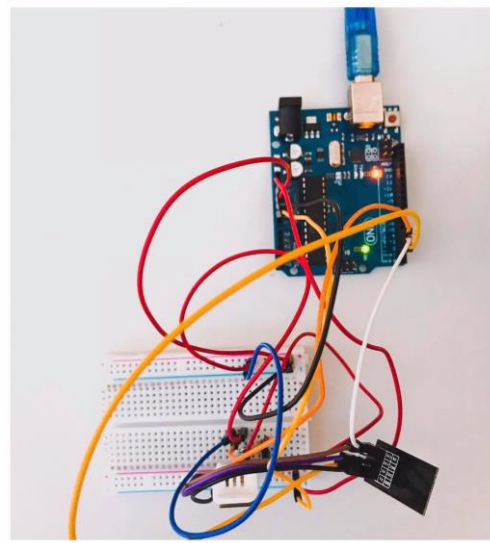
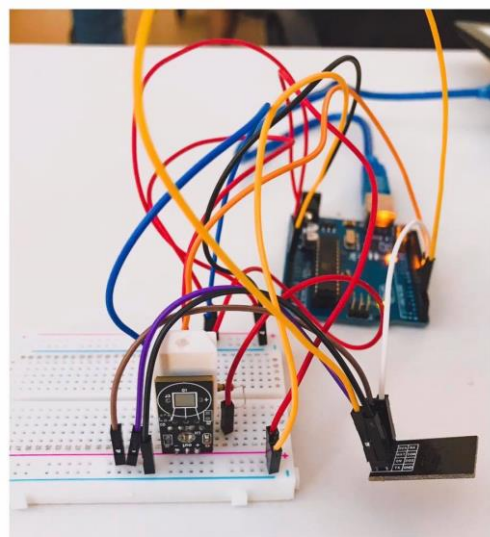


# МЕТЕОРОЛОГИЧНА СТАНЦИЯ ЗА ПЧЕЛИНИ С ARDUINO UNO

---



*Изготвили:*  
*Дуйгу Дуран / Николай Димитров / Десислава Бранимирова*

---

# Съдържание:

- ⑧ Описание на проекта -  
Метеорологична станция за пчелини с  
Arduino UNO
- ⑧ Съставни части
- ⑧ Блокова схема
- ⑧ Електрическа схема
- ⑧ Сглобяване на метеорологичната  
станция
- ⑧ Програмиране на  
метеорологичната станция /сорс код/
- ⑧ Заключение

# Описание на проекта – Метеорологична станция за пчелини с Arduino UNO

Метеорологична станция за пчелини е функционалност, която добавихме към нашето мобилно приложение - дневник за пчеларя (My Bees Diary - <https://github.com/Software-Maniacs/MyBeesDiary/tree/master/Bees%20Diary/My%20Bees%20Diary/My%20Bees%20Diary>). Тя осигурява набиране на представителни данни за температурата и влажността на въздуха за целия принадлежащ регион на съответния пчелин. Измерването се извършва с помощта на DHT22 сензор. То отчита температура от -40° до 80° C и относителна влажност на въздуха в диапазон 0% – 99%. Има предварително калибриран цифров изход и се отличава с голяма надеждност и стабилност. Сензорът е монтиран върху платка с размери 40mm x 15,4mm. Разполага с троен интерфейсен конектор и монтажен отвор. Захранва се и работи с напрежение 3,3V - 5V.

Нашият проект е прототип на метеорологична станция, защото за осъществяването на комуникацията между сървъра и устройството, използвахме Wi-Fi модул ESP8266. Той заедно със сензора е свързан към Arduino UNO платка. Модулът е изграден с Wi-Fi 32-битов микроконтролер ESP8266 - datasheet. Има сериен интерфейс, два GPIO порта, допълнителна Flash памет и вградена антена. Платката е с размери 25mm x 15mm, с монтиран от долната ѝ страна интерфейсен конектор, на който са изведени портовете за захранване и комуникация.

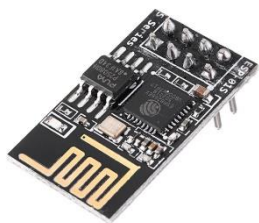
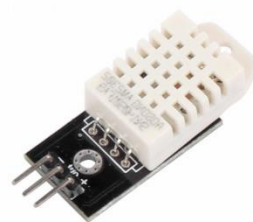
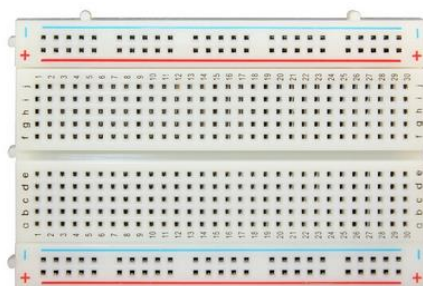
За реализацията на метеорологичната станция за пчелини много по-удачен вариант е използването на LoRaWAN устройство, вместо Wi-Fi модул, защото е предизвикателство създаването на висококачествено Wi-Fi мрежово покритие на труднодостъпни, отдалечени локации, където се намират в повечето случаи пчелините. За сметка на това LoRaWAN устройствата осигуряват много добро покритие на труднодостъпни места, захранващите батерии имат дълъг живот (устройствата могат да издържат дълга експлоатация с една батерия достигаща в много от случаите 10+ години). Също така предоставят възможности за евтина геолокации на обектите и относително ниски инвестиционни разходи за изграждане на LoRaWAN мрежа.



# Съставни части

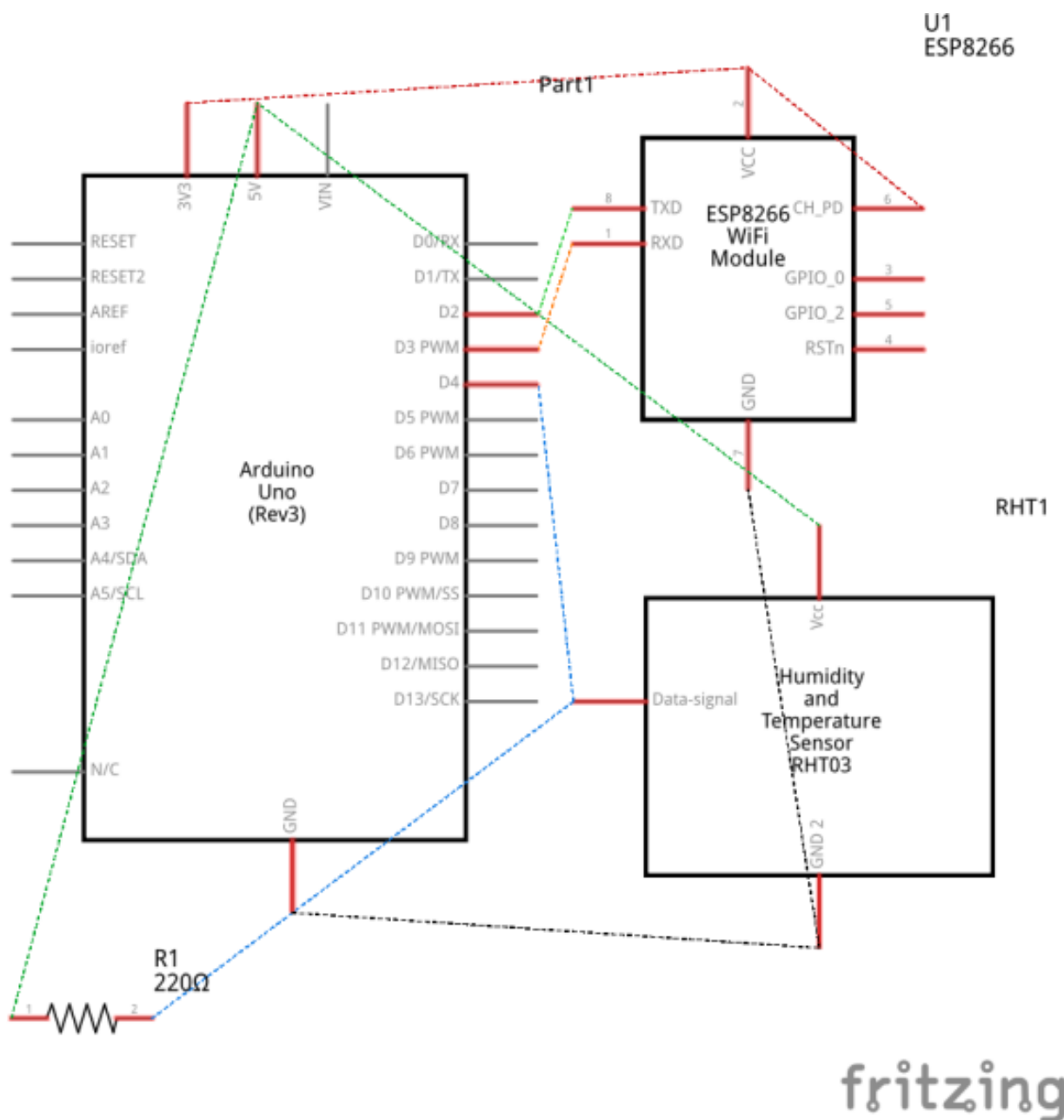
В изработката на проекта са използвани следните части:

- ∞ Arduino UNO платка
- ∞ Прототипна платка Breadboard с 400 гнезда
- ∞ Сензор за влажност и температура - DHT22, модул
- ∞ Wi-Fi модул ESP8266
- ∞ Проводници с конектори мъжки - женски и мъжки – мъжки
- ∞ Резистор – 220  $\Omega$

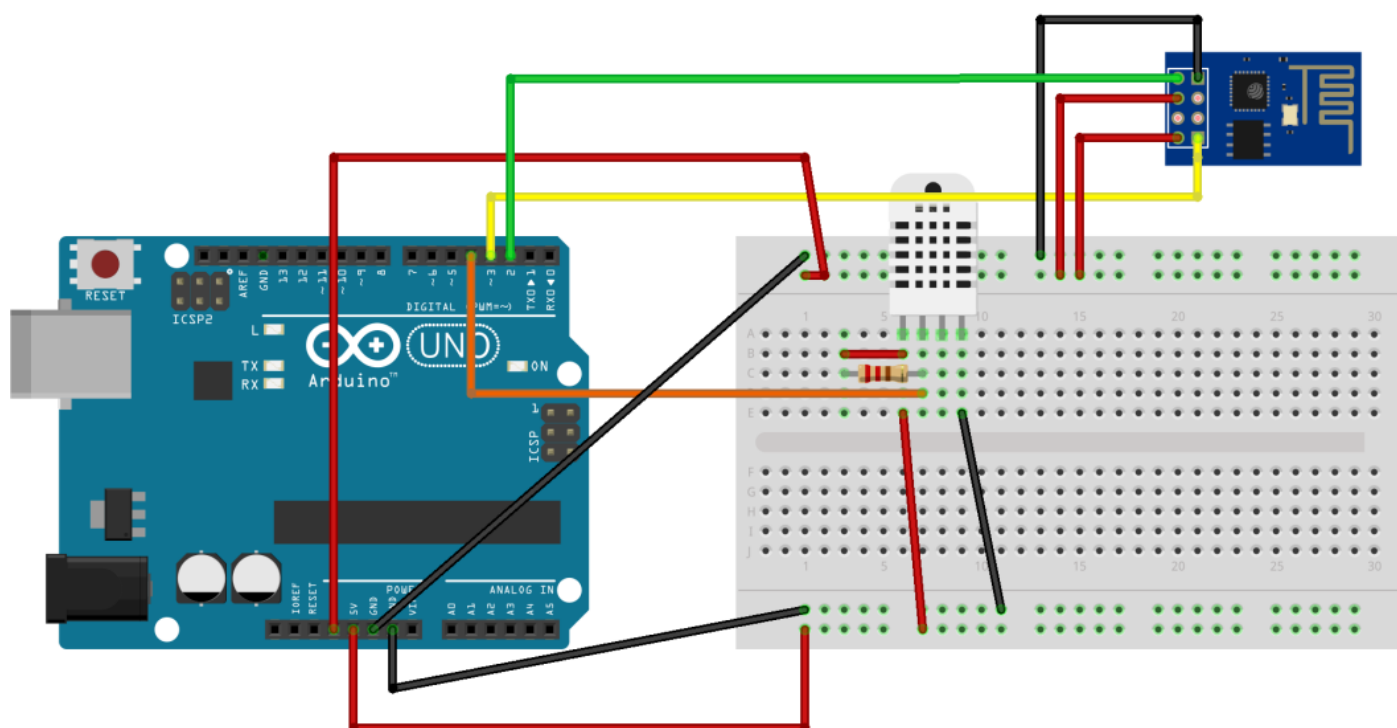




# Блокова схема



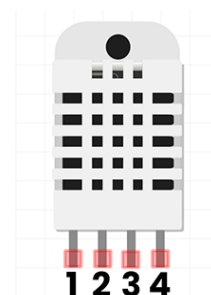
# Електрическа схема



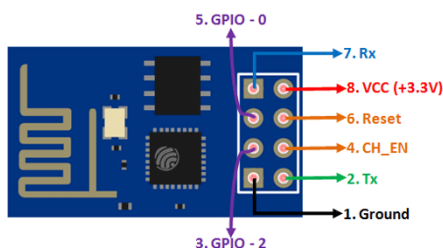
fritzing

# Сглобяване на метеорологическата станция

DHT22 сензорът има 4 пина като първият от тях е захранващият пин (VCC), който с помощта на Breadboard, свързваме с 5V захранващия пин на Arduino UNO платката. Вторият пин на сензора (DATA/SDA) е свързан към микроконтролера и е предназначен за прехвърляне на данни от сензора към контролера. Той се свързва с дигиталния пин (в нашия случай – пин D4) на платката, за да се осъществи обмяна на информация. Чрез него ние можем както да четем, така и да извеждаме информация. Третият пин на сензора (NC), познат още като „no connection pin“, остава несвързан. Четвъртият пин (GND) е заземителният. Той се свързва към GND пина на платката.



Wi-Fi модулът ESP8266 има 8 пина. Първият пин е заземителният (GND), който с помощта на Breadboard, се свързва към GND пина на платката. Вторият пин (TX) се свързва с дигиталния пин (D2) на платката и чрез него се осъществява качването на написаната програма. Четвъртият пин (CH\_EN) осигурява комуникацията между устройствата. Той заедно с осми пин (VCC) се свързват с 3,3V захранващия пин на Arduino UNO. Седми пин (RX) се свързва към ШИМ пина (D3) на платката, преобразуваща дигиталните сигнали в аналогови. Така се осъществява пълния трансфер на информация.



**\*Забележка:** Добре е ESP8266 да се свързва чрез регулатор на напрежението, понеже работи на 3,3V и съществува риск от неподходящо свързване с Arduino. Част от чиповете продължават да работят дори и на 5V, но за някои от тях по-високото напрежение може да доведе до повреда.

ПИН	Arduino UNO	5V	D4	GND
	DHT22	VCC	DATA/SDA	GND

ПИН	Arduino UNO	GND	D2	3,3V	3,3V	D3
	ESP8266	GND	TX	CH_EN	VCC	RX

# Програмиране на метеорологичната станция

## /сорс код/

```
#include <ESP8266WiFi.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>

#define DHTPin D4
#define DHTTYPE DHT22

DHT dht(DHTPin, DHTTYPE);

const char[] ssid = "myNetworkName";
const char[] password = "myNetworkPassword";

int status = WL_IDLE_STATUS;

char server[] = "www.petar-petrov.com"

String postData1;
String postData2;
String postVariable1 = "Temperature(°C)= ";
String postVariable2 = "Humidity in % = ";

WiFiClient client;
```

Скицата започва с добавянето на **ESP8266WiFi.h** библиотека. Тя предоставя специфични Wi-Fi методи, които викаме, за да се свържем с мрежата. Следващите библиотеки са предназначени за **DHT22** сензора и ни позволяват четенето на информация от него.

След това дефинираме номера на пина, към който е свързан сензора (в нашия случай - дигитален пин 4) и типа на сензора, след което следва инициализацията на сензора.

Докато конфигурираме **ESP8266** в STA режим, той се присъединява към съществуващата Wi-Fi мрежа. Следователно, ние трябва да му предоставим SSID и паролата на нашата мрежа и създаваме променлива за статуса ѝ. След това запазваме в масив от символи сървър, към който се свързваме, а в низове – стойностите на данните, измерени от сензора, които се изпращат към сървър чрез POST заявки.

```
void setup() {
  Serial.begin(115200);
  delay(100);

  pinMode(DHTPin, INPUT);
  dht.begin();

  while(status != WL_CONNECTED) {
    Serial.println("Connecting to ");
    Serial.println(ssid);
    status = WiFi.begin(ssid, password);
    delay(1000);
  }

  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());
  IPAddress ip = WiFi.localIP();
  IPAddress gateway = WiFi.gatewayIP();
  Serial.print("Got IP: ");
  Serial.print(ip);
}
```

Проверяваме статуса на мрежата и ако модулът все още не е свързан, while цикълът се повтаря до неговото свързване. При успешна връзка името и IP-то се изписват на серийния монитор.



```

void loop() {
  dhtObject.read(DHTPin);
  delay(100);
  float temperature = dhtObject.temperature;
  dhtObject.read(DHTPin);
  float humidity = dhtObject.humidity;

  postData1 = postVariable1 + temperature;
  postData2 = postVariable2 + humidity;

  if(client.connect(server, 80)){
    client.println("POST /apiary/api.php HTTP/1.1");
    client.println("Host: www.petar-petrov.com");
    client.println("Content-Type: application/x-www-form-urlencoded");
    client.println("Content-Length");
    client.println(postData1.length());
    client.println(postData2.length());
    client.println();
    client.print(postData1);
    client.print(postData2);
  }

  if(client.connected){
    client.stop();
  }
  Serial.println(postData1);
  Serial.println(postData2);

  delay(3000);
}

```

Стойности за температура и влажност се прочитат от сензора и се запазват в променливи.

При успешна връзка със сървъра данните се изпращат чрез POST заявка на следното API - <https://petar-petrov.com/apiary/api.php>, извеждайки се в JSON формат.

След това в мобилното приложение се създава инстанция на класа WebClient, чрез който се теглят данните от API-то. Чрез класовете WeatherStationTemperature.cs и WeatherStationHumidity.cs се

десериализират данните. Създава се променлива за стартиране на обратен цикъл с цел да се открие последния запис, направен от метеорологичната станция, чието apiary\_id да съвпада с id-то на избрания пчелин. При първото съвпадение цикълът приключва и се извежда информацията под формата на заглавие в страницата. Ако не е открито съвпадение се извежда съобщението – „Няма поставена метеорологична станция.“



## Заклучение

Проектът “Метеорологична станция за пчелини с Arduino UNO” е стъпка към навлизането на технологиите в по-широк мащаб в селското стопанство - конкретно в пчеларството. Той може да се надгради с добавянето на сензори за измерване нивото на звука и теглото на всеки един кошер. По този начин ще се предотвратяват роенето на пчелите, пчеларят ще знае кои пчелни семейства се нуждаят от поставянето на магазини. Това ще облекчи работата на пчеларя, намалявайки броя на прегледите.

