

Mendelova univerzita v Brně
Provozně ekonomická fakulta

Návrh bezdrôtového ovládacieho modulu s ESP8266

Bakalárska práca

Vedúci práce:
Ing. Robert Rouš

Adam Földvári

Brno 2016

Podakovanie patrí pánovi Ing. Robertovi Roušovi za jeho cenné rady a za to, že mi umožnil túto prácu realizovať.

Čestné prehlásenie

Prehlasujem, že som túto prácu: **Návrh bezdrôtového ovládacieho modulu s ESP8266**

vypracoval samostatne a všetky použité pramene a informácie sú uvedené v zozname použitej literatúry. Súhlasím, aby moja práca bola zverejnená v súlade s § 47b zákona č. 111/1998 Sb., o vysokých školách v znení neskorších predpisov, a v súlade s platnou *Smernicou o zverejňovaní vysokoškolských záverečných prác*.

Som si vedomý, že sa na moju prácu vzťahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brne má právo na uzavretie licenčnej zmluvy a použitie tejto práce ako školského diela podľa § 60 odst. 1 Autorského zákona.

Ďalej sa zaväzujem, že pred spísaním licenčnej zmluvy o využitie diela inou osobou (subjektom) si vyžiadam písomné stanovisko univerzity o tom, že predmetná licenčná zmluva nie je v rozpore s oprávnenými záujmami univerzity a zaväzujem sa uhradiť prípadný príspevok na úhradu nákladov spojených so vznikom diela, a to až do ich skutočnej výšky.

Brno 16. mája 2016

.....

Abstract

Földvári, A. Design of wireless operating module using ESP8266. Bachelor thesis. Brno, 2016.

The bachelor thesis focuses on the design and implementation of wireless operating module and operating application. First part of thesis introduces reader to the used technology, the following section describes the construction and implementation of the operating module. The result of this thesis is a operating wireless WiFi module that collects data from sensors and control connected electrical device with relay through the web application.

Keywords

ESP8266, home automation, IoT, NodeMCU, MQTT

Abstrakt

Földvári, A. Návrh bezdrôtového ovládacieho modulu s ESP8266. Bakalárska práca. Brno, 2016.

Bakalárska práca je zameraná na tvorbu bezdrôtového ovládacieho modulu a ovládacej aplikácie. Prvá časť práce oboznamuje čitateľa o použitých technológiách, nasledujúca časť popisuje konštrukciu a implementáciu modulu. Výsledkom práce je ovládací bezdrôtový WiFi modul, ktorý zbiera dáta zo senzorov a ovláda pripojené elektrické zariadenie pomocou relé, prostredníctvom webovej aplikácie.

Kľúčové slová

ESP8266, domáca automatizácia, IoT, NodeMCU, MQTT

Obsah

1	Úvod a cieľ práce	7
1.1	Úvod	7
1.2	Cieľ práce	7
2	Analýza problému	9
2.1	Domáca automatizácia	9
2.2	Metodika riešenia	10
3	Teória	12
3.1	Mikrokontrolér	12
3.1.1	ESP-8266	12
3.1.2	ESP-8266 verzia 12F	13
3.1.3	ESPlorer	14
3.2	Prevodník USB-TTL Serial	14
3.3	Senzory a relé	15
3.3.1	Senzor teploty a vlhkosti	15
3.3.2	Senzor svetelnej intenzity	15
3.3.3	Relé	16
3.4	MQTT protokol	16
3.4.1	MQTT Broker	17
3.4.2	MqttSpy	18
3.5	DipTrace	18
3.6	Internet of Things	19
3.7	Webová aplikácia pre ovládanie a správu ESP modulov	20
3.7.1	Nette	20
3.7.2	MySQL	21
3.7.3	Paho.js	21
3.7.4	Highcharts.js	21
4	Návrh riešenia	22
4.1	Zapojenie zariadenia ESP8266	22
4.1.1	Bežná prevádzka	22
4.1.2	Vypalovanie nového firmvéru	22
4.2	Voľba firmvéru	24
4.2.1	Firmvér NodeMCU	24
4.3	Prvá komunikácia	25
4.3.1	AT príkazy	25
4.3.2	Vstavané funkcie firmvéru	25
4.4	Programovanie modulu	26
4.4.1	LUA	27
4.5	Návrh funkčného modulu	27
4.5.1	Regulátor napätia AMS1117-3.3V	28

4.5.2	Relé	30
4.5.3	Fotorezistor	31
4.6	Návrh programu obsluhujúceho modul	31
4.6.1	config.lua	33
4.6.2	setup.lua	33
4.6.3	app.lua	34
4.6.4	init.lua	35
4.6.5	Využitá pamäť	35
4.6.6	Stabilita modulu	36
4.7	Doska plošných spojov	36
4.8	Webová aplikácia	38
4.8.1	Užívateľské rozhranie	39
4.8.2	Princíp fungovania	39
5	Záver	41
5.1	Zhrnutie	41
5.2	Zhodnotenie výsledkov	41
5.3	Budúce využitie a možné rozšírenia	42
6	Literatúra	43
	Prílohy	46
A	Obrázky	47
A.1	ESP8266 v. 12F	47
A.2	Modul	47
A.3	Webová aplikácia	51
B	Priložené CD	53

1 Úvod a cieľ práce

1.1 Úvod

Túto prácu som si zvolil v spojitosti s odborom, ktorý študujem a mojimi osobnými záujmami. Inteligentne riadené domácnosti sa čoraz rýchlejšie dostávajú do popredia a povedomia mladých ľudí a začínajúcich rodín. Tempo života sa čoraz viac zrýchluje, ľudia sú čím ďalej, tým viac zaneprázdnení, preto využívajú výdobytky modernej doby k uľahčeniu, zrýchleniu životného tempa a k nadobudnutiu vyššej kontroly nad vzdialenými prostriedkami (napr. vzdialená regulácia teploty, inteligentné zásuvky a pod.). Inteligentná domácnosť, ako už z názvu vyplýva, bude domácnosť, ktorá je monitorovaná určitými prostriedkami a pomocou týchto prostriedkov budú môcť byť ovládané jej jednotlivé prvky, ako napríklad monitorovanie teploty a regulácia kúrenia v danej časti domu.

Touto problematikou sa pochopiteľne zaoberalo mnoho odborníkov predom mnou, taktiež existuje mnoho komerčných riešení, ktoré sú predražené a častokrát neprispôbitelné do praxe. Rozhodol som sa vydať cestou, ktorú vo svojej knihe podrobne popisuje kanadský elektrotechnik a programátor Tom Igoe, ktorý píše, ako pomocou MCU¹ – napríklad Arduino, je možné veci v domácnosti prispôbiť tak, aby medzi sebou dokázali určitým spôsobom komunikovať a tak vytvoriť funkčný celok (Tom Igoe, 2011). Pri riešení problémov, ktoré popisuje vo svojej knihe, je teoreticky možné použiť akýkoľvek mikrokontrolér. Ponuka rôznych zariadení od rôznych výrobcov je naozaj široká. Na trhu nájdeme rôzne riešenia, ktoré sa od seba líšia či už veľkosťou, výkonosťou, alebo počtom použiteľných vstupov a výstupov. Taktiež je dostupné široké spektrum príslušenstva, ako napríklad digitálne senzory, displeje, snímače, spínače, atď.

Pomocou týchto periférií je teda možné vytvoriť určitý prototyp zariadenia, ktorý bude plne zodpovedať požiadavkam reality.

1.2 Cieľ práce

Systémy riadenia inteligentných domov je možné zhrnúť pod pojem – domáca automatizácia. Tento pojem v svojej knihe popisuje Marco Schwartz, ako všetko čo si dokážeme predstaviť na ovládanie a automatizovanie domácnosti. (Marco Schwartz, 2014) Kniha obsahuje niekoľko zaujímavých nápadov na domácu automatizáciu pomocou platformy Arduino. Popisuje tvorbu niekoľkých samostatných systémov napríklad bezpečnostný systém a ich následné prepojenie a vytvorenie tzv. chytrého automatizovaného domu.

Táto práca sa zaoberá konštrukciou modulu pre monitorovanie rôznych častí domu a následným ovládaním určitého pripojeného prvku. Cieľom je navrhnúť jednoduchý modul, ktorý komunikuje prostredníctvom WiFi s riadiacim rozhraním

¹MCU – Micro Controller Unit

(webová aplikácia), modul ďalej obsahuje potrebnú výbavu senzorov, snímačov a relé. Táto jednotka bude obsahovať program, ktorý získava údaje z jednotlivých snímačov a je schopný komunikácie s riadiacim rozhraním. Riadiace rozhranie má za úlohu komunikovať so všetkými modulmi v dome a umožňovať pomocou grafického užívateľského rozhrania dom monitorovať a ovládať. Riadiace rozhranie teda nielen prijíma a zhromažďuje údaje, ale aj vysiela riadiace príkazy jednotlivým modulom, napríklad zopnúť relé, alebo zmerať a odoslať dáta zo senzorov.

Moduly budú postavené na platforme ESP-8266 a na komunikáciu budú využívať vstavaný WiFi modul, budú obsahovať senzory vlhkosti, teploty, svetla a relé na ovládanie pripojeného elektrického zariadenia.

2 Analýza problému

V tejto kapitole je analyzovaná problematika domácej automatizácie, ktorej sa táto práca úzko týka. Čitateľovi je priblížená problematika domácej automatizácie z pohľadu rôznych autorov a samotných spôsobov respektíve ponímania preberanej problematiky. Následne je čitateľ oboznámený s metodikou riešenia.

2.1 Domáca automatizácia

Automatizácia domácností má vo všeobecnosti niekoľko výhod. Umožňuje domácnosť monitorovať, ovládať a dokonca môže výrazne prispieť k zníženiu nákladov. V dnešnej modernej dobe je možné robiť veľmi veľa vecí na diaľku, ako napríklad nakupovanie prostredníctvom internetu, avšak keď je potrebné v dome pred odchodom pozhasínať, je stále potrebné skontrolovať všetky miestnosti, v ktorých sa svieti. Automatizovaná domácnosť nám môže ušetriť peniaze, napríklad aj za kúrenie. Ak sme odišli na dlhšiu dobu z domu preč a zabudli sme v niektorej z miestností vypnúť kúrenie a táto miestnosť disponuje inteligentným ovládacím modulom, je možné kúrenie vypnúť. Edward Soper v svojej knihe popisuje základy domácej automatizácie, pričom svoje príklady zariadení stavia na platforme firmy X10. Jedná sa o sústavu zásuvkových modulov, ktoré medzi sebou komunikujú prostredníctvom existujúcej kabeľáže elektrorozvodov. Užívateľ tak vlastne ovláda zariadenia pripojené v zásuvkách pomocou diaľkového ovládača, jednotlivé zásuvky sú navzájom prepojené a adresovateľné. Tým pádom je možné, napríklad rozsvetovať, alebo zhasínať pripojené lampy. K tomuto systému existujú rôzne rozšírenia, ako napríklad časovanie (rozsvetovanie svetiel počas večerných hodín, alebo zapnutie respektíve vypnutie elektrického kúrenia). (Edward Soper, 2005)

Podstatnou časťou v domácej automatizácii tvorí určitá forma siete, ktorá umožňuje zariadeniam medzi sebou komunikovať a svojim spôsobom tým prispieť k plneniu požiadavkov užívateľa. Z tejto ideí vychádza aj Tom Igoe vo svojej knihe – „Making things talk“, kde používa rôzne druhy sieťovej komunikácie medzi ovládanými modulmi, ako napríklad rádiové vlny, štandardnú ethernetovú lan sieť a v závere WiFi sieť. Na realizáciu projektov využíva mikrokontrolér Arduino UNO, ku ktorému pripája rôzne komunikačné „shields“ (RJ-45 shield, WiFi shield, Bluetooth shield,..). V knihe sú podrobne popísané riešenia problémov, či už zo stránky elektrotechnickej, alebo programátorskej, ďalej sú v nej vysvetlené podrobné vzťahy medzi jednotlivými súčasťami budovaných systémov. (Tom Igoe, 2011)

Z obdobných princípov vychádza aj Danny Briere vo svojej knihe s názvom „Smart home for dummies“, kde na začiatku popisuje, aké siete sa nachádzajú v našich domoch a ako ich môžeme využiť k tomu, aby sa náš domov stal lepším a pohodlnejším miestom na oddych a život. Zaoberá sa nielen ovládaním jednotlivých prvkov domácností, ale aj multimedializáciou (streamovaním videa a audia) a taktiež aj tvorbou bezpečnostného systému. Jeho kniha však pozostáva z použitia komerčných prvkov a produktov vyrobených a určených pre automatizáciu domá-

nosti, ktoré sú ale neprimerane drahé a tak prerábanie domu podľa tejto knihy vyjde na nemalú sumu peňazí. (Danny Briere, 2005)

2.2 Metodika riešenia

Práca je rozdelená na niekoľko častí. Prvá časť sa zaoberá rešeršou použitých technológií a vysvetlením základných princípov. V tejto časti budú popísané jednotlivé pojmy súvisiace s domácou automatizáciou pomocou MCU.

V ďalších častiach bude písané o praktickej konštrukcii prototypu modulu, naprogramovania programu, ktorý bude modul obsluhovať. Navrhnutie dosky plošných spojov v programe DipTrace a tvorbu jednoduchej ovládacej webovej aplikácie. V týchto častiach budú uvedené podrobné obrázky schém, rozpisy zapojenia a iné. Praktická časť tejto práce bude riešená následovným postupom:

1. Zoznámenie sa so zariadením ESP8266
2. Návrh prototypu modulu
3. Skonštruovanie prototypu modulu
4. Implementácia ovládacieho programu
5. Návrh dosky plošných spojov
6. Tvorba ovládacej webovej aplikácie

Výčet hlavných použitých technológií – hardvér:

- MCU ESP8266 v.12
- USB-TTL prevodník
- DHT11 – digitálny senzor teploty a vlhkosti
- fotorezistor
- elektromagnetické relé pre MCU (spínané pomocou tranzistora)
- regulátor napätia
- potenciometer, odpory, kondenzátory

Výčet hlavných softvérových nástrojov a technológií ktoré sú v práci použité:

- ESPlorer (verzia 0.2.0-rc2) – nástroj pre tvorbu a nahrávanie skriptov do ESP8266
- ESP flasher (verzia 0.1.0)– nástroj pre vypalovanie firmvéru do flash pamäte
- LUA NodeMCU firmware (verzia master build, 9 modules, float, 24.1.2016)
- MQTT – komunikačný protokol

- MqttSpy (verzia 0.4.0)– nástroj pre komunikáciu s modulom
- DipTrace (verzia 3.0)
- PHP framework Nette (verzia 2.3.6), MySQL (verzia 5.0.11), Paho.js (verzia 3.1), Highcharts (verzia 4.2.3)

V závere, poslednej časti tejto práce, bude písané o efektívnosti, využiteľnosti a použiteľnosti takéhoto modulárneho systému v praxi.

3 Teória

Táto kapitola sa zaoberá vysvetlením základných princípov, popisom samotných zariadení a použitých technológií nutných pre konštrukciu a implementáciu bezdrôtového modulu použitím zariadenia ESP8266 a ovládaciú webovú aplikáciu.

3.1 Mikrokontrolér

Mikrokontrolér je výpočtová jednotka, ktorá v sebe obsahuje niekoľko rôznych obvodov. Tieto obvody plnia úlohy bežiaceho programu, prostredníctvom sady inštrukcií. Srdcom MCU je CPU² teda hlavná výpočtová jednotka, ktorá vykonáva hlavné výpočty. Avšak hodnoty, ktoré má spočítavať, musia byť určitým spôsobom predpripravené, na čo slúžia registre. Vstupné registre pre vstupné dáta a výstupné registre pre výstupné dáta. Postup operácií v CPU určuje radič, ktorý taktiež určuje typ vykonávanej operácie a radí dáta do jednotlivých registrov. MCU taktiež obsahuje vnútorné pamäťové bunky, ktoré uchovávajú užívateľov program. Tento program na MCU beží v nekonečnom cykle. Vonkajšie pamäťové bunky slúžia k uchovávaniu dát bežiacim programom. V závislosti na zvolenej vnútornej pamäti poznáme viacero druhov MCU. (John Davies, 2008)

- programovateľné – pamäť mikrokontroléru je možné zmazať a nahráť nanovo
- neprogramovateľné – pamäť mikrokontroléru nie je možné prepísať, ani zmazať, program bol nahratý počas výrobného procesu mikročipu

MCU sa medzi sebou líšia rôznymi vlastnosťami, napríklad taktovacou frekvenciou CPU, šírkou spracovávaného slova (napríklad 8-bit, 16-bit, 32-bit), veľkosťou použiteľnej pamäte a iné. Vyhotovenie mikrokontrolérov je štandardne ako zaliaty mikročip, z ktorého vychádzajú kontakty, ktoré predstavujú konektivitu MCU jednotky s okolitými súčiastkami. Pomocou týchto pinov je mikročip napájaný, pripojený na vonkajšiu dátovú zbernicu a taktiež môže ovládať programovateľné vstupy respektíve výstupy. (John Davies, 2008)

Prvý mikrokontrolér sa objavil v roku 1971 a jednalo sa o MCU spoločnosti Intel presnejšie Intel 4004, od tohto roku mnoho spoločností vyrobilo rôzne typy a druhy mikrokontrolérov. Stali sa neodlúčiteľnou súčasťou modernej techniky a umožnili vyvinutie rôznych komplexných a zložitých systémov. (John Davies, 2008)

3.1.1 ESP-8266

ESP je mikrokontrolér ktorý pochádza z čínskej spoločnosti Espressif Systems, kombinuje v sebe WiFi modul s plnohodnotným programovateľným MCU. K dnešnému dňu existuje 12 verzií, ESP 1–12, ktoré sa od seba líšia rozmermi, pin-outom³, ty-

²CPU – Core Processing Unit

³pin-out – piny MCU, ktoré sú vyvedené na dosku plošných spojov

pom WiFi antény, veľkosťou pamäte a počtom GPIO⁴ pinov. Skladá sa z mikročipu Cadence Tensilica LX106, ktorý v sebe používa 32-bitové CPU bežiacie na frekvencii 80 MHz. Procesoru disponuje 64 KiB inštrukčnej RAM pamäte a 96 KiB dátovej RAM. Mikrokontrolér má k dispozícii externú QSPI⁵ flash pamäť a to v závislosti od verzie 512 KiB–4 MiB, taktiež v závislosti od verzie ESP sa líši počet vyvedených GPIO pinov na dosku plošných spojov, samotný mikročip ale disponuje 16timi GPIO pinmi. WiFi modul podporuje IEEE 802.11 b/g/n štandardy a je schopný naraz obsluhovať 5 TCP spojení. Samozrejmosťou je autentifikácia prostredníctvom WEP alebo WPA/WPA2. (Neil Kolban, 2015)

ESP8266 obsahuje taktiež jeden 10-bitový prevodník z analógového signálu na digitálny. Pracovné napätie je 3,3 V a jeho odber prúdu sa pohybuje v rozmedzí 0,5 uA–215 mA. Na doske plošných spojov sa v závislosti od verzie modulu nachádza buď vstavaná anténa, alebo U.FL konektor pre externú anténu. (Espressif Systems IOT Team, 2014)

Aplikácie pre ESP sa štandardne píše v jazyku C++, mimo iné, v jazyku C++ existujú knižnice, ktoré uľahčujú rôzne rutinné činnosti ako napríklad pripojenie a odpojenie sa od WiFi siete. Na trhu existujú alternatívy, ktoré umožňujú programovanie v iných jazykoch, či už MicroPython, alebo napríklad programovací jazyk LUA. Kód napísaný v programovacom prostredí na počítači sa skompiluje a prostredníctvom sériovej komunikácie sa vypáli do pamäte flash MCU, kde zostane zapísaný aj po odpojení zariadenia od elektrickej energie. V dnešnej dobe málokto výrobca bežných osobných počítačov poskytne sériový RS232 port, ktorý bol v minulosti štandardom v komunikácii medzi zariadeniami. Preto existujú tzv. USB – Serial konvertori, ktoré ako z ich názvu vypovedá, konvertujú komunikáciu z USB (Universal Serial Bus) na sériovú komunikáciu. (Espressif Systems IOT Team, 2014)

3.1.2 ESP-8266 verzia 12F

Verzia 12 modulu ESP je vhodná pre túto prácu, nakoľko disponuje zabudovanou anténou v PCB (doska plošných spojov), LED diódou signalizujúcou stav zariadenia, celkom 22 vyvedenými pinmi, z ktorých je 11 GPIO. Funkcionalita jednotlivých pinov je v tabuľke 1, ich vyobrazenie na doske PCB je možné vidieť v prílohách na strane 47, na obrázku 7. Piny 9 – 14 v mojej práci nevyužívam, preto sa im nebudem ďalej venovať.

Táto verzia ESP modulu má 4 MiB flash pamäte, ktoré sú plne k dispozícii pre účely tejto práce. Navyše má veľmi kompaktné rozmery 16 mm * 24 mm, podrobné rozvrhnutie rozmerov dosky mikrokontroléru je tiež možné vidieť v prílohách na strane 47, na obrázku 7. (Neil Kolban, 2015)

⁴GPIO – General Purpose Input Output

⁵QSPI – Quad Serial Input Output

Tabuľka 1: Popis pinov ESP8266 verzia 12

Pin	Popis
VCC	Vstupné napätie 3,3 V
GND	Uzemnenie
CH_PD	Čip zapnutý
REST	Reštart
ADC	Vstup do konvertoru z analóg na digitál
TXD	Vysielanie sériových dát
RXD	Prijímanie sériových dát
GPIO 13	GPIO pin
GPIO 12	GPIO pin
GPIO 14	GPIO pin
GPIO 16	GPIO pin
GPIO 4	GPIO pin
GPIO 5	GPIO pin
GPIO 0	GPIO pin, voľba boot režimu
GPIO 2	GPIO pin, voľba boot režimu
GPIO 15	GPIO pin, voľba boot režimu

3.1.3 ESPlorer

Jedná sa o vývojové prostredie pre ESP moduly, umožňuje programovanie v jazykoch LUA, MicroPython a podporuje AT príkazy. Toto IDE⁶ v sebe zahŕňa monitor sériovej komunikácie, editor pre programovanie a nahrávanie skriptov do modulu, rozhranie pre základnú AT komunikáciu. Ďalej ladiace nástroje ako napríklad – logovaciu okno, softvérový reset ESP8266, formátovanie pamäte, vypisovanie adresárovej a súborovej štruktúry, a iné. (Rui Santos, 2015)

Jedná sa o aplikáciu vyvinutú skupinou nadšencov ESP modulov, je naprogramovaná v jazyku Java a pre správne fungovanie potrebuje minimálne verziu 7 Java SE (Standard Edition).

3.2 Prevodník USB-TTL Serial

Toto zariadenie je v tejto práci veľmi podstatné, nakoľko ESP je programované pomocou TTL (Transistor-Transistor Logic) sériovej linky. Jedným z najznámejších výrobcov týchto prevodníkov je firma FTDI a ich súčasťou prevodník TTL-232R. Toto zariadenie poskytuje rozhranie medzi USB sériovou komunikáciou a komunikáciou sériovou na úrovni RS232 UART (Universal Asynchronous Receiver/Trans-

⁶IDE – Integrated Development Environment

mitter). Tento prevodník v sebe komponuje mikrokontrolér, ktorý zaistuje preklad signálu z USB na štandardnú sériovú linku. Je napájaný z USB portu počítača teda 5 V a je možné pomocou týchto 5 V napájať aj mikrokontrolér ako napríklad Arduino. Avšak v tejto práci je použité zariadenie ESP8266, ktorého pracovné napätie je 3,3 V. Preto je potrebné toto napätie znížiť na úroveň vhodnú pre ESP modul. Štandardne USB na UART–Serial prevodníky ponúkajú možnosť voľby výšky výstupného napätia a to v 2 hodnotách, 5 V a 3,3 V, problém nastáva pri maximálnom možnom odbere prúdu prostredníctvom prevodníku, ktorý je ako výrobca udáva od 75 mA–100 mA. (Future Technology Devices International Ltd, 2010)

Tieto hodnoty nemusia byť dostačujúce pre správnu funkcionality ESP modulu v určitých situáciách (slabý signál WiFi, veľa externých periférií, atď.). Preto je vhodné použiť oddelený regulátor s väčším maximálnym odberom prúdu a ESP teda nenapájať prostredníctvom prevodníku. Na druhej strane je táto možnosť bezpečnejšia, nakoľko možnosť poškodenia USB rozhrania počítača pri nesprávnom zapojení ESP modulu je o niečo nižšia. (Neil Kolban, 2015)

3.3 Senzory a relé

Táto podkapitola je zameraná na externé periférie modulu, ktoré slúžia k ovládaniu a zbieraniu dát – o vlhkosti, teplote a svetelnosti.

3.3.1 Senzor teploty a vlhkosti

Senzor teploty a vlhkosti je možné nájsť v rôznych prevedeniach, od analógových až po digitálne. ESP8266 verzia 12 používa digitálne, ale aj analógové senzory. Pomocou špeciálnych open-source knižníc je možné z nich spracovávať hodnoty. Senzor bude umiestnený mimo zariadenie, aby podával čo najvernejšie údaje reality. Najjednoduchšie z hľadiska softvérovej podpory je použiť senzor DHT-11, ktorý má rozsah meranej teploty od 0° C do 50° C a rozsah meranej vlhkosti od 20% do 90%. Rozsah pracovného napätia tohto senzoru je 3,3 V–5 V. Avšak pri pracovnom napätí 3,3 V je maximálna dĺžka vodičov od MCU k DHT senzoru, podľa dokumentácie, do 1 m. (Adafruit Industries, 2015)

3.3.2 Senzor svetelnej intenzity

ESP disponuje 10 bitovým prevodníkom analógového signálu na digitálny, čo sa dá využiť pri práci s analógovými senzormi. Ako senzor intenzity svetla je možné použiť štandardnú odporovú súčiastku – foto–rezistor. V tejto práci bude slúžiť na určenie úrovne svetla v miestnosti. Je potrebné si uvedomiť, že úroveň svetelnosti sa môže miestnosť od miestnosti líšiť. Preto je vhodné dať používateľovi možnosť nastaviť si citlivosť foto-senzoru podľa vlastných požiadavok. Toto bude realizované pomocou potenciometra, ktorý bude ovládať vstupné napätie prichádzajúce na fotorezistor a ADC prevodník bude snímať až výslednú hodnotu vychádzajúcu z fotorezistora.

3.3.3 Relé

Relé je jeden z najzákladnejších elektronických ovládacích členov. Skladá sa z vstupu a z časti, ktorá je ovládaná spínaním relátka. Súčasné elektromagnetické relé – uspôsobené pre MCU, používa logickú hodnotu pre ovládanie tranzistora, ktorý následne ovláda spínanie kladivka vnútri medzi jednotlivými kontaktami. Ak je na vstupe logická 1 (na vstupe je napätie), kladivko pritiahne elektromagnet. Ak zas 0 (na vstupe nie je napätie), kontakty sa nedotýkajú. Umožňuje 2 spôsoby zapojenia spínaného zariadenia NC (Normally Closed) a NO (Normally open). NC znamená, že ak je na logickom vstupe 0, výstupné kontakty – NC a Common sú spojené. V opačnom prípade, keď je na logickom vstupe 1, relátko sa zopne a prepojí výstupné kontakty NO a Common. Piny relé sú uvedené v nasledujúcom výpise. (Paul Horowitz, Winfield Hill, 2015)

- Vstupné piny elektomagnetického relé, uspôsobeného pre MCU:
 - Vin – vstupné napájanie elektromagnetu
 - Gnd – záporný pól
 - Log – Logický vstup
- Výstupné / ovládané piny:
 - NC – Normally closed, Log je 0 – NC a Common sú spojené
 - Common – spínaný pin medzi NC a NO
 - NO – Normally open, Log je 1 – NO a Common sú spojené

3.4 MQTT protokol

MQTT⁷ je protokol, ktorý vznikol v roku 1999 pre tzv. machine-to-machine komunikáciu. Jedná sa o protokol, ktorý umožňuje výmenu správ prostredníctvom existujúcej infraštruktúry internetu, využívajúc TCP/IP štandardov. MQTT protokol je nízko záťažový a umožňuje výmenu správ na veľké vzdialenosti vo veľmi krátkom čase a to aj na miestach, kde je prístup na internet výrazne obmedzený – či už dátovo, alebo rýchlostne. (Valerie Lampkin, Leonardo Olivera, 2012)

Podľa knihy „Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ Telemetry“ od spoločnosti IBM, MQTT so sebou prináša nasledujúce výhody. (Valerie Lampkin, Leonardo Olivera, 2012)

- posunutie hraníc podniku rozšírením konektivity – inteligentné zariadenia
- poskytuje škálu nastavení, ktoré sú optimalizované pre správy senzorov a vzdialených zariadení

⁷MQTT – Message Queuing Telemetry Transport

- údaje, ktoré sú odoslané a prijaté, sú vždy relevantné
- umožňuje širokú rozširiteľnosť a prepojitelnosť rôznych druhov zariadení rôznych firiem

Kým sa MQTT protokol nepoužíval, údaje sa vymieňali prostredníctvom HTTP request-ov a response-ov (požiadavka a odpoveď), čo nebolo príliš efektívne a spoľahlivé riešenie. MQTT protokol, tým ako je navrhnutý, minimalizuje náročnosť na výkon zariadenia, siete a taktiež aj samotné servery. Server slúži ako prostredník pri výmene MQTT správ. MQTT komunikácia funguje na princípe „publish message“ a „subscribe message“, v preklade ide o jednoduchý princíp zverejňovania a sledovania nových správ na MQTT serveri. MQTT server sa tiež nazýva MQTT broker, čo v preklade z angličtiny je možné chápať ako MQTT dohadzovač, respektíve sprostredkovateľ správ medzi zariadeniami.

MQTT protokol zaujal svoju pozíciu v segmente IoT⁸ sietí, jedná sa o siete, ktoré využívajú internet k výmene údajov zo senzorov, k prepojeniu zariadení, vozidiel, domov, atď. Projekt „internetu vecí“ umožňuje komplexnejšie prepojenie informatiky a človeka. Protokol MQTT teda prispieva k zlepšeniu komunikácie nielen medzi samotnými zariadeniami, ale aj medzi človekom a skupinou zariadení. (Valerie Lampkin, Leonardo Olivera, 2012)

V tejto práci bude MQTT protokol slúžiť ako sprostredkovateľ komunikácie medzi užívateľom (aplikáciou) a ESP modulmi. Údaje zo senzorov modulu a riadiace príkazy užívateľa budú doručované prostredníctvom tohto protokolu. Taktiež je potrebné využiť služby jedného z neplatených MQTT broker-ov, ktorý bude prostredníkom pri výmene správ.

3.4.1 MQTT Broker

Existuje mnoho typov MQTT sprostredkovateľov. Takýto server zabezpečujúci výmenu dát môže fungovať na lokálnej úrovni – zriadenie vlastného MQTT serveru, alebo na úrovni verejnej – využitie služieb tretích strán súbežne s ostatnými užívateľmi. Voľba jednej, alebo druhej možnosti závisí od veľkého množstva faktorov, ako napríklad – bezpečnosť, spoľahlivosť, rýchlosť, potreba vzdialeného ovládania cez internet, QoS, atď.

Medzi najznámejších verejných poskytovateľov takýchto služieb patria spoločnosti ako IBM, Mosquitto a Amazon. Spoločnosť Mosquitto disponuje aj verziou programu pre zriadenie vlastného lokálneho serveru. Jednotliví poskytovatelia sa od seba odlišujú rôznymi vlastnosťami, ako napríklad dynamické vznikanie a zanikanie tém – podľa stavov zariadení (uvediem na príklade). (Peter R. Egli, 2015)

Príklad: Zariadenie prejde z činného stavu do stavu nečinnosti, vytvorí na serveri novú tému s názvom nečinnosť, do ktorej bude pridávať údaje zo zariadenia v stave nečinnosti (porucha, respektíve chybové hlásenie). Po uvedení do činné-

⁸IoT – Internet of Things (Internet vecí)

ho stavu vznikne na serveri nová téma, do ktorej sa zverejňujú údaje zo senzorov zariadenia v stave činnom. Po určitom čase téma o nečinnosti zariadenia zanikne.

Taktiež môže byť umožnené zadanie tzv. „last will“ (poslednej vôle) určitému uzlu v sieti (v tomto prípade ESP modul). Táto posledná vôle môže byť napríklad správa do určitej témy na serveri o odpojení zariadenia zo siete spôsobeného napríklad výpadkom elektrickej energie. (Valerie Lampkin, Leonardo Olivera, 2012)

3.4.2 MqttSpy

Jedná sa o jednoduchú open source aplikáciu, ktorá slúži k monitorovaniu MQTT komunikácie na jednotlivých témach. Táto aplikácia je napísaná v programovacom jazyku Java, po vyplnení prihlasovacích údajov k MQTT sprostredkovateľovi je možné prezerat a posielat nové správy do ľubovolnej témy. (Kamil Baczkowicz, 2015)

MqttSpy je veľmi užitočný nástroj, ktorý ušetrí mnoho času pri vývoji a ladení programu pre ESP modul.

3.5 DipTrace

DipTrace je softvérový balík spoločnosti Novarm Limited, ktorý umožňuje návrh vlastných elektrotechnických schém, tvorbu vlastných súčiastok a ich symbolov, návrh dosiek plošných spojov. Obsahuje knižnicu s obrovským počtom existujúcich štandardných elektronických súčiastok, ako napríklad kondenzátory, odpory, konektory, diódy, základné mikročipy a mnoho ďalších. Každý symbol má priradenú svoju maticu, ktorá reprezentuje veľkosť, tvar a kontakty pre pájanie reálnej súčiastky. Táto matica potom reprezentuje danú súčiastku pri tvorbe návrhu dosky plošných spojov. (Novarm Limited, 2016)

Pri tvorbe plošného spoja je k dispozícii nástroj „3D Preview“, ktorý vyrenduje dosku a spolu s ňou aj súčiastky, ktoré majú priradený svoj 3D model. Pomocou tohto nástroja je teda možné predísť prípadným rozmerovým konfliktom na doske, ktoré bez 3D modelu nie je možné odhaliť. (Novarm Limited, 2016)

V tejto práci je použitý nasledujúci DipTrace softvér:

- DipTrace Component editor
- DipTrace Patern editor
- DipTrace Schematic
- DipTrace PCB layout

Všetky elektrické schémy v tejto práci sú vytvorené pomocou programu DipTrace Schematic.

DipTrace je platený softvér, existuje však „freeware“ vydanie, ktoré umožňuje použitie všetkých funkcií a knižníc, avšak s limitom dvoch signálových vrstiev, maximálnym počtom 300 pinov a nekomerčného použitia. (Novarm Limited, 2016)

3.6 Internet of Things

Pojem IoT, definuje dynamickú globálnu infraštruktúru siete pracujúcu s rôznymi objektami (inteligentný dom, auto, senzorový modul.), ktoré sú prostredníctvom takejto infraštruktúry schopné vymieňať dáta medzi sebou. Tento segment zažíva obrovský „boom“ od roku 2013, kedy sa začali vo veľkom používať zariadenia komunikujúce bezdrôtovo, alebo cez internet a vytvárajú tak funkčné celky. Príklady aplikácií, ktoré patria do segmentu IoT sa nachádzajú v nasledujúcom výčte. (Ovidiu Vermesan, Peter Friess, 2014)

- Chytré monitorovanie vody alebo plodín (kvalita vody, vlhkosť pôdy, a iné)
- Chytré monitorovanie zdravia (starostlivosť o športovcov alebo starších ľudí, kvalita spánku, a iné)
- Chytré bývanie (úspora energií, chytré spotrebiče, a iné)
- Chytré monitorovanie okolia (znečistenie vzduchu, detekcia požiarov, a iné)
- Chytrá výroba (monitorovanie efektivity produkčných liniek, sledovanie zvierat na farmách, a iné)
- Chytrá energia (monitorovanie a ovládanie efektivity fotovoltických článkov, monitorovanie prietoku vody, a iné)
- Chytré budovy (inteligentný požiarový alarm, kontrola vnútornej klímy)
- Chytrý transport a mobilita (platba NFC kartami, celková auto-diagnostika vozidiel, a iné)
- Chytrý priemysel (predikcia údržby a opravy strojov, „Machine2Machine“⁹ automatizácia, a iné)
- Chytré mesto (digitálny video monitoring ulíc, monitorovanie a riadenie premávky vozidiel, a iné)

Súčasný segment IoT firiem je pomerne široký. Je však potrebné brať na zreteľ fakt, že produkty IoT segmentu slúžia rôznym skupinám ľudí, ktorých je možné rozdeliť na tri skupiny:

- Jednotlivci
- Komunity ľudí (mestá, krajiny, spoločnosť ako taká)
- Podniky

Je zrejmé, že potreby jednotlivca sa budú výrazne odlišovať od potreby komunit ľudí alebo samotných podnikov, preto je na trhu široké spektrum spoločností, ktoré sa

⁹Machine to machine – komunikácia medzi dvoma a viacerými strojmi

zaoberajú rôznymi cieľovými skupinami. Navyše IoT segment so sebou prináša obrovský trh, na ktorom sa môže uchytiť ne jeden zaujímavý nápad. IoT však nie je len o prepojených zariadeniach, ale taktiež o prepojení ľudí s daným funkčným celkom, so snahou svojim spôsobom uľahčiť, sprijemniť, alebo zefektívniť život jednotlivca, alebo komunity. (Ovidiu Vermesan, Peter Friess, 2014)

3.7 Webová aplikácia pre ovládanie a správu ESP modulov

Webová aplikácia slúži ako komunikačné rozhranie človeka (používateľa) a jednotlivých modulov v domácnosti. Táto aplikácia zhromažďuje údaje z jednotlivých modulov, ukladá dáta do databázy a poskytuje užívateľovi priestor k prezeraniu získaných údajov, manipuláciu s jednotlivými modulmi a možnosť ovládať pripojené zariadenie prostredníctvom relé. Pre vývoj takejto aplikácie bolo zvolené webové prostredie, nakoľko je možné ho používať na rôznych platformách bez nutnosti vývoja samostatných aplikácií.

- PC s ľubovoľným operačným systémom a webovým prehliadačom
- Smartfón s operačným systémom Android alebo iOS
- Herné konzoly s webovým prehliadačom
- a tak ďalej

Pre vývoj hlavnej funkcionality webovej aplikácie bol zvolený český PHP framework Nette. Nakoľko framework Nette sám o sebe nepodporuje MQTT výmenu správ, bol použitý MQTT klient Paho vyvíjaný spoločnosťou Eclipse, ktorý umožňuje MQTT komunikáciu pre rôzne platformy a programovacie jazyky (JavaScript, Java, C/C++, .Net, a ďalšie). Dáta zo senzorov sú ukladané do MySQL databázy, pre prezentáciu nameraných dát je použitá Java Script-ová knižnica „Highcharts.js“, pomocou ktorej sú vykresľované grafy nameraných hodnôt pre každý pripojený modul. Pre dizajn webovej aplikácie bol použitý css framework Bootstrap, ktorému sa v tejto práci nebudem bližšie venovať.

Úspešným prepojením týchto webových technológií dôjde k vytvoreniu funkčnej webovej aplikácie pre správu ESP modulov.

3.7.1 Nette

PHP framework Nette je tretí najoblúbenejší framework na celom svete¹⁰, navyše je open source aj pre komerčné použitie. Výhoda použitia frameworku pri tvorbe webovej aplikácie spočíva v tom, že veľa práce týkajúcej sa napríklad bezpečnosti (cross-site scripting, prihlasovanie, a iné) odvedli už vývojári Nette frameworku a je teda možné priamo sa zamerať na vývoj aplikácie. Navyše aplikácia vytvorená pomocou frameworku môže byť jednoducho rozšíriteľná a upravovateľná. Nette

¹⁰prieskum magazínu SitePoint „Best PHP Framework for 2015“

používa a podporuje najmodernejšie webové technológie (napríklad AJAX/AJAJ, SEO, MVC, a ďalšie) a neustále sa vyvíja a posúva vpred. (Nette Foundation, 2015)

Nette framework obsahuje ladiaci nástroj Tracy, ktorý slúži k rýchlemu odhaleniu a odladeniu vzniknutej chyby, čo je pri vývoji webových aplikácií určite veľké plus. (Nette Foundation, 2008)

3.7.2 MySQL

MySQL je vo svete najpopulárnejší „open source“ databázovo relačný systém správy dát. Podporuje štandardnú syntax jazyka SQL, ako grafické užívateľské rozhranie sa najčastejšie používa prostredie aplikácie phpMyAdmin, nakoľko samotné MySQL GUI nedisponuje. (Jack Potte, 2014)

3.7.3 Paho.js

Ako bolo spomenuté komunikačný MQTT klient Paho je vyvíjaný spoločnosťou Eclipse. Tento klient je kľúčový pri získavaní správ od modulov z MQTT brokeru do webovej aplikácie. Pri spustení webovej aplikácie sa tento java script-ový klient prihlási prostredníctvom WebSocket na MQTT broker do vybraných tém a následne je užívateľovi umožnené komunikovať s modulmi. Pomocou tohto JavaScript-u sú vysielané požiadavky na dáta zo senzorov a na prepnutie stavu relátka, naopak prijímané správy su spracovávané a následne sa na ne určitým spôsobom reaguje (napríklad zanesenie dát do databázy). (Eclipse Foundation, 2015)

3.7.4 Highcharts.js

Táto Java script-ová knižnica umožňuje tvorbu rôznych typov grafov. Pre nekomerčné účely je kompletne zdarma. Medzi jednu z najväčších výhod patrí kompatibilita so zariadeniami, grafy sa vykresľujú na všetkých moderných zariadeniach (tablet, smartfón, a ďalšie) a ich internetových prehliadačoch. Umožňujú dynamické pre-kresľovanie dát, taktiež je možné výsledný graf vytlačiť, alebo vyexportovať ako obrázok, čo sa môže použiť napríklad pri mesačnej sumarizácii nákladov na vedenie domácnosti. (Highcharts, 2016)

4 Návrh riešenia

V tejto kapitole je popísané praktické riešenie problematiky práce. Zaoberá sa návrhom prototypu zariadenia a jeho zapojením, ďalej implementáciou ovládacieho programu samotných modulov, návrhom dosky plošných spojov modulu a ovládacej webovej aplikácie v PHP frameworku Nette.

4.1 Zapojenie zariadenia ESP8266

Mikrokontrolér, ktorý je použitý v zariadení ESP8266 v.12, rozlišuje dva spôsoby zapojenia vo vývojárskom režime. Prvé zapojenie je zapojenie v bežnej prevádzke, kedy je možné priamo nahrávať naprogramované skripty do pamäte modulu.

Druhé zapojenie sa používa pri vypalovaní nového firmvéru do mikrokontroléru. Jedná sa teda o manuálny výber bootovacej možnosti modulu.

4.1.1 Bežná prevádzka

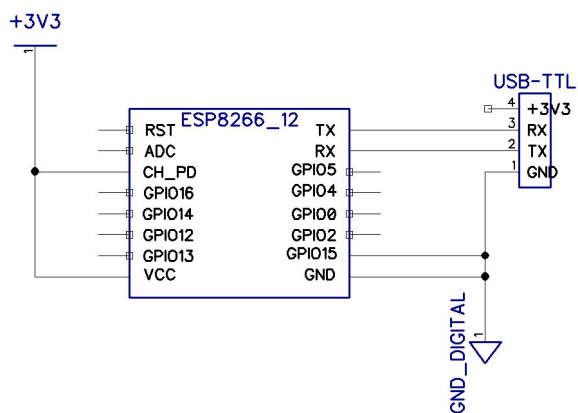
V bežnej prevádzke vo vývojárskom režime je pripojený USB-TTL prevodník na zariadenie ESP. USB prevodník má primárne 4 piny, dva tvorí napájanie (+3,3 V a GND) a dva piny zaberá sériová komunikácia, konkrétne piny Rx (Receiver) a Tx (Transmitter). Zapojenie je nasledovné: napájanie prevodníku sa pripája na vstupné napájanie modulu a zem (+3,3 V na Vcc a GND na GND), sériová komunikácia z prevodníku sa pripája na kríž so sériovou komunikáciou modulu (Tx na Rx a Rx na Tx). Ďalej je potrebné pripojiť na vetvu 3,3 V pin CH_PD (chip powered) modulu, ktorý uvedie mikrokontrolér do bežnej prevádzky. Taktiež je potrebné pripojiť pin GPIO15 na GND, čím sa zvolí boot v bežnej prevádzke. Prehľadné zapojenie je vyobrazené v tabuľke 2 a taktiež v schéme zapojenia na obrázku 1.

Tabuľka 2: Pripojenie pinov v bežnej prevádzke

USB prevodník	Vcc	GND	CH_PD	GPIO 15	Tx	Rx
+3,3 V	X		X			
Tx						X
Rx					X	
GND		X		X		

4.1.2 Vypalovanie nového firmvéru

Pri tomto zapojení je nutné pripojiť prevodník tak, ako pri bežnej prevádzke, to znamená +3,3 V na Vcc, GND na GND, Tx na Rx a Rx na Tx. Navyše pri tomto zapojení je nutné modulu oznámiť, že sa čip bude prepaľovať, preto sa nemá bootovať firmvér. GPIO 15 sa pripája na zem, alebo sa necháva nepripojený, GPIO 0 sa pripája na zem a GPIO 2 na 3,3 V. V tomto režime sa celá pamäť mikrokontroléru

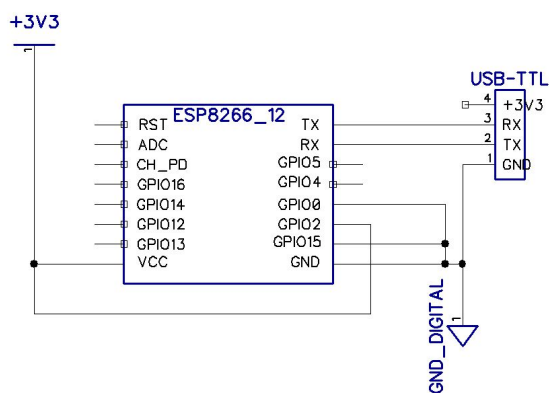


Obrázok 1: Zapojenie pri bežnej prevádzke ESP

zmaže a nahrá sa do nej zvolený firmvér. Prehľadné zapojenie je v tabuľke 3 a na obrázku 2.

Tabuľka 3: Pripojenie pinov pri vypaľovaní firmvéru

USB prevodník	Vcc	GND	GPIO 15	GPIO 0	GPIO 2	Tx	Rx
+3,3 V	X				X		
Tx							X
Rx						X	
GND		X	X	X			



Obrázok 2: Zapojenie pri vypaľovaní firmvéru do ESP

4.2 Voľba firmvéru

K dispozícii je niekoľko firmvérov, ktoré sú od seba pomerne odlišné. Líšia sa použitím rôznych druhov kompilátorov skriptov, druhmi použitých knižníc a vstavaných funkcií, taktiež sa líšia svojou stabilitou, odolnosťou voči rušeniu a podobne. (Jens Ch. Brynildsen, 2015) V mojej práci používam firmvér spoločnosti NodeMCU, ktorý podporuje programovanie skriptov v jazyku LUA, disponuje dostatočne širokou škálou knižníc a vstavaných funkcií. Firmvér firmy Espressif IoT mal v mojom prípade problém s časovaním CPU na jednom z troch modulov, čo spôsobovalo náhodné reštartovanie počas behu programu v cykle. Tento firmvér zároveň nepodporoval nahrávanie jednotlivých súborov do ESP modulu počas normálnej prevádzky, ale bolo nutné firmvér vypáliť znova spolu so zdrojovým kódom programu. Firmvér spoločnosti NodeMCU umožňuje nahrávanie skriptov počas zapojenia v bežnej prevádzke.

4.2.1 Firmvér NodeMCU

Spoločnosť NodeMCU ponúka možnosť zostaviť si firmvér podľa svojich vlastných potrieb. Ponúka na výber knižnice, ktoré sa zahrnú do binárneho súboru firmvéru, ktorý sa vypáli do flash pamäte MCU. Je teda možné nepotrebnú funkcionálnosť modulu vynechať a ušetriť tým miesto pre premenné a beh samotného programu. Taktiež je k dispozícii možnosť voľby pre výber celočíselného alebo firmvéru s desatinným rozvojom, výber závisí na tom, s akými hodnotami sa bude pracovať. V tejto práci bol použitý firmvér s desatinným rozvojom ¹¹. Skripty je možné písať v programovacom jazyku LUA, firmvér ďalej podporuje vytvorenie jedného TCP serveru a nadviazanie 5-tich TCP spojení. Výhodou tohto firmvéru je podpora nahrávania skriptov za behu v normálnom režime, bez nutnosti zmeny konfigurácie pripojenia. (NodeMCU, 2015)

K firmvéru je k dispozícii užitočný nástroj s názvom ESPlorer ¹², ktorý v sebe kombinuje monitor sériovej komunikácie, IDE pre kompiláciu a nahrávanie LUA skriptov do ESP modulu. Tento nástroj umožňuje rôzne nastavenia, ako napríklad rýchlosť sériovej komunikácie, nastavenie a zistenie IP adresy, zistenie MAC adresy, softvérový reštart modulu a iné. Taktiež umožňuje kontrolu verzie a aktuálnosti firmvéru s aktuálnou verziou. (Rui Santos, 2015)

Ďalší užitočný softvér firmy NodeMCU je ESP8266 flasher, ako z názvu vyplýva, je nástroj, ktorý slúži k extrahovaniu binárneho súboru firmvéru a jeho následné vypálenie do mikrokontroléru. Má prehľadné GUI a dokáže informovať o vzniknutých chybách počas procesu vypalovania firmvéru.

¹¹použitý firmvér – NodeMCU Master build, 9 modules, float, 24.1.2016

¹²ESPlorer – dostupné z: <https://github.com/4refr0nt/ESPlorer>

4.3 Prvá komunikácia

Prvotnú komunikáciu s modulom som realizoval prostredníctvom monitoru sériového portu (napr. PuTTY). Ak je modul správne zapojený v bežnom režime prevádzky a je nastavená správna rýchlosť komunikácie – v mojom prípade 9600 Baud-ov (rýchlosť sa líši v závislosti od verzie a firmvéru modulu), je možné prostredníctvom sériového monitoru osloviť modul pomocou AT (attention) príkazov. Ak je s modulom všetko v poriadku, na odoslaný príkaz „AT“ odpovie správou „OK“. Pomocou AT príkazov je možné modul pripojiť na WiFi, alebo WiFi sieť vytvoriť, taktiež je možné zisťovať rôzne informácie, napríklad o výrobcovi, alebo verzii firmvéru, ktorou modul disponuje. (Espressif Systems IOT Team, 2016)

4.3.1 AT príkazy

AT, alebo tiež Hayesové príkazy, sú príkazy, pomocou ktorých sa upútava pozornosť modulu. Pochádzajú z anglického slova – attention, boli vyvinuté pre komunikáciu s Hayesovými modemami v roku 1981. Príkazy, sú krátke reťazce, ktoré väčšinou obsahujú slovo AT a plnia rôzne funkcie. AT príkazy pre ESP modul sa čiastočne odlišujú od tých pôvodných modemových. Niekoľko príkazov však ostalo nezmenených (overenie stavu zariadenia, zistenie verzie firmvéru). (LSR, 2015)

Pre uskutočnenie takejto komunikácie s modulom je nutné použiť určitý komunikačný kanál, v tomto prípade to je sériová linka sprostredkovaná USB prevodníkom a sériovým monitorom. Pri nadväzovaní komunikácie je nutné dbať na správne nastavenie prenosovej rýchlosti. Prenosová rýchlosť je nastavená na firmvéri ESP modulu, bez prvotnej komunikácie ju nie je možné zmeniť, je teda potrebné rýchlosťou prispôbiť sériový monitor a samotný USB prevodník.

AT príkazy pre ESP modul sa delia na niekoľko druhov – základné, WiFi príkazy a TCP/IP príkazy. Základné príkazy overujú funkčnosť modulu a poskytujú základné informácie o firmvéri. WiFi príkazy umožňujú meniť režimy fungovania (WiFi stanica, prístupový bod alebo oboje súčasne), prezerat dostupné siete a pripájať sa k nim. TCP/IP AT príkazy umožňujú vytvárať pripojenia, posielat dáta, alebo overovat funkčnosť existujúcich spojení.

4.3.2 Vstavané funkcie firmvéru

Vstavané funkcie firmvéru sa používajú v programovaných skriptoch. Slúžia podobne ako AT príkazy, umožňujú využívat ich funkcionality za behu zariadenia bez nutnosti komunikácie cez sériovú linku – prostredníctvom vytvoreného skriptu. Vstavané funkcie sa taktiež dajú rozdelit na niekoľko druhov – základné, funkcie obsluhujúce WiFi a funkcie, ktorých knižnice obsahuje firmvér. Medzi základné funkcie samozrejme patrí overenie stavu zariadenia, vypísanie dostupných informácií (verzia firmvéru, dátum vypálenia, čas od posledného reštartu, a iné), alebo softvérové reštartovanie zariadenia. Funkcie obsluhujúce WiFi, podobne ako pri AT príkazoch,

majú obdobnú funkcionálnosť – umožňujú pripojenie, respektíve vytvorenie siete, nastavenie rôznych parametrov prístupového bodu a iné.

Funkcie ktorých volanie je podmienené existenciou daných knižníc vo firmvéri, je široké spektrum. Môže sa jednať napríklad o knižnicu GPIO, ktorá umožňuje ovládať piny a nastavovať rôzne režimy GPIO pinov (vstup/výstup, high/low), alebo knižnicu HTTP, ktorá umožňuje vytvoriť HTTP server, alebo používať metódy GET, POST, PUT, atď. Taktiež sa môže jednať o knižnicu určitej rady senzorov, napríklad DHT, ktorá po pripojení senzoru z rady DHT umožňuje čítať dáta z takýchto senzorov. K dispozícii je pomerne široká škála knižníc, ktorú je možné využívať a zahrnúť do zostavenia požadovaného firmvéru. Takéto knižnice sú ihneď k dispozícii, nie je teda nutné knižnicu importovať, alebo prepájať spolu s naprogramovaným skriptom. (NodeMCU, 2015)

V mojej práci sú pre mňa obzvlášť dôležité knižnice GPIO, MQTT, DHT a ADC. Samozrejmosťou sú povinné knižnice ako node, file, WIFI, tmr a UART. Príklad použitia vstavanej funkcie je v nasledujúcom kóde.

Kód 1: Lokálna funkcia pre overenie pridelenia IP adresy

```
--lokalna funkcia overi ci je pridelenia IP
-- ak ano vypise detaily, ak nie caka na pridelenie
local function ma_ip()
    if wifi.sta.getip() == nil then --ip nepridelenia
        print( 'Warning_SETUP_ma_ip():_ ' ..
              'IP_adresa_nebola_pridelena,_cakam...' )
    else
        tmr.stop(1)
        print('ID_modulu: '..node.chipid())
        print('MAC: '..wifi.sta.getmac())
        print('Pripojene_k_sieti:_ '..config.SSID['ssid'])
        print('IP: '..wifi.sta.getip())
        app.start()
    end
end
```

4.4 Programovanie modulu

Ako bolo spomínané, modul je programovaný prostredníctvom sériovej linky. Firmvér NodeMCU umožňuje nahrávanie programu za behu modulu a teda nie je nutné modul prepájať do režimu pre vypaľovanie firmvéru a spolu s kódom nahrávať aj firmvér. Program sa do modulu nahráva ako súbor, respektíve skript programovacieho jazyku LUA. Postup nahratia programu do modulu je nasledovný:

1. uloženie súboru
2. zmazanie predchádzajúcej kópie

3. odoslanie súboru do modulu ako parameter funkcie *makefile()*
4. spustenie súboru v module ako *dofile()*
5. kompilácia skriptu v module

V module je možné mať uložených niekoľko skriptov, navyše je možné spustiť niekoľko skriptov súbežne, ale treba brať ohľad na veľkosť pamäte RAM, nakoľko pri jej vyčerpaní dôjde k reštartovaniu modulu.

Inicializačný skript – teda skript, ktorý sa spustí, po spustení modulu sa vytvorí tým, že takýto skript sa pomenuje kľúčovým slovom *init*, teda „*init.lua*“. Skript s takýmto názvom sa spustí následne po načítaní firmvéru po štarte mikrokontroléru.

4.4.1 LUA

Programovací jazyk LUA vznikol v roku 1993, bol určený k rozširovaniu existujúcich aplikácií a k tvorbe skriptov. Jedná sa o odľahčený, imperatívny a procedurálny programovací jazyk, ktorého použitie je veľmi široké. Používa sa pri práci s animáciami a grafickými operáciami v hrách, ďalej tiež pre ovládanie robotov a robotických ramien. (Roberto Ierusalimschy, 2006)

Lua však podporuje malé množstvo dátových typov, konkrétne sa jedná o boolean, číselný dátový typ float a reťazec znakov. Zvyšné dátové štruktúry ako sú zoznamy, záznamy, polia, množiny-sú reprezentované natívnou dátovou štruktúrou-tabuľkou. V podstate sa jedná o heterogénne asociatívne pole, ktoré je možné implementovať tak, aby sa správalo ako jedna zo spomínaných dátových štruktúr. (Roberto Ierusalimschy, 2006)

Skripty naprogramované v programovacom jazyku LUA, nie sú interpretované priamo, ale sú kompilované do tzv. bytecode. Tento bytecode sa následne spúšťa na virtuálnom stroji (LUA virtual machine) v ESP module. Zvyčajne sa tento bytecode vytvára za behu skriptu, je však možné skript skompilovať tzv. offline kompiláciou, ktorá vytvorí bytecode súbor. Takýto pred kompilovaný súbor, môže viesť k zvýšeniu výkonu a zmenšeniu priestoru zaberanej pamäte RAM. (Roberto Ierusalimschy, 2006)

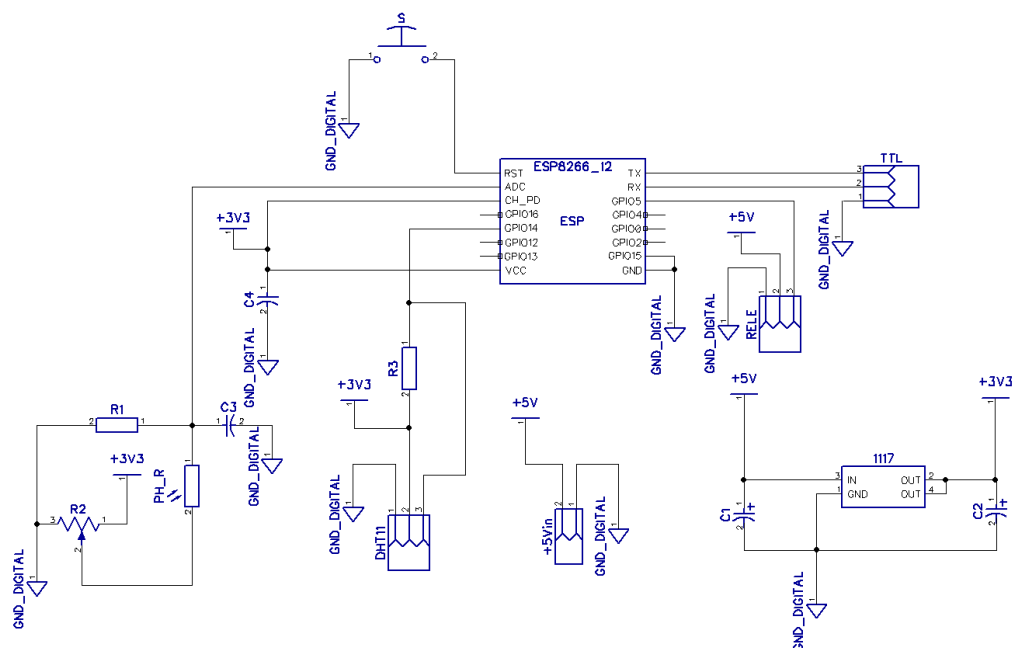
4.5 Návrh funkčného modulu

Táto časť práce sa venuje navrhnutiu funkčného modulu prostredníctvom MCU ESP8266 v.12-F a jednotlivých periférií. Jedná sa o:

- regulátor napätia AMS1117-3.3V
- ESP8266 v.12-F
- senzor teploty a vlhkosti – DHT11
- senzor úrovne svetelnosti – fotorezistor

- relé pre ovládanie pripojeného elektrického zariadenia

Schéma zapojenia modulu je vyobrazená na obrázku 3.



Obrázok 3: Vysledná schéma modulu

4.5.1 Regulátor napätia AMS1117-3.3V

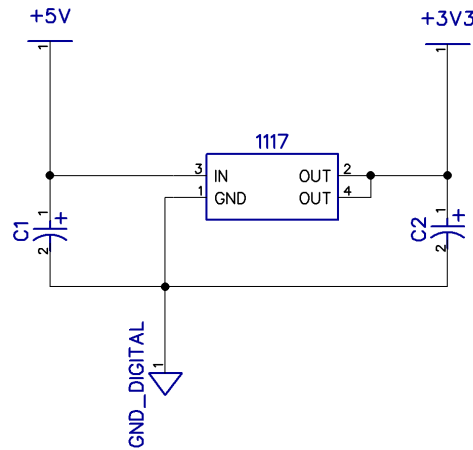
Pre túto prácu bolo potrebné navrhnuť regulátor vstupného napätia modulu z niekoľkých dôvodov. V praxi (najmä v domácnostiach) sa až tak často nevyskytujú zdroje s napätím 3,3 V, častejšie sa nájdu zdroje s hodnotou 5 V (napríklad nabíjačky s USB výstupom). Ďalším z dôvodov je cenová dostupnosť 5 V zdrojov v rôznych prevedeniach či už ako USB nabíjačky, alebo ako moduly pripevné na DIN lištu¹³ v elektrických rozvodoch. Vstupná hodnota napätia modulu bola teda ustanovená na 5 V, avšak pracovné napätie ESP8266 je 3,3 V, preto je nutné toto napätie regulovať.

Pre reguláciu z 5 V na 3,3 V, bol zvolený regulátor AMS1117-3.3 V v SMD prevedení SOT-223¹⁴, ktorý má maximálnu toleranciu vstupného napätia až 15 V. Pre správnu funkcionálnu a stabilitu regulovaného napätia je nutné použitie vstupného a výstupného kondenzátora. V dokumentácii regulátoru AMS1117 je uvedené, že je nutné použiť kondenzátory s minimálnou hodnotou 22 μF , v praxi sa však ukázalo

¹³TS 35 DIN lišta – podľa ČSN EN 60715 má tvar U, šírku 35 mm a hĺbku 7,5 mm

¹⁴SOT-223 – SMD prevedenie určené pre dosky plošných spojov – 4 piny

že je lepšie použiť hodnotu aspoň dvojnásobnú. Boli použité hodnoty $47\ \mu\text{F}$ 25V, ktoré stačia na stabilný chod regulátora bez napätového šumu. AMS1117 dodáva maximálny prúd 1 A, čo pre funkcionality modulu a prípadné spínanie relé úplne stačí. (Advanced Monolithic Systems, 2010) Schéma napätového regulátora je na obrázku 4.



Obrázok 4: Schéma regulátora napätia AMS1117-3.3 V

Z meraní odberu prúdu modulu vyšlo, že súčasná celková konfigurácia spotrebovávajú 70–100 mA. V dokumentácii ESP modulu je uvedené, že maximálny odber MCU je 215 mA. Samotný ESP modul spotrebovávajú v momentálnej konfigurácii 50 mA, predpokladaný maximálny prúd, ktorý bude dodávať regulátor do celého obvodu, sa teda bude pohybovať maximálne do hodnoty 300 mA. V dokumentácii k regulátoru AMS1117-3.3 V je uvedené, že každý vyprodukovaný 1 W zvýši teplotu o 55°C (Θ_{jc}). Je teda potrebné vypočítať aký výkon bude regulátor dodávať, toto je možné vypočítať zo vzťahu $P = U_{drop} I_{max}$. Kde P je výkon, U_{drop} je napätový rozdiel medzi vstupným a výstupným napätím regulátora a I_{max} je maximálny prúd, ktorý bude obvodom pretekať.

$$\begin{aligned}
 U_{in} &= 5\text{ V}; U_{out} = 3,3\text{ V} \\
 U_{drop} &= U_{in} - U_{out} = 1,7\text{ V} \\
 I_{max} &= 300\text{ mA} = 0,3\text{ A} \\
 \Theta_{jc} &= 55^\circ\text{C/W} \\
 P_{max} &= U_{drop} I_{max} \\
 P_{max} &= 1,7\text{ V} * 0,3\text{ A} = 0,51\text{ W} \\
 T_n &= P_{max} \Theta_{jc} = 28,05^\circ\text{C}
 \end{aligned} \tag{4.1}$$

Z výpočtu 4.1 vychádza, že maximálny teplotný nárast (T_n), voči okolitému prostrediu bude $28,05^\circ\text{C}$. V reále sa tak táto maximálna hodnota bude v súčte s izbovou

teplotou pohybovať približne na úrovni 50° C. Vyžiarené teplo je však možné čiastočne odvieť na zadnú stranu dosky plošných spojov a tým regulátor čiastočne ochladiť.

4.5.2 Relé

Pri výbere vhodného relé, je potrebné dbať na to, že samotný elektromagnet v relé by nemal byť priamo napájaný GPIO pinom. Ako výrobca mikročipu uvádza, maximálny odber prúdu na jeden GPIO pin je 12 mA, čo je pre zopnutie elektromagnetu nedostatočné a mohlo by dôjsť k poškodeniu samotného mikrokontroléru a jeho periférií. (Espressif Systems IOT Team, 2014)

Existuje niekoľko možností, ako tento problém vyriešiť, z ktorých sú práve dve najvhodnejšie.

Ako prvé riešenie, je použitie tranzistoru a externého zdroja pre spínanie elektromagnetu v relátku. V tomto prípade mikro kontrolér ovláda tranzistor ku ktorému je pripojené externé napájanie, tranzistor v tomto prípade funguje, ako spínač pre napájanie relé.

Druhé riešenie je použitie SSR¹⁵ relé, jedná sa o polovodičový prvok pre spínanie pripojeného zariadenia. Na rozdiel od štandardného elektromagnetického relé, SSR nemá žiadne mechanické časti, ktoré by ovládali spínanie kontaktov pripojeného zariadenia. Čo so sebou prináša niekoľko výhod, tichá prevádzka – na rozdiel od elektromagnetického relé, ktorého spínanie sprevádza tradičný klikací zvuk. Vyšší počet spínacích cyklov, ktorý sa taktiež odvíja od toho, že SSR neobsahuje pohyblivé mechanické časti ale elektronický obvod, ktorý ovláda vonkajšie svorky. Vyššia spínacia rýchlosť – taktiež sa odvíja od absencie mechanických častí. Medzi ďalšie výhody patrí menší rozmer voči štandardnému relé alebo stykaču pri zachovaní spínaného výkonu. SSR má však aj radu nevýhod, medzi ktoré sa radí napríklad úbytok napätia na spínanom zariadení. Navyše v niektorých prípadoch je potrebné SSR prvky chladiť. Štandardne sa odber prúdu SSR prvkov pohybuje od 5-15 mA, závisí však na konkrétnych parametroch SSR relé. (Paul Horowitz, Winfield Hill, 2015)

Výsledný modul, respektíve jeho PCB prevedenie, je uspôsobené pre použitie oboch spomínaných riešení. Elektromagnetické relé však bude musieť byť vybavené možnosťou externého napájania a logickým vstupom privedeným na tranzistor pre zaistenie čo najmenšieho odberu prúdu na daný GPIO pin. Inými slovami, takéto elektromagnetické relé musí byť uspôsobené pre použitie s MCU.

V tejto práci je použité elektromagnetické relé spínané pomocou tranzistora z externého zdroja napätia 5 V. Maximálny prúd ktorý môže toto relé spínať je 10 A, na štandardných elektrických rozvodoch 240 V.

¹⁵SSR – Solid State Relay

4.5.3 Fotorezistor

Fotorezistor¹⁶, je súčiastka ktorej rezistivita je závislá a premenlivá na základe úrovne svetelnej intenzity. Rezistivita tejto súčiastky sa v úplnej tme spravidla pohybuje v rádoch mega ohmov, avšak pri dopade fotónov respektíve paprskov svetla na svetlo citlivú plochu sa rezistivita môže pohybovať v stovkách ohmov.

Fotorezistor pracuje na fotoelektrickom princípe. Jedná sa o svetelné a elektrické javy, pri ktorých sa materiály na ktoré dopadá svetlo stávajú viac vodivé.

Existuje niekoľko druhov fotorezistorov, odlišujú sa od seba fyzickou veľkosťou, veľkosťou odporu, alebo zložením foto rezistívneho materiálu. (Paul Horowitz, Winfield Hill, 2015)

V tejto práci je použitý fotorezistor s rezistivitou v tme približne 2.5–3 MΩ. Je zapojený prostredníctvom potenciometru viď obrázok 3, ktorý slúži na manuálne – jemné doladenie citlivosti fotorezistoru.

4.6 Návrh programu obsluhujúceho modul

Táto časť práce sa zaoberá návrhom programu v programovacom jazyku LUA, ktorý riadi modul. Tento program sa rozdeľuje na niekoľko programových modulov – samostatných súborov, s názvami:

- config.lua – definícia premenných nevyhnutných pre fungovanie programu
- setup.lua – modul zaisťujúci pripojenie k WiFi
- app.lua – hlavná programová časť
- init.lua – spustenie programu

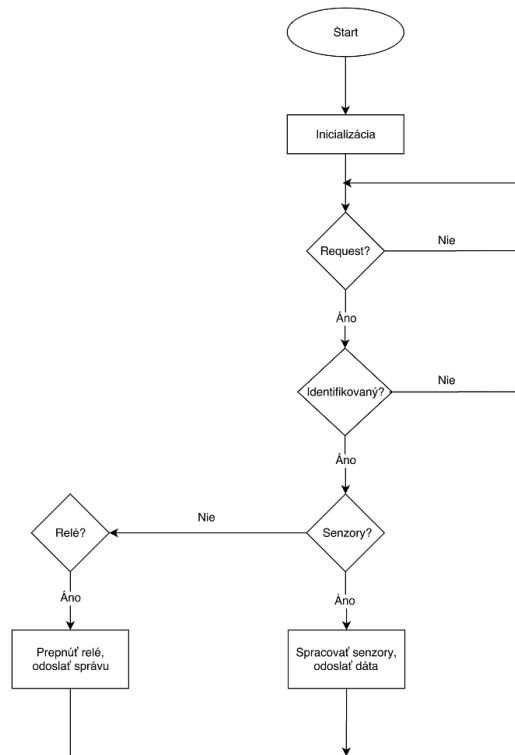
Spustenie programu po štarte modulu, vrátane nečakaného reštartovania je zabezpečené pomenovaním spúšťacieho skriptu názvom „init.lua“.

Skript zbiera údaje zo senzorov a posiela ich na MQTT server do špecifickej témy v momente, ako prijme požiadavku na dáta. Ovládanie relé je taktiež prostredníctvom MQTT serveru a špecifickej témy pre relé. Modul sleduje jednotlivé témy na MQTT serveri a reaguje na ne špecifickými akciami. Fungovanie skriptu je znázornené na obrázku 5.

Popis jednotlivých operácií z diagramu:

1. Štart – bootovanie firmvéru modulu, spustenie inicializačného skriptu (init.lua).
2. Inicializácia – touto časťou je myslená časť kódu v skriptoch, kedy prebieha inicializácia, respektíve nastavovanie modulu. Tu prebieha inicializácia premenných a nastavenia ako pripojenie do WiFi, pripojenie na MQTT server a spustenie hlavnej časti skriptu.

¹⁶Fotorezistor – tiež známe ako foto-odpor, rezistor závislý na svetelnej intenzite alebo fotobunka



Obrázok 5: Workflow diagram programu

3. Request? – v tejto časti modul čaká, kedy sa v téme požiadavkov na MQTT serveri objaví požiadavok. Pokiaľ sa spracuje nejaký požiadavok, program sa posunie ďalej, ak sa žiaden požiadavok neobjaví, modul ďalej čaká.
4. Identifikovaný? – bol prijatý požiadavok, v tejto časti sa overuje, či je prijatý požiadavok určený pre onen konkrétny modul. Ak je identifikácia úspešná, program sa posunie ďalej, ak nie, vracia sa späť do časti čakania na požiadavok.
5. Senzory? – v tejto časti je požiadavok prijatý a identifikovaný, je potrebné určiť, či sa jedná o požiadavok na dáta zo senzorov. Ak áno, program postúpi k metóde, ktorá obsluží odoslanie údajov na MQTT server, ak nie, overuje sa, či sa jedná o požiadavok pre relé.
6. Spracovať senzory, odoslať dáta – v tomto bloku modul zmeria údaje zo senzorov a odošle ich do vytvorenej témy na MQTT server.
7. Relé? – v tomto rozhodovacom bloku sa nejedná o požiadavku na senzory, overuje sa, či sa naozaj jedná o požiadavku na relé. Ak áno, spustí sa metóda na obsluženie relé.
8. Prepnúť relé, odoslať správu – v tejto metóde sa uvedie relé do stavu požadovaného v požiadavke a následne sa odošle správa na MQTT server.

V nasledujúcich podkapitolách, sú popísané jednotlivé programové moduly.

4.6.1 config.lua

V programovom module „config.lua“ sú definované statické globálne premenné pre ostatné programové moduly. Obsahuje premenné definujúce prístupové parametre na pripojenie k WiFi sieti, definované piny vyvedených senzorov, počiatkový stav relátka a iné.

Jedná sa o tzv. „best-practice“ postup definovania konfiguračného súboru, ktorý umožňuje zmenu hodnôt premenných na jednom mieste.

4.6.2 setup.lua

V tomto programovom module sú implementované funkcie obsluhujúce korektné pripojenie k WiFi sieti. Pre správnu funkcionality modulu je kľúčové, aby bol pripojený k WiFi sieti (parametre definované v config.lua), preto je pripájanie nadradené ostatnej funkcionality modulu. Pokiaľ modul nie je pripojený k WiFi, logicky, nie je možné sa s ním spojiť pomocou MQTT správ a ovládať ho.

Modul obsahuje tri funkcie:

- *ma_ip()*
- *pripoj_wifi(siete)*
- *spust()*

Funkcie sú stručne popísané v nasledujúcich odrážkach.

1. Funkcia ***ma_ip()*** v časovom intervale 2,5 sekundy overuje, či modul má pridelenú IP adresu. Pokiaľ áno, dôjde k vypísaniu detailov o pripojení a spusteniu hlavnej funkcionality aplikácie (z app.lua funkcia module.start()). Ak nie, čaká ďalších 2,5 sekundy a znova overuje pridelenie IP adresy. Funkcia *ma_ip()* je volaná vo funkcii *pripoj_wifi(siete)*, prostredníctvom časovača.
2. Funkcia ***pripoj_wifi(siete)***, vyžaduje parameter siete (pole s SSID WiFi sietí v dosahu modulu). Táto funkcia nastaví a spustí vstavané funkcie firmvéru obsluhujúce pripájanie k WiFi sieti, naplní ich parametrami z modulu config.lua a prostredníctvom časovača začne v 2,5 sekundovom intervale zisťovať, či bola pridelená IP adresa volaním funkcie *ma_ip()*.
3. Funkcia ***spust()*** nastaví modul do režimu pripájania k sieti a spustí vyhľadanie WiFi sietí v dosahu modulu. Predá vyhladané siete do parametru funkcie *pripoj_wifi(siete)*. Následne sa spustí pripájanie k sieti, definovanej v súbore config.lua.

4.6.3 app.lua

V module `app.lua` je implementovaná hlavná funkcionálna aplikácia. Programový modul obsahuje funkcie slúžiace na získavanie a spracovávanie dát zo snímačov, ovládanie relé, pripájanie k MQTT serveru, odosielanie a prijímanie (spracovávanie) MQTT správ. Navyše obsahuje jednu lokálnu premennú s názvom „`inst`“, ktorá slúži pre uloženie instance MQTT objektu vznikajúceho vo funkcii `spust_mqtt()`. Výčet funkcií modulu `app.lua`:

- `zmeraj_dht()`
- `teplota_vlhkost()`
- `rele()`
- `svetelnost()`
- `mqtt_data_senzory()`
- `odosli_senzory()`
- `prihlas_do_mqtt()`
- `sprava_prijata(conn, topic, data)`
- `spust_mqtt()`

V nasledujúcich odrážkach sú jednotlivé funkcie stručne popísané.

1. Funkcia `zmeraj_dht()` pomocou zabudovanej knižnice DHT načíta hodnoty zo senzora DHT11 do premenných. V tejto funkcii sa ďalej zistí, či sú namerané hodnoty v poriadku, alebo či pri meraní došlo k chybe kontrolného súčtu, prípadne či vypršal čas pre zmeranie hodnôt. Vracia zmeranú hodnotu, prípadne chybu.
2. Funkcia `teplota_vlhkost()` v svojom tele volá funkciu `zmeraj_dht()` a zisťuje, či pri meraní nedošlo k chybe, ak áno, meranie sa opakuje. Meranie sa maximálne opakuje 20x, ak napriek tomu sú hodnoty nenamerané, jedná sa o chybu senzora a pravdepodobne je potrebné ho vymeniť respektíve skontrolovať správnosť jeho zapojenia.
3. Funkcia `rele()` slúži k prepínaniu relátka a k odosielaniu stavu, v ktorom sa relé nachádza. Zmeny sa vykonávajú pomocou knižnice GPIO a jej metódy `write`.
4. Funkcia `svetelnost()` slúži k zmeraniu hodnoty z fotorezistora. Využíva knižnicu ADC, ktorá číta hodnoty zo vstavaného AD prevodníka.
5. Funkcia `mqtt_data_senzory()` získava a formátuje dáta zo sensorov a relé, pre odoslanie na MQTT server.

6. Funkcia *odosli_senzory()* pomocou knižnice MQTT odošle namerané hodnoty na MQTT server do témy „esp_modul/data“. Taktiež na sériovú linku odošle informáciu o úspechu, či neúspechu, čo je využité na ladenie funkcionality.
7. Funkcia *prihlas_do_mqtt()* sa pomocou MQTT knižnice prihlási do dvoch tém „esp_modul/id MCU“ a „esp_modul/“. Ďalej funkcia odošle na sériovú linku informáciu o úspechu či neúspechu danej operácie.
8. Funkcia *sprava_prijata(conn, topic, data)* má tri vstupné parametre, z ktorých topic a data sú podstatné pre vykonanie operácie závislej na prijatých dátach. Obsahuje rozvetvenú podmienku „if ... then“, pomocou ktorej sa rozhoduje o akú prijatú správu sa jedná a ako sa na ňu odpovie.
9. Funkcia *spust_mqtt()* uloží instanciu MQTT objektu do lokálnej premennej „inst“. Nastaví funkciu, ktorá sa má zavolať pri prijatí správy (funkcia *sprava_prijata(conn, topic, data)*). Nastaví „last will testament“-správa, ktorá sa pridá do témy „esp_modul/module“ a informuje o tom, že sa daný modul odpojil. Na záver sa spustí metóda pripájania, ktorá v sebe zavolá funkciu *prihlas_do_mqtt()* a prihlási modul do jednotlivých tém na serveri.

4.6.4 init.lua

V inicializačnom súbore sú nalinkované všetky potrebné súbory (config.lua, setup.lua, app.lua) a je spustený modul setup.lua funkciou *spust()*, ktorá spustí pripájanie k WiFi sieti. Pri každom zapnutí modulu teda prebehne proces pripájania k WiFi a následne sa spustí hlavná funkcionality (pripojenie k MQTT serveru a čakanie na prijatie requestu).

4.6.5 Využitá pamäť

Pomocou jednoduchšej vstavanej funkcie z knižnice „file“, ktorá vracia „LUA table“, je možné zistiť aké súbory sa nachádzajú v súborovom systéme modulu. V mojom prípade sa jedná o štyri vyššie spomínané programové skripty, ktoré neskompilované dohromady zaberajú 7608 B. Celková zabraná pamäť modulu s neskompilovanými súbormi je 13052 B, podrobný prehľad sa nachádza v tabuľke 4. Skompilovaním súborov je možné veľkosť zabranej pamäte o niečo zmenšiť, v mojom prípade konkrétne o 1833 B. Súbor init.lua nakoľko sa jedná o inicializačný skript, nie je možné skompilovať, pretože v štádiu bootovania modulu sa tento súbor vynútené kompiluje. Súbor init.lua, teda nie je možné predkompilovať a tým ušetriť ďalšie miesto v pamäti modulu.

Kód 2: Výpis použitej pamäte jednotlivými súbormi

```
-- vypis zaberanej pamäte jednotlivymi subormi  
l = file.list()
```

Tabuľka 4: Prehľad využitia pamäte jednotlivými súbormi

názov	neskompilovaný	skompilovaný
init	263 B	-
config	512 B	416 B
setup	1 451 B	1236 B
app	5 382 B	3 860 B
spolu	7 608 B	5 775 B

```
for k,v~in pairs(l) do
    print("nazov:"..k..",┘velkost:"..v)
end
```

Pre zistenie celkovej dostupnej pamäte RAM slúži funkcia „node.heap()“, ktorej návratová hodnota je počet voľných bajtov pamäte RAM. Pri neskompilovaných skriptoch sa po čerstvom naboťovaní modulu jednalo o hodnotu 24 860 B (24.27 KiB), voľnej pamäte RAM. Pri predkompilovaných programoch do byte kódu je voľnej pamäte RAM o niečo viac 26 008 B (25.39 KiB).

4.6.6 Stabilita modulu

Stabilita modulu je najmä ovplyvnená stabilitou napájacieho zdroja a jeho schopnosťou dodávania dostatočného množstva prúdu pre fungovanie modulu. Odber prúdu ESP je najviac ovplyvnený intenzitou WiFi signálu, pri nedostatočnej sile signálu môže narásť odber prúdu až na 215 mA pri vysielaní dát. Napájací zdroj teda musí byť schopný dodávať dostatočnú hodnotu prúdu pre správny a stabilný chod modulu – minimálne 300 mA.

Stabilita modulu je taktiež ovplyvnená optimalizáciou kódu v spúšťaných skriptoch, kde záleží na správnom použití cyklu v kombinácii so vstavanou funkciou časovača vo firmvéri. Samotná funkcia „tmr.delay(x)“ spôsobí, že procesor ESP sa dostane do tzv. non-interrupt módu¹⁷. V praxi je teda lepšie použiť alternatívu funkcie „tmr.alarm(id,interval,mode,function call)“, pomocou ktorej je možné v ľubovoľnom časovom intervale zadanom v μs volať ľubovoľnú funkciu bez tzv. „busy loop“ procesora.

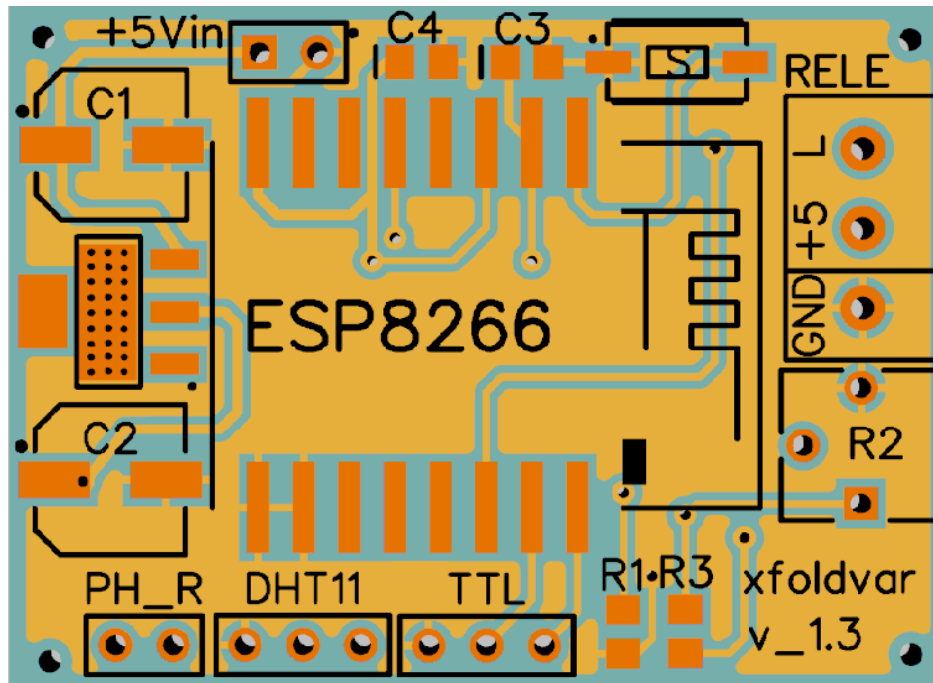
4.7 Doska plošných spojov

PCB respektíve dosku plošných spojov je navrhnutá v programe Diptrace. Rozmery dosky sú 41 mm * 30 mm, na jej ploche sa nachádza:

- plocha pre ESP8266
- napätový regulátor s kondenzátormi

¹⁷non-interrupt mód – mód v ktorom sú blokované všetky prerušenia a inštrukcie

- fotorezistor, potenciometer a smd rezistor
- svorky pre relé, DHT11 a TTL komunikáciu
- pushbutton pre takzvaný „hardreset“
- stabilizačné kondenzátory a pull-up rezistor



Obrázok 6: Finálna podoba PCB modulu

Pred návrhom samotnej dosky, bolo potrebné vytvoriť symbol (do schém) zariadenia ESP8266 v. 12, vytvoriť k nemu príslušnú maticu pre PCB s popisom pinov. Následne bolo potrebné preniesť schému reálneho zapojenia do programu DipTrace Schematic, z ktorého bolo možné následne prejsť na návrh samotného PCB. Pri návrhu bolo dbané na čo najmenšie rozmery výslednej dosky.

Doska je štandardná, dvojvrstvová, na zadnej strane má taktiež vedené cesty, všetky komponenty sa však nachádzajú na strane prednej. Je navrhnutá pre vstupné napätie 5 V a použitie relé s tranzistorom, nie je však problém použiť vhodné SSR relé (maximálny odber 12 mA, spínacie napätie pod 3,3 V). Senzory je možné pripojiť priamo na dosku, alebo je možné ich vyvieť prostredníctvom kábla – napríklad DHT11 pri napätí 3,3 V do 1 m. Pri tvorbe PCB bolo potrebné zohľadniť návrhové pravidlá niektorého z výrobcov PCB dosiek¹⁸. Tieto pravidlá sa týkajú napríklad minimálnej šírky medenej cesty, šírky vrtaných dier a podobne. Doska plošných spojov je vyobrazená na obr. 6, zoznam použitých súčiastok je v tabuľke 5.

¹⁸v mojom prípade sa jedná o spoločnosť OSHpark

Tabuľka 5: Prehľad použitých súčiastok na PCB

Označenie	Súčiastka	Hodnota
C1	kondenzátor	47 μ F
C2	kondenzátor	47 μ F
C3	kondenzátor	100 nF
C4	kondenzátor	100 nF
R1	rezistor	10 k Ω
R2	potenciometer	50 k Ω
R3	rezistor	10 k Ω
S	reset tlačítko	-
1117	napätový regulátor	15 V na 3,3 V
ESP8266	mikrokontrolér	-

Do práce som taktiež vymodeloval 3D model modulu, pomocou programu Dip-Trace a jeho modulu, ktorý umožňuje priradovať daným matriciam 3D modely. Vzniknutý model sa nachádza v prílohách ako obrázok číslo 10. Zadná strana dosky plošných spojov sa nachádza taktiež v prílohách ako obrázok číslo 8.

4.8 Webová aplikácia

Táto časť práce sa zaoberá tvorbou webovej aplikácie, ktorá slúži k správe a ovládaniu ESP modulov. Webová aplikácia je vytvorená v PHP frameworku Nette. Použitie vývojového frameworku je výhodou pri prípadnom rozširovaní alebo úprave aplikácie. Aplikácia sa rozdeľuje do nasledujúcich častí:

- Úvodná stránka a prihlasovanie
- Správa užívateľov
- Správa modulov
- Správa miestností
- Prehľad modulov
- Prehľad miestností

Úvodná stránka a prihlasovanie slúži k podaniu základných informácií užívateľovi a možnosť prihlásenia sa do aplikácie. Bez prihlásenia užívateľa nie je umožnené ani prezeráť, ani ovládať jednotlivé súčasti tejto aplikácie, autorizácia užívateľa je teda kľúčová pri používaní tejto aplikácie.

Sekcia **správa užívateľov** slúži k spravovaniu užívateľov aplikácie. Umožňuje vytvorenie a prípadné zmazanie užívateľa z databázy, čím je vlastne ošetrovaný prístup užívateľov do riadiaceho rozhrania.

Správa modulov umožňuje vyhľadanie pripojených modulov k danému MQTT brokeru, ich načítanie do databázy aplikácie, prípadne ich odstránenie z databázy. Taktiež je možné z tejto sekcie zistiť, či daný modul je priradený k určitej miestnosti, prípadne zmazať namerané hodnoty jednotlivých modulov z databázy.

Správa miestností umožňuje tvorbu miestností, definovanie názvu ovládaného zariadenia a priradenie modulu k danej miestnosti. V tejto sekcii je tiež možné miestnosť zmazať a modul tzv. „uvolniť“, následne je možné tento modul priradiť do inej miestnosti.

Prehľad modulov slúži k zobrazeniu všetkých modulov, ktoré sú úspešne pripojené k aplikácii. Je možné si prezrieť posledné namerané hodnoty z daných modulov, zistiť či daný modul je, alebo nie je priradený k miestnosti.

Prehľad miestností slúži k zobrazovaniu prehľadu vytvorených miestností, ovládaniu pripojených zariadení k modulu a prehliadaniu grafov nameraných hodnôt v jednotlivých miestnostiach.

4.8.1 Užívateľské rozhranie

Užívateľské rozhranie je vytvorené pomocou CSS a HTML frameworku Bootstrap, je plne rezponzívne pre rôzne veľkosti zobrazovacieho zariadenia, respektíve rozlíšenia. Tak isto aj zobrazované grafy sa prispôbujú veľkosti okna webového prehliadača. Grafy je možné exportovať ako obrázok v rôznych formátoch (pdf, jpeg, svg, png), taktiež je možné graf priamo vytlačiť, čo umožňuje napríklad tvorbu prehľadov, alebo súhrnov.

Užívateľské rozhranie je veľmi jednoduché a ľahko sa v ňom orientuje. Navyše je možné ho jednoducho rozšíriť pri prípadnej pridanej funkcionalite aplikácie. Obrázky užívateľského rozhrania sa nachádzajú v prílohách, konkrétne sa jedná o obrázky 14,15 a 16.

4.8.2 Princíp fungovania

Webová aplikácia, podobne ako navrhnutý modul sa po spustení pripojí na MQTT sprostredkovateľa. MQTT sprostredkovateľ nerozlišuje medzi pripojeným modulom a aplikáciou, navonok teda nie je badať žiadny rozdiel. Avšak dôležitá je implementácia JavaScript-ového kódu, implementovaného pomocou knižnice Paho.js. V tejto implementácii je definované, kam na MQTT broker sa má aplikácia prihlásiť, do ktorých tém má prispievať správy a z ktorých tém má správy čítať.

Odosielanie a prijímanie správ sa deje pomocou JavaScript-ových funkcií, prijímanie správ je prepojené s PHP backend-ovou časťou aplikácie, ktorej JavaScript-ová funkcia predá prijaté údaje a tie sa následne uložia do databázy.

Nakoľko je použitý bezplatný MQTT broker od spoločnosti Amazon, ktorý umožňuje v reálnom čase mať pripojených len 10 klientov, je pri práci s aplikáciou

potrebné dbať na počet aktívnych modulov a počet používateľov aplikácie. Ak administrátor vytvorí ďalší účet do webovej aplikácie pre správu ESP modulov, bude možné mať súčasne pripojených 8 modulov, jedného administrátora a jedného bežného užívateľa. Samozrejme, v prípade potreby, nie je problém zvýšiť maximálnu kapacitu súčasne pripojených klientov na MQTT broker a to zaplatením určitého poplatku, stanoveného spoločnosťou Amazon.

Tým, že aplikácia je naprogramovaná v PHP frameworku Nette, je možné ju jednoducho rozšíriť, alebo samotnú aplikáciu pridať ako rozšírenie do existujúceho systému (napríklad webový informačný systém pre správu hotela).

5 Záver

Táto kapitola sa zaoberá výsledkami práce a zhodnotením dosiahnutia stanovených cieľov.

5.1 Zhrnutie

Práca sa zaoberala navrhnutím ovládacieho modulu pomocou MCU ESP8266. Na úvod bolo potrebné sa zoznámiť so všeobecnou problematikou mikro-kontrolérov a ich základnými pojmami. Nakoľko zadanie práce vymedzuje konkrétny mikro-kontrolér, bolo potrebné sa oboznámiť s jeho špecifickými vlastnosťami, ktoré boli nevyhnutné pre realizáciu tejto práce. V práci boli prepojené rôzne technológie, čoho výsledkom je modulárny ovládací a monitorovací systém vhodný pre domácu automatizáciu.

K práci bol skonštruovaný fyzický prototyp modulu a implementovaný program, ktorý je nevyhnutný pre fungovanie modulu. Pomocou programu DipTrace bola navrhnutá doska PCB so všetkými perifériami, ktoré má modul obsahovať. Navyše popri tvorbe modulu a jeho implementácii bola vytvorená webová aplikácia, ktorá slúži k ovládaniu, spravovaniu a prezeraniu zozbieraných údajov z jednotlivých modulov.

V sekcii návrh riešenia je popísaný fyzický návrh modulu, vysvetlená implementácia programu pre modul, návrh dosky plošných spojov a vysvetlenie funkcionality webovej aplikácie.

Po splnení cieľa práce sa dospelo k záveru, že mikro-kontrolér ESP8266 je svojimi vlastnosťami vhodný k domácej automatizácii. Obrovskou výhodou je, že modul disponuje vstavaným WiFi rozhraním a nie je teda nutné moduly fyzicky prepájať kabeľážou.

5.2 Zhodnotenie výsledkov

Aby bolo možné zhodnotiť výsledky práce, je potrebné analyzovať ciele, ktoré boli stanovené na začiatku.

Výsledný modul by mal byť schopný zbierať dáta o teplote, vlhkosti a svetelnosti. Bolo teda nevyhnutné, aby obsahoval potrebnú senzorovú výbavu a implementovaný program, ktorý tieto dáta spracuje.

Ďalej malo byť užívateľovi umožnené ovládať (zapnúť/vypnúť) určité pripojené zariadenie pomocou relé. Modul mal byť schopný tieto zozbierané dáta zo senzorov odosielať a prijímať požiadavky na prepínanie relé. Komunikačným kanálom pre výmenu údajov mala byť WiFi sieť. Pre splnenie tejto časti bolo nutné vytvoriť určité komunikačné rozhranie, v tomto prípade sa jedná o MQTT sprostredkovateľa, kam sa posielajú všetky správy z modulov a aj všetky správy ktoré sú určené modulom.

Na záver bola vytvorená jednoduchá, prehľadná webová aplikácia, ktorá umožňuje prezeráť získané údaje z modulov a ovládať jednotlivé relé.

Ciele boli teda všetky splnené.

5.3 Budúce využitie a možné rozšírenia

Ak budeme hľadať využitie pre tento modulárny systém, je možné nájsť veľké množstvo možností a miest využitia. Moduly je možné použiť napríklad pri automatizácii v domácnostiach, firmách, hoteloch atď. Všade tam, kde je nutné, alebo vhodné takýto systém použiť.

Možnosti rozšírení sú rôznorodé. Moduly je napríklad možné rozšíriť o ďalšiu senzorovú prípadne relé výbavu a tým pádom vytvoriť modul, ktorý bude v širšom spektre monitorovať (napríklad pohyb, výskyt nebezpečných plynov, a ďalšie) respektíve ovládať ďalšie elektrické zariadenie (pridané relé znamená ďalšie zariadenie, ktoré je možné ovládať). Možné rozšírenia sa týkajú aj webovej aplikácie, ktorá samotná môže byť rozširovaná (vizualizácia senzorov, a iné), alebo ako bolo naznačené, môže byť rozšírením k existujúcemu webovému systému.

Rozšíriť je možné aj samotný program pre ESP modul. Napríklad je možné implementovať programový modul, ktorý bude mať za úlohu sprostredkovať možnosť aktualizácie ostatných programových modulov v ESP, čo prináša niekoľko výhod. Napríklad pri zistení chyby v implementácii programu, nie je nutné každý modul samostatne odpojiť a aktualizovať – tento proces by sa odohral prostredníctvom webovej aplikácie. Pri zámene sensorovej výbavy, by mohlo byť možné aktualizovať metódy, ktoré komunikujú so senzormi.

6 Literatúra

- ADAFRUIT INDUSTRIES *DHTxx Sensors[datasheet]*. 2015. Dostupné z: <https://learn.adafruit.com/downloads/pdf/dht.pdf>.
- ADVANCED MONOLITHIC SYSTEMS *AMS1117 - 1A low dropout voltage regulator[datasheet]*. 2010. Dostupné z: <http://www.advanced-monolithic.com/pdf/ds1117.pdf>.
- BACZKOWICZ, K.. *MqttSpy - Getting started* In: GitHub[online]. 2015, [cit. 2016-03-14]. Dostupné z: <https://github.com/kamilfb/mqtt-spy/wiki/GettingStarted>.
- BRIERE, D. *Smart Homes For Dummies*. Indianapolis: For Dummies, 2005. 432 s. ISBN 978-0470165676.
- BRYNILDSEN, CH. JENS. *ESP8266 and stability* In: Flash Gamer[online]. 24.3.2015, [cit. 2016-03-28]. Dostupné z: <http://flashgamer.com/hardware/comments/esp8266-and-stability>.
- DAVIES, J. *MSP430 Microcontroller Basics*. Oxford: Newnes, 2008. 668 s. ISBN 978-0-7506-8276-3.
- ECLIPSE FOUNDATION. *Paho.js* In: Eclipse[online]. 2015, [cit. 2016-03-14]. Dostupné z: <https://eclipse.org/paho/clients/js/>.
- EGLI, P. R. *AN INTRODUCTION TO MQTT, A PROTOCOL FOR M2M AND IoT APPLICATIONS*. Indigoo.com, 2015. Dostupné z: http://www.indigoo.com/dox/wsmw/1_Middleware/MQTT.pdf.
- ESPRESSIF SYSTEMS *ESP8266EX Datasheet[datasheet]*. 2014. Dostupné z: https://www.adafruit.com/images/product-files/2471/OA-ESP8266_Datasheet__EN_v4.3.pdf.
- ESPRESSIF SYSTEMS *Espressif IoT SDK: Programming Guide[guide]*. 2014. Dostupné z: http://www.mikrocontroller.net/attachment/245197/2C-SDK-Espressif_IoT_SDK_Programming_Guide_v0.9.5.pdf.
- ESPRESSIF SYSTEMS *ESP8266 AT Instruction Set*. 2016, Dostupné z: <http://bbs.espressif.com/download/file.php?id=1085>.
- FUTURE TECHNOLOGY DEVICES INTERNATIONAL LTD *TTL-232R-PCB[datasheet]*. 2010. Dostupné z: http://www.ftdichip.com/Support/Documents/DataSheets/Cables/DS_TTL-232R_PCB.pdf.
- HIGHCHARTS. *Highcharts* In: Highcharts[online]. 2016, [cit. 2016-03-14]. Dostupné z: <http://www.highcharts.com/products/highcharts>.

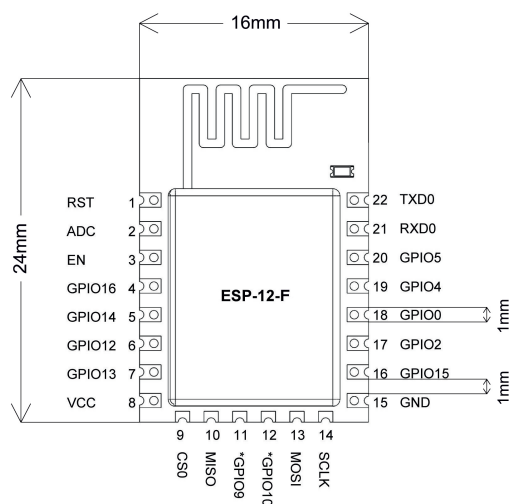
- HOROWITZ, P., HILL, W. *The Art of Electronics*. Vyd. 3. New York: Cambridge, 2015. 1125 s. ISBN 978-0-521-80926-9.
- IERUSALIMSKY, R. *Programming in LUA 2nd edition*. Lua.org, 2006, 328 s. ISBN 978-8590379829.
- IGOE, T. *Making things talk*. Sebastopol: O'REILLY, 2011. 496 s. ISBN 978-1-449-39243-7.
- KOLBAN, N. *Kolban's book on ESP-8266*. 2015. 410 s. Dostupné z: <http://neilkolban.com/tech/esp8266/>.
- LAMPKIN, V., OLIVERA, L. *Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ Telemetry*. First edition. Poughkeepsie, NY: IBM International Technical Support Organization, 2012. 249 s. ISBN 0738437085.
- LSR *Innovative Alternatives to AT Command Sets*. 2015, Dostupné z: <https://www.lsr.com/downloads/wp/2015/innovative-alternatives.pdf>.
- NETTE FOUNDATION. *Slovníček pojmov* In: Nette[online]. 2008, [cit. 2016-03-14]. Dostupné z: <https://doc.nette.org/cs/2.3/glossary>.
- NETTE FOUNDATION. *Nette prednosti* In: Nette[online]. 2015, [cit. 2016-03-14]. Dostupné z: <https://nette.org/cs/#toc-features>.
- NODEMCU *NodeMCU API Documentation*. In: Nodemcu[online]. 2015, [cit. 2016-02-18]. Dostupné z: <http://nodemcu.readthedocs.io/en/>.
- NOVARM LIMITED. *DipTrace software* In: Diptrace[online]. [cit. 2016-03-14]. Dostupné z: <http://diptrace.com/diptrace-software/>.
- POTTE, J. a kol. *MySQL* In: WikiBooks[online]. 2014, [cit. 2016-03-14]. Dostupné z: <https://en.wikibooks.org/wiki/MySQL>.
- SANTOS, R. *Getting Started With The ESPlorer IDE*. 2015, Dostupné z: <http://esp8266.ru/download/esp8266-doc/GettingStartedwiththeESPlorerIDE-RuiSantos.pdf>.
- SCHWARTZ, M. *Arduino Home Automation Projects : Automate your Home using the powerful Arduino Platform*. Birmingham: Packt Publishing, 2014. 131 s. ISBN 1783986069.
- SMITH, A. G. *Introduction to Arduino A piece of cake!*. London: CreateSpace, 2011. 170 s. ISBN 1463698348.
- SOPER, E. *Absolute Beginner's Guide to Home Automation*. Indianapolis: Que, 2005. 384 s. ISBN 0789732076.

VERMESAN, O., FRIESS, P. *Internet of Things - From Research and Innovation to Market Deployment*. Aalborg: River Publishers, 2014. 355 s. ISBN 978-87-93102-94-1.

Prílohy

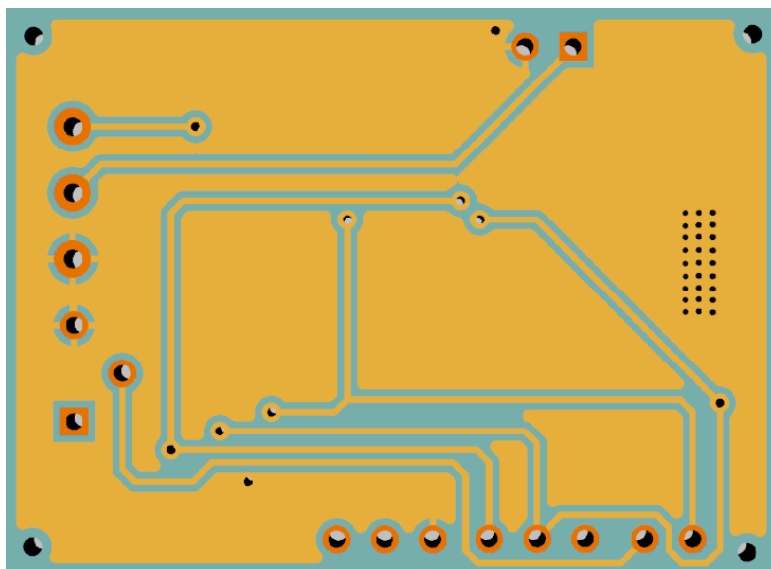
A Obrázky

A.1 ESP8266 v. 12F

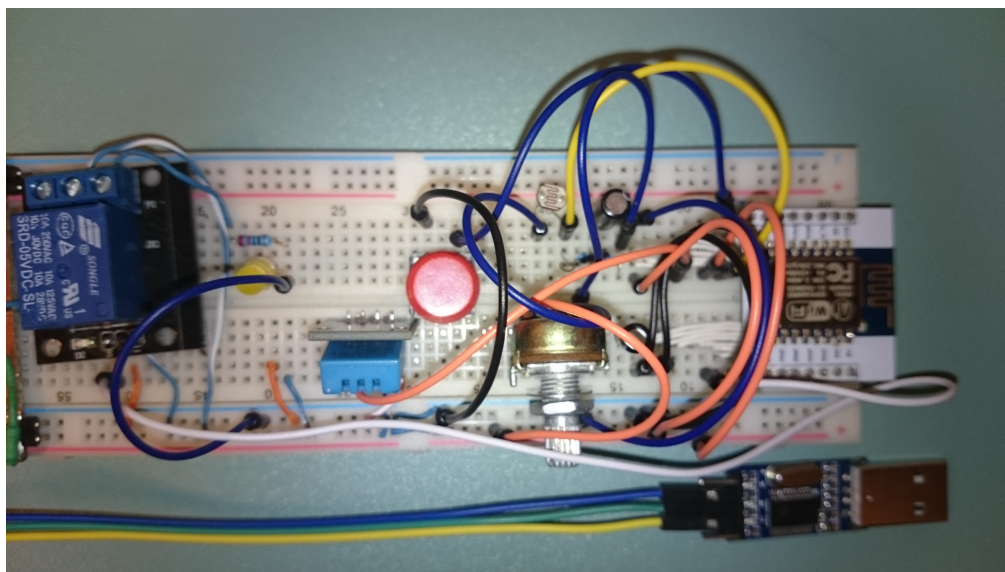


Obrázok 7: ESP-12F

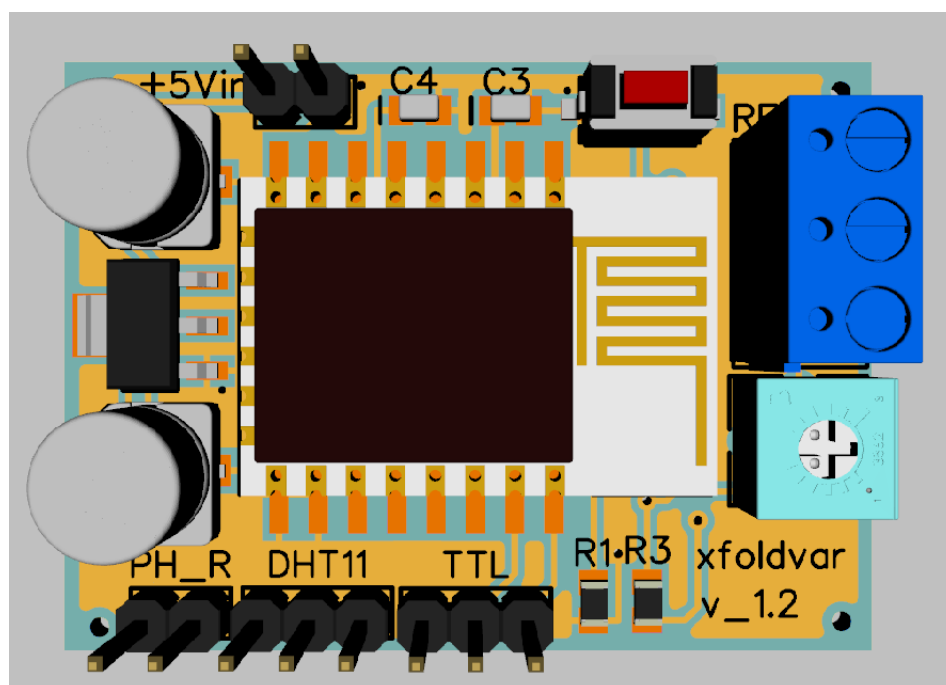
A.2 Modul



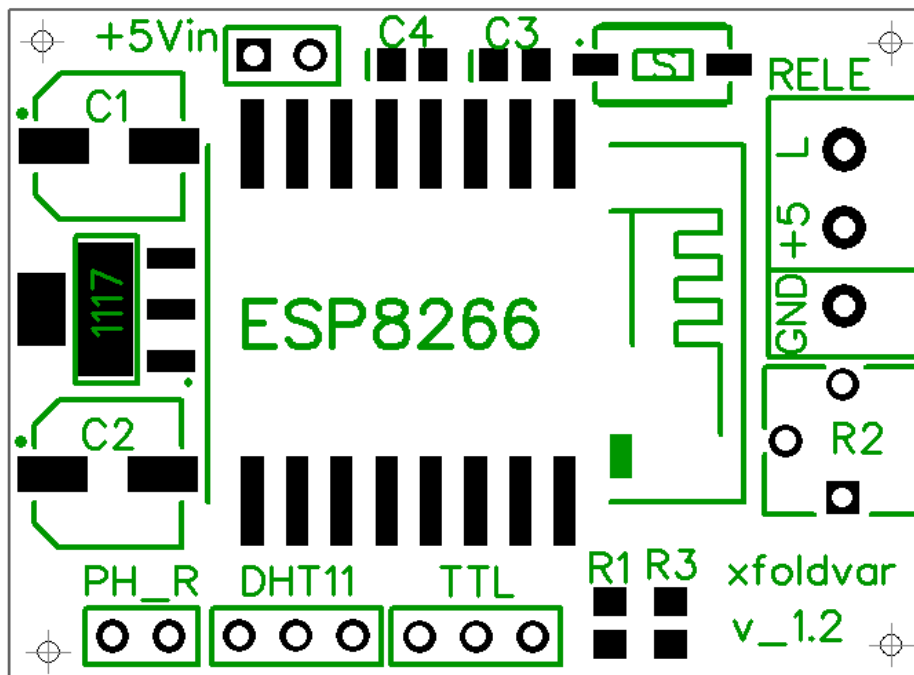
Obrázok 8: Zadná strana dosky plošných spojov



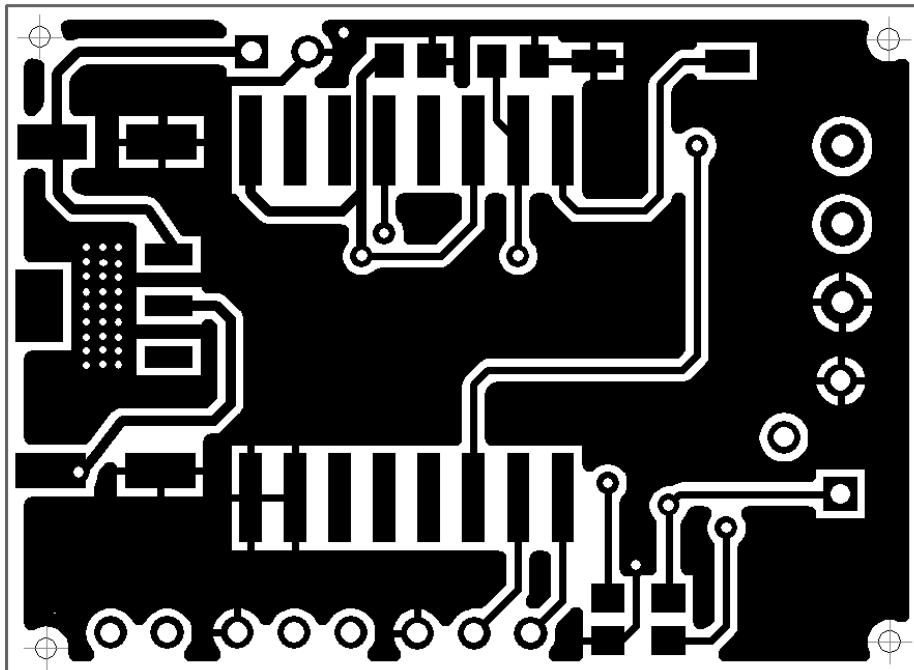
Obrázok 9: Fotka modulu na vývojovej doske



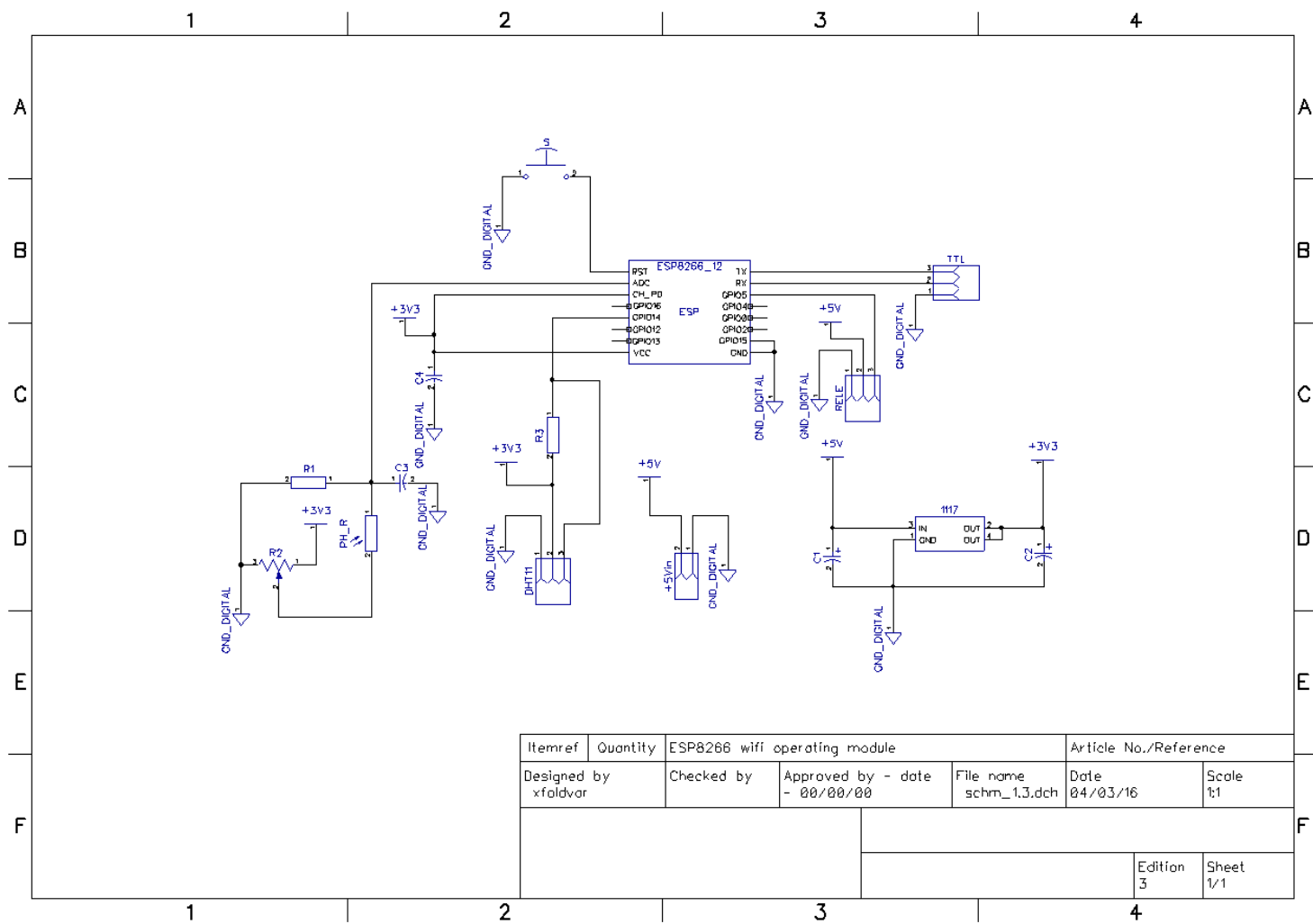
Obrázok 10: 3D model modulu na doske plošných spojov



Obrázok 11: Vrchná potlač PCB dosky



Obrázok 12: Detail návrhu PCB dosky



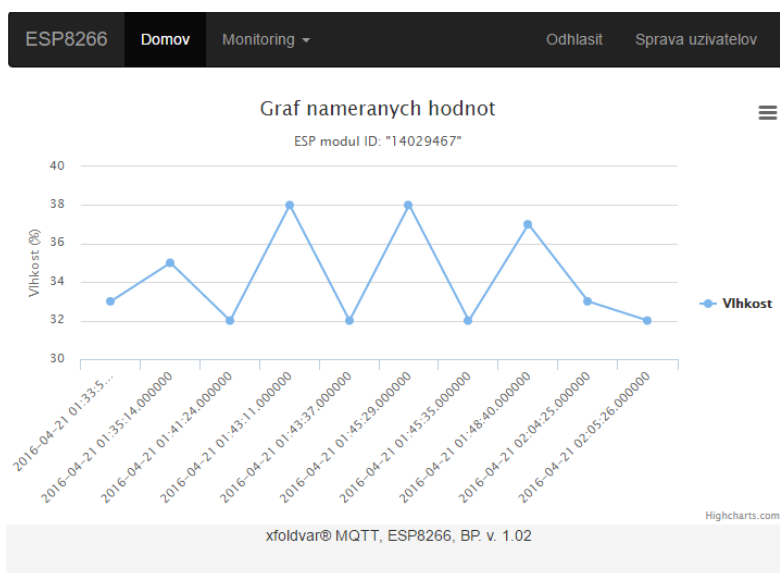
Obrázok 13: Kompletná schéma zapojenia

A.3 Webová aplikácia

Miestnosť	Teplota	Vlhkosť	Svetlo	Zariadenie
Kuchyna	26	32	Šero	kurenie
Obyvacka	25	32	Tma	Telka

xfoldvar@ MQTT, ESP8266, BP. v. 1.02

Obrázok 14: Webová aplikácia prehľad miestností



Obrázok 15: Webová aplikácia zobrazenie grafu

ESP8266 Domov Monitoring ▾ Odhlasiť Správa užívateľov

V tejto sekcii sa manipuluje s modulmi - mazanie meraní, menenie zariadení pripojených k rele, odstránenie modulu, automatické pridávanie nových modulov.

ID modulu	Mazanie meraní	Zmena zariadenia na rele	Odstránenie modulu
1159477	Zmaz merania modulu	<input type="text" value="Telka"/> Zmeniť	Zmaz modul
14029467	Zmaz merania modulu	<input type="text" value="kureníe"/> Zmeniť	Zmaz modul

Automaticky pridaj nové moduly

xfoldvar@MQTT, ESP8266, BP. v. 1.02

Obrázok 16: Webová aplikácia správa modulov

B Priložené CD

Priložené CD obsahuje:

- Zdrojové súbory pre ESP modul.
- Zdrojové súbory webovej aplikácie.
- Prácu v elektronickej podobe.