



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY**

**A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

**ÚSTAV RADIOELEKTRONIKY**

DEPARTMENT OF RADIO ELECTRONICS

## BEZDRÁTOVÁ STANICE S PŘIPOJENÍM DO SÍTĚ IOT

WIRELESS STATION WITH CONNECTION TO IOT NETWORK

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

Richard Kubíček

**VEDOUCÍ PRÁCE**

SUPERVISOR

doc. Ing. Jaromír Kolouch, CSc.

BRNO 2016

# Bakalářská práce

bakalářský studijní obor **Elektronika a sdělovací technika**

Ústav radioelektroniky

**Student:** Richard Kubiček

**ID:** 164753

**Ročník:** 3

**Akademický rok:** 2015/16

**NÁZEV TÉMATU:**

## Bezdrátová stanice s připojením do sítě IoT

### POKYNY PRO VYPRACOVÁNÍ:

Navrhněte stanici pro připojení senzorů a ovladačů k síti IoT (Internet of Things) pomocí WiFi modulu ESP8266. Vytvořte odpovídající knihovnu pro mikrokontroléry AVR Atmel. Analyzujte nebo navrhněte vhodný protokol pro zabezpečení přenosu dat. Připravte podklady pro realizaci stanice.

Pro demonstraci funkčnosti navrženého řešení realizujte domácí meteostanici pro měření teploty a vlhkosti a stanici s reléovým výstupem pro ovládání síťových spotřebičů jako jsou osvětlení a elektrické topení. Zobrazení naměřených údajů a ovládání spotřebičů bude provedeno pomocí www rozhraní.

### DOPORUČENÁ LITERATURA:

[1] MANN, B. C pro mikrokontroléry. Praha: BEN, 2003.

[2] HÖLLER, J.; TSIATSIS, V. From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence. Oxford: Elsevier, 2014.

**Termín zadání:** 8.2.2016

**Termín odevzdání:** 26.5.2016

**Vedoucí práce:** doc. Ing. Jaromír Kolouch, CSc.

**Konzultant bakalářské práce:** Ing. Ondřej Pavelka, Honeywell

**doc. Ing. Tomáš Kratochvíl, Ph.D., předseda oborové rady**

### UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **ABSTRAKT**

Tato práce se zabývá návrhem WiFi stanice pro připojení senzorů a ovladačů s připojením do sítě IoT. Stanice využívá WiFi modul ESP8266. Jsou zde popsány jednotlivé komponenty, ze kterých se stanice skládá. Práce zahrnuje popis vytvořené knihovny pro mikrokontrolér umožňující ovládání tohoto modulu. Dále jsou součástí práce podklady pro výrobu této stanice.

## **KLÍČOVÁ SLOVA**

WiFi modul, ESP8266, Atmel AVR, DS18B20, DHT11, meteostanice, OLED displej

## **ABSTRACT**

This thesis describes the design of WiFi station for connection sensors and drivers with Internet of Things connection. The station uses WiFi module ESP8266. Description of component which the station consists is also part of the thesis. Description of created library for control module ESP8266 by microcontroller is also part of thesis. Finally, documents for production of this station are also part of the thesis.

## **KEYWORDS**

WiFi module, ESP8266, Atmel AVR, DS18B20, DHT11, meteostation, OLED display

KUBÍČEK, R. *Bezdrátová stanice s připojením do sítě IoT*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2016. 36 s., 6 s. příloh. Vedoucí bakalářské práce doc. Ing. Jaromír Kolouch, CSc..

# PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma Bezdrátová stanice s připojením do sítě IoT jsem vypracoval samostatně pod vedením vedoucího bakalářské práce, a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne .....

.....

(podpis autora)

# PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu doc. Ing. Jaromíru Kolouchovi, CSc. a odbornému konzultantovi panu Ing. Ondřejovi Pavelkovi za odborné vedení, konzultace, cenné rady a návrhy k práci.

V Brně dne .....

.....

(podpis autora)

# OBSAH

|                                             |             |
|---------------------------------------------|-------------|
| <b>Seznam obrázků</b>                       | <b>viii</b> |
| <b>Úvod</b>                                 | <b>1</b>    |
| <b>1 Internet of Things</b>                 | <b>2</b>    |
| 1.1 Internet věcí dnes a v budoucnosti..... | 2           |
| 1.2 Požadavky Internetu věcí.....           | 2           |
| 1.3 Řešení nedostatku IP adres .....        | 2           |
| <b>2 Modul ESP8266</b>                      | <b>4</b>    |
| 2.1 Verze modulů.....                       | 4           |
| 2.2 Parametry.....                          | 6           |
| 2.3 Elektrotechnické připojení.....         | 6           |
| 2.4 Ovládání.....                           | 6           |
| 2.5 Zabezpečení dat.....                    | 6           |
| <b>3 Arduino IDE</b>                        | <b>8</b>    |
| <b>4 NodeMCU</b>                            | <b>10</b>   |
| 4.1 Skripty .....                           | 10          |
| <b>5 AT příkazy</b>                         | <b>11</b>   |
| 5.1 Struktura .....                         | 11          |
| 5.1.1 Složení příkazů.....                  | 11          |
| 5.1.2 Základní pravidla.....                | 11          |
| 5.2 Konfigurace UART .....                  | 12          |
| 5.3 Inicializace modulu .....               | 12          |
| 5.4 WiFi instrukce.....                     | 12          |
| 5.5 Příkazy TCP/IP .....                    | 13          |
| 5.6 Nastavení modulu.....                   | 15          |
| <b>6 WiFI stanice</b>                       | <b>16</b>   |
| 6.1 Mikrokontrolér ATmega32.....            | 16          |
| 6.2 Teplotní senzor DS18B20 .....           | 17          |

|           |                                                      |           |
|-----------|------------------------------------------------------|-----------|
| 6.3       | Čidlo vlhkosti DHT11 .....                           | 18        |
| 6.4       | Bistabilní relé .....                                | 18        |
| 6.5       | Displej.....                                         | 19        |
| <b>7</b>  | <b>Zkušební deska</b>                                | <b>20</b> |
| 7.1       | Hardware .....                                       | 20        |
| 7.2       | Obslužný program .....                               | 20        |
| 7.3       | Obslužný server.....                                 | 21        |
| 7.4       | Výroba prototypu .....                               | 22        |
| 7.5       | Testování prototypu.....                             | 23        |
| <b>8</b>  | <b>Knihovna pro ESP8266</b>                          | <b>24</b> |
| 8.1       | Struktura knihovny .....                             | 24        |
| 8.2       | Hlavní část knihovny.....                            | 25        |
| 8.3       | Uživatelské funkce .....                             | 26        |
| <b>9</b>  | <b>Realizace WiFi stanice</b>                        | <b>28</b> |
| 9.1       | Napájecí zdroj .....                                 | 28        |
| 9.2       | Schéma zapojení stanice .....                        | 28        |
| 9.3       | Layout DPS.....                                      | 29        |
| 9.4       | Realizace meteostanice.....                          | 31        |
| 9.4.1     | Řídicí program meteostanice .....                    | 31        |
| 9.4.2     | Internetový server.....                              | 33        |
| <b>10</b> | <b>Závěr</b>                                         | <b>36</b> |
|           | <b>Literatura</b>                                    | <b>37</b> |
|           | <b>Seznam symbolů, veličin, zkratk a cizích slov</b> | <b>38</b> |
| <b>A</b>  | <b>Modul DC-DC měniče TPS62160</b>                   | <b>40</b> |
| <b>B</b>  | <b>Realizace výsledné stanice</b>                    | <b>41</b> |

# SEZNAM OBRÁZKŮ

|             |                                                              |    |
|-------------|--------------------------------------------------------------|----|
| Obrázek 2-1 | ESP8266 ve verzi ESP-01 [14].....                            | 5  |
| Obrázek 2-2 | ESP8266 ve verzi ESP-07 [15].....                            | 5  |
| Obrázek 3-1 | Vývojové prostředí Arduino IDE.....                          | 8  |
| Obrázek 6-1 | Procesor ATmega32 v pouzdře TQFP44 [13].....                 | 16 |
| Obrázek 6-2 | Vývojový diagram čtení teploty z DS18B20.....                | 17 |
| Obrázek 6-3 | Kombinované čidlo DHT11.....                                 | 18 |
| Obrázek 6-4 | Časové impulsy na bistabilním relé, převzato z [5].....      | 19 |
| Obrázek 7-1 | Vývojový diagram obslužného programu prototypu .....         | 21 |
| Obrázek 7-2 | Výsledná podoba prototypu.....                               | 23 |
| Obrázek 8-1 | Struktura knihovny.....                                      | 24 |
| Obrázek 9-1 | Vizualizace osazeného plošného spoje.....                    | 30 |
| Obrázek 9-2 | Vizualizace plošného spoje stanice ze strany bottom .....    | 30 |
| Obrázek 9-3 | Displej se zobrazenými údaji.....                            | 31 |
| Obrázek 9-4 | Osazený plošný spoj s připojeným displejem a teploměrem..... | 31 |
| Obrázek 9-5 | Vývojový diagram finálního zařízení.....                     | 32 |
| Obrázek 9-6 | Zobrazená webová stránka .....                               | 33 |
| Obrázek 9-7 | Internetová stránka s grafy naměřených hodnot.....           | 34 |



# ÚVOD

Cílem této práce bylo navrhnout podklady pro realizaci WiFi stanice pro připojení senzorů a ovladačů s připojením do sítě IoT. Součástí návrhu stanice bylo vytvořit odpovídající knihovnu umožňující ovládání WiFi modulu ESP8266. Pro demonstraci funkčnosti stanice a knihovny byla vytvořena domácí meteostanice umožňující měření teploty, vlhkosti a spínání relé pro uživatelská zařízení.

Aby uživatel nebyl závislý pouze na výsledcích měření, které, jak bude později uvedeno, se ukládají na internetovém serveru, je součástí zařízení také displej zobrazující aktuální naměřené údaje.

V následujících kapitolách jsou postupně uvedeny detaily o síti IoT, o samotném modulu ESP8266 a podklady pro realizaci stanice.

# 1 INTERNET OF THINGS

Internet věcí (IoT) je označení pro přístroje a zařízení, které jsou nějakým způsobem připojeny k internetu.

## 1.1 Internet věcí dnes a v budoucnosti

V současné době připojení zařízení k internetu umožňuje kromě čtení údajů ze zařízení i jejich ovládání. Tato zařízení se postupně stávají běžnější součástí našich životů. Jejich úkolem je ušetřit jejich majitelům čas při běžných činnostech.

Pravděpodobně největším tahákem této kategorie jsou takzvané inteligentní domy. Jejich hlavní předností je možnost řízení jakýchkoliv parametrů na dálku. Jinými slovy je možné před odchodem z práce si v domácnosti zapnout vytápění. Obdobně lze řídit světelné okruhy, vypínat a zapínat zásuvky a podobně vše ostatní co lze v domech najít.

## 1.2 Požadavky Internetu věcí

Hlavním a tím nejdůležitějším požadavkem pro funkčnost IoT je připojení k internetu. Bohužel je to prozatím také největší slabina, obzvláště v České republice. Stále se totiž najdou oblasti na území České republiky, kde úroveň připojení k internetu není příliš vysoká.

Tato slabina by mohla být dle [1] odstraněna spoluprací firem T-Mobile a SIGFOX, které by mohly pokrýt Českou republiku sítí 868 MHz. Ušetřila by se tak především energie spojená s přístupem k bezdrátovému internetu, tak jak jej známe dnes. Zařízení pro IoT by tak s napájením pomocí dvou tužkových baterií mělo vydržet díky nižšímu odběru v chodu více jak 10 let.

Dalším důležitým problémem této sítě jsou IP adresy. Pokud by každé jednotlivé zařízení mělo přiřazenu svoji IP adresu, došlo by k jejich rychlému vyčerpání.

## 1.3 Řešení nedostatku IP adres

V případě domácí sítě má každé zařízení svoji IP adresu a uživatel je nucen se ke každému z nich připojovat jednotlivě. Z tohoto důvodu je uživatelsky přívětivější a prakticky realizovatelnější využití tzv. centrální stanice.

Centrální stanice je jediné zařízení, mající svoji IP adresu. Veškerá zařízení jsou nějakým způsobem připojena k této centrální stanici. Tato připojení lze realizovat jak pomocí kabelu, po sběrnici, tak i bezdrátově. Každé z těchto řešení má svoje pro a proti.

Jedním z řešení připojení je použití kabelu a komunikačních protokolů jako jsou RS232, RS485, nebo sběrnice SPI, I2C, atd. Toto řešení je spolehlivé, nedochází ke ztrátě dat. Je navíc také možné ve svazku s těmito kabely vést i napájení cílového zařízení. Bohužel použití tohoto řešení sebou nese značnou nevýhodu, tedy nutnost

rozvádět po místnostech kabeláž. Množství mědi v kabelech tak stoupá s počtem použitých senzorů, či počtem chytrých zařízení.

V případě propojení pomocí bezdrátových technologií ušetříme na kabeláži. Je zde ovšem problém s napájením bezdrátových senzorů, protože spotřeba bezdrátových technologií je vysoká. Propojení s centrální stanicí je možné pomocí technologií, jako Bluetooth, WiFi, tedy 2,4 GHz síť, atd. Všechny tyto technologie pracují ve stejném kmitočtovém pásmu, což znamená, že může dojít k zahlcení pásma.

## 2 MODUL ESP8266

ESP8266 je pravděpodobně nejdostupnější a s cenou pohybující se okolo 100 korun za kus nejlevnější WiFi modul. Pochází z dílny firmy Espressif Systems z Šanghaje. Čipy ESP8266 byly vyvinuty pravděpodobně v letech 2013 nebo 2014. Přesný rok vývoje nebyl nalezen.

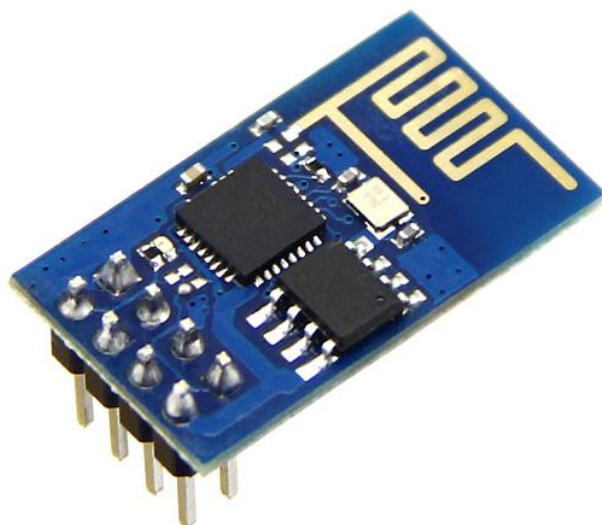
V současnosti je ke koupi několik verzí modulů ESP8266. Je samozřejmě také možné zakoupit i samotný čip. Ovšem zde vyvstává požadavek na korektně navrženou desku plošných spojů pro tento čip. Již jen díky tomu, že se jedná o bezdrátovou technologii, je nutné mimo jiné správně přizpůsobit anténu a samotný čip.

### 2.1 Verze modulů

Firma Espressif rozlišuje moduly jako ESP-01, ESP-07, ESP-12, atd. Verze ESP-01 je znázorněna na obrázku 2-1 a verze ESP-07 na obrázku 2-2. Každý z nich je něčím specifický a má výhody i nevýhody. Jako nejzákladnější verzi modulu lze pokládat ESP-05. Tato verze nemá vyvedeny použitelné GPIO piny, avšak je zde k dispozici U-FL konektor pro připojení externí 2,4 GHz antény.

Jiné verze mají k dispozici 2 a více GPIO pinů. Existují verze modulů, které potřebují připojit externí anténu, vyleptanou na stejném plošném spoji, ke kterému je připájen dotyčný modul. Lze také nalézt verze s anténou, vyleptanými na plošném spoji modulu, případně keramické antény nebo alespoň U-FL konektor.

Každá z verzí má většinou jinou obměnu FLASH paměti. Dříve vyvinuté moduly mají menší paměť, zatímco novější ji mají větší. Větší paměť má výhodu, že lze vytvořit vlastní program ve vývojovém prostředí Arduino IDE a ten poté do modulu nahrát.



Obrázek 2-1 ESP8266 ve verzi ESP-01 [14]



Obrázek 2-2 ESP8266 ve verzi ESP-07 [15]

## 2.2 Parametry

Protože ESP8266 pochází z Číny, je poněkud obtížnější nalézt oficiální technickou dokumentaci přeloženou alespoň do angličtiny. Nicméně byl nalezen technický list pod hlavičkou výrobce avšak s vodoznakem confidential - důvěrné.

Jak již bylo řečeno, komunikace probíhá v bez licenčním pásmu ISM 2,4 GHz. Modul podporuje protokol 802.11 ve standardech b, g i n. Je zde podpora jak P2P sítě, tak i vytvoření přístupového bodu. Čip má již implantován TCP/IP stack umožňující připojení do WiFi sítě a obdržení IP adresy. Lze také nalézt v čipu i integrovaný teplotní senzor. Kromě podpory UARTu je zde i hardwarová podpora rozhraní SPI.

Čip je navržen tak aby umožňoval vysokou úroveň integrace, takže k provozu potřebuje pouze externí rezistory, kondenzátory a krystal. Byl vyroben pro použití v mobilní nositelné elektronice.

Co se týče elektronických parametrů, tak při napájení 3,3 V při přenosu informací rychlostí 1Mbps a výstupním výkonem 19,5 dBm by měl mít modul odběr 215 mA. Tato hodnota by podle technické dokumentace měla být maximální.

## 2.3 Elektrotechnické připojení

Pro správnou funkci modulu je nutné kromě připojení napájení připojit určité vývody buďto na napájení nebo na GND.

Na napájení je nutné připojit vývod, popsáný jako EN nebo CH\_PD. Bez tohoto propojení není modul funkční. Také je vhodné připojit resetovací vývod přes rezistor o hodnotě 10 kΩ na napájecí napětí. Toto zajistí, aby se program v procesoru samovolně nerestartoval. K restartu programu tak dojde pouze, pokud je aktivován softwarový watchdog. V případě že tedy dojde k chybě a program se zacyklí, dojde k restartu modulu.

Některé verze firmware, se kterými je modul dodáván vyžadují dále i uzemnění vývodu GPIO15.

## 2.4 Ovládání

Pro řízení samotného modulu se používá standardně AT příkazů. Tyto příkazy se zasílají do modulu rozhraním UART. Tyto příkazy budou později podrobně popsány.

Další možností jak řídit modul je napsání vlastního řídicího programu. To je možné provést za pomoci vývojového prostředí Arduino IDE, nebo díky platformě NodeMCU.

## 2.5 Zabezpečení dat

V dnešním moderním světě je zabezpečení dat, a především citlivých dat velice důležité. Proto je nutné zabezpečit data i v tomto případě.

Sám modul ESP8266 svým vlastním zabezpečovacím protokolem nedisponuje,

nebo to alespoň nebylo zjištěno. Z tohoto důvodu je nutné data zabezpečit již při jejich odesílání do modulu a dešifrovat je na straně přijímače (respektive serveru).

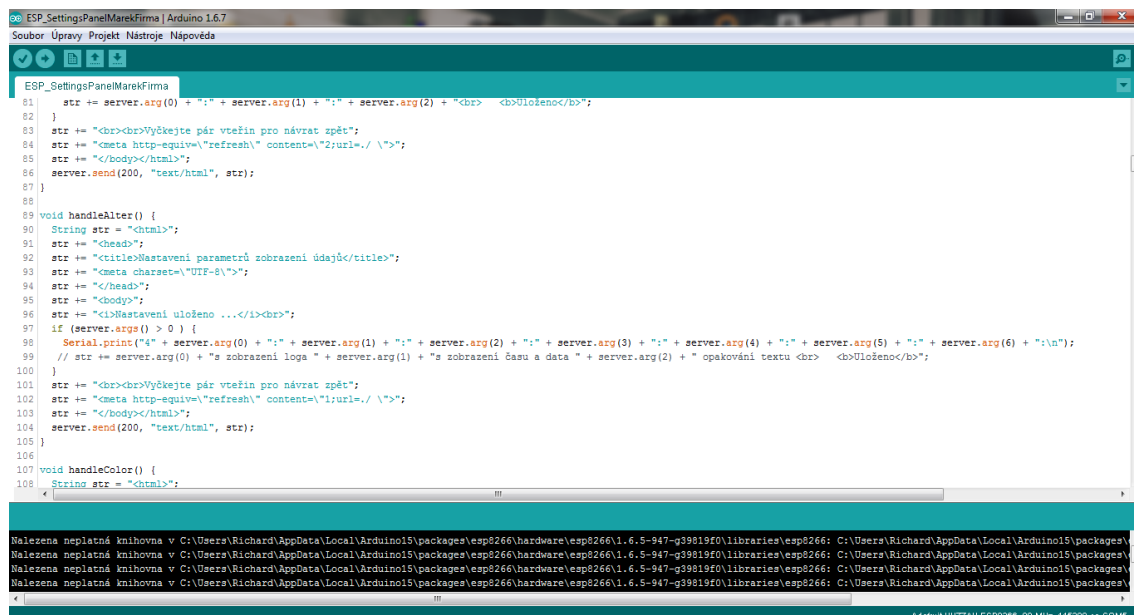
Samotná WiFi komunikace je zabezpečena již ze samotného principu WiFi sítě, takže tuto část komunikace není nutné řešit.

Pravděpodobně nejvhodnější způsob jak zabezpečit data odesílaná modulem na server nebo na jiný přijímač je úprava dat. Takovou úpravu lze provést například použitím operace XOR, tedy exkluzivní disjunkce. Při tomto způsobu šifrování určíme klíčové slovo, na které společně s daty použijeme tuto operaci. Na straně přijímače (serveru) již stačí použít stejné klíčové slovo s přijmutými daty a aplikovat na ně opět operaci XOR.

I takové na pohled jednoduché šifrování je dostatečné k tomu, aby pokud dojde k odposlechnutí dat, byly tato data k ničemu.

### 3 ARDUINO IDE

Arduino IDE, na obrázku 3-1, je vývojové prostředí primárně určené pro vytvoření programů a aplikací na platformě Arduino, respektive Genuino.



Obrázek 3-1 Vývojové prostředí Arduino IDE

Programování v tomto prostředí je uzpůsobeno pro začátečníky z řad programátorů. Je zde použita mírně upravená forma jazyka C++, jazyk wire. Tato úprava umožňuje psát vlastní zdrojové kódy, které jsou velice srozumitelné i úplným začátečníkům. Pro tento způsob programování není nutná znalost registrů daného mikrokontroléru, protože jeho veškeré ovládání probíhá přes integrované funkce.

Pro zahájení programování modulu ESP8266 v tomto prostředí je nutné nainstalovat definici tohoto modulu. Společně s touto definicí dojde i k instalaci knihoven, nutných pro funkci WiFi části procesoru. Stažení této definice je možné z [2]. Je zde k dispozici i detailní popis instalace. Aby bylo možné tyto knihovny nainstalovat, je nutné mít již nainstalované Arduino IDE od verze 1.6.4, jehož součástí je správce desek. Po této instalaci je možné započít vývoj vlastního programu.

Do výsledného programu lze také přidat téměř jakoukoliv knihovnu, kompatibilní s Arduinem. Jediné omezení je ve velikosti programu, který se vleze do čipu. Lze využívat periférie jako je UART, SPI a I2C. Tyto periférie mohou být použité jak na pinech, k nim náležící, nebo i na ostatních pinech, kde bude tato periférie virtualizována. Protože Arduino IDE obsahuje také svůj vlastní kompilátor, není nutné žádný jiný instalovat. Nahrání programu do cílového modulu je provedeno za pomoci sériového rozhraní. To je však možné pouze pokud je čip v programovacím módu. Do tohoto módu je vstoupeno připojením pinu GPIO0 na GND před připojením napájení.

Jakmile je program úspěšně nahrán, dojde po restartu modulu k jeho startu.



Pro ukázkou může být uveden následující úsek programu, vytvářející WiFi přístupový bod a vypisující IP adresu modulu přes sériovou linku.

```
void initWiFi(void) {  
  
    WiFi.softAP("AP_SSID","AP_PSWD"); //vytvoreni AP s danymi parametry  
    IPAddress myIP = WiFi.softAPIP(); //promenne myIP je predana IP  
                                   adresa  
    Serial.print("AP IP address: "); // vypis textu na UART  
    Serial.println(myIP);           // vypis IP adresy modulu na UART  
  
}
```

Výhodou tohoto způsobu řízení je možnost provozovat zařízení pouze s modulem. Není zde potřeba řídicí procesor. Příkladem takového zařízení může být například bateriový senzor teploty, obsahující kromě modulu pouze čidlo teploty. Dochází tak k minimalizaci ceny výsledného zařízení.

Přes veškeré výhody tohoto využití modulu je zde také jedna obrovská nevýhoda. Při snaze zobrazit uživateli velké množství údajů v podobě webové stránky se může stát, že při odesílání stránky do prohlížeče dojde k zaplnění paměti RAM modulu a k aktivaci vnitřního watchdog časovače, jehož přetečení způsobí restart modulu. Tato nevýhoda byla pozorována při snaze využít tento modul pro nastavování parametrů textového informačního panelu, kdy již procesoru nezbýval čas na vykreslování údajů na displej a na obsluhu modulu pomocí AT příkazů. Tyto pády modulu byly vyřešeny tak, že se z původně jediné stránky pro nastavování asi 15 parametrů, musela stát komplexnější stránka s více podstránkami. Na druhou stranu bylo možné přenést na modul i měření teploty, kdy za pomoci vhodné knihovny mohlo být uskutečněno měření teploty ze senzoru DS18B20 a její odesílání po sériovém rozhraní do řídicího mikrokontroléru.

## 4 NODEMCU

NodeMCU je operační systém, běžící na modulu ESP8266. Jeho ovládání je provedeno přes tzv. Lua příkazy. Způsob ovládání systému vzdáleně připomíná příkazovou řádku, známou z Microsoft Windows.

Tento systém není běžnou součástí modulů. Je zapotřebí jej manuálně nahrát do modulu. To je možno provést pomocí aplikace ESP8266 Flasher, kterou je možno nalézt na [3].

Jakmile je systém úspěšně nahrán, je možné se k modulu připojit přes sériové rozhraní a začít jej používat. Lua příkazy mají podobnou strukturu, jako programy napsané v jazyce Bascom. Použití tohoto systému je o něco složitější, co se týče znalostí programátora, než při použití předchozího Arduino IDE, avšak pokud se uživatel nebrání změnám, neměl by být pro něj tento systém problém. Opět se zde přistupuje k jednotlivým perifériím objektově (podobně je tomu Arduino IDE). Příkladem použití tohoto systému může být následující ukázka převzatá z [4]. Tato ukázka slouží pro vytvoření přístupového bodu a vypsání základních údajů přes sériovou linku.

```
-- Global Variables (Modify for your network)
ssid = "my_ssid"
pass = "my_pass"

-- Configure Wireless Internet
print('\nAll About Circuits init.lua\n')
wifi.setmode(wifi.STATION)
print('set mode=STATION (mode='..wifi.getmode()..')\n')
print('MAC Address: ',wifi.sta.getmac())
print('Chip ID: ',node.chipid())
print('Heap Size: ',node.heap(),'\n')

-- wifi config
start wifi.sta.config(ssid,pass)
-- wifi config end
```

Používání tohoto systému má tu nevýhodu, že je nutné mít k modulu pro automatickou funkci připojení řídicí procesor. Je tak navýšena cena výsledného zařízení.

### 4.1 Skripty

NodeMCU nabízí také možnost napsat funkční Lua skript a ten poté nahrát do modulu, podobně jako je tomu v případě programování přes Arduino IDE. Tento skript bude poté načten a aktivován. Na [3] lze nalézt několik funkčních skriptů k různým perifériím, které lze připojit k modulu. Mezi těmito předpřipravenými perifériemi je čidlo teploty DS18B20, OLED displej s knihovnou u8glib.

Použitím těchto skriptů odpadá nutnost připojení řídicího procesoru.

## 5 AT PŘÍKAZY

Každá verze firmware, umožňující ovládání modulu pomocí AT příkazů má svůj vlastní seznam příkazů. Většina z nich, alespoň ty základní a nejdůležitější jsou standardizovány. Tento standard je důležitý proto, aby nedocházelo k problémům v případě sériové výrobě zařízení, a nebylo tak nutné pokaždé měnit ovládací program. Nicméně se může stát, že některé z níže uvedených příkazů nebudou k dispozici.

Stává se, že některý příkaz je podporován jen daným firmware. Většinou však tyto příkazy, které nejsou podporovány, jsou spíše podružné, a nejsou bezpodmínečně nutné k chodu programu a celého zařízení.

### 5.1 Struktura

Příkaz, který je posílán do modulu přes rozhraní UART, a to buď z počítače, nebo z mikrokontroléru je složen ze dvou částí. První částí je uvození AT příkazu, za kterým následuje samotný příkaz.

#### 5.1.1 Složení příkazů

Příkaz může mít až 4 typy zadání, tedy až 4 typy odpovědí. První z nich je takzvaný test. V tomto případě odpovědí na příkaz složený podle následujícího je seznam možných a použitelných parametrů tohoto příkazu.

```
AT+<x>=?
```

Další možností je příkaz typu dotaz, jehož odpovědí je aktuálně nastavený parametr.

```
AT+<x>?
```

Třetí možností je příkaz typu nastavit, nastavující předávaný parametr. Odpovědí u většiny příkazů, který tento typ podporují, je OK.

```
AT+<x>=<. . .>
```

Posledním druhem příkazů je spouštěcí příkaz. Ten spustí danou požadovanou akci.

```
AT+<x>
```

V těchto příkazech představuje <x> název daného příkazu a <. . .> představuje uživatelem definovaný parametr.

#### 5.1.2 Základní pravidla

Při zadávání příkazů je nutné dodržet několik pravidel. Jedním z nich je citlivost na velikost písmen v příkazu. Jedná se o zadání příkazu. To musí být provedeno s použitím velkých písmen. V opačném případě dojde k chybě příkazu, protože firmware jej nerozezná.

Na konci každého příkazu musí být jeho ukončení. To se provádí použitím ukončovacích značek *CR* a *LF*. Tyto značky jsou provedeny posláním znaků `\r\n` na

konci příkazu. Takto je zaručeno odeslání řádku s příkazem do modulu.

V případě, že je součástí příkazu textový řetězec, je nutné, aby byl uzavřen do uvozovek. Příkladem takového příkazu může být vytvoření AP.

## 5.2 Konfigurace UART

Pro umožnění komunikace pomocí AT příkazů je nutné, aby byla sériová linka správně nakonfigurována. Přenosová rychlost linky je pro většinu modulů nastavena na 115200 baud. Některé z firmware umožňují toto nastavení změnit. Počet bitů v každém znaku je nastaven na 8 bitů (data bits) a je nastaven 1 stop bit (počet bitů ukončujících znak).

## 5.3 Inicializace modulu

Jedná se o příkazy, u nichž není povinnost je použít, nicméně jsou vhodné, protože jejich odpovědi jsou užitečné informace o hardware a software modulu.

Základní instrukcí je příkaz **AT**. Odpovědí na tento příkaz je OK. Lze tak zjistit, zda je správně nastavena sériová linka a modul je správně zapojen.

Instrukce restartu **AT+RST** provede restartování modulu. Tento příkaz se také provede, pokud je aktivován watchdog. Odpovědí na tento příkaz je ready. Tato odpověď značí, že je modul připraven k použití. Součástí odpovědi bývá i informace o vývojáři firmware, datu jeho publikování a informace o hardware.

Informaci o verzi daného firmware v podobě 8 bytů dlouhého čísla získáme příkazem **AT+GMR**.

Odeslání jakéhokoliv příkazu, i když chybného způsobí jeho předání zpět. Jinak řečeno, kromě odpovědi na příkaz uživatel případně mikrokontrolér přijme i samotný příkaz. Tuto odpověď lze vypnout příkazem **ATE0**.

## 5.4 WiFi instrukce

Jedná se o sadu příkazů, nastavujících parametry WiFi připojení. Některé z těchto parametrů jsou podmíněny určitým předchozím nastavením. V případě že se nejedná o speciální příkaz, a je možné jej provést, odpovědí na tento příkaz je OK.

Režim, ve kterém modul pracuje je nastaven příkazem **AT+CWMODE=<x>**. Kde x je číslo zvoleného režimu. Režim 1 je použit pro práci v režimu klienta, kdy se modul může připojit k WiFi síti a k internetovému serveru. Režim 2 nastaví modul pro práci v AP módu. V tomto případě dojde k vytvoření přístupového bodu s nastavenými parametry. Poslední režim, číslo 3, značí použití obou módů současně.

Pro připojení k existující WiFi síti je zapotřebí aby CWMODE příkaz byl nastaven na 1 nebo 3. Poté již může být zadán příkaz **AT+CWJAP=<SSID>,<PSWD>** který připojí modul k WiFi síti a obdrží IP adresu. SSID symbolizuje název WiFi sítě a PSWD její heslo. Je-li zapotřebí zjistit zda je modul k nějaké síti připojen lze použít obměnu tohoto příkazu **AT+CWJAP?** Odpovědí na tuto obměnu je v případě aktivního

připojení název sítě. V opačném případě ERROR.

V případě, že je zapotřebí zjistit zde je daná WiFi síť k dispozici lze použít příkaz **AT+CWLAP=<SSID>,<MAC>,<CH>** který odpoví v případě dostupnosti sítě OK. V opačném případě odpoví ERROR. Je zde také možnost vypsát všechny sítě, které jsou k dispozici. To je provedeno příkazem **AT+CWLAP**. Odpovědí na tento příkaz je seznam sítí v následujícím formátu: **+CWLAP:<ECN>,<SSID>,<RSSI>,<MAC>**. Symbol ECN představuje způsob zabezpečení sítě (0 - bez hesla, 1 – WEP, 2 - WPA PSK, 3 – WPA2 PSK, 4 – WPA/WPA2 PSK), SSID název sítě, RSSI sílu signálu a MAC představuje mac adresu přípojného bodu.

V případě nutnosti odpojení se od WiFi sítě lze použít **AT+CWQAP**.

Pro vytvoření přístupového bodu je nutné aby CWMODE byl nastaven na 2 nebo 3. Poté je možné použít následující příkaz: **AT+CWSAP=<SSID>,<PSWD>,<CH>,<ECN>** Symbol ECN představuje způsob zabezpečení sítě podobně jako je tomu u příkazu CWLAP, SSID název sítě, PSWD heslo sítě a CH představuje kanál, na kterém bude síť vysílána. Zde je vhodné nastavit odlišný kanál, než na kterém vysílají sítě v okolí modulu, aby nedocházelo ke vzájemnému ovlivňování.

Pro zjištění IP adres aktuálně připojených zařízení k vytvořenému přístupovému bodu je možné použít instrukci **AT+CWLIF**.

Vytvořený přístupový bod má aktivní DHCP přiřazení IP adres. To je možno vypnout, stejně jako je možno vypnout získávání IP adresy při módu klienta. Instrukce má následující podobu: **AT+CWDHCP=<MODE>,<EN>** Symbol MODE představuje jednotlivé pracovní módy dle definice u příkazu CWMODE, EN představuje povolení DHCP pro EN = 0 a zakázání DHCP pro EN = 1.

Modul může mít 2 rozdílné mac adresy. Jedna je pro mód stanice a druhá je pro mód přístupového bodu. Pro zjištění adresy módu klienta je použit příkaz **AT+CIPSTAMAC?** Ten odpoví mac adresou. Pokud je zapotřebí tuto adresu změnit, je použito **AT+CIPSTAMAC=<MAC>** kde MAC představuje zamýšlenou adresu, která se má nastavit. Podobně lze nastavit či zjistit mac adresu pro mód přístupového bodu. Tyto příkazy jsou **AT+CIPAPMAC?** a **AT+CIPAPMAC=<MAC>**.

Pokud je zapotřebí změnit IP adresu v módu klienta lze tuto změnu provést příkazem **AT+CIPSTA=<IP>** případně zjistit současnou IP adresu lze příkazem **AT+CIPSTA?** Změna IP adresu může být provedena, pouze pokud to umožňuje přípojný bod. Symbol IP zde představuje zamýšlenou IP adresu.

Podobně lze změnit IP adresu modulu v případě módu přístupového bodu. Tato možnost je důležitá v případě, že je zapotřebí definovat adresu, přes kterou se přistupuje k modulu. Tato instrukce má strukturu **AT+CIPAP=<IP>** případně lze zjistit aktuální IP adresu **AT+CIPAP?** Podobně jako při nastavení IP adresy módu klienta, symbol IP značí zamýšlenou IP adresu.

## 5.5 Příkazy TCP/IP

Jedná se o sadu příkazů, umožňující nastavení parametrů TCP/IP stacku, odesílání dat, apod.

Za předpokladu že je modul připojen k síti, případě je aktivní mód přístupového bodu, lze získat informace o připojení pomocí **AT+CIPSTATUS** Odpovědí na tento příkaz je STATUS:<STAT> a +CIPSTATUS:<ID>,<TYPE>,<ADDR>,<PORT>,<MODE> Symbol STAT představuje aktuální stav připojení (2 – získaná IP adresa, 3 – modul je připojen a 4 – modul je odpojen), ID představuje číslo spojení za předpokladu více možných aktivních připojení, TYPE udává typ připojení, na výběr je mezi TCP a UDP připojením, ADDR je IP adresa ke které je modul připojen, PORT značí číslo portu na kterém je adresa připojena a TYPE udává v jakém módu modul pracuje (0 – mód klienta a 1 – mód přístupového bodu).

V případě že je zapotřebí umožnit více spojení, což je nutné minimálně, pokud je nastaven mód klienta a přístupového bodu současně, je nutné provést příkaz, který umožní více aktivních spojení **AT+CIPMUX=<MODE>** Pro umožnění jediného spojení je nastaven MODE na 0, zatímco pro více aktivních spojení je MODE nastaven na 1. Počet spojení je omezen na 4.

V případě že je nastavena práce v módu klienta je možné připojit se k serveru, ať již k místnímu nebo ke vzdálenému. Pokud je povoleno pouze jedno spojení je struktura příkazu pro připojení následující **AT+CIPSTART=<TYPE>,<ADDR>,<PORT>** Při povolení většího počtu připojení je do příkazu vloženo navíc číslo spojení které se má navázat **AT+CIPSTART=<ID>,<TYPE>,<ADDR>,<PORT>** Symbol TYPE udává typ spojení (TCP nebo UDP), ADDR je IP adresa cíle, PORT je číslo portu, přes který na vzdálený cíl je přistupováno. ID je číslo používaného spojení.

Za předpokladu že je spojení navázáno lze odeslat data. Je nutné znát předem jejich délku. Pro jediné spojení je použito **AT+CIPSEND=<LENGTH>** a pro vícenásobné spojení je použito **AT+CIPSEND=<ID>,<LENGTH>** Pokud je umožněno vysílání, odpovědi na tento příkaz je znak ≥. Symbol ID značí číslo spojení a LENGTH délku vysílané zprávy. Pokud je přijat znak > je možné poslat zprávu o již předané délce.

Pro ukončení aktivního spojení lze použít příkaz **AT+CIPCLOSE** případně **AT+CIPCLOSE=<ID>** ID představuje číslo uvažovaného spojení. Nicméně k ukončení spojení dojde i samovolně po uplynutí časového limitu, který má server nastaven.

Pro zjištění aktuálně používaných IP adres v jednotlivých komunikačních módech a jejich mac adres lze použít instrukci **AT+CIFSR**.

V případě že je povoleno více aktivních spojení, lze vytvořit TCP server, ke kterému se lze připojit. Server lze vytvořit pomocí **AT+CIPSERVER=<MODE>,<PORT>** kde MODE znamená pro 1 vytvoření a pro 0 vymazání serveru a PORT je číslo portu na kterém server bude aktivní.

Pokud je řeč o serveru, je vhodné nastavit tzv. timeout, čili maximální čas během kterého může vzdálený klient dostat odpověď. Tuto dobu je možno nastavit pomocí **AT+CIPSTO=<TIME>** kde TIME je požadovaný čas v sekundách (lze nastavit 0 – 7200 vteřin).

Modul může také data přijímat. Pokud se tak stane, jsou data odeslána na sériový port společně s jejich délkou v podobě: +IPD,<LEN>:<DATA> a pro případ že je povoleno více spojení, tak +IPD,<ID>,<LEN>:<DATA> Symbol ID značí identifikátor spojení, LEN délku přijímaných dat a DATA obsahuje samotná přijímaná data.

## 5.6 Nastavení modulu

Další příkazy, které mohou být užitečné pro práci s modulem, slouží například pro nastavení přenosové rychlosti po sériové lince nebo pro aktualizaci firmware.

Pro aktualizaci aktuálního firmware, je-li aktualizace k dispozici a je-li modul připojen k síti internet, lze použít **AT+CIUPDATE** Tento příkaz poté pokud je to možné odpoví jednotlivými kroky aktualizací.

Ke změně přenosové rychlosti slouží příkaz **AT+CIOBAUD=<SPEED>** přičemž **SPEED** je požadovaná rychlost.

## 6 WIFI STANICE

Výsledná stanice, jejíž návrh je cílem této práce obsahuje kromě modulu ESP8266 přijímacího AT příkazy také řídicí mikrokontrolér AVR ATmega32. Mikrokontrolér kromě řízení WiFi modulu také čte aktuální teplotu ze senzoru DS18B20, provádí čtení vlhkosti z čidla DHT11, čte údaje z dalších připojených senzorů a řídí spínání relé. Tyto komponenty byly vybrány s ohledem na cíl práce.

Mikrokontrolér také umožňuje naměřené údaje, kromě jejich poslání pomocí AT příkazů na k tomu určený internetový server, zobrazit na displeji.

### 6.1 Mikrokontrolér ATmega32

Jedná se o 8bitový mikrokontrolér AVR od firmy Atmel. Tento procesor je na obrázku 6-1. Kromě rozhraní UART, se používá pro komunikaci s modulem ESP8266, disponuje mikrokontrolér také sériovými sběrnici SPI, I2C, atd. Procesor také obsahuje dva 8bitové a jeden 16bitový časovač, které lze využít pro časování aplikací. Pro měření externího analogového napětí je k dispozici osmi kanálový 10bitový AD převodník. K dispozici je mnoho dalších periférií jako PWM, časovač Watchdog atd.



Obrázek 6-1 Procesor ATmega32 v pouzdře TQFP44 [13]

Napsat řídicí program pro mikrokontrolér je možné prakticky ve všech programovacích jazycích. Poté záleží pouze na použitém překladači, který přeloží program do šestnáctkové soustavy formátu hex. Ten je poté možno nahrát například pomocí rozhraní ISP (In System Programming – programování bez vyjmutí procesoru z cílové desky) do procesoru.

Procesor může pracovat jak s externím krystalem o taktu maximálně 16 MHz, nebo i bez krystalu.

Program pro realizaci WiFi stanice s modulem ESP8266 a tímto procesorem je napsán ve vývojovém prostředí firmy Atmel, Atmel Studio.



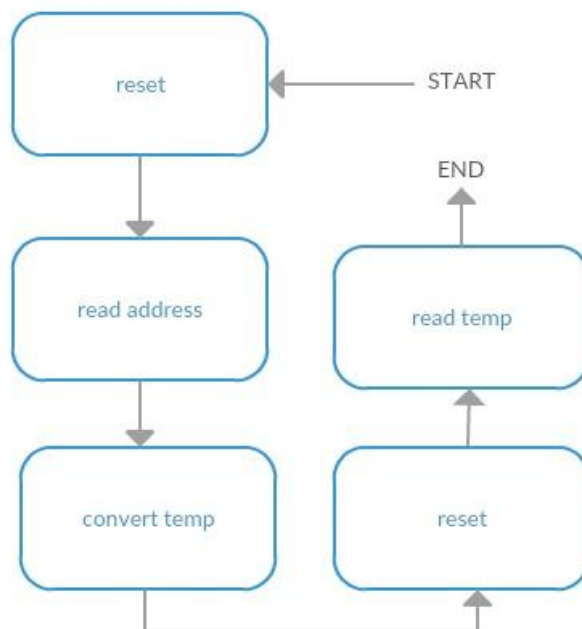
## 6.2 Teplotní senzor DS18B20

Čidlo DS18B20 je digitální, až 12bitový senzor teploty od firmy Maxim Integrated. Tento senzor lze zakoupit v nejpoužívanějším pouzdře TO-92 (používáno pro čidla na kabelu), nebo v pouzdře SO-8. Rozsah měřených teplot je od  $-55^{\circ}\text{C}$  do  $125^{\circ}\text{C}$ . S procesorem komunikuje pomocí sběrnice 1wire. Jedná se o jedno vodičové propojení, kdy jsou logické hodnoty vysílány po sběrnici tak, že po inicializaci odesílání příkazu dojde po určité době k odeslání dané logické hodnoty. Následující ukázka, znázorňuje odeslání bitu po sběrnici, jehož hodnota je parametrem funkce.

```
void therm_write_bit(uint8_t bit){  
    //Pull line low for 1uS  
    THERM_LOW();  
    THERM_OUTPUT_MODE();  
    _delay_us(5);  
    //If we want to write 1, release the line (if not will keep low)  
    if(bit) THERM_INPUT_MODE();  
    //Wait for 60uS and release the line  
    _delay_us(55);  
    THERM_INPUT_MODE();  
}
```

Každý kus tohoto čidla má ve své integrované ROM paměti uložen unikátní 64bitový identifikátor. Lze tak zapojit paralelně na jednu sběrnici více čidel a ty poté adresovat.

Postup čtení teploty je patrný z obrázku 6-2.



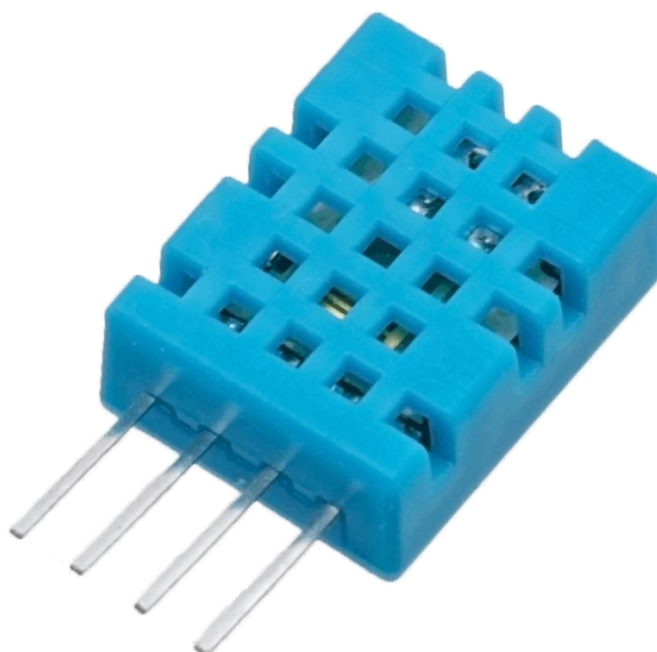
Obrázek 6-2 Vývojový diagram čtení teploty z DS18B20

## 6.3 Čidlo vlhkosti DHT11

Jedná se o vlhkostní a teplotní senzor, zobrazený na obrázku 6-3, měřící relativní vlhkost o rozsahu 20 – 90% s přesností  $\pm 5\%$ . Podobně jako u DS18B20 se jedná o jedno vodičové propojení. Na rozdíl od předchozího čidla, DHT11 není možné adresovat. Proto na jedné sběrnici může být připojeno pouze jedno.

Komunikace se senzorem probíhá na stejném principu jako u DS18B20. Tedy změnou hodnoty na pinu. Kromě vlhkosti měří čidlo i teplotu. U měření teploty ovšem dochází k větší nejistotě než u předchozího čidla a měření je možné pouze od 0-50°C.

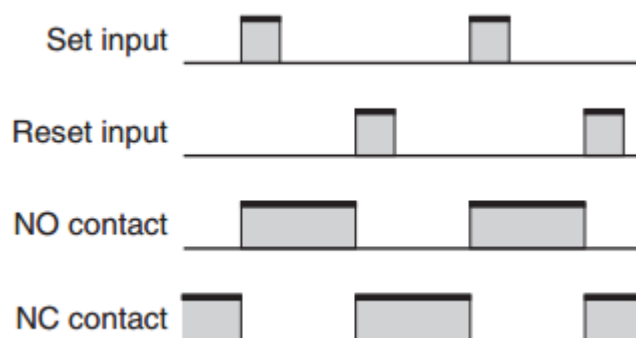
Pro dosažení větší přesnosti měření je vhodné použít namísto DHT11 senzor DHT22, komunikující stejným způsobem, ale je přesnější a dražší.



Obrázek 6-3 Kombinované čidlo DHT11

## 6.4 Bistabilní relé

Bistabilní relé je spínací prvek, který lze jednoduše používat v bateriově poháněných zařízeních pro spínání. Pro sepnutí výstupního kontaktu není nutné, aby napětí na spínací cívce bylo přivedeno trvale, ale stačí pouze nastavovací impuls. Některé relé mají 2 vstupy. Jeden je použit pro nastavovací impuls a resetovací impuls, tak jak je vidět na obrázku 6-4. Jiné typy těchto relé se řídí polaritou impulsu mezi dvěma vývody.



Obrázek 6-4 Časové impulsy na bistabilním relé, převzato z [5]

## 6.5 Displej

Pro zobrazení informací o aktuálním stavu stanice, případně pro zobrazení aktuálních naměřených hodnot je vhodné doplnit stanici displejem. Nejdostupnější a nejpoužívanější displej v podobných zařízeních je znakový displej.

Alternativou znakového displeje je grafický displej, případně lépe vypadající OLED displej. OLED displej využívá technologii organických elektroluminiscenčních diod. Tyto displeje lze nejčastěji najít v MP3 přehrávačích. Protože je použitý materiál v jednotlivých subpixelech z organické hmoty, přivedením napětí dojde k jeho rozsvícení. Díky tomu tyto panely nepotřebují oproti LCD displejům své vlastní podsvícení. Další výhodou je dokonalé zobrazení černé, kdy daný pixel nesvítí, k čemuž se váže i nevýhoda těchto panelů, tedy energetická náročnost při rozsvícení pixelu.

V této práci je použit modul OLED displeje, na obrázku 9-3, umístěný na plošném spoji, s rozlišením 128 x 64 px. Komunikace s tímto displejem, napájeným 3,3 V, probíhá pomocí sériové sběrnice I2C.

## 7 ZKUŠEBNÍ DESKA

Pro testování modulu a jednotlivých periférií byla vytvořena zkušební deska (dále jen prototyp) WiFi stanice. Na tomto prototypu byl také testován obslužný software modulu ESP8266.

### 7.1 Hardware

Plošný spoj prototypu byl navrhnut v návrhovém programu Altium Designer.

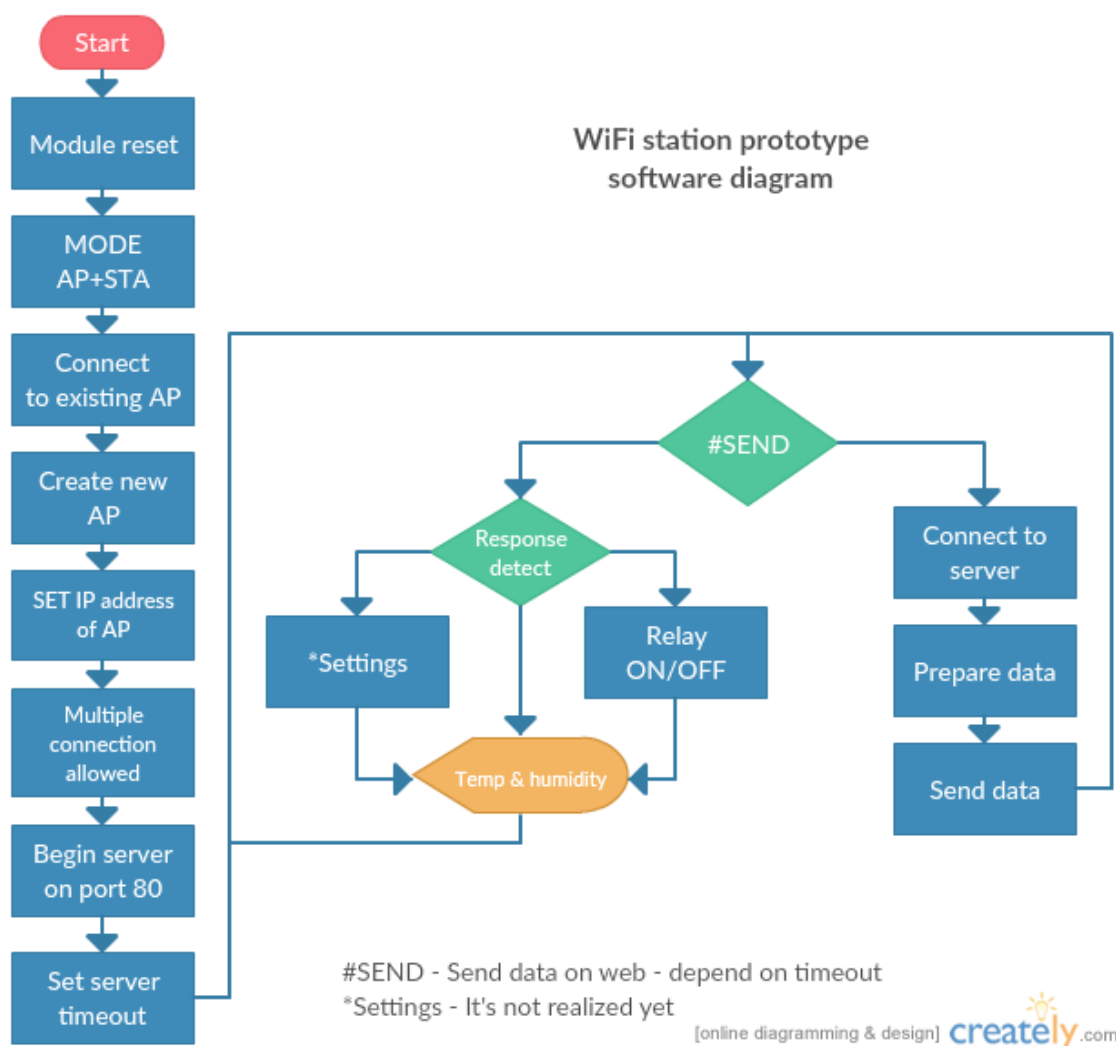
Jako napájecí zdroj byl použit modul se spínaným zdrojem LM2596 [8], jehož výstup byl nastaven na 5 V. Toto napětí bylo poté stabilizováno na 3,3V lineárním stabilizátorem TS1117. Volba tohoto napájení se nakonec ukázala jako ne příliš vhodná. Pro výslednou aplikaci bude vhodnější použít zdroj s výstupem přímo 3,3 V.

Do návrhu byl také umístěn vstup AD převodníku pro měření vstupního napětí i s příslušnými děliči. V prvním návrhu byl součástí i USB vstup tvořený USB konektorem a převodník FTDI232. Díky tomu byl do návrhu umístěn i přepínač sériové linky tvořený čtyř kanálovým multiplexorem 74HC4052.

Schéma i motiv plošného spoje jsou umístěny v příloze.

### 7.2 Obslužný program

Pro prototyp byl vytvořen software, na němž byl modul testován, a byla zjištěna posloupnost jednotlivých příkazů. Tento program je popsán vývojovým diagramem na obrázku 7-1.



Obrázek 7-1 Vývojový diagram obslužného programu prototypu

Jakmile dojde program do bodu, na obrázku 7.1 označeném jako #SEND, může se uživatel za pomoci externího zařízení (notebook, tablet, chytrý telefon, ...) připojit k vytvořené WiFi síti stanice. Připojení se provede zadáním IP adresy do adresního řádku vyhledávače. IP adresa je napevno nastavena na 192.168.4.1. Po zobrazení stránky s aktuálními naměřenými údaji je možné řídit spínání relé.

Odesílání naměřených údajů na internetový server je proces periodicky opakující se v určitém časovém intervalu. V případě, že je možno se připojit na server, jsou změřena data a následně odeslána na server.

### 7.3 Obslužný server

Data jsou na server odesílána ve formě protokolu GET. Hodnoty jsou předávány ve formě parametrů v adresním řádku. Příkladem takového způsobu předávání dat může být následující parametr adresy internetové stránky, kde XX představuje předanou hodnotu teploty.

```
http://station.smartsys.cz/receiver.php?temp=XX
```

Takto předaná data jsou zpracována skriptem v souboru receiver.php. Tento PHP skript vezme předané hodnoty a uloží je společně s aktuálním časem a datem do textového souboru.

Tyto uložené údaje lze poté později zpracovat do statistik, případně do grafů.

V případě, že není k dispozici webhosting nebo server, na kterém lze provozovat podobný sběr údajů, bylo by nutné naměřené údaje uchovávat alespoň na paměťové kartě pro pozdější přečtení hodnot.

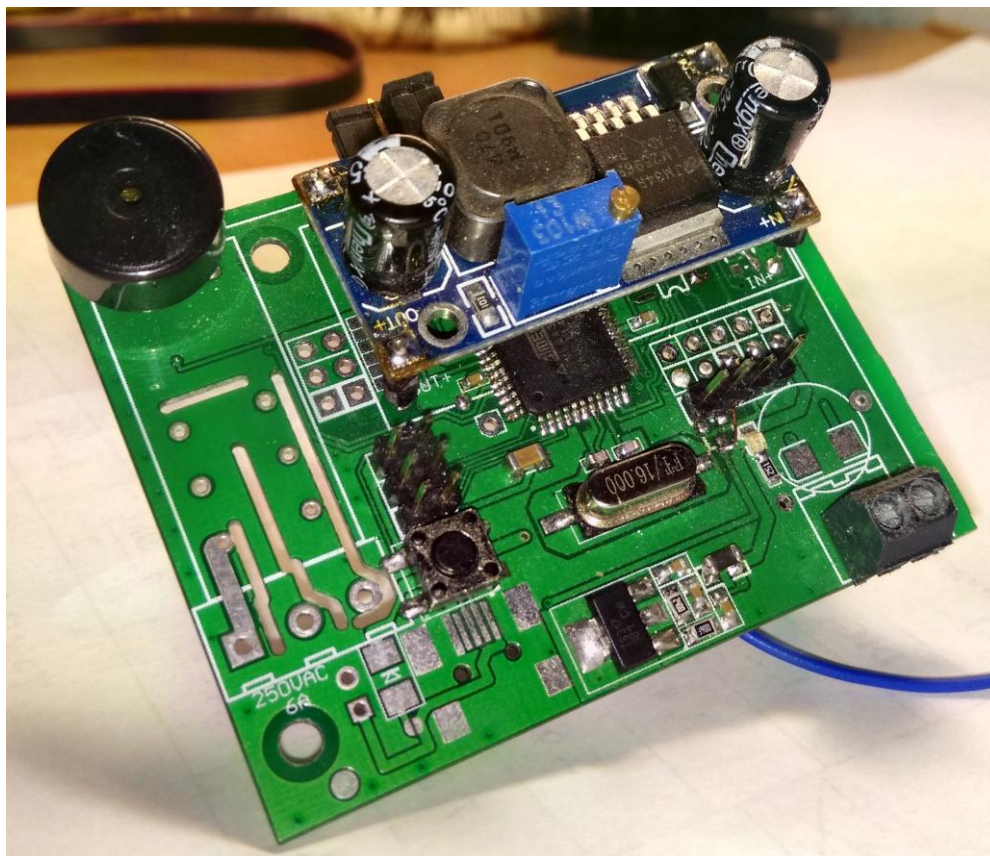
## 7.4 Výroba prototypu

Výroba plošného spoje byla zadána firmě Itead Studio [6]. Za asi 700 Kč bylo obdrženo 10 ks plošných spojů. S těmi mohlo být později experimentováno dle libosti.

Většina součástek byla na plošném spoji osazena technologií pájení v parách. Výhodou použití této technologie pájení je, že ve spojích nevznikají prasklinky jako při klasickém pájení páječkami. Tato metoda je také šetrnější ke krystalům, z hlediska teploty pájení, a k polovodičovým součástkám.

Při osazování byly zjištěny nedostatky plošného spoje způsobené především tím, že se jednalo o první návrh plošného spoje v návrhovém programu Altium Designer. Jedním z nich je vynechání vývodu s označením enable ze schématické značky obvodu U102 74HC4052. Tato chyba měla za následek nepřítomnost signálu na výstupu obvodu. Musela být odstraněna odškrabáním zelené masky plošného spoje a vývod propojen na zemnicí plochu. Další chybou bylo rozlité zemní plochy i pod část WiFi modulu kde se nachází keramická anténa. To má za následek nežádoucí odrazy signálu a snížení tak dosahu WiFi signálu. Jedinou možnou opravou je vyvarovat se tomuto jevu u výsledné podoby desky. Protože bylo zakoupeno jiné bistabilní relé, než na které byl motiv navržen, nemohlo být osazeno.

Na obrázku 7-2 je vyobrazen osazený plošný spoj prototypu.



Obrázek 7-2 Výsledná podoba prototypu

## 7.5 Testování prototypu

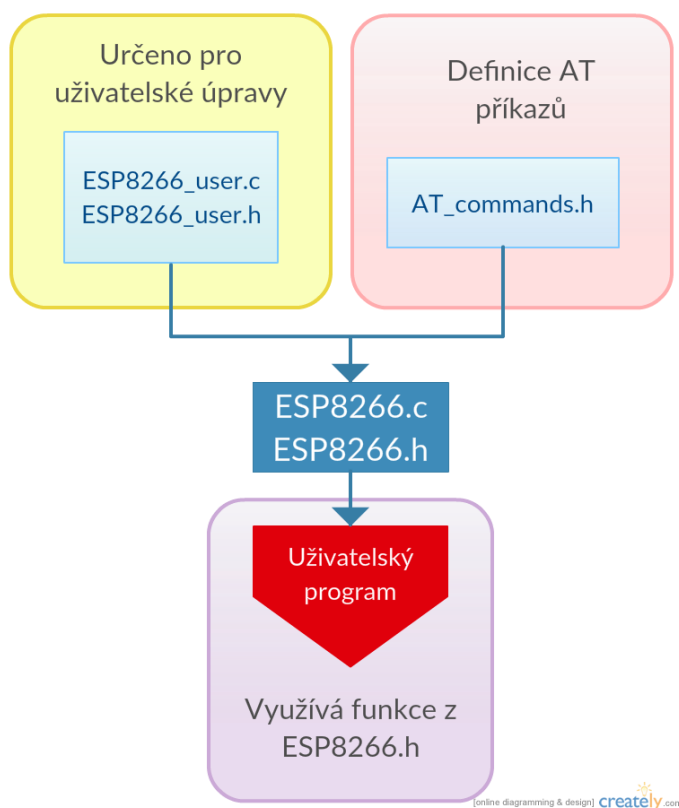
Testování prototypu, obzvláště pak stability připojení k externímu serveru probíhalo periodickým odesíláním naměřené teploty, respektive jakékoliv hodnoty na server. Toto odesílání se opakovalo vždy po třech až čtyřech vteřinách. Záznam z testování je k dispozici na [7].

Z výsledků testování vyplynulo, že až na několik výpadků, kdy nedošlo k odeslání dat je spojení stabilní. Zmíněné výpadky nemusely být přímo výpadky, ale i zpoždění připojení k serveru. U testovacího software nebyla tato chyba ošetřena. Modul je tedy možné využívat i k dlouhodobým záměrům měření.

## 8 KNIHOVNA PRO ESP8266

Jedním z cílů této práce bylo vytvořit knihovnu pro mikrokontrolér, umožňující komunikaci a obsluhu WiFi modulu ESP8266. Knihovna byla vytvořena pro mikrokontroléry AVR, přesto je možné ji převzít do mikrokontrolérů jiné architektury a za pomoci direktivy `#define` pro některé funkce ji začít používat.

### 8.1 Struktura knihovny



Obrázek 8-1 Struktura knihovny

Knihovna se skládá celkem z pěti souborů. Pro lepší pochopení struktury knihovny slouží obrázek 8-1. Soubory ESP8266.c a ESP8266.h obsahují nejdůležitější funkce nutné pro inicializaci a obsluhu modulu. Jejich součástí jsou i výčtové typy používané pro lepší navigaci mezi jednotlivými kroky. Soubor se zdrojovým kódem ESP8266.c není určen pro uživatelskou úpravu. V hlavičkovém souboru ESP8266.h je vhodné upravit definici funkcí pro odesílání a příjem dat ze sériového rozhraní.

Soubory ESP8266\_user.c a ESP8266\_user.h obsahují funkce, které jsou jako jediné určeny pro úpravu uživatelem.

Hlavičkový soubor AT\_commands.h obsahuje nejdůležitější AT příkazy, a jejich obměny pro různé módy ve kterých může modul pracovat. V tomto souboru jsou i



definovány název a heslo vytvářeného přístupového bodu a název s heslem přístupového bodu, ke kterému se bude modul připojovat.

## 8.2 Hlavní část knihovny

Základem knihovny jsou tedy soubory ESP8266.c a ESP8266.h. Celá tato knihovna se může na první pohled zdát jako velice obsáhlá a zbytečně dlouhá. Avšak tato obsáhlost je způsobena pouze větším množstvím příkazů switch, díky nimž je provedena vždy pouze daná část dle nastaveného *workingMode*. Podobného efektu lze docílit i využitím definic pro překladač, kdy by se používaly definované podmínky typu `#ifdef` a `#endif`. Avšak, nebyla by zde možnost programově měnit pracovní mód (tedy nastavovat proměnou *workingMode*).

Nejčastěji využitou funkcí je **SendCommand**. Vstupními parametry jsou příkaz k odeslání, za který se automaticky doplní zakončovací znaky CR a LF, dvě možné verze správné odpovědi a dvě možné verze odpovědi při chybě. Po odeslání příkazu následuje while cyklus. Ten zajistí, že přijímání odpovědi bude probíhat, dokud podmínky nevyhodnotí, že přijatá data obsahují jednu ze čtyř odpovědí. V případě, že je splněna některá z podmínek, ukončí se cyklus a funkce vrátí OK (definováno jako 1) nebo BAD (definováno jako 0).

Pro nastavení pracovního módu, tedy jestli má modul pracovat jako klient, přístupový bod, nebo jako kombinace obojího slouží funkce **ESP8266\_setWorkongMode**, která má jako vstup parametr z výčtového typu *ESP8266\_mode*. Z tohoto výčtového typu lze zvolit CLIENT (modul pracuje jako klient), AP (modul pracuje jako přístupový bod), nebo AP\_CLIENT (modul vytvoří přístupový bod a zároveň se umí připojit do jiné existující sítě). Pro zjištění jaký je aktuálně nastavený mód lze využít funkci **ESP8266\_getWorkingMode**, která vrátí odpovídající hodnotu z výčtového typu.

Knihovna dále využívá flag *busy*, který lze využít jako ochranu proti kolizím mezi příkazy a detekci zaneprázdnění modulu. Tento flag je nastaven vždy, když má dojít k odeslání příkazu a po zpracování odpovědi je vymazán. Pro přístup k tomuto flagu lze využít funkce **ESP8266\_clearBusy** a **ESP8266\_setBusy**, jejichž funkce vychází z názvu. Stav tohoto flagu lze zjistit pomocí funkce **ESP8266\_isBusy**, která vrátí 1 pro nastavený flag a 0 pro vymazaný flag.

Dvě funkce, které ovládají napájení a resetovací pin modulu, **ESP8266\_Power** se vstupním parametrem 1 pro zapnuto a 0 pro vypnuto, a funkce **ESP8266\_makeRST**, která vytvoří hardwarový reset modulu. Tyto funkce jsou psané sice obecně, a lze je změnou v hlavičkové souboru přemapovat na jiné piny, ale jsou určeny pro architekturu AVR. Architektura ARM většinou využívá pro ovládání GPIO pinů struktury, takže by bylo nutné tyto funkce pozměnit.

Pro základní inicializaci modulu slouží funkce **ESP8266\_init**, která využívá globálně nastavenou proměnou *workingMode*. Podle její hodnoty dojde k inicializaci dané části. Návratovou hodnotou této funkce v případě úspěchu je *INIT\_OK* z výčtového typu *ESP8266\_state*. Tento výčtový typ může nabývat následujících hodnot odpovídajících jednotlivým krokům inicializace: *INIT\_OK* (úspěšná inicializace), *RESET* (chyba při restartování modulu), *AT* (chyba při základní odpovědi

modulu – není v ukázce použito), MODE (chyba v nastavení pracovního režimu), JOINAP (chyba při připojení k existujícímu přístupovému bodu), CREATEAP (chyba při vytvoření AP), MUX (chyba povolení většího množství aktivních spojení), SERVER (chyba vytvoření serveru) a TIMEOUT (chyba při nastavení maximálního času pro nahrávání stránky do prohlížeče). V případě selhání některého z kroků inicializace vrací jinou hodnotu z tohoto výčtového typu danou krokem, kdy došlo k selhání. Tato funkce nemá návratovou hodnotu pouze pro příkaz připojení k již existujícímu přístupovému bodu. V případě, že se nelze připojit, je nastavena globální proměnná *APNotExist*. Toto muselo být voleno jako kompromis, protože bylo zjištěno, že doba, kterou trvá modulu připojení se k přístupovému bodu, je odlišná, pro každý přístupový bod.

Pro odeslání jakýchkoliv dat na internetovou stránku slouží funkce **ESP8266\_sendDataOnWeb**. Tato funkce vrací OK v případě úspěšného odeslání a BAD v případě chyby. Jediným účelem této funkce je ověřit, zda jsou správné podmínky k odesílání na web a volání uživatelské funkce, která bude popsána později. Funkce ověřuje přítomnost flagu busy, správný *workingMode* a zda je modul připojen k existujícímu přístupovému bodu. Tato funkce je většinou volána například v časovači, pokud je zapotřebí periodické odesílání dat, nebo může být volána na základě nějaké akce, jako například stisknutí tlačítka a podobně.

Funkce, která by měla běžet v nekonečné smyčce, starající se o obsluhu příchozích požadavků, je **ESP8266\_detectResponse**. Ta po zjištění příchodu určitého požadavku vykoná určitou akci. Aby bylo možno upravit si zobrazovanou www stránku, tato funkce volá uživatelsky definovanou funkci **userDetectResponse**.

### 8.3 Uživatelské funkce

Odeslání dat na internetový server má na starosti uživatelská funkce **userFunctionForSendData**. Zde je nejdříve nutné definovat adresu, ke které se má modul připojit a port na kterém má proběhnout spojení. Dále je zapotřebí definovat délku odesílaných dat. Délka dat se spočítá tak, že se sečte počet symbolů odesílaných dat bez mezer a k tomuto součtu se přičte 1.

Příkladem odesílaných dat může být následující příkaz služby GET.

```
GET /receiver.php?temp=25&hum=31&crc=086 HTTP/1.1\r\nHost:
station.smartsys.cz\r\n
```

Tento příkaz otevře soubor receiver.php s parametry temp=25, hum=31 a crc=086. Otevření se provede za pomoci služby HTTP verze 1.1 na adrese station.smartsys.cz. Pro tento příkaz je součet znaků bez mezer 79. Do příkazu AT+CIPSEND se tedy odešle 80. O zpracování parametrů se stará právě otevíraný soubor receiver.php. Tato funkce má návratové hodnoty pro případ úspěšného odeslání dat, i pro případ selhání. Tyto návratové hodnoty pocházejí z výčtového typu *ESP8266\_web*. Prvním krokem odeslání dat je připojení se ke vzdálenému serveru. V případě neúspěchu funkce vrací CONNET. PREPARE je vráceno v případě, že selže odeslání počtu zamýšlených znaků. Posledním krokem je odeslání dat na server. Selhání tohoto kroku má za následek, že tato funkce vrací SEND. Naopak v případě úspěšného odeslání dat funkce vrací WEB\_OK.

Pro detekci připojení uživatele k zařízení slouží funkce **userDetectResponse**, ve které proběhne příjem ze sériového rozhraní a detekce hledané sekvence. Například pro zobrazení www stránky je nutné detekovat sekvenci `GET /`. Po nalezení této sekvence je volána funkce **userShowOnePage**.

Ve funkci **userShowOnePage** dojde k odeslání počtu znaků, které se budou vysílat, a poté k odeslání samotné www stránky v podobě HTML kódu. V příkladu knihovny obsahuje www stránka informaci o aktuální teplotě, vlhkosti a stavu relé.

## 9 REALIZACE WIFI STANICE

Na základě zkušeností s prototypem byla vytvořena WiFi stanice s možností připojení senzorů a spínáním relé. Pro demonstraci funkčnosti byla vytvořena meteostanice, jejímž jádrem je navržená WiFi stanice.

Podobně jako u prototypu je použit mikrokontrolér AVR ATmega32, modul ESP8266 ve verzi ESP-07 a bistabilní relé. Přímě na plošný spoj stanice byly umístěny senzor teploty DS18B20 a senzor vlhkosti DHT11.

### 9.1 Napájecí zdroj

Jako napájecí zdroj lze použít stejnosměrný nebo i střídavý zdroj tak, aby po usměrnění nepřesáhlo jeho napětí 17 V.

Toto napětí je maximální vstupní napětí pro použitý spínaný stabilizátor. Je použit modul vlastní výroby spínaného zdroje s výstupním napětím 3,3 V a maximálním výstupním proudem 1 A. Modul je osazen integrovaným obvodem TPS62160 od firmy Texas Instruments v osmi pinovém pouzdru WSON. Účinnost tohoto obvodu se pohybuje v okolí 80 % vždy v závislosti na odebíraném proudu a vstupním napětí.

Použitý integrovaný obvod má synchronní usměrňovač, takže není nutné mít připojenou externí diodu, čímž pádem dojde k úspoře jak finanční, tak i k úspoře místa na plošném spoji. Díky vysoké spínací frekvenci měniče, typicky  $f = 2,25 \text{ MHz}$  lze použít filtrační tlumivku s malou indukčností a tím i malým rozměrem, typicky  $L = 2,2 \mu\text{H}$ .

Použitý modul o velikosti 23,8 x 10,7 mm je zapojen dle doporučení výrobce v katalogovém listu.

Kvůli náročnosti návrhu layoutu DPS pro spínaný zdroj a možnosti osazení jiného zdroje napětí, je ve WiFi stanici použit modul, který je za pomoci pinové lišty osazen nad můstkovým usměrňovačem. Schéma zapojení a layout tohoto modulu, nakreslené v programu Altium Designer, jsou uvedeny v příloze A.

### 9.2 Schéma zapojení stanice

Schéma stanice je rozděleno do dvou částí. První částí je napájecí obsahující napájecí konektory, značku pro spínaný měnič, lineární stabilizátor a filtrační prvky. Druhá část je řídicí. Obsahuje mikrokontrolér, WiFi modul, relé a vstupně výstupní konektory.

Na stanici lze osadit dva typy konektorů pro napájení. Je možno osadit šroubovací konektor typu ARK, nebo konektor typu jack, který je přítomný u napájecích adaptérů. Aby vstup nebyl závislý na polaritě, vstupní napětí je usměrněno diodovým můstkem. Za tímto můstkem následují zmíněný spínaný měnič s výstupním napětím 3,3 V. Paralelně s tímto měničem je zapojen lineární stabilizátor LM317, respektive TS1117 s výstupním napětím 5V pro relé. Obě tyto napětí jsou dále filtrována pomocí článku

ferit a kondenzátor. Ferit 120R/100MHz je zapojen do série a paralelně k zemi je zapojen kondenzátor 100nF.

V řídicí části je zapojen procesor, ke kterému jsou zapojeny další periférie. Modul ESP8266 je k procesoru připojen, tedy UART, přes dva nulové rezistory umožňující odpojit modul od procesoru. Napájení modulu je připojeno přes dva unipolární tranzistory s N kanálem, které jsou řízeny z mikrokontroléru.

Pro připojení periférií přes rozhraní I2C slouží konektor JP101, na kterém je kromě signálu I2C, SDA a SCL přítomno i napájení, které lze volit mezi 3,3V a 5V. Konektor JP100 slouží pro připojení tří analogových vstupů s možností osadit děliče napětí, na jejichž výstupu bude maximálně 3,3 V tak, aby nedošlo k poškození mikrokontroléru. Tyto děliče napětí jsou navíc jištěny zenerovou diodou. Součástí stejného konektoru jsou i dva digitální vstupy respektive výstupy. Protože programování procesoru se provádí pomocí ISP, je na plošném spoji vyvedeno i SPI. Díky těmto konektorům lze ke stanici připojit téměř libovolný senzor případně ovladač.

## 9.3 Layout DPS

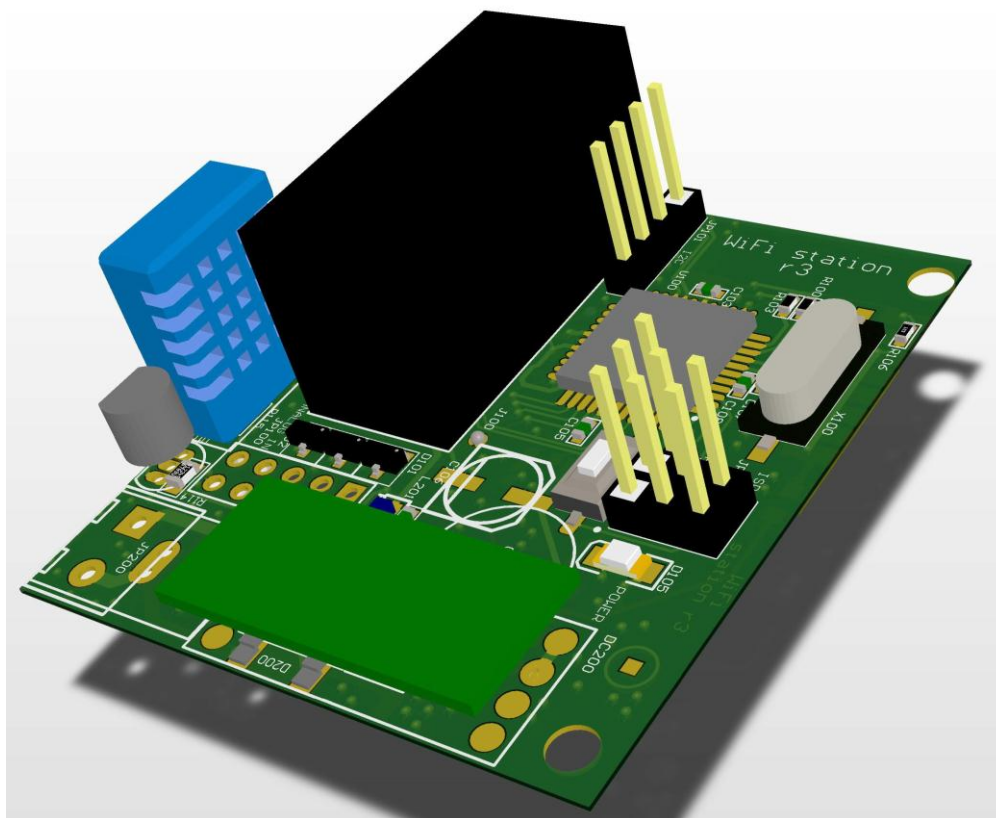
Jak již bylo řečeno, v prototypu došlo k chybnému rozlití polygonu země u modulu ESP8266. Toto bylo napraveno vytvořením clearance, tedy plochy, na které nebude signál GND rozlity.

Také došlo k uspořádání konektorů a senzorů pro větší praktičnost použití plošného spoje. Pinová lišta pro rozhraní I2C byla nahrazena konektorem typu PSH02-04PG, tak aby mohlo dojít k případnému odpojení periférií.

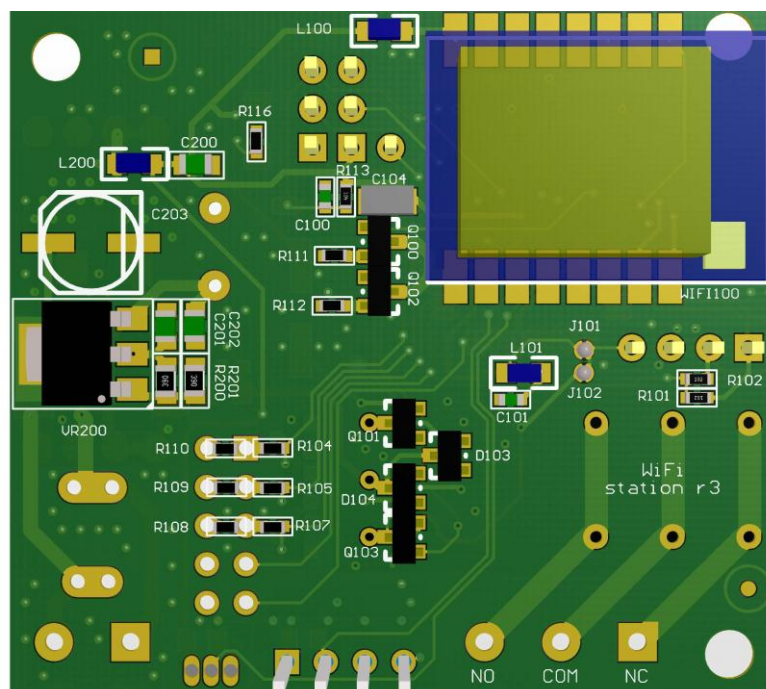
Teploměr DS18B20 byl umístěn mimo plošný spoj stanice, do kovové trubičky. Připojení teploměru je provedeno připájením konektoru PSH02-03WG na pozici teploměru.

Plošný spoj byl navržen s ohledem na univerzálnost použití, takže je možné jej využít jako stanici pro měření a odesílání dat, bezdrátovou stanici s relé anebo jako v případě demonstrace stanice kombinovaně.

Plošný spoj, jehož vizualizace jsou na obrázcích 9-1 a 9-2, má velikost 50 x 45 mm a byl vyroben ve firmě Itead Studio [6]. Schéma i motiv plošného spoje, nakreslené v programu Altium Designer, společně s rozpiskou součástek jsou uvedeny v příloze B.



Obrázek 9-1 Vizualizace osazeného plošného spoje



Obrázek 9-2 Vizualizace plošného spoje stanice ze strany bottom

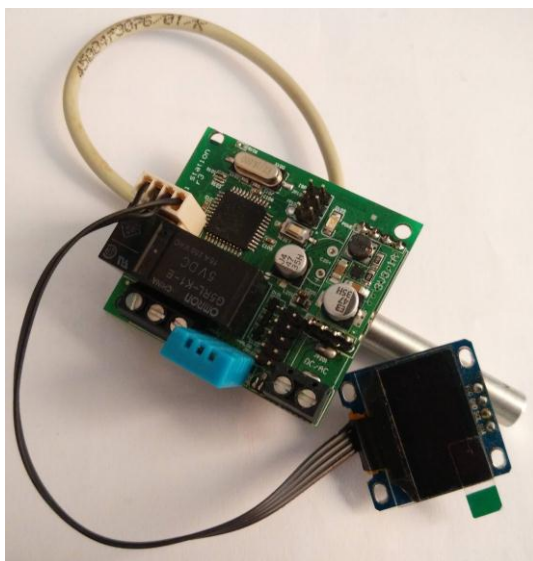
## 9.4 Realizace meteostanice

Díky vytvořenému návrhu stanice byla vyrobena meteostanice umožňující měření teploty, vlhkosti a spínání relé pro uživatelská zařízení. Obrázky 9-3 a 9-4 zobrazují realizovanou meteostanici.

Osazení plošného spoje proběhlo za použití technologie pájení v parách. Stejnou technologií se osazoval i modul spínaného měniče.



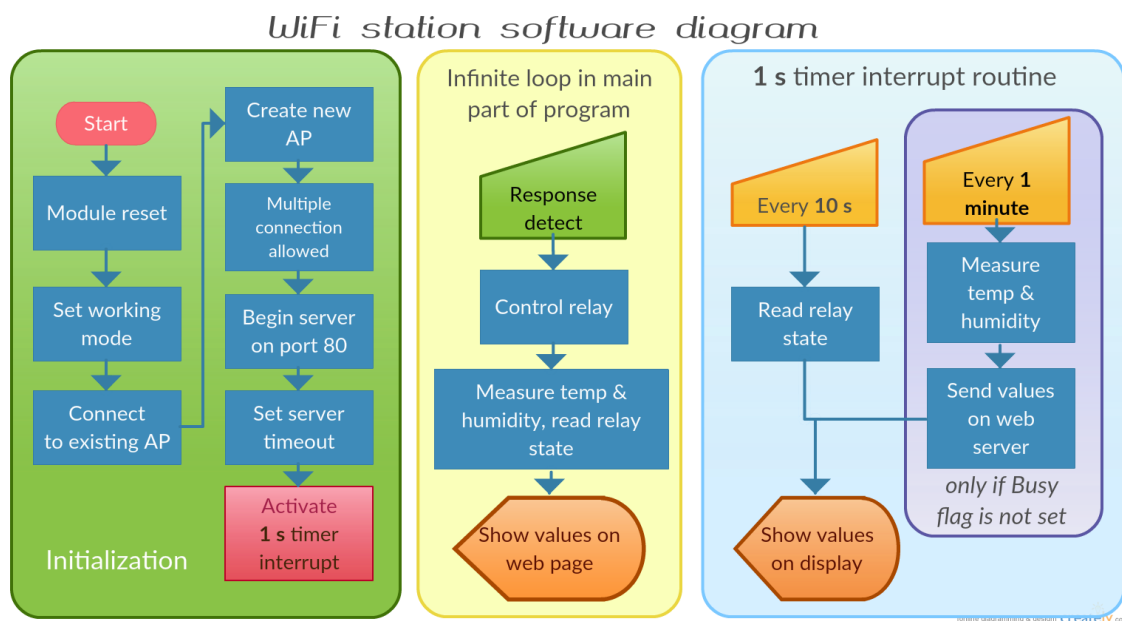
Obrázek 9-3 Displej se zobrazenými údaji



Obrázek 9-4 Osazený plošný spoj s připojeným displejem a teploměrem

### 9.4.1 Řídicí program meteostanice

Do řídicího mikrokontroléru ATmega32 byl nahrán program, který má implementovanou vytvořenou knihovnu pro ESP8266. Dále umožňuje čtení teploty, vlhkosti a ovládání relé. Naměřené hodnoty a aktuální stav relé jsou zobrazovány na OLED displeji.



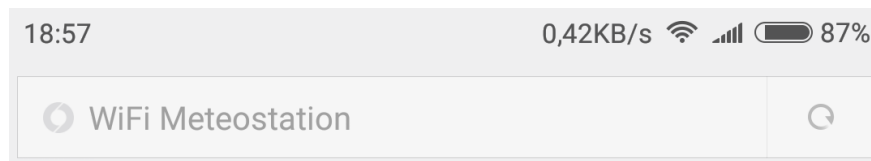
Obrázek 9-5 Vývojový diagram finálního zařízení

Tento program je popsán diagramem na obrázku 9-5. Jak lze vidět, je program rozdělen do celkem tří částí. První částí je inicializace, kdy dojde k základnímu nastavení WiFi modulu. Tato část proběhne ihned po startu mikrokontroléru a dojde až do bodu aktivace časovače. Do tohoto bodu program dojde pouze v případě, že nedojde k chybě v odpovědi modulu ESP8266. V opačném případě dojde k restartování procesoru, a tím i k restartování modulu.

Další důležitou částí programu je možnost zobrazovat webové stránky v případě připojení se ke stanici. Připojení je umožněno přihlášením klienta k WiFi síti nazvané *Meteostation* s heslem *smartmeteo*. Poté je v prohlížeči možno zadat adresu <http://192.168.4.1>, čímž dojde k zobrazení stránky, jako na obrázku 9-6. Toto je zajištěno částí programu, která běží v nekonečné smyčce procesoru. Je v ní zajištěno přijímání dat z modulu a jejich zpracování. V případě, že jsou v přijatých datech nalezeny určité řetězce, jako `GET /` nebo `GET /rele` dojde ke změření teploty a vlhkosti a k odeslání webové stránky do prohlížeče. V případě že je nalezen řetězec `GET /rele` dojde k přepnutí stavu relé a toto je taktéž indikováno na webové stránce.

Pokud zrovna neprobíhá zobrazování webové stránky, dochází k přerušení časovače ve vteřinových intervalech. Každých deset vteřin dojde ke čtení stavu relé, který je uložen v paměti EEPROM. Tento stav je následně zobrazen na OLED displeji. Každou celou minutu od začátku čtení časovače dojde ke změření teploty a vlhkosti. Tyto hodnoty jsou poté odeslány na internetový server, kde jsou uloženy a zpracovány. Součástí přenášených dat je ochrana proti zneužití a ztrátě informací. Jedná se o cyklický redundantní součet CRC, kdy dojde k sečtení hodnoty teploty a vlhkosti. Tato hodnota je přenesena na server. Ten pokud tento součet souhlasí, uloží naměřené údaje.





## WiFi Meteostation

powered on ESP8266

Actual temp: 22 °C

Actual humidity: 37 %

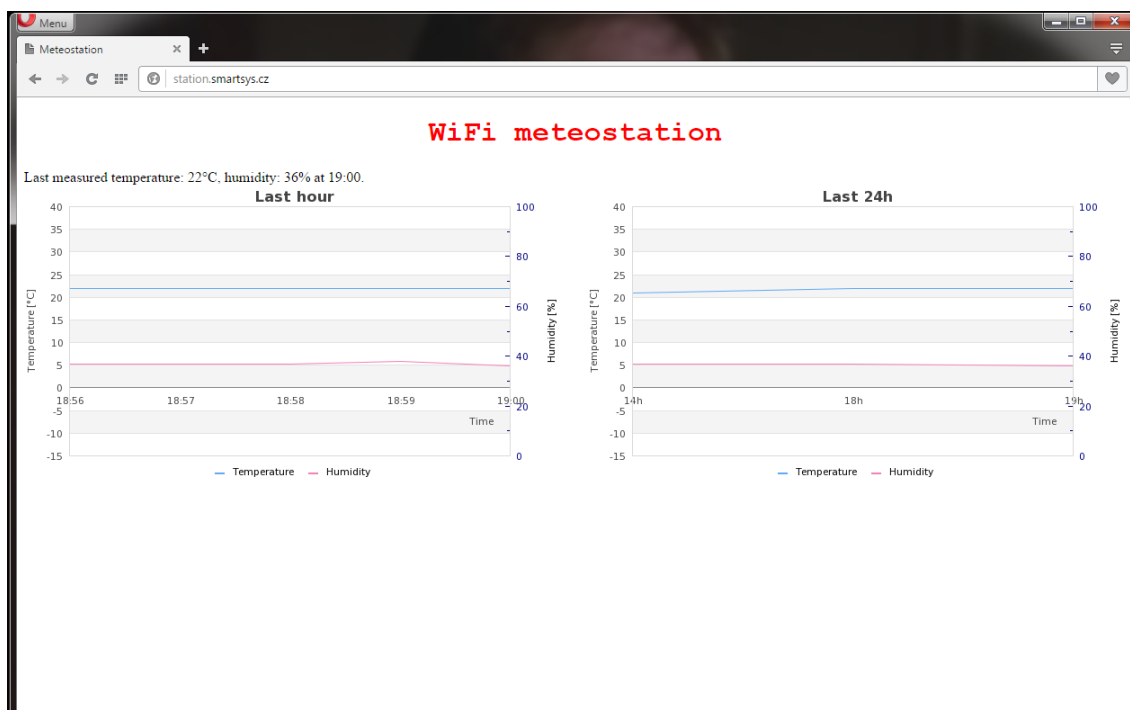
Relay is OFF

Obrázek 9-6 Zobrazená webová stránka

### 9.4.2 Internetový server

Pro ukládání dat na internet existuje velké množství již hotových řešení, které jsou dostupné široké veřejnosti. Příkladem takového serveru může být stránka ThingSpeak [9]. Tento portál byl vytvořen právě za účelem sbírání a prezentace dat z IoT zařízení. Po založení účtu na tomto portálu lze vytvořit kanály určené pro jednotlivá zařízení a po obdržení adresy, přes kterou je umožněno nahrávání dat, lze tento portál začít používat. Příkladem takového sběru dat je kanál sbírající data z čidla DHT 11 [10].

Pro lepší možnost modifikace a prezentace dat byla vytvořena stránka Meteostation [11] na serveru SmartSys.cz. Tato stránka je také vyobrazena na obrázku 9-7. Stránka byla postavena na technologii jazyka PHP s využitím knihovny JpGraph [12] umožňující vytvoření velkého množství druhů grafů reprezentujících vstupní data. Tyto grafy jsou následně díky této knihovně převedeny do obrázku, které lze zobrazit uživateli.



Obrázek 9-7 Internetová stránka s grafy naměřených hodnot

Ukládání dat ze stanice na server obstarává php skript `receiver.php`. Tento skript očekává tři parametry ve formátu `receiver.php?temp=25&hum=31&crc=086`, kde 25 je změřená teplota, 31 je změřená vlhkost a 086 je CRC ochranný součet. Za předpokladu že se shoduje CRC součet s hodnotami teploty a vlhkosti, jsou data uložena do textového souboru s aktuálním datem. Těmto datům je navíc přidána časová značka, tak aby hodnoty teploty a vlhkosti v grafu mohly být seřazeny a zobrazeny i s časem. Z těchto souborů mohou být následně načteny data.

Pro toto konkrétní zařízení dochází k vytvoření dvou spojnicových grafů při každém načtení internetové stránky [11]. Jeden zobrazuje data z poslední hodiny. Druhý graf zobrazuje údaje za posledních 24 hodin. Při načtení této stránky dojde k otevření příslušných textových souborů a k jejich procházení. Do grafu zobrazující údaje z poslední hodiny se použijí pouze údaje, jejichž časová značka není starší než 60 minut. Je tedy využito funkce jazyka PHP, která umí vrátit aktuální čas nebo datum. Graf zobrazující posledních 24 hodin, prochází příslušné soubory a používá první naměřenou hodnotu v každé hodině. Skripty procházející naměřené údaje, umí otevřít i soubor ze dne předchozího a načíst z něj příslušné hodnoty, pokud je to zapotřebí.

Bylo by vhodné, aby se dalo relé na stanici spínat i přes toto internetové rozhraní. To by ovšem vyžadovalo mnohem častější komunikaci stanice se serverem, aby byla snížena latence.

Místo ukládání dat do textových souborů lze naměřené údaje ukládat také do databáze SQL. Toto řešení je modernější a více flexibilní. Procházení dat z databáze je jednodušší a nebylo by zapotřebí psát skript, který prochází více souborů apod. Avšak toto řešení má nevýhodu spočívající v tom, že databázový server musí být k dispozici. Toho je jednoduché dosáhnout na internetovém serveru. Avšak v případě hostování tohoto řešení na domácím serveru, například na mini počítači Raspberry Pi [17] bývá

problém pro nezkušené uživatele vytvořit a spravovat databázi.

Pokud by však bylo toto řešení s grafy hostováno na zmiňovaném mini počítači umístěném ve vlastní síti, nebyl by problém snížit latenci a umožnit tak spínání relé i ze stránky s relé. V případě internetového hostování je možné latenci také snížit, avšak mohl by nastat problém s poskytovatelem internetové připojení i s majitelem hostingu týkající se velkého množství požadavků.

Latence modulu a serveru byla testována na odlišném projektu, konkrétně na Chytré zásuvce umožňující měřit výkon koncového zařízení a toto zařízení i spínat. Spínání bylo provedeno přes internetovou stránku. Bylo zjištěno, že bezpečná doba, po které lze opakovat připojení na server, jehož majitelem je společnost Wedos, jsou tři vteřiny. Tato doba se ukázala být pro spínání relé jako dostatečná. Avšak není to nejmenší doba spojení. V případě modifikace obslužného programu lze tuto dobu ještě více zkrátit.

## 10 ZÁVĚR

V práci byla navržena stanice pro využití WiFi modulu ESP8266 umožňující připojit senzory a ovladače do sítě IoT.

První kapitola se zabývá problematikou sítě Internet of Things, jejími výhodami a nevýhodami. Následující kapitola popisuje samotný modul a je zde pojednáváno o možnostech zabezpečení komunikace. Třetí a čtvrtá kapitola obsahují informace o napsání vlastního obslužného programu pro modul ESP. Pátá kapitola je jakýmsi manuálem s popisem AT příkazů, které lze použít pro ovládání modulu. V další kapitole lze nalézt popis jednotlivých komponent, ze kterých byl sestaven prototyp a nakonec výsledná stanice. Sedmá kapitola se věnuje poznatkům z prototypu WiFi stanice. Následující kapitola pojednává o vytvořené knihovně umožňující ovládat modul ESP8266. V deváté kapitole lze nalézt popis kombinované WiFi stanice.

Popisy programování vlastního obslužného programu uvedené v kapitolách tři a čtyři nejsou tak obsáhlé jako manuál k AT příkazům z kapitoly pět. Důvodem k tomu je fakt, že ovládáním modulu ESP8266 pomocí AT příkazů lze dosáhnout využití jeho plného potenciálu.

Díky vytvořenému návrhu WiFi stanice mohla být pro demonstraci funkčnosti vytvořena domácí meteostanice měřící teplotu a vlhkost. Také umožňuje spínat relé.

Po domluvě s vedoucím práce byla pro demonstraci vytvořena jedna univerzální WiFi stanice. Tímto byla ověřena funkčnost modulu ESP8266 pro režim přístupového bodu a klienta současně. Toto zařízení měří teplotu a vlhkost a tyto údaje odesílá na internetový server, s adresou `station.smartsys.cz`, kde je možné si prohlédnout grafy prezentující naměřené údaje. Pokud je inicializováno připojení klienta ke stanici a zadána do prohlížeče adresa `192.168.4.1` je zobrazena stránka umožňující spínat relé.

Součástí práce měla být analýza případně návrh protokolu zabezpečení. Toto bylo provedeno pouze teoreticky v kapitole 2.5. V praxi se bohužel nepodařilo toto řešení implementovat na internetový server patrně díky neúplné znalosti jazyka PHP. Proto byla zvolena alespoň ochrana proti ztrátě dat pomocí CRC ochrany.

# LITERATURA

- [1] Portál T-PRESS. [online]. 2015 [cit. 2015-12-05] Dostupné z: <http://t-press.cz/en/press-releases/press-news-archive/sigfox-a-new-wireless-network-for-the-internet-of-things-in-the-czech-republic.html>.
- [2] Portál Sparkfun, ESP8266 Thing Hookup Guide. [online] 2015 [cit. 2015-12-05] Dostupné z: <https://learn.sparkfun.com/tutorials/esp8266-thing-hookup-guide>.
- [3] Portál GitHub. NodeMCU. [online] 2015 [cit. 2015-12-05] Dostupné z: <https://github.com/nodemcu>
- [4] Portál All about circuits. Patrick Lloyd, *How to Make an Interactive TCP Server with NodeMCU on the ESP8266* [online] 2015 [cit. 2015-12-05] Dostupné z: <http://www.allaboutcircuits.com/projects/how-to-make-an-interactive-tcp-server-nodemcu-on-the-esp8266/>
- [5] Portál TME, datasheet G5RL-U/-K [online] [cit. 2015-12-06] Dostupné z: <http://www.tme.eu/cz/Document/0fc2b45042a4ad6cc204d1c2ceba74ca/G5RL-U-K-EN.pdf>
- [6] Portál Itead Studio. [online] [cit. 2015-12-06] Dostupné z: <https://www.itead.cc>
- [7] Portál SmartSys.cz, *Záznam z testování prototypu*, Kubiček R. [online] 2015 [cit. 2015-12-06] Dostupné z: <http://station.smartsys.cz/myFile.txt>
- [8] Portál Itead Studio. *LM2596 DC-DC Buck converter Step-Down Power Module Output 1,25 V – 35 V*. [online] [cit. 2015-12-09] Dostupné z: <https://www.itead.cc/power/dc-dc/lm2596-dc-dc-buck-converter-step-down-power-module-output-1-25v-35v.html>
- [9] Portál ThingSpeak. [online] [cit. 2016-05-20] Dostupné z: <http://thingspeak.com>
- [10] Portál ThingSpeak, *ESP8266 DHT11 Module*, sgconcept [online] [cit. 2016-05-20] Dostupné z: <https://thingspeak.com/channels/44345>
- [11] Portál SmatSys.cz, *Meteostation*, Kubiček R. [online] 2016 [cit.: 2016-05-20] Dostupné z: <http://station.smartsys.cz>
- [12] Portál JpGraph. [online] [cit.: 2016-05-20] Dostupné z: <http://jpgraph.net>
- [13] Portál TME. [online] [cit.: 2016-05-22] Dostupné z: <http://tme.eu>
- [14] Portál hamshop.cz. [online] [cit.: 2016-05-22] Dostupné z: <http://hamshop.cz>
- [15] Portál Banggood.com, prodej ESP8266 ESP-07. [online] [cit.: 2016-05-22] Dostupné z: <http://www.banggood.com/ESP8266-ESP-07-Remote-Serial-Port-WIFI-Transceiver-Wireless-Module-p-961247.html>
- [16] Portál PTshop, Senzor teploty a vlhkosti vzduchu DHT11. [online] [cit.: 2016-05-22] Dostupné z: <http://www.ptshop.cz/Senzor-teploty-a-vlhkosti-vzduchu-DHT11-d148.htm>
- [17] Portál Raspberry Pi. [online] [cit.: 2016-05-22] Dostupné z: <https://www.raspberrypi.org>

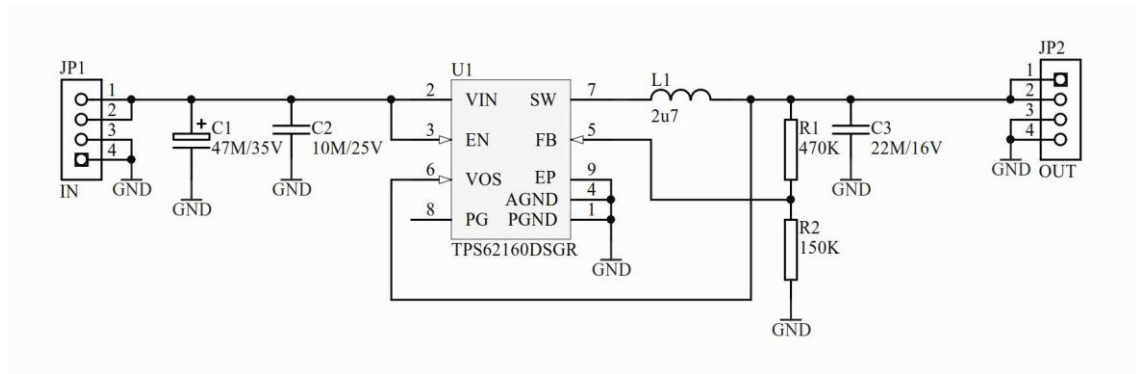
# SEZNAM SYMBOLŮ, VELIČIN, ZKRATEK A CIZÍCH SLOV

|              |                                                                                                                    |
|--------------|--------------------------------------------------------------------------------------------------------------------|
| AD převodník | Převod analogové hodnoty na digitální hodnotu – číslo.                                                             |
| AP           | Access point - přístupový bod v síti WiFi.                                                                         |
| ARM          | Architektura 32 a více bitových procesorů                                                                          |
| baud         | Jednotka přenosové rychlosti udávající počet změn přenosového média za jednu vteřinu.                              |
| Bluetooth    | Otevřený standard pro bezdrátovou komunikaci propojující dvě a více zařízení.                                      |
| FLASH paměť  | Paměť ke které je možno libovolně přistupovat, používaná v paměťových médiích.                                     |
| GPIO         | Označení pinu, jehož úroveň lze definovat v software.                                                              |
| I2C          | Sériová sběrnice pro komunikaci dvěma vodiči, označována také jako TWI – Two Wire Interface.                       |
| IP adresa    | Číslo identifikující síťové rozhraní v síti.                                                                       |
| latence      | Reakční čas; doba uplynulá mezi akcí a reakcí                                                                      |
| OLED displej | Zobrazovací displej využívající technologii organických elektroluminiscenčních diod.                               |
| P2P          | Označení síťového propojení, ve kterém spolu klienti komunikují přímo, bez serveru.                                |
| PWM          | Pulsně šířková modulace.                                                                                           |
| ROM          | Typ paměti, jejíž obsah je dán při výrobě, tento obsah nelze změnit.                                               |
| RS-232       | Standard pro sériovou komunikaci mezi dvěma zařízeními.                                                            |
| RS-485       | Standard pro sériovou komunikaci mezi dvěma zařízeními používaný především v průmyslovém prostředí.                |
| SQL          | Dotazovací jazyk využívaný pro práci s daty v databázích                                                           |
| SPI          | Sériové periferní rozhraní používané pro komunikaci mezi řídicím mikroprocesorem a ostatními integrovanými obvody. |
| TCP/IP stack | Kompletní sada síťových protokolů nutných pro komunikaci v síti.                                                   |
| UART         | Sériová komunikace založená na RS-232 ale s napěťovými úrovněmi vhodnými pro mikroprocesory.                       |
| Watchdog     | Časovač, který resetuje program při jeho zaseknutí.                                                                |

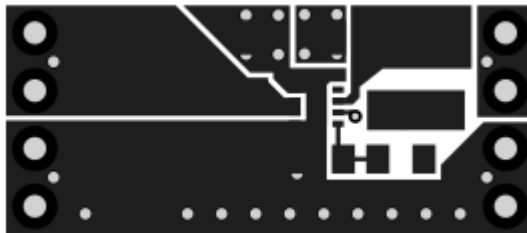
WiFi                      Označení pro standardy IEEE 802.11 popisující bezdrátovou síť.

## A MODUL DC-DC MĚNIČE TPS62160

### A.1 Schéma zapojení

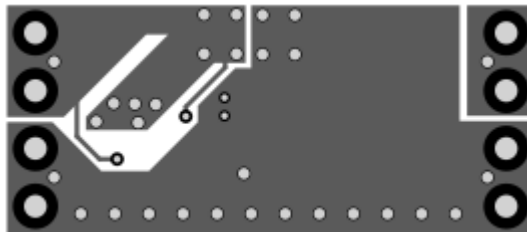


### A.2 Deska plošného spoje – top



Rozměr desky 23,8 x 10,7 [mm], měřítko M2:1

### A.3 Deska plošného spoje – bottom

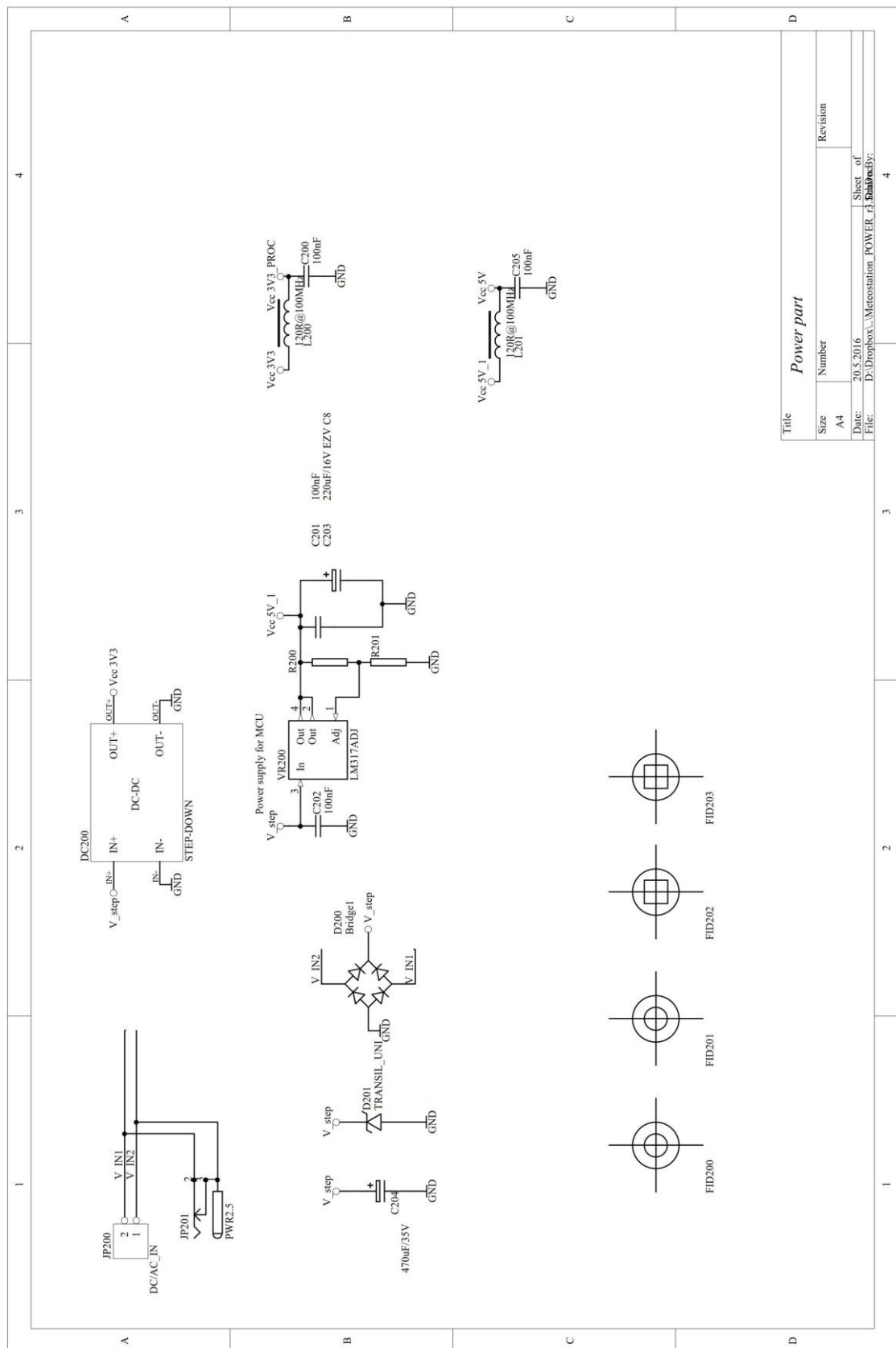


Rozměr desky 23,8 x 10,7 [mm], měřítko M2:1

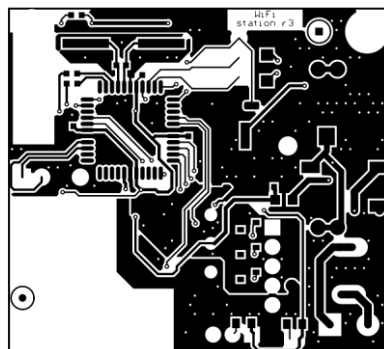




## B.2 Schéma zapojení zdrojové části

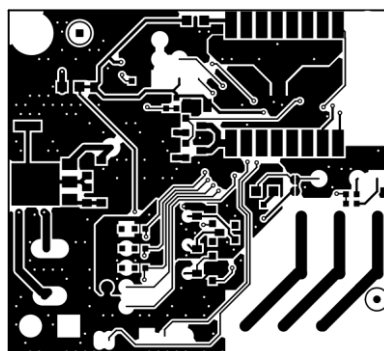


### B.3 Deska plošného spoje – top



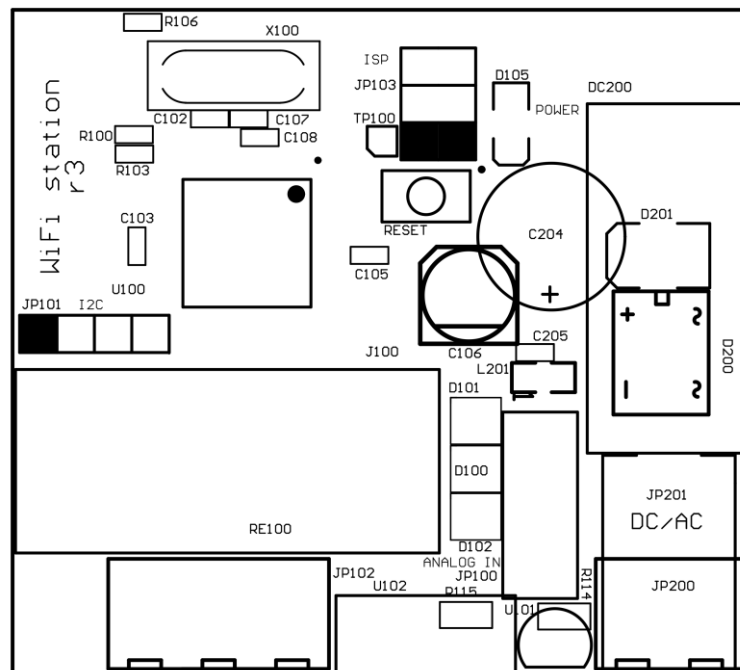
Rozměr desky 50 x 45 [mm], měřítko M1:1

### B.4 Deska plošného spoje – bottom

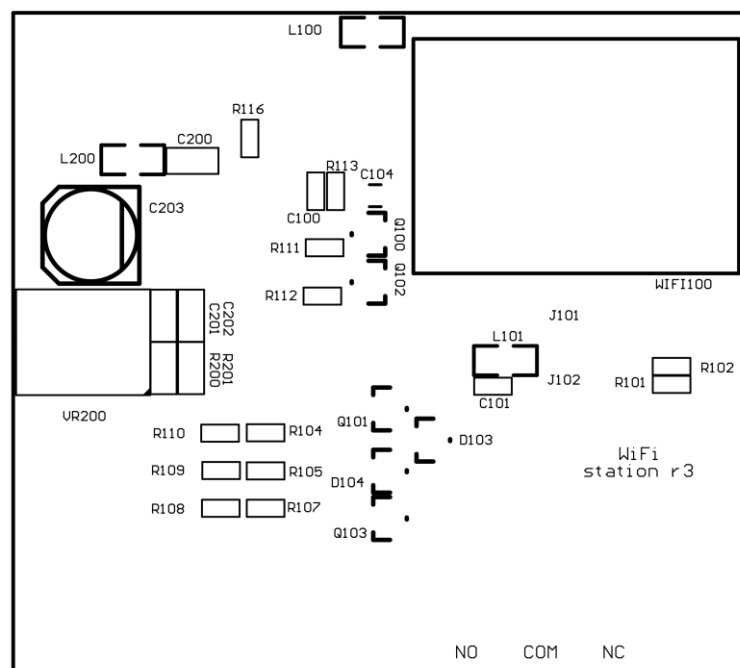


Rozměr desky 50 x 45 [mm], měřítko M1:1

## B.5 Osazovací výkres plošného spoje - top



## B.6 Osazovací výkres plošného spoje - bottom



## B.7 Seznam součástek

| Označení                     | Hodnota        | Pouzdro            | Množství |
|------------------------------|----------------|--------------------|----------|
| C100, C101, C205             | 100nF          | 0603               | 3        |
| C102, C107                   | 22p            | 0604               | 2        |
| C103, C105, C108             | 100n           | 0605               | 3        |
| C104                         | 4u7/6,3V       | 1206               | 1        |
| C106                         | 470uF/35V      | ALU C 6,3 mm       | 1        |
| C200, C201, C202             | 100nF          | 0805               | 3        |
| C203                         | 220uF/16V      | ALU C 6,3 mm       | 1        |
| C204                         | 470uF/35V      | CP 5,08 O10x16 V   | 1        |
| D100, D101, D102             | BZX84C3V6      | SOT-23             | 3        |
| D103, D104                   | BAS20          | SOT-24             | 2        |
| D105                         | POWER          | 1206               | 1        |
| D200                         | Bridge1        | DB-1S              | 1        |
| D201                         | TRANSIL_UNI    | SMB                | 1        |
| DC200                        | STEP-DOWN      | module             | 1        |
| FID200, FID201               | FID_CIRCLE     | FID-CIRCLE1MM      | 2        |
| FID202, FID203               | FID_SQUARE     | FID-SQUARE1MM      | 2        |
| J100                         | GND-AGND       | BYPASS-MINI-CLOSED | 1        |
| J101                         | 3V3            | BYPASS-MINI-CLOSED | 1        |
| J102                         | 5V             | BYPASS-MINI-CLOSED | 1        |
| JP100                        | 2X5            | HDR2X5A            | 1        |
| JP101                        | I2C            | HEADER-1X4         | 1        |
| JP102                        | RELE           | AK500-3            | 1        |
| JP103                        | PROG           | HEADER-2X3         | 1        |
| JP200                        | DC/AC_IN       | AK500-2            | 1        |
| JP201                        | PWR2.5         | KLD-0202           | 1        |
| L100, L101, L200, L201       | 120R@100MHz    | 0805               | 4        |
| Q100, Q102                   | BSS138         | SOT-23             | 2        |
| Q101, Q103                   | BCR512         | SOT-23             | 2        |
| R100, R103, R104, R105, R107 | 0R             | 0603               | 5        |
| R101, R102                   | 1K             | 0604               | 2        |
| R106                         | 10k            | 0605               | 1        |
| R108, R109, R110             | N/A            | 0606               | 3        |
| R111, R112                   | 0R11           | 0607               | 2        |
| R113                         | 100k           | 0608               | 1        |
| R114, R115                   | 4k7            | 0805               | 2        |
| R116                         | 270R           | 0805               | 1        |
| R200, R201                   |                | 0805               | 2        |
| RE100                        | Bistable relay | GL-K1-E            | 1        |
| S100                         | RESET          | SW-PB_3.5x6m_SMD   | 1        |
| TP100                        | SS             | TESTPIN            | 1        |
| U100                         | ATmega32A      | TQFP44             | 1        |
| U101                         | DS18B20        | TO-92A             | 1        |
| U102                         | DHT-11         | DHT-11             | 1        |
| VR200                        | LM317ADJ       | SOT223             | 1        |
| WIFI100                      | ESP8266-07     | ESP8266-12         | 1        |
| X100                         | 16MHz          | XTAL-RU49/4        | 1        |