



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

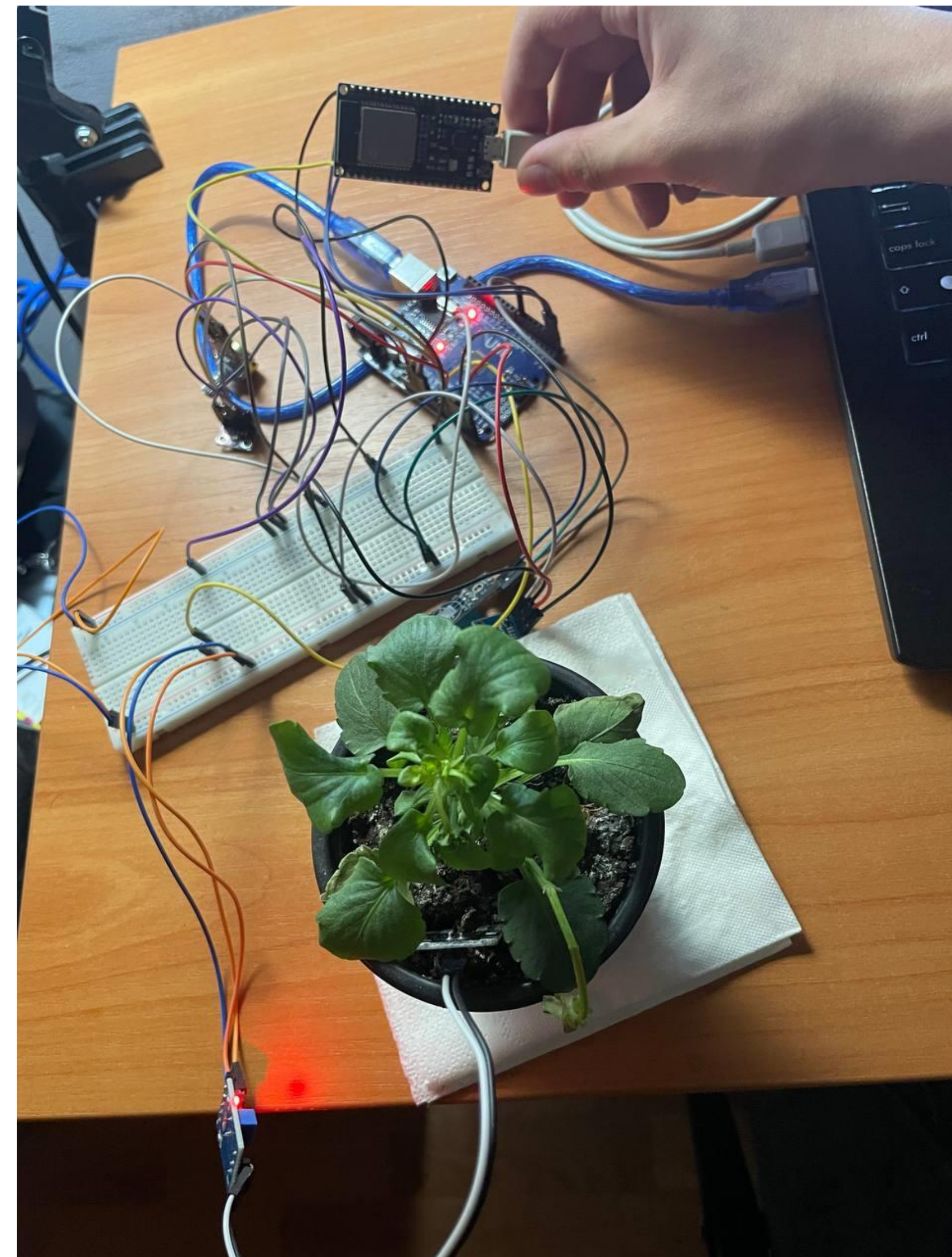
ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС ДЛЯ ОБРАБОТКИ ДАННЫХ КЛИМАТИЧЕСКИХ СЕНСОРОВ

Автор: Кайралап Досбол
студент 4 курса ФКН ПМИ 197

Руководитель: Попов Виктор Юрьевич
Профессор, департамент бизнес-информатики

Москва, 2022

ОПИСАНИЕ ПРОЕКТА



ТЕРМИНЫ, ОПРЕДЕЛЕНИЯ

1. Микроконтроллер (или просто плата) — небольшая плата с собственным процессором и памятью.
2. Arduino — аппаратно-программные средства для построения систем автоматики и робототехники.
3. Сенсор (датчик) - средство измерения физического, химического параметра, преобразующее контролируемый параметр в удобный для использования сигнал, как правило электрический.
4. Arduino IDE [4]— это программная среда разработки, использующая C++ и предназначенная для программирования всех плат ряда Ардуино (Arduino).
5. Скетч (sketch) - обозначение кода (программы) для Arduino. Это единичный проект, который загружается и выполняется платой. В данной работе я регулярно буду пользоваться именно этим термином.

ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

Работа состояла из 4 частей: сборка цепи, написание скетча, создание приложения и сервера.

Для реализации проекта требовалось выполнить ряд задач:

- 1) изучить научную литературу и существующие аналоги;
- 2) определить технологии, которые будут использованы при разработке программы для микроконтроллера;
- 3) построить модель цепи (например на платформе tinkercad);
- 4) приобрести все необходимые детали (плата Arduino Uno, датчики, резисторы, провода и прочее) и собрать цепь;
- 5) написать код для микроконтроллера;
- 6) создать приложение на Android, сделать удобный интерфейс;
- 7) собрать и запустить в облаке сервер с постоянным ip;
- 8) организовать передачу данных с Arduino на телефон с помощью Bluetooth и WiFi;
- 9) протестировать систему и произвести отладку программы;
- 10) подготовить отчёт и необходимую документацию

АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ

- Уникальная система по уходу за растениями от Koubachi
- Life Control
- Умная теплица от Clover Tech



ВЫБОР ТЕХНОЛОГИЙ И СРЕДСТВ РАЗРАБОТКИ

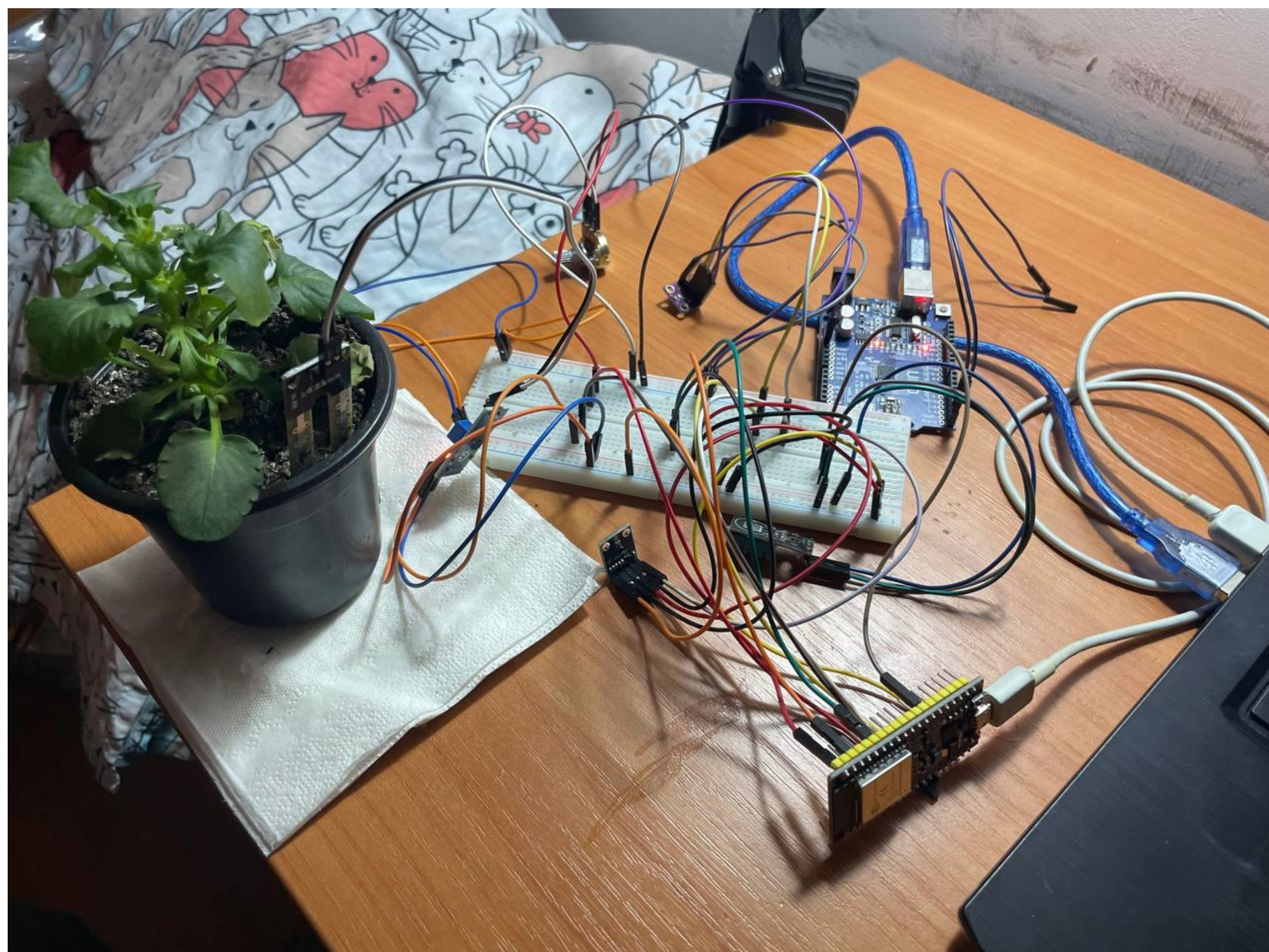
Воспользовался официальным софтом Software Arduino IDE 1.8.15 для написания и отладки кода. Отсюда по проводу можно загрузить код на плату и следить за выводом данных с датчиков с помощью монитора порта.

Использовал 7 модулей:

1. GY-BME P 280 - сенсор температуры, давления и влажности воздуха;
2. BH1750 - цифровой люксметр (датчик освещения);
3. 080 МН - датчик влажности почвы;
4. DHT 11 - датчик влажности воздуха, но также измеряющий температуру;
5. BK32-31 - модуль передачи данных по bluetooth, это китайский аналог модуля HC05;
6. B10K - потенциометр;
7. ESP32 WROOM 32D - это WiFi + Bluetooth модуль передачи данных

АРХИТЕКТУРА ПРОЕКТА

Аппаратная часть



ПЕРВАЯ ОНЛАЙН ВЕРСИЯ ПРОЕКТА

показывают критичность температуры

автополив растения

датчик температуры

датчик освещения

ночное освещение

экран показывает время работы и температуру

```
1 #include <Servo.h>
2 #include <LiquidCrystal.h>
3
4 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
5 double poliv_time = 15;
6 double val, temper, Temp, time1, time2, poliv = poliv_time / 3;
7 int i = 1, pos = 0;
8 Servo servo_6;
9
10 //функция возвращает температуру с датчика
11 double get_temp() {
12     temper = (double) (analogRead(A1));
13     temper = ((temper - 100) / 8);
14     return temper + 10;
15 }
16
17 void setup() {
18     Serial.begin(9600);
19     servo_6.attach(6, 500, 2500);
20
21     //подключаем пины
22     pinMode(A0, INPUT);
23     pinMode(A1, INPUT);
24     pinMode(11, OUTPUT);
25     pinMode(6, OUTPUT);
26     pinMode(10, OUTPUT);
27     pinMode(7, OUTPUT);
28     pinMode(8, OUTPUT);
29     pinMode(9, OUTPUT);
30
31     //выводим на экран постоянные символы
32     lcd.begin(16, 2);
33     lcd.setCursor(0, 0);
34     lcd.print("Time:");
35     lcd.setCursor(0, 1);
36 }
```

Монитор последовательного интерфейса

Онлайн модель первой версии проекта (с кодом) на платформе <https://www.tinkercad.com/things/ketjWcIihep-proekt-umnoe-rastenie/editel>

СЕРВЕРНАЯ ЧАСТЬ - СКЕТЧ (КОД)

```
void mytemp() { //температура
    float t = dht.readTemperature();
    float t2 = bme.readTemperature();
    Serial.print("Температура = ");
    Serial.print(t);
    Serial.println(" *C");
}
void myillum() { //освещённость
    float osv = lightMeter.readLightLevel();
    Serial.print("Освещение: ");
    Serial.print(osv);
    Serial.println(" lx");
}
void mypress() { //давление в гексапаскалях
    float davlen = bmp.readPressure();//bmp, bme
    Serial.print("Давление = ");
    Serial.print(davlen / 100.0);
    Serial.println(" hPa");
}
```

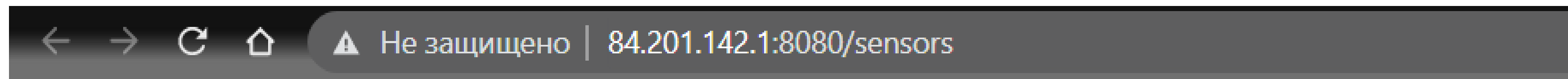
```
144 HTTPClient http; //Объявить объект класса HttpClient
145 WiFiClient client;
146 http.begin(client, "http://84.201.142.1:8080/sensors/update");//Укажите адрес запроса
147
148 http.addHeader("Content-Type", "application/json");//application/json //"/update"
149 int httpPost = http.POST(Vivod);
150
151 Serial.println(httpPost);
152 String payload2 = http.getString();
153 Serial.println(payload2);
```

Скетч - это код, написанный на Arduino IDE и загруженный непосредственно на микроконтроллер. Скетч представляет из себя код написанный на языке Arduino, разбитый на блоки с удобными комментариями (каждый блок для отдельного датчика).



СЕРВЕРНАЯ ЧАСТЬ - ОБЛАЧНЫЙ СЕРВЕР

Адрес сервера и json данные



```
{"temperature":26,"illumination":49.17,"pressure":101.4,"humidity":60,"soilMoisture":50,"time":""}
```

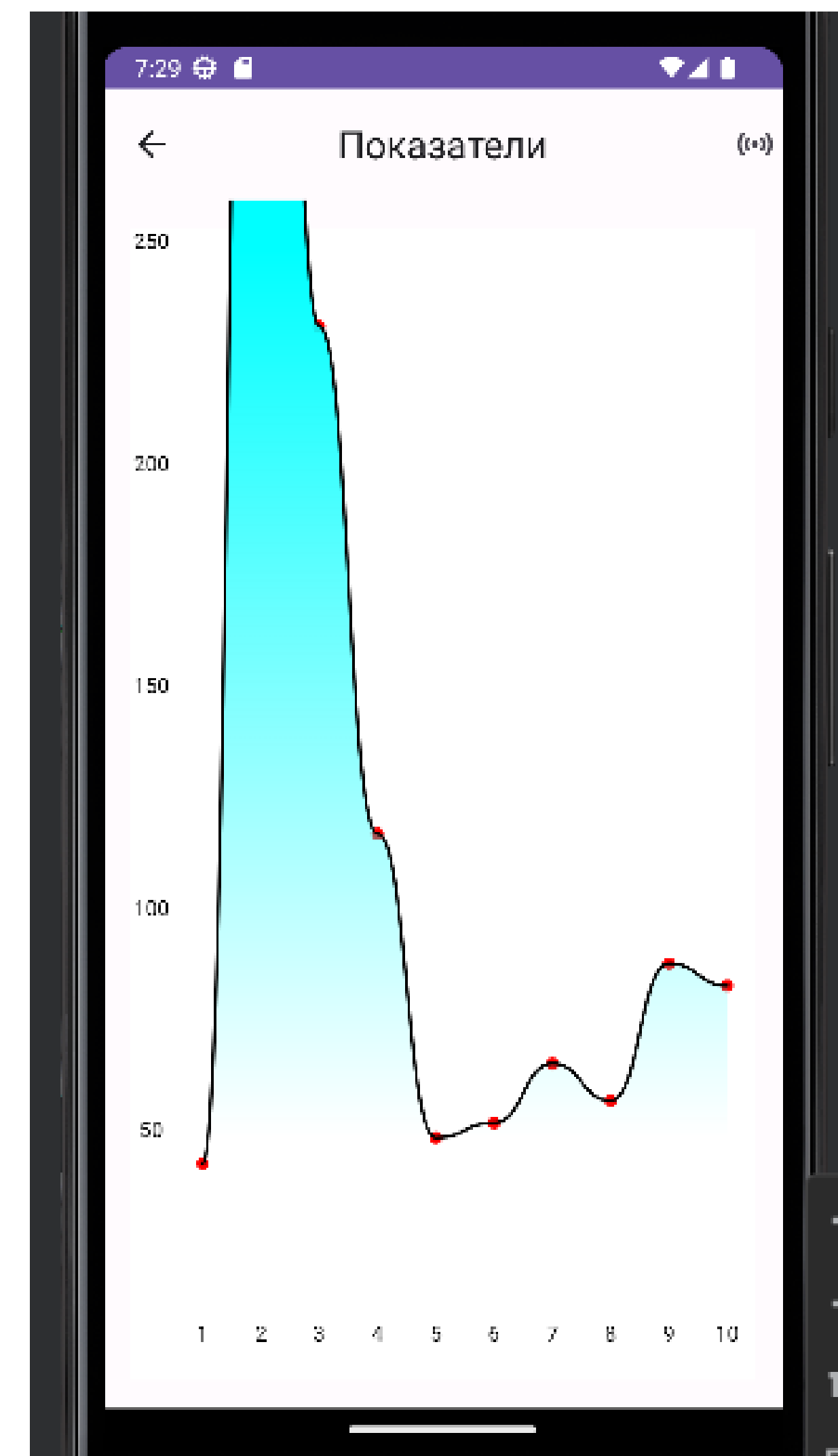
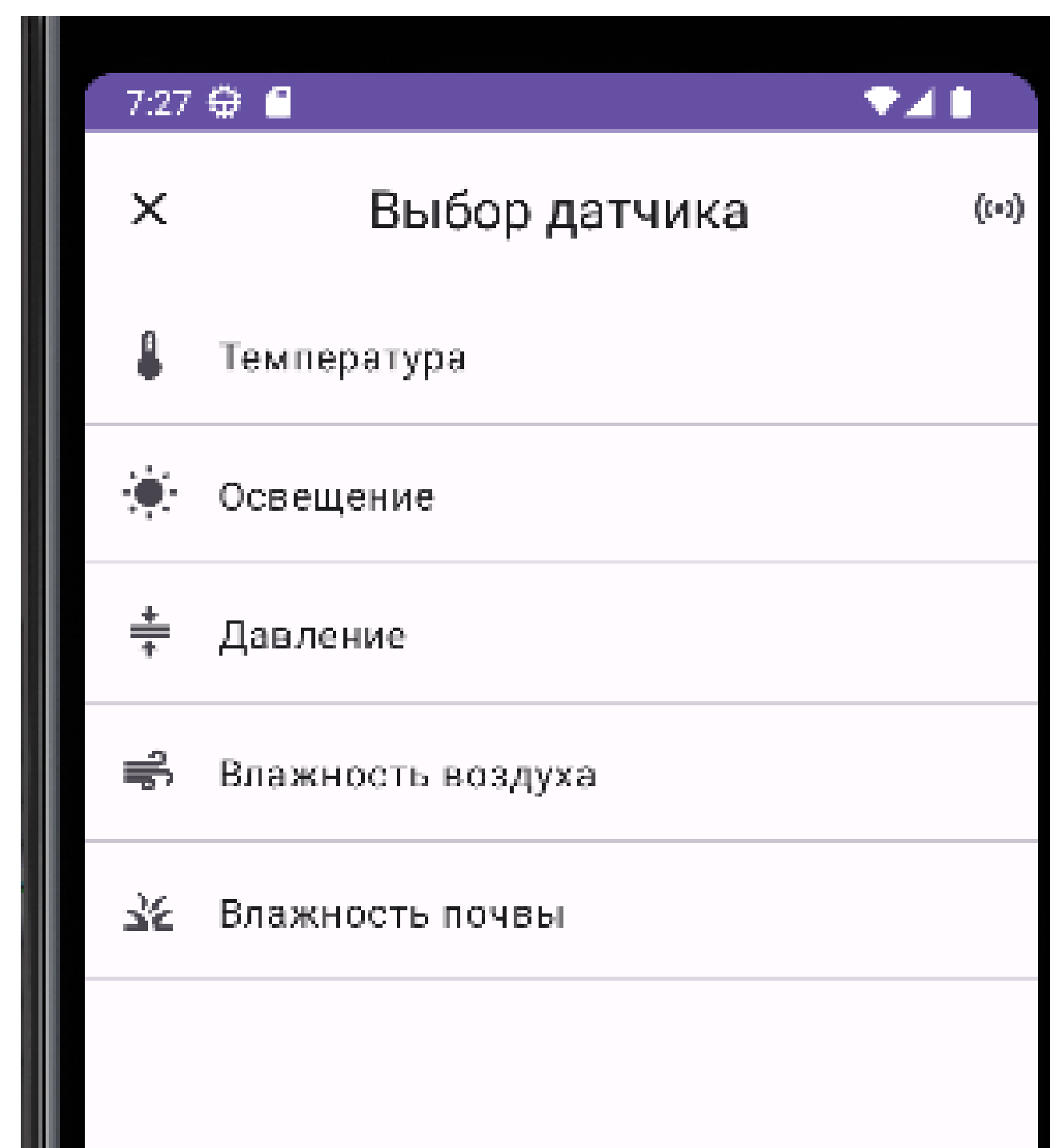
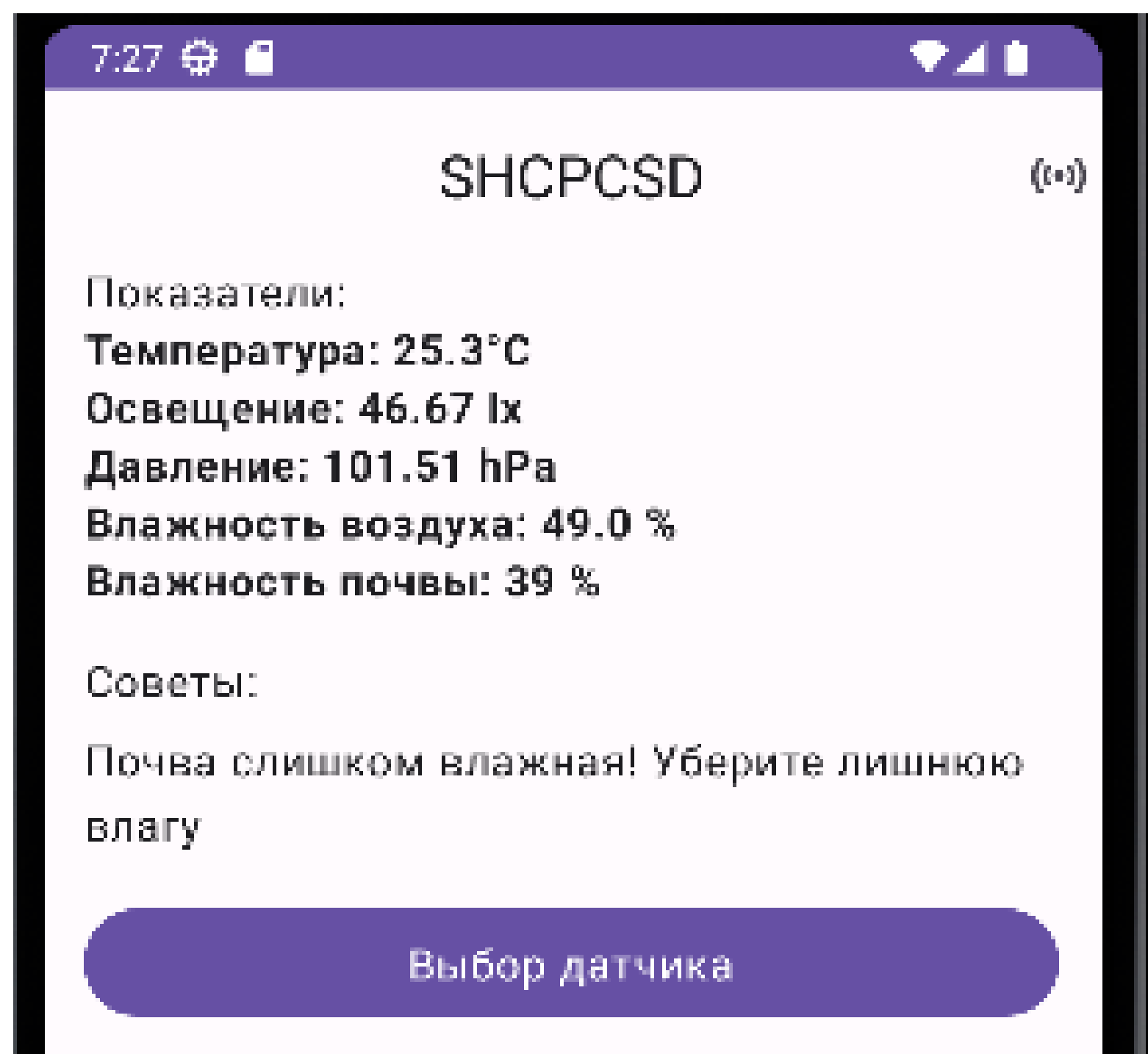
Yandex cloud

<input type="checkbox"/>	Имя	Статус	ОС	Платформа	vCPU	Доля vCPU	RAM	Прерываемая	Размер дисков	Внутренний IPv4	Публичный IPv4
<input type="checkbox"/>	dudos	Running		Intel Ice Lake	2	50 %	2 ГБ	да	8 ГБ	10.129.0.11	84.201.142.1

Часть кода сервера,
отвечающая за
обновление данных

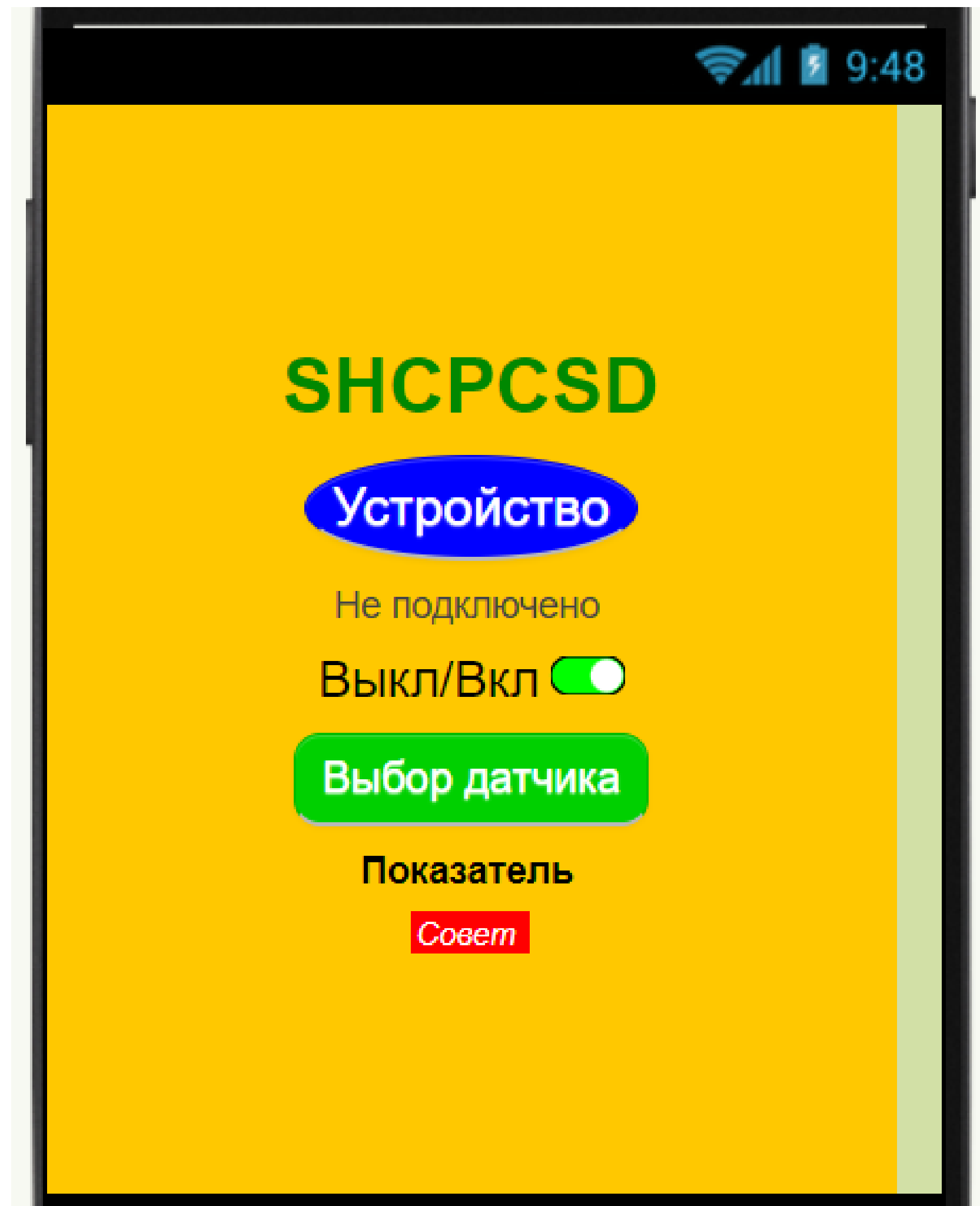
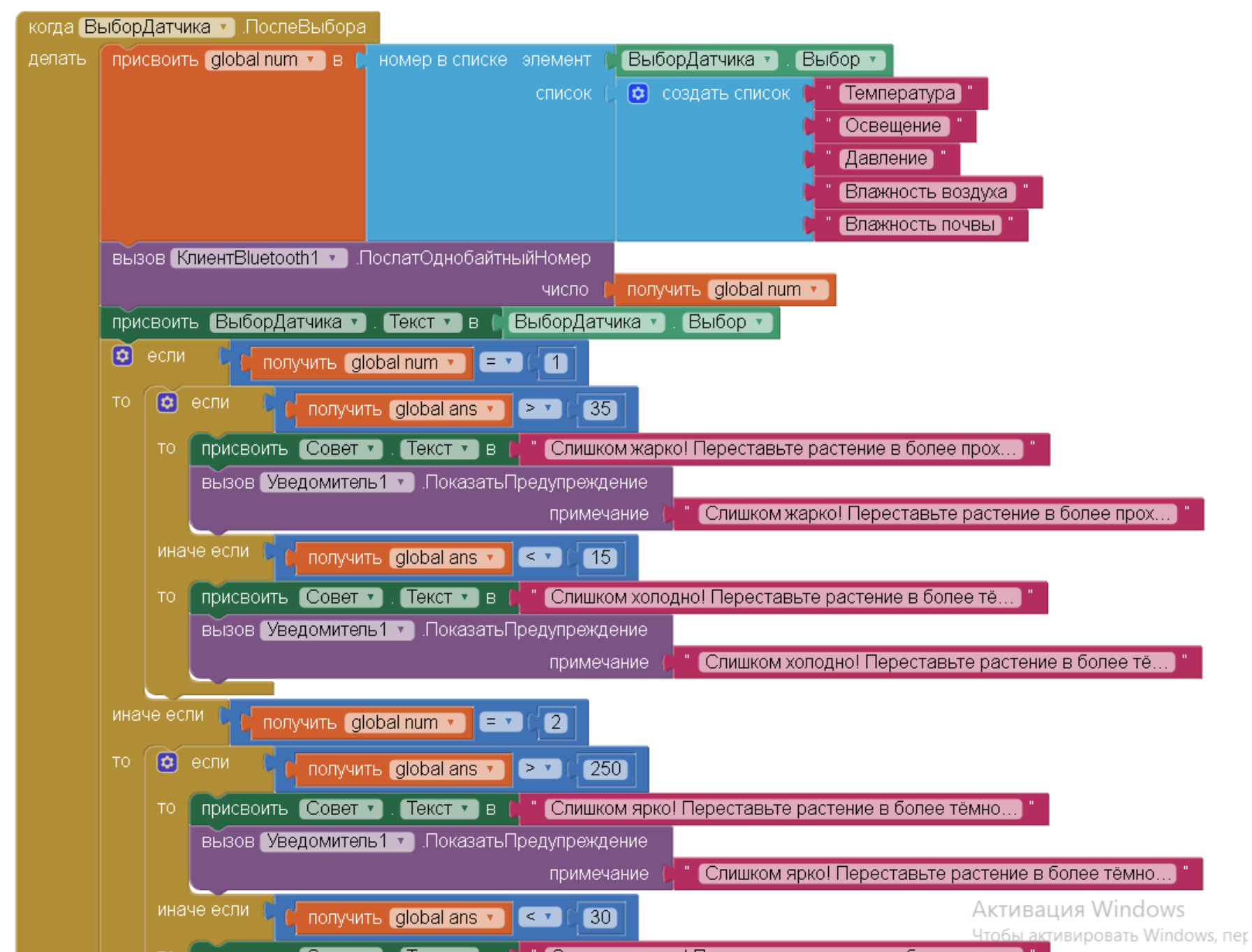
```
func updateSensors(w http.ResponseWriter, r *http.Request) { 1 usage
    body, err := ioutil.ReadAll(r.Body)
    var ans Sensors
    err = json.Unmarshal(body, &ans)
    if err != nil : err *
    sensors = ans
    dt := time.Now()
    writeIntoFile(fmt.Sprintf("#{ans.Temperature},#{ans.Illumination},#{a
    log.Println(ans)
    return
}
```


ПРИЛОЖЕНИЕ



РЕАЛИЗАЦИЯ BLUETOOTH ПРИЛОЖЕНИЯ

Для создания приложения использовал очень удобный сервис с понятным интерфейсом и простой сборкой - MIT app inventor. Интерфейс приложения собирается по-блочно, так же как и сама логика исполнения. Полную реализацию приложения можно найти в репозитории.





НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Почта: dkayralap@edu.hse.ru

Телефон.: +7 9151304015

Кайралап Досбол ПМИ197

Программно-аппаратный комплекс для обработки данных климатических сенсоров