

Учебно-методического пособие
«Разработка программного обеспечения для платформы
ARDUINO»

по 2 лабораторной работе для образовательных программ по
направлениям

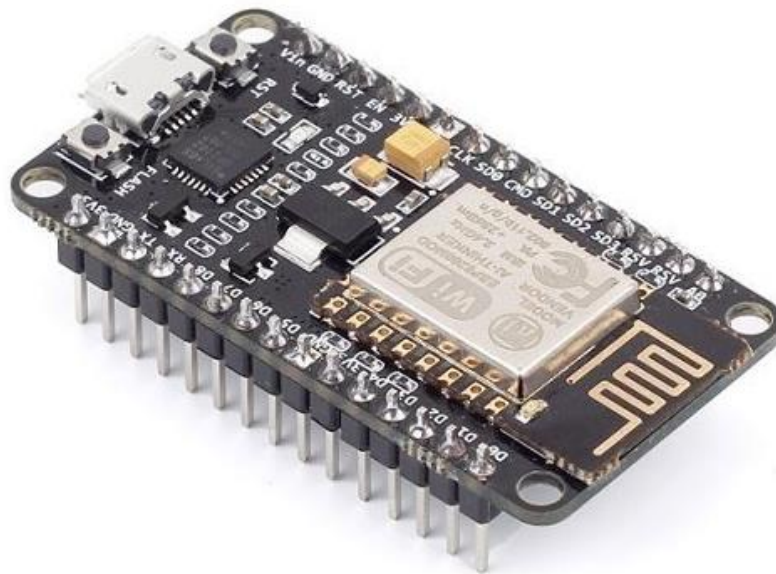
бакалавриата 02.03.03, 09.03.04 (дисциплина "Инженерия
программного обеспечения встраиваемых систем, систем ЦОС и
систем реального времени")

магистратуры 09.04.04 (дисциплина "Системы кодирования
информации и цифровой обработки сигналов")

Оглавление

NodeMCU (ESP8266)	3
Характеристики NodeMCU	4
Подключение NodeMCU	5
Подключение датчика температуры и влажности DHT22	10
Требуемые инструменты.....	10
Схема подключения.....	10
Установка библиотек	10
Программа взаимодействия	12
Подключение датчика DHT22 и передача данных на web-сервер	14
Требуемые инструменты.....	14
Схема подключения.....	14
Установка библиотек	15
Программа взаимодействия	16
Результат.....	19
Создание web-страницы	20
Задания к лабораторной работе	23
Вводная информация (ко всех темам)	23
Тема 1. Охрана дома	23
Тема 2. Защита дома от пожара	24
Тема 3. Комфортный дом.....	24
Тема 4. Управление предметами по будильнику.....	25
Тема 5. Защита от затопления дома	25
Тема 6. Комфортный дом для родных	26

NodeMCU (ESP8266)



NodeMCU — это платформа на основе модуля ESP8266. Плата предназначена для удобного управления различными схемами на расстоянии посредством передачи сигнала в локальную сеть или интернет через Wi-Fi. На базе данной платы можно создать «Умный дом», при этом будет настроено управление датчиками и параметрами через мобильное устройство или веб-приложение.

Характеристики NodeMCU

Размер платы NodeMCU — 6 * 3 см. Плата довольно небольшого размера, что позволяет использовать её во многих проектах. «Ножки» NodeMCU расположены так, что не возникает трудностей в её установке на макетную плату.

На лицевой части платы разъем Micro USB, с помощью которого в контроллер заливают скетчи или подают питание от powerbank-а или компьютера.

Рядом с разъемом располагаются две кнопки: «Flash» и «Reset». Кнопка «Flash» используется для отладки, а кнопка «Reset» для перезагрузки платы.

Больше всего места на плате занимает чип ESP8266, на котором установлен микропроцессор с тактовой частотой 80 МГц (можно разогнать до 160 МГц). Плата имеет 4 мегабайта Flash-памяти.

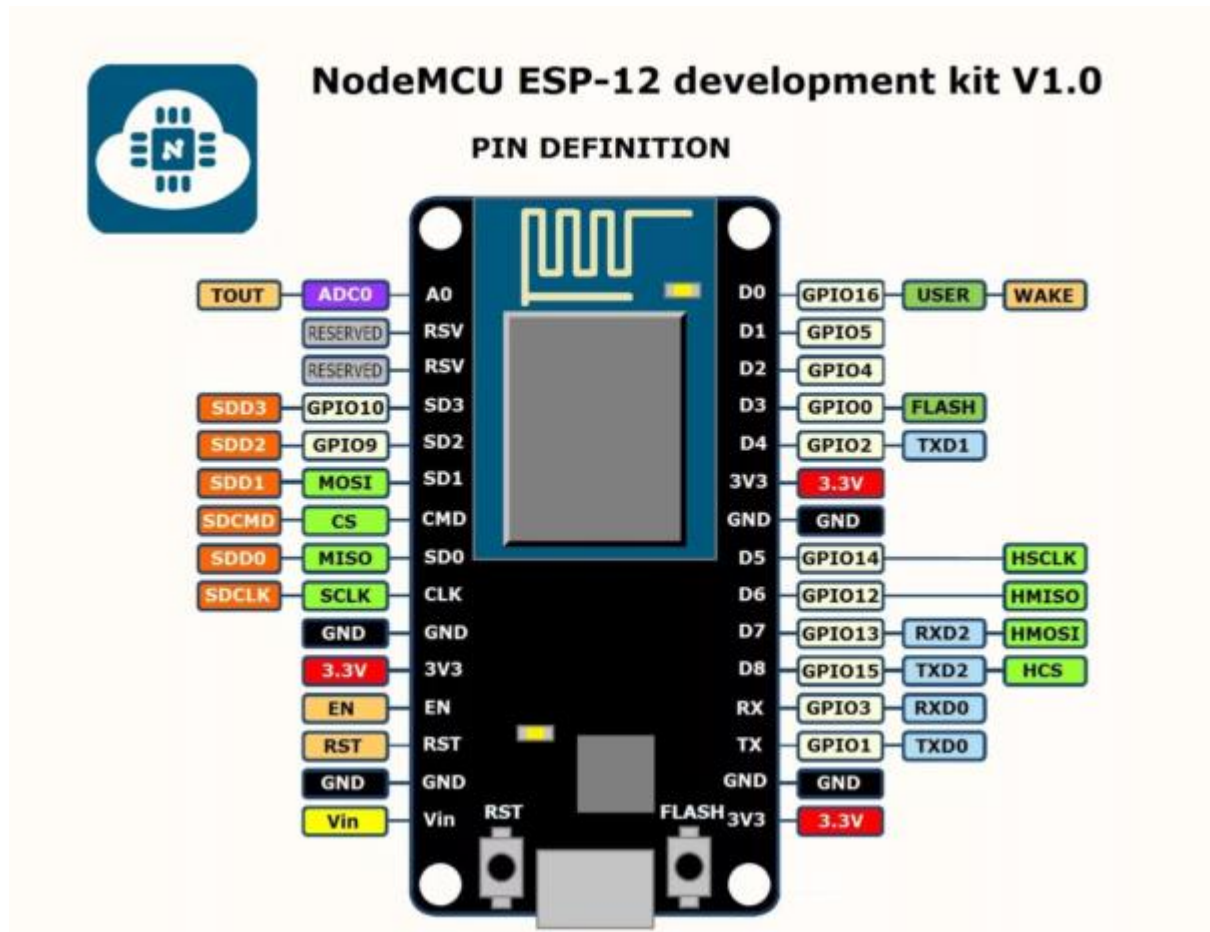
Для питания на плату можно подавать напряжение от 5 до 12 В, но рекомендуется от 10 В. Можно питать как от Micro USB, так и от контакта Vin (от 5В.). Также существуют дополнительные платы расширения для удобного питания модулей.

Плата потребляет небольшое количество энергии. Это позволяет использовать ее с автономным питанием.

NodeMCU имеет 11 портов ввода-вывода общего назначения.

Некоторые из портов имеют дополнительные функции:

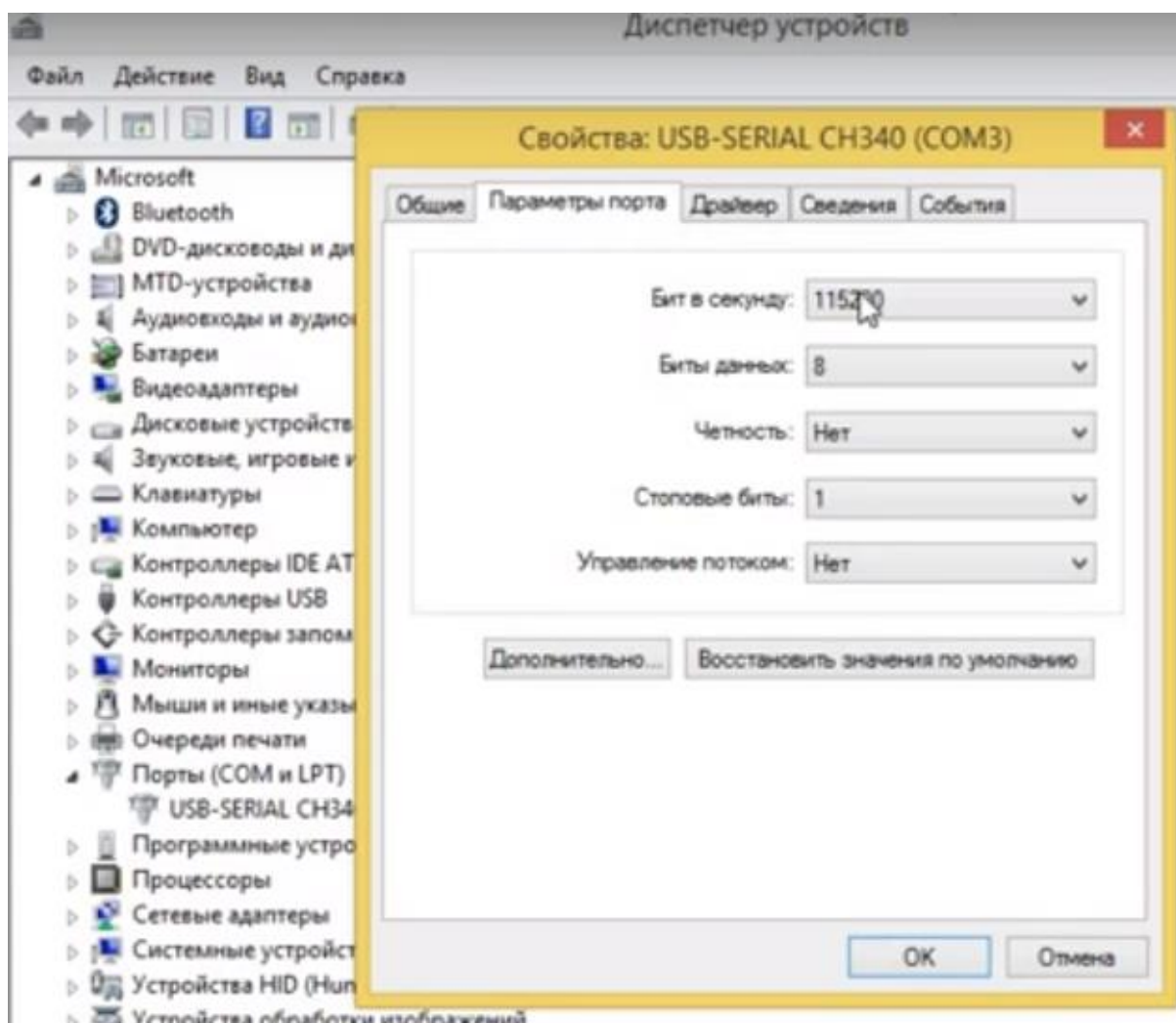
- D9, D10 — UART
- D1, D2 — I²C/TWI
- D5–D8 — SPI
- D1–D10 — выходы с ШИМ (PWM)
- A0 — аналоговый вход с АЦП.



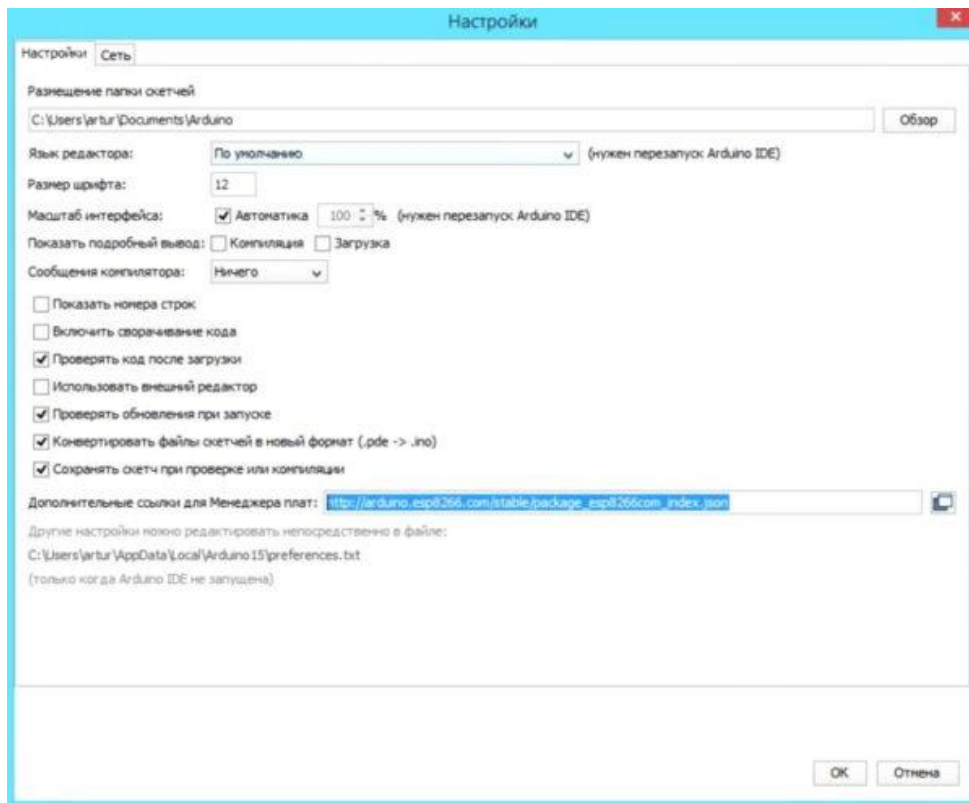
Подключение NodeMCU

Подключаем плату NodeMCU к компьютеру с помощью USB кабеля:

- сначала необходимо установить драйвер CH340G для операционной системы. Скачать драйвер можно на этой странице: <http://www.ek21.ru/esp8266/> (конец страницы). Для того чтобы убедиться, что драйвер установлен корректно. Нужно проверить COM порт. Плата должна быть в этот момент подключена к компьютеру. Заходим в раздел «Диспетчер устройств» -> «Порты (COM и LPT)» и проверяем, что там появился порт и его номер. Далее заходим «Свойства» данного порта -> «Параметры порта» и меняем значение «Бит в секунду» на 115200. И переходим к настройке среды разработки.



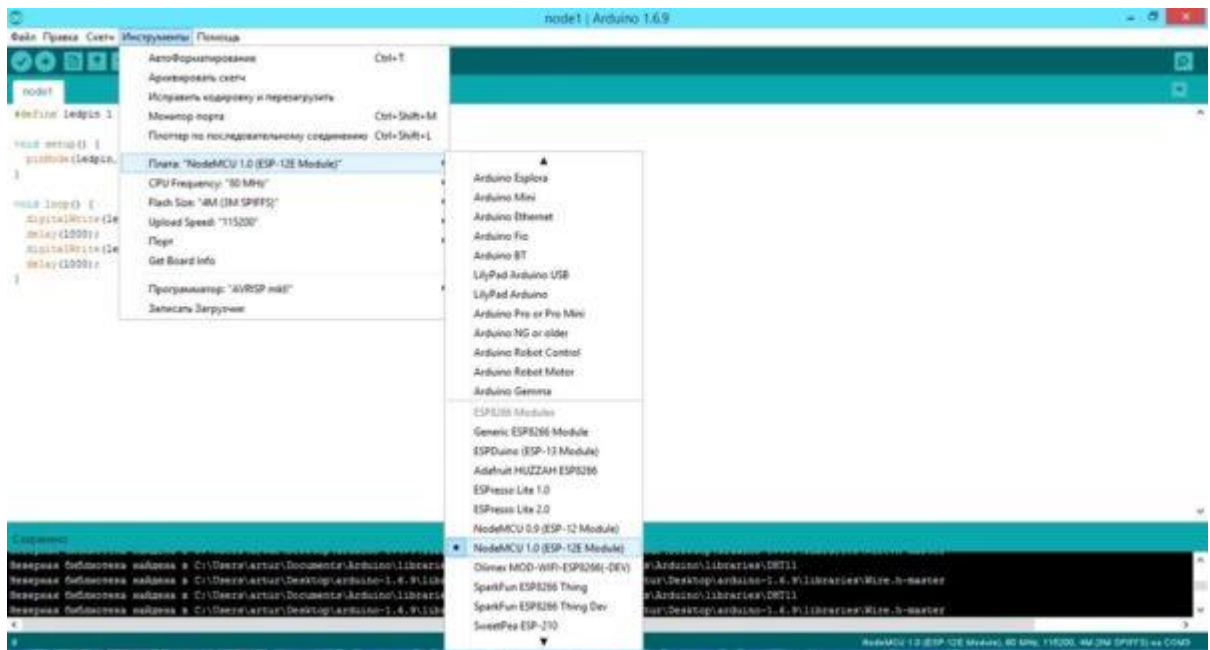
Открываем Arduino IDE. Заходим в раздел «Файл» -> «Настройки» и находим строчку «Дополнительные ссылки для менеджера плат» и вставляем туда следующую ссылку http://arduino.esp8266.com/stable/package_esp8266com_index.json или http://download.igrr.me/stable/package_esp8266com_index.json,



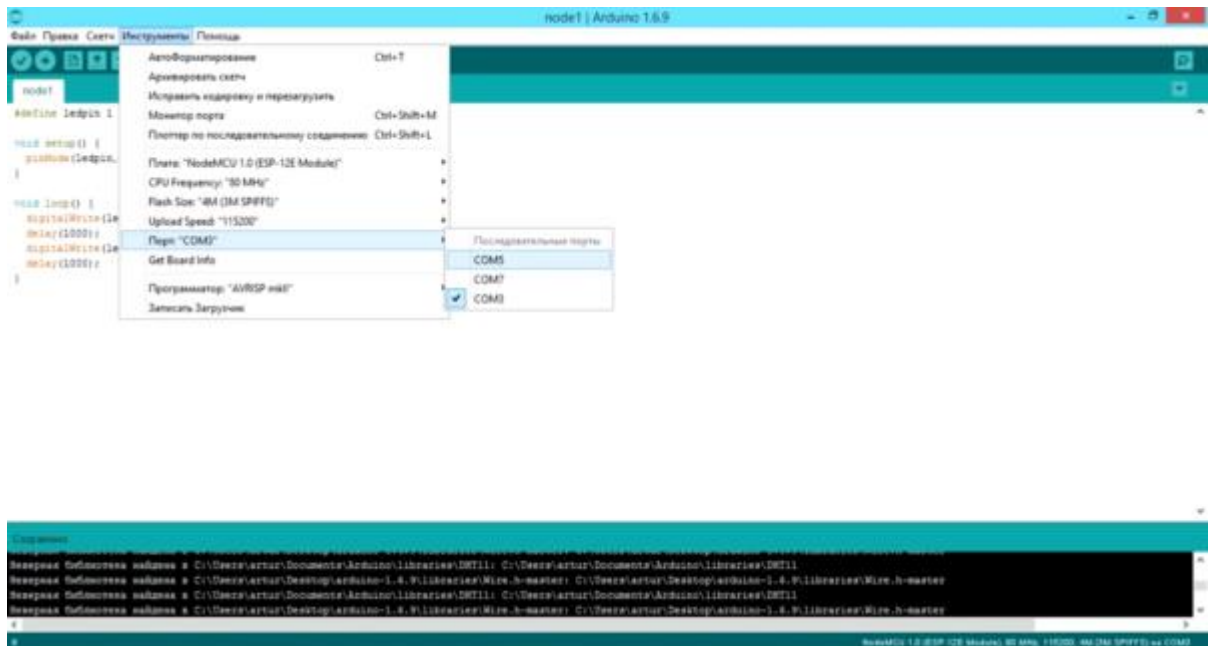
- Далее переходим в раздел «Инструменты» -> «Плата» -> «Менеджер плат», где выбираем «esp8266» и скачиваем последнюю версию



- После установки переходим в следующий раздел «Инструменты» -> «Плата» и находим в списке Node MCU и выбираем с последней версией



• Так же в том же разделе необходимо зайти в подраздел «Порт» и выбрать тот, в который подключена плата. Это должен быть тот же порт, который был указан в «Диспетчер устройств»



• И после настройки среды для работы с платой можно проверить её работоспособность. Для это можно запустить шаблон скетча. Заходим в раздел «Файл» -> «Примеры» -> «ESP8266WiFi» -> «WiFiScan». После открывшийся скетч заливаем в плату:

```
#include "ESP8266WiFi.h"
void setup() {
  Serial.begin(115200);
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(100);
}
```



```

Serial.println("Setup done");
}

void loop() {
  Serial.println("scan start");

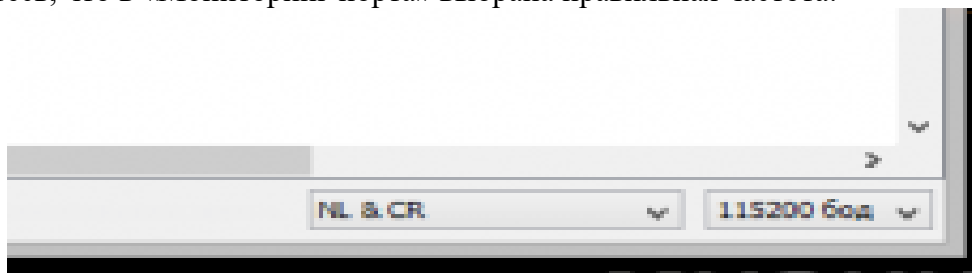
  // WiFi.scanNetworks will return the number of networks found
  int n = WiFi.scanNetworks();
  Serial.println("scan done");
  if (n == 0) {
    Serial.println("no networks found");
  } else {
    Serial.print(n);
    Serial.println(" networks found");
    for (int i = 0; i < n; ++i) {
      // Print SSID and RSSI for each network found
      Serial.print(i + 1);
      Serial.print(": ");
      Serial.print(WiFi.SSID(i));
      Serial.print(" (");
      Serial.print(WiFi.RSSI(i));
      Serial.print(")");
      Serial.println((WiFi.encryptionType(i) == ENC_TYPE_NONE) ? " " : "*");
      delay(10);
    }
  }
  Serial.println("");

  // Wait a bit before scanning again
  delay(5000);
}


```

После удачной загрузки скетча открываем «Инструменты» -> «Мониторинг порта» и наблюдаем список доступных Wifi – сетей.

Убедитесь, что в «Мониторинг порта» выбрана правильная частота:



Результат выполнения программы:

 COM4

```
4: MTSRouter_2.4GHz_003593 (-89) *
5: DIR-815 (-61) *
6: TP-LINK_AE3A (-75) *
7: Keenetic-7854 (-70) *
8: ваанда (-79) *
9: HTC Portable Hotspot DDF5 (-76) *
10: ROSTELECOM (-89) *

scan start
scan done
11 networks found
1: TP-LINK_5F8EEA (-87) *
2: RT-WiFi-D8EE (-90) *
3: VDV (-86) *
4: DIR-815 (-57) *
5: TP-LINK_AE3A (-79) *
6: ваанда (-71) *
7: Keenetic-7854 (-67) *
8: MTSRouter-D4F25E (-93) *
9: HTC Portable Hotspot DDF5 (-72) *
10: T@LISM@N (-92) *
11: ROSTELECOM (-90) *
```

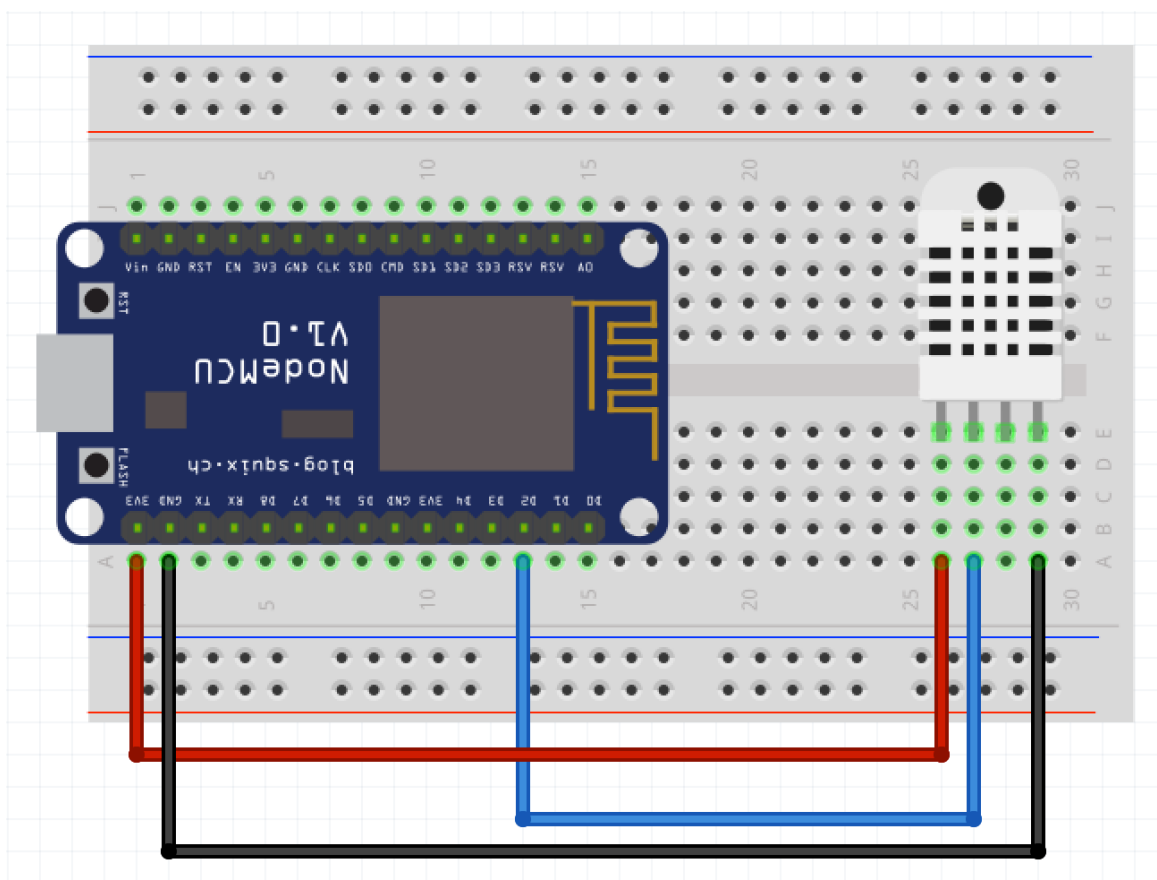
☒ Автопрокрутка ☐ Показывать отметки времени

Подключение датчика температуры и влажности DHT22

Требуемые инструменты

- провода папа-папа
- макетная плата
- датчик DHT22
- провод Micro-USB
- модуль NodeMCU

Схема подключения



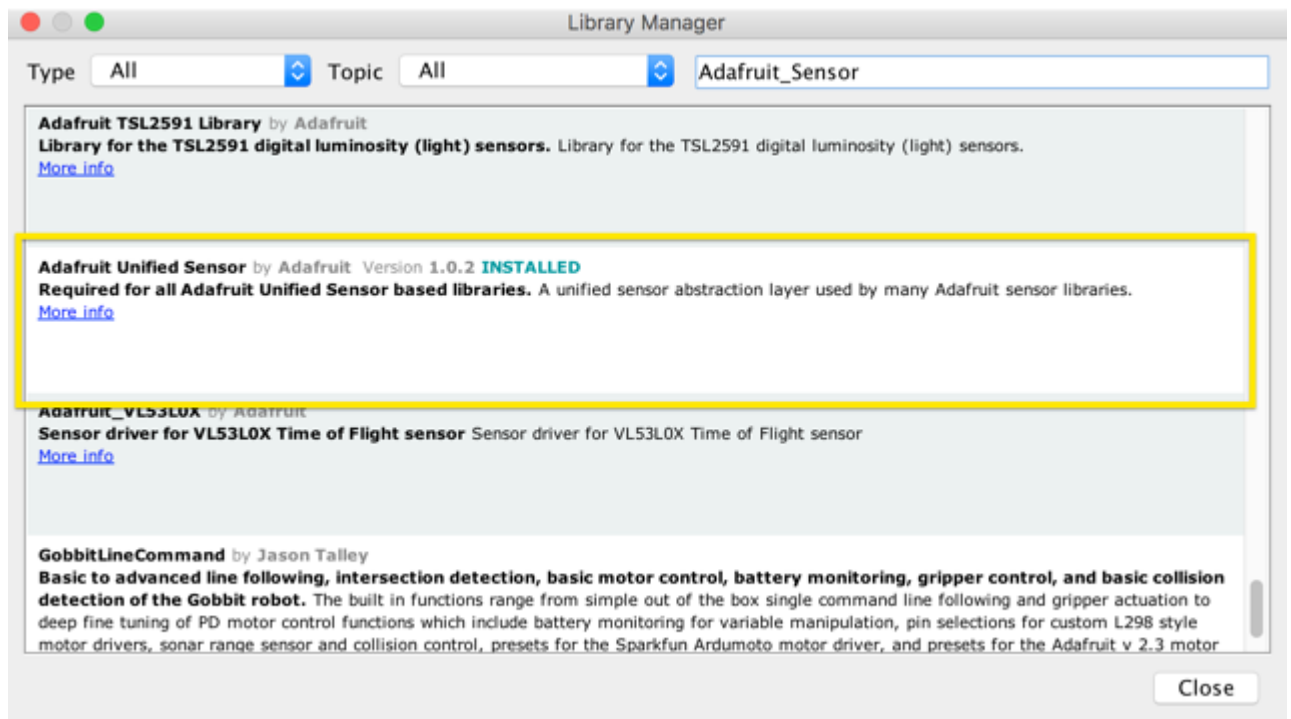
Перед работой расположите модуль так, чтобы передняя часть (с отверстиями) DHT22 была обращена к вам.

- Самый левый штифт DHT22 (+) - это положительный штифт. Вы должны подключить его к 3v3 или Vin на MCU.
- Второй вывод DHT22 (out)- это вывод данных. Вы должны подключить его к D2 на MCU.
- Третий вывод DHT22 (слева) ничего не делает. Если он имеется на датчике.
- Последний штифт DHT22 (-) - это штифт заземления. Он должен быть подключен GND.

Установка библиотек

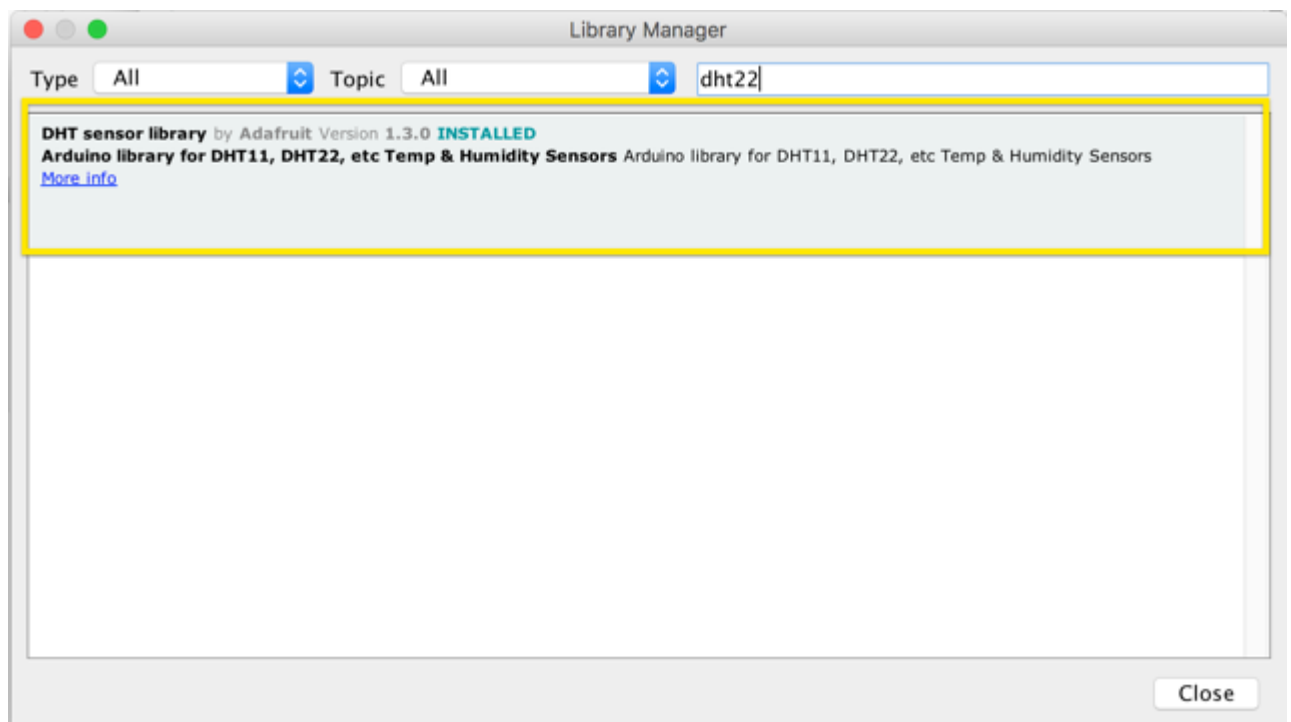
Для работы с датчиком нужно установить две библиотеки. Чтобы прочитать датчик, нужно использовать библиотеку Adafruit DHT22. Его можно установить с помощью менеджера библиотек Arduino. Он состоит из двух компонентов.

Во- первых, нужно загрузить библиотеку унифицированных датчиков Adafruit:

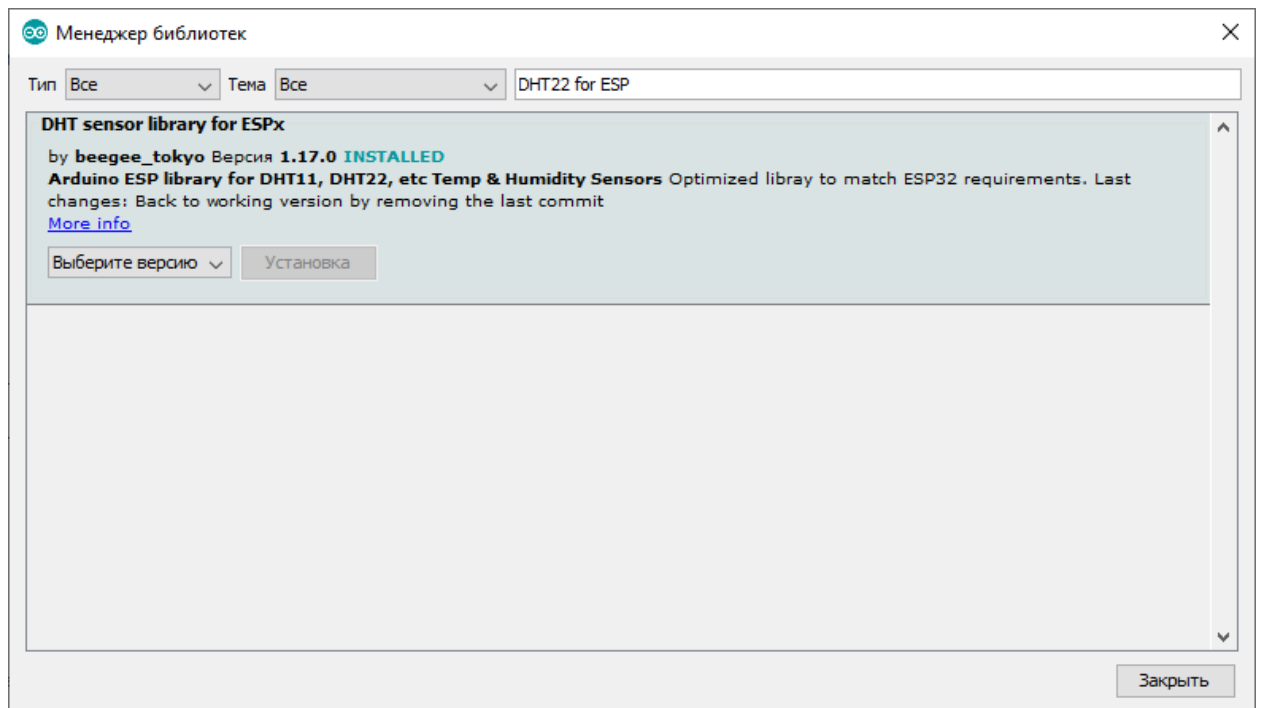


Во-вторых, установить две библиотеки для самого датчика DHT:

1) Общая библиотека датчика DHT22:



2) Библиотека датчика DHT22 для модуля ESPxx:



Программа взаимодействия

```
#include "DHT.h"
#define DHTPIN 2    // what digital pin the DHT22 is conected to
#define DHTTYPE DHT22 // there are multiple kinds of DHT sensors

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  Serial.setTimeout(2000);
  while(!Serial) { }
  Serial.println("Device Started");
  Serial.println("-----");
  Serial.println("Running DHT!");
  Serial.println("-----");
}

void loop() {
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  float f = dht.readTemperature(true);
  float hif = dht.computeHeatIndex(f, h);
  float hic = dht.computeHeatIndex(t, h, false);
  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(t);
  Serial.print(" *C ");
```

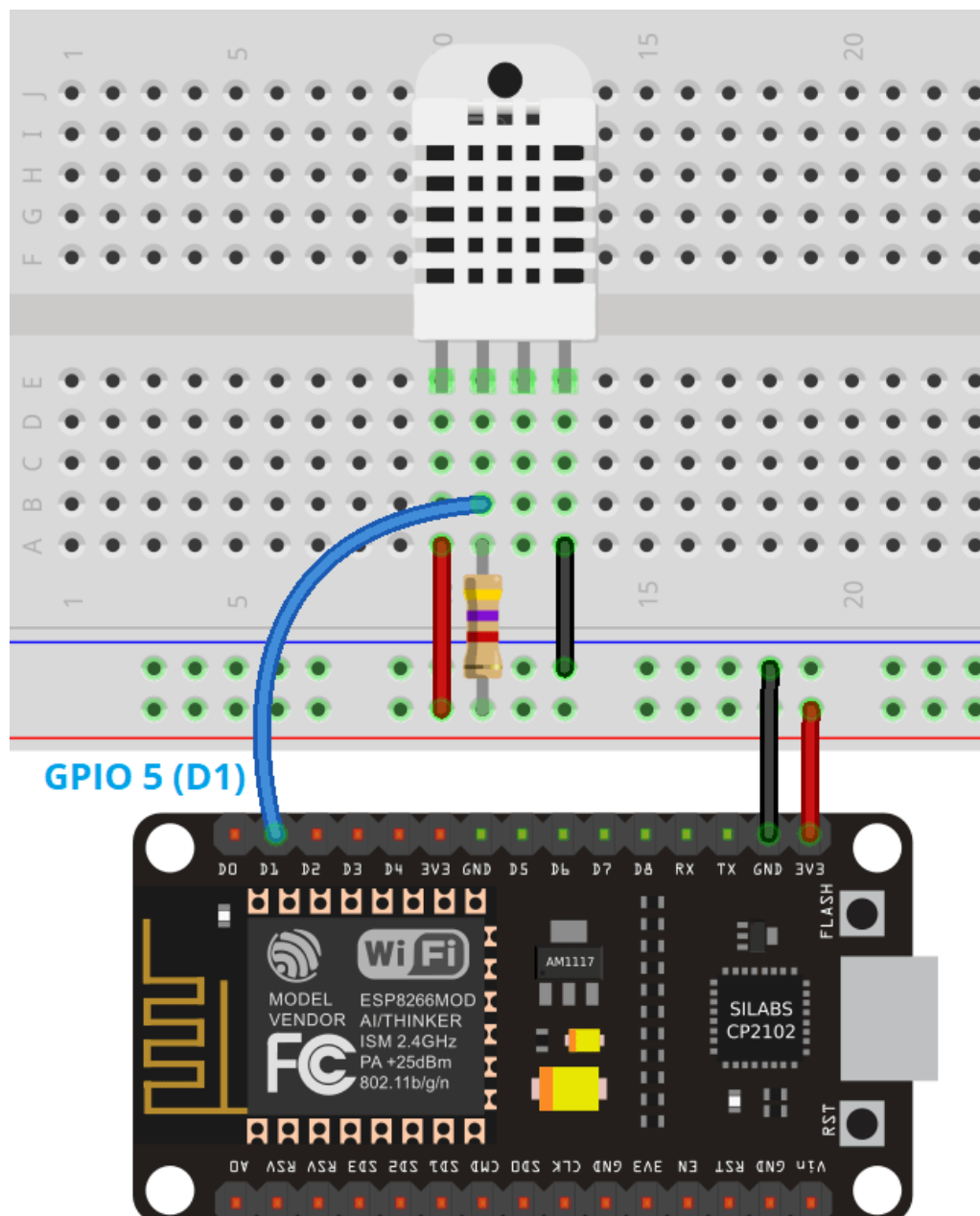
```
Serial.print(f);  
Serial.print(" *F\t");  
Serial.print("Heat index: ");  
Serial.print(hic);  
Serial.print(" *C ");  
Serial.print(hif);  
Serial.println(" *F");  
}
```

Подключение датчика DHT22 и передача данных на web-сервер

Требуемые инструменты

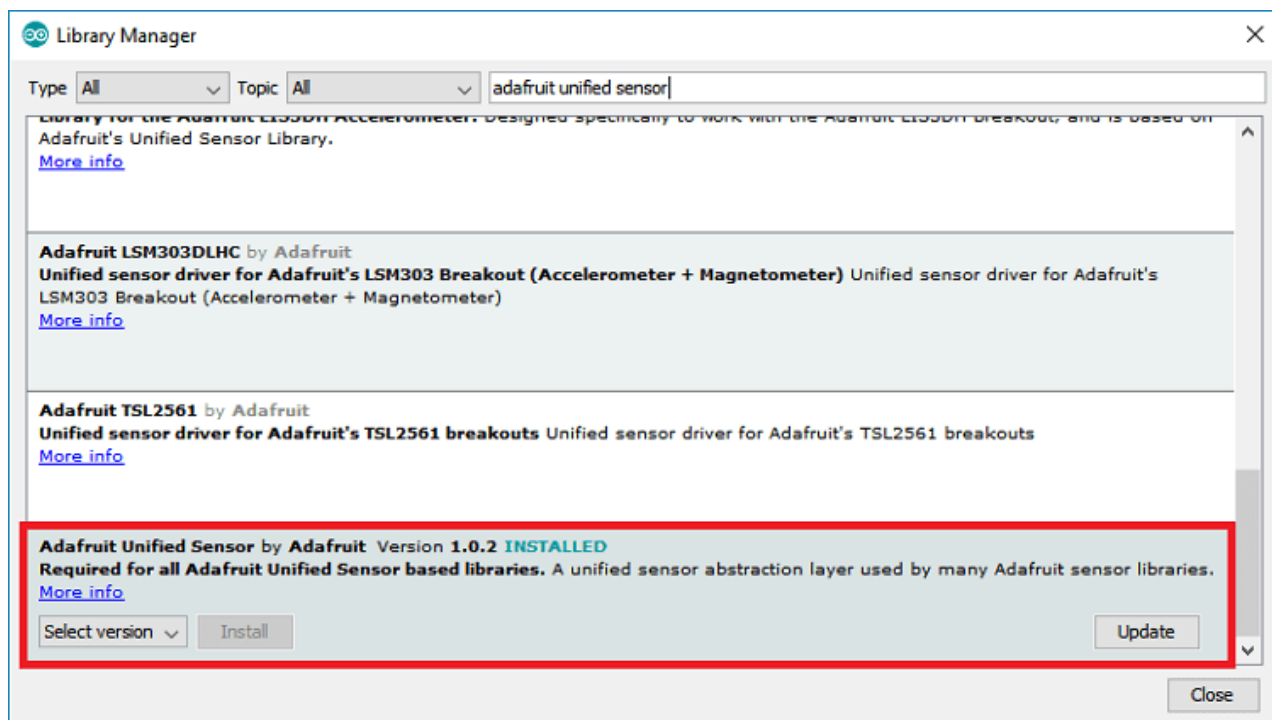
- провода папа-папа
- макетная плата
- датчик DHT22
- провод Micro-USB
- модуль NodeMCU
- резистор 4.7 кОм

Схема подключения



Установка библиотек

1) После установки библиотек DHT нужно установить библиотеки Adafruit введите «Adafruit Unified Sensor» в поле поиска. Прокрутите путь вниз, чтобы найти библиотеку и установить ее.



2) Для создания веб-сервера мы будем использовать библиотеку ESPAsyncWebServer, которая обеспечивает простой способ создания асинхронного веб-сервера.

Библиотека ESPAsyncWebServer недоступна для установки в Arduino IDE Library Manager. Вам нужно установить его вручную.

Выполните следующие шаги для установки библиотеки ESPAsyncWebServer:

1. Нажмите здесь, чтобы загрузить библиотеку ESPAsyncWebServer. У вас должна быть папка .zip в папке «Загрузки»
2. Распакуйте папку .zip и вы должны получить *ESPAsyncWebServer-мастер* папку
3. Переименуйте вашу папку в *ESPAsyncWebServer*
4. Переместите папку *ESPAsyncWebServer* в папку библиотек установки Arduino IDE.

3) Для библиотеки ESPAsyncWebServer требуется библиотека ESPAsyncTCP. Выполните следующие шаги, чтобы установить эту библиотеку:

1. Нажмите здесь, чтобы загрузить библиотеку ESPAsyncTCP. У вас должна быть папка .zip в папке «Загрузки»
2. Распакуйте папку .zip и вы должны получить *ESPAsyncTCP - мастер* папку
3. Переименуйте вашу папку в *ESPAsyncTCP*

4. Переместите папку *ESPAsyncTCP* в папку установочных библиотек Arduino IDE.
5. Откройте Arduino IDE

Программа взаимодействия

```
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <Hash.h>
#include <ESPAsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
// Replace with your network credentials
const char* ssid = "REPLACE_WITH_YOUR_SSID";
const char* password = "REPLACE_WITH_YOUR_PASSWORD";

#define DHTPIN 5
#define DHTTYPE DHT22

DHT dht (DHTPIN, DHTTYPE);

float t = 0.0;
float h = 0.0;

// Create AsyncWebServer object on port 80

AsyncWebServer server(80);
unsigned long previousMillis = 0;

// Updates DHT readings every 10 seconds
const long interval = 10000;

const char index_html[] PROGMEM = R"rawliteral(
<!DOCTYPE HTML><html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.2/css/all.css"
integrity="sha384-
fnmOCqbTlWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr"
crossorigin="anonymous">
  <style>
    html {
      font-family: Arial;
      display: inline-block;
      margin: 0px auto;
      text-align: center;
    }
    h2 { font-size: 3.0rem; }
    p { font-size: 3.0rem; }
    .units { font-size: 1.2rem; }
```

```

.dht-labels{
    font-size: 1.5rem;
    vertical-align:middle;
    padding-bottom: 15px;
}
</style>
</head>
<body>
<h2>ESP8266 DHT Server</h2>
<p>
    <i class="fas fa-thermometer-half" style="color:#059e8a;"></i>
    <span class="dht-labels">Temperature</span>
    <span id="temperature">%TEMPERATURE%</span>
    <sup class="units">&deg;C</sup>
</p>
<p>
    <i class="fas fa-tint" style="color:#00add6;"></i>
    <span class="dht-labels">Humidity</span>
    <span id="humidity">%HUMIDITY%</span>
    <sup class="units">%</sup>
</p>
</body>
<script>
setInterval(function ( ) {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("temperature").innerHTML = this.responseText;
        }
    };
    xhttp.open("GET", "/temperature", true);
    xhttp.send();
}, 10000 ) ;

setInterval(function ( ) {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("humidity").innerHTML = this.responseText;
        }
    };
    xhttp.open("GET", "/humidity", true);
    xhttp.send();
}, 10000 ) ;
</script>
</html>>rawliteral";
// Replaces placeholder with DHT values
String processor(const String& var) {
    //Serial.println(var);
    if(var == "TEMPERATURE"){
        return String(t);
    }
}

```

```

else if(var == "HUMIDITY"){
    return String(h);
}
return String();
}

void setup(){
    Serial.begin(115200);
    dht.begin();

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

    // Print ESP8266 Local IP Address
    Serial.println(WiFi.localIP());

    server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
        request->send_P(200, "text/html", index_html, processor);
    });
    server.on("/temperature", HTTP_GET, [](AsyncWebServerRequest *request){
        request->send_P(200, "text/plain", String(t).c_str());
    });
    server.on("/humidity", HTTP_GET, [](AsyncWebServerRequest *request){
        request->send_P(200, "text/plain", String(h).c_str());
    });

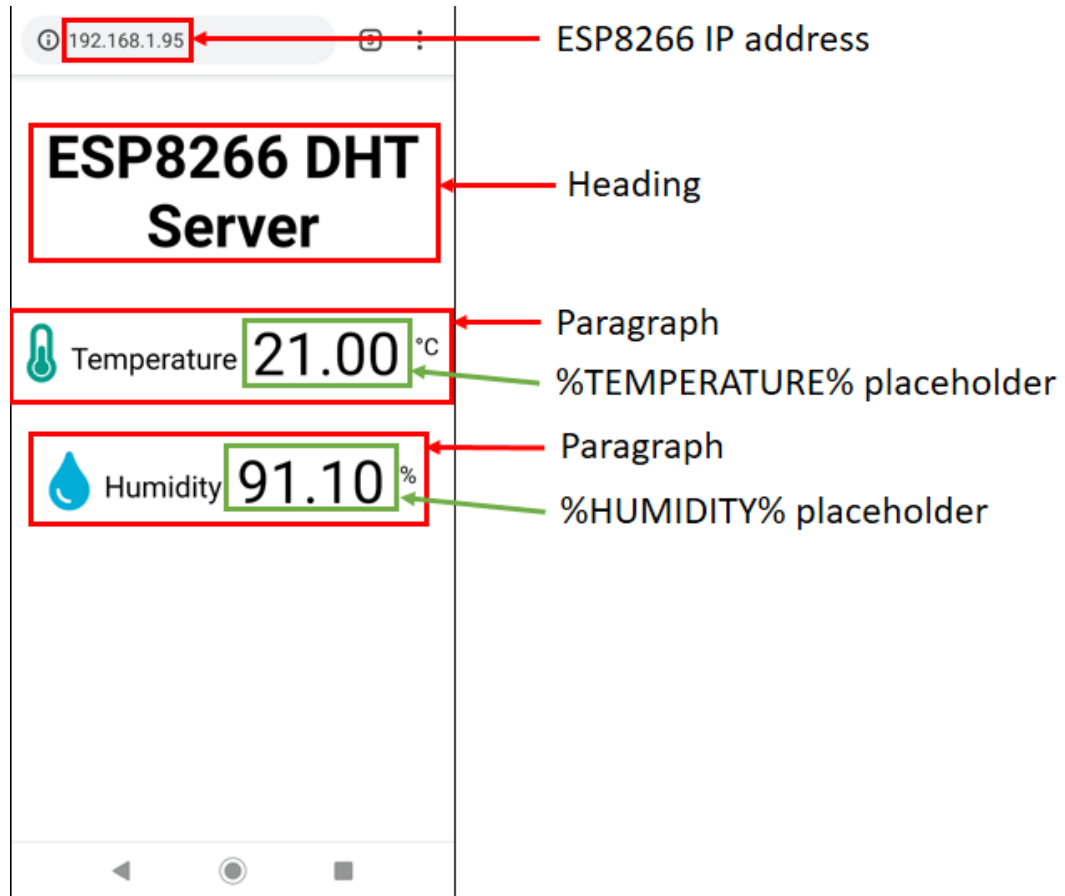
    server.begin();
}

void loop() {
    unsigned long currentMillis = millis();
    if (currentMillis - previousMillis >= interval) {
        previousMillis = currentMillis;
        float newT = dht.readTemperature();
        if (isnan(newT)) {
            Serial.println("Failed to read from DHT sensor!");
        }
        else {
            t = newT;
            Serial.println(t);
        }
        float newH = dht.readHumidity();
        if (isnan(newH)) {
            Serial.println("Failed to read from DHT sensor!");
        }
        else {
            h = newH;

```

```
Serial.println(h);  
}  
}  
}
```

Результат



Веб-страница показывает один заголовок и два абзаца. Есть параграф для отображения температуры, а другой - для отображения влажности. Есть также две иконки для стилизации страницы.

Создание web-страницы

Весь текст HTML со включенными стилями хранится в переменной `index_html`. Теперь мы пройдемся по тексту HTML и посмотрим, что делает каждая часть.

Тег `<meta>` делает вашу веб-страницу отзывчивой в любом браузере.

Тег `<link />` нужен для загрузки иконки с сайта [fontawesome](https://fontawesome.com/).

Между тегами `<style>` `</ style>` мы добавляем CSS для стилизации веб-страницы.

Внутри тегов `<body>` `</ body>` мы добавляем контент веб-страницы.

`<h2>` `</ h2>` теги добавить заголовок веб - страницы. В этом случае текст «ESP8266 DHT сервер», но вы можете добавить любой другой текст.

Есть два абзаца. Один для отображения температуры, а другой для отображения влажности. Абзацы разделены тегами `<p>` и `</ p>`.

Абзац для температуры следующий:

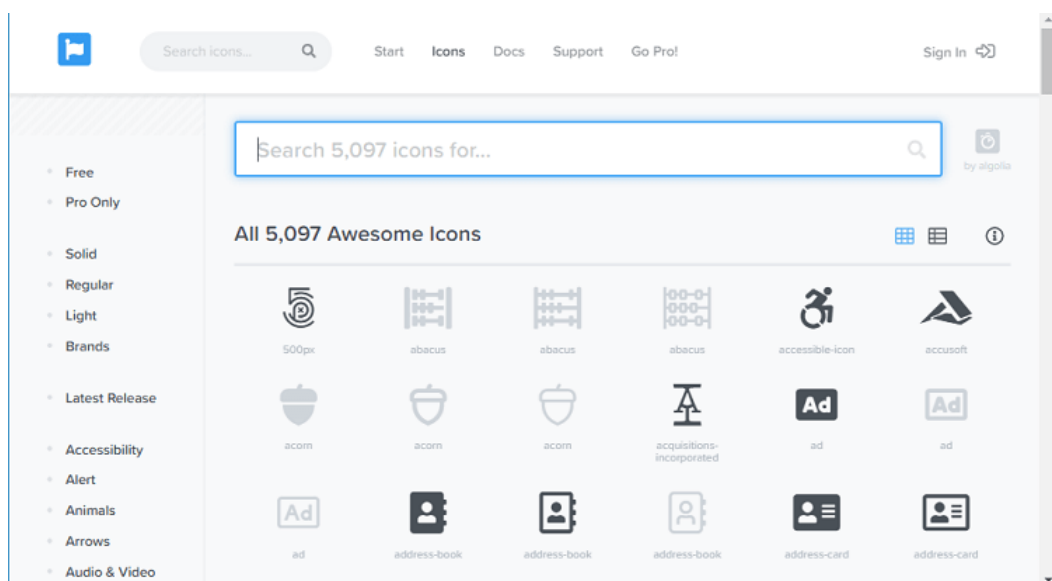
```
<p>
  <i class="fas fa-thermometer-half" style="color:#059e8a;"></i>
  <span class="dht-labels">Temperature</span>
  <span id="temperature">%TEMPERATURE%</span>
  <sup class="units">°C</sup>
</p>
```

И абзац для влажности находится на следующем фрагменте:

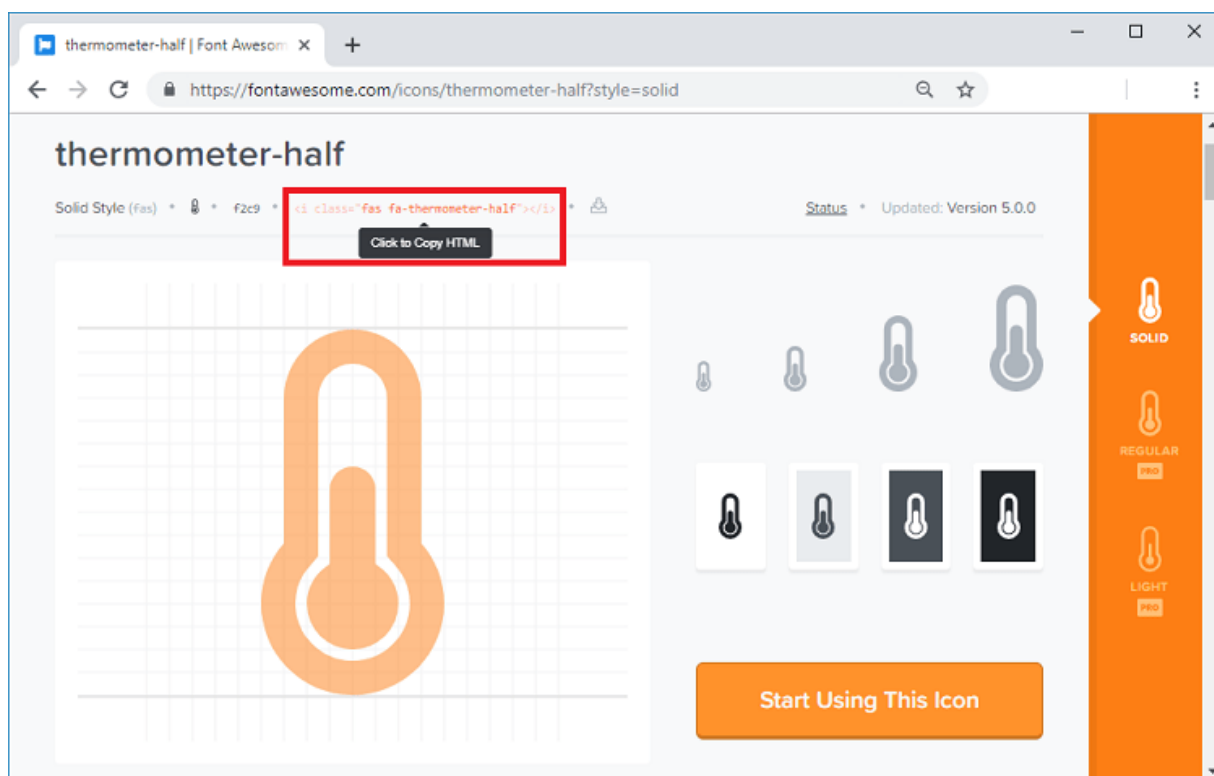
```
<p>
  <i class="fas fa-tint" style="color:#00add6;"></i>
  <span class="dht-labels">Humidity</span>
  <span id="humidity">%HUMIDITY%</span>
  <sup class="units">%</sup>
</p>
```

В `<i>` теги отображаются значки [fontawesome](https://fontawesome.com/).

Чтобы выбрать значки, перейдите на [веб-сайт Font Awesome Icons](https://fontawesome.com/). Найдите значок, который вы ищете.



Нажмите на нужный значок. Затем вам просто нужно скопировать предоставленный текст HTML.



Чтобы выбрать цвет, вам просто нужно передать параметр стиля с шестнадцатеричным цветом следующим образом:

```
<i class="fas fa-tint" style="color:#00add6;"></i>
```

Текст `TEMPERATURE` между знаками `%` является заполнителем для значения температуры. Это означает, что этот текст `%TEMPERATURE%` подобен переменной, которая будет заменена фактическим значением температуры от датчика DHT. Заполнители в тексте HTML должны находиться между знаками `%`.

На нашей веб-странице есть код JavaScript, который автоматически обновляет температуру и влажность каждые 10 секунд.

Сценарии в тексте HTML должны находиться между тегами `<script>< script>`.

Чтобы обновить температуру на фоне, у нас есть функция `setInterval()`, которая запускается каждые 10 секунд.

По сути, он делает запрос в URL `/temperature`, чтобы получить последние показания температуры.

Получив это значение, он обновляет HTML-элемент, идентификатор которого — `temperature`.

```
if (this.readyState == 4 && this.status == 200) {  
    document.getElementById("temperature").innerHTML = this.responseText;  
}
```

Таким образом, этот предыдущий раздел отвечает за асинхронное обновление температуры. Тот же процесс повторяется для показаний влажности.

Функция **processor()**, заменит заполнители в нашем HTML-тексте фактическими значениями температуры и влажности.

```
String processor(const String& var){  
  if(var == "TEMPERATURE"){  
    return String(t);  
  }  
  else if(var == "HUMIDITY"){  
    return String(h);  
  }  
  return String();  
}
```

Когда веб-страница запрашивается, мы проверяем, есть ли в HTML-заполнителях. Если он находит заполнитель **%TEMPERATURE%**, мы возвращаем температуру, которая хранится в переменной **t**.

Задания к лабораторной работе

Вводная информация (ко всех темам)

1) Во всех заданиях необходимо разработать программно-аппаратную систему на базе NodeMCU ESP8266 и дополнительной периферии, которая обеспечивает получение информации от определенных датчиков, ее обработку и формирование управляющих воздействий на внешние устройства управления или отображения.

2) Алгоритм функционирования системы, "связывающий" данные, получаемые от датчиков, и управляющие воздействия, которые выдаются на устройства управления/отображения, необходимо разработать и описать самостоятельно.

3) Для взаимодействия платформы NodeMCU с большинством периферийных устройств требуется использование драйверов устройств, которые включены в дополнительные библиотеки (см. сайт дисциплины).

4) Для получения работоспособной аппаратной платформы, правильного подключения некоторых периферийных устройств может потребоваться использование дополнительных радиодеталей, не указанных в задании, но доступных в наборе комплектующих.

5) Взаимодействие со всеми датчика и вывод полной информации проходит через веб-приложение или мобильное приложение на Ваше усмотрение.

Тема 1. Охрана дома

Нужно защитить дом от несанкционированного доступа. Для этого использовать датчик движения. Но так как датчик движения будет реагировать не только на людей, а еще и на пробегающих мимо животных, то нужно сделать дополнительные проверки. Датчик касания, который включит Вашу охранную систему. И дополнительная проверка для доступа в дом по магнитному ключу или магнитной карточке. Световыми и звуковыми эффектами оповестите хозяев либо о успешном доступе в дом ими самими, либо о незаконном доступе.

Данные с входных датчиков отображаются в приложении и подается сигнал на выходные устройства. Пользователю в приложении отображаются сообщения о датчиках:

- о положении поставленной/снятой сигнализации;
- сообщение о том, кто вошел по карточке/ключу;
- произошло ли движение.

Датчики (сенсоры) для съема данных	Устройства управления / отображения
1) Датчик касания 2) HC-SR501 (PIR датчик) 3) RFID МОДУЛЬ RC522 13.56MHZ + КАРТА + БРЕЛОК	1) Светодиоды 2) Пьезоизлучатель

Тема 2. Защита дома от пожара

Нужно защитить дом от несчастных случаев в доме. В доме со время отсутствия хозяев может возникнуть пожар. Поэтому нужно предупредить, что такая ситуация появилась. Но не кому не хочется узнать, что уже его дом горит. Поэтому для начала можно просто сигнализировать о том, что в доме появились газы разных разновидностей. Для этого используйте датчик. А для того чтобы пользователю было известно, какой газ и где появился, используйте монитор для вывода значений и звуковую сигнализацию о уже начинающемся пожаре.

Данные с входных датчиков отображаются в приложении и подается сигнал на выходные устройства. Пользователю в приложении отображаются сообщения о датчиках:

- о появлении огня;
- о появлении дыма/газа и тд. И какой уровень этого показателя;
- о состоянии температуры и влажности в помещении и о необходимости предпринять различные меры.

Датчики (сенсоры) для съема данных	Устройства управления / отображения
1) Датчик пламени (огня) 2) Датчик широкого спектра газов MQ-2 3) DHT22 Temperature-Humidity Sensor (HDC1080)	1) Светодиоды 2) Пьезоизлучатель

Тема 3. Комфортный дом

Для комфортного проживания в доме, нужно чтобы воздух был чистым, умеренно влажным, теплым, а были растения. Но за этим всегда нужен контроль. Поэтому отслеживайте влажность и температуру воздуха. Освещенность в комнате. Чтобы растения всегда были вовремя политы. Нужно сообщать пользователю очень открыто и явно, если показатели вышли за границы, если нет, просто выводить на дисплей, чтобы пользователь был ознакомлен с ситуацией. Если света в комнате становится мало, то включить свет.

Данные с входных датчиков отображаются в приложении и подается сигнал на выходные устройства. Пользователю в приложении отображаются сообщения о датчиках:

- о влажности почвы и о необходимости предпринять те или иные меры;
- о темноте в помещении;
- о состоянии температуры и влажности в помещении и о необходимости предпринять различные меры.

Датчики (сенсоры) для съема данных	Устройства управления / отображения
1) DHT22 Temperature-Humidity Sensor (HDC1080) 2) Датчик влажности почвы 3) Фоторезистор	1) Светодиоды 2) Пьезоизлучатель

Тема 4. Управление предметами по будильнику

Хочется проснуться утром, а на кухне тебя уже ожидает свежесваренный кофе. Но мы пойдем пока не много меньше. Автоматизируем рутинную работу. Вам нужно в определенное время на улице включать свет и так же выключать. Если вы легли спать и забыли поставить офис на сигнализацию, включите датчик движения в нужное время. Ну и конечно пользователь должен видеть и регулировать время выставления будильников. Сделайте две кнопочки выставления времени для будильника (одна изменяет часы, другая изменяет минуты). А так же если на улице становится темно, включить во время свет в доме или на улице.

Данные с входных датчиков отображаются в приложении и подается сигнал на выходные устройства. Пользователю в приложении отображаются сообщения о датчиках:

- настройка и отображение времени в приложении;
- о темноте в помещении.

Датчики (сенсоры) для съема данных	Устройства управления / отображения
1) Часы реального времени DS3231 2) Фоторезистор	1) Светодиоды 2) Пьезоизлучатель

Тема 5. Защита от затопления дома

Трубы имеют свойство ржаветь, а стиральная машинка имеет свойство ломаться. И хорошо, если Вы будете в этот момент дома, а если нет, придется восстанавливать не только прохудившиеся трубы и стиральную машинку, но ещё и возмещать ремонт соседям и себе. Поэтому для контроля таких ситуаций будем использовать датчик влажности, который сообщит нам, о том, что где-то произошла протечка воды. А если Вы любите купаться в полные воды ванной, но при этом легко забываете вовремя выключить воду или Вас в этот момент могут отвлечь, то будем использовать датчик определения уровня воды. И конечно нужно сразу сигнализировать о произошедшей ситуации.

Данные с входных датчиков отображаются в приложении и подается сигнал на выходные устройства. Пользователю в приложении отображаются сообщения о датчиках:

- о уровне воды и о необходимости предпринять различные меры;
- о влажности почвы и о необходимости предпринять те или иные меры;
- о протечке в помещении и о необходимости предпринять различные меры.

Датчики (сенсоры) для съема данных	Устройства управления / отображения
1) Датчик уровня воды (угловой) 2) МОДУЛЬ ДАТЧИКА ПРОТЕЧКИ FC-37 3) Датчик влажности почвы	1) Светодиоды 2) Пьезоизлучатель

Тема 6. Комфортный дом для родных

Хорошо когда родные живут близко и есть возможность всегда поддерживать все показатели дома для комфортного и безопасного проживания. А что же делать, если хочется всегда знать состояние дома, но возможности нет!? Тогда на помощь придет система позволяющая отслеживать сложные показатели и сообщать о необходимости дополнительных действий. Система позволит измерять атмосферное давление и температуру в помещении, позволит контролировать влажность и при необходимости оповещать пользователей о важности проветривания комнаты. А также не мало важное измерять уровень пыли в здании. Ведь это так важно, особенно если в помещении маленькие дети.

Данные с входных датчиков отображаются в приложении и подается сигнал на выходные устройства. Пользователю в приложении отображаются сообщения о датчиках:

- о атмосферном давлении в помещении;
- о уровне запыленности помещения;
- о влажности и температуре помещения;
- о необходимости предпринять то или иное действие пользователю.

Датчики (сенсоры) для съема данных	Устройства управления / отображения
1) Датчик BMP180 2) Датчик пыли 3) Датчик HDC1080 (DHT22)	1) Светодиоды 2) Пьезоизлучатель