
Интернет вещей

— azure —

Задачи

- Подключить модуль NodeMCU к интернету
- Установить соединение с облачным IoT сервером azure
- Передавать телеметрию с датчика на сервер
- Принимать управляющие команды с IoT сервера

Подключение WiFi NodeMCU

```
#include <ESP8266WiFi.h>
```

```
...
```

```
WiFi.begin("Имя сети", "Пароль");
```

```
while (WiFi.status() != WL_CONNECTED)
```

```
{
```

```
    delay(500);
```

```
}
```

Протокол связи с IoT сервером

Используется сетевой протокол MQTT

Особенности:

- Работает поверх TCP/IP
- Архитектура издатель - подписчик
- Снижена нагрузка на сеть
- Работа с постоянным обрывом связи
- Нет ограничений на формат данных

Подключение библиотек

Для работы с IoT сервисом azure нужно:

ArduinoJson <https://github.com/bblanchon/ArduinoJson>

Time <https://github.com/PaulStoffregen/Time>

AzureIoTHubMQTTClient <https://github.com/andriyadi/AzureIoTHubMQTTClient>

Синхронизируй время

Перед подключением к сервису IoT, нужно установить точное время.

```
NTP.begin(NTP_DEFAULT_HOST);  
  
while (NTP.getTime() < 30000)  
{  
    delay(500);  
}
```

Подключаем библиотеку

```
#include <ESP8266WiFi.h> //вайфай
```

```
#include <WiFiClientSecure.h> //клиент для подключения
```

```
#include <AzureIoTHubMQTTClient.h> //сервер IoT
```

Создаем объект сервер

```
WiFiClientSecure tlsClient; //вайфай
```

```
AzureIoTHubMQTTClient client(tlsClient, IOTHUB_HOSTNAME,  
DEVICE_ID, DEVICE_KEY); //клиент для облака
```

IOTHUB_HOSTNAME - адрес подключения

DEVICE_ID - имя девайса на сервере IoT

DEVICE_KEY - ключ подключения

Привязка управляющих команд от сервера

//Формат команды таков:

```
{"Name": "[COMMAND_NAME]", "Parameters": [PARAMETERS_JSON_ARRAY]}
```

```
client.onCloudCommand("название команды", [функция обработки]);
```

Обработка команды

```
{"Name":"led","Parameters":{"Activated":0}}
```

```
void onLedCommand(String cmdName, JsonVariant jsonValue)
{
    JsonObject& jsonObject = jsonValue.as<JsonObject>();
    if (jsonObject.containsKey("Parameters"))
    {
        auto params = jsonValue["Parameters"];
        auto isAct = (params["Activated"]);
    }
}
```

Соединение с сервером IoT

```
void setup()
```

```
{
```

```
...
```

```
    client.begin();
```

```
...
```

```
}
```

Цикл обработки

```
void loop()  
{  
    client.run();  
  
    ...  
  
    delay(10);  
}
```

Проверка соединения

```
if (client.connected())  
{  
    //Есть связь  
    ...  
}
```

Формирование данных для отправки

```
AzureIoTHubMQTTClient::KeyValueMap keyVal =
```

```
{  
    {"MTemperature", temp},  
    {"MHumidity", humidity},  
    {"DeviceId", DEVICE_ID},  
    {"EventTime", currentTime}  
};
```

Отправка данных на сервер

```
client.sendEventWithKeyVal(keyVal);
```