

홈 보호 시스템 *From Home*

캡스톤 디자인



손현지

정보보호과 201741023



목차

소개

시스템 주제, 이름, 개발 배경, 목적을 설명하고 기존의 시스템과 차별점에 대해 설명

5p

기능

시스템의 기능, 시스템 동작 방식, 기능 구현에 따른 시스템 개발 일정을 설명

8p

구현 상황

중간 발표 때의 구현상황을 리뷰하고 완료된 구현 상황을 설명

27p

결론

시스템 완성 후 결론

28p

시스템 소개

시스템 주제

홈 보호 시스템

집에 침입자가 있는지 파악해 사용자에게 알림

시스템 개발 배경

영화 '도어락'을 통해 아이디어 구상

낯선 사람의 침입을 미리 안다면 큰 위험 예방

시스템 이름

From Home

집으로부터 오는 연락이라는 뜻을 가짐

시스템 목적

① 사전에 위험을 예방할 수 있게 함



② 가족에게 귀가 여부를 알려
서로의 안전을 파악할 수 있게 함



기존 시스템과 차별점

세콤 홈 블랙박스

원격으로 난방을 켜거나 가스 밸브를 잠그는 등 생활 편리 기능도 제공
자석 감지기, 열선 감지기를 통해 실내 침입 여부를 감지

서비스 요금 : 월 **69,000원**

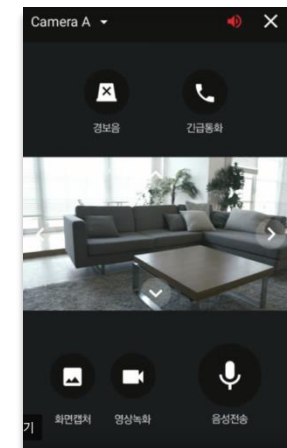


kt 기가 IoT 홈캠

집 24시간 모니터링 가능

스마트폰 앱을 통해 홈캠으로 촬영되는 영상을 실시간으로 볼 수 있음

이용 요금 : **1만원 대(계약 조건에 따라 상이)**



시스템 기능



- ① 지문 인식을 통해 적외선 인체 감지 센서 해제
 - ② 가족에게 귀가 정보 메시지 전송
- ※ 허가되지 않은 사람 존재 시 앱에 표시

적외선 인체 감지 센서 해제 못함

- ① 센서에 감지 됨
- ② 감지된 정보 기록

시스템 동작 방식

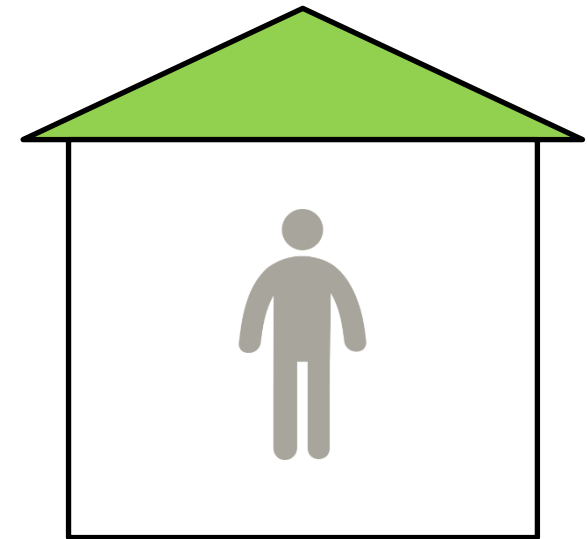


시스템 동작 방식



사용자

침입자 알림

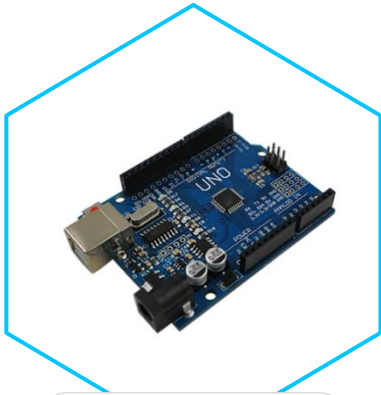


집에 침입자 존재

시스템 개발 일정

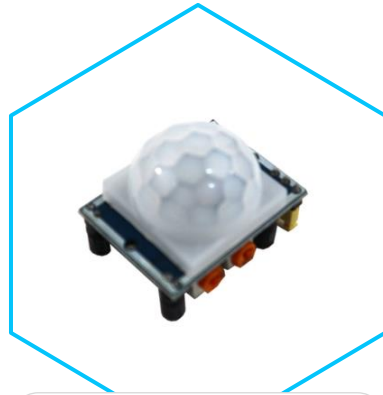
주차	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
아이디어 고안 후 주제 확정															
착수 보고서 작성 및 제출															
장치 제작 후 작동 확인															
중간 발표															
앱 개발															
아두이노 코드 추가 작성															
집 현관 모형 제작															
최종 테스트															
최종 보고서 작성 및 제출															
기말 발표															

중간 발표 구현 상황



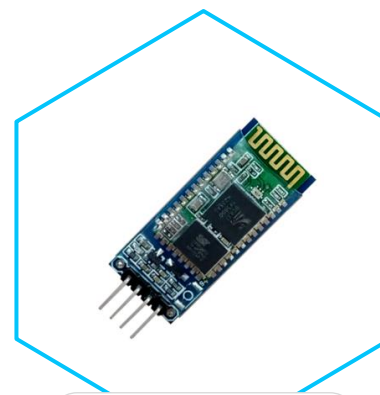
아두이노보드

적외선 인체 감지 센서 블루투스 모듈
연결



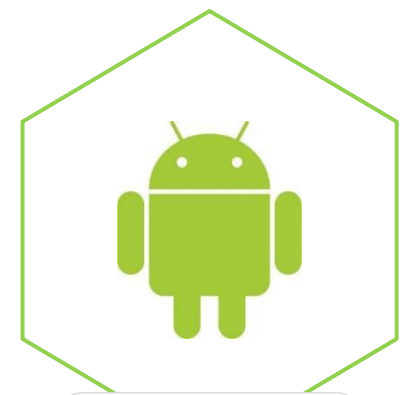
적외선 인체감지 센서

침입자 존재 여부를 판단



블루투스 모듈

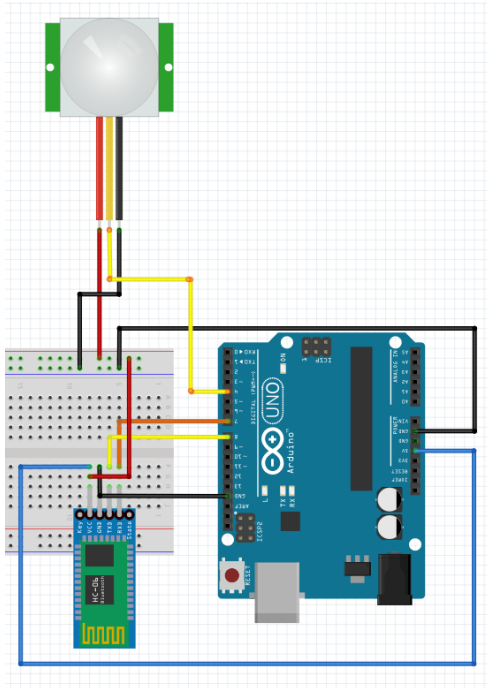
보드에 연결된 센서와 앱이
서로 통신할 수 있도록 함



안드로이드 앱

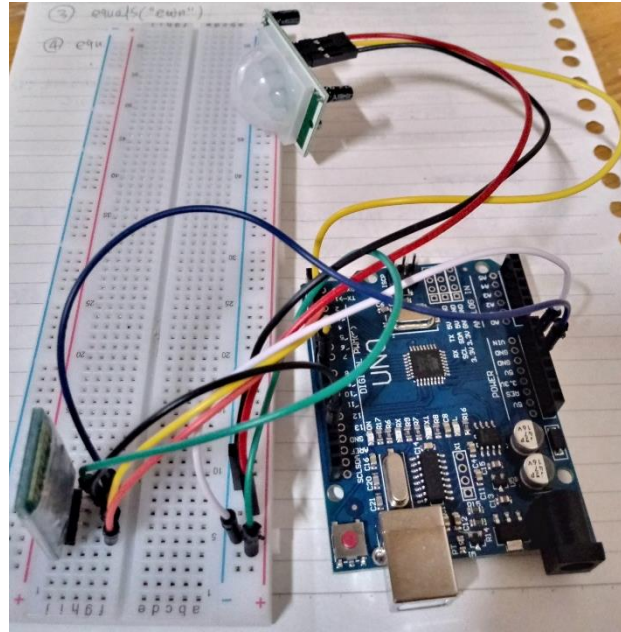
적외선 인체 감지 센서 off
&
침입자 존재 시 알림 받음

중간 발표 구현 상황



장치 설계도

Fritzing 프로그램 이용해 설계



제작한 장치

설계도를 기준으로 제작



블루투스 앱

블루투스 모듈과 통신하는 앱

현재 구현 상황



100%

개발 진척도

기말 발표 때의 구현 상황

전체 개발 진척도 100%

장치 제작, 앱 개발 완료



100%

장치 제작



100%

앱 개발

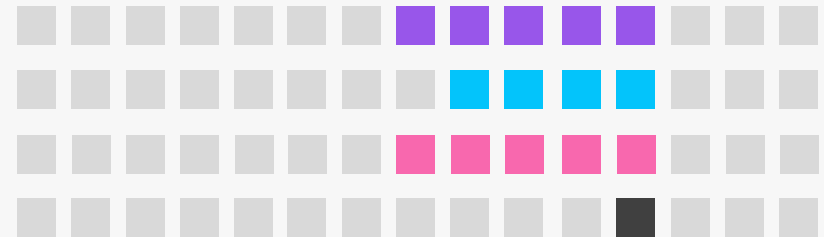
개발 일정

앱 개발 5주

코드 추가 작성 4주

모형 제작 5주

테스트 1주



현재 구현 상황 아두이노 코딩

적외선 인체 감지 센서 관련 변수 선언 부분

- count 변수는 침입자 존재 여부를 기록하는 변수
- state는 집의 상황을 앱으로 전달할 때 사용하는 변수

```
//인체감지센서
int inputPin = 4; //인체감지센서를 4번핀으로 설정
int pirState = LOW; //센서 초기상태는 움직임이 없음을 가정
int val = 0; //센서 신호의 판별을 위한 변수
int count = 0; //허가 되지 않은 사람이 들어왔다는 기록

char state = ' ';
```

적외선 인체 감지 센서 동작 관련 부분

- 적외선 인체 감지 센서에 사람이 감지될 경우
count 변수의 1 값이 들어감



```
void loop(){
  digitalWrite(inputPin,HIGH); //인체감지센서 ON
  val = digitalRead(inputPin); //인체감지센서로부터 값 읽어오기

  if(val==HIGH){ //사람 지나갈 때 |
    if(pirState==LOW){
      pirState = HIGH;
      count = 1; //허가 받지 않은 사람 들어옴
      Serial.println(count);
    }
  }else{ //시나가고 난 뒤에
    if(pirState==HIGH){
      pirState = LOW; //값 원래대로 돌림 (움직임이 없다고 가정)
    }
  }
}
```

현재 구현 상황 아두이노 코딩

앱과 블루투스 통신 부분 - 침입자 체크

- 앱에서 보내는 데이터를 읽어와 data 변수에 저장
- data변수의 값이 'c'이면 앱에서 침입자 존재 여부 확인
- count변수 값이 1이면 침입자 존재, 0이면 없음
- 침입자 존재하면 앱으로 'e'라는 값을 보냄

```
if (bluetooth.available()) { //블루투스로 데이터 수신
    byte data = bluetooth.read();
    if(data == 'c'){
        switch(count){
            case 1:
                state = 'e';
                bluetooth.write(state);
                Serial.write(state);
                count = 0;
                break;
            case 0:
                state = 'n';
                bluetooth.write(state);
                Serial.write(state);
                break;
        }
    }
}
```

앱과 블루투스 통신 부분 - 적외선 인체 감지 센서 해제

- 앱에서 지문 인증 과정을 거침
- 지문 인증 성공했으면 적외선 인체 감지 센서 해제 가능
- delay()를 이용해 사용자가 지나갈 수 있도록 함

```
}else if(data == '0'){
    digitalWrite(inputPin,LOW); //인체감지센서 OFF
    Serial.write(data);
    delay(5000); //허가된 사람이 지나가는 시간 (현재 5초로 설정)
}
}
```

현재 구현 상황 안드로이드 앱

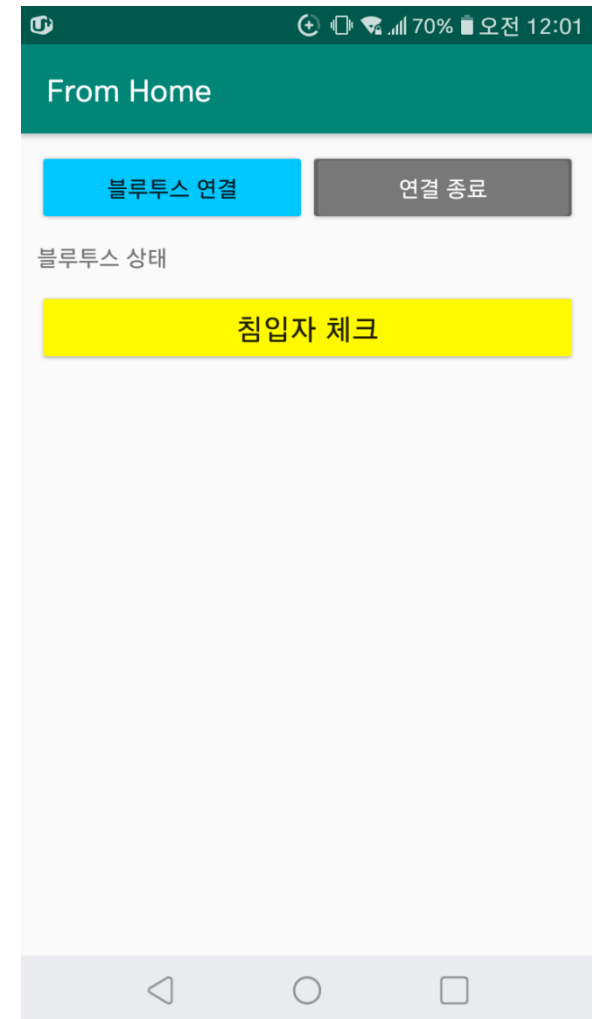
From Home 앱 Main 화면

- 블루투스 연결, 종료 버튼으로 블루투스 제어
- 블루투스 상태에 따라 TextView에 표시됨
- 침입자 체크 버튼을 통해 현재 집에 침입자 존재 여부 확인

```
buttonBluetoothConnect.setOnClickListener(new Button.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        bluetoothOn();  
    }  
});
```

```
buttonBluetoothDisconnect.setOnClickListener(new Button.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        bluetoothOff();  
    }  
});
```

```
buttonCheckStranger.setOnClickListener((v) -> {  
    if(threadConnectedBluetooth != null) {  
        threadConnectedBluetooth.write( str. "c");  
    }  
    else {  
        Toast.makeText(getApplicationContext(), text. "블루투스 장치와 연결이 되어 있지 않습니다.", Toast.LENGTH_LONG).show();  
    }  
});
```

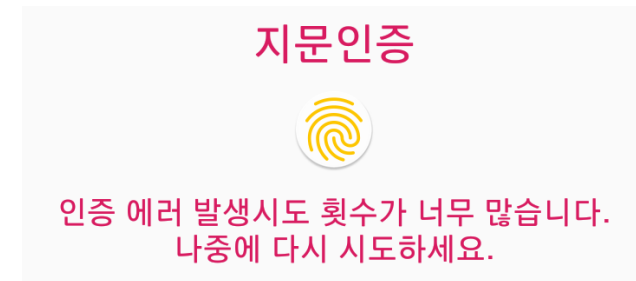
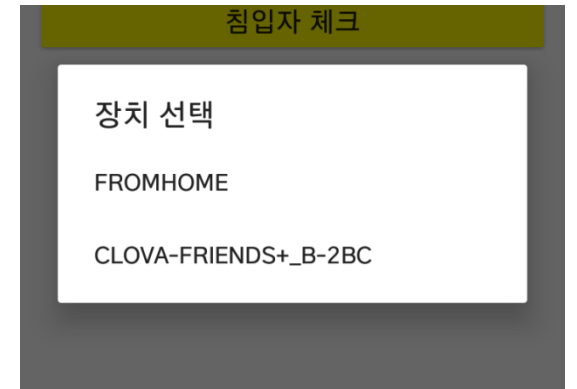
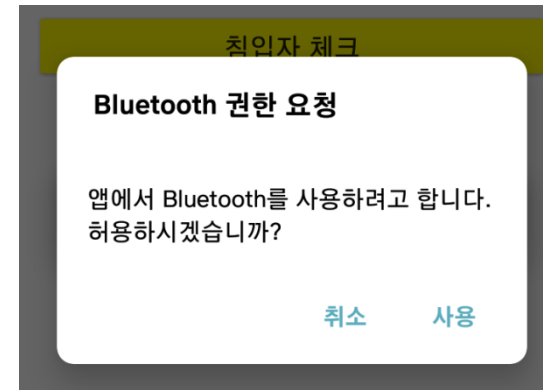


현재 구현 상황 안드로이드 앱

From Home 앱에서 필요한 권한

- 블루투스에 필요한 권한 : BLUETOOTH, BLUETOOTH_ADMIN
- 지문 인식에 필요한 권한 : USE_FINGERPRINT
- AndroidManifest.xml에 작성

```
<uses-permission android:name="android.permission.BLUETOOTH" />  
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />  
<uses-permission android:name="android.permission.USE_FINGERPRINT" />
```

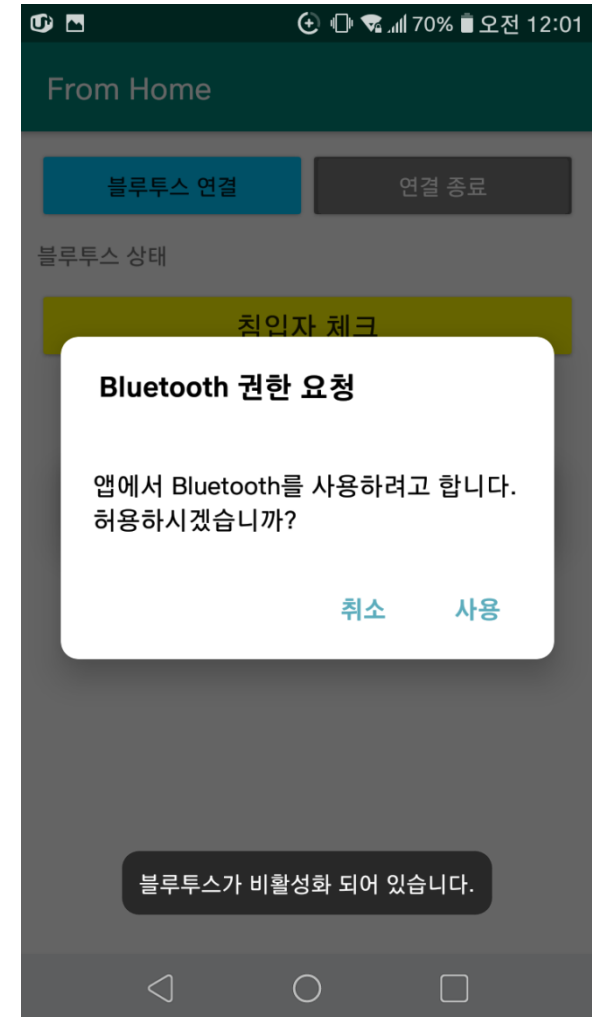


현재 구현 상황 안드로이드 앱

블루투스 연결 버튼 눌렀을 때 1

- 블루투스를 지원하는 기기라면 블루투스 활성화
- 블루투스 활성화 되면 "활성화"라고 토스트 메시지 나옴
- 블루투스 활성화 되면 연결 가능한 장치 목록 보여주는 함수 호출 → listPairedDevices();

```
void bluetoothOn() {  
    if(blueetoothAdapter == null) {  
        Toast.makeText(getApplicationContext(), text "블루투스를 지원하지 않는 기기입니다.", Toast.LENGTH_LONG).show();  
    }  
    else {  
        if(blueetoothAdapter.isEnabled()) {  
            // Toast.makeText(getApplicationContext(), "블루투스가 이미 활성화 되어 있습니다.", Toast.LENGTH_LONG).show();  
            textViewBluetoothStatus.setText("활성화");  
            listPairedDevices();  
        }  
        else {  
            Toast.makeText(getApplicationContext(), text "블루투스가 비활성화 되어 있습니다.", Toast.LENGTH_LONG).show();  
            Intent intentBluetoothEnable = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);  
            startActivityForResult(intentBluetoothEnable, BT_REQUEST_ENABLE);  
        }  
    }  
}
```

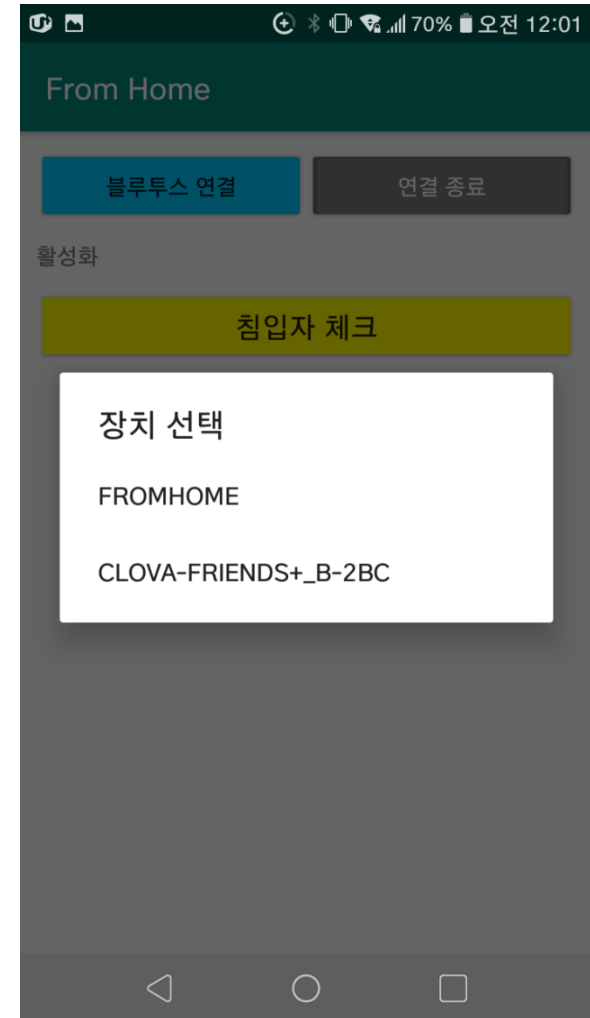


현재 구현 상황 안드로이드 앱

블루투스 연결 버튼 눌렀을 때 2

- 핸드폰에 저장된 페어링 목록을 가져와 listPairedDevices에 저장하고 다이얼로그 화면으로 출력
- 다이얼로그 화면에서 선택하면 connectSelectedDevice 함수가 호출됨

```
void listPairedDevices() {  
    if (bluetoothAdapter.isEnabled()) {  
        pairedDevices = bluetoothAdapter.getBondedDevices();  
  
        if (pairedDevices.size() > 0) {  
            AlertDialog.Builder builder = new AlertDialog.Builder(context, this);  
            builder.setTitle("장치 선택");  
  
            listPairedDevices = new ArrayList<String>();  
            for (BluetoothDevice device : pairedDevices) {  
                listPairedDevices.add(device.getName());  
                //mListPairedDevices.add(device.getName() + "\n" + device.getAddress());  
            }  
            final CharSequence[] items = listPairedDevices.toArray(new CharSequence[listPairedDevices.size()]);  
            listPairedDevices.toArray(new CharSequence[listPairedDevices.size()]);  
  
            builder.setItems(items, (dialog, item) -> {  
                connectSelectedDevice(items[item].toString());  
            });  
            AlertDialog alert = builder.create();  
            alert.show();  
        } else {  
            Toast.makeText(getApplicationContext(), text: "페어링된 장치가 없습니다.", Toast.LENGTH_LONG).show();  
        }  
    } else {  
        Toast.makeText(getApplicationContext(), text: "블루투스가 비활성화 되어 있습니다.", Toast.LENGTH_SHORT).show();  
    }  
}
```

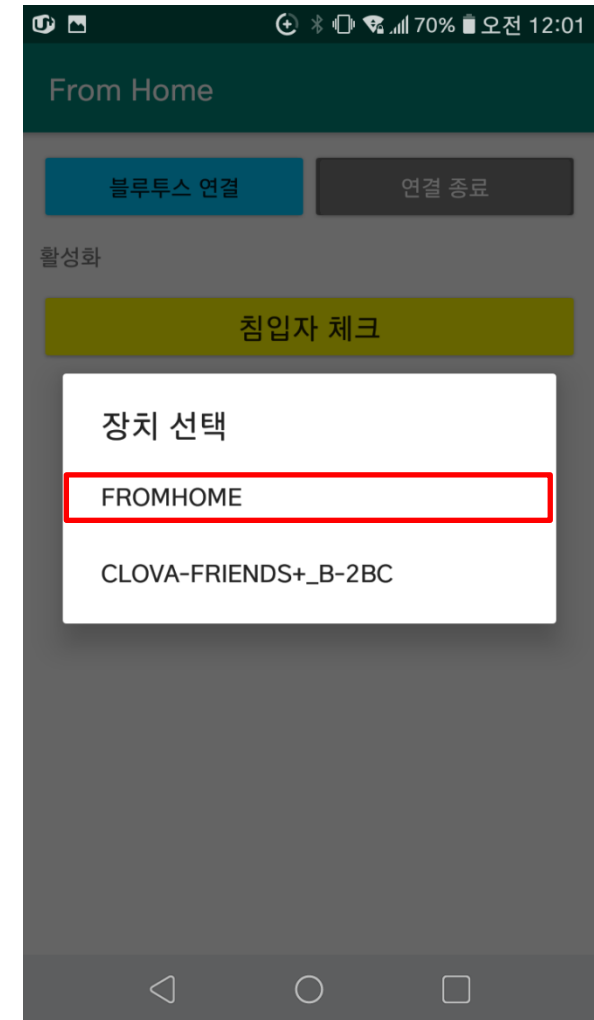


현재 구현 상황 안드로이드 앱

블루투스 연결 버튼 눌렀을 때 3

- 다이얼로그 화면에서 장치를 선택
- 장치가 선택되면 connectSelectedDevice 함수를 통해 해당 장치와 연결

```
void connectSelectedDevice(String selectedDeviceName) {  
    for(BluetoothDevice tempDevice : pairedDevices) {  
        if (selectedDeviceName.equals(tempDevice.getName())) {  
            bluetoothDevice = tempDevice;  
            break;  
        }  
    }  
    try {  
        bluetoothSocket = bluetoothDevice.createRfcommSocketToServiceRecord(BT_UUID);  
        bluetoothSocket.connect();  
        threadConnectedBluetooth = new ConnectedBluetoothThread(bluetoothSocket);  
        threadConnectedBluetooth.start();  
        bluetoothHandler.obtainMessage(BT_CONNECTING_STATUS, arg1: 1, arg2: -1).sendToTarget();  
    } catch (IOException e) {  
        Toast.makeText(getApplicationContext(), text: "블루투스 연결 중 오류가 발생했습니다.", Toast.LENGTH_LONG).show();  
    }  
}
```



현재 구현 상황 안드로이드 앱

침입자 체크 버튼 눌렀을 때 1

- 블루투스로 장치와 연결된 상태에서 침입자 체크 버튼 클릭
- 아두이노로 'c'를 threadConnectedBluetooth의 write를 이용해 보냄

```
buttonCheckStranger.setOnClickListener((v) -> {  
    if(threadConnectedBluetooth != null) {  
        threadConnectedBluetooth.write( str: "c");  
    }  
    else {  
        Toast.makeText(getApplicationContext(), text: "블루투스 장치와 연결이 되어 있지 않습니다.", Toast.LENGTH_LONG).show();  
    }  
});
```

```
public void write(String str) {  
    byte[] bytes = str.getBytes();  
    try {  
        outputStream.write(bytes);  
    } catch (IOException e) {  
        Toast.makeText(getApplicationContext(), text: "데이터 전송 중 오류가 발생했습니다.", Toast.LENGTH_LONG).show();  
    }  
}  
  
public void cancel() {  
    try {  
        socket.close();  
    } catch (IOException e) {  
        Toast.makeText(getApplicationContext(), text: "소켓 해제 중 오류가 발생했습니다.", Toast.LENGTH_LONG).show();  
    }  
}
```

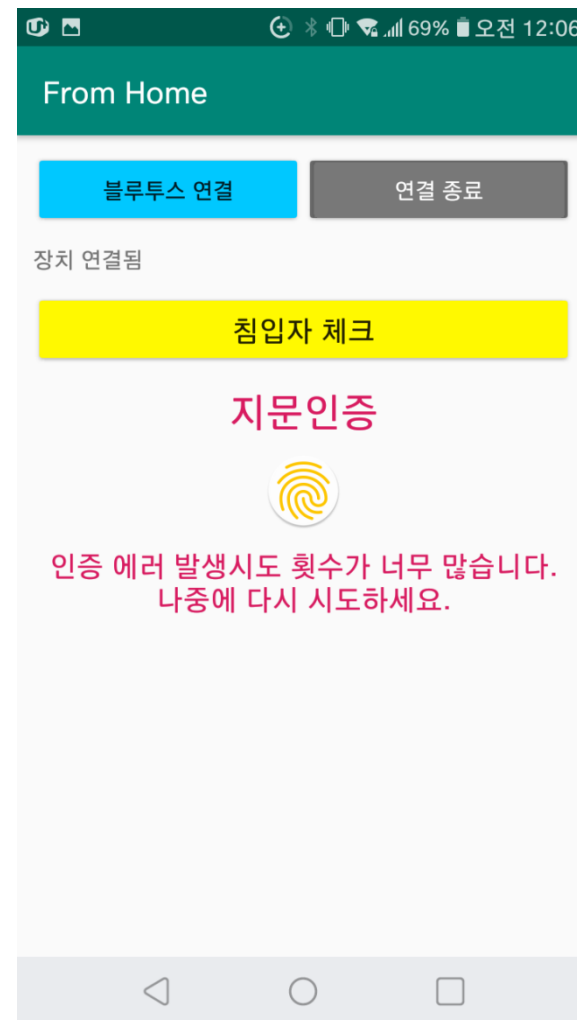


현재 구현 상황 안드로이드 앱

침입자 체크 버튼 눌렀을 때 2

- 아두이노에서 침입자 존재 여부를 보냄 (존재하면 "e", 존재 하지 않으면 "n")
- 아두이노에서 보낸 값을 checkHome 함수를 이용해 확인
- 존재한다면 신고할 수 있는 화면을 없다면 지문 인식을 받을 수 있는 화면 보여줌

```
void checkHome(String receiveData){  
    LinearLayoutExistStranger = (LinearLayout)findViewById(R.id.linearlayoutExistStranger);  
    LinearLayoutNoStranger = (LinearLayout)findViewById(R.id.linearlayoutNoStranger);  
    LinearLayoutPass = (LinearLayout)findViewById(R.id.linearlayoutPass);  
  
    if(receiveData.equals((String)"e")){  
        Toast.makeText(getApplicationContext(), text "침입자 존재", Toast.LENGTH_SHORT).show();  
  
        LinearLayoutNoStranger.setVisibility(View.GONE);  
        LinearLayoutPass.setVisibility(View.GONE);  
        LinearLayoutExistStranger.setVisibility(View.VISIBLE);  
    }else if(receiveData.equals((String)"n")){  
        Toast.makeText(getApplicationContext(), text "침입자 없음", Toast.LENGTH_SHORT).show();  
  
        LinearLayoutExistStranger.setVisibility(View.GONE);  
        LinearLayoutNoStranger.setVisibility(View.VISIBLE);  
    }else{  
        Toast.makeText(getApplicationContext(), receiveData, Toast.LENGTH_SHORT).show();  
    }  
}
```

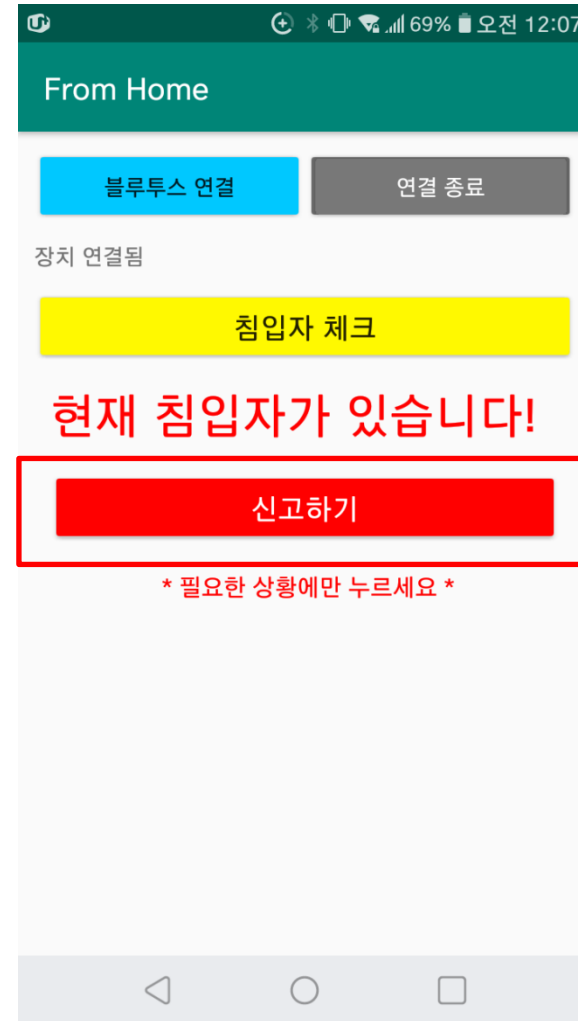


현재 구현 상황 안드로이드 앱

침입자가 존재할 때

- 침입자가 있다는 문구를 보여줌
- 신고하기 버튼을 통해 112로 신고할 수 있도록 해줌

```
buttonCall = (Button)findViewById(R.id.buttonCall);  
buttonCall.setOnClickListener((v) -> {  
    Intent intent = new Intent(Intent.ACTION_DIAL,Uri.parse("tel:119"));  
    startActivity(intent);  
});
```



현재 구현 상황 안드로이드 앱

침입자 없을 때 1

- 지문 인증을 받도록 함
- 지문 인증 관련 파일 생성 (FingerprintHandler.java)
- 지문 인증은 최소 버전이 마시멜로

```
public FingerprintHandler(Context context){
    this.context = context;
}

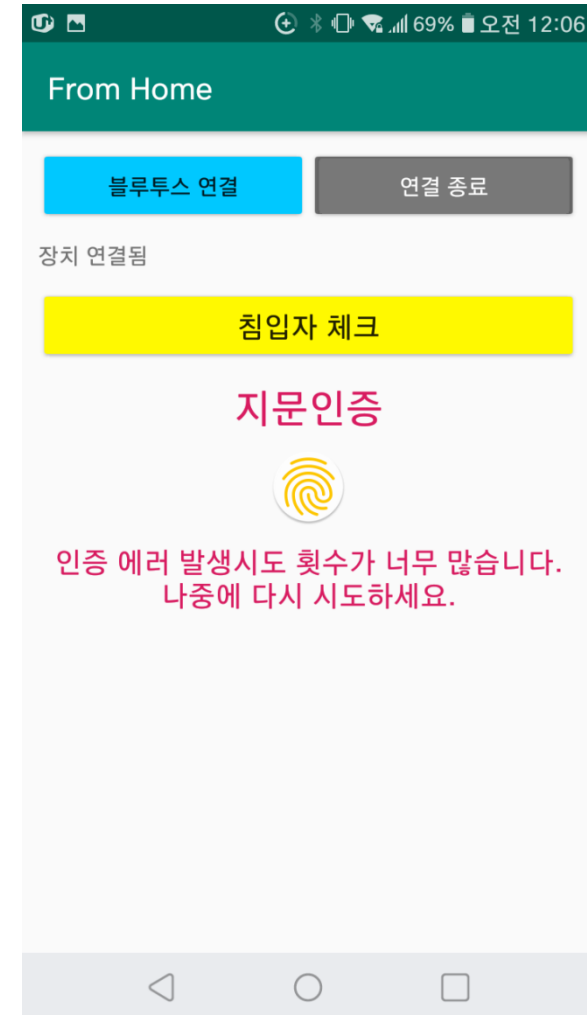
//메소드들 정의
public void startAuth(FingerprintManager fingerprintManager, FingerprintManager.CryptoObject cryptoObject){
    cancellationSignal = new CancellationSignal();
    fingerprintManager.authenticate(cryptoObject, cancellationSignal, flags: 0, callback: this, handler: null);
}

@Override
public void onAuthenticationError(int errorCode, CharSequence errString) {
    this.update( s: "인증 에러 발생" + errString, b: false);
}

@Override
public void onAuthenticationHelp(int helpCode, CharSequence helpString) {
    this.update( s: "Error : " + helpString, b: false);
}

@Override
public void onAuthenticationSucceeded(FingerprintManager.AuthenticationResult result) {
    this.update( s: "앱 접근이 허용되었습니다.", b: true);
}

@Override
public void onAuthenticationFailed() {
    this.update( s: "인증 실패", b: false);
}
```



침입자 없을 때 2

- 지문 인 증은 최소 버전이 마시멜로
- 핸드폰이 버전을 충족하고 등록된 지문이 있는지 확인

```
imageViewFingerprint = (ImageView) findViewById(R.id.imageViewFingerprint);
textViewGuidance = (TextView) findViewById(R.id.textViewGuidance);

if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
    fingerprintManager = (FingerprintManager) getSystemService(FINGERPRINT_SERVICE);
    keyguardManager = (KeyguardManager) getSystemService(KEYGUARD_SERVICE);

    if (!fingerprintManager.isHardwareDetected()) {
        textViewGuidance.setText("지문을 사용할 수 없는 핸드폰입니다.");
    } else if (ContextCompat.checkSelfPermission(context, this, Manifest.permission.USE_FINGERPRINT) != PackageManager.PERMISSION_GRANTED) {
        textViewGuidance.setText("지문 사용을 허가해주세요.");
    } else if (!keyguardManager.isKeyguardSecure()) {
        textViewGuidance.setText("등록된 지문이 없습니다.");
    } else { // 모든 관문을 성공적으로 통과(지문인식을 지원, 지문 사용 허가, 잠금 화면 설정, 지문 등록되어있을 때)
        textViewGuidance.setText("손가락을 홈버튼에 올려주세요.");

        generateKey();
        if (cipherInit()) {
            cryptoObject = new FingerprintManager.CryptoObject(cipher);
            // 핸들러 실행
            FingerprintHandler fingerprintHandler = new FingerprintHandler(context, this);
            fingerprintHandler.startAuth(fingerprintManager, cryptoObject);
        }
    }
}
```


현재 구현 상황 안드로이드 앱

침입자 없을 때 3

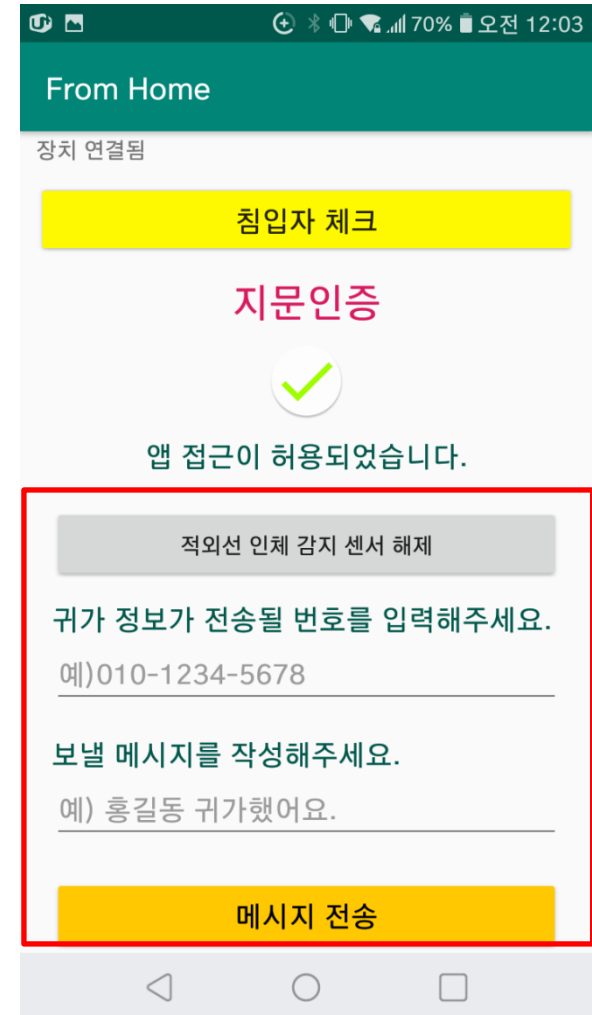
- 지문 인증에 성공했을 때 화면 보여줌

```
private void stopFingerAuth(){
    if(cancellationSignal != null && !cancellationSignal.isCanceled()){
        cancellationSignal.cancel();
    }
}

private void update(String s, boolean b){
    final TextView textviewGuidance = (TextView)((Activity)context).findViewById(R.id.textviewGuidance);
    final ImageView imageviewFingerprint = (ImageView)((Activity)context).findViewById(R.id.imageviewFingerprint);
    final LinearLayout linearlayoutPass = (LinearLayout)((Activity)context).findViewById(R.id.linearlayoutPass);

    // 안내 메시지 출력
    textviewGuidance.setText(s);

    if(b == false){
        textviewGuidance.setTextColor(ContextCompat.getColor(context, R.color.colorAccent));
    }else{
        // 지문 인증 성공
        textviewGuidance.setTextColor(ContextCompat.getColor(context, R.color.colorPrimaryDark));
        imageviewFingerprint.setImageResource(R.mipmap.done);
        linearlayoutPass.setVisibility(LinearLayout.VISIBLE);
    }
}
```



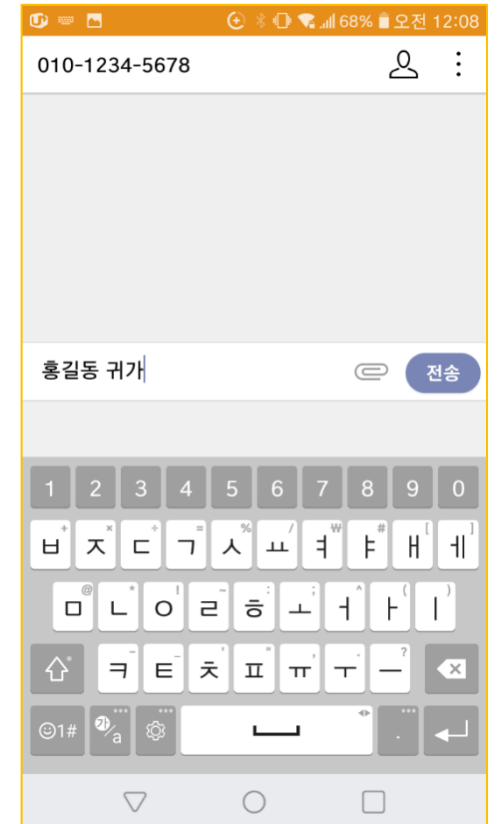
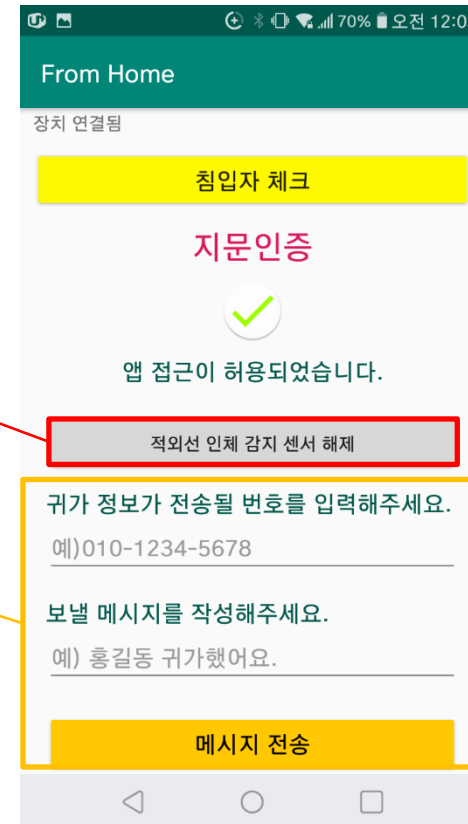
현재 구현 상황 안드로이드 앱

침입자 없을 때 4

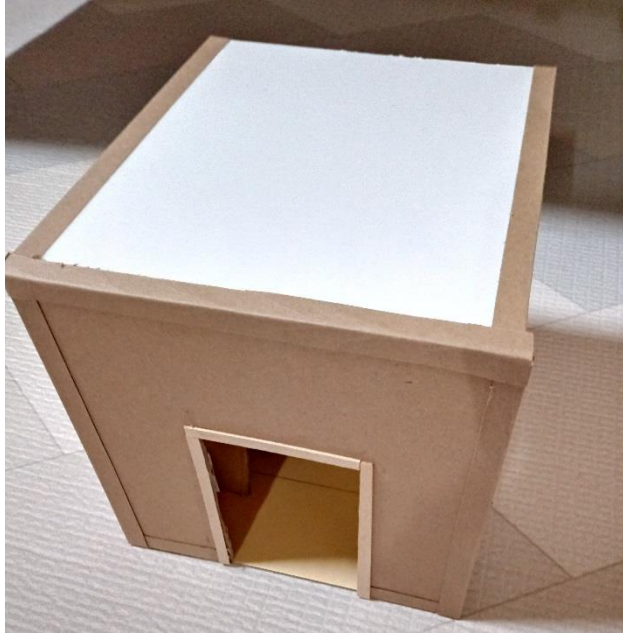
- 적외선 인체 감지 센서 해제를 누르면 아두이노로 값을 보내 센서 해제
- 번호와 내용을 입력하고 메시지 전송을 누르면 메시지 전송 화면으로 감

```
buttonSensorOff.setOnClickListener(new Button.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        if(threadConnectedBluetooth != null) {  
            threadConnectedBluetooth.write(str: "0");  
        }  
    }  
});
```

```
public void sendSmsIntent(String number){  
    try{  
        Uri smsUri = Uri.parse("sms:"+number);  
        Intent sendIntent = new Intent(Intent.ACTION_SENDTO, smsUri);  
        sendIntent.putExtra(name: "sms_body", editTextSms.getText().toString());  
        startActivity(sendIntent);  
    }catch (Exception e){  
        e.printStackTrace();  
    }  
}
```



현재 구현 상황 **현관문 모형**



전체 모습
하드보드지로 제작



정면
현관문 앞 모습으로
문 안에 적외선 인체 감지 센서



뒷면
전체 장치를 안에 설치
USB 포트와 연결해 전력 공급

현재 구현 상황 동영상



침입자 존재 하지 않음



침입자 존재

결론

From Home

장치를 통해 침입자 존재 여부를 판단한 뒤 사용자에게 알려 위험을 사전에 방지
낮은 비용으로 안전을 지킴
가족에게 연락이 오지 않을 경우 무슨 일이 있는지 파악할 수 있게 함

개선할 점

처음 목표는 데이터베이스를 사용해 사용자 정보를 저장
정보를 저장하면 가족들의 귀가 정보를 앱에서 표시할 수 있음
지문 인식을 지원하지 않는 기기의 경우 대체할 부분이 필요

참고 문헌

참고한 부분		참고한 사이트 주소
장치	적외선 인체 감지 센서	에듀이노 > 적외선 인체 감지 센서(HC-SR501) https://eduino.kr/product/detail.html?product_no=104&cate_no=62&display_group=1
	블루투스 모듈	에듀이노 > 블루투스 모듈(HC-06) https://eduino.kr/product/detail.html?product_no=27&cate_no=55&display_group=1
앱	블루투스 통신	Bugwhale World > 안드로이드 블루투스 애플리케이션 만들기 https://bugwhale.com/android-bluetooth-application/
	지문 인식	사소한 시선 > 안드로이드 지문 인증 사용하기(1) : Fingerprint Authentication https://m.blog.naver.com/cosmosjs/221381903806
	전화 걸기	명우니 닷컴 > [Do it;Android]안드로이드 - 버튼 클릭 시 전화 걸기 https://myeonguni.tistory.com/1440
	문자 메시지 보내기	혁준 블로그 > 안드로이드 Intent로 SMS, MMS, Email 보내기 https://ghj1001020.tistory.com/750
제품 비교	세콤 홈 블랙박스	에스원 > 세콤 > 세콤 홈 블랙박스 https://www.s1.co.kr/service/service01_01.do
	kt 기가 IoT 홈캠	kt > 상품서비스 > 기가IoT > 스마트홈 https://product.kt.com/wDic/productDetail.do?ItemCode=1064
기타	현관문 모형	HATSAE 핫새 > 미니어쳐 하드보드지로 룸박스 만들기 1편 https://youtu.be/480_MV1wLA0
	PPT 템플릿	조땡 http://pptbizcam.co.kr/