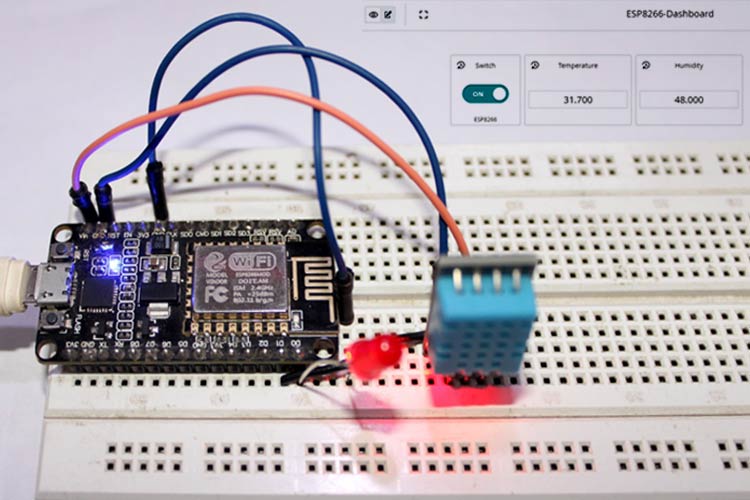
*28апр25*

<https://microkontroller.ru/esp8266-projects/nachalo-raboty-s-oblakom-arduino-cloud-iot-s-pomoshhyu-nodemcu-esp8266/?ysclid=m59jqakz8w209888156>

**Начало работы с облаком Arduino (Cloud IoT) с помощью NodeMCU ESP8266**

Опубликовано [25.12.2021](https://microkontroller.ru/esp8266-projects/nachalo-raboty-s-oblakom-arduino-cloud-iot-s-pomoshhyu-nodemcu-esp8266/) автором [admin-new](https://microkontroller.ru/author/admin-new/)

В 2019 году платформа Arduino представила мировому сообществу свое облачное хранилище данных Arduino Cloud IoT, которое значительно упрощает хранение данных и представляет полный комплекс услуг в рамках концепции интернета вещей (IoT). Оно включает в себя аппаратные и программные средства, облачные сервисы и базу знаний. Данное облачное хранилище данных позволяет пользователям собирать, анализировать и строить виде графиков данные от датчиков, обработчиков событий и т.д.



В данной статье мы рассмотрим подключение платы NodeMCU ESP8266 к облаку Arduino (Arduino Cloud IoT). В данное облако мы будем передавать данные, считываемые с датчика DHT11, также в этом облаке мы установим переключатель, с помощью которого мы будем управлять светодиодом, подключенным к NodeMCU.

**Общие сведения об облачном хранилище Arduino**

Платформа Arduino Cloud IoT позволяет пользователям писать коды своих программ, компилировать и загружать их в платы непосредственно из браузера, подключать устройства интернета вещей и использовать виртуальную приборную панель (dashboard) в режиме реального времени. Она автоматически формирует необходимый скетч при подключении к ней нового устройства. Платформа имеет следующие основные свойства.

**Автоматически генерируемый код**: платформа Arduino Cloud IoT автоматически генерирует скетч в соответствии с настройками, задаваемыми пользователем при подключении устройства к облаку и, таким образом, она устраняет барьеры для пользователей, которые не знакомы с программированием, и расширяет возможности разработчиков всех возрастов и опыта.

**Поддержка технологии Plug & Play (подключи и играй)**: используя облачную платформу интернета вещей Arduino, вы можете писать код, компилировать и загружать его прямо из своего браузера, подключать устройства Интернета вещей и создавать информационные панели в режиме реального времени. Таким образом, у вас не будет необходимости использовать Arduino IDE для программирования оборудования.

**Мобильная панель мониторинга "На ходу (‘On-the-go’ )"**: она позволяет пользователям получать доступ, проверять данные и управлять удаленным мониторингом датчиков из любого места с помощью доступных виджетов.

**Интеграция с другими платформами**: Arduino Cloud IoT может быть интегрирована с Amazon Alexa, Google Sheets, IFTTT и ZAPIER, что позволяет пользователям программировать и управлять устройствами с помощью голоса, электронных таблиц, баз данных и автоматизировать оповещения с помощью веб-хуков (webhooks). Платформа также позволяет разработчикам создавать собственные приложения с использованием API Интернета вещей Arduino с добавлением пользовательских веб-хуков в конечные точки для большей гибкости.

Обладая всеми этими функциями, облачный интернет вещей Arduino также имеет некоторые ограничения. Один из них заключается в том, что он работает только с определенными платами Arduino и модулями ESP. Все поддерживаемые платы перечислены ниже.

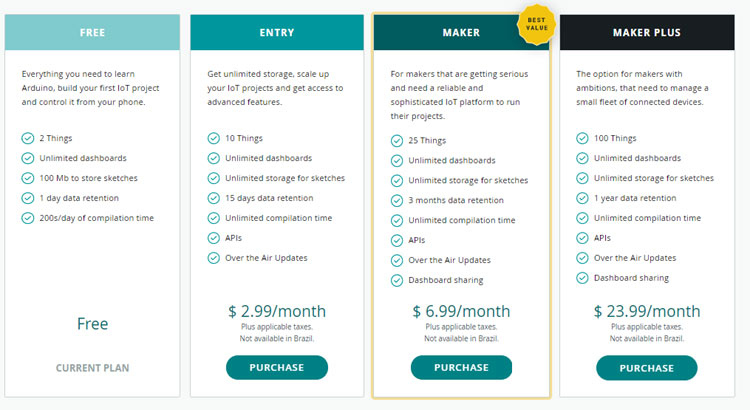
**Платы Arduino:**

* Arduino NANO 33 IoT;
* Arduino MKR Wi-Fi 1010;
* Arduino MKR WAN 1310;
* Arduino MKR WAN 1300;
* Arduino MKR NB 1500;
* Arduino MKR GSM 1400;
* Arduino MKR1000;
* Arduino Portenta H7 (M7 core);
* Arduino Nano RP2040 Connect.

**Модули ESP и другие платы:**

* ESP8266;
* ESP32;
* Pelion;
* Generic LoRa.

При бесплатном использовании Arduino Cloud IoT вы можете добавлять в платформу только два устройства и при этом время компиляции для вас будет ограничено 200 секундами в день. Если вы хотите расширить возможности использования платформы, то вы можете выбрать один из платных тарифов, представленных на рисунке ниже.



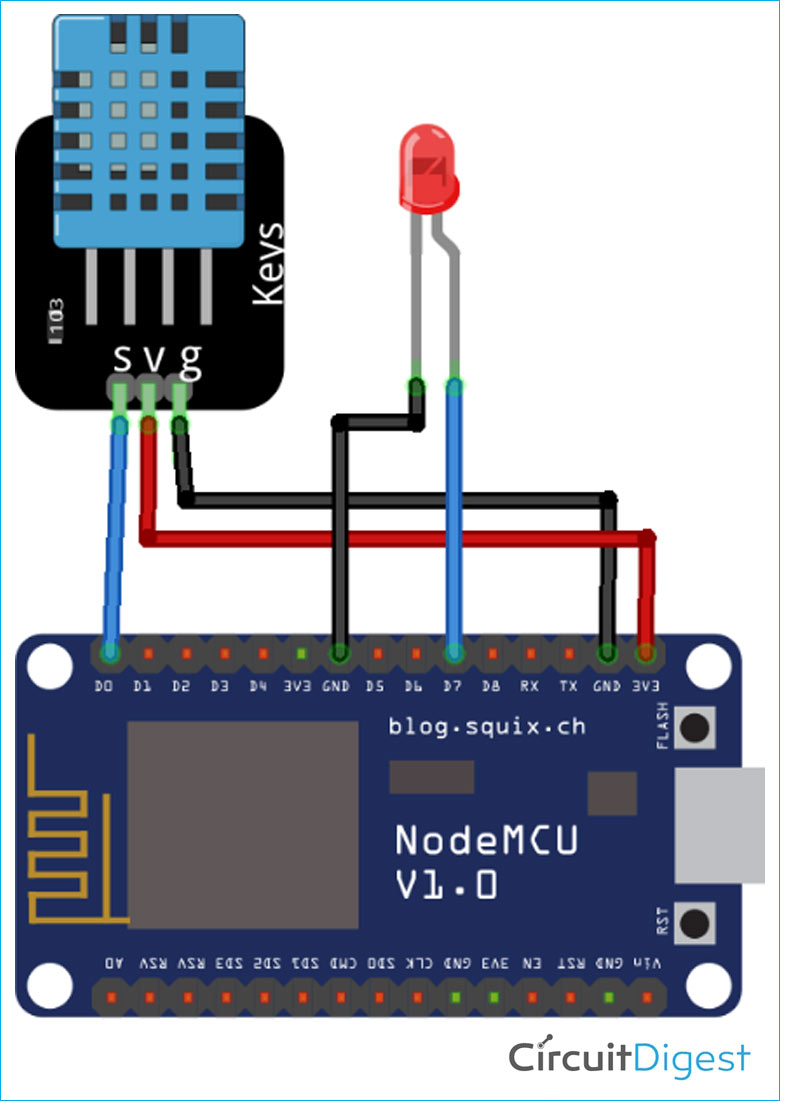
**Необходимые компоненты**

1. NodeMCU ESP8266 ([купить на AliExpress](https://alii.pub/6weo2l?erid=2SDnjc5mqdS)).
2. Датчик DHT11 ([купить на AliExpress](https://alii.pub/75qtbe?erid=2SDnjdsMX1E)).
3. Светодиод (5mm) ([купить на AliExpress](https://alii.pub/72p93p?erid=2SDnjdYXVum)).

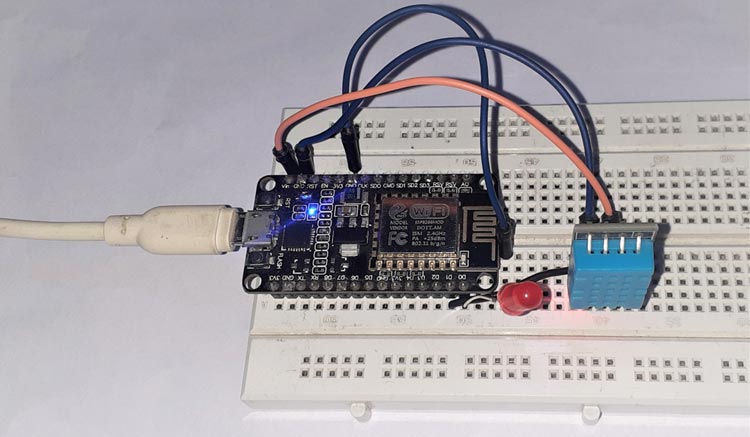
Реклама: ООО "АЛИБАБА.КОМ (РУ)" ИНН: 7703380158

**Схема проекта**

Схема для передачи данных от датчика DHT11 в облако Arduino с помощью NodeMCU ESP8266 представлена на следующем рисунке.

[](https://microkontroller.ru/wp-content/uploads/2021/12/esp8266-dht11-circuit-diagram.jpg)Соединения в схеме очень просты. Датчик DHT11 получает питание через контакты 5V и GND платы NodeMCU ESP8266. Контакт данных датчика DHT11 подключен к контакту D0 платы, а положительный вывод светодиода – к контакту D7 платы.

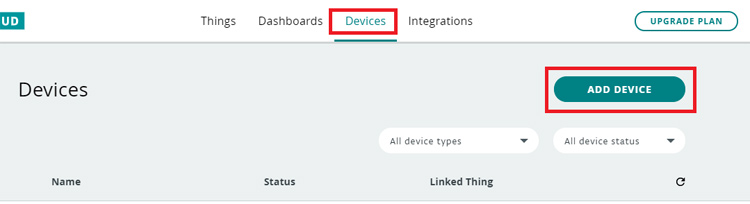
Внешний вид собранной конструкции проекта показан на следующем рисунке.



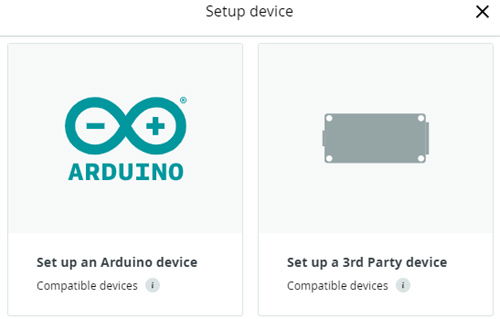
**Установка Arduino Cloud IoT**

Для начала вам необходимо создать себе аккаунт (если у вас его еще нет) на [платформе Arduino Cloud IoT](https://create.arduino.cc/iot/). После этого выполните следующую последовательность шагов.

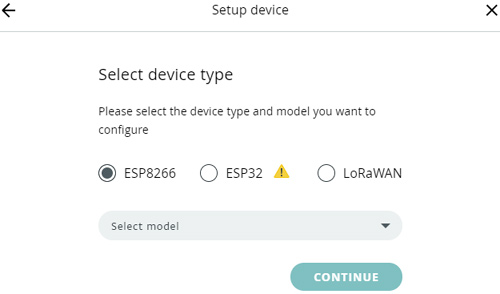
**Шаг 1**. После создания аккаунта добавьте в него новое устройство, а именно NodeMCU ESP8266. Для этого перейдите на вкладку "Devices" (устройства) и в ней нажмите "Add device".



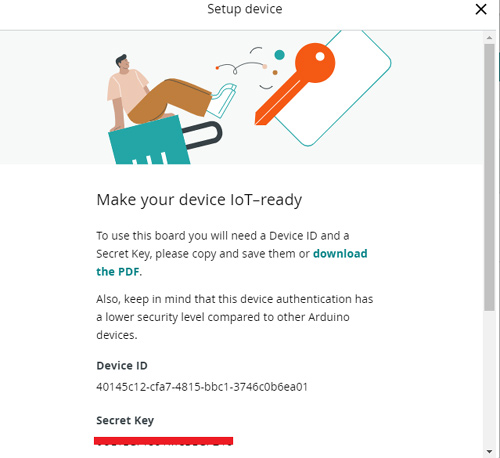
Затем нажмите на "Set up a 3rd party device".



На следующем шаге выберите используемую нами плату разработки. Для этого сначала в качестве типа устройства (device type) выберите ESP8266, а потом в выпадающем меню выберите NodeMCU1.0.

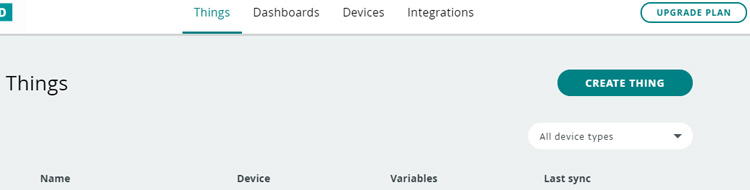


Далее переименуйте устройство по своему усмотрению и нажмите на next. Появится всплывающее окно с идентификатором устройства (Device ID) и секретным ключом (Secret Key) для вашего устройства. Вы можете либо вручную сохранить эти данные в блокноте либо нажать на “download the PDF” чтобы сохранить их в формате PDF. Учтите, что если вы потеряете (забудете) секретный ключ (secret key), то его невозможно будет восстановить.

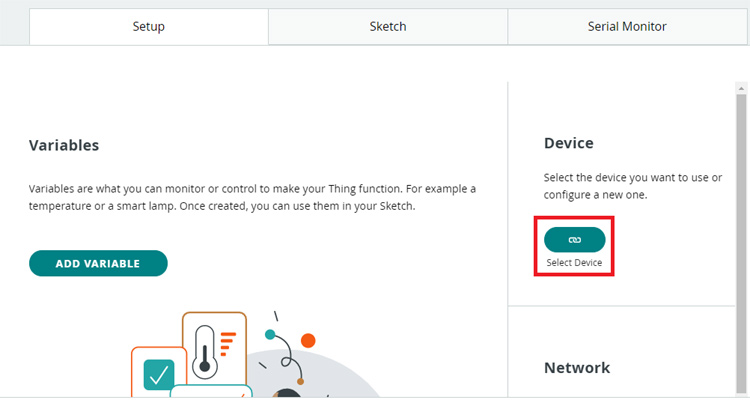


После этого нажмите на CONTINUE чтобы добавить устройство.

**Шаг 2**. После добавления устройства необходимо создать предмет/вещь (Thing). Для этого перейдите на вкладку "Things" и в ней нажмите на ‘Create Thing’.

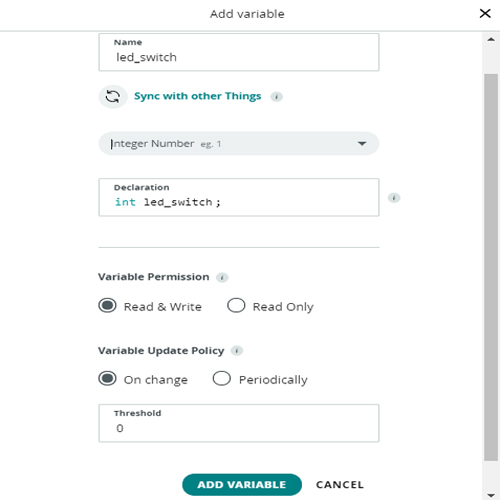


Теперь, на вкладке Things, у нас есть три опции, которые необходимо настроить – это Variables (переменные), Device (устройство) и Network (сеть). Но сначала свяжем наше устройство с нашим предметом (Thing). Это можно сделать нажав иконку связывания (выделена красным прямоугольником на рисунке ниже) в секции "Device". После этого откроется окно, в котором ранее созданное вами устройство (Device) должно быть доступно для выбора.

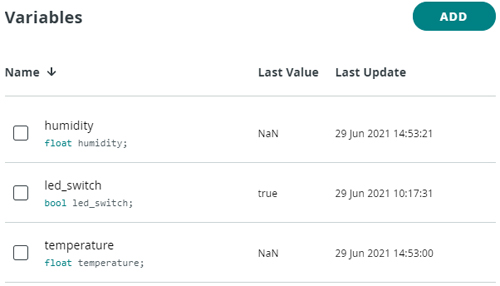


После того как устройство будет связано, нам необходимо будет добавить некоторые переменные, которые будут использоваться в коде программы. Для этого нажмите на кнопку "Add variable". После этого откроется окно, в котором вам необходимо заполнить информацию по переменной. В нашем проекте мы будем использовать три переменные, одна из которых будет использоваться для управления светодиодом, а две другие – для хранения значений температуры и влажности.

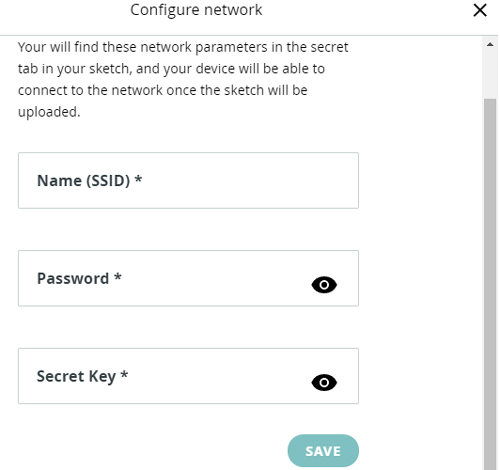
Сначала создадим переключатель светодиода (led\_switch). Тип данных у его будет целый (int), его значение можно будет только считывать (read-only), опция обновления (update policy) для него будет включена. После этого нажмите на кнопку "Add variable" (добавить переменную).



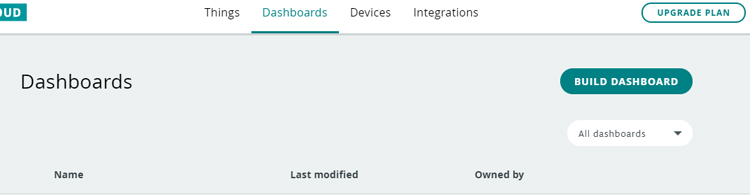
Аналогичным образом добавьте переменные для хранения значений температуры (‘temperature’) и влажности (‘humidity’). Тип данных для этих переменных будет float, их можно будет считывать и записывать (read & write), опция обновления для них также будет включена.



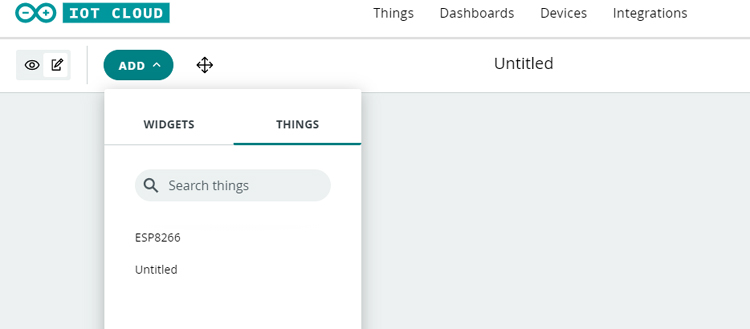
На следующем шаге необходимо ввести данные для доступа к вашей сети Wi-Fi (ее имя и пароль к ней) и секретный ключ (secret key), который был выдан вам во время установки вашего устройства. Для этого нажмите на кнопку ‘Configure’ в разделе "Network Section". Введите необходимую информацию и нажмите на ‘Save.’



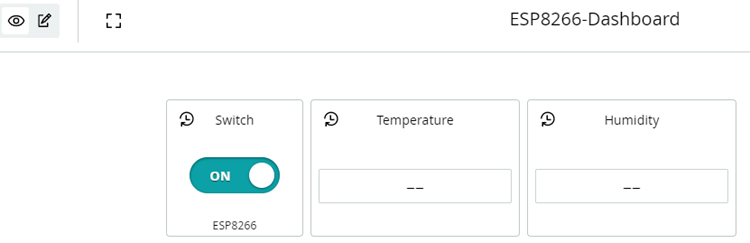
**Шаг 3**. После того как мы добавили устройство и переменные мы можем приступать к созданию информационной панели (dashboard). Для этого перейдите на вкладку ‘Dashboard’ и в ней нажмите на кнопку ‘Build Dashboard’.



Для добавления виджетов на информационную панель (dashboard) нажмите на иконку с карандашом в верхнем левом углу, затем нажмите на кнопку ‘Add’, затем перейдите на "Things" и выберите предмет (Thing). После этого выберите все переменные и нажмите на "Add widgets" (добавить виджеты).



В результате на экране вы должны получить следующую картину:

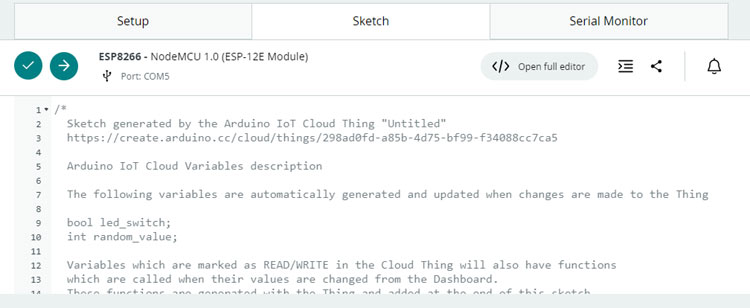


**Программирование NodeMCU ESP8266 для передачи данных в облако Arduino**

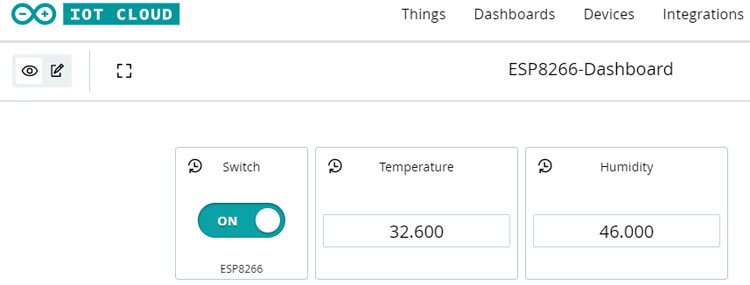
После того как мы закончили с настройками платформы Arduino Cloud IoT следующим нашим шагом будет программирование платы NodeMCU ESP8266 для считывания данных с датчика DHT11 и передачи их в облако Arduino (Arduino Cloud). Чтобы сделать это перейдите на вкладку "Sketch".

Когда вы добавляете любую переменную в предметы (Things), скетч в облаке автоматически обновляется в соответствии с добавленными переменными. Большая часть кода программы (скетча) будет автоматически написана платформой, нам необходимо будет всего лишь добавить несколько строк для работы с датчиком DHT11. Полный код программы приведен в конце статьи.

Когда код программы будет готов, выберите плату (Board) и порт (Port) и нажмите кнопку загрузить (upload) в верхнем левом углу.



После этого вы сможете управлять светодиодом и мониторить данные с датчика DHT11 используя платформу (Arduino Cloud IoT).



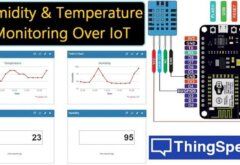
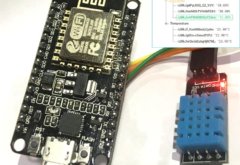
**Исходный код программы (скетча)**

|  |
| --- |
|  |
| #include "thingProperties.h"  #include "DHT.h"  #define DHTTYPE DHT11   // DHT 11  #define dht\_dpin 1  DHT dht(dht\_dpin, DHTTYPE);  int LED = D0;  void setup() {    // Initialize serial and wait for port to open:    Serial.begin(9600);    pinMode(LED, OUTPUT);    // This delay gives the chance to wait for a Serial Monitor without blocking if none is found (задержка для работы с последовательным портом)    delay(1500);    // Defined in thingProperties.h    initProperties();    // Connect to Arduino IoT Cloud (подключаемся к облаку)    ArduinoCloud.begin(ArduinoIoTPreferredConnection);    /\*       The following function allows you to obtain more information       related to the state of network and IoT Cloud connection and errors       the higher number the more granular information you’ll get.       The default is 0 (only errors).       Maximum is 4  \*/    setDebugMessageLevel(2);    ArduinoCloud.printDebugInfo();  }  void loop() {    ArduinoCloud.update();    // Your code here (ваш код необходимо вставить здесь)    temperature = dht.readTemperature();    humidity = dht.readHumidity();  }  void onLedSwitchChange() {    // Do something (делаем что-нибудь)    if(led\_switch){      digitalWrite(LED, LOW);    }    else{      digitalWrite(LED, HIGH);    }  }  void onTemperatureChange() {  }  void onHumidityChange() {  } | |

**Видео, демонстрирующее работу проекта**

[Источник статьи](https://circuitdigest.com/microcontroller-projects/getting-started-with-arduino-cloud-iot-connect-esp8266-to-arduino-cloud-iot)

**Похожие статьи**

[[](https://microkontroller.ru/esp8266-projects/nachalo-raboty-s-esp-now-esp8266-nodemcu-v-arduino-ide/)Начало работы с ESP-NOW (ESP8266 NodeMCU) в Arduino IDE](https://microkontroller.ru/esp8266-projects/nachalo-raboty-s-esp-now-esp8266-nodemcu-v-arduino-ide/) [[](https://microkontroller.ru/esp8266-projects/peredacha-s-pomoshhyu-nodemcu-esp8266-dannyh-temperatury-i-vlazhnosti-na-thingspeak/)Передача с помощью NodeMCU ESP8266 данных температуры и влажности на ThingSpeak](https://microkontroller.ru/esp8266-projects/peredacha-s-pomoshhyu-nodemcu-esp8266-dannyh-temperatury-i-vlazhnosti-na-thingspeak/) [[](https://microkontroller.ru/esp8266-projects/peredacha-s-pomoshhyu-nodemcu-esp8266-dannyh-temperatury-i-vlazhnosti-na-google-firebase/)Передача с помощью NodeMCU ESP8266 данных температуры и влажности на Google Firebase](https://microkontroller.ru/esp8266-projects/peredacha-s-pomoshhyu-nodemcu-esp8266-dannyh-temperatury-i-vlazhnosti-na-google-firebase/)