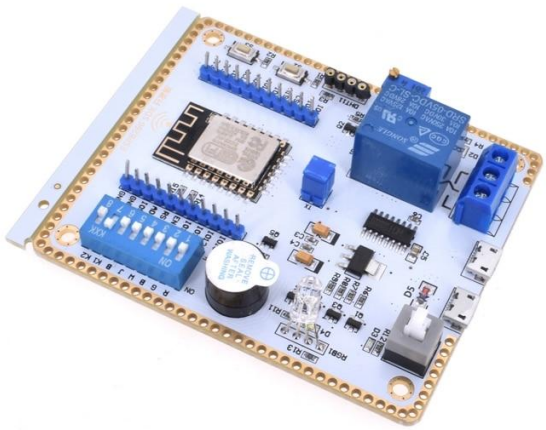
ESP8266 WiFi vývojová platforma

POPIS

Jedná se o vývojovou platformu, která se skládá z kontroléru s čipem ESP8266 a prototypové desky. Deska je osazena základními komponentami a moduly, které umožní uživateli vyzkoušet si možnosti platformy ESP8266. Komponenty jsou připojeny na přepínač, díky kterému je lze deaktivovat.



Základní charakteristika:

- Vývojová deska ESP8266 s anténou
- Prototypová deska s komponentami:
  - Relé
  - Bzučák
  - RGB LED a bílá LED dioda
  - 10K Potenciometr
  - Modul pro měření teploty/vlhkosti
  - Vypínací přepínač

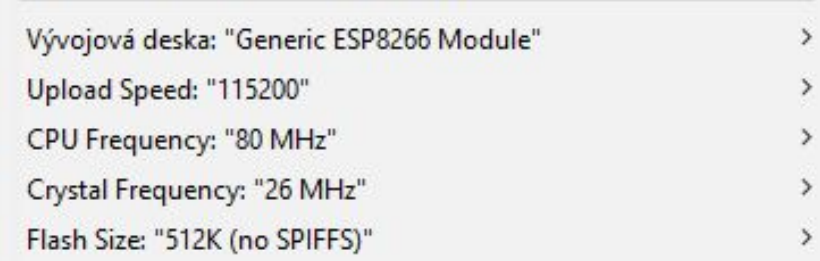
SPECIFIKACE

Hlavní čip	ESP8266	**Rozsah měření vlhk. vzduchu**	20 až 95 % RH
Napájení	5 V	**Přesnost měření teploty**	± 2 °C
Logická napěťová úroveň	3,3 V	**Přesnost měření vlhkosti**	± 5 % RH
USB TTL	CH340G	**Typ bzučáku**	aktivní
Relé	SRD-05VDC-SL-C	**Rozměry desky**	84 x 70 x 17 mm
Spínaný proud (při 24 VDC)	do 5 A	**Rozměry mont. otvorů**	3 mm
Snímač teploty	DHT11	**Rozteč mont. otvorů**	72 x 58 mm
Rozsah měření teplot	0 až 50 °C	**Hmotnost**	46 g

PROGRAMOVÁNÍ

Před připojením zařízení k PC (přes mini USB) je nutné nainstalovat ovladače pro USB TTL převodník CH340G. Ovladače lze stáhnout z popisu produktu na e-shopu.
Aby byl uživatel schopný programovat platformu ESP8266 za použití vývojového prostředí Arduino IDE, je nutné aby stáhl balíček **esp8266**. Jednotlivé kroky dané procedury jsou uvedeny v článku, na který je odkazováno na stránce produktu. Po instalaci balíčku následují další kroky:

Nastavení Arduino IDE
V nabídce vývojových desek vybere uživatel modul Generic ESP8266 Module. Další parametry již není nutné měnit (kromě COM portu).



© 2019

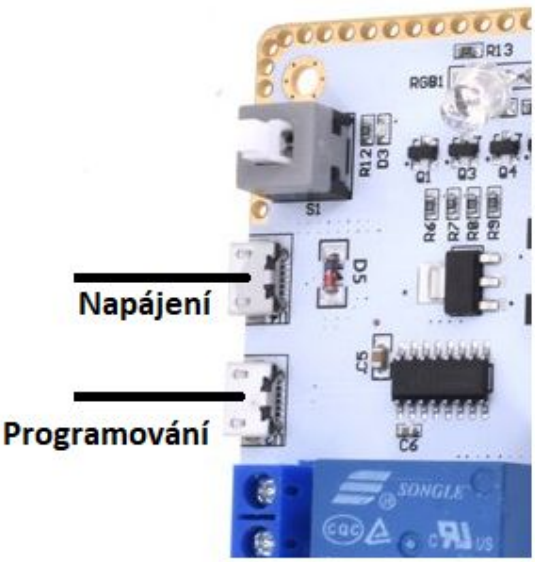
Nahrávání programu
V případě, že má uživatel v úmyslu nahrát program do ESP8266, musí před připojením napájení přepnout přepínače **K1 a K2 do stavu ON**. V případě, že by tak nebylo učiněno, nelze do desky nahrát program. Pokud se přepínače K1 a K2 nachází ve stavu ON, může uživatel zařízení připojit k PC a nahrát do něj program.
Po nahrání programu uživatel musí vrátit přepínače K1 a K2 do polohy OFF a modul restartovat. Po restartu se již spustí nahraný program.

Přepínače
Přepínač S1 slouží pro zapínání a vypínání vývojové desky.

Komponenta s osmi přepínači slouží pro deaktivaci některých komponent (stav OFF) a vyvolání programovacího režimu:

- **R**: Vypne červené světlo na RGB LED diodě
- **G:** Vypne zelené světlo na RGB LED diodě
- **B:** Vypne modré světlo na RGB LED diodě
- **W**: Vypne bílou LED diodu
- **J**: Deaktivuje relé
- **B**: Deaktivuje bzučák
- **K1 a K2**: slouží pro spuštění programovacího režimu (jsou spojeny s tlačítky S2 a S3)

USB konektory



© 2019

UKÁZKA PROGRAMU

Ke správné kompilaci programu je nutné stáhnout knihovnu **DHT.h**.

```
#include "DHT.h"
#define RELE 16
#define BILA_LED 14
#define BUZZER 5
#define TRIMMER A0 //snižování - CCW
#define DHT_PIN 4
//RGB LED
#define CERVENA 15
#define ZELENA 13
#define MODRA 12

uint8_t counter = 1;
DHT dht(DHT_PIN, DHT11);

void setup() {
  delay(500);
  dht.begin();
  Serial.begin(9600);
  pinMode(RELE, OUTPUT);
  pinMode(CERVENA, OUTPUT);
  pinMode(ZELENA, OUTPUT);
  pinMode(MODRA, OUTPUT);
  pinMode(BILA_LED, OUTPUT);
  pinMode(BUZZER, OUTPUT);
  Serial.println("\nTestovani periferii modulu");
  Serial.println("**************************");
}

void loop() {
  testFunction();
}

void testFunction(){
  switch(counter){
    case 1:
    pinHandler(CERVENA, "RGB LED - CERVENA");
    break;
    case 2:
    pinHandler(ZELENA, "RGB LED - ZELENA");
    break;
    case 3:
    pinHandler(MODRA, "RGB LED - MODRA");
    break;
```


© 2019

```
    case 4:
    pinHandler(BILA_LED, "BILA LED");
    break;
    case 5:
    pinHandler(RELE, "RELE SEPNUTO");
    break;
    case 6:
    pinHandler(BUZZER, "VYSTRAZNY ZVUK");
    break;
    case 7:
    Serial.print("Hodnota analog. vstupu - trimmer: ");
    Serial.println(analogRead(TRIMMER));
    delay(3000);
    break;
    case 8:
    Serial.print("Teplota: ");
    Serial.println(dht.readTemperature());
    delay(3000);
    break;
    default:
    counter = 0;
    break;
  }
  counter++;
}

void pinHandler(uint8_t pinNum, const char* str){
  digitalWrite(pinNum, HIGH);
  Serial.println(str);
  delay(3000);
  digitalWrite(pinNum, LOW);
}
```


© 2019