Reception number	

Academicity Capstone Design Result report

조선대학교 SW중심대학사업단 4차 산업혁명, 조선대학교가 이끌어갑니다.

SW Institute					
Task type	☐Technical solution type		■Problem excavation		
Task area	☐AI	☐Information protection	☐VR/AR	■IOT	☐Big Data
Task name	Cuniture		Team name	IOTainment	
Period of execution	2021. 3. 3. ~ 2021. 6. 9. (15 weeks)				
Coaching staff	Affiliation	National Program of Excellence in Software center	Name	Namhong Park	
	Contact	parknh@chosun.ac.kr	E-mail	010-7181-6040	
Participating agency	Name	KEPCO KDN	Participating agency Human resources man	Department / position	Deputy General Manager
	Representative	Hee-Pil Jang		Name	Hee-Pil Jang

Participating students	Major	Grade	Class Number	Name	Contact
	Computer Engineering	4	20154284	Minki Park	
	Computer Engineering	4	20164263	Yeonghak Yu	
	Computer Engineering	4	20184384	Jeongyeon Yu	
	Computer Engineering	4	20184435	Minhee Hong	

Submit a result report according to the Design Support Plan of the University Campus Support Business Program, SWU-centric Business Support Project.

2021 year 06 month 16 day

Coaching Staff : Namhong Park (sign)

Representative Student : Minki Park (sign)

Chosun University SW Master of Education

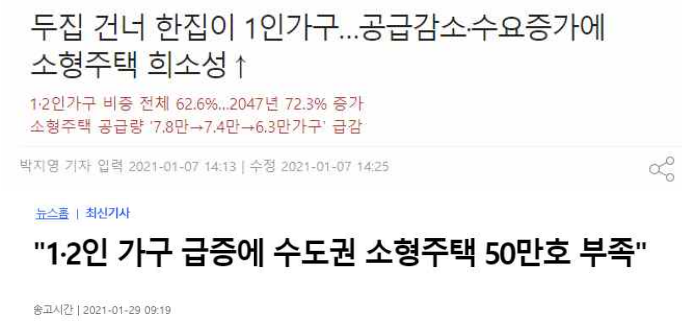
1. Task Overview

A. Task Summary

【IDEA Background and Necessity】

① **The generalization of small houses** due to the increase in the number of single-person households

According to the resident registration statistics released recently by the Ministry of Public Administration and Security, **the number of single-person households** was 9,063,362, accounting for 39.2% of the total households. When the proportion of two-person households is added, it reached **62.6%** of the total. As a result, the structure of the residential space itself is changing to a smaller scale. However, as the size of the house **becomes smaller,** the activities that can be done in it are bound to **be limited.**



(Source: NewDaily Economy)

② **Increased Need to secure space** due to **COVID-19**

We asked 1,000 men and women between the ages of 19 and 59 nationwide about the meaning of home and changes in daily life after 'COVID-19', and **the change in the amount of time spent at home increased significantly** after 'social distancing' was implemented. In particular, it was investigated that the **'value of a house'** is being **emphasized** more as a space for psychological stability and various activities.

(Source:Trend Monitor)

In an environment where we have no choice but to work at home due to the corona virus, we wanted to make **Querencia** in the homes of modern people to solve the restrictive problems caused by cramped spaces.

(**Qerencia**, which means refuge or shelter in Spanish, is an architectural term that means a place where cattle rest for a while before the final battle with a matador at a bullfighting arena, but it is also an architectural term that means a place of rest for modern people who are tired of busy daily life.)

B. Group members and roles

Number	Class	Name	Roles	Participation degree(%)
1	Computer Engineering	Minki Park	General Manager (SW & HW Development. Writing a Report, Presentation)	25 %
2	Computer Engineering	Yeonghak Yu	HW Development (Product Design & Making, Assembling Parts)	25 %
3	Computer Engineering	Jeongyeon Yu	SW Development (Google Assistant & Arduino Connection)	25 %
4	Computer Engineering	Minhee Hong	HW Development (Assembling Parts), Writing a Reprot	25 %

C. Contents of weekly tasks

Week	Contents of weekly tasks	Enterprise Participation status	Participation Name of Enterprise
1	Group Composition	-	
2	Role Sharing and IDEA Conference (1)	-	
3	IDEA Conference (2)	-	
4	IDEA Conference (3)	-	
5	SW & HW Design	○	KEPCO KDN
6	SW Development - Application	-	
7	Hw Development - Motor Combination	-	
8	Prototype Test	-	
9	Interim Announcement & Order Necessary Products	○	KEPCO KDN
10	Products check and Testing IDEA Conference (4) SW IDEA Competition Participation	-	
11	IDEA Change	-	
12	SW Development - Arduino & Google Assistant SW Development - Bluetooth & Application	-	
13	SW & HW Redesign HW Development - Motor Combination SW IDEA Competition Presentation	-	
14	Parts Combination & Final Test	-	
15	Writing a Final Report	-	
16	Final Presentation	○	KEPCO KDN

○ Task Log

① IDEA Planning

- 2021.03.10. Role sharing and Brainstorming

② Project Document preparation & Review

- 2021.03.17. Brainstorming and Writing an Idea plan
- 2021.03.24. Writing a topic selection report
- 2021.03.31. Final topic selection through meeting with a Professor
- 2021.04.07. Brief project presentation
- 2021.04.14. Final review of the assignment report
- 2021.04.15. Final approval to professor

③ Study

- 2021.04.22. SW & HW design required for project progress
- 2021.04.23. Android Application Development (SW)
- 2021.04.24. Motor Test (HW)
- 2021.04.26. Prototype Test

④ Preparation for Presentation

- 2021.04.28. Interim Announcement

⑤ IDEA Change & Product Inspection, Parts Testing

- 2021.05.04. IDEA Conference
- 2021.05.06. IDEA Change
- 2021.05.13. Order Necessary parts for prototype
- 2021.05.19. Inspect ordered necessary parts for prototype, Product inspection, Testing
- 2021.05.16. SW IDEA Competition Document Submission
- 2021.05.21. Submit an assignment expense statement

⑥ SW IDEA Competition

- 2021.06.04. SW IDEA Competition Announcement & Awards

⑦ SW & HW Development / System Build / Bug fix

- 2021.05.29. SW & HW Redeisgn
- 2021.05.31. Bluetooth App Development for Test Model (SW)
- 2021.06.02. Arduino Program Development for Test Model (SW)
- 2021.06.03. Making a Embeded closet for Test Model (HW)
- 2021.06.04. Making a house for Test Model (HW)
- 2021.06.07. Linear Motor Design (HW) & Bluetooth App Development (SW)
- 2021.06.08. Servo Motor Design (HW)
- 2021.06.09. Google Assistant Development (SW)
- 2021.06.10. Servo Motor Redesign (HW)
- 2021.06.11. Making a Embeded closet (HW)
- 2021.06.12. Making a lower part of the house (HW)
- 2021.06.13. Making a lower part of the house (HW)
- 2021.06.14. Combination of motor and closet (HW)
- 2021.06.15. House wall construction, furniture making (HW)
- 2021.06.16. Interior decoration, Landscape composition (HW)

⑧ Final Test

- 2021.06.16. Final Test

⑨ Preparation for Presentation

- 2021.06.12. ~ 2021.06.16. Writing a final report
- 2021.06.16. ~ 2021.06.22. Prepare presentation materials
- 2021.06.23. Presentation

2. Task result

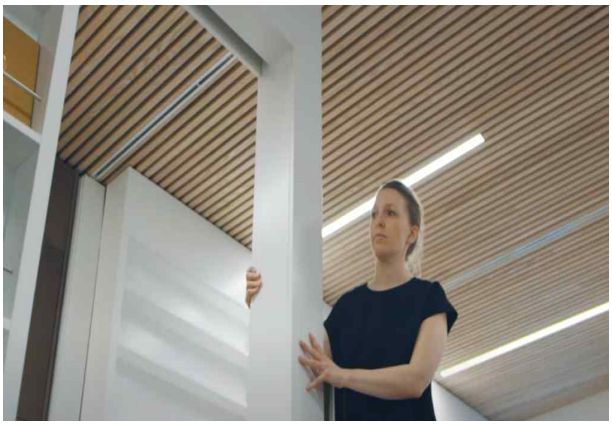
A. Task contents

【Problems with existing technologies/products - Manual Operation】

The concept of moving a wall or furniture to reconfigure the structure of the house has existed in the past. However, it is inconvenient to have to operate it **manually.**

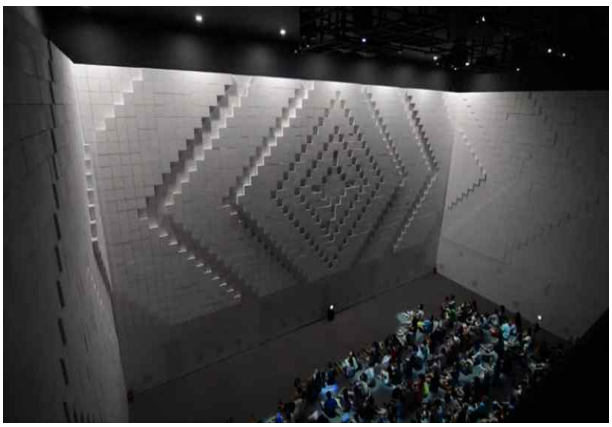
(Source:YouTube, Moving walls transform home into office in Tokyo apartment)

(Source:YouTube, Pivoting wall adds/subtracts rooms in NYC modular micro-flat)

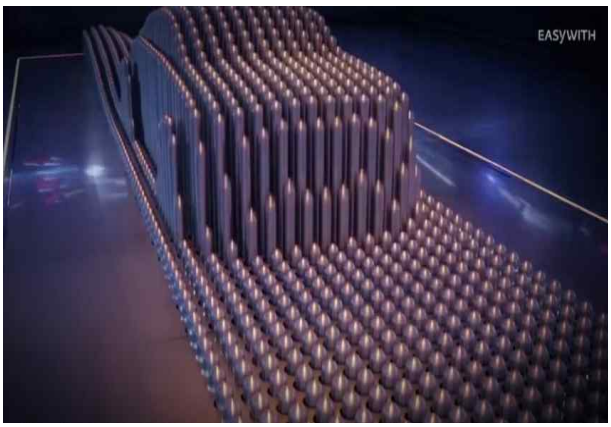


(Source:YouTube, Reconfigurable apartment allows residents to transform their living spaces)

Then, while watching Hyundai Motor's **Matrix Wall** video, We thought that if We could make the wall move automatically, the residents would be able to use it comfortably.



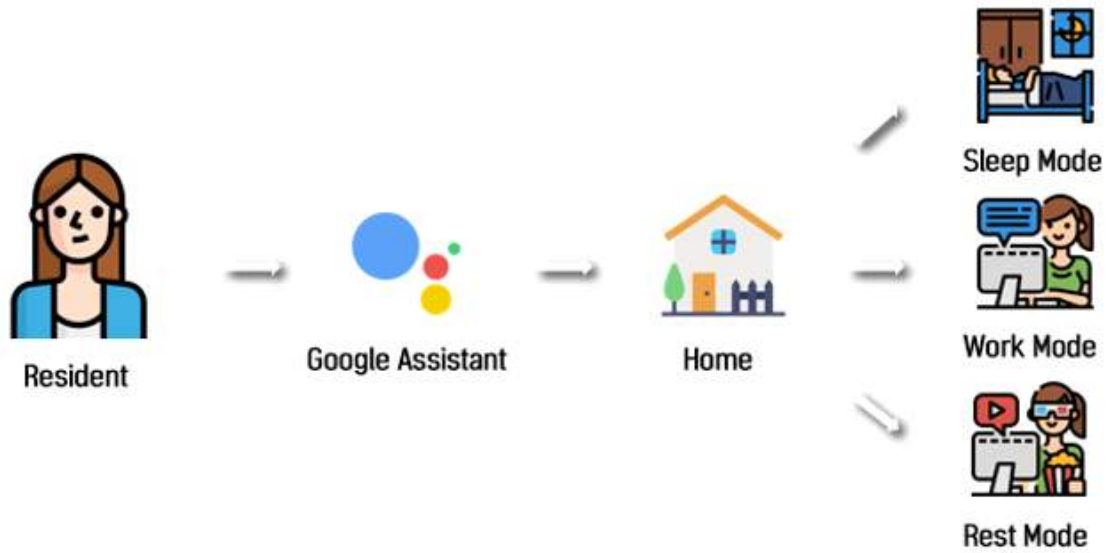
(Source: YouTube, Hyundai Hyper-Matrix Wall)



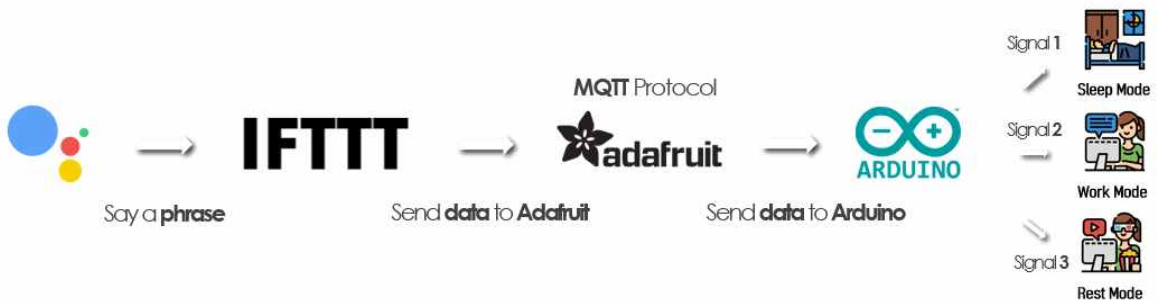
(Source: YouTube, Kinetic Installation for Hyundai Motorstudio Goyang, Korea 2017)

- Task Performance & Result Photography

1) Overview



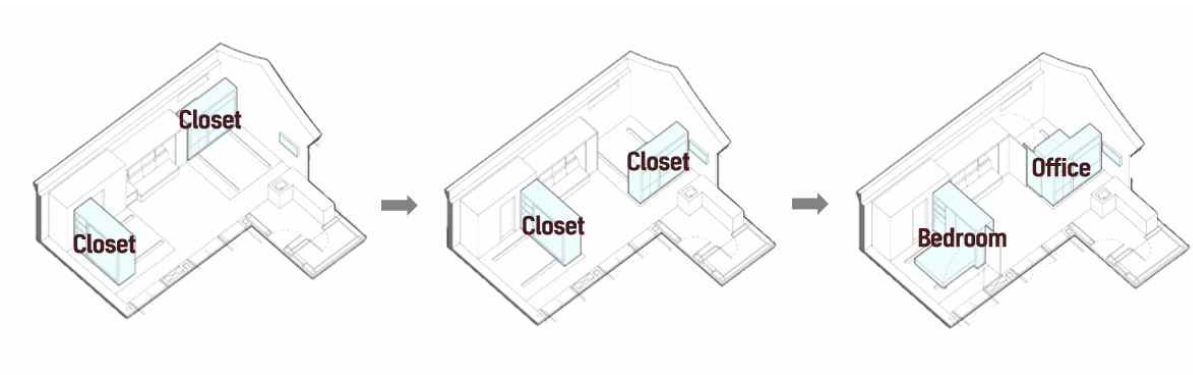
Conceptual Diagram



Google Assistant Operation Procedure

Google Assistant : The user calls command
IFTTT : Receive commands and operate adafruit
Adafruit : Passing a specific value to the Arduino during operation
ARDUINO : Hardware behavior when a specific value is passed

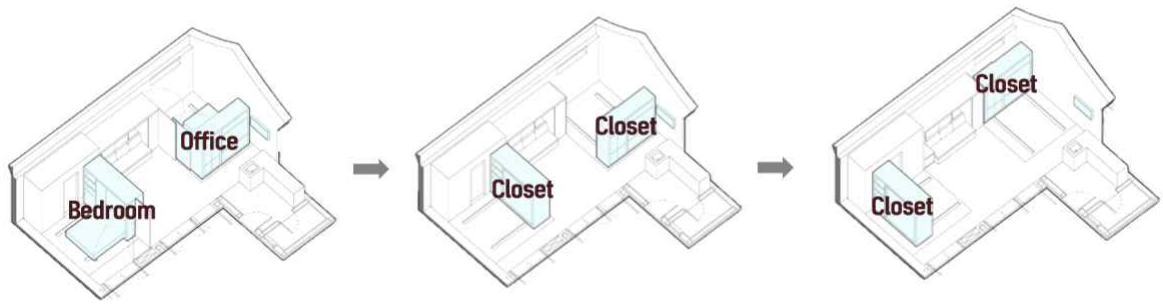
2) Estimated Drawing



Situation 1

"Hello Google, Make a bed for me to sleep."

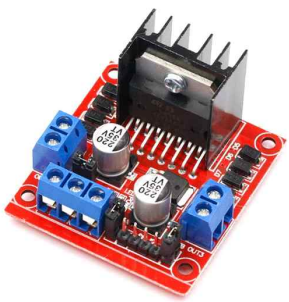
"Hello Google, Make a desk for me to study."

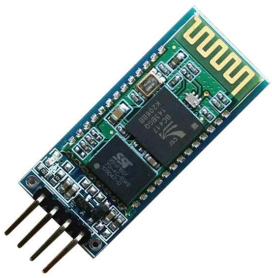


Situation 2

"Hello Google, My friends are coming to play, So can you clean up the room?"

3) Parts Selection

Part	Product	Description & Reason for selection
MCU	 [WIFI ESP8266 D1 R1 Board]	A board that can transmit sketches wirelessly based on the most used ESP8266 Using Google Assistant over Wi-Fi
Actuator	 [Electric cylinder small linear motor(KGU-3429) Stroke 140mm 12V]	It is the cheapest among linear motors, and stable movement is possible due to its low speed. Used to move the closet
	 [MG995 servo motor]	High portability, simple angle control. Stronger and more stable than the typical SG90 motor. Used to control beds and desks
Motor Driver	 [L298N]	Motor driver module for motor control. Essential for projects that require motor control Used to program the direction control of the linear motor in Arduino.

Part	Product	Description & Reason for selection
Sensor	 [LED]	A basic semiconductor device that converts electric current into light Used to illuminate the light after making a desk
	 [Ultrasonic sensor]	A sensor that can measure the distance by sending a high-frequency sound over 20kHz and measuring the time difference when it is reflected and returned Used to precisely control the moving distance of the closet
Battery	 [Bexel Batteries / 6V]	Reliable battery while providing high output power to a wide range of devices Used to stably operate a linear motor that requires high output.
	 [DURACELL Batteries / 9V]	The ideal general-purpose battery when you want to provide reliable power to your everyday devices. Used to supply stable power to the servomotor and Bluetooth connected to the Arduino
communication device	 [HC-06]	TX/RX communication, A product that provides two-way communication among various Bluetooth modules, is inexpensive and provides the most stable connection. Used for connection with a smartphone to work as an app

4) Code

4.1 Arduino

File Name	Content

bedarduino.ino	#include "config.h"
	#include "Servo.h"
	#include <SoftwareSerial.h>
	
	Servo servo_b;
	
	// 피드 이름 설정
	AdafruitIO_Feed *counter = io.feed("control");
	
	// 핀 번호
	// #침대#
	// Motor_A1 // D1
	// Motor_A2 // D2
	// Motor_A_Ena // D3
	
	// servo_b // D5
	
	// trigPin_b // D6
	// echoPin_b // D7
	
	SoftwareSerial BTSerial(D9, D10); // 블루투스 Tx, Rx // D9, D10
	
	void setup() {
	  pinMode(D1, OUTPUT);
	  pinMode(D2, OUTPUT);
	  pinMode(D3, OUTPUT);
	
	  servo_b.attach(D5); // 침대 서보
	
	  pinMode(D6, OUTPUT);
	  pinMode(D7, INPUT);
	
	  Serial.begin(115200);
	  BTSerial.begin(9600);

```
  while(! Serial);

  Serial.print("Connecting to Adafruit IO");

  io.connect();

  counter->onMessage(handleMessage);

  while(io.mqttStatus() < AIO_CONNECTED) {
    Serial.print(".");
    delay(500);
  }

  counter->get();

  Serial.println();
  Serial.println(io.statusText());
}

void loop() {
  io.run();
  ///// 어플 ## 블루투스 이용 했을 때
  if(BTSerial.available()) // 블루투스에서 입력된 값이 있다.
  {
    char ch = BTSerial.read(); // 그러면 읽어온 값을 ch에 받는다.
    Serial.println(ch); // 제대로 입력되는지 확인하기 위해 시리얼 모니터에
입력된 값 써준다.
    if(ch == 'a') // 침대 ## 모터 움직이고 침대 내려오기
    {
      // 초음파 거리 재기
      while(1){
        float duration_b, distance_b;
        digitalWrite(D6, HIGH);
        delay(10);
        digitalWrite(D6, LOW);

        duration_b = pulseIn(D7, HIGH);
        distance_b = ((float)(340 * duration_b) / 10000) / 2;

        digitalWrite(D1, HIGH);      // Motor 방향설정1
        digitalWrite(D2, LOW);       // Motor 방향설정2
        digitalWrite(D3, HIGH);      // Motor On
```

```
        if(distance_b < 25){ // 침대 센서 가까이 오면 멈추기
          digitalWrite(D3, LOW);
          break;
        }
      }
      delay(500);
      servo_b.writeMicroseconds(1000); // 서보모터 움직이기 침대
내려오기
    }

    if(ch == 'b') // 침대 ## 침대 올라가고 모터 움직이기
    {
      servo_b.writeMicroseconds(2000); // 서보모터 움직이기 침대
올라가기
      delay(500);
      while(1){
        float duration_b, distance_b;
        digitalWrite(D6, HIGH);
        delay(10);
        digitalWrite(D6, LOW);

        duration_b = pulseIn(D7, HIGH);
        distance_b = ((float)(340 * duration_b) / 10000) / 2;

        digitalWrite(D1, HIGH);      // Motor 방향설정1
        digitalWrite(D2, LOW);       // Motor 방향설정2
        digitalWrite(D3, HIGH);      // Motor On

        if(distance_b >= 37){ // 침대 멀리 가면 멈추기
          digitalWrite(D3, LOW);
          break;
        }
      }
    }
  }
}

void handleMessage(AdafruitIO_Data *data) {
  Serial.print("received <- ");
  Serial.println(data->value());
  int state = data->toInt();

  ///// 구글 어시스턴트 ## 구글 어시스턴트 이용 했을 때
```

```
if(state == 1){ // 침대 ## 모터 움직이고 침대 내려오기
  while(1){
    float duration_b, distance_b;
    digitalWrite(D6, HIGH);
    delay(10);
    digitalWrite(D6, LOW);

    duration_b = pulseIn(D7, HIGH);
    distance_b = ((float)(340 * duration_b) / 10000) / 2;

    digitalWrite(D1, HIGH);       // Motor 방향설정1
    digitalWrite(D2, LOW);        // Motor 방향설정2
    digitalWrite(D3, HIGH);       // Motor On

    if(distance_b < 25){ // 침대 센서 가까이 오면 멈추기
      digitalWrite(D3, LOW);
      break;
    }
  }
  delay(500);
  servo_b.writeMicroseconds(1000); // 서보모터 움직이기 침대 내려오기
}

else if(state == 2){ // 침대 ## 침대 올라가고 모터 움직이기
  servo_b.writeMicroseconds(2000); // 서보모터 움직이기 침대 올라가기
  delay(500);
  while(1){
    float duration_b, distance_b;
    digitalWrite(D6, HIGH);
    delay(10);
    digitalWrite(D6, LOW);

    duration_b = pulseIn(D7, HIGH);
    distance_b = ((float)(340 * duration_b) / 10000) / 2;

    digitalWrite(D1, HIGH);       // Motor 방향설정1
    digitalWrite(D2, LOW);        // Motor 방향설정2
    digitalWrite(D3, HIGH);       // Motor On

    if(distance_b >= 37){ // 침대 멀리 가면 멈추기
      digitalWrite(D3, LOW);
      break;
    }
```

	} } }

config.h	// adafruit 이름, 키
	#define IO_USERNAME "JGatsby"
	#define IO_KEY "aio_ekfk14xJ9hhrk76ls6mAe4wk0ugG"
	
	// wifi
	#define WIFI_SSID "yyh"
	#define WIFI_PASS "1q2w3e4r"
	
	#include "AdafruitIO_WiFi.h"
	
	#ifdefined(USE_AIRLIFT) ||
	defined(ADAFRUIT_METRO_M4_AIRLIFT_LITE) || \
	defined(ADAFRUIT_PYPORTAL)
	
	#if !defined(SPIWIFI_SS)
	#define SPIWIFI SPI
	#define SPIWIFI_SS 10
	#define NINA_ACK 9
	#define NINA_RESETN 6
	#define NINA_GPIO0 -1
	#endif
	AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID, WIFI_PASS, SPIWIFI_SS,
	NINA_ACK, NINA_RESETN, NINA_GPIO0, &SPIWIFI);
	#else
	AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID, WIFI_PASS);
	#endif

deskarduino.ino	#include "config.h"
	#include "Servo.h"
	
	Servo servo_d;
	
	// 피드 이름 설정
	AdafruitIO_Feed *counter = io.feed("control");
	
	// 핀 번호
	// #책상#
	// Motor_A1 // D1
	// Motor_A2 // D2
	// Motor_A_Ena // D3

```
// servo_d // D5

// trigPin_d // D6
// echoPin_d // D7

// LED // D9

void setup() {
  pinMode(D1, OUTPUT);
  pinMode(D2, OUTPUT);
  pinMode(D3, OUTPUT);

  servo_d.attach(D5); // 침대 서보

  pinMode(D6, OUTPUT);
  pinMode(D7, INPUT);

  pinMode(D9, OUTPUT);

  Serial.begin(115200);

  while(! Serial);

  Serial.print("Connecting to Adafruit IO");

  io.connect();

  counter->onMessage(handleMessage);

  while(io.mqttStatus() < AIO_CONNECTED) {
    Serial.print(".");
    delay(500);
  }

  counter->get();

  Serial.println();
  Serial.println(io.statusText());
}

void loop() {
  io.run();
  // ********여기부터
```

```
  ///// 어플 ## 블루투스 이용 했을 때
  if(Serial.available())
  {
    char ch = Serial.read(); // 그러면 읽어온 값을 ch에 받는다.
    Serial.println(ch); // 제대로 입력되는지 확인하기 위해 시리얼 모니터에 입력된 값 써준다.
    if(ch == 'a') // 책상 ## 모터 움직이고 책상 내려오기
    {
      // 초음파 거리 재고
      while(1){
        float duration_d, distance_d;
        digitalWrite(D6, HIGH);
        delay(10);
        digitalWrite(D6, LOW);

        duration_d = pulseIn(D7, HIGH);
        distance_d = ((float)(340 * duration_d) / 10000) / 2;

        digitalWrite(D1, HIGH);      // Motor 방향설정1
        digitalWrite(D2, LOW);       // Motor 방향설정2
        digitalWrite(D3, HIGH);      // Motor On

        char ch = Serial.read(); // 그러면 읽어온 값을 ch에 받는다.
        if(ch == 'c') // 책상 ## 책상 정지
        {
          digitalWrite(D3, LOW);
          break;
        }
        if(distance_d < 12.5){ // 책상 센서 가까이 오면 멈추기
          digitalWrite(D3, LOW);
          break;
        }
      }
      delay(500);
      servo_d.writeMicroseconds(1000); // 서보모터 움직이기 책상 내려오기
      delay(500);
      digitalWrite(D9, HIGH); // LED ON
    }

    if(ch == 'b') // 책상 ## 책상 올라가고 모터 움직이기
    {
      digitalWrite(D9, LOW); // LED OFF
```

```
        delay(500);
        servo_d.writeMicroseconds(2000); // 서보모터 움직이기 책상 올라가
기
        delay(500);
        while(1){
          float duration_d, distance_d;
          digitalWrite(D6, HIGH);
          delay(10);
          digitalWrite(D6, LOW);

          duration_d = pulseIn(D7, HIGH);
          distance_d = ((float)(340 * duration_d) / 10000) / 2;

          digitalWrite(D1, HIGH);       // Motor 방향설정1
          digitalWrite(D2, LOW);        // Motor 방향설정2
          digitalWrite(D3, HIGH);       // Motor On

          char ch = Serial.read(); // 그러면 읽어온 값을 ch에 받는다.
          if(ch == 'c') // 책상 ## 책상 정지
          {
            digitalWrite(D3, LOW);
            break;
          }
          if(distance_d >= 24){ // 책상 멀리 가면 멈추기
            digitalWrite(D3, LOW);
            break;
          }
        }
      }

      if(ch == 'c') // 책상 ## 책상 정지
      {
        digitalWrite(D3, LOW);
      }
    }
    // *******여기까지는 없어도 됨
}

void handleMessage(AdafruitIO_Data *data) {
  Serial.print("received <- ");
  Serial.println(data->value());
  int state = data->toInt();
```

```
///// 구글 어시스턴트 ## 구글 어시스턴트 이용 했을 때
if(state == 3){ // 책상 ## 책상 움직이고 침대 내려오기
  while(1){
    float duration_d, distance_d;
    digitalWrite(D6, HIGH);
    delay(10);
    digitalWrite(D6, LOW);

    duration_d = pulseIn(D7, HIGH);
    distance_d = ((float)(340 * duration_d) / 10000) / 2;

    digitalWrite(D1, HIGH);      // Motor 방향설정1
    digitalWrite(D2, LOW);       // Motor 방향설정2
    digitalWrite(D3, HIGH);      // Motor On

    if(distance_d < 12.5){ // 책상 센서 가까이 오면 멈추기
      digitalWrite(D3, LOW);
      break;
    }
  }
  delay(500);
  servo_d.writeMicroseconds(1000); // 서보모터 움직이기 침대 내려오기
  delay(500);
  digitalWrite(D9, HIGH); // LED ON
}

else if(state == 4){ // 책상 ## 책상 올라가고 모터 움직이기
  digitalWrite(D9, LOW); // LED OFF
  delay(500);
  servo_d.writeMicroseconds(2000); // 서보모터 움직이기 책상 올라가기
  delay(500);
  while(1){
    float duration_d, distance_d;
    digitalWrite(D6, HIGH);
    delay(10);
    digitalWrite(D6, LOW);

    duration_d = pulseIn(D7, HIGH);
    distance_d = ((float)(340 * duration_d) / 10000) / 2;

    digitalWrite(D1, HIGH);      // Motor 방향설정1
    digitalWrite(D2, LOW);       // Motor 방향설정2
    digitalWrite(D3, HIGH);      // Motor On
```

	if(distance_d >= 24){ // 책상 멀리 가면 멈추기
	digitalWrite(D3, LOW);
	break;
	}
	}
	}
	}

config.h	// adafruit 이름, 키
	#define IO_USERNAME "JGatsby"
	#define IO_KEY "aio_ekfk14xJ9hhrk76ls6mAe4wk0ugG"
	
	// wifi
	#define WIFI_SSID "yyh"
	#define WIFI_PASS "1q2w3e4r"
	
	#include "AdafruitIO_WiFi.h"
	
	#ifdefined(USE_AIRLIFT) ||
	defined(ADAFRUIT_METRO_M4_AIRLIFT_LITE) || \
	defined(ADAFRUIT_PYPORTAL)
	
	#if !defined(SPIWIFI_SS)
	#define SPIWIFI SPI
	#define SPIWIFI_SS 10
	#define NINA_ACK 9
	#define NINA_RESETN 6
	#define NINA_GPIO0 -1
	#endif
	AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID, WIFI_PASS, SPIWIFI_SS,
	NINA_ACK, NINA_RESETN, NINA_GPIO0, &SPIWIFI);
	#else
	AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID, WIFI_PASS);
	#endif

4.2 Android Studio

File Name	Content
MainActivity.java	package com.example.arduino_bluetooth; import androidx.appcompat.app.AppCompatActivity; import androidx.core.app.ActivityCompat; import android.Manifest; import android.app.AlertDialog; import android.bluetooth.BluetoothAdapter; import android.bluetooth.BluetoothDevice; import android.bluetooth.BluetoothSocket; import android.content.BroadcastReceiver; import android.content.Context; import android.content.DialogInterface; import android.content.Intent; import android.content.IntentFilter; import android.os.Bundle; import android.os.Handler; import android.util.Log; import android.view.View; import android.view.WindowManager; import android.widget.AdapterView; import android.widget.ArrayAdapter; import android.widget.Button; import android.widget.ImageView; import android.widget.ListView; import android.widget.TextView; import android.widget.Toast; import java.io.IOException; import java.io.UnsupportedEncodingException; import java.lang.reflect.Method; import java.util.ArrayList; import java.util.List; import java.util.Set; import java.util.UUID; public class MainActivity extends AppCompatActivity { String TAG = "MainActivity"; UUID BT_MODULE_UUID =

```
UUID.fromString("00001101-0000-1000-8000-00805F9B34FB"); //
"random" unique identifier

        Button mBtnConnect;
        TextView textStatus;
        Button btnSleep;
        BluetoothAdapter btAdapter;
        Set<BluetoothDevice> pairedDevices;
        ArrayAdapter<String> btArrayAdapter;
        ArrayList<String> deviceAddressArray;
        Handler mBluetoothHandler;
        List<String> mListPairedDevices;

        ImageView imageView;
        ImageView imageView2;
        ImageView imageView3;
        ImageView imageView4;

        int imageIndex = 0;

        private final static int REQUEST_ENABLE_BT = 1;
        private final static int BT_MESSAGE_READ = 2;
        private final static int BT_CONNECTING_STATUS = 3;
        BluetoothDevice mBluetoothDevice;
        BluetoothSocket btSocket = null;
        ConnectedThread connectedThread; // 소켓 스레드

        @Override
        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            // 상태바 없애기

getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCR
EEN,

WindowManager.LayoutParams.FLAG_FULLSCREEN);

            setContentView(R.layout.activity_main);

            imageView = findViewById(R.id.imageView);
            imageView2 = findViewById(R.id.imageView2);
            imageView3 = findViewById(R.id.imageView3);
            imageView4 = findViewById(R.id.imageView4);
```

```
            // Get permission 위치 권한이 있어야, 나중에 블루투스로
주변 장치 검색이 가능
            String[] permission_list = {
                    Manifest.permission.ACCESS_FINE_LOCATION,

Manifest.permission.ACCESS_COARSE_LOCATION
            };
            ActivityCompat.requestPermissions(MainActivity.this,
permission_list, 1);

            // Enable bluetooth
            btAdapter = BluetoothAdapter.getDefaultAdapter();
            if (!btAdapter.isEnabled()) {
                Intent enableBtIntent = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
                startActivityForResult(enableBtIntent,
REQUEST_ENABLE_BT);
            }

            // variables

            btnSleep = (Button) findViewById(R.id.btn_sleep);
            mBtnConnect = (Button) findViewById(R.id.btnConnect);
            //         mBtnConnect = (Button)
findViewById(R.id.btnConnect);

            mBtnConnect.setOnClickListener(new
Button.OnClickListener() {
                @Override
                public void onClick(View view) {
                    listPairedDevices();
                }
            });

            // 핸들러?
            mBluetoothHandler = new Handler() {
                public void handleMessage(android.os.Message msg)
{
                    if (msg.what == BT_MESSAGE_READ) {
                        String readMessage = null;
                        try {
                            readMessage = new String((byte[])
```

```
msg.obj, "UTF-8");
                                } catch (UnsupportedEncodingException e) {
                                    e.printStackTrace();
                                }
                            }
                        }
                    };

                }

                void listPairedDevices() {
                    if (btAdapter.isEnabled()) {
                        pairedDevices = btAdapter.getBondedDevices();

                        if (pairedDevices.size() > 0) {
                            android.app.AlertDialog.Builder builder = new
android.app.AlertDialog.Builder(this);
                            builder.setTitle("장치 선택");

                            mListPairedDevices = new ArrayList<String>();
                            for (BluetoothDevice device : pairedDevices) {
                                mListPairedDevices.add(device.getName());
                                //mListPairedDevices.add(device.getName()
+ "\n" + device.getAddress());
                            }
                            final CharSequence[] items =
mListPairedDevices.toArray(new
CharSequence[mListPairedDevices.size()]);
                            mListPairedDevices.toArray(new
CharSequence[mListPairedDevices.size()]);

                            builder.setItems(items, new
DialogInterface.OnClickListener() {
                                @Override
                                public void onClick(DialogInterface dialog,
int item) {

connectSelectedDevice(items[item].toString());
                                }
                            });
                            AlertDialog alert = builder.create();
                            alert.show();
                        } else {
```

```
                    Toast.makeText(getApplicationContext(),
"페어링된 장치가 없습니다.", Toast.LENGTH_LONG).show();
                }
            } else {
                Toast.makeText(getApplicationContext(), "블루투스가
비활성화 되어 있습니다.", Toast.LENGTH_SHORT).show();
            }
        }

        public void onClickButtonPaired(View view) {

            btArrayAdapter.clear();
            if (deviceAddressArray != null &&
!deviceAddressArray.isEmpty()) {
                deviceAddressArray.clear();
            }
            pairedDevices = btAdapter.getBondedDevices(); // 기존에
페어링된 디바이스 목록을 가져온다
            if (pairedDevices.size() > 0) {
                // There are paired devices. Get the name and
address of each paired device.
                for (BluetoothDevice device : pairedDevices) {
                    String deviceName = device.getName();
                    String deviceHardwareAddress =
device.getAddress(); // MAC address
                    btArrayAdapter.add(deviceName);

deviceAddressArray.add(deviceHardwareAddress);
                }
            }
        }

        public void onClickButtonSearch(View view) { // 기기
검색하기
            // Check if the device is already discovering
            if (btAdapter.isDiscovering()) {
                btAdapter.cancelDiscovery();
            } else {
                if (btAdapter.isEnabled()) {
                    btAdapter.startDiscovery(); // 주변 기기 검색
                    btArrayAdapter.clear();
                    if (deviceAddressArray != null &&
!deviceAddressArray.isEmpty()) {
```

```
                    deviceAddressArray.clear();
                }
                IntentFilter filter = new
IntentFilter(BluetoothDevice.ACTION_FOUND);
                registerReceiver(receiver, filter);
            } else {
                Toast.makeText(getApplicationContext(),
"bluetooth not on", Toast.LENGTH_SHORT).show();
            }
        }
    }

    // 버튼 눌렀을 때 값 나오도록
    public void onClickButtonSleep(View view) {
        if (connectedThread != null) {
            connectedThread.write("a");
            Toast.makeText(this.getApplicationContext(), "a눌림.",
Toast.LENGTH_SHORT).show();
        }
        imageIndex = 1;
        changeImage();
    }

    public void onClickButtonSleepOff(View view) {
        if (connectedThread != null) {
            connectedThread.write("b");

        }
        imageIndex = 2;
        changeImage();
    }

    public void onClickButtonDesk(View view) {
        if (connectedThread != null) {
            connectedThread.write("c");
        }
        imageIndex = 3;
        changeImage();
    }

    public void onClickButtonDeskOff(View view) {
        if (connectedThread != null) {
            connectedThread.write("d");
```

```
            }
            imageIndex = 4;
            changeImage();
        }

        public void onClickButtonBedStop(View view) {
            if (connectedThread != null) {
                connectedThread.write("e");
            }
        }

        public void onClickButtonDeskStop(View view) {
            if (connectedThread != null) {
                connectedThread.write("f");
            }

        }



        // Create a BroadcastReceiver for ACTION_FOUND.
        private final BroadcastReceiver receiver = new
BroadcastReceiver() {
            public void onReceive(Context context, Intent intent) {
                String action = intent.getAction();
                if (BluetoothDevice.ACTION_FOUND.equals(action)) {
                    // Discovery has found a device. Get the
BluetoothDevice
                    // object and its info from the Intent.
                    BluetoothDevice device =
intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
                    String deviceName = device.getName();
                    String deviceHardwareAddress =
device.getAddress(); // MAC address
                    btArrayAdapter.add(deviceName);

deviceAddressArray.add(deviceHardwareAddress);
                    btArrayAdapter.notifyDataSetChanged();
                }
            }
        };



        @Override
```

```
        protected void onDestroy() {
            super.onDestroy();

            // Don't forget to unregister the ACTION_FOUND
receiver.
            unregisterReceiver(receiver);
        }

        public class myOnItemClickListener implements
AdapterView.OnItemClickListener {

            @Override
            public void onItemClick(AdapterView<?> parent, View
view, int position, long id) {
                Toast.makeText(getApplicationContext(),
btArrayAdapter.getItem(position), Toast.LENGTH_SHORT).show();

                textStatus.setText("try...");

                final String name =
btArrayAdapter.getItem(position); // get name
                final String address =
deviceAddressArray.get(position); // get address
                boolean flag = true;

                BluetoothDevice device =
btAdapter.getRemoteDevice(address);

                // create & connect socket
                try {
                    btSocket = createBluetoothSocket(device);
                    btSocket.connect();
                } catch (IOException e) {
                    flag = false;
                    textStatus.setText("connection failed!");
                    e.printStackTrace();
                }

                // start bluetooth communication
                if (flag) {
                    textStatus.setText("connected to " + name);
                    connectedThread = new
ConnectedThread(btSocket);
```

```
                    connectedThread.start();
                }

            }
        }

        //
        void connectSelectedDevice(String selectedDeviceName) {
            for (BluetoothDevice tempDevice : pairedDevices) {
                if
(selectedDeviceName.equals(tempDevice.getName())) {
                    mBluetoothDevice = tempDevice;
                    break;
                }
            }
            try {
                btSocket =
mBluetoothDevice.createRfcommSocketToServiceRecord(BT_MODULE
_UUID);
                btSocket.connect();
                connectedThread = new
ConnectedThread(btSocket);
                connectedThread.start();

mBluetoothHandler.obtainMessage(BT_CONNECTING_STATUS, 1,
-1).sendToTarget();
            } catch (IOException e) {
                Toast.makeText(getApplicationContext(), "블루투스
연결 중 오류가 발생했습니다.", Toast.LENGTH_LONG).show();
            }
        }

        private BluetoothSocket
createBluetoothSocket(BluetoothDevice device) throws IOException {
            try {
                final Method m =
device.getClass().getMethod("createInsecureRfcommSocketToServiceR
ecord", UUID.class);
                return (BluetoothSocket) m.invoke(device,
BT_MODULE_UUID);
            } catch (Exception e) {
                Log.e(TAG, "Could not create Insecure RFComm
Connection", e);
```

	}
	return
	device.createRfcommSocketToServiceRecord(BT_MODULE_UUID);
	}
	
	private void changeImage() {
	if (imageIndex == 1) {
	imageView.setVisibility(View.VISIBLE);
	imageView2.setVisibility(View.INVISIBLE);
	imageView3.setVisibility(View.INVISIBLE);
	imageView4.setVisibility(View.INVISIBLE);
	
	} else if (imageIndex == 2) {
	imageView.setVisibility(View.INVISIBLE);
	imageView2.setVisibility(View.VISIBLE);
	imageView3.setVisibility(View.INVISIBLE);
	imageView4.setVisibility(View.INVISIBLE);
	
	} else if (imageIndex == 3) {
	imageView.setVisibility(View.INVISIBLE);
	imageView2.setVisibility(View.INVISIBLE);
	imageView3.setVisibility(View.VISIBLE);
	imageView4.setVisibility(View.INVISIBLE);
	
	} else if (imageIndex == 4) {
	imageView.setVisibility(View.INVISIBLE);
	imageView2.setVisibility(View.INVISIBLE);
	imageView3.setVisibility(View.INVISIBLE);
	imageView4.setVisibility(View.VISIBLE);
	
	}
	}
	}

ConnectedThread.java	package com.example.arduino_bluetooth; import android.bluetooth.BluetoothSocket; import android.os.Handler; import android.os.SystemClock; import java.io.IOException; import java.io.InputStream; import java.io.OutputStream;

```
public class ConnectedThread extends Thread {
    private final BluetoothSocket mmSocket;
    private final InputStream mmInStream;
    private final OutputStream mmOutStream;



    public ConnectedThread(BluetoothSocket socket) {
        mmSocket = socket;
        InputStream tmpIn = null;
        OutputStream tmpOut = null;

        // Get the input and output streams, using temp objects
because
        // member streams are final
        try {
            tmpIn = socket.getInputStream();
            tmpOut = socket.getOutputStream();
        } catch (IOException e) {
        }

        mmInStream = tmpIn;
        mmOutStream = tmpOut;
    }

    @Override
    public void run() {
        byte[] buffer = new byte[1024];  // buffer store for the
stream
        int bytes; // bytes returned from read()
        // Keep listening to the InputStream until an exception
occurs
        while (true) {
            try {
                // Read from the InputStream
                bytes = mmInStream.available();
                if (bytes != 0) {
                    buffer = new byte[1024];
                    SystemClock.sleep(100); //pause and wait for
rest of data. Adjust this depending on your sending speed.
                    bytes = mmInStream.available(); // how many
bytes are ready to be read?
```

	bytes = mmInStream.read(buffer, 0, bytes); // record how many bytes we actually read
	}
	} catch (IOException e) {
	e.printStackTrace();
	
	break;
	}
	}
	}
	
	/* Call this from the main activity to send data to the remote device */
	public void write(String input) {
	byte[] bytes = input.getBytes(); //converts entered String into bytes
	try {
	mmOutStream.write(bytes);
	} catch (IOException e) {
	}
	}
	
	/* Call this from the main activity to shutdown the connection */
	public void cancel() {
	try {
	mmSocket.close();
	} catch (IOException e) {
	}
	}
	}

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" android:background="@color/colorBlack" tools:context=".MainActivity"> <FrameLayout

```
        android:layout_width="match_parent"
        android:layout_height="300dp"
        android:layout_weight="6">

        <ImageView
            android:id="@+id/imageView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:visibility="invisible"
            app:srcCompat="@drawable/btn_sleepon"
            android:layout_gravity="center" />

        <ImageView
            android:id="@+id/imageView2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:visibility="invisible"
            app:srcCompat="@drawable/btn_deskon"
            android:layout_gravity="center" />

        <ImageView
            android:id="@+id/imageView3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:visibility="invisible"
            app:srcCompat="@drawable/btn_sleepoff"
            android:layout_gravity="center" />

        <ImageView
            android:id="@+id/imageView4"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:visibility="invisible"
            app:srcCompat="@drawable/btn_deskoff"
            android:layout_gravity="center" />

    </FrameLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:orientation="horizontal">
```

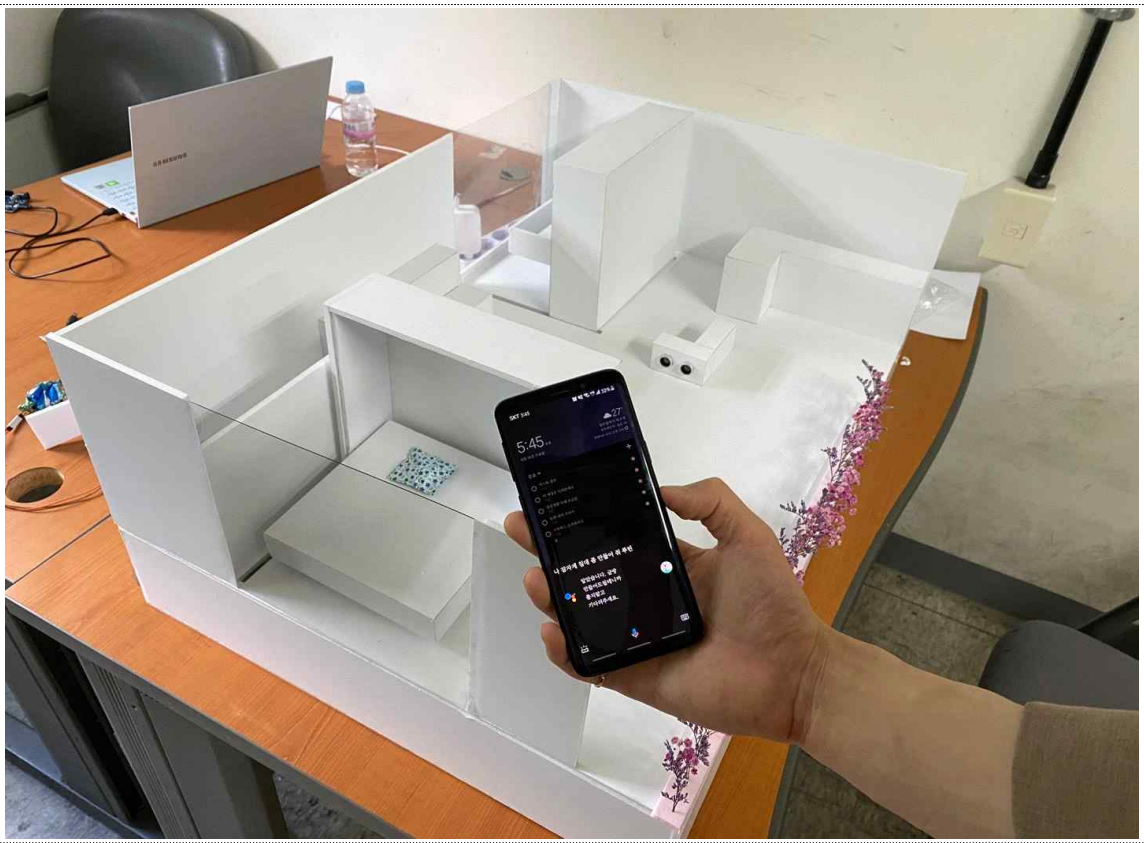
```
        <Button
            android:id="@+id/btn_sleep"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:fontFamily="@font/sebang_gothic"
            android:onClick="onClickButtonSleep"
            android:text="SLEEP ON"
            android:textColor="#2F223C"
            android:layout_margin="3dp"
            android:background="@drawable/shadow"/>

        <Button
            android:id="@+id/btn_sleep_back"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:fontFamily="@font/sebang_gothic"
            android:onClick="onClickButtonDesk"
            android:text="STUDY on"
            android:layout_margin="3dp"
            android:background="@drawable/shadow"/>
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:orientation="horizontal">

        <Button
            android:id="@+id/btn_hobby"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:fontFamily="@font/sebang_gothic"
            android:onClick="onClickButtonSleepOff"
            android:text="SLEEP OFF"
            android:layout_margin="3dp"
            android:background="@drawable/shadow"/>

        <Button
```

```
            android:id="@+id/button2"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:fontFamily="@font/sebang_gothic"
            android:onClick="onClickButtonDeskOff"
            android:text="STUDY  OFF"
            android:layout_margin="3dp"
            android:background="@drawable/shadow"/>
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:orientation="horizontal">

        <Button
            android:id="@+id/btn_bed_stop"
            android:layout_width="10dp"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:background="#00ff0000"
            android:onClick="onClickButtonBedStop"  />

        <Button
            android:id="@+id/btnConnect"
            android:layout_width="60dp"
            android:layout_height="40dp"
            android:layout_gravity="center"
            android:background="@drawable/btn_bluetooth"
            android:fontFamily="@font/sebang_gothic"  />

        <Button
            android:id="@+id/btn_desk_stop"
            android:layout_width="10dp"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:background="#00ff0000"
            android:onClick="onClickButtonDeskStop"  />
    </LinearLayout>

</LinearLayout>
```

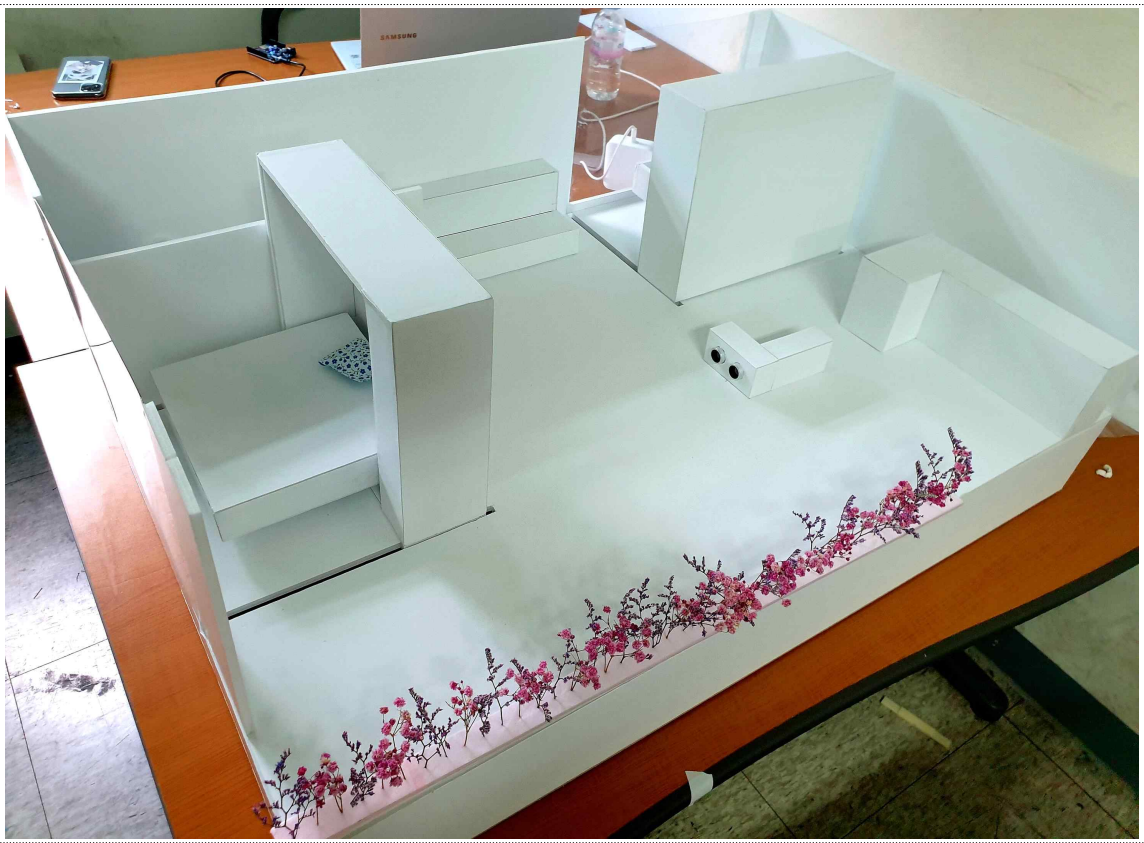
5) Task Result

Living room - initial state

1-1) Bedroom Closet

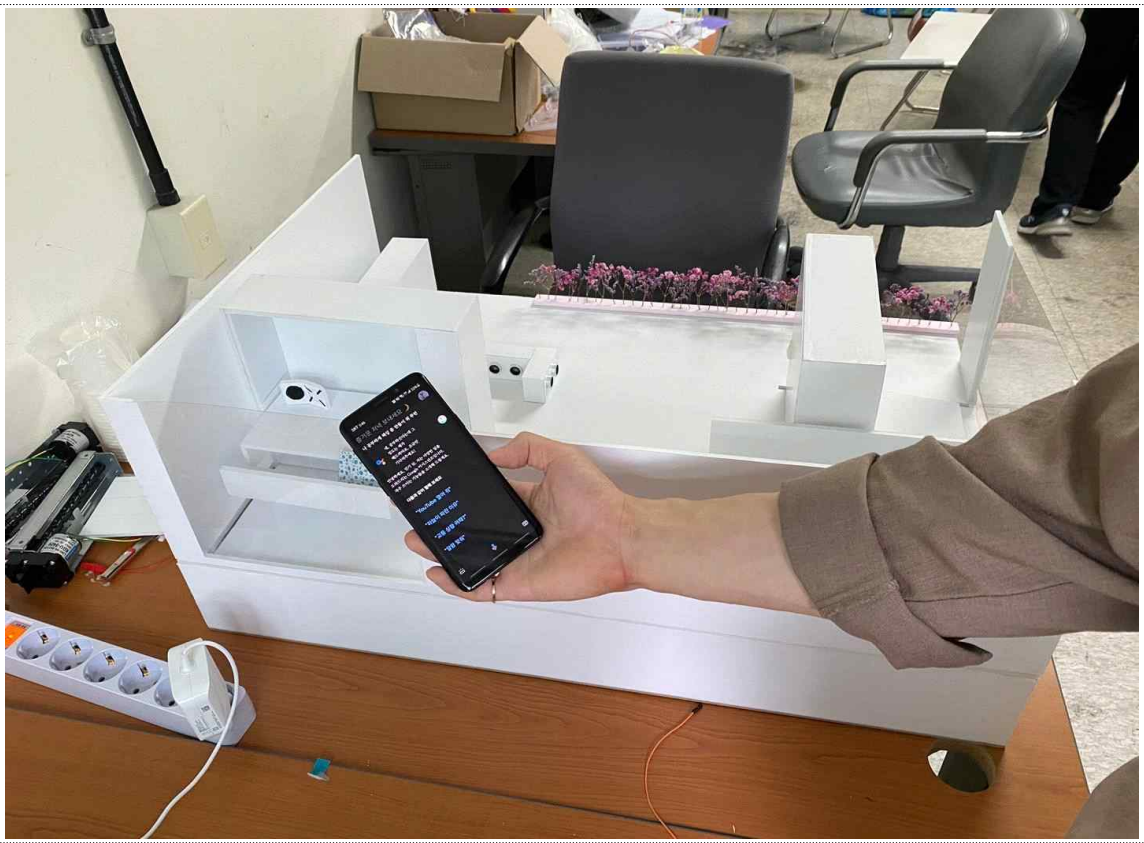


1-2) Create a bedroom with Google Assistant

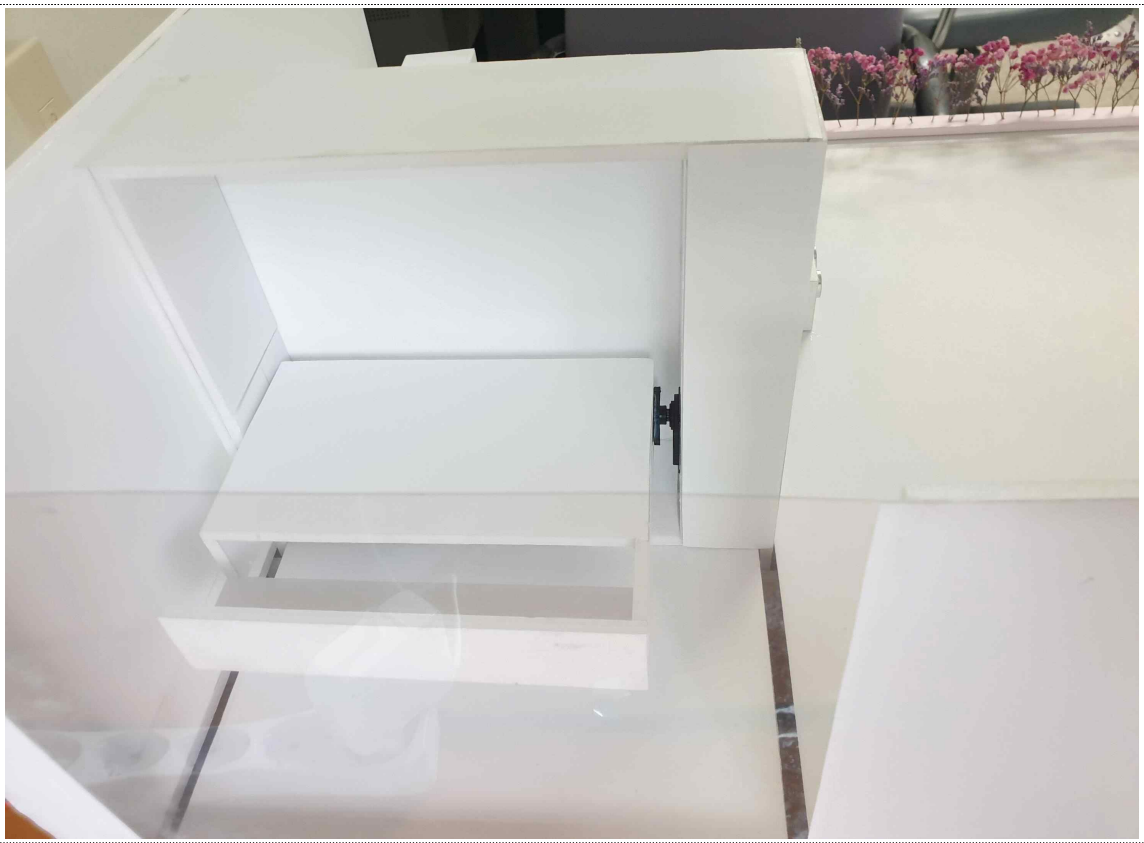


1-3) Complete bedroom creation

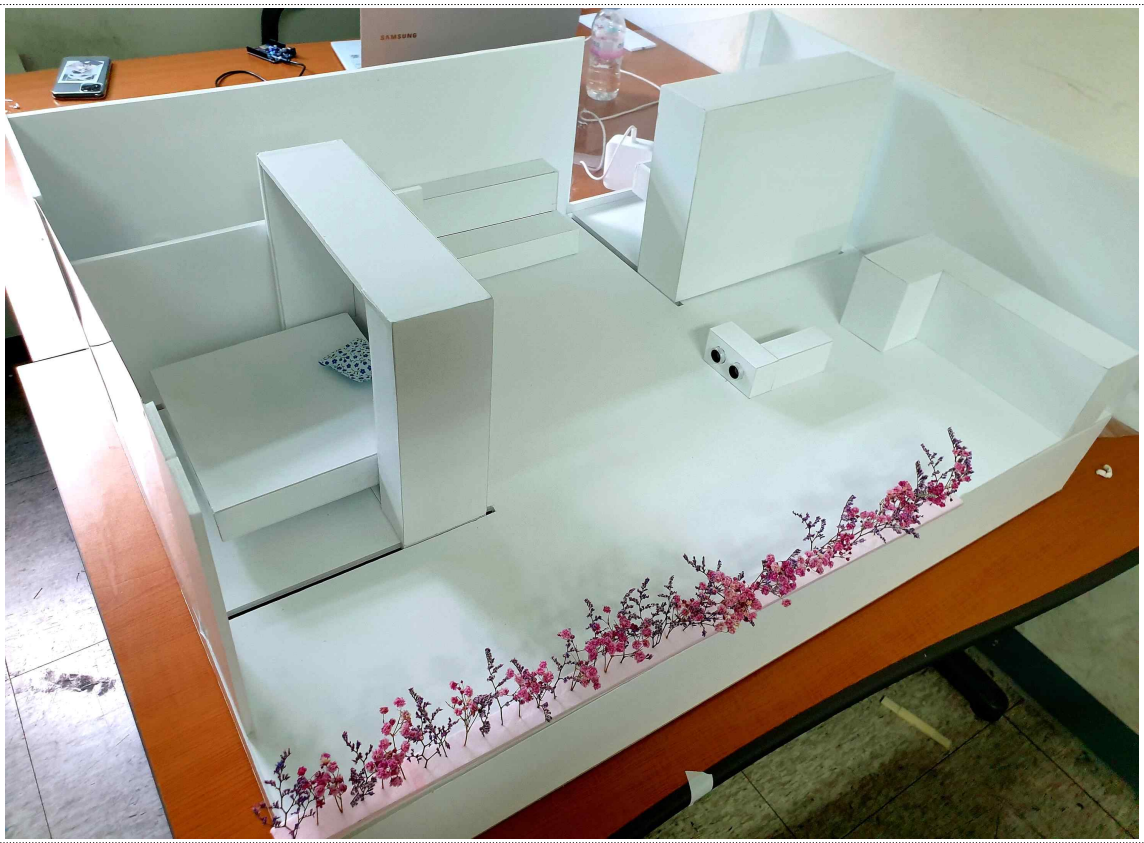
2-1) Office Closet



2-2) Create an office with Google Assistant



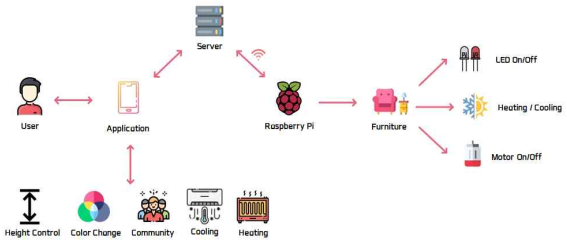
2-3) Complete office creation



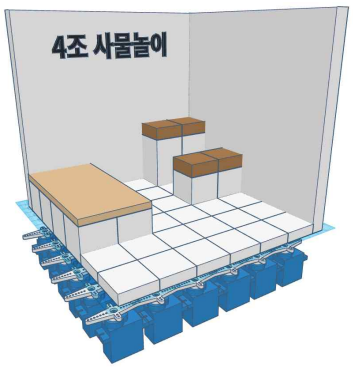
3) Completion

B. Process

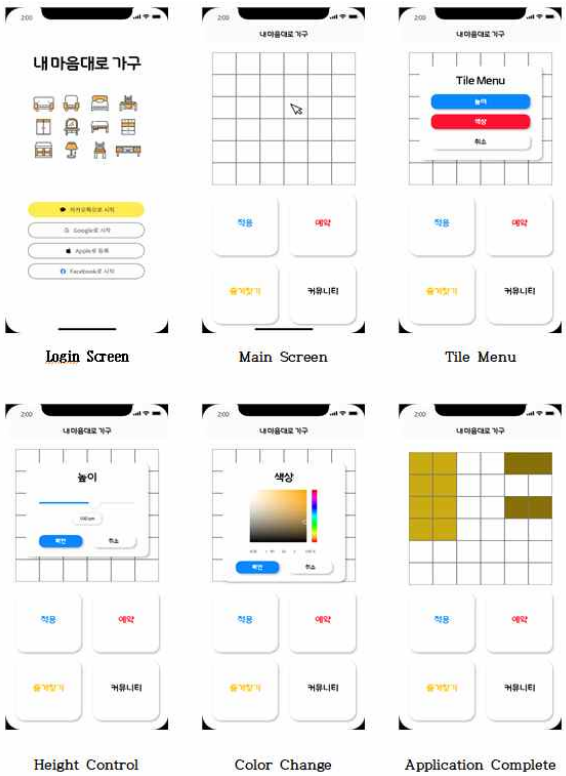
IDEA Conference



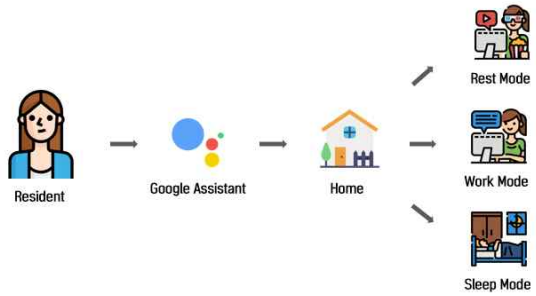
1st IDEA Overview



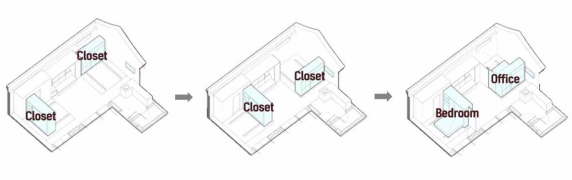
1st IDEA Expected Drawing



1st IDEA Application UI

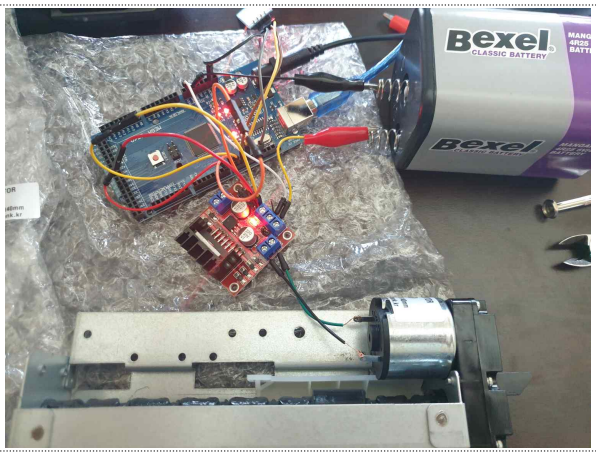


2nd IDEA Overview - IDEA Change

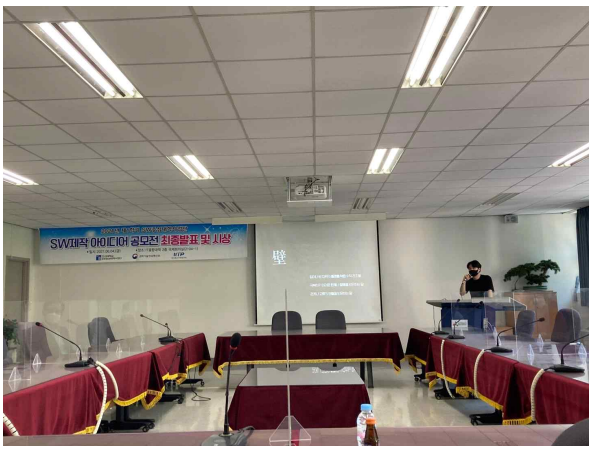


2nd IDEA Expected Drawing

Order Necessary Products



Parts check and Testing



SW IDEA Competition Participation

제 2021-2호

상 장

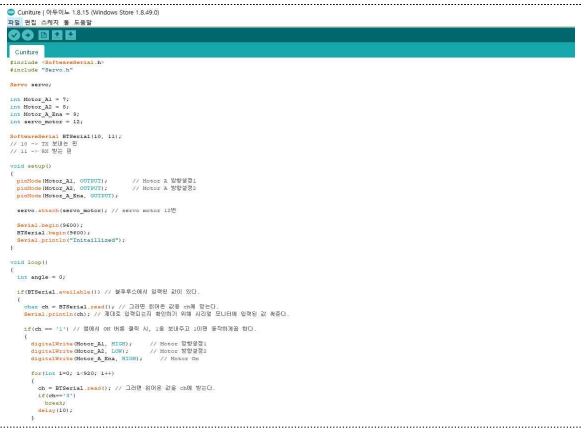
팀명: IoTainment
소속: IT융합대학 컴퓨터공학과
학번: 20154284
이름: 박민기

위 학생은 SW중심대학사업단에서 주최하는 SW제작 아이디어 공모전에서 금상을 수상하였기에 이 상장과 부상을 드립니다.

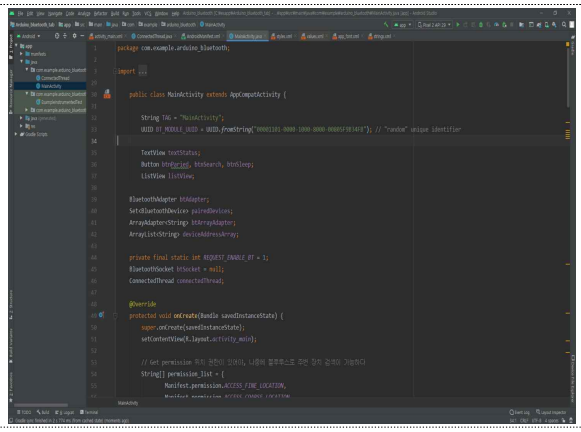
2021년 6월 4일

조선대학교SW중심대학사업단장

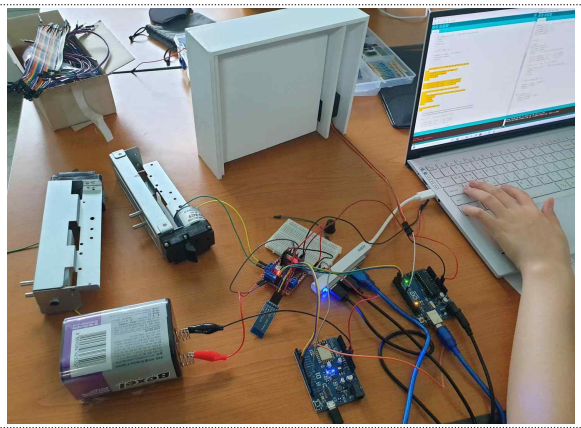
SW IDEA Competition - Gold Prize



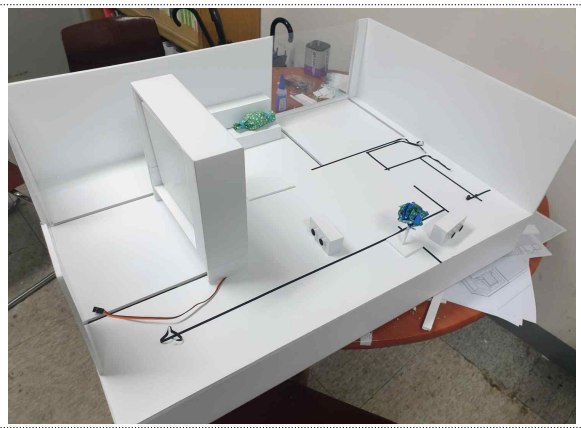
SW Development - Arduino



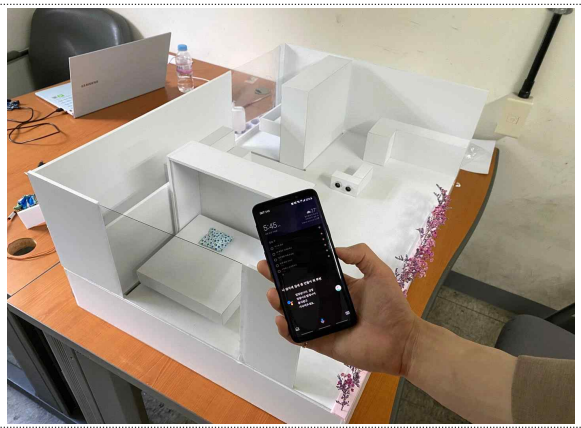
SW Development - Android Studio



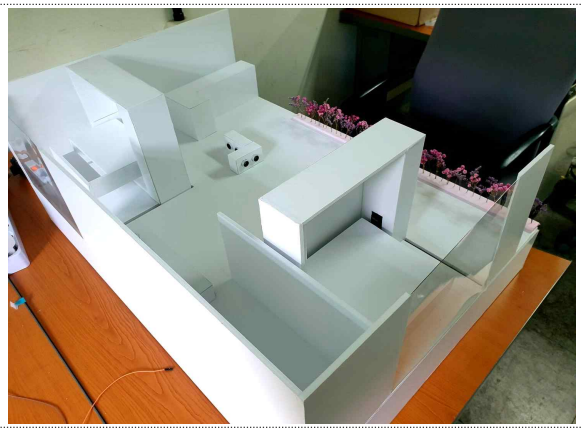
HW Development - Closet



HW Development - House Framework



Final Test



Completion

C. Expected effects and utilization measures

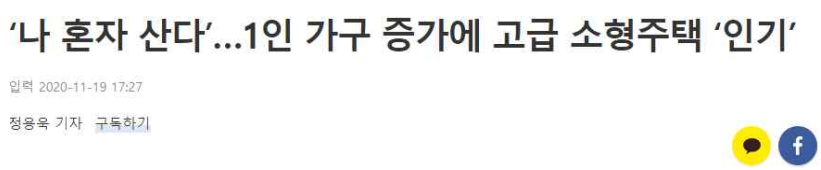
【IDEA Expectation】

① IDEA that **can satisfy the current architectural market trend**

(Source : EconomicRivew)

If a business has supply **but no demand**, it is **a failed item**. However, as I mentioned earlier, the number of single-person households is constantly increasing and the real estate market is keeping pace with the changing housing trends. In addition, as the demand for small houses increases, luxury small houses are also getting a good response from consumers.

(Source: SeoulEconomy TV)



(Source: EToday)

Small house sales were also found to be active. As a result of the analysis of 'monthly housing sales transaction status' published by the Korea Appraisal Board, the nationwide housing sales volume from January to September this year was 1.48 million. Of these, 607,917 cases, or 41.07%, were found to be small sizes of less than 60 square meters.

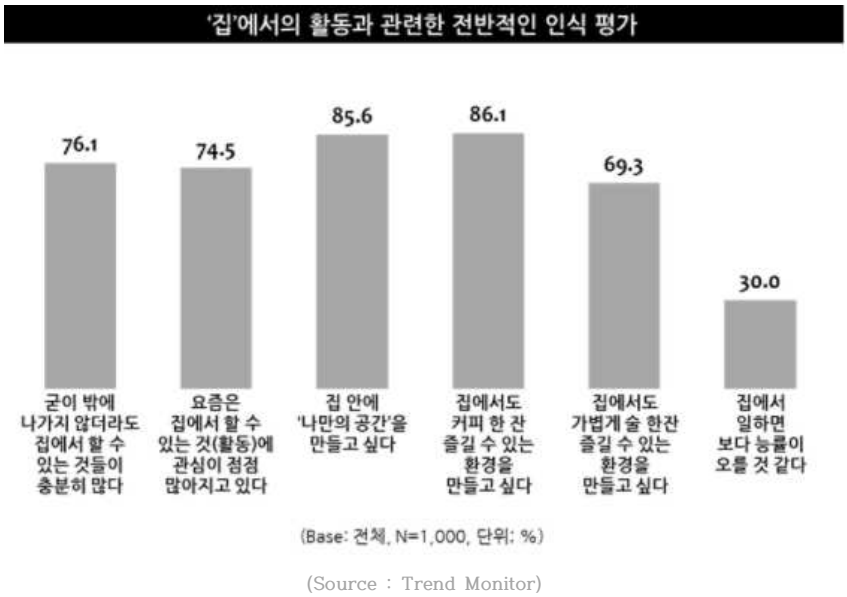
The price of small homes continues to rise. According to the real transaction price disclosure system of the Ministry of Land, Infrastructure and Transport, the 31 m^2 type of 'Samsung Hillstate 1 Complex' located in Samseong-dong, Gangnam-gu, Seoul rose by 291 million won from 970 million won in October last year to 1.198 billion won in October this year. In addition, the 49.618 m^2 type dedicated to 'Cheongdam Xi' in Cheongdam-dong, which was traded at 1.65 billion won in July last year, was changed to 1.8 billion won in June this year.

In fact, 'Pent Hill Nonhyeon', which was sold in Gangnam-gu, Seoul in October last year, **has been sold out in a short period of time, which is unusual for a high-end residential facility.**

This is the idea that **can best satisfy the trends in the construction industry.**

② IDEA to **create your own space** in the home

Due to COVID-19, which shows no sign of ending, people are still forced to work at home. When we analyzed a survey related to this, **85.6%** wanted to **create their own space** in their home. And our ideas are those that can fully satisfy **the needs** of people **who want to express their individuality.**

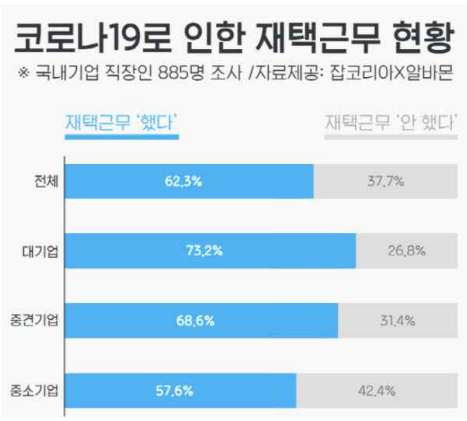


(Source : Trend Monitor)

③ IDEA to **increase the efficiency of work** in a limited space

The current situation is that **telecommuting** is **increasing in a situation** where it is difficult to go out due to the corona virus. However, humans are sensitive animals to their surroundings.

Therefore, when the bed, a resting space, and the desk, the work space, coexist, the efficiency is bound to decrease. As people answered in the statistic above, only 30% of those who said that they think they will be more efficient if they work. So, **'Cuniture'** is an idea that can provide efficient use of space to increase work efficiency even in a limited space.



(Source : JobKorea, Albamon)

④ **A space of potentia** from material labor to immaterial labor

Architect Jean Grenier said, **'The hour of night is the time of creation'.** Looking back on our history, during the industrialization period, sleep was a symbol of laziness, and a society that was hostile to laziness. However, beyond informatization, in the era of the 4th industrial revolution, industrial sites and labor that require **creativity** rather than diligence and sincerity have emerged.

The Italian philosopher, Antonio Negri, views modern labor as immaterial labor. In modern society, the hegemony of labor in the existing industry no longer has a great meaning, and the hegemony of immaterial labor has emerged. Immaterial labor leads to verbal labor and affective labor, which has **made night time for creativity important.**

In other words, his own time when he is not working, his leisure time, and his time at night has **become the time to create his own powers.** The problem is that it is difficult to even have a space of your own to be used to acquire such capabilities, and even if you have space, it is a barren environment in which you cannot have affection for the place.

Therefore, our **'Cuniture'** helps to **reduce the cost burden on space, allows each space to be configured according to the purpose in order to utilize more space,** and is an idea to **design a creative space to develop individual capabilities.**

【Future plans using the results of this project】

To be honest, We are not in the Department of Architecture. So we think that there are many difficulties in reality for us to do business with these results right now.

Therefore, we would like to submit it to **'The World Embeded Software Contest 2021'** hosted by 'the Ministry of Trade, Industry and Energy' to be recognized for the excellence of the idea, improve it, and prove that it is a really valuable idea.

D. Knowledge property of this task result (patents, utility models, trademarks, design, etc.)

Knowledge reappointment title	Applicant	Date of application	Application number

E. Knowledge of intellectual property rights plan

Consent to Application Plans	
☑ Agreement	☐ Disagree

※ Support for application of knowledge rights after evaluation of project groups according to agreed application plan

Summary of Design Studio Capstone Design Tasks

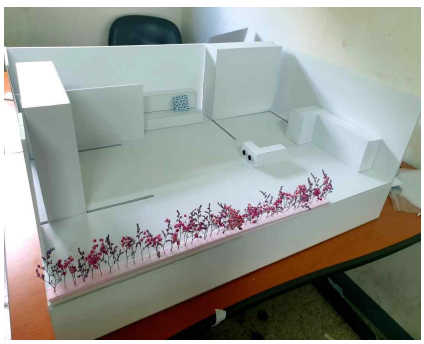
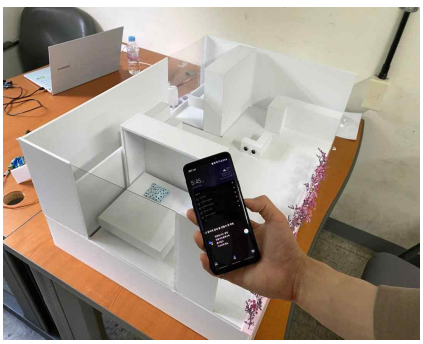
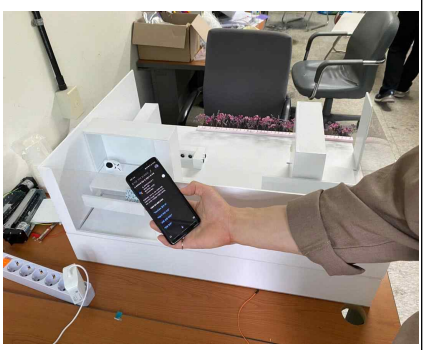
Task name	Cuniture		
Team name	IoTainment		
Coaching staff	NamHong Park		
Participating students	Minki Park Yeonghak Yu Jeongyeon Yu Minhee Hong	property rights	

Purpose

① The generalization of small houses due to the increase in the number of single-person households
② Increased Need to secure space due to COVID-19

Task contents

By moving the wall to create a new space called '**Querencia**', various environments can be created even in a limited space.


Application and Expected Effect

① IDEA that can **satisfy the current architectural market trend**
② IDEA to **create your own space** in the home
③ IDEA to **increase the efficiency of work** in a limited space
④ **A space of potentia** from material labor to immaterial labor