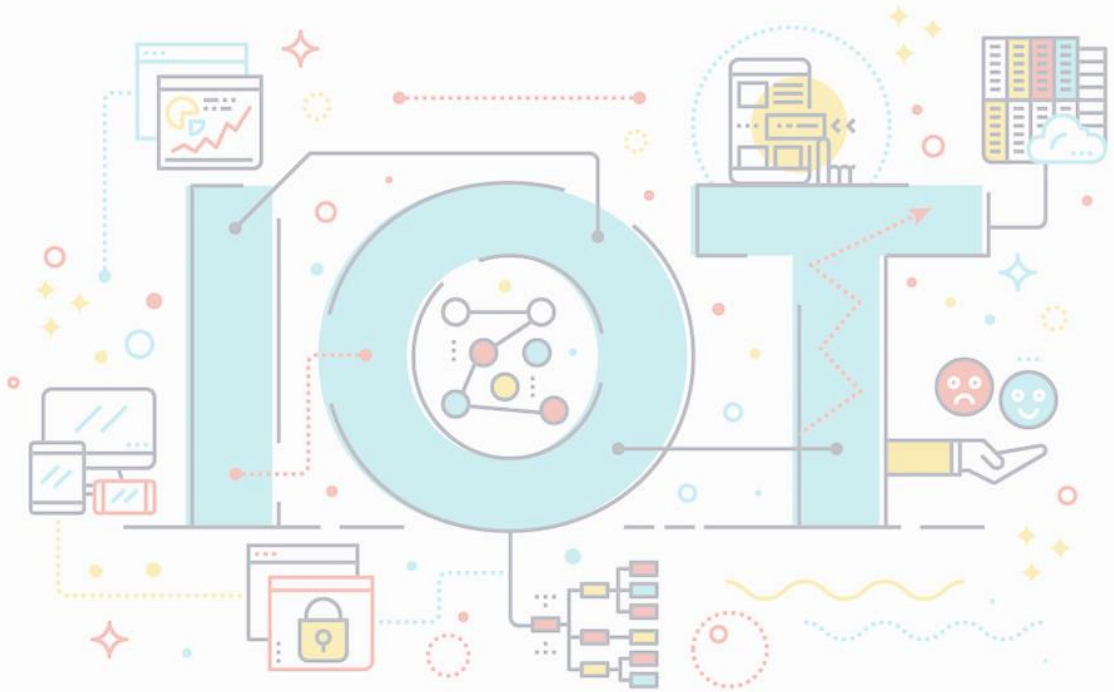
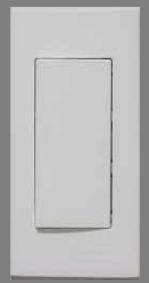



# IOT 소프트웨어 응용
# 4팀 최종 발표

## 원격 전등 제어

14109353 유호균
14109364 임정후

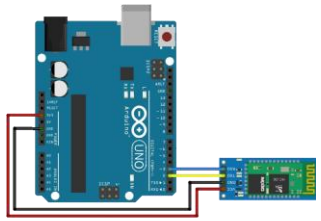
# 제작 목적



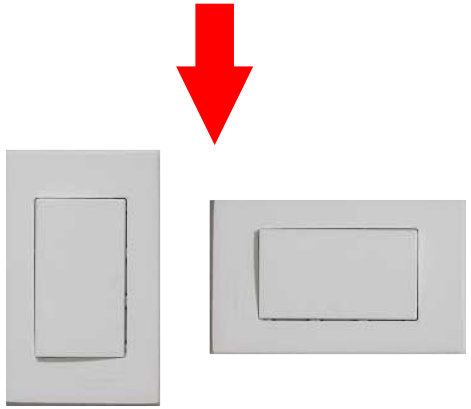
## 원격으로 전등 제어

### 아두이노 장치

서보 모터



UNO + BT모듈



**일반적인 가정용 전등 스위치에 부착해서 사용**

배선을 새로 연결하지 않아도 됨

# 기본 동작

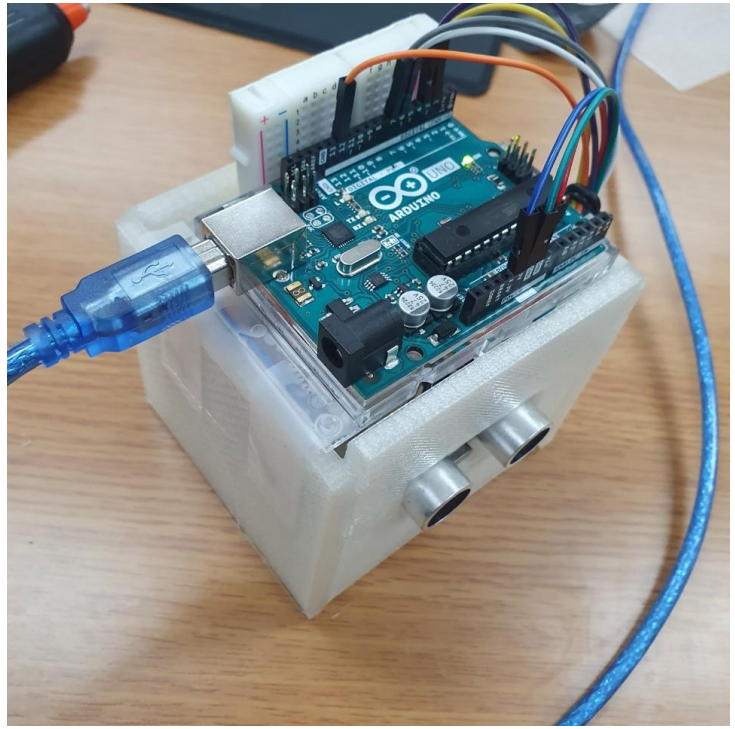
부착된 장치의 대기 상태에서
앱을 통해 신호를 보내면
모터의 회전을 통해 전등 스위치 조작

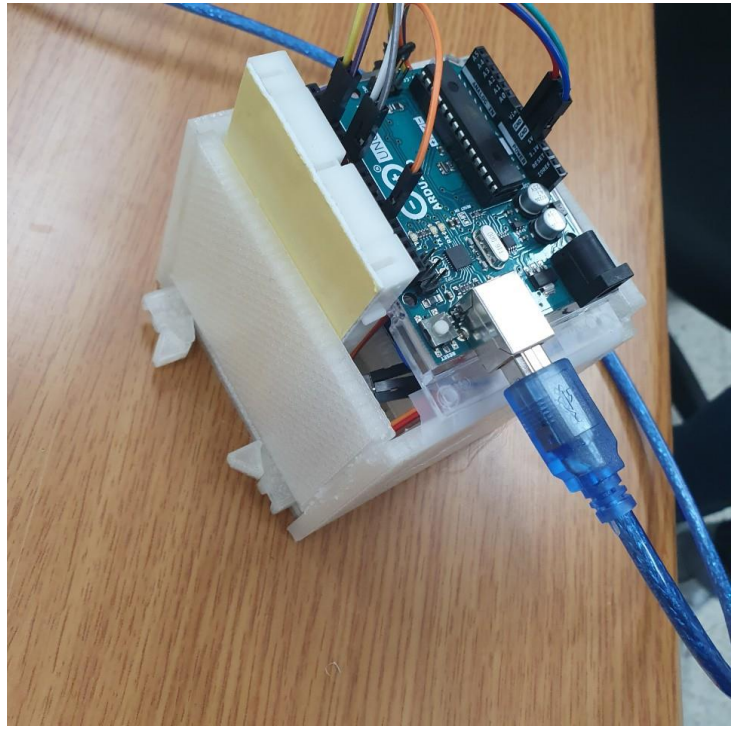
# 타이머

앱을 통해 일정 시간이 흐른 뒤
장치를 동작하게 하여
불이 켜지게 혹은 꺼지게 할 수 있음

# 직접 조작

초음파 센서 혹은 버튼을 이용해
장치를 분리하지 않고도
스위치를 조작할 수 있도록 함



# 프로젝트 기능도

스마트폰

회전량 결정
- 작은 스위치
- 중간 스위치
- 큰 스위치

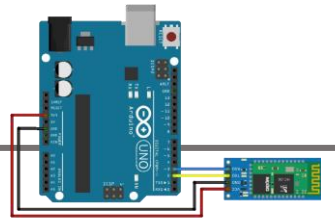
+

전등 상태 결정
- 점등
- 소등

+

타이머 설정
- 시간 설정

초음파 센서 입력



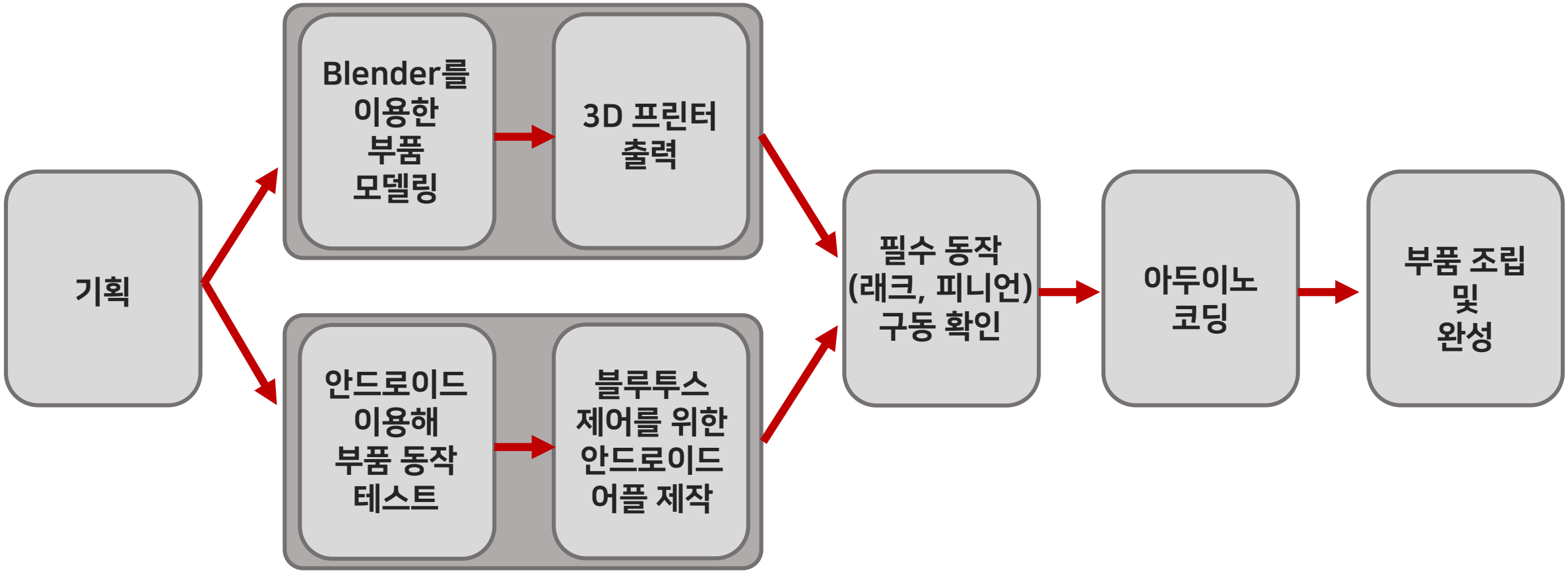
추가 입력을 받았는가?
- Yes
- No

타이머 설정 여부
- Yes
- No

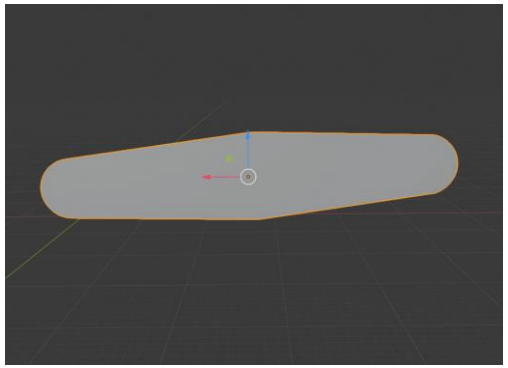
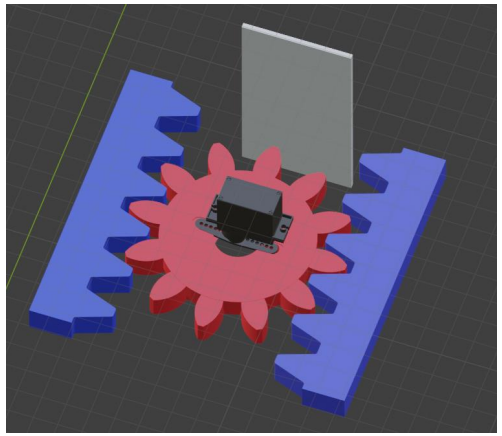
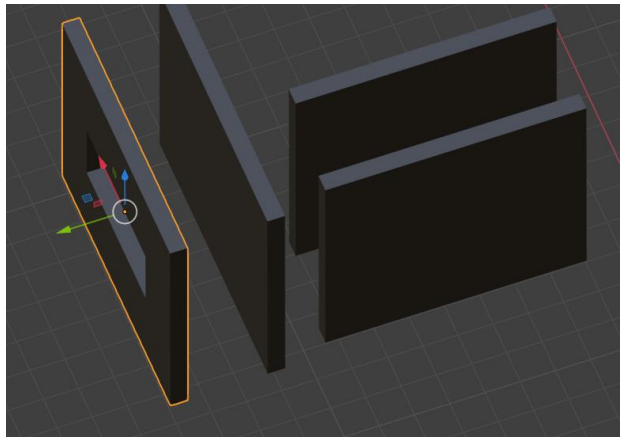
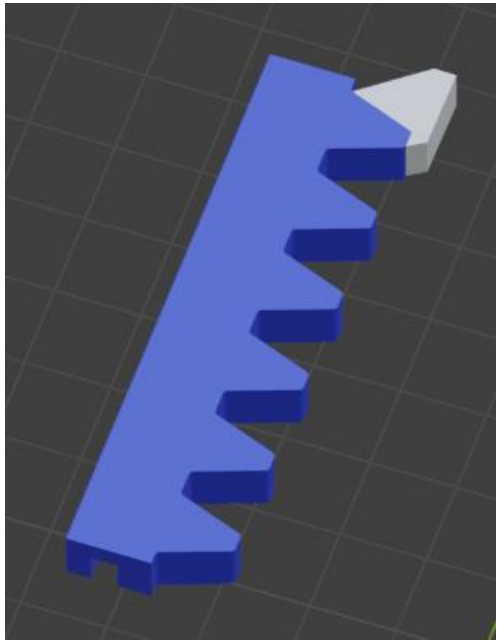
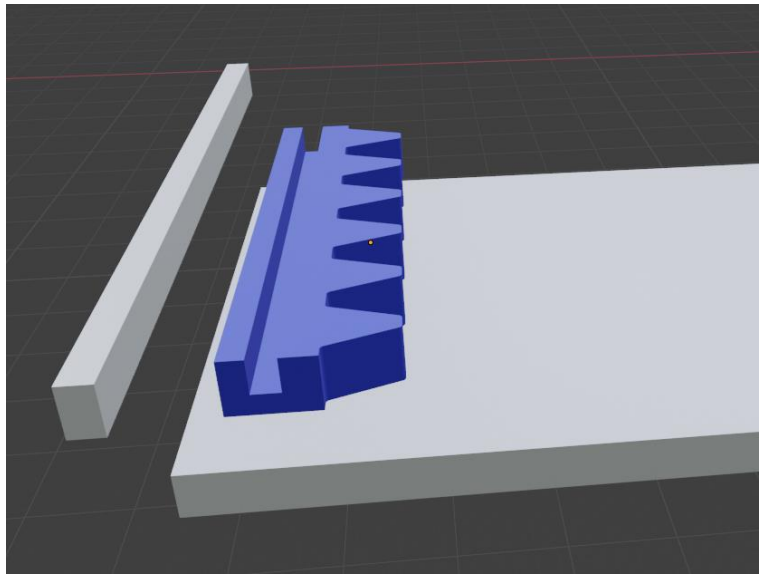
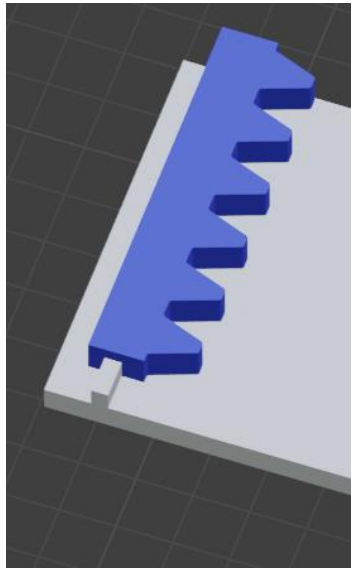
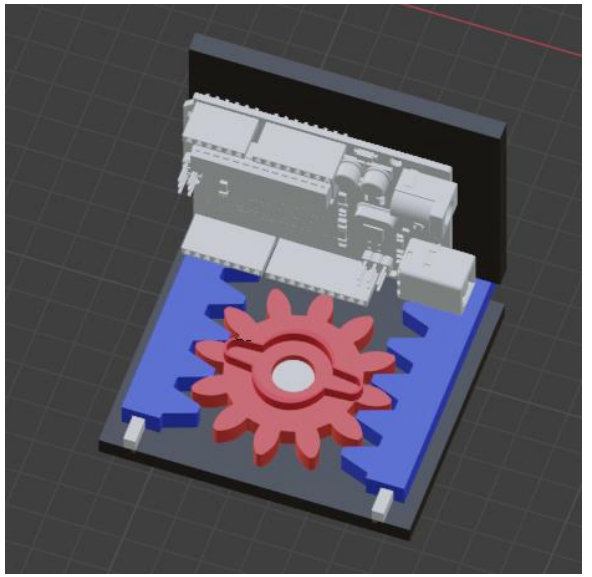
동작

Time off

# 프로젝트 설계 흐름

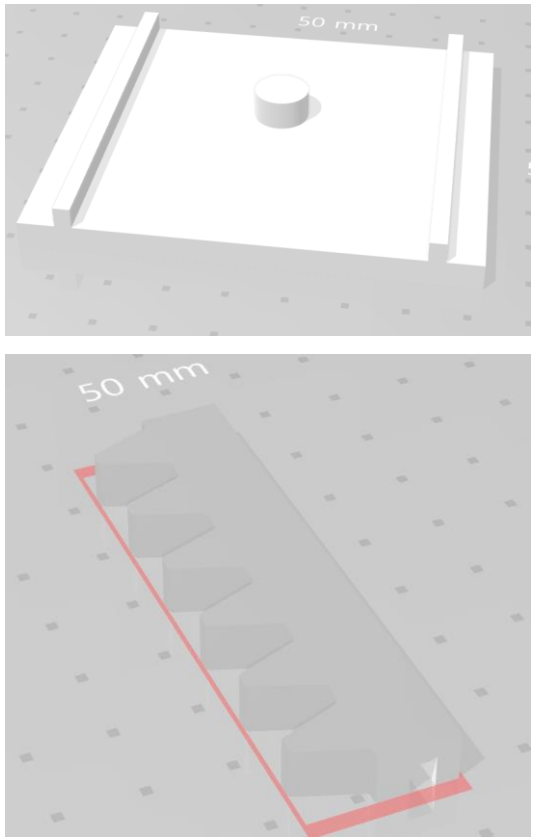

기획
Blender를 이용한 부품 모델링
3D 프린터 출력
안드로이드 이용해 부품 동작 테스트
블루투스 제어를 위한 안드로이드 어플 제작
필수 동작 (래크, 피니언) 구동 확인
아두이노 코딩
부품 조립 및 완성

# Blender를 이용한 부품 모델링

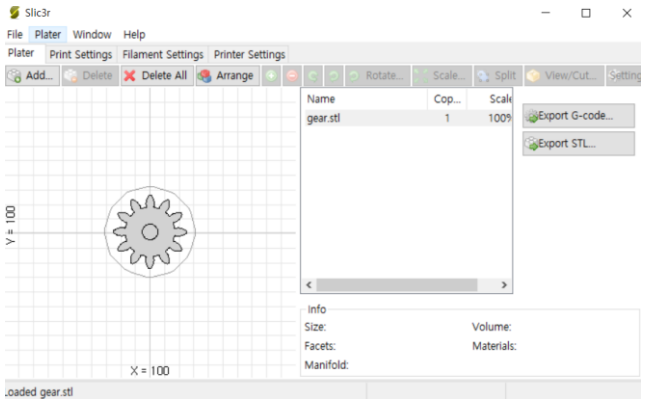
# 3D 프린터 출력

## .stl 파일로 변환

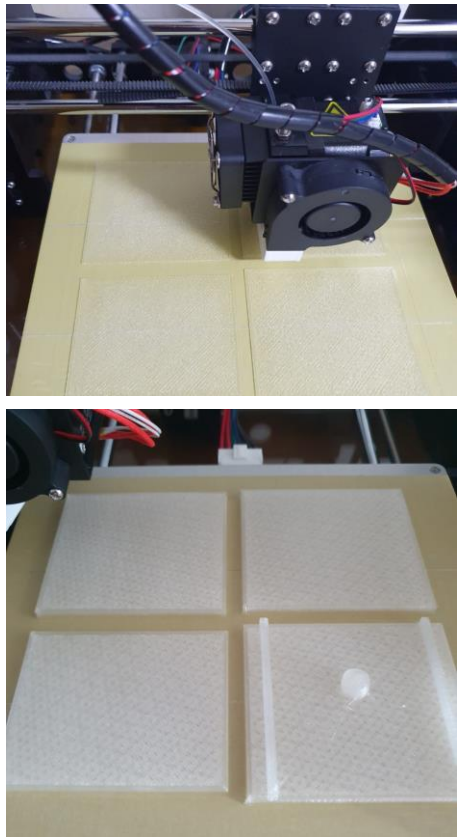



## G-code 파일로 변환

플라스틱 적층식 프린터(FDM)에서 3D 모델을 출력시키기 위해 파일 변환



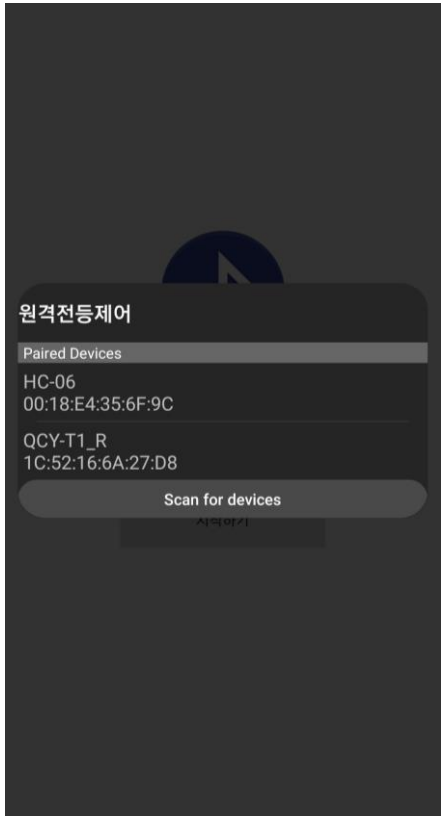

## 출력



# 안드로이드 이용한 부품 테스트 및 어플 제작

시작 화면

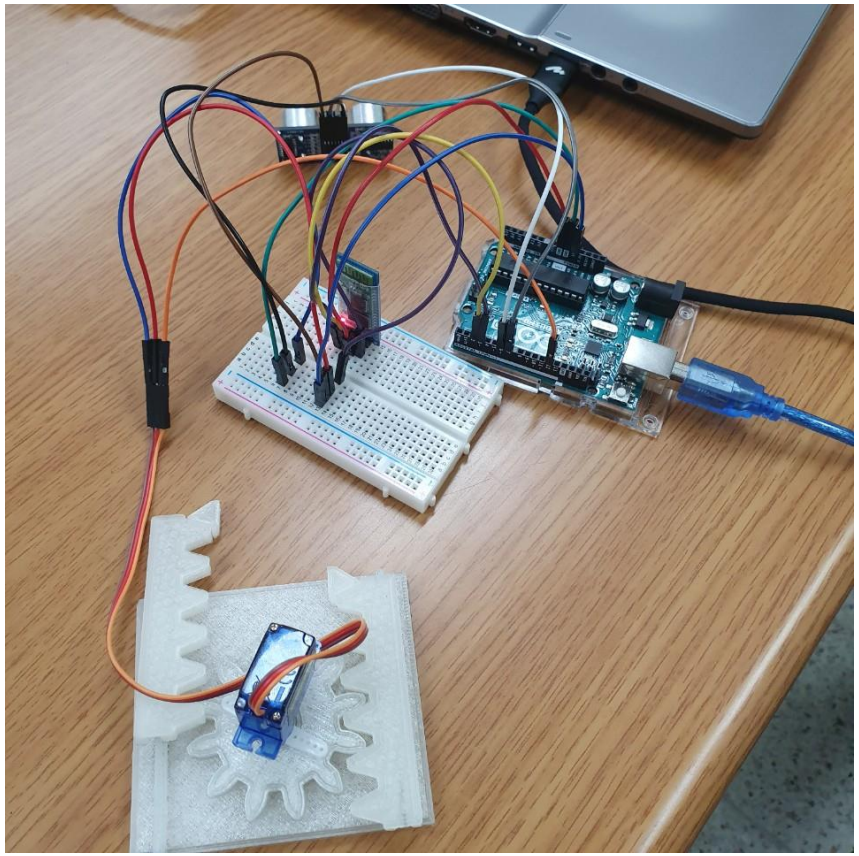

블루투스 연결




기능 화면


스위치 조절

# 필수 동작 확인(래크, 피니언)



원격 전등 제어 장치가 동작하는데 있어

가장 핵심이 되는 부분이므로

조립 전 출력된 부품들이 잘 맞물려 동작하는지 확인하고

이상이 있는 부분은 후처리를 통해 가공 수정

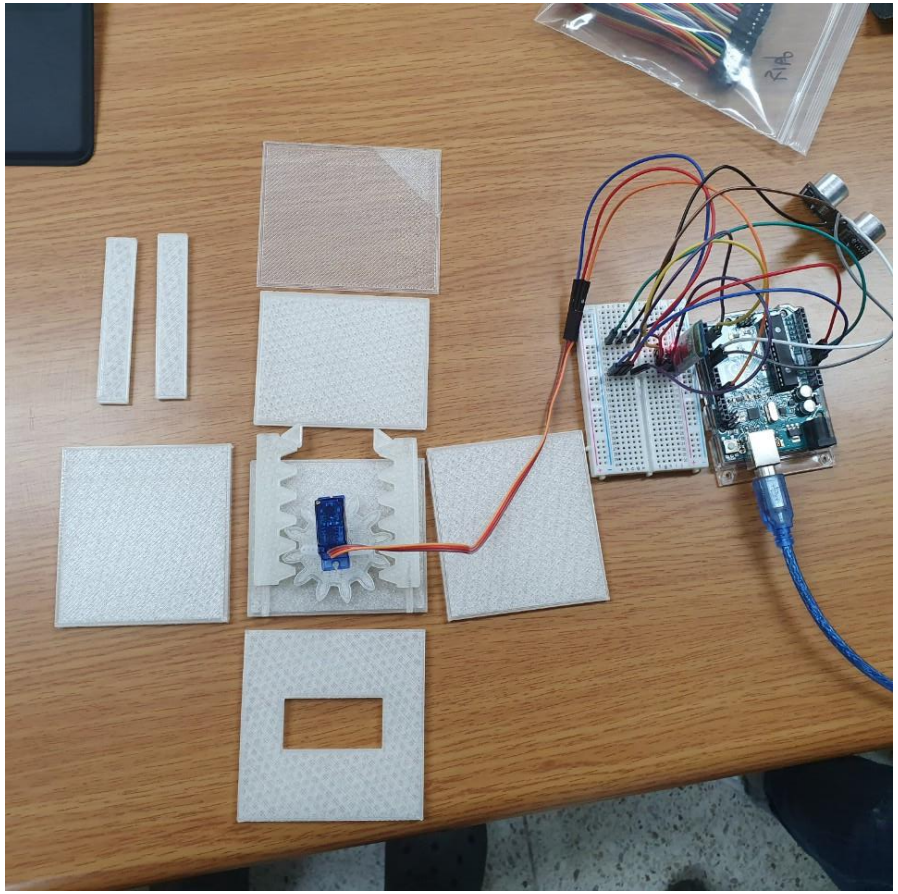
# 아두이노 코딩

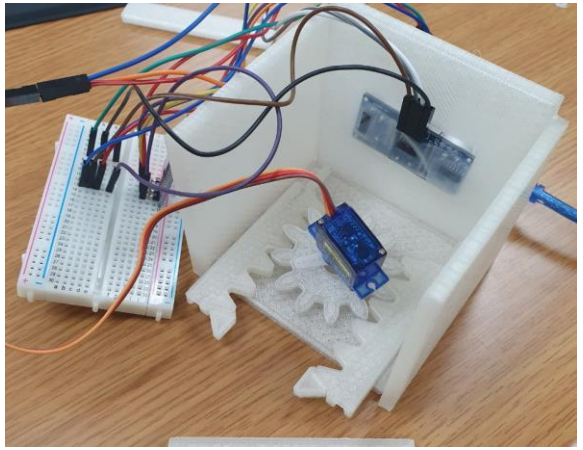
```
/*
 * myString[0] : 동작모드 결정. 0=on, 1=off, 2=setSwitchHeight,
 * myString[1] : 스위치 높이 결정. 0=short, 1=mid, 2=high
 * myString[2]...myString[n] : Time정보(분)
 */

#include <SoftwareSerial.h>
#include "Servo.h"

Servo myservo;
int blueTx = 2;  // 블투 Tx
int blueRx = 3;  // 블투 Rx
int trig = 7;
int echo = 6;

char switchSize = '1';  // 스위치 크기 초기값
boolean currentState = true; // True = on, False = off
unsigned long previousTime, currentTime;  // For Timer
int sonicFlag = 0;  // For Ultrasonic Sensor

SoftwareSerial mySerial(blueTx, blueRx);  // 시리얼 통신을 위한
String myString="";  // 받는 문자열
String temp;

void setup() {
  myservo.attach(12);  // 서보 시그널 핀설정
  myservo.write(66);  // 서보 초기각도 설정
  Serial.begin(9600);
  mySerial.begin(9600);  // 블루투스 시리얼 개방
  pinMode(trig, OUTPUT);  // 초음파 송신부
  pinMode(echo, INPUT);  // 초음파 수신부
}

void loop() {
  digitalWrite(trig, LOW);
  digitalWrite(echo, LOW);
  delayMicroseconds(2);
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);

  unsigned long duration = pulseIn(echo, HIGH);
  float distance = duration / 29.0 / 2.0;
  if(distance <= 5 && sonicFlag == 0)  // 초음파 센서 핸들링
  {
    Serial.println(currentState);
    myString = "3";
```

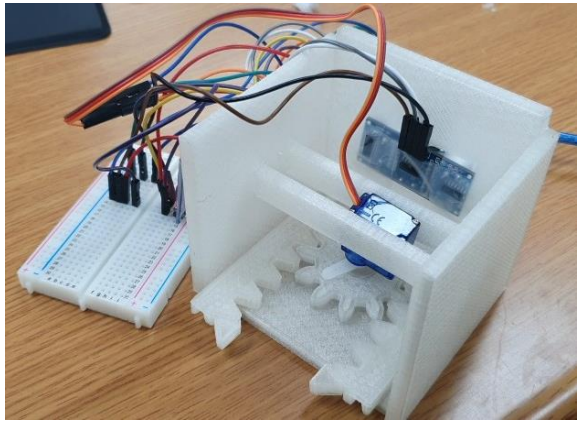
```
while(mySerial.available())  //mySerial 값이 있으면
{
  char myChar = (char)mySerial.read();  // mySerial int형식의 값을 c
  myString += myChar;   // 수신되는 문자열을 myString에 모두 붙임 (1바
  delay(5);           // 수신 문자열 끊김 방지
  if(myString[0] == "2")
  {
    switchSize = myString[1];
    myString = "";
  }
  previousTime = millis();  // Timer Set
}

if(!myString.equals(""))  // myString 값이 있다면
{
//  Serial.println("input value: " + myString); // 시리얼모니터에 my

  if(myString.length() <= 2)  // 단순 of,off 및 초음파 센서 동작
  {
    if(myString[0] == '0')  // myString 값이 '0' 이라면
    switchOn(myString[1]);
    else if(myString[0] == '1')
      switchOff(myString[1]);
    else if(myString[0] == '3')
    {
      if(currentState == 1)  // 현재상태에서 toggle 시킴. 스위치 off
        switchOff(switchSize);

      else if(currentState == 0)  // 스위치 on
        switchOn(switchSize);

    }
      myString = "";  // myString 변수 초기화
  }
  else if(myString.length() > 2)  // 타이머 동작 기능
  {
    String myStringTemp = myString;
    temp = myString;
    myString = "";  // myString 변수 초기화
    temp.remove(0,2);
    currentTime = temp.toInt();  // 숫자로 변환
    currentTime = currentTime*1000;
    Serial.println(currentTime);
    Serial.println(previousTime);

    while(1)
    {
```

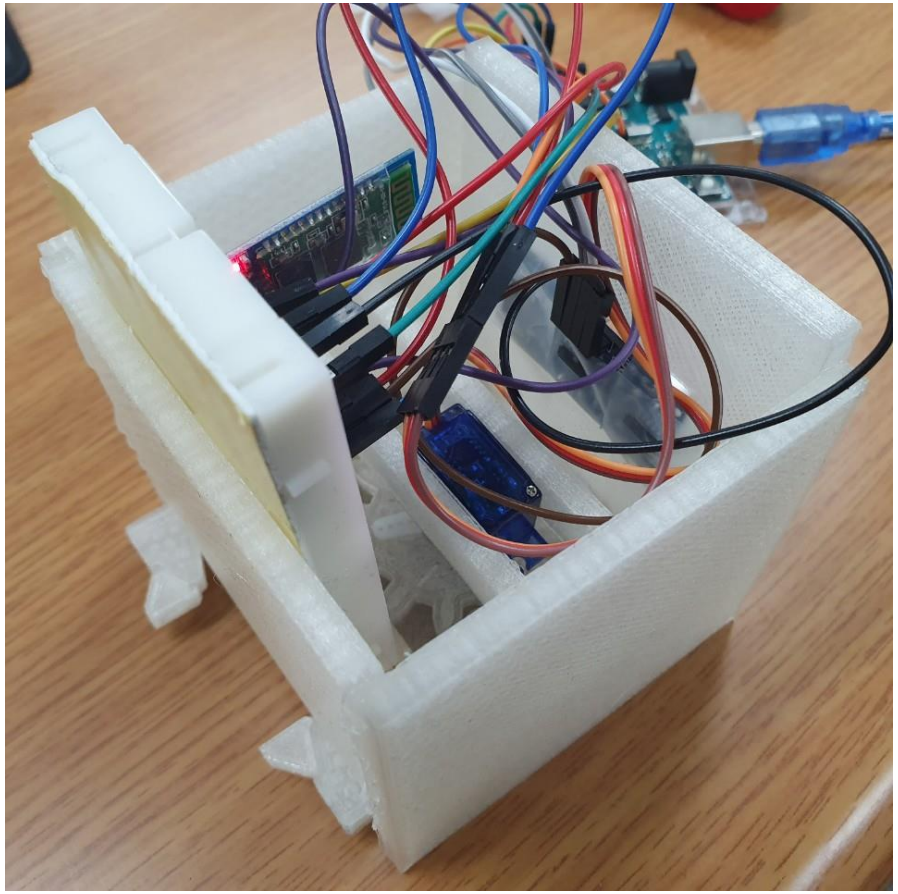
```
void switchOn(char s) {
  Serial.println("ON");
  Serial.println(s);
  if(s == '0')
  {
    myservo.write(76);  // 각도 10도로 움직임
    delay(400);
    myservo.write(66);
  }
  else if(s == '1')
  {
    myservo.write(86);  // 각도 20도로 움직임
    delay(400);
    myservo.write(66);
  }
  else if(s == '2')
  {
    myservo.write(106);  // 각도 30도로 움직임
    delay(400);
    myservo.write(66);
  }
  currentState = true;

}

void switchOff(char s) {
  Serial.println("OFF");
   Serial.println(s);
  if(s == '0')
  {
    myservo.write(56);  // 각도 10도로 움직임
    delay(400);
    myservo.write(66);
  }
  else if(s == '1')
  {
    myservo.write(46);  // 각도 20도로 움직임
    delay(400);
    myservo.write(66);
  }
  else if(s == '2')
```

# 부품 조립 및 완성

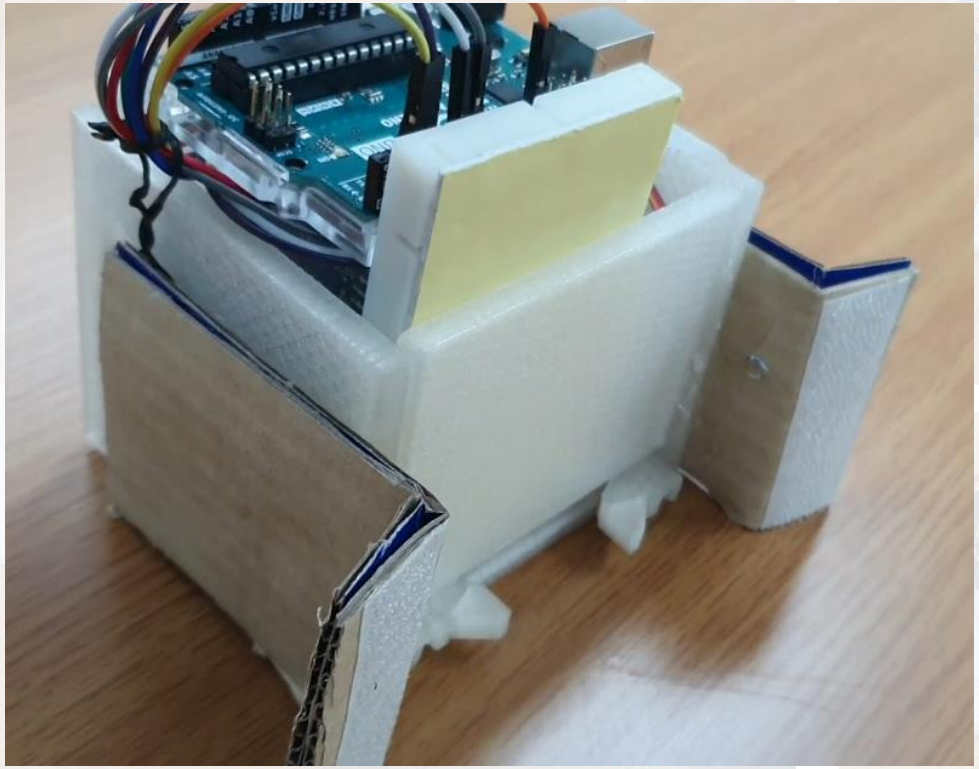


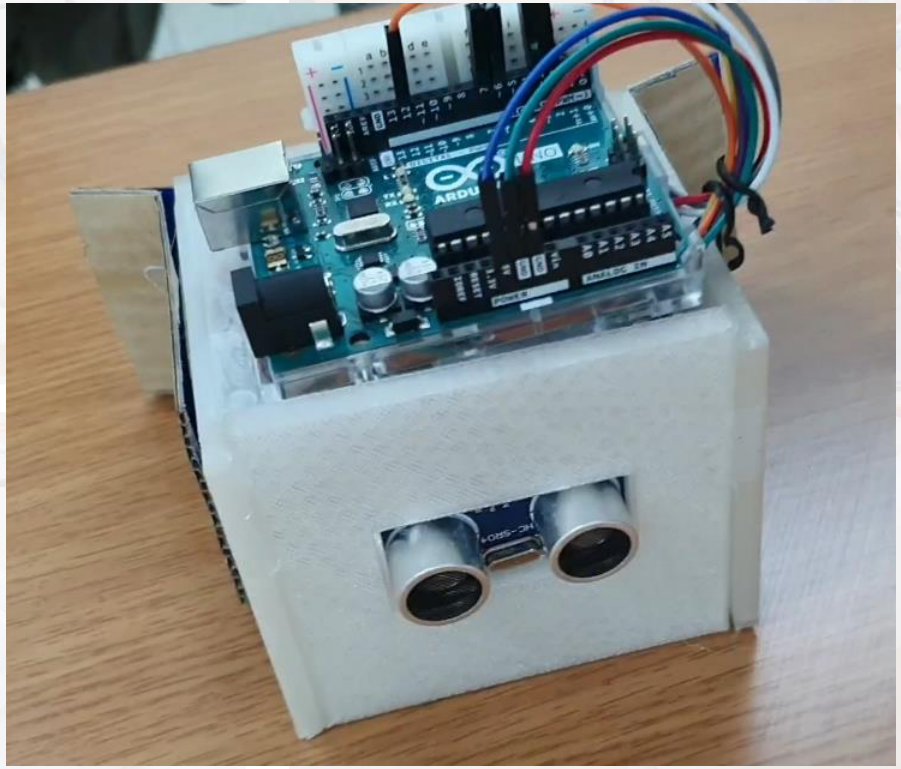






# 부품 조립 및 완성

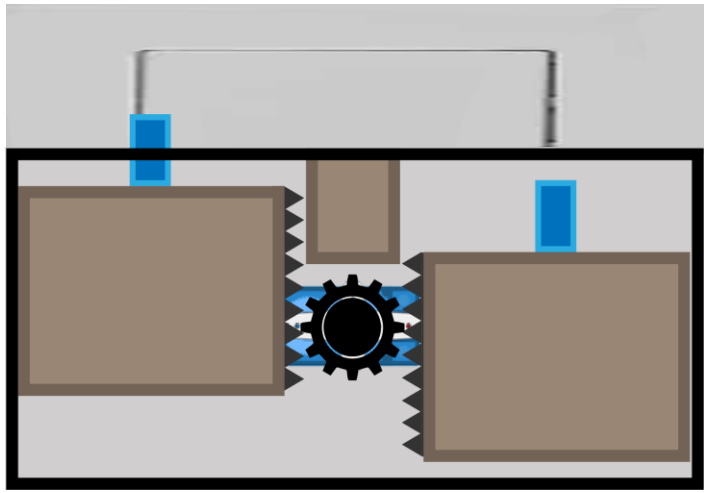


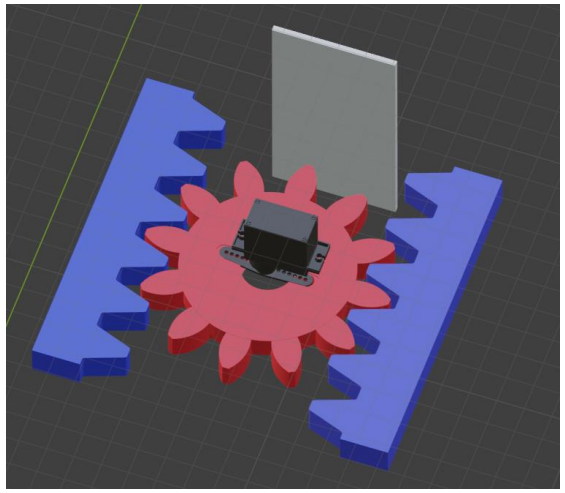


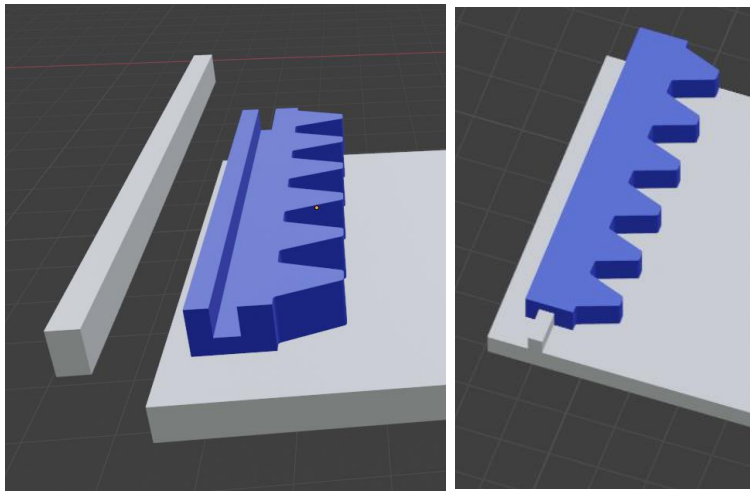
# 기술적 문제점 및 해결

보다 안정된 장치의 동작을 위해 래크와 피니언 기어를 이용하고자 했는데
단순히 기어만 있는 상태라면 언제나 일관된 움직임을 기대할 수 없음.

-> 래크 기어가 일정한 움직임을 할 수 있도록 추가적인 모델링 진행.



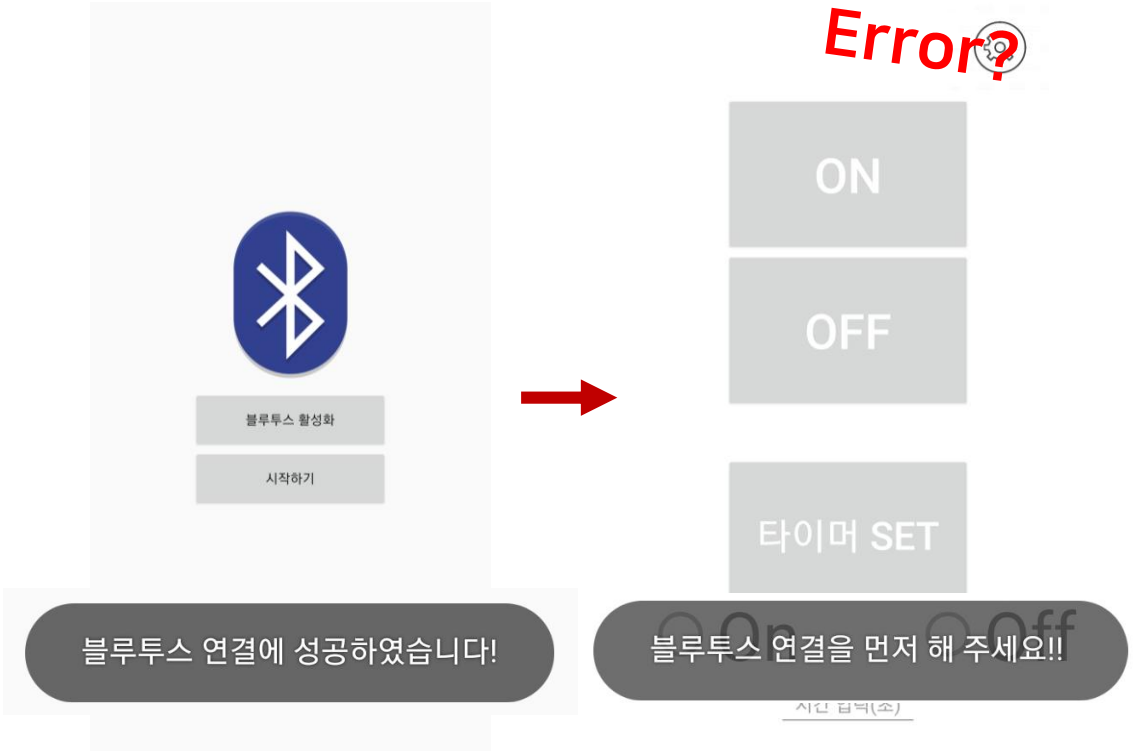




# 기술적 문제점 및 해결

**어플에서 새 액티비티를 열면 블루투스 연결이 종료되는 현상.**
**어플을 백그라운드 상태로 돌리면 블루투스 연결이 종료됨.**

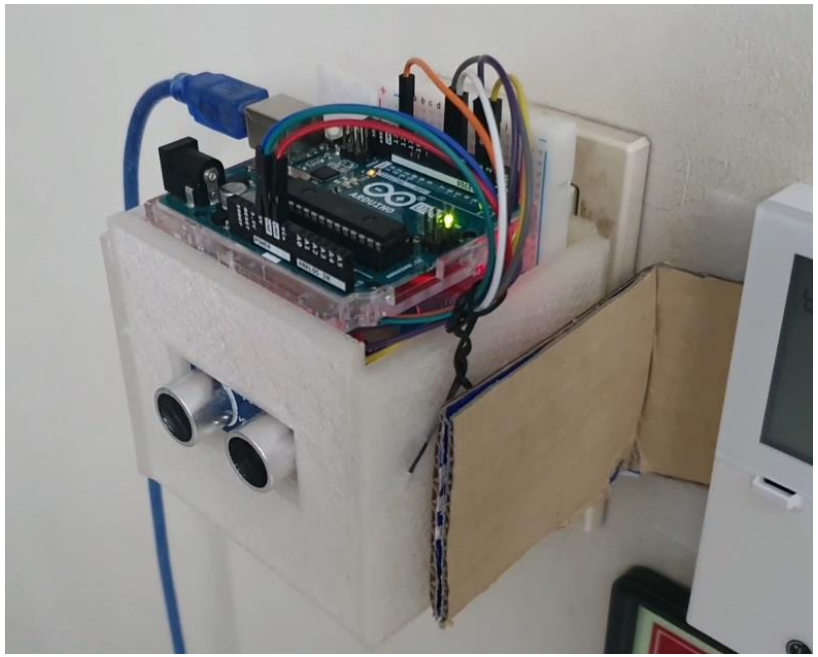
**-> 블루투스 연결 및 통신을 서비스 작업으로 구현.**



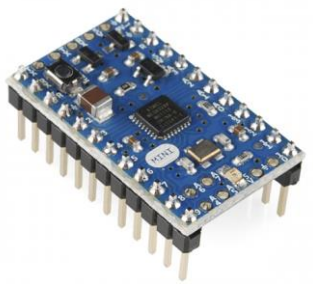
# 장치 경량화

만들어진 최종 프로젝트 결과물은 아래 사진과 같이 상당한 부피를 가짐.
기능 연산을 담당하는 보드를 아두이노 미니와 같은 소형 보드를 사용하고,
배선 설계를 더 최적화 시키면 더 활용하기 편할 것으로 생각됨.

실 부착된 모습



경량화를 위한 계획



경량
프레임

효율적인
배선

# 팀원 역할

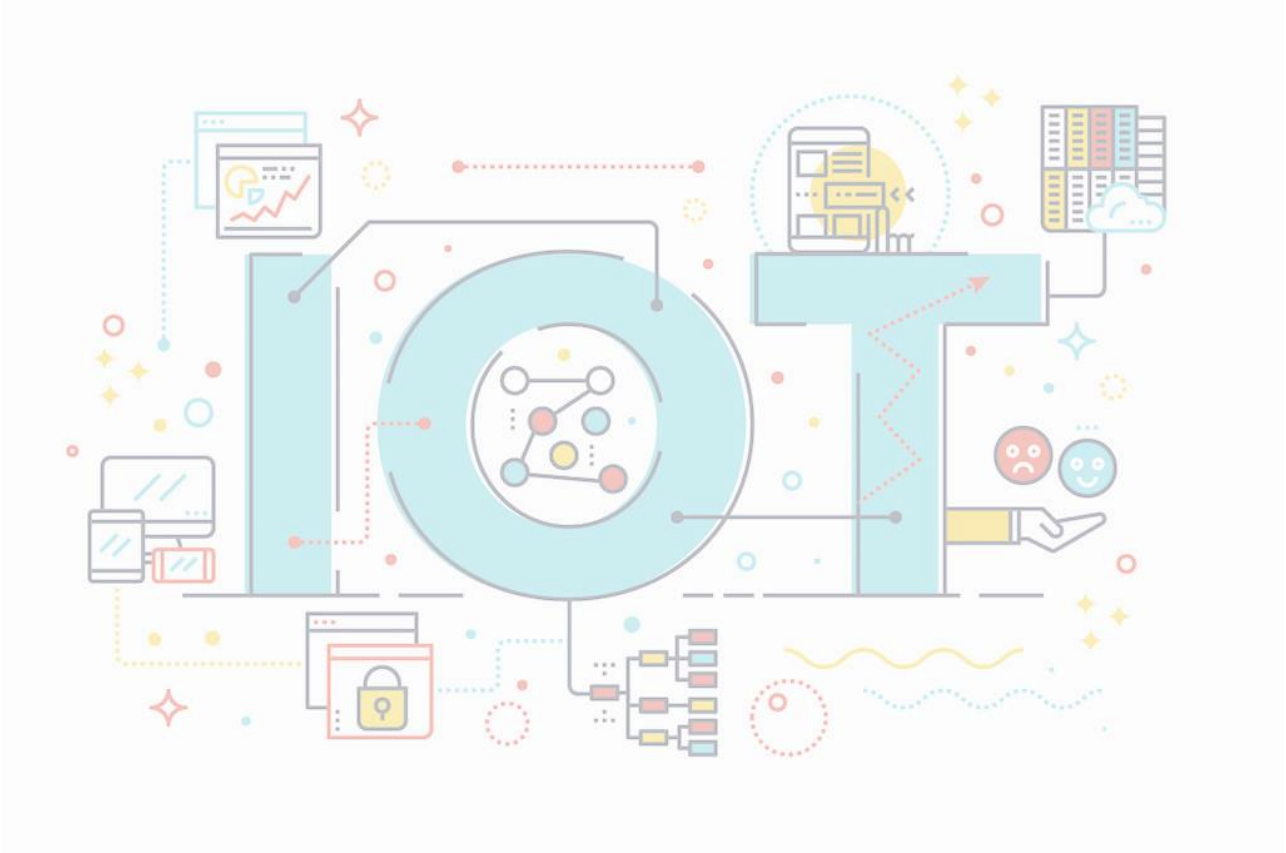
## 임정후

- Blender 이용한 모델링
- 3D 프린터로 출력
- ppt 제작 및 발표

## 유호균

- Sketch 이용한 아두이노 프로젝트 코딩
- 부품 조립 및 완성
- 동영상 촬영 및 편집

- 안드로이드 코딩
- Serial Bluetooth Terminal 이용한 부품 검수
- 발표




# 감사합니다