

# Deep-Sleep

## verwendete Bauteile

- [ESP8266 Mikrokontroller](#)

## Code

```
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>

char ssid[] = "WLAN-SSID";          // your network SSID (name)
char password[] = "WLAN-PASS"; // your network password
#define TELEGRAM_BOT_TOKEN "yyyyyy-zzzz"

// This is the Wifi client that supports HTTPS
WiFiClientSecure client;
UniversalTelegramBot bot(TELEGRAM_BOT_TOKEN, client);

void setup() {
  boolean sleep = false;
  boolean first = true;

  Serial.begin(115200);
  Serial.println("initial");
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(LED_BUILTIN, LOW);

  // Set WiFi to station mode and disconnect from an AP if it was previously
  // connected
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(100);

  // attempt to connect to Wifi network:
  Serial.print("Connecting Wifi: ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
}
```

```
Serial.println("WiFi connected");
Serial.print("IP address: ");
Serial.println(WiFi.localIP());

// Only required on 2.5 Beta
client.setInsecure();

bot.longPoll = 10;

int numNewMessages = bot.getUpdates(bot.last_message_received + 1);

while (true) {
  numNewMessages = bot.getUpdates(bot.last_message_received + 1);
  if (!numNewMessages && sleep) {
    Serial.println("sleeping");
    ESP.deepSleep(60e6); //sleep for 10 seconds max 71min
  }
  else {
    Serial.println("else");
    for (int i = 0; i < numNewMessages; i++) {
      String chat_id = String(bot.messages[i].chat_id);
      String text = bot.messages[i].text;

      if (first) {
        if (sleep) {
          bot.sendMessage(chat_id, "sleep : true\n", "Markdown");
        } else {
          bot.sendMessage(chat_id, "sleep : false\n", "Markdown");
        }
        boolean first = false;
      }

      if (text == F("/ip")) {
        bot.sendMessage(chat_id, "local IP: 127.0.0.1\n", "Markdown");
      }
      else if (text == F("/temp")) {
      }
      else if (text == F("/wake")) {
        bot.sendMessage(chat_id, "sleep : false\n", "Markdown");
        sleep = false;
      }
      else if (text == F("/sleep")) {
        bot.sendMessage(chat_id, "sleep : true\n", "Markdown");
        sleep = true;
      }
      else {
        Serial.println("else");
        if (sleep) {
          bot.sendMessage(chat_id, "sleep : true\n", "Markdown");
        } else {
```

```
        bot.sendMessage(chat_id, "sleep : false\n", "Markdown");
    }
    bot.sendMessage(chat_id, "/sleep : go to sleep for 1 minute\n",
"Markdown");
    bot.sendMessage(chat_id, "/wake : stay awake\n", "Markdown");
    }
    }
    }
}

void loop() {
}
```

From:

<https://wiki.haberland.it/> - **haberland.it**

Permanent link:

<https://wiki.haberland.it/doku.php?id=projekte.haberland.it:arduino:projekte:esp-deep-sleep&rev=1554477636>

Last update: **2020/05/12 11:44**

