // 웹뷰
    private WebSettings webSettings; // 웹뷰 세팅

    private Button btnOn; //cctv 켜기 버튼
    private Button btnOff; //cctv 끄기 버튼
    private Button btnReport; //신고하기 버튼
    private Button btnAlarmOn; //알림 켜기 버튼
    private Button btnAlarmOff; //알림 끄기 버튼

    //알람이On/Off인지를 저장하는 변수
    public static boolean isAlarm = true;
    //움직임 감지Count
    public static int count = 0;
    //MqttClient
    private MqttClient mqttClient;

    // 학교 서버(라즈베리 파이) : 192.168.15.84
    // private String camUrl = "http://192.168.15.84:8081";
    //mqtt server url(라즈베리파이)
```

```

// private String mqttUrl = "tcp://192.168.15.84:1883";

private String camUrl = "http://192.168.10.117:8081";
private String mqttUrl = "tcp://192.168.10.117:1883";

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    cctvweb = (WebView)findViewById(R.id.cctvweb);
    btnOn = (Button)findViewById(R.id.btnOn);
    btnOff = (Button)findViewById(R.id.btnOff);
    btnReport = (Button)findViewById(R.id.btnReport);
    btnAlarmOn = (Button)findViewById(R.id.btnAlarmOn);
    btnAlarmOff = (Button)findViewById(R.id.btnAlarmOff);

    webSettings = cctvweb.getSettings();
    webSettings.setJavaScriptEnabled(true);
    webSettings.setBuiltInZoomControls(true);
    webSettings.setSupportZoom(true);

    cctvweb.loadData("<img style = '-webkit-user-select: none;margin: auto;background-color: hsl(0, 0%, 25%);' src = 'http://192.168.10.117:8081/' width='640' height='480'>",
    "text/html", "UTF-8");

    try {
        mqttClient = new MqttClient(mqttUrl,MqttClient.generateClientId(),null);
        mqttClient.connect();
        mqttClient.subscribe("detection");
    } catch (MqttException e) {
        e.printStackTrace();
        Log.i("MQTT connect fail", "다시 연결 필요!!");
    }

    //mqttclient message
    mqttClient.setCallback(new MqttCallback() {
        @Override
        public void connectionLost(Throwable cause) {
            Log.d("connectionLost","Mqtt ReConnect");
        }
    });

    //mqtt message가 도착했을 때
    @Override
    public void messageArrived(String topic, MqttMessage message) throws Exception {

        //알람이On일때
        if(isAlarm){
            //움직임 감지가 연속으로3번된다면 알람
            Log.i("토픽: ", topic +"\t메세지: " + message.toString());
            if(count == 0){
                Log.i("count ==> ", count+"");
                count++;
            }else if(count == 1){
                Log.i("count ==> ", count+"");
                count++;
            }else if(count == 2){
                Log.i("count ==> ", count+"");
                //알람 생성
                createNotification();
                count = 0;
            }
        }
    }
}

```

```

}

}

@Override
public void deliveryComplete(IMqttDeliveryToken token) {

}

});

//CCTV 켜기 버튼
btnOn.setOnClickListener(new View.OnClickListener(){
@Override
public void onClick(View view) {
Log.i("btnOn", "btnOnClick");
isAlarm = true;
cctvweb.setVisibility(View.VISIBLE);
}
});

//CCTV 끄기 버튼
btnOff.setOnClickListener(new View.OnClickListener(){
@Override
public void onClick(View view) {
Log.i("btnOff", "btnOffClick");
isAlarm = false;
cctvweb.setVisibility(View.INVISIBLE);
}
});

//신고하기 버튼
btnReport.setOnClickListener(new View.OnClickListener(){
@Override
public void onClick(View v) {
AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);
builder.setTitle("신고");
builder.setMessage("신고하시겠습니까?");
builder.setPositiveButton("Yes", new DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialogInterface, int i) {
Intent intent = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:112"));
startActivity(intent);
}
});
builder.setNegativeButton("No", new DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialog, int which) {

}
});
AlertDialog alertDialog = builder.create();
alertDialog.show();
}
});

//알람 켜기 버튼
btnAlarmOn.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
isAlarm = true;
Toast.makeText(getApplicationContext(), "알람이 켜졌습니다.",
Toast.LENGTH_SHORT).show();
}
}

```

```

});

//알림 켜기 버튼
btnAlarmOff.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
isAlarm = false;
Toast.makeText(getApplicationContext(),"알람이 꺼졌습니다.",
Toast.LENGTH_SHORT).show();
}
});
}

//알림 생성 함수
private void createNotification(){

Intent notificationIntent = new Intent(this,MainActivity.class);
PendingIntent notificationPendingIntent =
PendingIntent.getActivity(this,0,notificationIntent,PendingIntent.FLAG_UPDATE_C
URRENT);
NotificationCompat.Builder builder = new NotificationCompat.Builder(this,
"default");

builder.setSmallIcon(R.drawable.motionicon);
builder.setTitle("움직임 감지!!");
builder.setText("움직임이 감지되었습니다. 확인 후 신고하세요!!");
builder.setContentIntent(notificationPendingIntent);

builder.setColor(Color.RED);
//사용자가 탭을 누르면 자동 제거
builder.setAutoCancel(true);

// 알림 표시
NotificationManager notificationManager = (NotificationManager)
this.getSystemService(Context.NOTIFICATION_SERVICE);
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
notificationManager.createNotificationChannel(new
NotificationChannel("default", "기본 채널",
NotificationManager.IMPORTANCE_DEFAULT));
}

// id값은 정의해야하는 각 알림의 고유한int값
notificationManager.notify(1, builder.build());

}

//알림 제거
private void removeNotification() {
NotificationManagerCompat.from(this).cancel(1);
}
}

```

## 4.1.2 activity\_main.xml

[표 3] activity\_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"

```

```

android:orientation="vertical"
tools:context=".MainActivity">

<WebView
android:id="@+id/cctvweb"
android:layout_width="match_parent"
android:layout_height="480dp"
/>

<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="horizontal"
>

<Button
android:id="@+id/btnOn"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="CCTV 켜기"
android:padding="10dp"
android:layout_margin="10dp"
/>

<Button
android:id="@+id/btnOff"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="CCTV 끄기"
android:padding="10dp"
android:layout_margin="10dp"
/>

<Button
android:id="@+id/btnReport"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="신고하기"
android:padding="10dp"
android:layout_margin="10dp"
/>

</LinearLayout>

<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="horizontal"
>

<Button
android:id="@+id/btnAlarmOn"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="알림 켜기"
android:padding="10dp"
android:layout_margin="10dp"
/>

<Button

```

```

    android:id="@+id/btnAlarmOff"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="알림 끄기"
    android:padding="10dp"
    android:layout_margin="10dp"
/>

</LinearLayout>

</LinearLayout>

```

## 4.2 아두이노

### 4.2.1 cam\_IOT.ino

[표 4] cam\_IOT.ino

```

#include <ESP8266WiFi.h> // WiFi
#include <PubSubClient.h> // mqtt

// 아두이노가 접속할 WiFi AP/password
//const char* ssid = "networking407_2.4G";
//const char* password = "networking407";
// mqtt 브로커 주소 & mqtt 브로커에 접속하는 클라이언트 이름
//const char* mqtt_server = "192.168.15.84";

const char* ssid = "Hyuckkjjjuuu";
const char* password = "123456789asd";
const char* mqtt_server = "192.168.10.117";
const char* clientName = "wonyong2"; // 중복되지 않도록 수정

WiFiClient espClient;           // WiFi 접속 객체
PubSubClient client(espClient); // mqtt 클라이언트 객체

// WiFi 접속-----
void setup_wifi() {
    delay(10);
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);

    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, password); // WiFi AP 접속 시도
    // WiFi AP 접속 시도
    while(WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    // WiFi AP 접속시 시리얼 모니터에 성공 메시지 출력
    Serial.println("");
}

```



```

Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}

// mqtt 브로커에 접속 시도-----
void reconnect() {
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");
        // mqtt 브로커 접속 시도
        if (client.connect(clientName)) {
            Serial.println("connected");
        } else {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println(" try again in 5 seconds");
            delay(5000); // 5초 마다 접속 재시도
        }
    }
}

// 아두이노 설정, WiFi AP 및 mqtt 브로커 접속-----
void setup() {
    //A0번이 인체감지센서 Input
    pinMode(A0, INPUT);
    Serial.begin(115200);
    setup_wifi(); // WiFi AP 접속
    client.setServer(mqtt_server, 1883); // mqtt 브로커에 접속
}

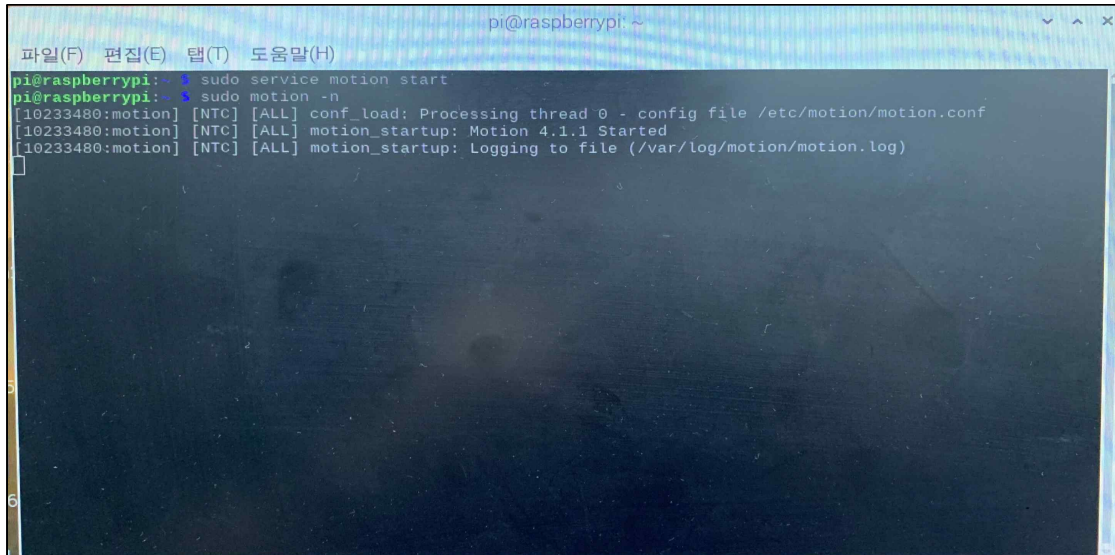
// mqtt 브로커 접속 시도-----
void loop() {
    if (!client.connected()) {
        reconnect();
    }
    client.loop(); // mqtt 브로커로부터 토픽 청취
    int pirValue = analogRead(A0);
    Serial.println(pirValue);
    //pirValue값이 900이상인 경우, 알람과 함께 detection 토픽 발행
    if(pirValue > 900){
        Serial.println("detection!!");
        client.publish("detection", "1"); //토픽발행!!!
    }else{
        Serial.println("no. detection!!");
    }

    delay(1000);
}

```

## 4.3 라즈베리 파이

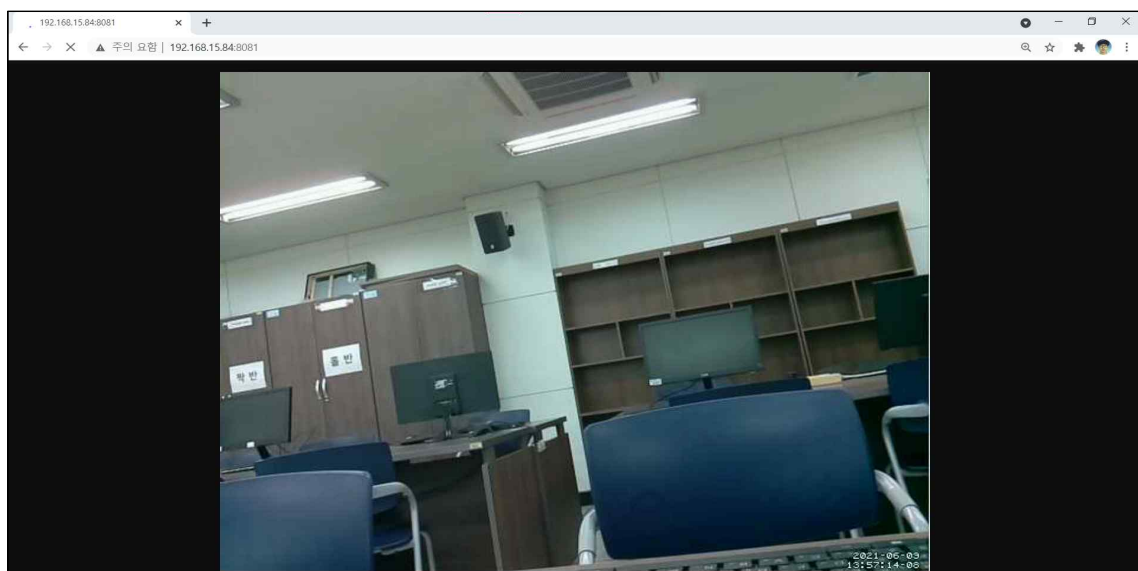
### 4.3.1 Motion 실행



```
pi@raspberrypi: ~  
pi@raspberrypi:~$ sudo service motion start  
pi@raspberrypi:~$ sudo motion -n  
[10233480:motion] [NTC] [ALL] conf_load: Processing thread 0 - config file /etc/motion/motion.conf  
[10233480:motion] [NTC] [ALL] motion_startup: Motion 4.1.1 Started  
[10233480:motion] [NTC] [ALL] motion_startup: Logging to file (/var/log/motion/motion.log)
```

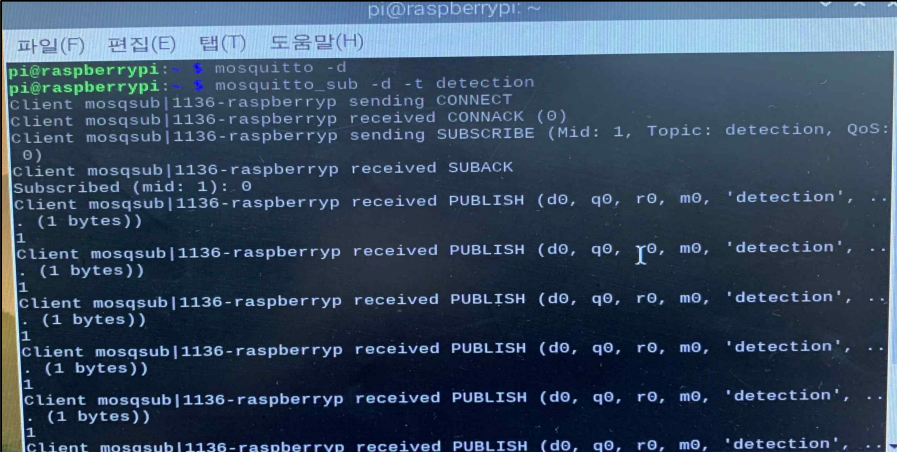
[그림 10] Motion 실행 화면

- Motion을 실행하기 위해 [그림 10]와 같이 “sudo service motion start”를 입력한 후, “sudo motion -n”을 입력하면 motion 스트리밍 서버가 실행됩니다.
- 실행된 motion 스트리밍 서버는 “http://라즈베리파이 IP주소:8081”에 접속하면 확인할 수 있습니다.



[그림 11] Motion 스트리밍 서버 화면

### 4.3.2 Mosquitto 실행



```
pi@raspberrypi: ~  
파일(F) 편집(E) 탭(T) 도움말(H)  
pi@raspberrypi:~$ mosquitto -d  
pi@raspberrypi:~$ mosquitto_sub -d -t detection  
Client mosqsub|1136-raspberryp sending CONNECT  
Client mosqsub|1136-raspberryp received CONNACK (0)  
Client mosqsub|1136-raspberryp sending SUBSCRIBE (Mid: 1, Topic: detection, QoS:  
0)  
Client mosqsub|1136-raspberryp received SUBACK  
Subscribed (mid: 1): 0  
Client mosqsub|1136-raspberryp received PUBLISH (d0, q0, r0, m0, 'detection', ..  
1  
1 (1 bytes))  
Client mosqsub|1136-raspberryp received PUBLISH (d0, q0, r0, m0, 'detection', ..  
1  
1 (1 bytes))  
Client mosqsub|1136-raspberryp received PUBLISH (d0, q0, r0, m0, 'detection', ..  
1  
1 (1 bytes))  
Client mosqsub|1136-raspberryp received PUBLISH (d0, q0, r0, m0, 'detection', ..  
1  
1 (1 bytes))  
Client mosqsub|1136-raspberryp received PUBLISH (d0, q0, r0, m0, 'detection', ..  
1  
1 (1 bytes))  
Client mosqsub|1136-raspberryp received PUBLISH (d0, q0, r0, m0, 'detection', ..
```

[그림 12] Mosquitto 실행 화면

- MQTT Broker인 Mosquitto를 실행하기 위해서는 [그림 12]과 같이 “mosquitto -d”를 입력하여 Mosquitto 서버를 실행합니다.
- 실행한 서버에서 본 프로젝트에서 사용하는 토픽인 “detection” 토픽을 구독하기 위해서 “mosquitto\_sub -d -t detection”의 명령어를 사용하면, 구독한 토픽이 발행되었을 때 확인 할 수 있습니다.

## 5. 참고자료

---

- LG U+ 펫케어

<https://www.uplus.co.kr/ent/shome/lotAnimalpackInfo.hpi>

- ADT캡스 뷰가드미니 2.0

[https://www.adtcaps.co.kr/caps\\_home/door\\_guard.asp?utm\\_source=naver&utm\\_medium=cpc-sa-pc&utm\\_campaign=%EC%BA%A1%EC%8A%A4%ED%99%88&utm\\_content=cctv-%EB%8C%80%ED%91%9C&utm\\_term=%ED%99%88CCTV&NaPm=ct%3Dkpp6k4zk%7Cci%3D0Am0002Cra1uL4Dbm0YO%7Ctr%3Dsa%7Chk%3D86066da4f2b4ea752f09a26a4553fc16cd406145](https://www.adtcaps.co.kr/caps_home/door_guard.asp?utm_source=naver&utm_medium=cpc-sa-pc&utm_campaign=%EC%BA%A1%EC%8A%A4%ED%99%88&utm_content=cctv-%EB%8C%80%ED%91%9C&utm_term=%ED%99%88CCTV&NaPm=ct%3Dkpp6k4zk%7Cci%3D0Am0002Cra1uL4Dbm0YO%7Ctr%3Dsa%7Chk%3D86066da4f2b4ea752f09a26a4553fc16cd406145)

- 인체감지센서 다루기

<https://rasino.tistory.com/194>

- 감지신호 알람 올리기

<https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=hcheong&logNo=221171346501>

- 카메라 연동

<https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=scw0531&logNo=220653503111>

- 안드로이드 알림 사용하기

<https://sh-itstory.tistory.com/63>