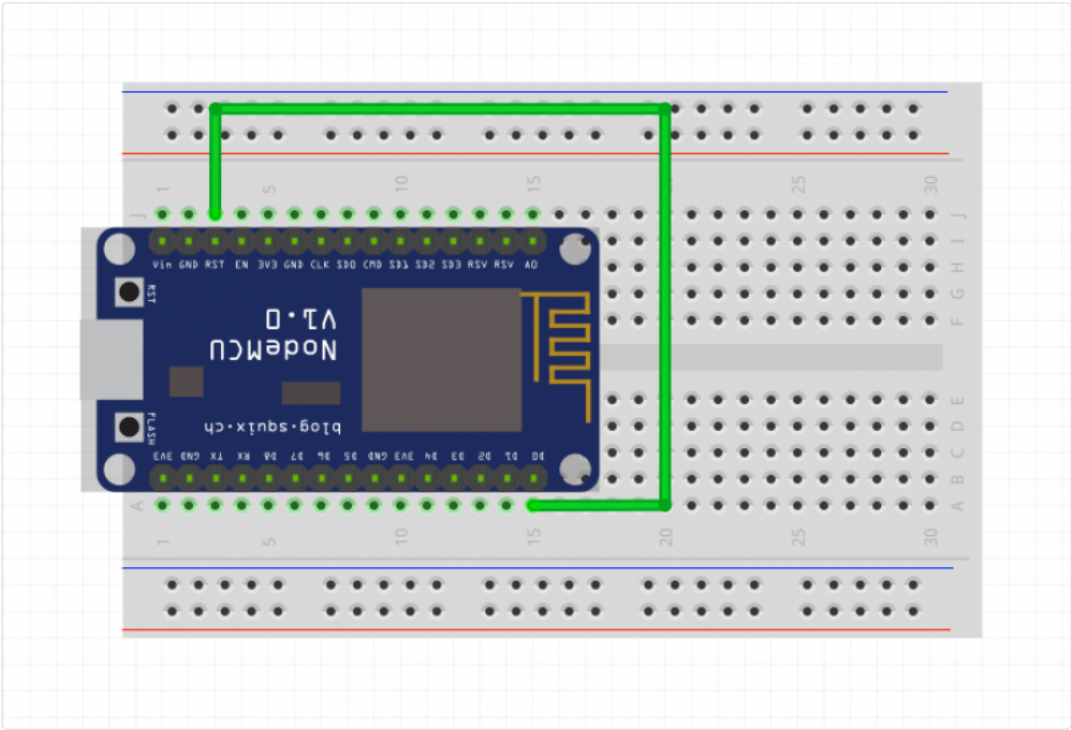


РЕЖИМ ЭКОНОМИИ ЭНЕРГИИ ESP8266 (DEEP SLEEP)

Режим экономии энергии ESP8266 (DEEP SLEEP)



(https://ngin.pro/uploads/posts/2017-08/1503646192_nodemcu-deep-sleep.png)
Итак, вы создали потрясающий проект, используя аппаратное обеспечение ESP8266, в котором есть LiPo аккумулятор, или вы просто подключили свой NodeMCU к USB-аккумулятору. Но после запуска вы понимаете, что питание от аккумулятора длится недолго.

Поскольку мы много времени не контролируем аппаратное обеспечение / компоненты в микроконтроллере, мы не можем оптимизировать оборудование. Но мы можем написать прошивку, которая будет отключать оборудование, чтобы сэкономить электроэнергию. Во время сна устройство потребляет гораздо меньше энергии, чем при пробуждении.

В этой статье мы сосредоточимся на режиме DEEP SLEEP (режим экономии электроэнергии) с ESP8266.

ТИПЫ СНА

Существует четыре режима сна для ESP8266: No-sleep, Modem-sleep, Light-sleep и Deep-sleep.

Table 1-1. Differences between 3 Sleep Modes			
Item	Modem-sleep	Light-sleep	Deep-sleep
Wi-Fi	OFF	OFF	OFF
System clock	ON	OFF	OFF
RTC	ON	ON	ON
CPU	ON	Pending	OFF
Substrate current	15 mA	0.4 mA	~ 20 µA
Average current	DTIM = 1	16.2 mA	1.8 mA
	DTIM = 3	15.4 mA	0.9 mA
	DTIM = 10	15.2 mA	0.55 mA

(https://ngin.pro/uploads/posts/2017-08/1503646212_esp8266_sleep_options.png)

Все они имеют разные функции.

NO-SLEEP

Установка No-Sleep режима значить что постоянно будет работать все оборудование. Очевидно, что это наиболее неэффективно и будет потреблять больше всего энергии.

MODEM-SLEEP

MODEM-SLEEP - это режим по умолчанию для ESP8266. Однако он включен только когда вы подключены к точке доступа.

В режиме «MODEM-SLEEP» ESP8266 отключает модем (WiFi) на как возможно дольше. Он отключает модем между интервалами DTIM Beacon. Этот интервал устанавливается вашим маршрутизатором.

LIGHT-SLEEP

Light-sleep выполняет ту же функцию, что и «MODEM-SLEEP», но также отключает системные часы и приостанавливает работу CPU. CPU не выключен. Это просто холостой ход.

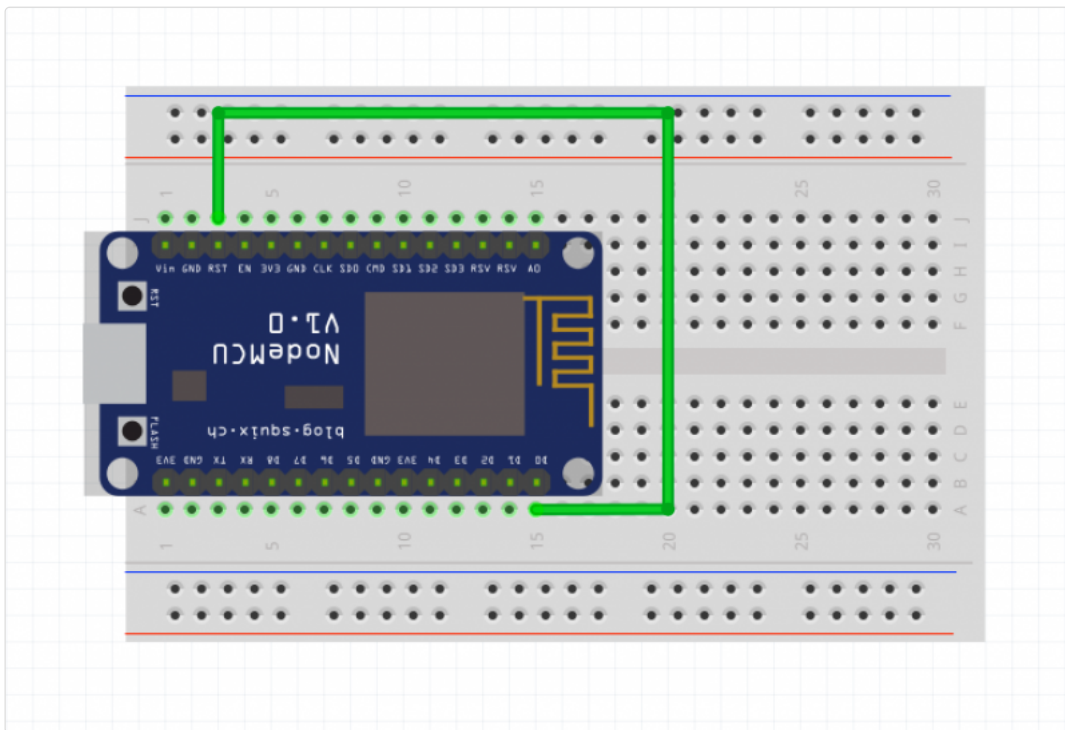
DEEP-SLEEP

Все выключено, но часы реального времени (RTC) включены потому что нужно контролировать время. Поскольку все выключено, это наиболее энергоэффективный вариант.

Если вам нужна дополнительная информация, обязательно ознакомьтесь с документацией (http://www.espressif.com/sites/default/files/9b-esp8266-low_power_solutions_en_0.pdf).

DEEP-SLEEP

MODEM-SLEEP и **LIGHT-SLEEP** полезны, если вам все еще нужно работать в ESP8266, и вам нужны дополнительные опции питания. Они также легко настраиваются в настройках WiFi ESP8266 (<https://github.com/esp8266/Arduino/blob/4897e0006b5b0123a2fa31f67b14a3fff65ce561/doc/esp8266vclass.md#other-function-calls>). Но, если вам нужно какое-то серьезное управление мощностью, Deep-sleep - это ваш вариант.



(https://nginx.pro/uploads/posts/2017-08/1503646192_nodemcu-deep-sleep.png)

С Deep-sleep структура наших приложений будет выполнять следующие шаги:

Выполните некоторые действия (считывание информации с датчика)
 Сон на n микросекунд
 Повторение

Важно отметить, что время сна указано в микросекундах (мкс).

Забавный факт, вы не можете отключить ESP8266 на вечно. Согласно SDK ESP8266, вы можете отключить его только на 4 294 967 295 мкс, что составляет около ~ 71 минуты.

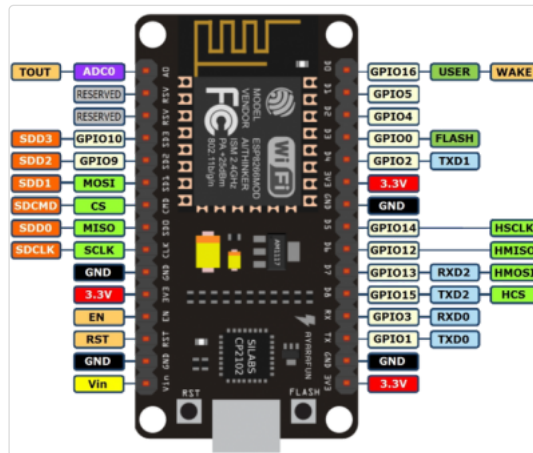
Теперь давайте посмотрим код. В этих примерах я собираюсь использовать IDE Arduino.

Давайте рассмотрим простой пример:

Несмотря на то, что мы ничего не подключаем, чтобы включить Deep-sleep, нам нужно связать контакт RST с GPIO 16 на ESP8266. На NodeMCU GPIO 16 представлен как D0.

Если мы посмотрим на распиновку для NodeMCU, мы увидим, что GPIO 16 является специальным выводом:

Вывод RST поддерживается сигналом HIGH при запуске ESP8266. Однако, когда контакт RST получает сигнал LOW, он перезапускает микроконтроллер. Когда ваше устройство находится в режиме глубокого сна, он отправит сигнал LOW на GPIO 16, когда таймер сна будет включен. Вам необходимо подключить GPIO 16 к RST для пробуждения (или сброса) устройства, когда Deep-sleep закончился.



(https://nginx.pro/uploads/posts/2017-08/1503646197_nodemcu_pins.png)

Вот пример кода:

```
/**
 * An example showing how to put ESP8266 into Deep-sleep mode
 */

void setup() {
  Serial.begin(115200);
  Serial.setTimeout(2000);
  // Wait for serial to initialize.
  while(!Serial) { }

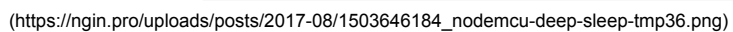
  Serial.println("I'm awake.");
  Serial.println("Going into deep sleep for 20 seconds");
  ESP.deepSleep(20e6); // 20e6 is 20 microseconds
}

void loop() {
}
```

В этом примере мы заходим в Serial, спим в течение 20 секунд и повторяем. Вы можете использовать этот пример в качестве шаблона для других программ.

DEEP-SLEEP W/ ПОДКЛЮЧЕНИЕ

В реальном мире мы хотим выполнить действие, например, сделать сетевой запрос, в то время как устройство отключено. Давайте рассмотрим пример отправки показаний датчика температуры в Losant каждые 20 секунд и сон между ними.



<https://ngin.pro/smart-house/300-rezhim-ekonomii-energii-esp8266-deep-sleep.html>

```
/**
 * An example showing how to put ESP8266 into Deep-sleep mode
 */
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <Losant.h>
// WiFi credentials.
const char* WIFI_SSID = "wifi-ssid";
const char* WIFI_PASS = "wifi-pass";
// Losant credentials.
const char* LOSANT_DEVICE_ID = "device-id";
const char* LOSANT_ACCESS_KEY = "access-key";
const char* LOSANT_ACCESS_SECRET = "access-key";
WiFiClientSecure wifiClient;
LosantDevice device(LOSANT_DEVICE_ID);
void connect() {
    // Connect to Wifi.
    Serial.println();
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(WIFI_SSID);
    WiFi.begin(WIFI_SSID, WIFI_PASS);
    // WiFi fix: https://github.com/esp8266/Arduino/issues/2186
    WiFi.persistent(false);
    WiFi.mode(WIFI_OFF);
    WiFi.mode(WIFI_STA);
    WiFi.begin(WIFI_SSID, WIFI_PASS);
    unsigned long wifiConnectStart = millis();
    while (WiFi.status() != WL_CONNECTED) {
        // Check to see if
        if (WiFi.status() == WL_CONNECT_FAILED) {
            Serial.println("Failed to connect to WiFi. Please verify credentials: ");
            delay(10000);
        }
        delay(500);
        Serial.println("...");
        // Only try for 5 seconds.
        if (millis() - wifiConnectStart > 15000) {
            Serial.println("Failed to connect to WiFi");
            return;
        }
    }
    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
    Serial.println();
    Serial.print("Connecting to Losant...");
    Serial.print("Authenticating Device...");
    HTTPClient http;
    http.begin("http://api.losant.com/auth/device");
    http.addHeader("Content-Type", "application/json");
    http.addHeader("Accept", "application/json");
    /* Create JSON payload to sent to Losant
     {
       "deviceId": "575ecf887ae143cd83dc4aa2",
       "key": "this_would_be_the_key",
       "secret": "this_would_be_the_secret"
     }
    */
    StaticJsonBuffer<200> jsonBuffer;
    JsonObject& root = jsonBuffer.createObject();
    root["deviceId"] = LOSANT_DEVICE_ID;
    root["key"] = LOSANT_ACCESS_KEY;
    root["secret"] = LOSANT_ACCESS_SECRET;
    String buffer;
    root.printTo(buffer);
    int httpCode = http.POST(buffer);
    if (httpCode > 0) {
        if (httpCode == HTTP_CODE_OK) {
            Serial.println("This device is authorized!");
        } else {

```

```
Serial.println("Failed to authorize device to Losant.");
if (httpCode == 400) {
    Serial.println("Validation error: The device ID, access key, or access secret is not in the proper format.");
} else if (httpCode == 401) {
    Serial.println("Invalid credentials to Losant: Please double-check the device ID, access key, and access secret.");
} else {
    Serial.println("Unknown response from API");
}
Serial.println("Current Credentials: ");
Serial.println("Device id: ");
Serial.println(LOSANT_DEVICE_ID);
Serial.println("Access Key: ");
Serial.println(LOSANT_ACCESS_KEY);
Serial.println("Access Secret: ");
Serial.println(LOSANT_ACCESS_SECRET);
return;
}
} else {
    Serial.println("Failed to connect to Losant API.");
    return;
}
http.end();
device.connectSecure(wifiClient, LOSANT_ACCESS_KEY, LOSANT_ACCESS_SECRET);
while (!device.connected()) {
    delay(1000);
    Serial.print(".");
}
Serial.println("Connected!");
Serial.println("This device is now ready for use!");
}

void reportTemp(double degreesC, double degreesF) {
    StaticJsonBuffer<200> jsonBuffer;
    JsonObject& root = jsonBuffer.createObject();
    root["tempC"] = degreesC;
    root["tempF"] = degreesF;
    device.sendState(root);
}

void setup() {
    Serial.begin(115200);
    Serial.setTimeout(2000);
    // Wait for serial to initialize.
    while (!Serial) { }
    Serial.println("Device Started");
    Serial.println("-----");
    Serial.println("Running Deep Sleep Firmware!");
    Serial.println("-----");
    connect();
    int temp = analogRead(A0);
    double degreesC = (((temp / 1024.0) * 3.2) - 0.5) * 100.0;
    double degreesF = degreesC * 1.8 + 32;
    Serial.println();
    Serial.print("Temperature C: ");
    Serial.println(degreesC);
    Serial.print("Temperature F: ");
    Serial.println(degreesF);
    Serial.println();
    reportTemp(degreesC, degreesF);
    Serial.println("Going into deep sleep for 20 seconds");
    ESP.deepSleep(20e6); // 20e6 is 20 microseconds
}

void loop() {
}
```

В этом примере мы выполняем следующие шаги:

Подключение к Wi-Fi (если не подключено)

Считывание датчика температуры из аналогового

Режим сна - 1мин

Повторение

Вывод

Теперь, когда вы знаете, как использовать режим экономии энергии (**DEEP-SLEEP**), вы можете сделать свое оборудование более энергоэффективным.

{POSTER_AVATAR}

ADMIN (HTTPS://NGIN.PRO/USER/ADMIN/)

11 426




ODMEN

16:41, 25 ЯНВАРЬ

норм. давайте статью о лайСлип

0

Ответить



TAPAC

09:16, 06 АПРЕЛЬ



Из практики. Nodemcu v3, дип слип максимальный 2100 * 1000000 микросекунд, т.е. 35 минут. Когда делал 2150 устройство просто не засыпает и все.

1

Ответить

Имя

E-Mail

B *I* U ~~S~~         

0

Я не робот

reCAPTCHA

Конфиденциальность - Условия использования

Добавить

(<https://plus.google.com/u/0/b/114310883523146199347/114310883523146199347/posts>) (<https://twitter.com/NginPro>) (https://www.facebook.com/Arduino-Geek-1129478897159111/?view_public_for=1129478897159111) (<https://vk.com/arduino geek>)