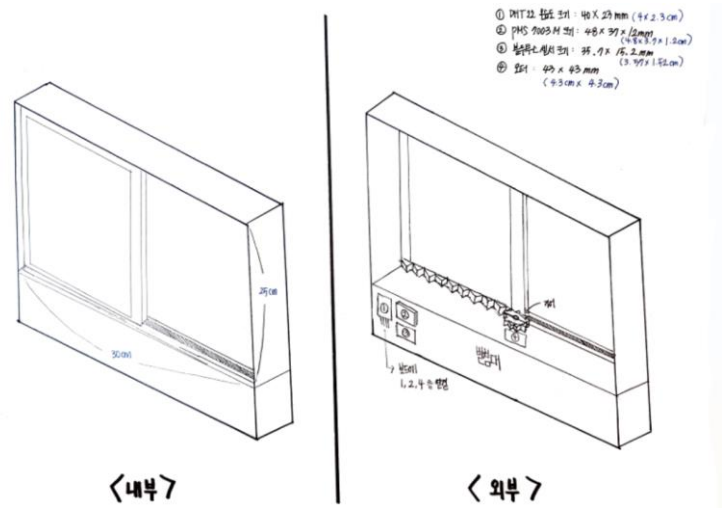
< 진행상황 최종 보고서 >

주제	날씨에 따라 옷차림을 알려주는 IOT 스마트 창문 제어 알리미
학번/이름	202345038/임채연,202345042/서지민
실습기간	**05.04~06.14**

<미세먼지 센서와 온습도 센서의 값 확인(외부)>

-외관 초기 구상도-



-외관 실제 사진-

버튼을 사용해서 모터를 사용해 창문을 열고 닫을 수 있게 하는 코드

```
#include <Stepper.h>
// Define pin numbers for motor connections
#define IN1 7
#define IN2 6
#define IN3 5
#define IN4 4

// Define pin numbers for button connections
#define BTN1 8
#define BTN2 9

// Number of steps per revolution for the motor
const int stepsPerRevolution = 2048;

// Define the number of steps to fully open/close the window
const int stepsToOpen = 5250; // Increase steps to open the
window
const int stepsToClose = -5250; // Increase steps to close
the window

// Define debounce delay time in milliseconds
```

실습내용	``` const unsigned long debounceDelay = 200; // Initialize the stepper library on pins 4 through 7 Stepper myStepper(stepsPerRevolution, IN4, IN2, IN3, IN1); // Variable to track window state bool isWindowOpen = false; // Variables to store the last debounce time for each button unsigned long lastDebounceTime1 = 0; unsigned long lastDebounceTime2 = 0; void setup() { // Initialize serial communication Serial.begin(9600); // Set button pins as input with internal pull-up resistors pinMode(BTN1, INPUT_PULLUP); pinMode(BTN2, INPUT_PULLUP); // Set the speed of the stepper motor (RPM) myStepper.setSpeed(15); } void loop() { // Get the current time in milliseconds unsigned long currentTime = millis(); // Check if BTN1 is pressed and debounce it if (digitalRead(BTN1) == LOW && (currentTime - lastDebounceTime1 > debounceDelay)) { lastDebounceTime1 = currentTime; // Update the last debounce time for BTN1 if (!isWindowOpen) { Serial.println("Opening window..."); myStepper.step(stepsToOpen); // Rotate to open the window isWindowOpen = true; // Update window state to open } } // Check if BTN2 is pressed and debounce it if (digitalRead(BTN2) == LOW && (currentTime - lastDebounceTime2 > debounceDelay)) { lastDebounceTime2 = currentTime; // Update the last debounce time for BTN2 ```

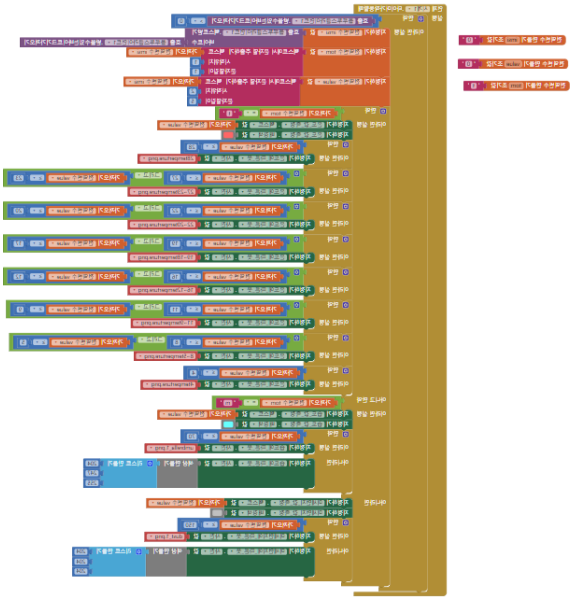
```
    if (isWindowOpen) {
      Serial.println("Closing window...");
      myStepper.step(stepsToClose); // Rotate to close the
window
      isWindowOpen = false; // Update window state to closed
    }
  }
}

ㄴ
```

미세먼지, 온습도 코드

```
#include <DHT.h>
#include <SoftwareSerial.h>
#include <Stepper.h>

const int stepsPerRevolution = 2048;
Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11); //
스텝모터 핀 설정
DHT mydht(A0, DHT22); // 핀을 A0 에 연결
SoftwareSerial HC06(2, 3); // TX : 2 번, RX : 3 번

// GP2Y1014AU 미세먼지 센서 핀
const int dustSensorPin = A1;
const int ledPower = 7;

void setup() {
  Serial.begin(9600); //PC-아두이노 간 통신라인
  HC06.begin(9600); // 아두이노-블루투스 모듈 간 통신라인
  mydht.begin();
  pinMode(dustSensorPin, INPUT);
  pinMode(ledPower, OUTPUT);
  myStepper.setSpeed(15); // 스텝모터 속도 설정
}

void loop() {
  // 온도 읽기
  float temperature = mydht.readTemperature();
  HC06.println("t" + String(temperature));
  delay(1000);

  // 습도 읽기
  float humidity = mydht.readHumidity();
  HC06.println("m" + String(humidity));
```

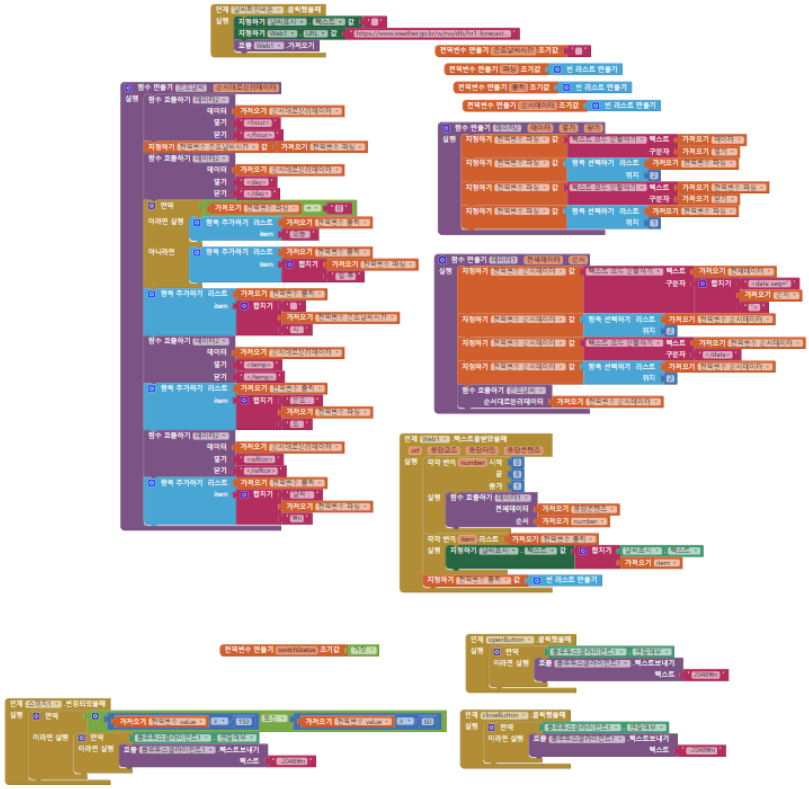
```
  delay(1000);

  // 미세먼지 읽기
  digitalWrite(ledPower, LOW); // LED 켜기
  delayMicroseconds(280);

  int dustValue = analogRead(dustSensorPin);
  delayMicroseconds(40);

  digitalWrite(ledPower, HIGH); // LED 끄기
  delayMicroseconds(9680);

  float voltage = dustValue * (5.0 / 1024.0); // 전압 변환
  float dustDensity = 0;

  // 먼지 밀도 계산
  if (voltage >= 0.6) {
    dustDensity = (voltage - 0.6) * 1000 / 5.0 * 0.5; //
mg/m³로 변환
  } else {
    dustDensity = 0; // 음수 값을 방지하기 위해 0으로 설정
  }

  // 미세먼지 농도에 따른 문자열 표시
  String airQuality;
  if (dustDensity <= 30) {
    airQuality = "좋음";
  } else if (dustDensity <= 80) {
    airQuality = "보통";
  } else if (dustDensity <= 150) {
    airQuality = "나쁨";
  } else {
    airQuality = "매우나쁨";
  }

  HC06.println("d" + String(dustDensity));
  delay(1000);

  // 블루투스 데이터 읽기 및 모터 제어
  if(HC06.available()){
    String data = HC06.readStringUntil('\n');
    int steps = data.toInt();
    Serial.println("Moving stepper by " + String(steps) + "
steps.");
    myStepper.step(steps);
  }
}
```

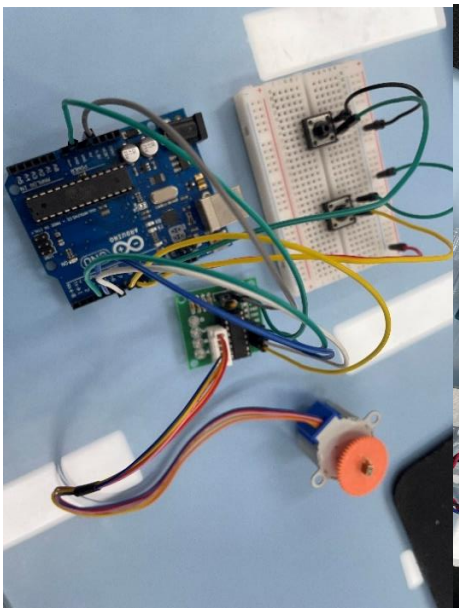
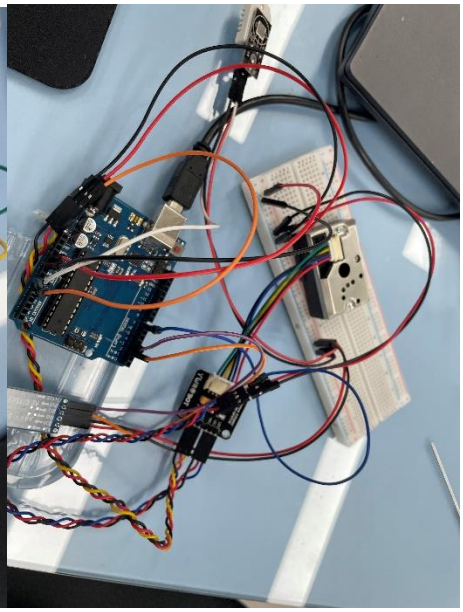
온습도 , 미세먼지 블록 코딩



블루투스 블록 코딩

API 날씨 정보 블록코딩 (스위치1, open, close 블록은 push button으로 대체함)



실습결과	버튼을 이용해 모터 작동  앱인벤터 결과 	미세먼지,온습도 


날씨에 따라 옷차림을 알려주는
IOT 스마트 창문 제어 알리미
창문을 통해 날씨에 따른 옷차림을 알 수 있어요!
인하공업전문대학
2023145038 임채연
2023145042 서지민


날씨에 따라 옷차림을 알려주는
IOT 스마트 창문 제어 알리미
- 미추홀 구의 날씨를 알려줘요!
수동제어
자동제어
- 자동제어 시, 습도가 60%, 미세먼지가 150이상일 때 닫아줘요!
온도 : 29.30 ℃
습도: 71.20 %
농도: 181.7
오늘의 옷차림은?
28℃ 이상
민소매, 반팔, 반바지 얇은 치마, 린넨 옷
온습도 센서
미세먼지 센서
블루투스 센서
스텝모터
- 습도 값이 60이상일 때, 그림이 나타나요!
- 미세먼지가 150(나쁨)이상일 때, 그림이 나타나요!