
INTERNET VĚCÍ V PRAXI

INTERNET OF THINGS IN PRACTICE

Milan ZAJÍČEK
ÚTIA AV ČR, v.v.i., zajicek@utia.cas.cz

Anotace

Předpokladem práce s mikrokontrolery je pochopení základních principů jejich komunikace s okolím a možné způsoby jejich programování. Mikrokontroler ESP8266 je jedním z neznámějších obvodů používaných v aplikacích pro internet věcí. Jednou z vývojových platforem, která se o ESP8266 opírá, je modulární systém Wemos D1 mini, který umožňuje přímou komunikaci s PC prostřednictvím USB portu a disponuje dostatkem vstupních a výstupních periférií pro stavbu jednoduchých IoT zařízení. Pro programování je použito vývojové prostředí ArduinoIde. Tento tutorial provede úplného začátečníka od instalace software až k sestavení termostatu pro spínání silové zátěže.

Annotation

Prerequisite of working with microcontrollers is understanding the basic principles of their communication with the environment and possible ways of their programming. The ESP8266 microcontroller is one of the most popular chips used in Internet of things applications. Wemos D1 mini is one of development platforms behind the ESP8266, which allows direct communication with the PC via a USB port and has plenty of input and output peripherals to build simple IoT devices. The ArduinoIde development environment is used for programming. The thermostatic switch is made as a result of this tutorial.

1. Potřebné vybavení

Pro stavbu jednoduchého IoT termostatu pro spínání silové zátěže je třeba jednak samotný mikrokontroler ESP8266 a dále modul pro měření teploty a spínací relé. Kromě tohoto hardwaru je třeba použít nějaké vývojové prostředí pro tvorbu programu umožňujícího definovat logiku práce mikrokontroleru.

1.1. ESP 8266 a vývojové desky

Mikrokontroler ESP8266 je zařízení obsahující 32 bit procesor, který může být taktován až na frekvenci 160 MHz. Pro komunikaci s okolím využívá WiFi připojení, programování probíhá pomocí sériového rozhraní. Využitelná velikost paměti RAM přímo na chipu je přibližně 50kB, její velikost závisí na velikosti paměti rezervované pro samotnou funkci ESP. Maximální velikost paměti, připojitelné k tomuto modulu pomocí SPI sběrnice, je až 16 MB. Obr. 1 ukazuje fotografie vlastního čipu ESP8266 a ukázkou dostupných provedení ESP modulů, které se liší hlavně rozměry a počtem i sortimentem vyvedených pinů. Napájecí napětí mikrokontroleru je 3,3V. Modul nemá žádnou přepětovou ochranu, takže nelze připojit přímo na pětivoltovou logiku. Jelikož ale velmi velké množství čidel používá právě napájecí napětí

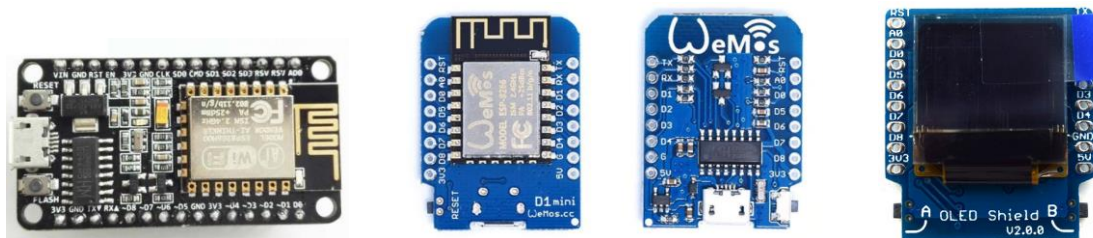
5V, je nutné v praxi řešit tuto nesrovnalost převodníky mezi napětovými hladinami. Příkon celého ESP8266 může být ve špičkách až 300 mA, v režimu hlubokého spánku, kdy běží pouze časovač, který odpočítává čas k probuzení je spotřeba pouze cca 20 μ A, což předurčuje zařízení i k použití při napájení z baterie či akumulátoru.



Obr. 5 Chip ESP8266 a jeho různá provedení pro použití v projektech

1.2. WemosD1 a další vývojové moduly

Na trhu je k dispozici řada vývojových modulů, jejichž srdcem je právě ESP8266. Mezi nejznámější patří WeMos D1 mini a NodeMCU. Oba tyto moduly je možné připojit k PC pomocí USB konektoru, jelikož obsahují převodníky napětí pro napájení ESP a také převodník USB - sériový port. Pro modul WeMos je k dispozici množství modulů s různými periferiemi, od tlačítka, přes senzory pohybu, vlhkosti a teploty až například po OLED display viz. obr.2.



Obr. 2 Vývojové desky využívající ESP8266 – NodeMCU, WeMos D1 mini a modul OLED displaye pro WeMos

Komfort práce s těmito vývojovými deskami je přímo předurčuje pro malé projekty a ladění funkce ESP před návrhem jednoúčelového řešení využívajícího pouze mikrokontroler.

1.3. Vývojové prostředí

Programování mikrokontroleru je možné několika způsoby. Lze použít interpretové jazyky, kdy je přímo v modulu nahrán kompilátor programovacího jazyka (Lua, microPython) a do ESP je v takovém případě nahráván textový zápis programu. Druhou variantou je použití kompilátoru, který vytvoří kontrolerem spustitelný binární kód (ArduinoIDE).

2. Práce s mikrokontrolerem

2.1. Instalace ArduinoIDE

Aktuální verze prostředí je dostupná na stránkách www.arduino.cc pro různé platformy. Jelikož ArduinoIDE je vývojové prostředí primárně určené pro platformu Arduino, je třeba doplnit definici použité vývojové desky (například WeMos) a v případě potřeby i dodatečné knihovny. K tomu slouží správce knihoven, který je jeho součástí vývojového prostředí (Menu: Projekt > Přidat knihovnu > Spravovat Knihovny).

2.2. Programování mikrokontroleru

Programovací jazyk ArduinoIde je obdobou jazyka C. V literatuře [1] existuje velké množství příkladů, na kterých je ukázáno, jakým způsobem lze s mikrokontrolerem a jeho perifériemi komunikovat. Rozhraní umožňuje používat již hotové příklady, které jsou součástí vývojového prostředí (Menu: Soubor > Příklady). Pro seznámení s funkcí knihoven je využití příkladů asi nejlepší cesta, jak na jejich základu rychle vytvořit funkční program dle vlastních představ.

2.3. Vstupy a výstupy

Ke komunikaci s okolím slouží jednotlivé piny vyvedené z modulu ESP tak, aby k nim bylo možné připojit zdroje signálu (v případě vstupů) anebo akční prvky (v případě výstupů). Tyto porty ovládatelné uživatelským programem jsou označovány zkratkou GPIO (General Purpose Input/Output) a je možné definovat jejich funkci. Specifické porty pak umožňují provoz jako sběrnice (např. I2C), nebo rozhraní sériové linky (porty Rx a Tx). Maximální proud přes jeden GPIO je 12 mA.

2.4. WiFi modul

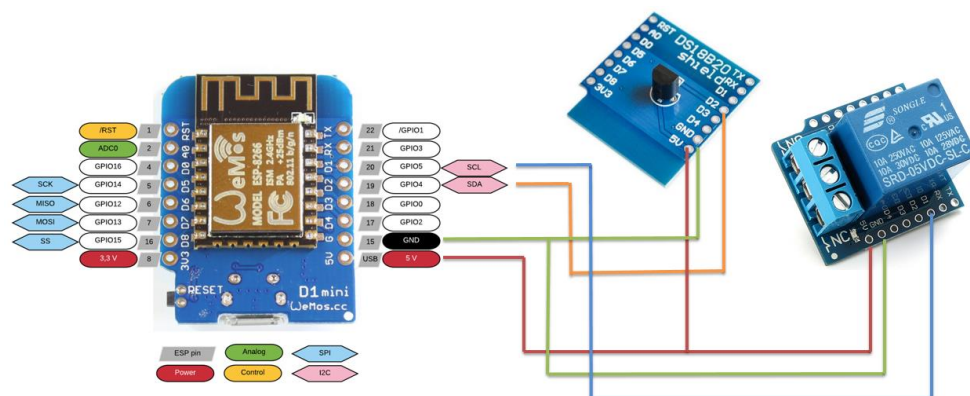
ESP8266 umožňuje prostřednictvím knihovny `ESP8266WiFi.h` práci v následujících konfiguracích: Station (připojování k AP, automatický reconnect), Soft AP (možnost tvorby sítě mezi jednotlivými ESP), Scan (umožňuje získat informace o dostupných sítích), Client (klasický WiFi klient), Client Secure (jako client + šifrování), Server (umožňuje propagovat služby běžící na ESP do internetu).

2.5. Přenos dat do internetu

Po připojení na internet je možné data z mikrokontroleru číst (pokud je konfigurován v režimu Server), anebo data odesílat a ukládat do databází [4]. Za zmínku stojí technologie mqtt, která umožňuje udržovat na jednom místě databázi stavů a hodnot IoT zařízení a v závislosti na nich provádět zpětně řízení IoT prvků. Online vizualizace dat je možná s použitím internetových služeb, jako <https://azure.microsoft.com/> nebo <https://thingspeak.com/>. Pro drobné projekty jsou obě tyto služby zdarma. Pro zasílání zpráv o změně stavu lze využít například službu <https://www.pushbullet.com/>.

3. Termostat pro spínání silové zátěže

Sestavení termostatického spínače pomocí modulu WeMos ukazuje schema na obr. 3. Modul teploměru DS18B20 (komunikující přes sběrnici OneWire) a spínacího relé jsou navrženy tak, aby toto propojení vzniklo pouhým naskládáním modulů na sebe.



Obr. 3 Zapojení modulů wemos jako relé spínané termostatem

Zdrojový kód programu pro ovládání relé tak, aby došlo k jeho sepnutí při překročení nastavené teploty, ukazuje obr. 4.

```
#include <OneWire.h>
#include <DallasTemperature.h>
#define DS18B20 4
#define relayPin 5
OneWire ourWire(DS18B20);
DallasTemperature sensors(&ourWire);
static int rozhodna_teplota = 31;
float teplota;

void setup() {
  Serial.begin(9600);
  Serial.println("\n\Start programu");
  pinMode(D1, OUTPUT);
  delay(1000);
  sensors.begin();
}

void loop() {
  sensors.requestTemperatures();
  teplota = sensors.getTempCByIndex(0);
  Serial.print(teplota);
  Serial.println(" st C");

  if (teplota > rozhodna_teplota){
    digitalWrite(relayPin, HIGH);
  }
  else
  {
    digitalWrite(relayPin, LOW);
  }

  delay(5000);
}
```

Obr. 4 Výpis zdrojového kódu pro relé spínané termostatem

Jak je patrné program se skládá ze tří částí. Jedná se o definice, část setup, která je provedena pouze jednou (při spuštění ESP) a část loop, která je prováděna nepřetržitě stále znovu, dokud nedojde k výpadku napájení nebo restartu zařízení. Uvedený program definuje hraniční teplotu 31°C, za pomoci knihovny DallasTemperature.h je jedenkrát za 5 s přečtena teplota z teplotního čidla a na základě vyhodnocení podmínky je v případě překročení hraniční teploty sepnuto relé.

4. Závěr

ESP8266 je mikrokontroler umožňující levný způsob, jak sledovat a řídit procesy na základě vlastní naprogramované logiky využívající komunikace s okolím. Vzhledem k bohatému spektru připojitelných periférií může najít uplatnění jak v jednoúčelových aplikacích drobné automatizace, tak jako robustní prostředek pro sběr dat a jejich odesílání

k dalšímu zpracování při relativně malé spotřebě. Za zmínku stojí i možnost používat tyto mikrokontrolery k výukovým účelům a studentským projektům, neboť tvorba IoT zařízení vyžaduje jak programátorské a elektrotechnické dovednosti, tak konstruktérské nadání a kreativní myšlení.

LITERATURA

- [20] Osborn, W. P.: **Learn to Program in Arduino C – 18 Lessons, from setup() to robots**, Armadillo Books, 2017, ISBN: 978-0-9981287-1-9
- [21] Javed, A: **Building Arduino Projects for the Internet of Things**, Apress, 2016, ISBN: 978-1-4842-1939-3
- [22] Schwartz, M.: **Internet of Things with ESP8266**, Packt publishing, 2016, ISBN: 978-1-78646-802-4
- [23] Bell, Ch.: **MySQL for the Internet of Things**, Springer, 2016, ISBN: 978-1-4842-1293-6

ONLINE ZDROJE

- [1] <https://arduino.cz/> - České stránky s řadou návodů a praktických ukázek
- [2] <https://arduino-shop.cz/> - Eshop se součástkami pro IoT
- [3] <https://www.arduino.cc/> - Oficiální stránky arduino – stažení ArduinoIde
- [4] https://mz.utia.cas.cz/CAE_Forum_2018/ - Materiály k tutoriálu