

특정 트리거를 활용한 스마트홈 시스템 구현

스마트자취방

문기준 안병욱 조형준 차상순 최은희

01

프로젝트 개요

02

배경 및 목적

03

작동방식

04

구현 상세

05

실행 영상

Problem

현재 보편적인 스마트홈 시스템은 스마트스피커를 중심으로 작동

하지만 청각장애인 등 언어로 특정 명령을 전달할 수 없는 사람들에게 있어서는 진입장벽 존재

특정 루틴을 작동시키기 위해
일일이 음성으로 전달

반복되는 루틴을 쉽게
적용할 수 있는 방법 부재

음성으로 전달이 불가능한 고객군의 경우,
높은 진입장벽 존재

Solution

스마트 자취방 ;

아두이노 센서의 물리적 신호를 특정 트리거로 활용한 스마트홈 시스템 개선



사용자의 동작에 의한 루틴

사용자의 동작이 트리거가 되었을 때
작동하는 패턴 활용



외부의 환경변화에 의해 작동하는 루틴

외부 환경변화를 데이터로 받았을 때
그에 따라 작동하는 패턴 활용

작동방식

Google Home

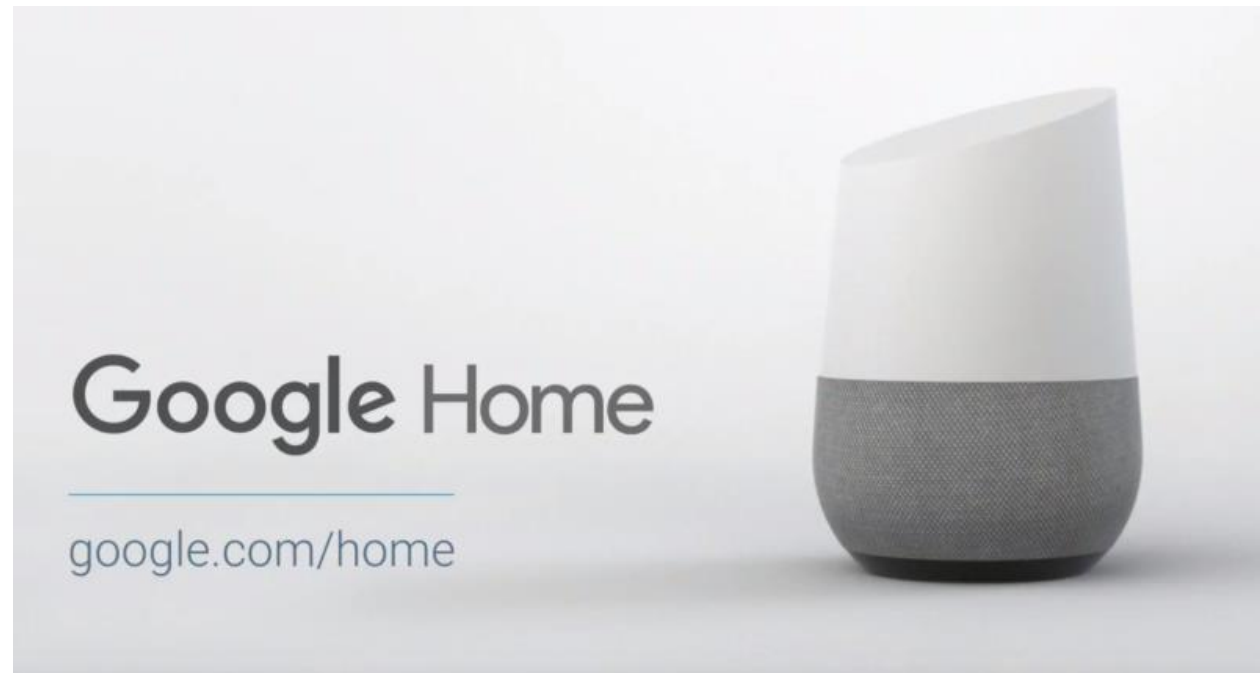
핵심은 구글 홈(Google Home)!

구글 홈에 입력을 전달하고,
구글 홈이 목표 기기에 신호를 보낸다.

Google Home

구글 홈은?

- ✓ 구글 어시스턴트를 활용한 스마트 스피커



Google Home

구글 홈의 기능

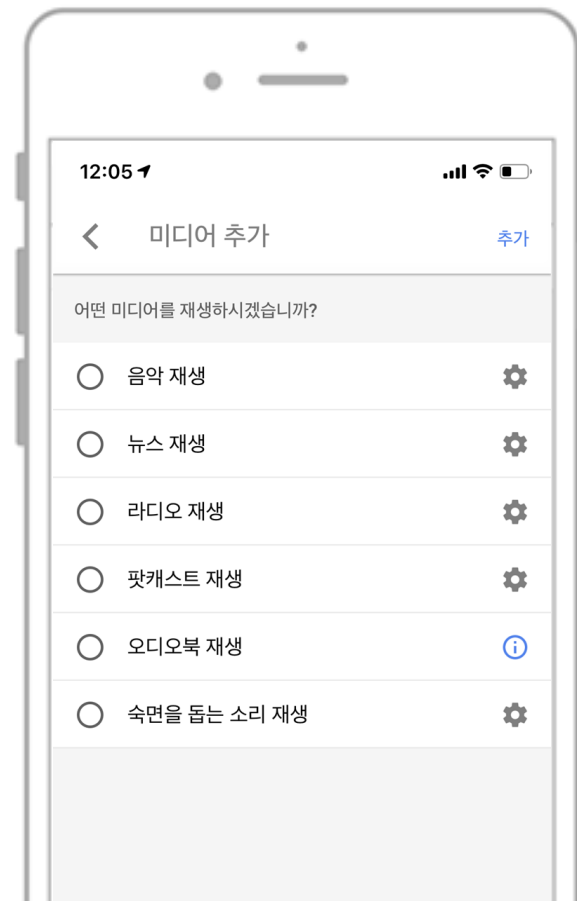
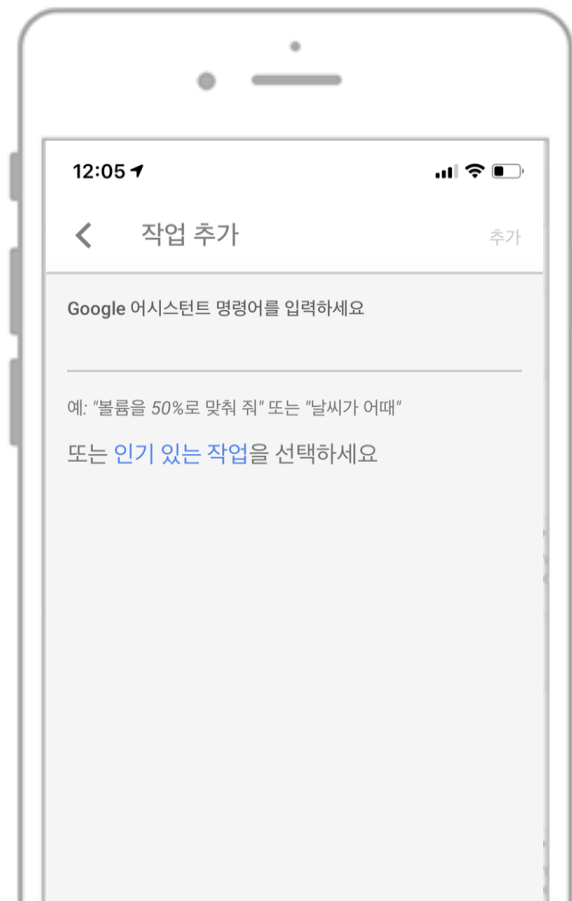
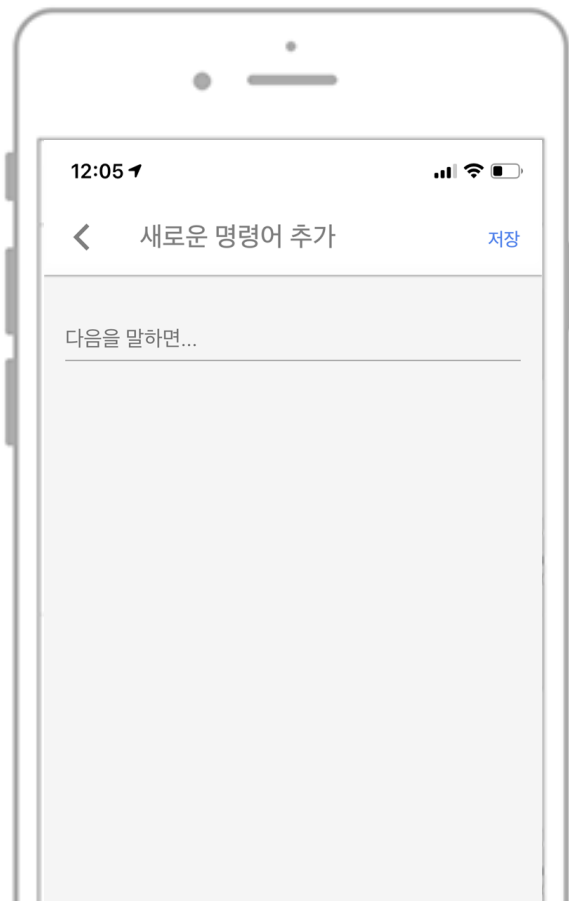
- ✓ 사용자가 직접 구글 어시스턴트의 루틴을 편집하여 필요한 기능을 설정할 수 있다.



Google Home

구글 홈의 기능

✓ 명령어, 작업, 미디어를 추가할 수 있다



구현 목표

구글 홈에 등록하고자 하는 루틴

1. 귀가 루틴

2. 외출 루틴

3. 미세먼지 확인 루틴

4. 공부 모드 루틴

5. 취침 루틴

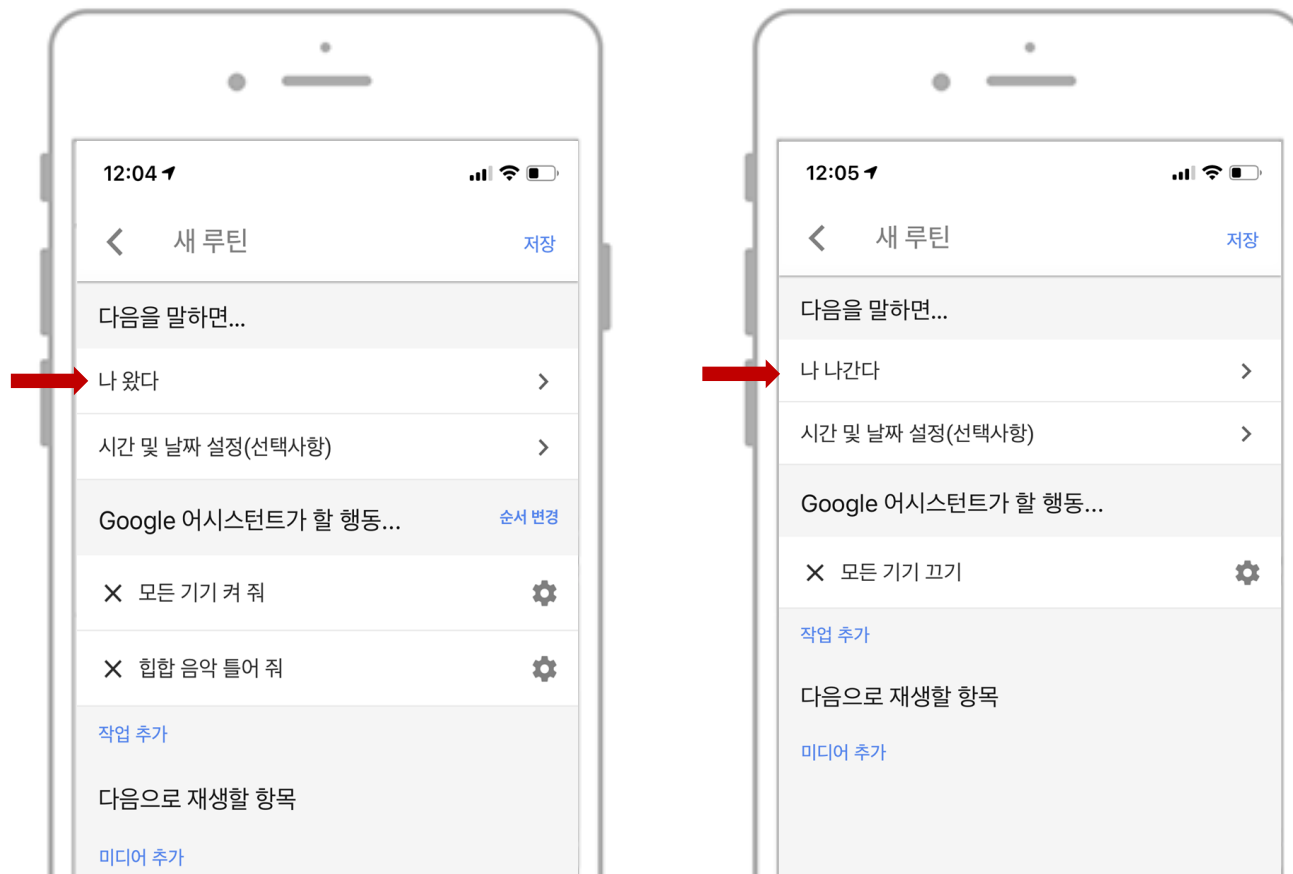
구현 목표

구글 홈에 등록하고자 하는 루틴

1. 귀가 루틴

2. 외출 루틴

✓ “나 왔다”, “나 나간다”의 음성을 입력받으면 기기의 전원을 켜거나 끄는 동작을 하는 루틴



구현 목표

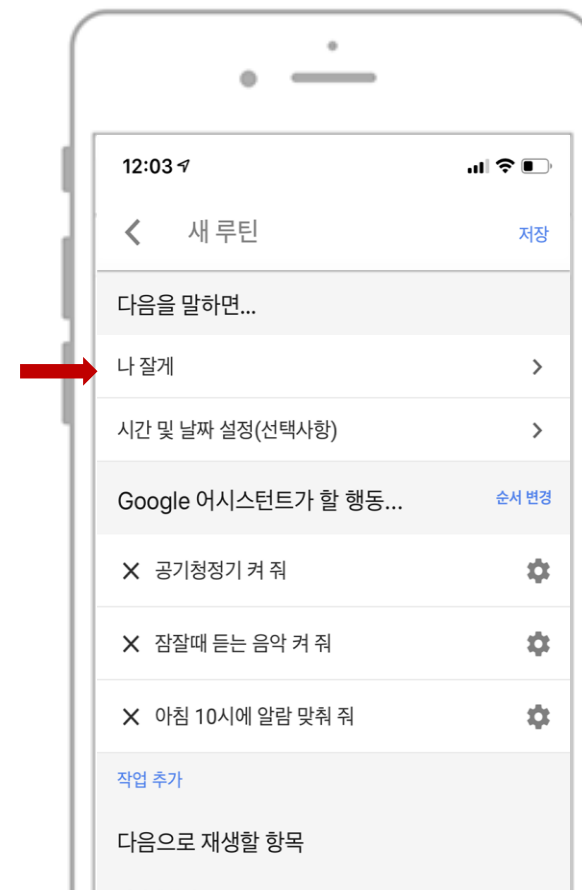
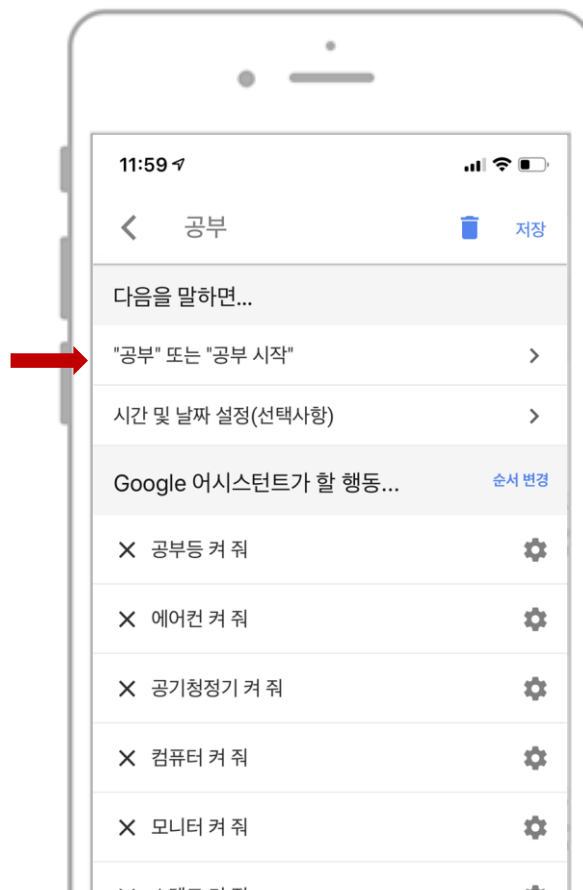
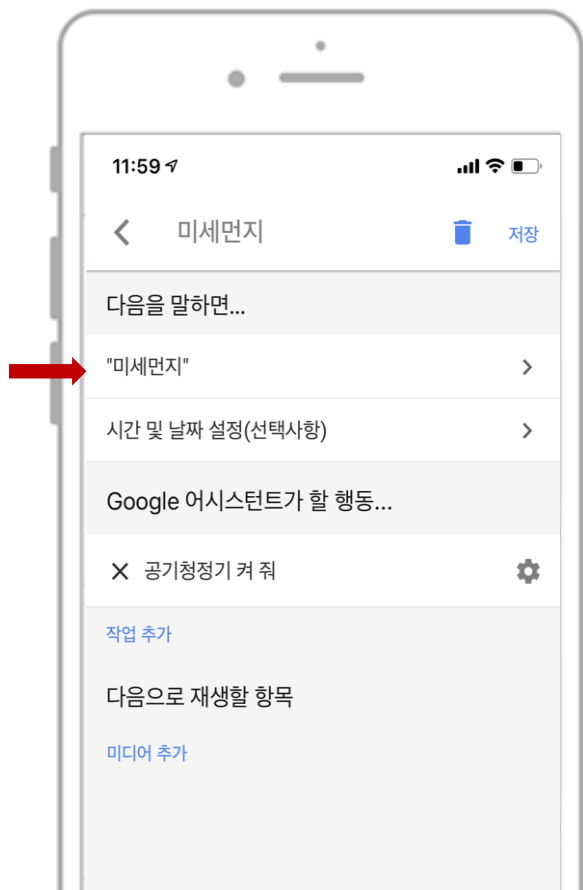
구글 홈에 등록하고자 하는 루틴

- ✓ "미세먼지", "공부", "나 잘게"라는 음성을 입력받으면
공기청정기 등의 전자기기를 켤 수 있도록 설정한 루틴

3. 미세먼지
확인 루틴

4. 공부모드
루틴

5. 취침 루틴



구현 목표

구글 홈에 등록하고자 하는 루틴

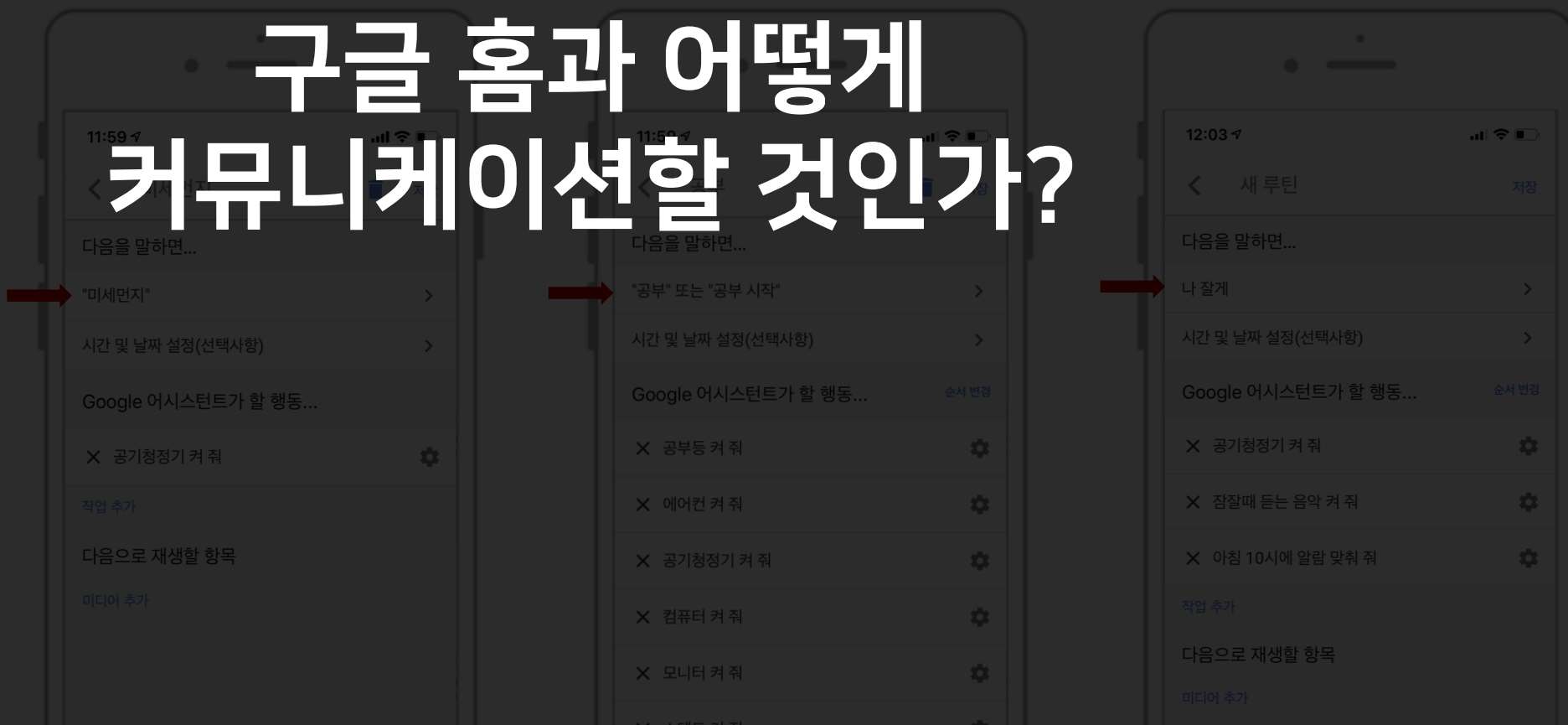
- ✓ "미세먼지", "공부", "나 잘게"라는 음성을 입력받으면 공기청정기 등의 전자기기를 켤 수 있도록 설정한 루틴

3. 미세먼지
확인 루틴

4. 공부 모드
루틴

5. 취침 루틴

구글 홈과 어떻게 커뮤니케이션할 것인가?



구현 방법

구글 홈과의 커뮤니케이션 과정

1. 구글 홈에 입력을 전달하는 과정
2. 구글 홈으로부터 목표 기기에 신호를 보내는 과정

구현 방법

구글 홈과의 커뮤니케이션 과정

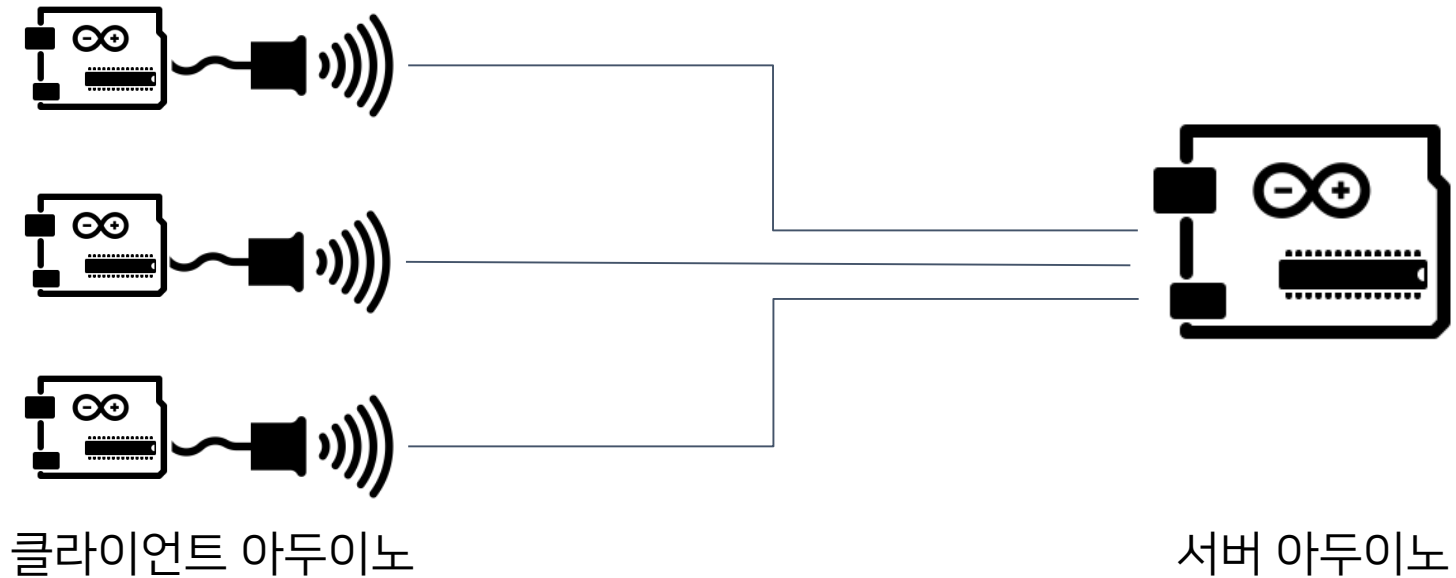
1. 구글 홈에 입력을 전달하는 과정
2. 구글 홈으로부터 목표 기기에 신호를 보내는 과정

구현 방법

구글 홈과의 커뮤니케이션 과정

1. 구글 홈에 입력을 전달하는 과정

2. 구글 홈으로부터 목표 기기에 신호를 보내는 과정

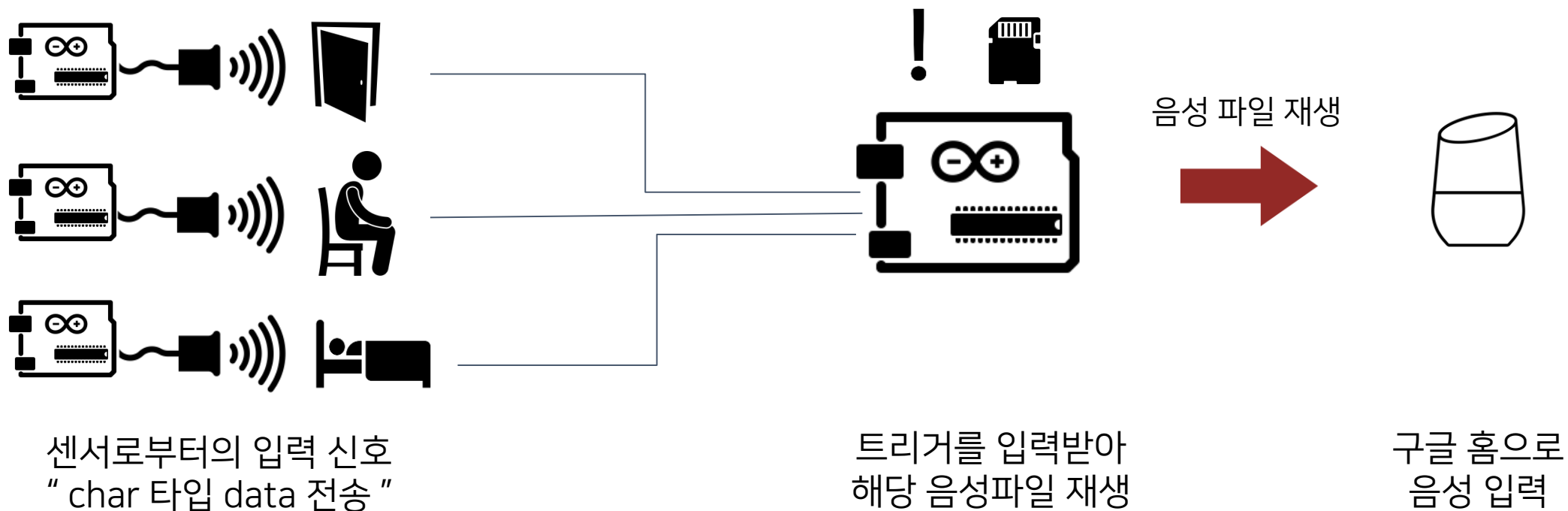


구현 방법

구글 홈과의 커뮤니케이션 과정

1. 구글 홈에 입력을 전달하는 과정

2. 구글 홈으로부터 목표 기기에 신호를 보내는 과정



구현 방법

구글 홈과의 커뮤니케이션 과정

1. 구글 홈에 입력을 전달하는 과정
2. 구글 홈으로부터 목표 기기에 신호를 보내는 과정

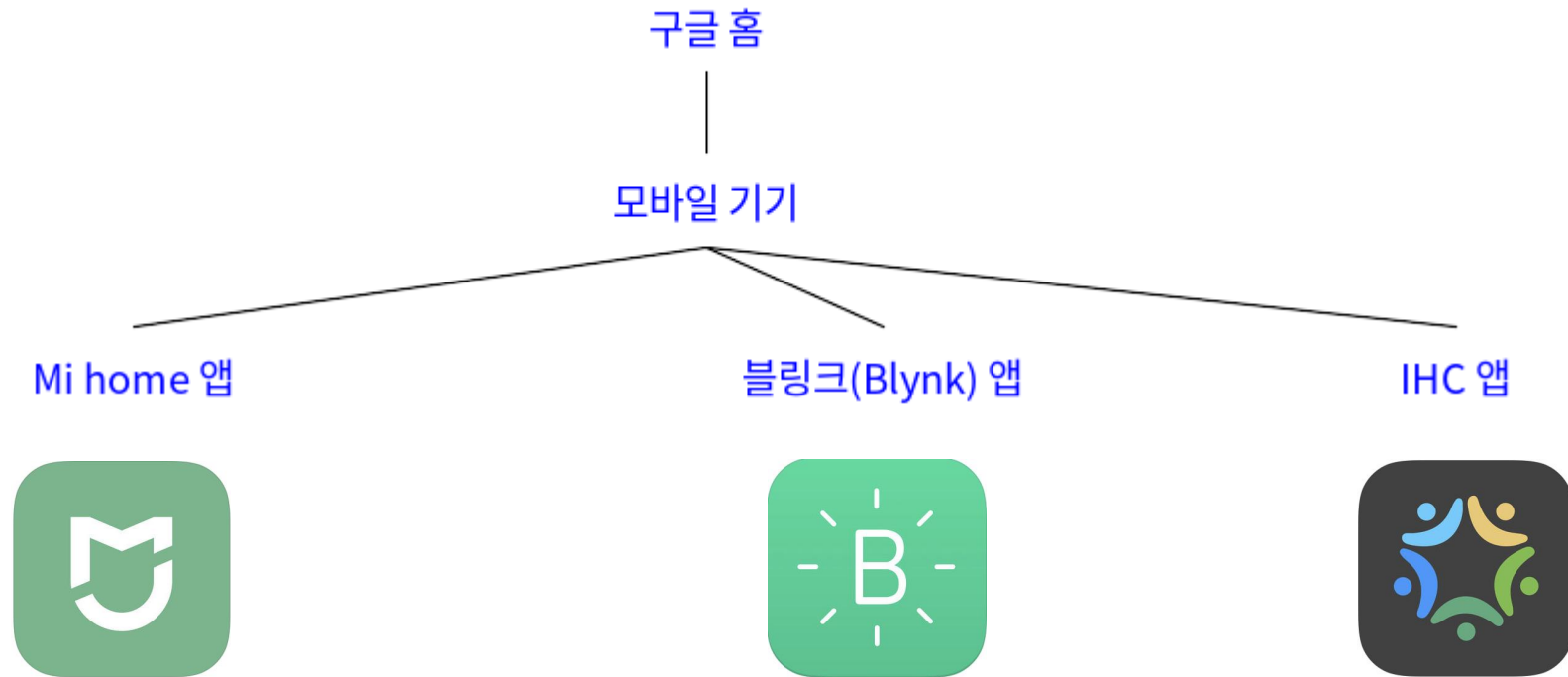
모바일 기기의 다양한 어플리케이션을 사용하면 가능!

구현 방법

구글 홈과의 커뮤니케이션 과정

1. 구글 홈에 입력을 전달하는 과정

2. 구글 홈으로부터 목표 기기에 신호를 보내는 과정

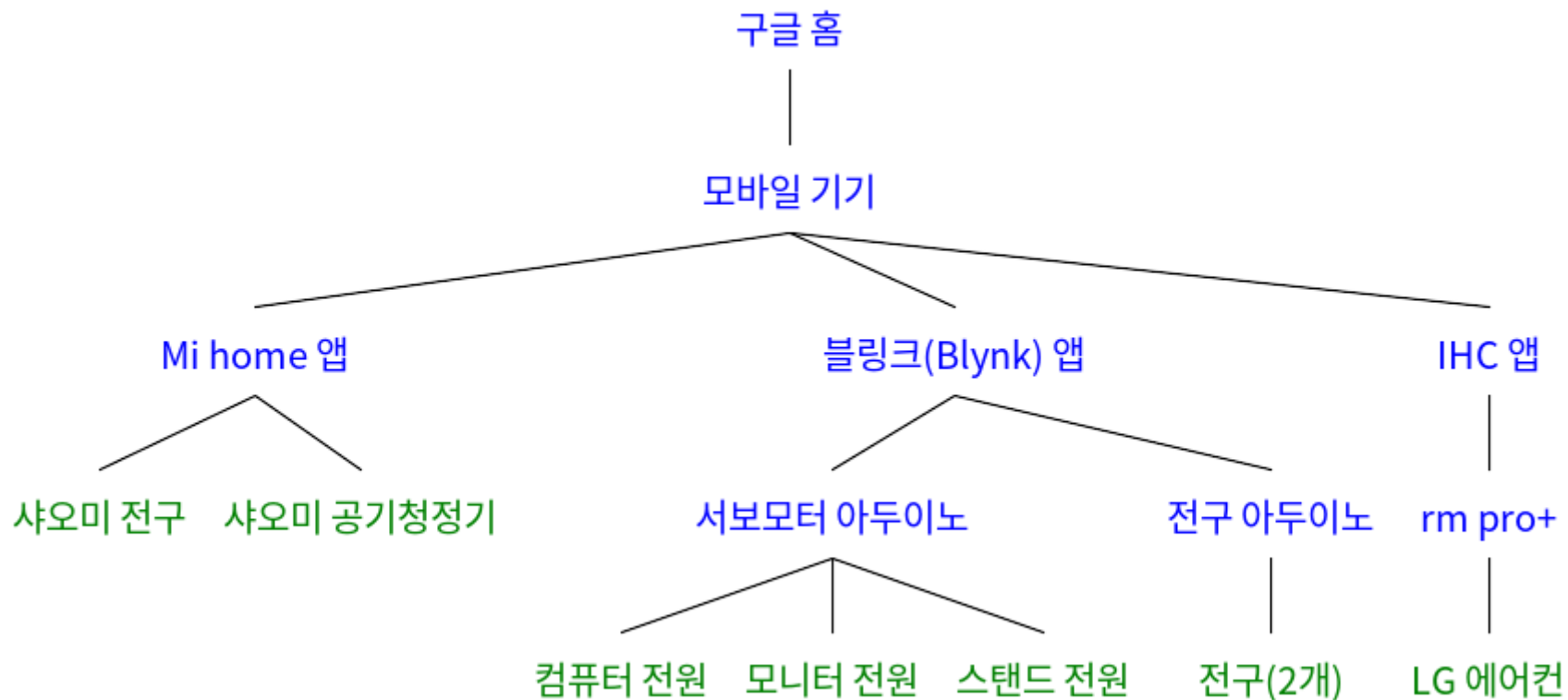


구현 방법

구글 홈과의 커뮤니케이션 과정

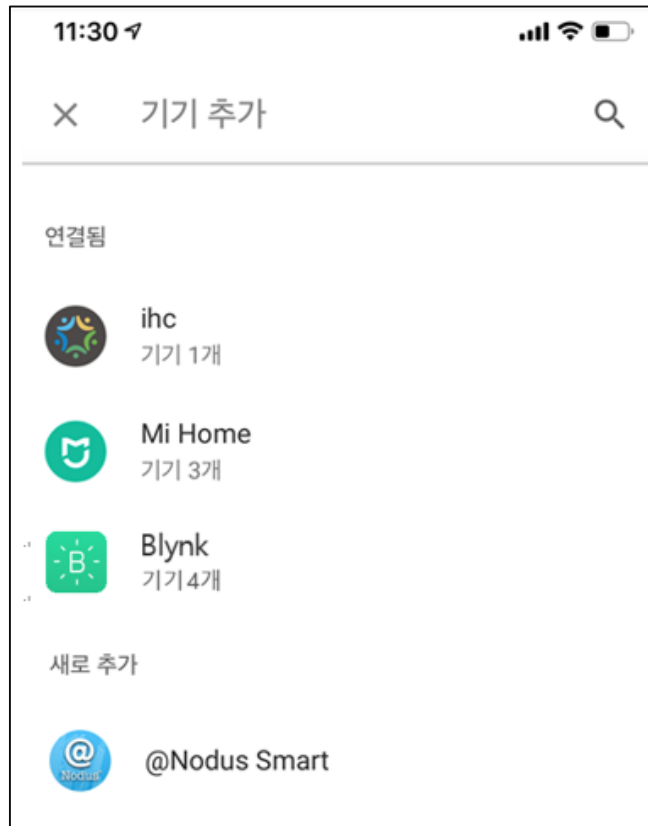
1. 구글 홈에 입력을 전달하는 과정

2. 구글 홈으로부터 목표 기기에 신호를 보내는 과정



활용 가능한 어플리케이션

- ✓ 구글 홈에 IHC, Mi Home, Blynk 연동한 화면
- ✓ 조명, 아두이노 스위치, 에어컨, 공기청정기 등을 손쉽게 제어 가능



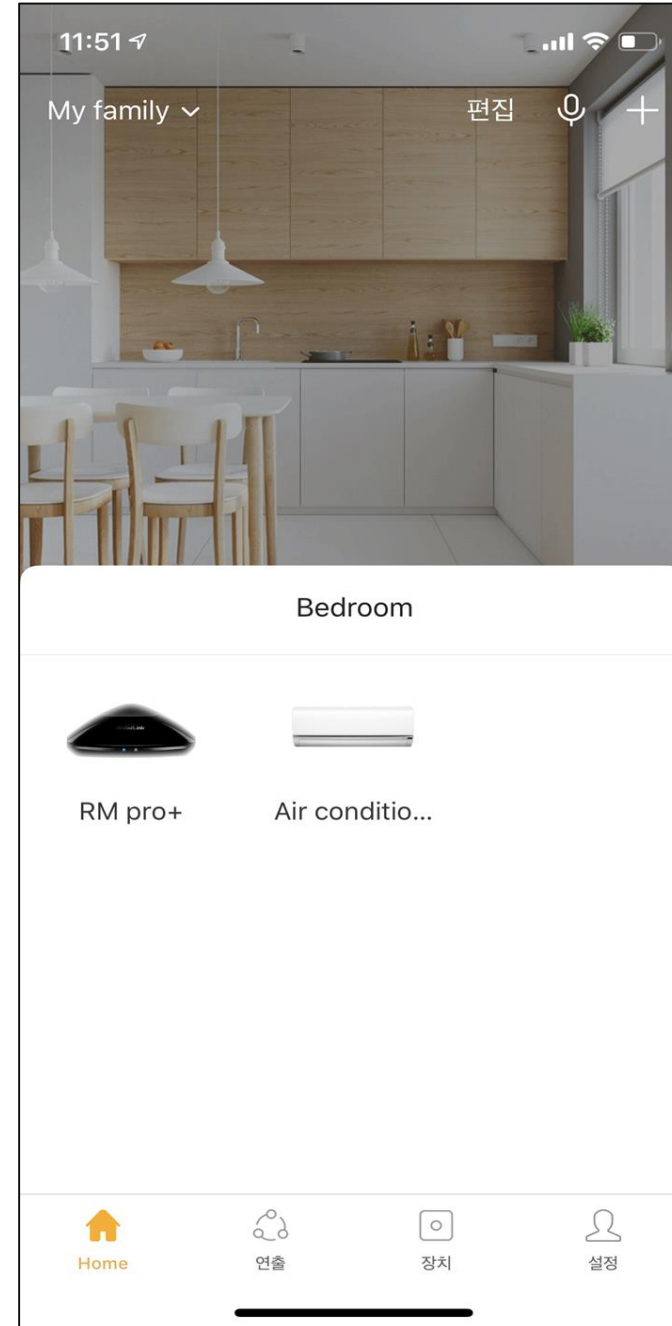
활용 가능한 어플리케이션



IHC앱을 통한 Broadlink RM pro+ 제어

IHC (Intelligent Home Center)

RM pro+의 적외선 신호를 통해
LG 에어컨과의 커뮤니케이션 가능

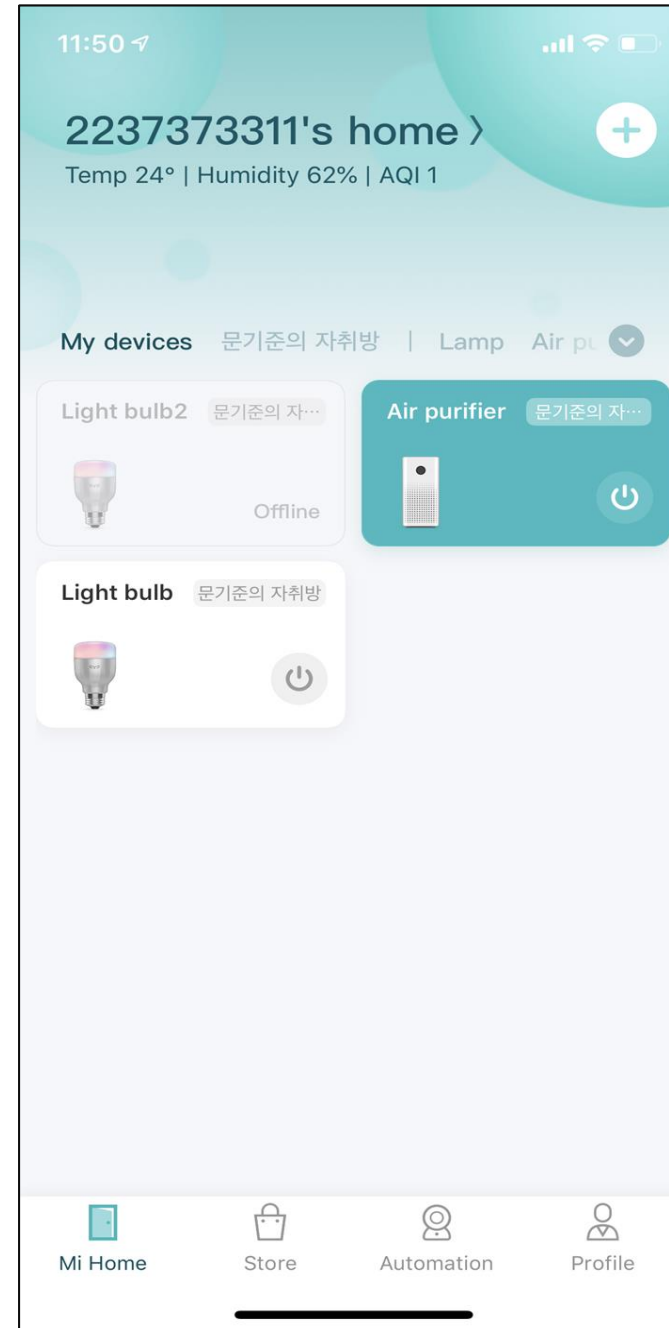


활용 가능한 어플리케이션



Mi Home

샤오미 전구, 샤오미 공기청정기 등
샤오미제품을 등록하여 쉽게 관리할
수 있도록 도와주는 어플리케이션

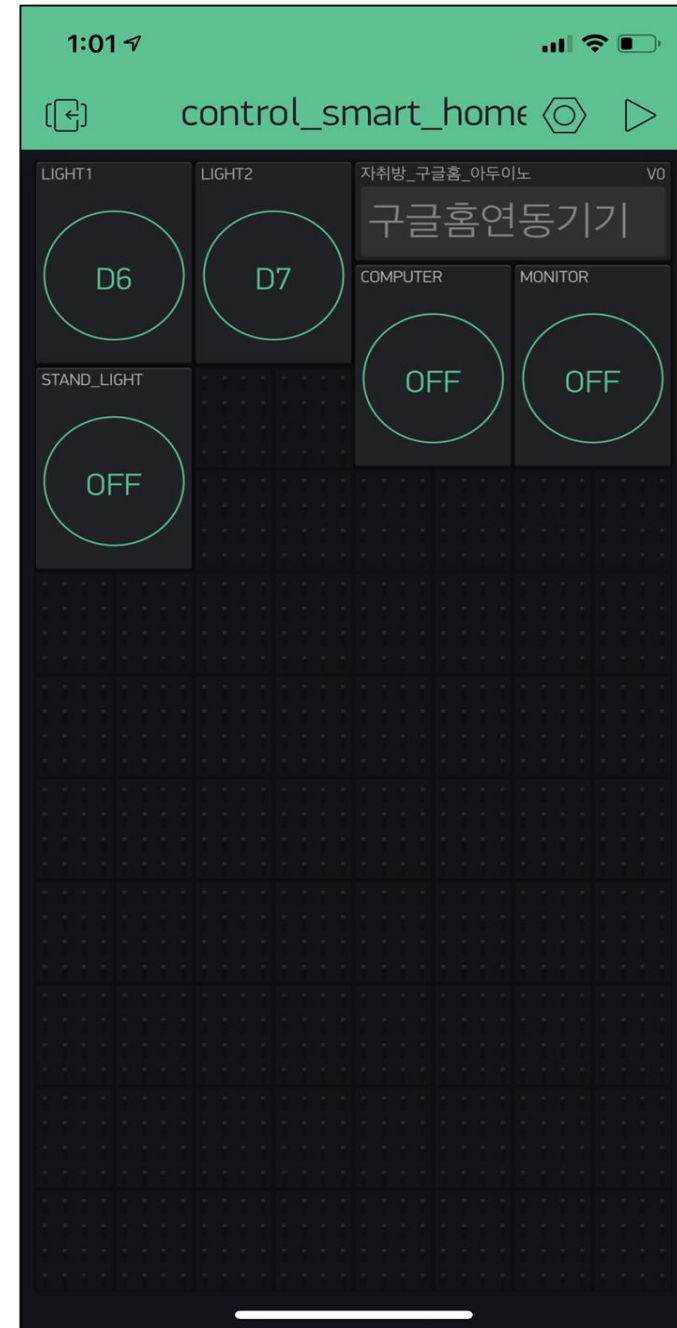


활용 가능한 어플리케이션



Blynk

구글 홈에 별도로 등록된 아두이노의 경우,
Wifi를 통해 IoT 디바이스를 등록할 수
있는 어플리케이션



활용 가능한 어플리케이션

- ✓ Blynk는 MQTT 통신을 위해 각 기기에 token 주소 부여
- ✓ Token 주소를 아두이노 스케치에 포함하여 업로드하면,
해당 아두이노가 MQTT프로토콜의 endpoint가 됨



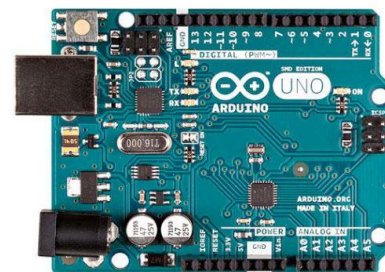
Google Home

명령어 전달



명령어 해석

MQTT 프로토콜
신호 전달



지정동작 수행





구현상세

구현목표

구글 홈에 등록하고자 하는 루틴

1. 귀가 루틴

2. 외출 루틴

3. 미세먼지 확인 루틴

4. 공부 모드 루틴

5. 취침 루틴

구현상세

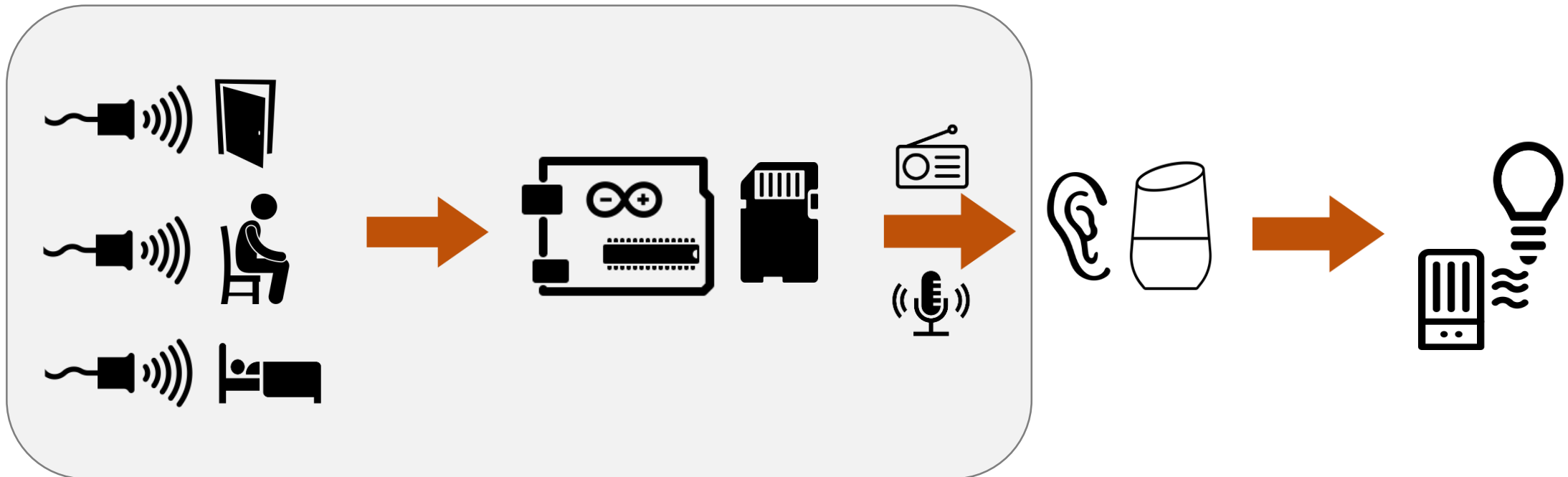
✓ 최종적으로 구글홈에 등록된 모든 기기

1. 샤오미 스마트 전구
2. 샤오미 공기 청정기
3. LG 에어컨
4. 컴퓨터 스위치
5. 모니터 스위치
6. 스탠드 스위치
7. 아두이노 전구

구현상세

1. 구글 홈으로 입력되는 아두이노 모듈 구성

- 1-1. 초음파 센서 (클라이언트)
- 1-2. 미세먼지 센서 (클라이언트)
- 1-3. 침대, 공부의자 압력 센서 (클라이언트)
- 1-4. 음성출력 아두이노 (서버)



귀가/외출 입력_초음파센서

1-1. 초음파 센서 (ultrasonic_sensor_wifi_client)

Implementation

- Ultrasonic Sensor input값 계산
- 값 변화 순서에 따라
server 아두이노에 1 또는 2 전송

** 1 : enter room, 2 : exit room

```
if(distance1 < 40 && check1 == 1 && check2 == 0){ //1번째 센서가 먼저 반응하면 exit
    exit_sig = 1;
}
if(distance2 < 40 && check1 == 0 && check2 == 1){ //2번째 센서가 먼저 반응하면 enter
    enter_sig = 1;
}

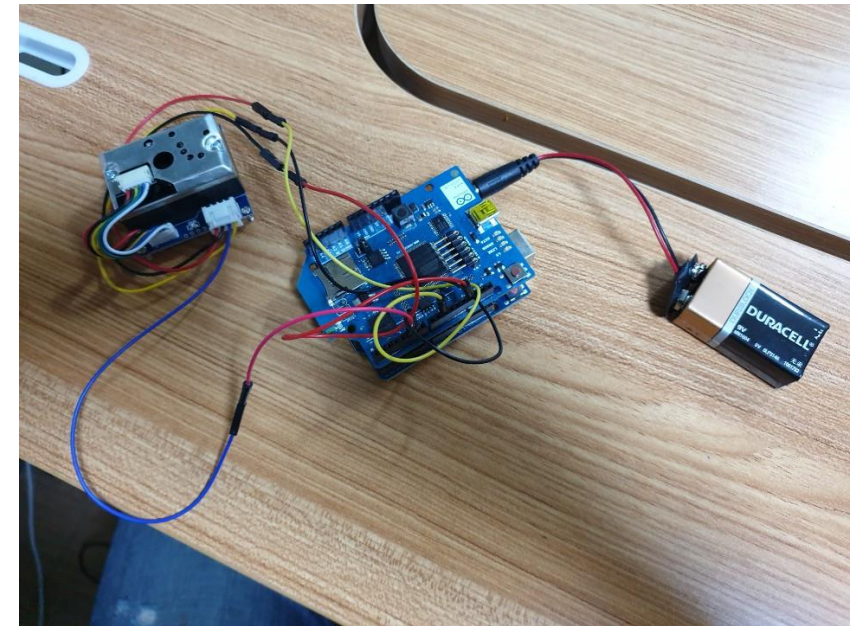
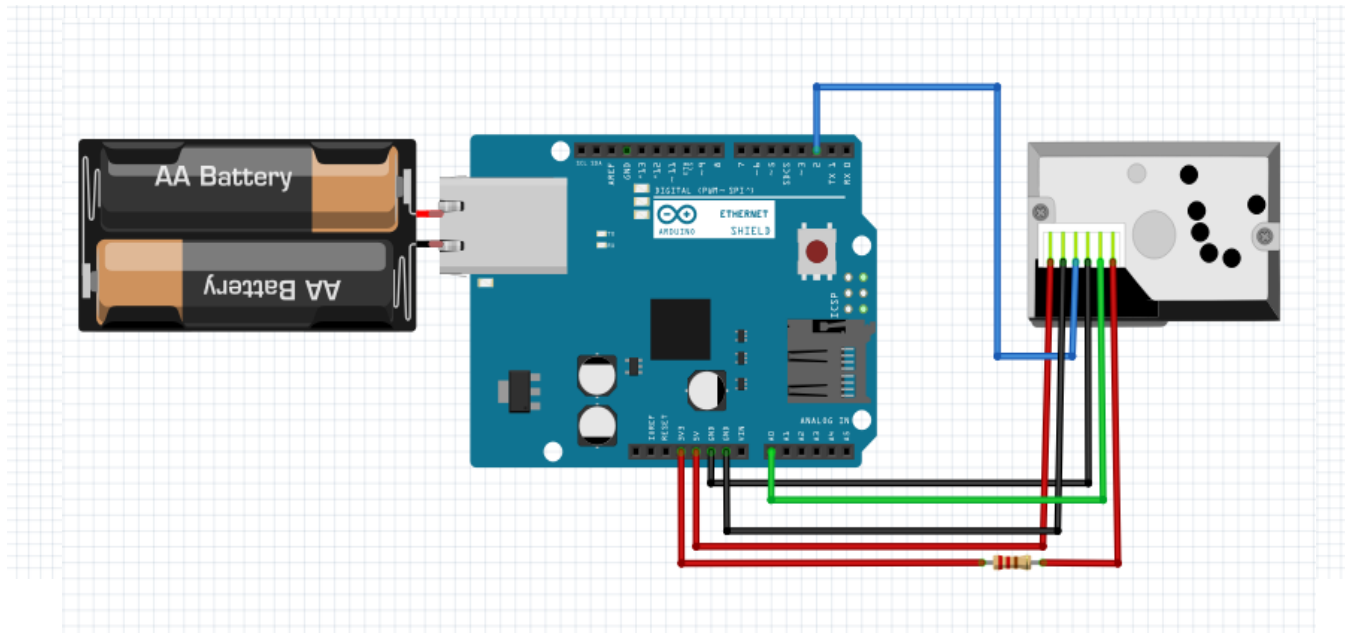
if(exit_sig){ //exit 시에 server에게 시그널 보내기
    client.connect(server, 80); // Connection to the server
    digitalWrite(ledPin, LOW); // to show the communication only (inverted logic)
    Serial.println(".");
    client.println("1\r"); // sends the message to the server when enter room
    String answer = client.readStringUntil('\r'); // receives the answer from the sever
    Serial.println("from server: " + answer);
    client.flush();
    digitalWrite(ledPin, HIGH);
    delay(2000); // client will trigger the communication after two seconds
    break;
}

if(enter_sig){ //enter 시에 server에게 시그널 보내기
    client.connect(server, 80); // Connection to the server
    digitalWrite(ledPin, LOW); // to show the communication only (inverted logic)
    Serial.println(".");
    client.println("2\r"); // sends the message to the server when exit room
    String answer = client.readStringUntil('\r'); // receives the answer from the sever
    Serial.println("from server: " + answer);
    client.flush();
    digitalWrite(ledPin, HIGH);
    delay(2000); // client will trigger the communication after two seconds
    break;
}
}
```

미세먼지 모드 입력_미세먼지 측정센서

1-2. 미세먼지 센서 (dust_sensor_wifi_client)

Target H/W - Dust sensor
- Arduino wi-fi shield



미세먼지 모드 입력_미세먼지 측정센서

1-2. 미세먼지 센서 (dust_sensor_wifi_client)

Implementation

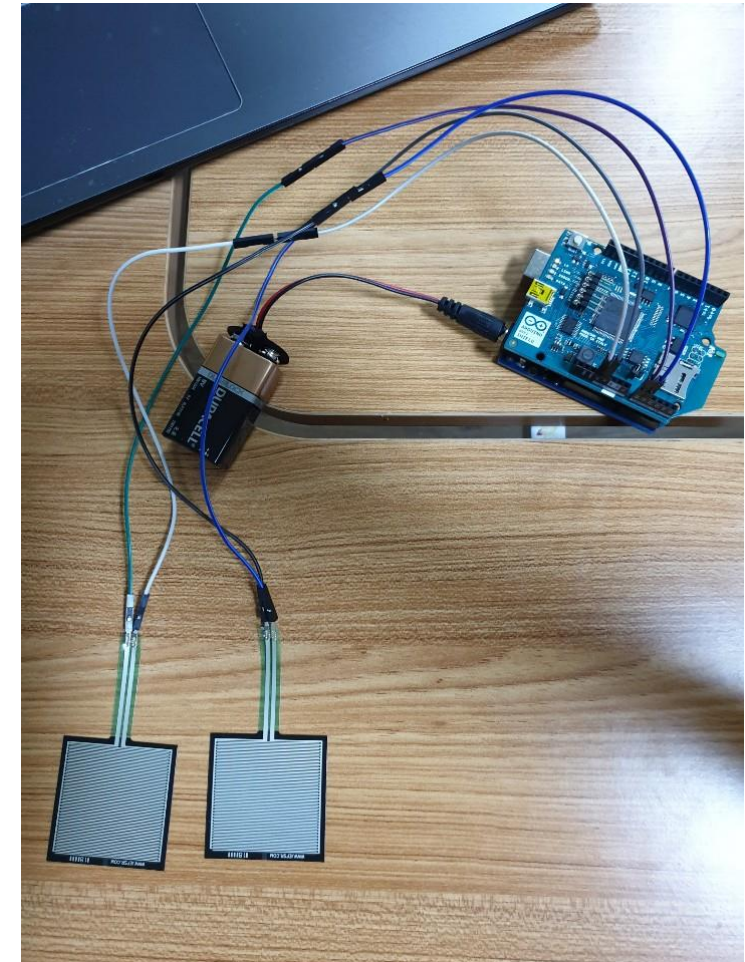
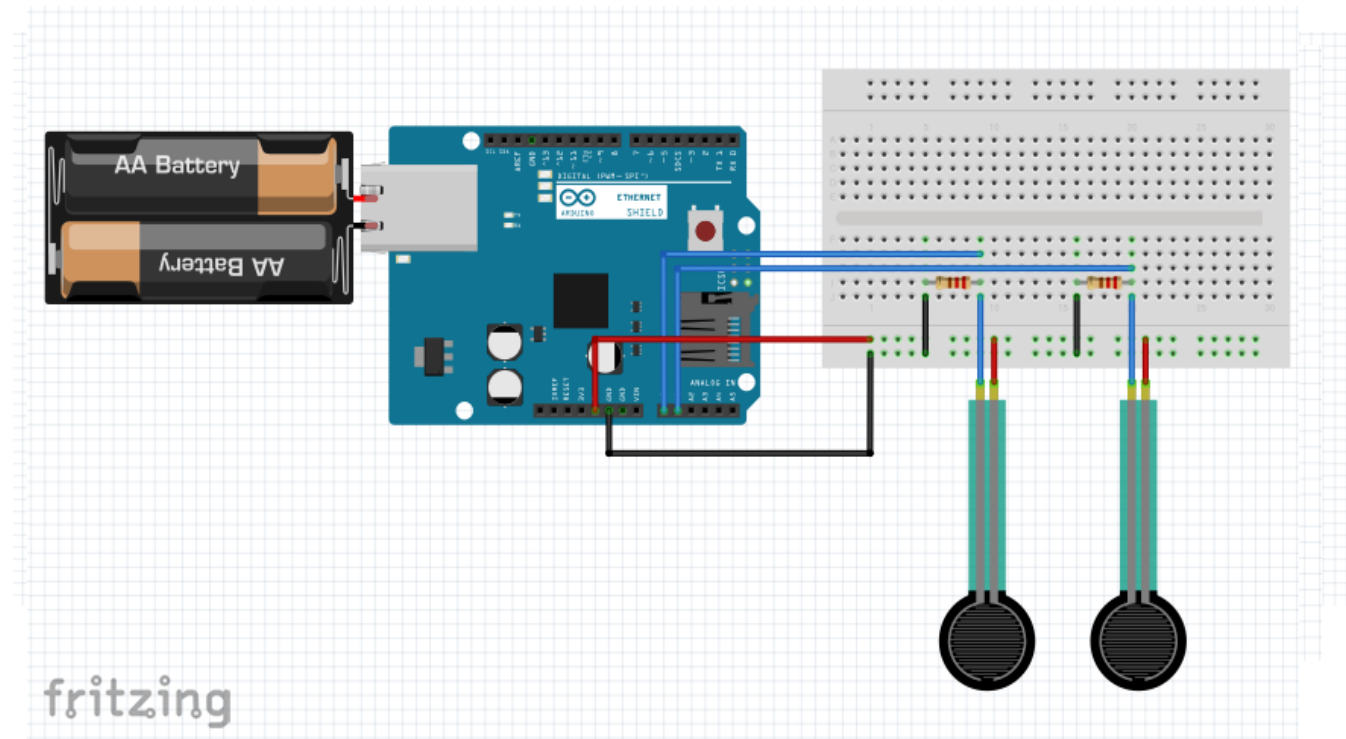
- Dust sensor input 값 계산
- dust_density 250 초과 시, server 아두이노에 3 전송

```
void loop () {  
  
    //미세먼지 센서에서 미세먼지 농도 계산  
    digitalWrite(ledPower, LOW); // power on the LED  
    delayMicroseconds(samplingTime);  
  
    voMeasured = analogRead(measurePin); // read the dust value  
  
    delayMicroseconds(deltaTime);  
    digitalWrite(ledPower, HIGH); // turn the LED off  
    delayMicroseconds(sleepTime);  
  
    // 0 - 3.3V mapped to 0 - 1023 integer values  
    // recover voltage  
    calcVoltage = voMeasured * (3.3 / 1024);  
  
    // linear equation taken from http://www.howmuchsnow.com/arduino/airquality/  
    // Chris Nafis (c) 2012  
    dustDensity = 0.17 * calcVoltage - 0.1; //최종적인 미세먼지 값 받기  
  
    if(dustDensity >= 250){ //미세먼지 농도가 250이상이면 server 아두이노에 메시지 보내기  
        client.connect(server, 80); // Connection to the server  
        digitalWrite(ledPin, LOW); // to show the communication only (inverted logic)  
        Serial.println(".");  
        client.println("3\r"); // sends the message to the server 3번 음악 파일 재생을 trigger  
        String answer = client.readStringUntil('\r'); // receives the answer from the sever  
        Serial.println("from server: " + answer);  
        client.flush();  
        digitalWrite(ledPin, HIGH);  
        delay(2000); // client will trigger the communication after two seconds  
    }  
}
```

공부/취침 모드 입력_압력센서

1-3. 압력 센서 (study/bed_mode_pressure_sensor_wifi_client)

Target H/W - Force Sensing Register(FSR)
- Arduino wi-fi shield



공부 모드 입력_압력센서

1-3. 압력 센서 (study/bed_mode_pressure_sensor_wifi_client)

Implementation

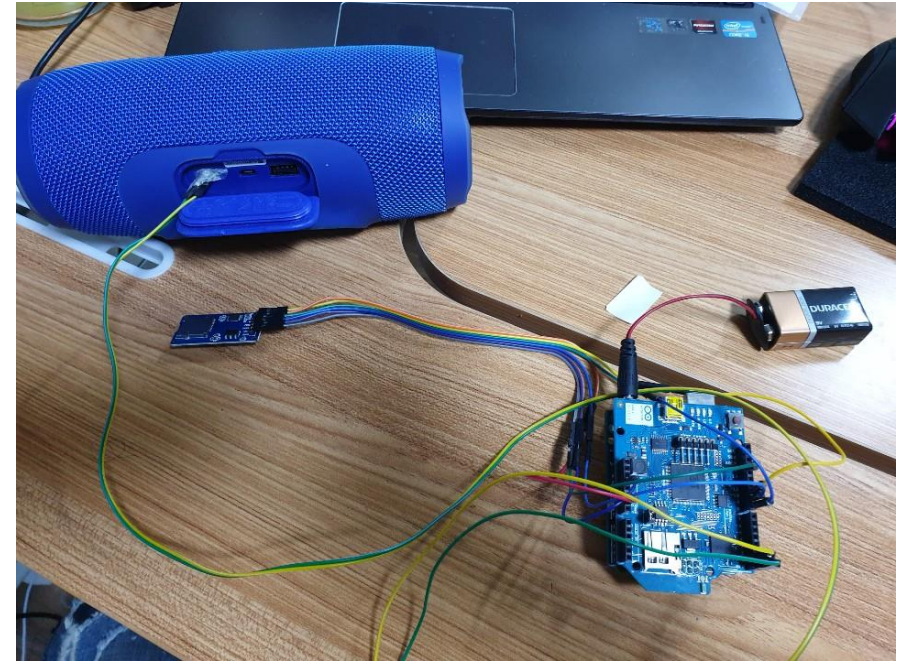
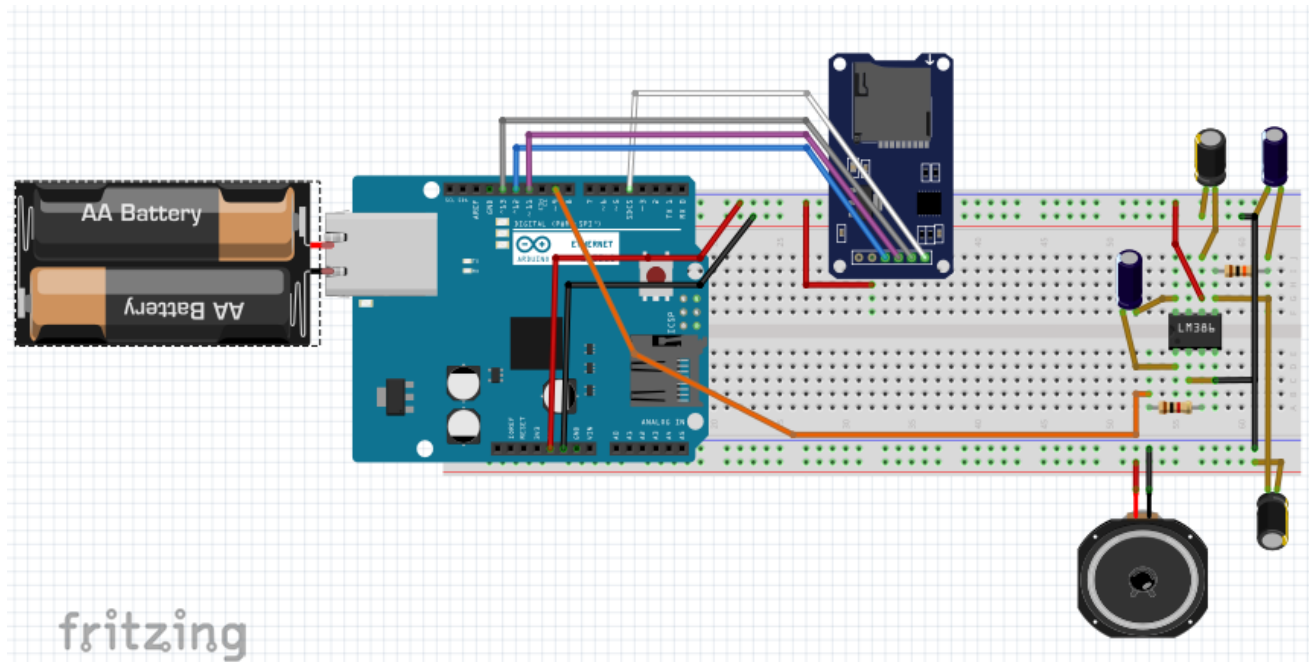
- 압력센서 input 계산
- force 값 100 초과 시,
server 아두이노에 5 전송

```
void loop () {  
  int fsrADC1 = analogRead(FSR_PIN1); //압력 센서1에서 값을 읽어옴//압력 센서에서 값을 읽어옴  
  int fsrADC2 = analogRead(FSR_PIN2); //압력 센서2에서 값을 읽어옴//압력 센서에서 값을 읽어옴  
  {  
    // Use ADC reading to calculate voltage:  
    float fsrV1 = fsrADC1 * VCC / 1023.0;  
    float fsrV2 = fsrADC2 * VCC / 1023.0;  
    // calculate FSR resistance 압력센서1:  
    float fsrR1 = R_DIV * (VCC / fsrV1 - 1.0);  
    float force1;  
    float fsrG1 = 1.0 / fsrR1; // Calculate conductance  
    // calculate FSR resistance 압력센서2:  
    float fsrR2 = R_DIV * (VCC / fsrV2 - 1.0);  
    float force2;  
    float fsrG2 = 1.0 / fsrR2; // Calculate conductance  
    if (fsrR1 <= 600) //압력센서1 최종적인 force값 계산  
      force1 = (fsrG1 - 0.00075) / 0.00000032639;  
    else  
      force1 = fsrG1 / 0.000000642857;  
    if (fsrR2 <= 600) //압력센서2 최종적인 force값 계산  
      force2 = (fsrG2 - 0.00075) / 0.00000032639;  
    else  
      force2 = fsrG2 / 0.000000642857;  
    if((force1 + force2) > 100){ //만약 사람이 가하는 압력이 총 100이 넘으면 있었다고 판단하여 server에게 메시지를 보내는 if문  
      client.connect(server, 80); // Connection to the server  
      digitalWrite(ledPin, LOW); // to show the communication only (inverted logic)  
      Serial.println(".");  
      client.println("5\r"); // sends the message to the server 5번 음악 파일 재생을 trigger//  
      String answer = client.readStringUntil('\r'); // receives the answer from the sever  
      Serial.println("from server: " + answer);  
      client.flush();  
      digitalWrite(ledPin, HIGH);  
      delay(2000); // client will trigger the communication after two seconds  
    }  
  }  
}
```


음성출력 아두이노

1-4. 음성출력 아두이노 (sd_card_audio_file_sever)

Target H/W - MicroSD adapter
- Arduino wi-fi shield



음성출력 아두이노

1-4. 음성출력 아두이노 (sd_card_audio_file_sever)

Implementation

- 센서 아두이노와 구글 홈을 통신하게 하는 중간 server 아두이노
- 클라이언트로부터 Blynk를 통해 input을 전달 받을 시,
➔ 정해진 루틴에 따라 미리 녹음된 음성 파일 재생

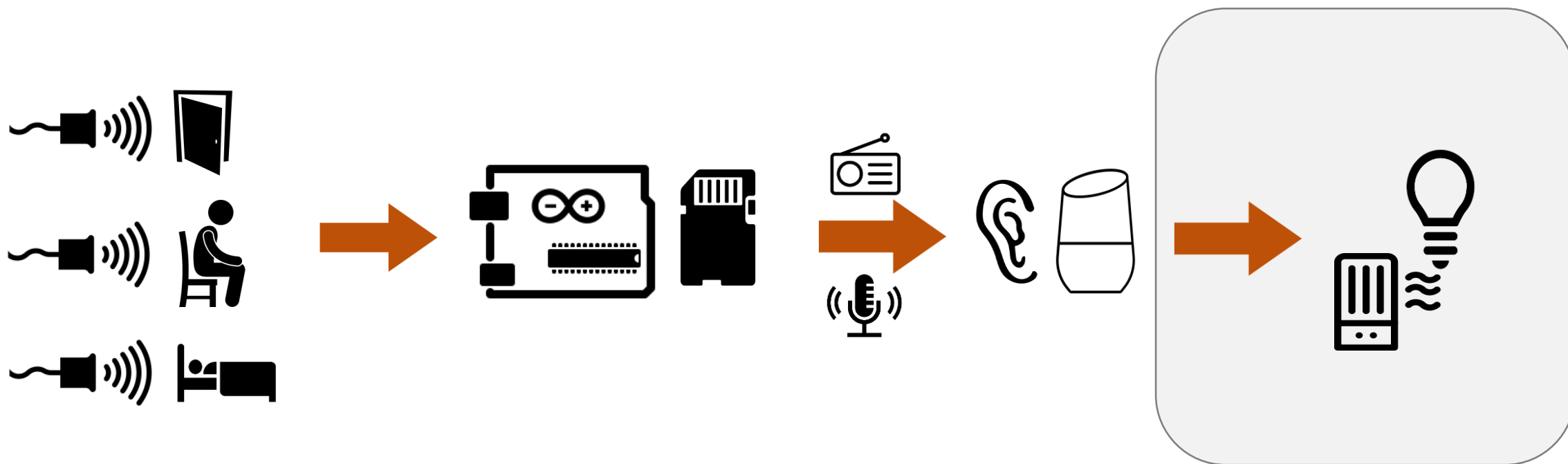
```
if ('#\r' == newChar) { //when client's data is end of a string
  // Blah blah, do whatever you want with inputs[i]
  //여기다 이제 input에 따른 동작 작성
  if(inputs[i] == "1")
    {music.play("enter_room.wav");} //Play song 1 : enter_아두이노
  if(inputs[i] == "2")
    {music.play("exit_room.wav");} //Play song 2 : exit_아두이노
  if(inputs[i] == "3")
    {music.play("dust.wav");} //Play song 3 : 미세먼지_아두이노
  if(inputs[i] == "4")
    {music.play("bed_time.wav");} //Play song 4 : 침대_압력센서_아두이노
  if(inputs[i] == "5")
    {music.play("studying_time.wav");} //Play song 5 : 공부_압력센서_아두이노
  // Empty the string for next time
  inputs[i][0] = NULL;
```

구현상세

2. 구글 홈으로 출력되는 아두이노 모듈 구성

2-1. 전구 아두이노

2-2. 서보모터 아두이노

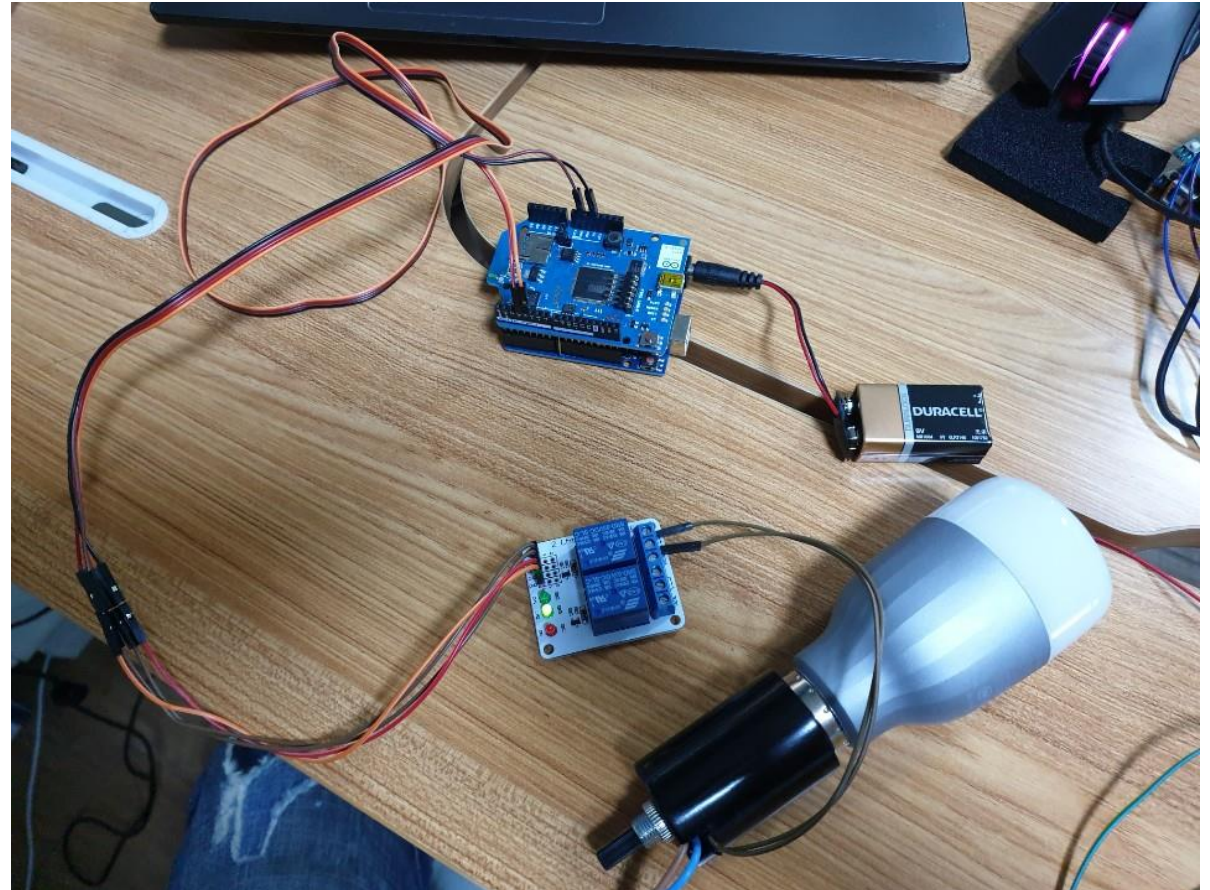
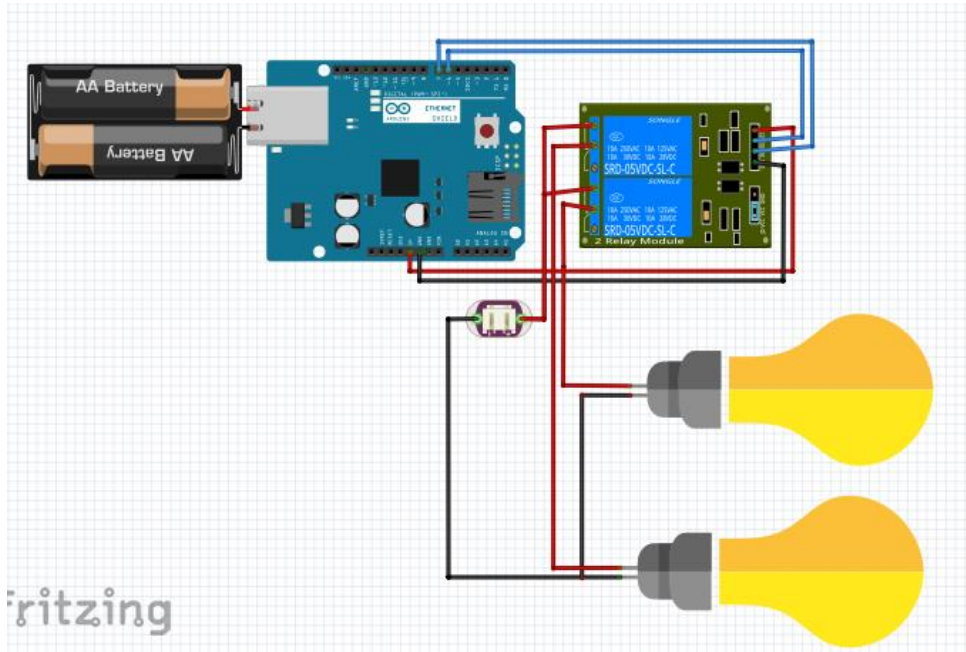


전구 아두이노

2-1. 전구 아두이노 (google_home_lightbulbs)

Target H/W - 2ch relay board

- Light bulb
- Arduino wi-fi shield



전구 아두이노

2-1. 전구 아두이노 (google_home_lightbulbs)

Implementation

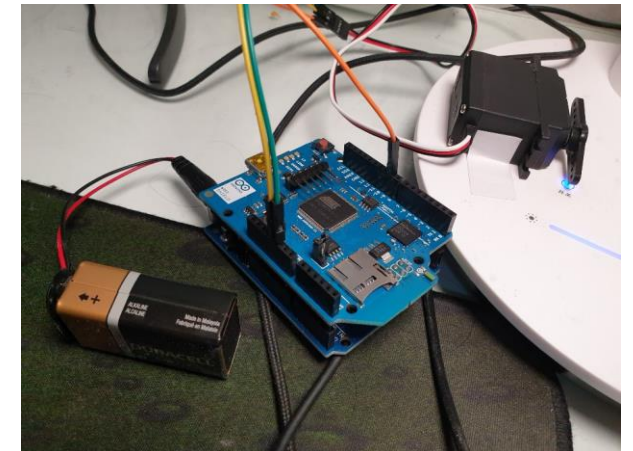
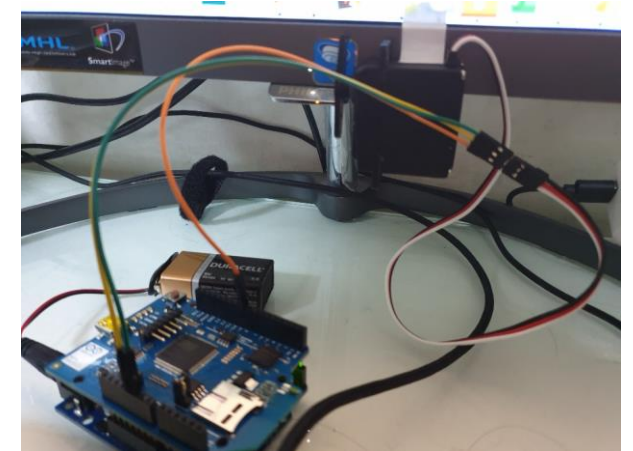
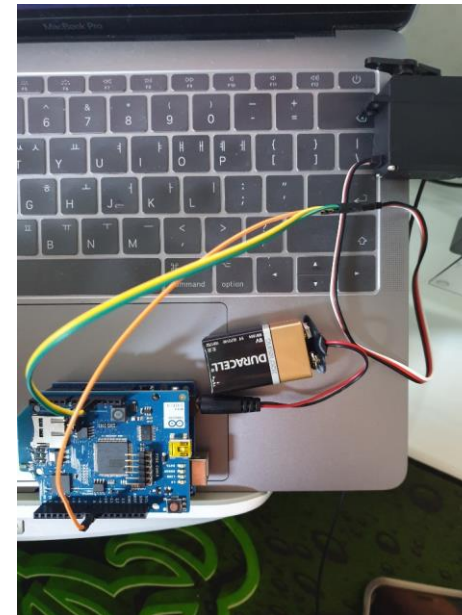
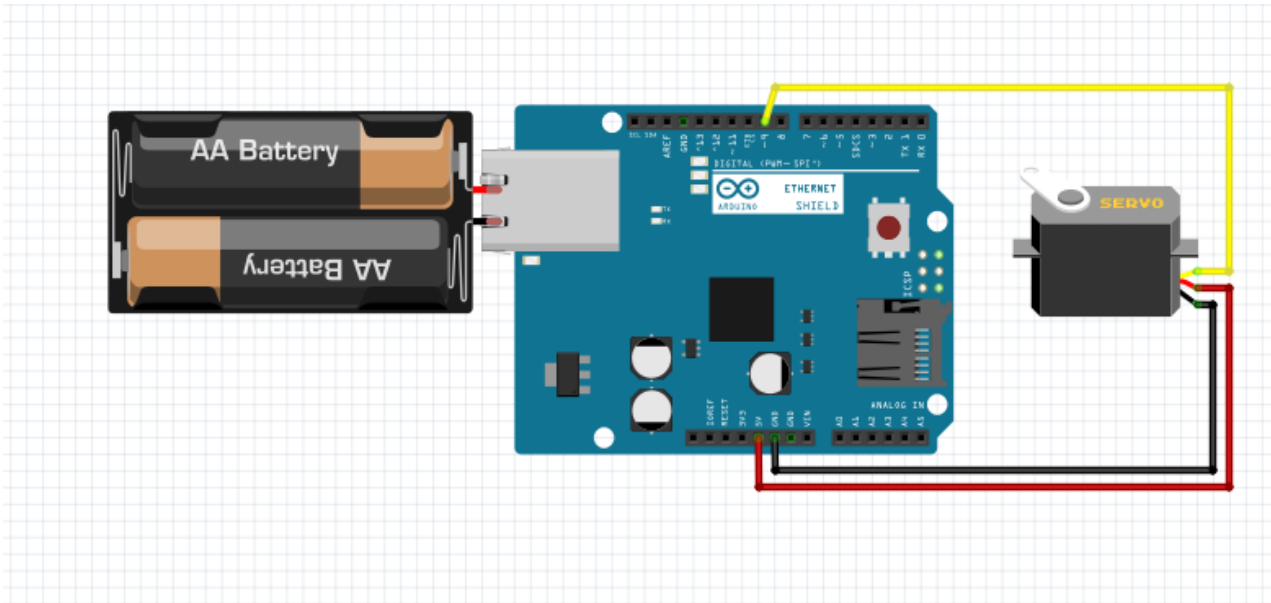
- Blynk로 부터 "on", "off"정보를 입력받고 해당 input에 따라 relay와 연결된 light bulb를 on/off
- Blynk는 구글 홈으로 부터 명령을 전달 받는다

```
BLYNK_WRITE(V0)    // it will run every time a string is sent by Blynk app Blynk어플을 통해 명령을 전달 받음
{
  s=param.asStr();
  Serial.print(s); //string sent by Blynk app will be printed on Serial Monitor
  if(s=="on")
  {
    digitalWrite(7, HIGH);    //Pin 7 has been set in setup() 전구1 켜기
    digitalWrite(6, HIGH);    //Pin 7 has been set in setup() 전구2 켜기
  }
  else if(s=="off")
  {
    digitalWrite(7, LOW);     //Pin 7 has been set in setup() 전구1 끄기
    digitalWrite(6, LOW);     //Pin 7 has been set in setup() 전구2 끄기
  }
  else{
    Serial.print("Say on or off");
  }
}
```


서보모터 아두이노

2-2. 서보모터 아두이노 (google_home_switch)

Target H/W – Servo motor Control
- Arduino wi-fi shield



**각각 모니터, 스탠드, 컴퓨터에 연결된 스위치의 모습

서보모터 아두이노

2-2. 서보모터 아두이노 (google_home_switch)

Implementation

- Blynk로부터 "on", "off" 정보를 입력받고 해당 input에 따라 servo motor가 물리적으로 switch를 on/off 실행
- Blynk는 구글 홈으로부터 명령을 전달 받는다

```
BLYNK_WRITE(V0)    // it will run every time a string is sent by Blynk app Blynk어플을 통해 명령을 전달 받음
{
  s=param.asStr();
  Serial.print(s); //string sent by Blynk app will be printed on Serial Monitor
  if(s=="on")
  {
    //서보모터를 90도까지 돌리고 돌아오는 코드
    for(angle = 0; angle < 90; angle += 1)    // command to move from 0 degrees to 180 degrees
    {
      servo_test.write(angle);                //command to rotate the servo to the specified angle
      delay(15);
    }
    delay(1000);

    for(angle = 90; angle>=1; angle--=5)    // command to move from 180 degrees to 0 degrees
    {
      servo_test.write(angle);                //command to rotate the servo to the specified angle
      delay(5);
    }
    delay(1000);
  }
  else if(s=="off")
  {
    //서보모터를 90도까지 돌리고 돌아오는 코드
    for(angle = 0; angle < 90; angle += 1)    // command to move from 0 degrees to 180 degrees
    {
      servo_test.write(angle);                //command to rotate the servo to the specified angle
      delay(15);
    }
    delay(1000);

    for(angle = 90; angle>=1; angle--=5)    // command to move from 180 degrees to 0 degrees
    {
      servo_test.write(angle);                //command to rotate the servo to the specified angle
      delay(5);
    }
    delay(1000);
  }
  else{
    Serial.print("Say on or off");
  }
}
```



실행영상

실행영상

1. 전체 구현 영상

<https://www.youtube.com/watch?v=9NORalsKSNs>

2. 방석 센서 동영상

<https://www.youtube.com/watch?v=nAqJ0uFoga8>

Thank You

임베디드시스템_아버지날보고있다면정답을알려조
문기준 안병욱 조형준 차상순 최은희