

Vývojová deska ESP8266 s OLED displejem

1. POPIS

Vývojová deska s WiFi modulem ESP8266EX umožňuje bezdrátovou komunikaci, analyzování WiFi sítí nebo podporu méně náročného webserveru. Pro čtení výstupních dat má modul navíc jednoduchý a dobře čitelný OLED displej. Tato vývojová platforma je vhodná pro průmyslové účely, domácí automatizaci nebo edukativní účely v rámci tvoření jednodušších uzlů pro čtení a odesílání naměřených dat, analyzování sítě, poskytování síťových služeb či ovládání jednodušších prvků automatizace.

Základní charakteristika:

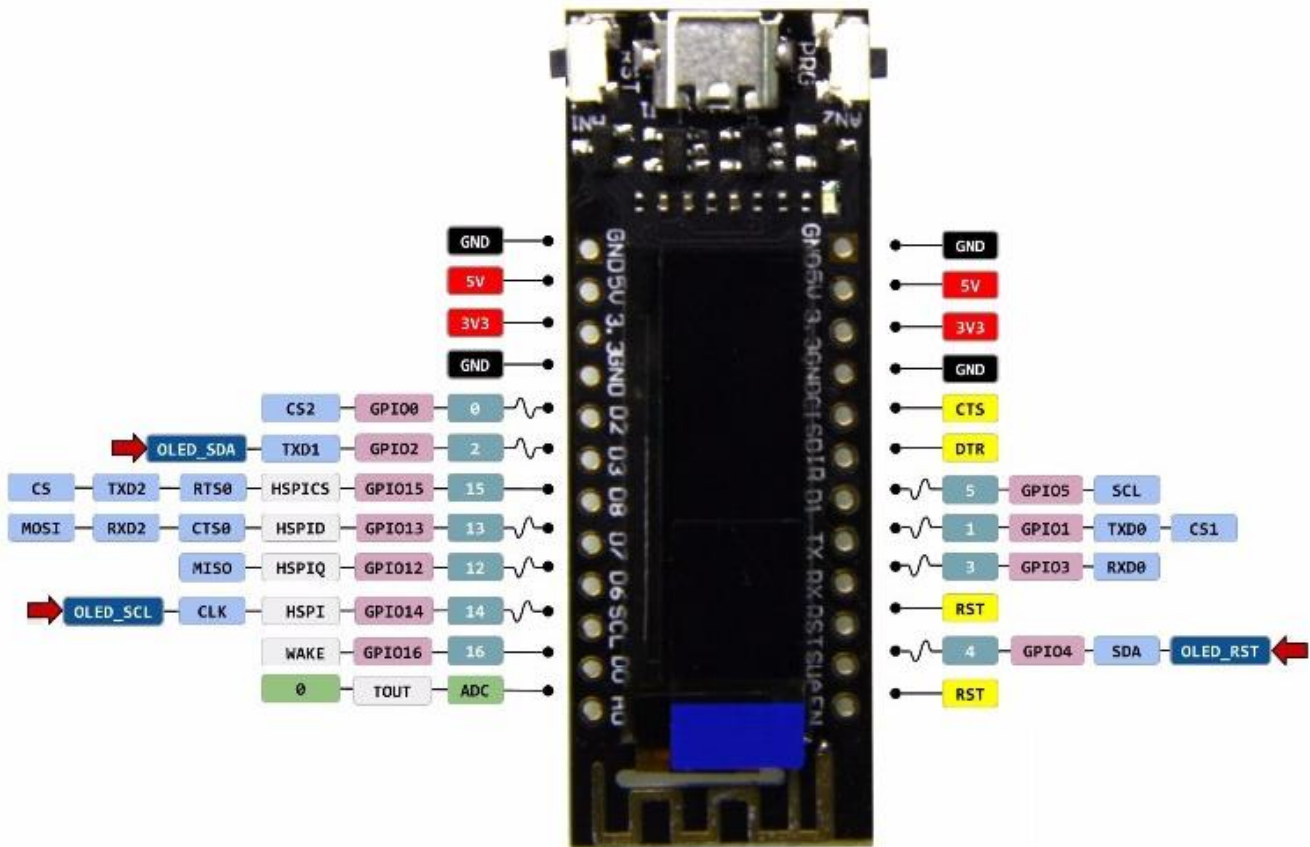
- komunikační protokoly FTP, HTTP, IPv4, TCP, UDP
- frekvenční pásmo 2,4 GHz
- OLED displej s úhlopříčkou 0,91"
- integrovaná WiFi anténa
- kompaktní rozměry



2. SPECIFIKACE

Hlavní čip	ESP8266EX	Vysílací výkon (802.11b)	20 dBm
Napájení USB	5 V	Standardy	IEEE 802.11b/g/n
Napájení VIN	3,3 až 7 V	Frekvenční pásmo	2,4 GHz
Nap. úroveň logiky	3,3 V	Rozlišení displeje	128 x 32 px
Pracovní proud	až 250 mA	Kontrolér displeje	SSD1306
Počet I/O pinů	9	Úhlopříčka displeje	0,91"
USB převodník	CP210X	Rozměry modulu (mm)	51 x 19 x 7

3. ZAPOJENÍ



Displej je zapojen na piny D3 (SDA), SCL (SCL) a SDA (RST).

Ovladače

Pro správnou emulaci COM port zařízení v PC je nutné stáhnout ovladač pro USB převodník CP210X, který je volně k dispozici na internetu.

Knihovny ESP8266 a U8g2lib

Vlastnosti Arduino IDE – vyplnění URL adresy

Správce dalších desek URL:

http://arduino.esp8266.com/staging/package_esp8266com_index.json

Stáhnutí knihovny ESP8266 a U8g2lib

Nainstalovat tyto knihovny lze přes správce knihoven v Arduino IDE – do vyhledávacího pole stačí napsat názvy knihoven a poté je nainstalovat.

Nahrávání programu

Tlačítko PRG

Pokud má uživatel v úmyslu nahrát do modulu program, musí během zasunutím vývojové platformy do USB držet stisknuté tlačítko PRG, které se nachází na desce modulu.

Tlačítko RST

Provádí restart zařízení.

Nastavení Arduino IDE

Vývojová deska: "NodeMCU 0.9 (ESP-12 Module)"	>
Flash Size: "4M (1M SPIFFS)"	>
Debug port: "Disabled"	>
Debug Level: "Žádný"	>
lwIP Variant: "v2 Lower Memory"	>
VTables: "Flash"	>
CPU Frequency: "80 MHz"	>
Upload Speed: "115200"	>
Erase Flash: "Only Sketch"	>
Port	>
Získat informace o Desce	

Nastavení portu je individuální. Uživatel nastaví takový port, který je zařízení přidělen.

00101 01001 00001 4. UKÁZKA PROGRAMU

Pro správnou funkci je nutné vyplnit "wifi_ssid", tedy název WiFi sítě uživatele a "wifi_password", což je heslo dané WiFi sítě.

```
#include <ESP8266WiFi.h> //For ESP8266
#include <ESP8266mDNS.h> //For OTA
#include <WiFiUdp.h> //For OTA
#include <ArduinoOTA.h> //For OTA
#include <Arduino.h>
#include <U8g2lib.h> // make sure to add U8g2 library and restart Arduino IDE
#include <SPI.h>
#include <Wire.h>

#define OLED_SDA 2
#define OLED_SCL 14
#define OLED_RST 4
U8G2_SSD1306_128X32_UNIVISION_F_SW_I2C u8g2(U8G2_R0, OLED_SCL, OLED_SDA, OLED_RST);
const char *text = "U8g2 OTA Demo"; // scroll this text from right to left

#define wifi_ssid "*****"
```

```

#define wifi_password "*****"
#define WiFi_hostname "ESP8266-TTGO"

//Necessary to make Arduino Software autodetect OTA device
WiFiServer TelnetServer(8266);

void setup_wifi() {
  delay(100);
  Serial.println("");
  Serial.print("Connecting to ");
  Serial.println(wifi_ssid);
  WiFi.hostname(WiFi_hostname);
  WiFi.begin(wifi_ssid, wifi_password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.print(" IP address: ");
  Serial.println(WiFi.localIP());
  Serial.print("Configuring OTA device...");
  TelnetServer.begin(); //Necessary to make Arduino Software autodetect OTA device

  ArduinoOTA.onStart([]() {Serial.println("OTA starting...");});
  ArduinoOTA.onEnd([]() {Serial.println("OTA update finished!");Serial.println("Rebooting...");});
  ArduinoOTA.onProgress([](unsigned int progress, unsigned int total) {Serial.printf("OTA in progress: %u%%\r\n", (progress /
(total / 100)));});
  ArduinoOTA.onError([](ota_error_t error) {
    Serial.printf("Error[%u]: ", error);
    if (error == OTA_AUTH_ERROR) Serial.println("Auth Failed");
    else if (error == OTA_BEGIN_ERROR) Serial.println("Begin Failed");
    else if (error == OTA_CONNECT_ERROR) Serial.println("Connect Failed");
    else if (error == OTA_RECEIVE_ERROR) Serial.println("Receive Failed");
    else if (error == OTA_END_ERROR) Serial.println("End Failed");
  });
  ArduinoOTA.begin();
  Serial.println("Wifi OK");
}

void setup() {
  Serial.begin(9600);
  setup_wifi();
  u8g2.begin();
}

void loop() {
  ArduinoOTA.handle();
  u8g2.clearBuffer(); // clear the internal memory
  u8g2.setFont(u8g2_font_8x13B_mf); // choose a suitable font
  u8g2.drawStr(0,10,text); // write something to the internal memory
  IPAddress myip= WiFi.localIP();
  String sFullip = String(myip[0] + "." + myip[1] + "." + myip[2] + "." + myip[3]);
  u8g2.drawStr(0,28,sFullip.c_str()); // write something to the internal memory
  u8g2.sendBuffer(); // transfer internal memory to the display
  delay(1000);
}

```