



## BAKALÁŘSKÁ PRÁCE

---

# Senzorické řešení chytré domácnosti s automatickou diagnostikou komunikace

---

*Autor:*  
Patrik NACHTMANN

*Vedoucí práce:*  
Ing. Martin BULÍN, MSc.

Závěrečná práce k získání akademického titulu *Bakalář (Bc.)*  
*v oboru Systémy pro identifikaci, bezpečnost a komunikaci*

Katedra kybernetiky

13. března 2020

# Declaration of Authorship

Česky, bude stačit kratší - starší bakalářky ze ZČU. Možná je dokonce oficiální, které se k textu přidá až po tisku, vyřešíme až na konci...

I, Patrik NACHTMANN, declare that this thesis titled, „Senzorické řešení chytré domácnosti s automatickou diagnostikou komunikace“ and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.

Signed:

---

Date:

---

*„Something supersmart.“*

Your hero

ZÁPADOČESKÁ UNIVERZITA

*Abstrakt*

Fakulta aplikovaných věd

Katedra kybernetiky

Bakalář (Bc.)

**Senzorické řešení chytré domácnosti s automatickou diagnostikou  
komunikace**

by Patrik NACHTMANN

Your abstract goes here...

## *Acknowledgements*

Your acknowledgements go here...

# Obsah

<b>Abstrakt</b>	<b>iii</b>
<b>1 Úvod</b>	<b>1</b>
1.1 State of the Art . . . . .	1
1.2 Thesis Objectives . . . . .	1
1.3 Thesis Outline . . . . .	1
<b>2 Hardwarové komponenty</b>	<b>2</b>
2.1 Mikročip ESP8266 . . . . .	3
2.2 Senzory . . . . .	9
2.2.1 Teplotní čidlo DS18B20 . . . . .	9
2.2.2 Vlhkoměr DHT11 . . . . .	10
2.2.3 Čidlo intenzity osvětlení TSL2591 . . . . .	12
2.2.4 Čidlo barometrického tlaku a teploty BME280 . . . . .	13
2.2.5 Pohybové čidlo AM312 . . . . .	14
2.2.6 Magnetické čidlo LS311B38 . . . . .	16
2.3 Raspberry Pi . . . . .	17
<b>3 Síťová komunikace a databáze</b>	<b>19</b>
3.1 Protokol MQTT . . . . .	19
3.2 Ukládání dat do databáze . . . . .	19
3.3 Webserver . . . . .	19
<b>4 Diagnostika a detekce anomálií</b>	<b>20</b>
4.1 Detekce chyb na úrovni ESP8266 . . . . .	20
4.2 Detekce anomálií na základě klasifikace . . . . .	20
4.3 Diagnostika stavu čidel na serveru . . . . .	21
<b>5 Webové rozhraní</b>	<b>22</b>
<b>6 Závěr</b>	<b>23</b>
6.1 Future Work . . . . .	23
<b>A1 Structure of the Workspace</b>	<b>24</b>

# Seznam obrázků

2.1	Vztahy mezi použitým hardwarem a měřenými veličinami . . . . .	2
2.2	Vývojová platforma NodeMCU s modulem ESP8266 . . . . .	3
2.3	Diagram odesílání zpráv na základě vzniku události . . . . .	5
2.4	Diagram periodicky opakovaného odesílání zpráv . . . . .	6
2.5	Diagram architektury kódu na microchipu ESP8266 . . . . .	7
2.6	Schéma obvodu platformy NodeMCU, dvou čidel DS18B20 a led diody na plošném spoji . . . . .	10
2.7	Fyzická realizace senzoru s moduly DS18B20 a DHT11 . . . . .	10
2.8	Schéma obvodu platformy NodeMCU, modulu DHT11 a led diody na plošném spoji . . . . .	11
2.9	Schéma obvodu platformy NodeMCU, modulu TSL2591 a led diody na plošném spoji . . . . .	12
2.10	Fyzická realizace senzoru s modulem TSL2591 . . . . .	13
2.11	Schéma obvodu platformy NodeMCU, modulu BME280 a led diody na plošném spoji . . . . .	14
2.12	Fyzická realizace senzoru s modulem BME280 . . . . .	14
2.13	Schéma obvodu platformy NodeMCU, modulu AM312, led diody a externího bzučáku na plošném spoji . . . . .	15
2.14	Fyzická realizace senzoru s modulem AM312 . . . . .	16
2.15	Schéma obvodu platformy NodeMCU, magnetického kon- taktu LS311B38 a led diody na plošném spoji . . . . .	17
2.16	Fyzická realizace senzoru s magnetickým kontaktem LS311B38	17
2.17	Jednočipový počítač Raspberry Pi 4 Model B . . . . .	18

## Seznam tabulek

# Použité zkratky a termíny

<b>IoT</b>	Internet of Things
<b>RPi</b>	Raspberry Pi
<b>ESP8266</b>	Mikročip, mikrokontrolér
<b>NodeMCU</b>	Vývojová platforma s čipem ESP8266
<b>Čidlo</b>	Komponenta k mikročipu (např.: teploměr DS18B20)
<b>Senzor</b>	Celek skládající se z mikročipu a čidla (např.: obvod ESP8266 + DS18B20)



# Kapitola 1

## Úvod

Přeložit do češtiny

Your intro... Thesis ref example: **bulin:2016**, Misc ref example: **smidl:pc**, Article ref example: **mcculloch:neuron**, Online webpage ref example: **online:xor\_solution**

### 1.1 State of the Art

### 1.2 Thesis Objectives

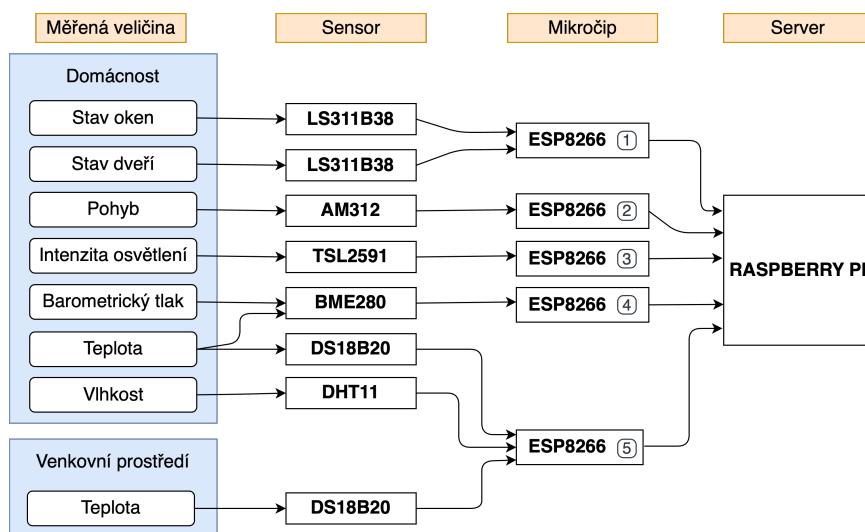
### 1.3 Thesis Outline

## Kapitola 2

# Hardware komponenty

V této kapitole je popsán veškerý použitý hardware v projektu senzorického řešení chytré domácnosti. V projektu je využito 5 mikročipů ESP8266, 8 čidel pro měření fyzikálních veličin a jeden počítač Raspberry Pi.

Měřené fyzikální veličiny lze rozdělit do dvou základních kategorií podle lokace. Veličiny, které jsou měřeny uvnitř domácnosti v rámci místnosti a veličiny venkovním prostředí. Mezi veličiny měřené uvnitř místnosti patří teplota, vlhkost, intenzita osvětlení, stav dveří a oken a barometrický tlak. V místnosti je dále monitorován pohyb osob. Ve venkovním prostředí je měřena teplota. Na Obr. 2.1 je zobrazen použitý hardware v závislosti na měřených veličinách.



OBRÁZEK 2.1: Vztahy mezi použitým hardwarem a měřenými veličinami

Ve snaze o co nejfektivnější využití hardwaru může jeden mikročip číst data z několika čidel. Tento přístup se využívá hlavně u čidel teploty a vlhkosti a u čidel pro monitorování stavu oken a dveří. U ostatních senzorů je zpravidla využitý jeden mikročip pro čtení dat z jednoho specifického čidla. Využitelnost jednoho mikročipu pro více čidel je závislá zejména na fyzickém umístění senzoru (např.: pohybový senzor musí být umístěn v rohu místnosti, kde není vhodné současně měřit jiné veličiny) a na výpočetní náročnosti. Jeden mikročip často zvládne načítat data pouze z jednoho čidla.

## 2.1 Mikročip ESP8266

Základním stavebním prvkem senzorů v tomto projektu je mikročip *ESP8266*. Existuje celá řada mírně odlišných mikrokontrolérů založených na tomto modulu, v principu se ale jedná o velmi levný mikročip s procesorem o frekvenci 80 Hz, flash pamětí typicky od 512 KiB do 4 MiB, 16 vstupně-výstupních GPIO piny a podporou sběrnic SPI, I<sup>2</sup>C a I<sup>2</sup>S.

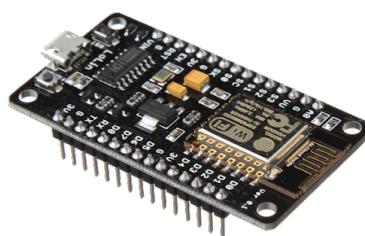


### Vlastnosti mikročipu

Hlavní přednosti toho čipu je podpora Wi-Fi standardu IEEE 802.11 b/g/n. K mikročipu stačí kávové přívést pouze napájení a veškerá komunikace může probíhat bezdrátově po místní síti. Tím odpadá nutnost předem připravené kabeláže v domě a rozšiřují se možnosti využití i v méně dostupných prostorech. Tento druh mikročipu byl zvolen hlavně kvůli kompatibilitě s celou řadou čidel, podpoře několika programovacích jazyků a v neposlední řadě kvůli široce rozšířené komunitě a dostupnosti nejrůznějších návodů pro DIY projekty.

### Vývojové platformy

Vývojové desky s mikročipem ESP8266 jsou vyráběny v několika provedeních. V projektu senzorického řešení chytré domácnosti jsem využil výhradně vývojovou platformu *NodeMCU* (na Obr. 2.2). Tato platforma je přizpůsobena pro pohodlnou komunikaci s mikročipem ESP8266 pomocí sériové linky. Platforma disponuje USB konektorem a vývody GPIO pinů. USB konektor slouží k napájení a zároveň k přenosu dat do paměti mikročipu. Pro zapojení externích komponent a vytváření obvodů lze pohodlně využít breadboard<sup>1</sup> a propojovací kably. Ve fázi vytváření prototypu tedy není nutné pájet.



OBRÁZEK 2.2: Vývojová platforma NodeMCU s modulem  
ESP8266

### Firmware mikročipu

Po zakoupení vývojové platformy je prvně potřeba nahrát aktuální firmware do flash paměti mikročipu. Před nahráním firmwaru je nutné nainstalovat ovladač pro komunikaci s USB rozhraním na desce. NodeMCU zpravidla

<sup>1</sup>Nepájivé kontaktní pole. Komponenta, která umožňuje sestavit prototyp obvodu a je vhodná pro prvotní experimentování.

vyžaduje ovladač CP2102 (alternativně CH340). Pro nahrání samotného firmware do mikročipu jsem zvolil nástroj *esptool.py*. Esptool je program postavený na Pythonu, který umí načíst informace o připojeném mikročipu, vymazat flash paměť, nahrát nový firmware a případně zálohovat flash paměť.

## Vývojové prostředí

Veškerá komunikace s čipem probíhá po USB rozhraní a je založena na principu nahrání skriptů do paměti čipu a následném restartování pro spuštění programu. Jakkoliv změna kódu znamená vymazání původního skriptu z paměti čipu a následném nahrání nového skriptu. Tato skutečnost mírně ztěžuje ladění kódu a odchytávání chyb, ale samotný skript pro mikrokontrolér nebývá příliš dlouhý a k ladění kódu lze efektivně využít příkazovou řádku. Pro ukládání skriptů do mikročipu je potřeba vhodný nástroj pro přístup do adresářové části paměti na čipu. Jedním z těchto nástrojů je *Mpfshell*. Mpfshehell je balíček postavený na Pythonu, který umožňuje editaci souborů uložených v paměti (nahrávání, mazání, vytváření složek apod.) a zpřístupňuje příkazovou řádku na mikročipu. Právě příkazová řádka je zásadní pro rychlé spuštění kódu a odladění chyb, je možné v ní přímo psát části kódu nebo spouštět jednotlivé skripty bez nutnosti restartu.

## Programování mikročipu

Samotné ESP8266 může být programováno v několika jazycích, předně v Arduino IDE, MicroPythonu nebo LUA. V tomto projektu jsou všechny mikrokontroléry programovány v jazyce MicroPython. V rámci sjednocení programovacích jazyků v celém projektu jsem vybral MicroPython právě z důvodu univerzálnosti. MicroPython vychází z Pythonu 3 a je přímo přizpůsoben pro programování mikrokontrolérů. Další části projektu (například backend webové vizualizace nebo klasifikátor dat) jsou postaveny na Pythonu 3, proto se MicroPython v rámci zachování jednotné struktury jeví jako logická volba. Nespornou výhodou tohoto jazyku je velká podpora v rámci komunity a množství návodů a knihoven pro komunikaci s jednotlivými čidly.

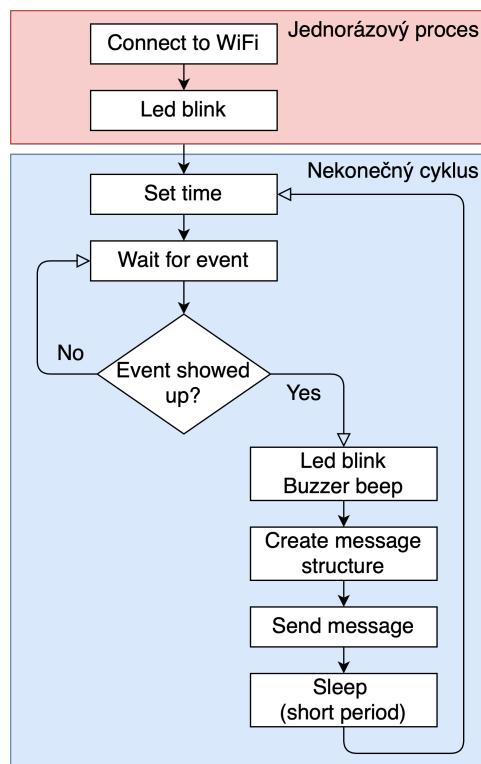
## Odesílání dat z mikročipu

V senzorech pro chytrou domácnost se uplatňují 2 základní přístupy odesílání dat na server. První je založen na odeslání zprávy na základě vzniku události a druhý přístup odesílá zprávy pravidelně s předem zvolenou periodou odesílání. V obou případech se po zapnutí senzoru mikročip nejprve připojí k Wi-Fi a problkne led dioadami. Tento proces proběhne jednorázově a pak senzor přejde do stavu, kdy je připraven načítat data ze senzorů a publikovat zprávy.

## Odesílání zpráv na základě vzniku události

Odesílání zpráv ze senzoru na server na základě vzniku události používá senzor pro detekci pohybu v místnosti a senzor pro monitorování stavu oken a dveří. Tento přístup je popsán v diagramu na Obr. 2.3. V nekonečném cyklu,

do kterého se senzor dostane po připojení k internetu, se prvně sesynchronizuje čas se světovým časem pomocí protokolu NTP<sup>2</sup>. Po synchronizaci senzor čeká, dokud se neobjeví událost, kterou má zaznamenat. Touto událostí je například otevření či zavření okna nebo dveří a nebo pohyb člověka v místnosti. V momentě, kdy událost nastane, čidlo problikne externí led diodou (v případě pohybového čidla se pro indikaci zapne ještě externí bzučák) a přejde do fáze vytváření zprávy. Po vytvoření struktury zprávy, která sestává z několika atributů (více v Kap. 3.1), se senzor připojí k MQTT brokeru a odesle zprávu. Po odeslání zprávy se uspí na 3 vteřiny a celý cyklus se opakuje. Uspání je v programu spíše jako preventivní opatření, aby se kód nežádáným způsobem někde nazacyklil. V případě pohybového senzoru je krátké uspání vhodné také pro to, aby čidlo neposílalo zbytečně moc zpráv, když v místnosti detekuje pohyb. Šetří se tím vytíženosť sítě a pro účel detekce pohybu není nutné odesílat informaci několikrát za vteřinu.



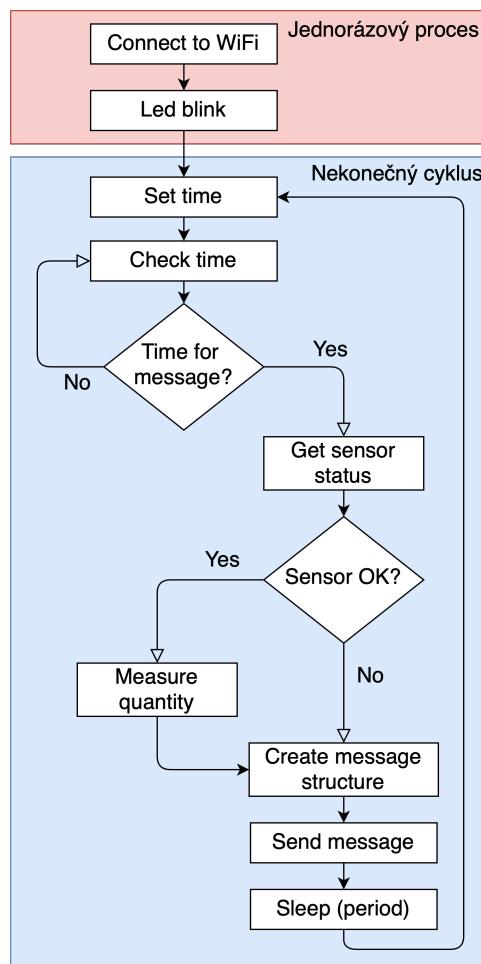
OBRÁZEK 2.3: Diagram odesílání zpráv na základě vzniku události

### Periodické odesílání zpráv

Pravidelné odesílání zpráv s předem danou frekvencí odesílání používají všechny ostatní senzory měřící veličiny, u kterých je vhodné měření po určité době opakovat. Jde o čidla vnitřní a venkovní teploty, vlhkosti, intenzity osvětlení a barometrického tlaku. U těchto senzorů je vhodné měření fyzičkálních veličin opakovat za účelem získání dostatečného množství dat pro následné trénování modelů a pro zajištění neustálé aktuálnosti hodnot ve webové vizualizaci. Vstupními parametry toho přístupu je perioda odesílání dat a určení okamžiku, ve kterém má proběhnout odeslání dat. V případě

<sup>2</sup>Network Time Protocol - protokol pro synchronizaci času po internetu

senzorů pro chytrou domácnost je perioda odesílání 1 minuta a okamžik, kdy se posílají data je vždy ve 30. vteřině (např.: po sobě jdoucí sekvence dat odeslaná v časech 14:25:30, 14:26:30, 14:27:30, ...). **Perioda odesílání dat každou minutu je kompromis mezi dostatečným množstvím dat a vytížeností výpočetní techniky** (vytíženost přenosové sítě, množství potřebné kapacity pro ukládání těchto dat, opotřebení čidel vlivem neustálého načítání dat apod.). Odesílání dat každou minutu vytvoří 1440 vzorků dat za 24 hodin. Po několika dnech sbírání dat už lze z tohoto množství dat počítat statistiky a trénovat klasifikátory. Specifická volba odesílání dat ve 30. vteřině je spíše návrhové rozhodnutí, které sjednocuje odesílání dat ze všech senzorů ve stejném okamžiku. Přístup periodického odesílání dat je popsán v diagramu na Obr. 2.4.



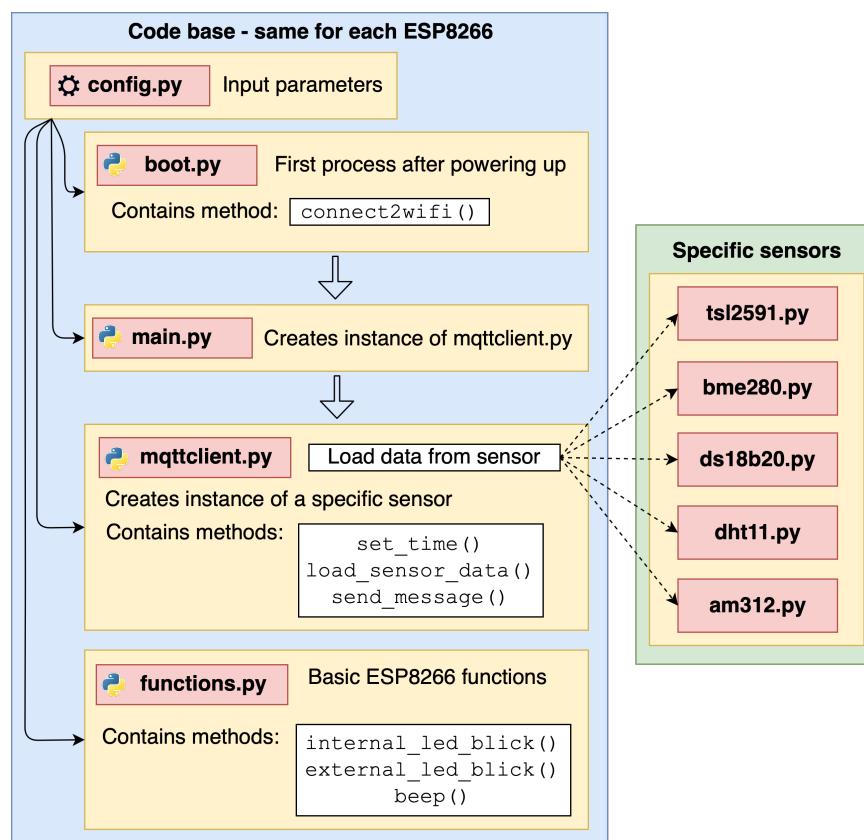
OBRÁZEK 2.4: Diagram periodicky opakovaného odesílání zpráv

V nekonečném cyklu, do kterého se senzor dostane po připojení k Wi-Fi, se nejdříve sesynchronizují vnitřní hodiny mikročipu se světovým časem pomocí protokolu NTP. Následně senzor čeká v cyklu na okamžik, kdy má odeslat zprávu (30. vteřina každé minuty). V momentě, kdy je podmínka splněna a je čas na zprávu, mikročip zjistí status senzoru a načte data (zjištění statusu čidla je přímo spojeno s načtením dat - pokud se podaří načíst data z čidla, status čidla je "OK", pokud dojde k chybě při pokusu o načtení dat, status čidla je "ERROR". Více o detekci chyb na úrovni mikročipu v Kap. 4.1). Po

načtení dat z čidla se vytvoří struktura zprávy (více o struktuře zprávy v Kap. 3.1) a zpráva se odešle na MQTT broker. Po odeslání se senzor uspí na čas  $t - 10[\text{sec}]$ , kde  $t$  je perioda odesílání. Například pro periodu odesílání 1 minuta se senzor uspí na 50 vteřin. Zbylých 10 vteřin, kdy senzor nespí, je určeno pro proces načtení dat ze senzoru (např.: u teplotního senzoru, který načítá data z několika čidel tento proces trvá v průměru 3 až 4 vteřiny), vytvoření zprávy a odeslání zprávy.

### Architektura kódu na mikročipu

Architektura kódu všech senzorů sestává ze stejného jádra. Při návrhu programu pro mikročipy byl kláden důraz na objektovou strukturu programování a co nejuniverzálnější využití. Proto je základ programu pro všechny senzory stejný a liší se jen komunikací s konkrétním čidlem. Tímto přístupem se podařilo sjednotit verze kódu ve všech senzorech a hlavně umožnit rychlé přidání dalších senzorů v budoucnu. Pokud bych do chytré domácnosti chtěl přidat další senzor, který bude měřit jinou fyzikální veličinu, nemusím programovat celou logiku senzoru znova nebo složitě vytrhávat části jinde použitého kódu, ale použiji stávající program, který pouze doplním o implementaci komunikace s konkrétním čidlem. Vizuální pohled na architekturu kódu na mikročipu je na Obr. 2.5.



OBRÁZEK 2.5: Diagram architektury kódu na microchipu  
ESP8266

Struktura kódu pro mikročip ESP8266 se skládá z 6 skript boot.py, main.py,

*mqttclient.py*, *functions.py*, jednoho skriptu pro komunikaci s čidlem a jednoho konfiguračního souboru. Skripty *boot.py* a *main.py* jsou pevně dané programovacím jazykem MicroPython.

- *config.py*

Skript *config.py* je konfigurační soubor se vstupními parametry. V tomto souboru jsou nadefinovány pevné parametry konkrétního senzoru. Jsou zde uloženy přístupové údaje pro připojení k Wi-Fi, parametry pro MQTT komunikaci a označení GPIO pinů podle reálného využití (mikročip a čidlo je napájené na plošném spoji a využití GPIO je pevně dané touto fyzickou realizací). Všechny parametry jsou uloženy v tomto konfiguračním souboru a zbytek kódu pouze odkazuje na tyto parametry.

- *boot.py*

Skript *boot.py* se po zapnutí mikrokontroléru spustí vždy jako první nezávisle na jakémkoliv jiném souboru. V tomto skriptu dochází k připojení k místní Wi-Fi síti. Po úspěšném připojení k síti senzor čtyřikrát zabliká interní led a následně i externí led diodou pro vizuální kontrolu.

- *main.py*

Druhým skriptem, který naběhne ihned po *boot.py* je *main.py*. Toto pořadí spuštění skriptů je pevně stanovené programovacím jazykem. Ve skriptu *main.py* se vytvoří instance třídy `MQTTClient`.

- *mqttclient.py*

Tato třída je v podstatě jádrem celého programu pro mikročip. Dochází zde k synchronizaci vnitřních hodin se světovým časem, načítání dat z čidla, vytváření struktury zprávy a publikování zprávy. Ve třídě `MQTTClient` se vytvoří instance třídy konkrétního čidla, která obstarává veškerou komunikaci s čidlem. Třída `MQTTClient` je tedy pro všechny mikročipy stejná, jednotlivé mikročipy se liší jen třídou pro komunikaci s konkrétním čidlem.

- *functions.py*

Skript *functions.py* je krátký skript, který zahrnuje obecné funkce mikročipu ESP8266. Umožňuje zapojit interní a externí led diody a zapnout externího bzučáku. Funkce jsou napsané univerzálně a načítají vstupní parametry ze souboru *config.py*. Z konfiguračního souboru načítá například číslo GPIO pinu externí led diody - každý senzor může mít externí led diodu umístěnou na jiném GPIO pinu, kvůli využití místa na plošném spoji, ale metoda pro rozsvícení diody je vždy stejná.

Soubory *tsl2591.py*, *bme280.py*, *ds18b20.py*, *dht11.py* a *am312.py* jsou skripty pro komunikaci s konkrétním čidly a mikročip obsahuje vždy jen ten komunikační skript, který pro obsluhu svého čidla potřebuje.

Tímto objektově orientovaným programováním bylo dosaženo maximální univerzálnosti a přehlednosti kódu. Všechny senzory v chytré domácnosti. Hlavní výhodou tohoto přístupu je snadná rozšiřitelnost kódu v budoucnu a díky udržení jedné vývojové větve je na každém mikročipu možné

snadno aktualizovat kód. Díky konfiguračnímu souboru je možné měnit vstupní parametry bez nutnosti zásahu do samotného kódu.

## 2.2 Senzory

V projektu chytré domácnosti bylo použito celkem 7 typů čidel pro měření fyzikálních veličin. Mezi tyto čidla patří teplotní čidlo *DS18B20*, vlhkoměr *DHT11*, čidlo intenzity osvětlení *TSL2591*, čidlo barometrického tlaku a teploty *BME280*, pohybové čidlo *AM312* a magnetické čidlo pro monitorování stavu oken a dveří *LS311B38*. Tyto fyzikální veličiny mají z pohledu chytré domácnosti význam pro automatizaci, zajištění bezpečnosti objektu a přehled o podmínkách **vevnitř** domu a okolí. Data z těchto senzorů jsou následně využita pro trénovaní možnosti klasifikaci jednotlivých hodnot a pro predikci měřených hodnot. Konkrétní senzory byly vybrány na základě kompromisu mezi cenou, funkčností a přesností měření.

### Postup fyzické realizace čidel

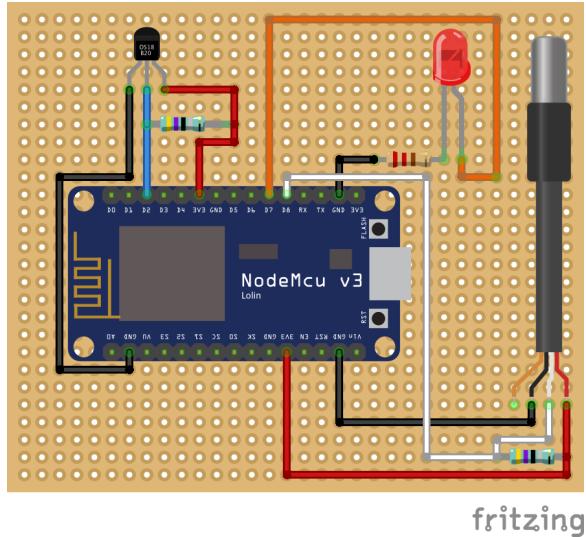
Prvním krokem při návrhu chytrých čidel je stanovení fyzikálních veličin, které chceme sledovat a otestovat spolehlivost čidel pro měření těchto veličin. Po vybrání konkrétních senzorů následuje sestrojení prototypu - zapojení obvodu na breadboardu. Po vytvoření funkčního obvodu může začít experimentování a ladění kódu pro načítání dat z čidla. Po úspěšném naprogramování komunikace s čidlem jsem přešel k vytvoření pevného fyzického obvodu napájením jednotlivých součástí na plošný spoj. Tento plošný spoj má nespornou výhodu ve své životnosti a odolnosti, na rozdíl od obvodu na breadboardu, kde se kabely mohou vlivem neopatrnosti vypojovat. Ke každému senzoru jsem přidal červenou externí led diodu pro indikaci různých změn a stavů. Uživatel v podstatě nemá žádný jiný způsob vizuální kontroly funkčnosti senzoru. Led diodu využívám jako signalizaci při úspěšném připojení k Wi-Fi a jako signalizaci odeslání zprávy. Vždy, když senzor odešle zprávu, dioda jednou blikne.

#### 2.2.1 Teplotní čidlo DS18B20

Teplotní čidlo Dallas DS18B20 se vyrábí ve více provedení, v projektu chytré domácnosti jsou použity 2 čidla, jedno v klasickém provedení pro měření vnitřní teploty a jedno voděodolné pro měření venkovní teploty. Rozsah měření tohoto čidla je od  $-55^{\circ}\text{C}$  do  $+125^{\circ}\text{C}$  s přesností  $\pm 0,5^{\circ}\text{C}$  v mezích od  $-10^{\circ}\text{C}$  do  $+85^{\circ}\text{C}$ . Doba nutná pro konverzi 12 bitové hodnoty teploty do digitálního čísla je 750 ms. Datová komunikace probíhá přes rozhraní One-Wire a pro přenos dat je využitý jeden pin. Každé čidlo DS18B20 má svoji unikátní 64 bitovou sériovou adresu. Díky tomu je umožněno vytvořit One-Wire obvod s několika čidly, které jsou připojeny k jednomu digitálnímu GPIO pinu a pomocí toho sériového čísla lze čidlo snadno identifikovat při programování komunikace čidla s mikročipem.

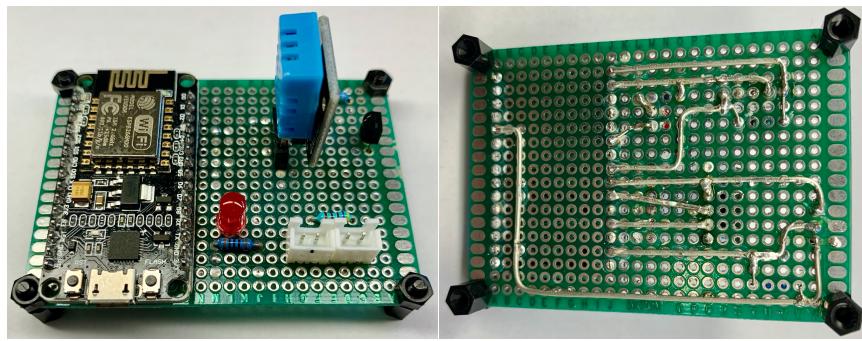
Čidlo je k mikročipu připojeno třemi dráty - jeden drát pro napájení, jeden pro uzemnění a jeden pro datovou komunikaci. Napájecí drát je na ESP8266 připájen k pinu 3,3V, zemnící drát ke GND a datová linka je připájena k digitálnímu pinu D4. Do obvodu je potřeba připojit rezistor s odporem

$4,7k\ \Omega$ , který spojuje datový a napájecí drát. Schéma zapojení obou čidel DS18B20 je zobrazeno na Obr. 2.6.



OBRÁZEK 2.6: Schéma obvodu platformy NodeMCU, dvou čidel DS18B20 a led diody na plošném spoji

Čidlo DS18B20 se osvědčilo přesností měření a důvěryhodností naměřených hodnot. Při prvotním testování jsem zapojil najednou více těchto čidel blízko sobě a naměřené hodnoty ze všech čidel se lišily v rámci desetin stupně. V pro chytré domácnosti jsou hodnoty teplot zaokrouhlovány na 2 desetinná místa, tudíž přesnost v rámci desetin dostačuje a odchylinky jsou zanedbatelné. V tomto projektu jsou teplotní čidla DS18B20 a vlhkostní snímač DHT11 z důvodu efektivního využití hardwaru a stejnémum umístění v místnosti připojeny k jednomu mikročipu ESP8266. Senzor skládající se z jedné platformy NodeMCU, jednoho vnitřního čidla DS18B20, jednoho venkovního čidla DS18B20, jednoho vlhkoměru DHT11 a externí led diody je umístěn v rohu místnosti. Venkovní čidlo má 1 m dlouhý kabel a je umístěno za oknem. Na Obr. 2.7 je vyfocena fyzická realizace zkonztruovaného senzoru.



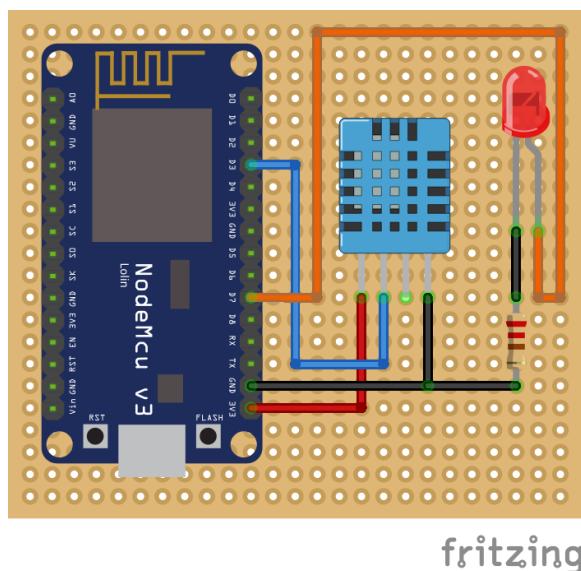
OBRÁZEK 2.7: Fyzická realizace senzoru s moduly DS18B20 a DHT11

## 2.2.2 Vlhkoměr DHT11

Modul DHT11 je určen pro měření vlhkosti a teploty uvnitř místnosti. Měřit teplotu umožňuje v rozmezí od  $0\text{ }^{\circ}\text{C}$  do  $+50\text{ }^{\circ}\text{C}$  s přesností  $\pm 2,0\text{ }^{\circ}\text{C}$ . Z

důvodem menšího rozsahu měření a hlavně nižší přesnosti je toto čidlo v projektu chytré domácnosti používáno k měření pouze vlhkosti. Vnitřní vlhkost je měřena v rozmezí 20 % až 90 % s přesností  $\pm 5\%$ . Chyba měření vlhkosti také není zcela zanedbatelná, nicméně pro základní představu o vlhkosti v místnosti postačuje. Často nezáleží na konkrétní hodnotě vlhkosti, ale spíše na změně této hodnoty v průběhu dne. Pokud je například celý den domácnost bez přítomnosti lidí, drží se vlhkost celý den na stejné hodnotě. Po příchodu člověka do místnosti se hodnota vlhkosti zvýší (sice mírně v rámci jednotek procent, ale s touto informací už je možné dále pracovat).

Čidlo DHT11 je k mikročipu připojeno třemi dráty - jeden drát pro napájení, jeden pro uzemnění a jeden pro přenos dat. Modul vyžaduje napětí v rozmezí 3,0 V až 5,5 V. Po zavedení napětí k modulu je před měřením vyžadováno minimálně 1 vteřinu počkat na kalibraci čidla. Tuto prodlevu je nutné brát v úvahu při každém načítání dat z čidla. Ze zkušenosti s komunikací s čidlem DHT11 raději používám 2 vteřinovou pauzu, při kratším intervalu se často stávalo, že čidlo nestihlo odpovědět a kód spadl do chyby. Metoda pro čtení hodnot z čidla DHT11 vrací hodnotu v procentech zaokrouhlenou na celá čísla. Schéma zapojení modulu DHT11 a vývojové platformy NodeMCU je zobrazeno na Obr. 2.8.



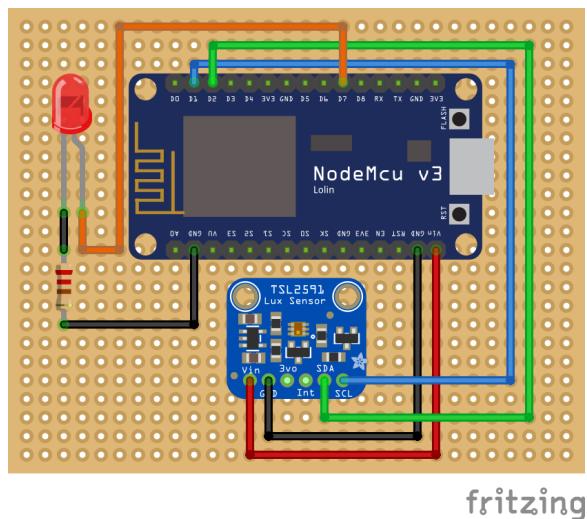
OBRÁZEK 2.8: Schéma obvodu platformy NodeMCU, modulu DHT11 a led diody na plošném spoji

Fotografie fyzické realizace senzoru využívající modul DHT11 je na Obr. 2.7. Modul DHT11 není úmyslně připájen přímo k plošnému spoji, ale je připojen přes konektor na desce. Během několika měsíců testovacího provozu přestalo fungovat jedno čidlo DHT11 a muselo být nahrazeno za jiné. Díky připojení přes dutinkovou lištu byla výměna modulu záležitostí několika vteřin. Tato výměna senzoru probíhá za provozu bez nutnosti restartu senzoru, senzor pomocí automatické detekce chyb pozná nefunkční čidlo a po výměně za nový kus sám naběhne do režimu čtení hodnot z modulu (více v Kap. 4.1).

### 2.2.3 Čidlo intenzity osvětlení TSL2591

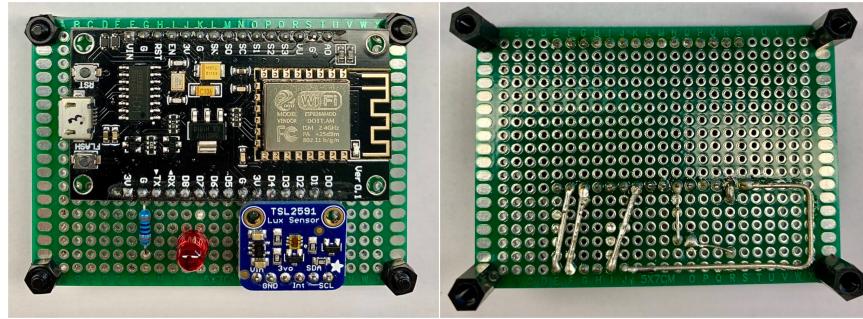
TSL2591 je čidlo sloužící pro měření intenzity osvětlení, které konvertuje intenzitu světla do digitálního výstupu přenášeného sběrnicí I<sup>2</sup>C. Výstupem modulu je hodnota intenzity osvětlení v jednotce lux s přesností na 4 desetinná místa. Toto čidlo je jedno z nejdražších v projektu chytré domácnosti, ale nabízí víceméně dokonalé rozpoznání intenzity osvětlení od 188  $\mu$ Lux do 88 000 Lux. Hodnota s přesností na 4 desetinná místa je zaokrouhlována na celá čísla, protože větší přesnost není vyžadována. Intenzita osvětlení v místnosti během dne se mění rádově o stovky luxů, v noci je osvětlení blízké 0 lux, přes den v rozmezí přibližně 30 až 500 lux a na přímém slunci je to několik desítek tisíc lux. Modul garantuje přesnost měření v teplotních podmínkách od -30 °C do +80 °C. Rozsah napájení desky s čidlem TSL2591 je od 3,3 V do 5,0 V. Velkou předností tohoto čidla je přítomnost diod pro měření infračerveného, celospektrálního a viditelného světla.

Každý modul TSL2591 má svou unikátní 7 bitovou adresu a komunikace s mikročipem ESP8266 probíhá přes I<sup>2</sup>C směrnicí. K čidlu je přivedeno napájení z 3,3 V výstupu na mikročipu, dále je napájení GND drátem a I<sup>2</sup>C komunikaci (výstupy SDA a SCL) s mikrokontrolérem zajišťují digitální GPIO piny D1 a D2. Schéma zapojení čidla TSL2591, platformy NodeMCU a externí led diody je zobrazeno na Obr. 2.9.



OBRÁZEK 2.9: Schéma obvodu platformy NodeMCU, modulu TSL2591 a led diody na plošném spoji

Modul TSL2591 se během testování v projektu chytré domácnosti osvědčil pro svou kvalitu zpracování, přesnost naměřených hodnot, neobvykle velkému rozsahu měření a spolehlivostí. Datová komunikace s tímto čidlem je výpočetně náročná, proto je použitý samostatný mikročip ESP8266 pro obsluhu jen tohoto modulu. Fotografie zkonstruovaného senzoru je na Obr. 2.10.



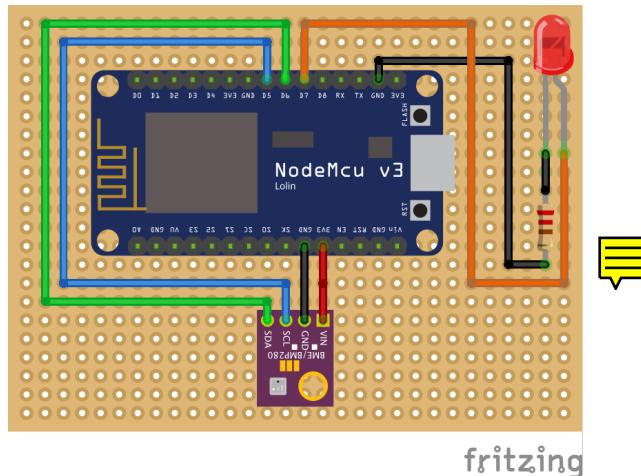
OBRÁZEK 2.10: Fyzická realizace senzoru s modulem  
TSL2591

#### 2.2.4 Čidlo barometrického tlaku a teploty BME280

Modul BME280 od výrobce BOSH slouží k měření vnitřní teploty a barometrického tlaku. Čidlo je schopno měřit okolní teplotu v rozmezí  $-40^{\circ}\text{C}$  do  $+85^{\circ}\text{C}$  a barometrický tlak v rozsahu 300 hPa až 1100 hPa. Hodnoty teploty jsou měřeny s přesností  $\pm 1^{\circ}\text{C}$  a hodnoty barometrického tlaku s přesností  $\pm 1 \text{ Pa}$ . Modul vrací zkonzervovanou hodnotu obou veličin s rozlišením na 2 desetinná místa.

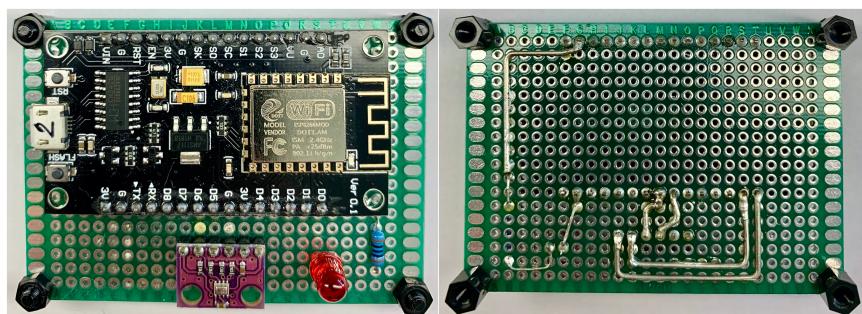
Při porovnání naměřených hodnot teploty z čidla BME280 a z DS18B20 jsem zjistil, že hodnoty se liší v průměru o méně než  $0,5^{\circ}\text{C}$ . Tento rozdíl je pro účely chytré domácnosti zanedbatelný a lze tedy předpokládat, že hodnoty teploty v místnosti jsou velmi přesné a čidlo DS18B20 pro běžné použití dostačuje, přestože je několikanásobně levnější, než modul BME280. Hodnoty barometrického tlaku je vhodné monitorovat s přesností na 2 desetinná místa, protože tato veličina se v místnosti v průběhu dne mění jen v minimálních mezích, za několik měsíců měření to průměrně bylo od 970 hPa do 980 hPa.

Komunikace s mikrokontrolérem probíhá po I<sup>2</sup>C sběrnici. Reakční doba modulu od požadavku na změření hodnoty k odeslání hodnoty činí 1 vteřinu. S tímto zpožděním je opět nutno počítat při každém načítání hodnot. Kvůli výpočetní náročnosti při komunikaci s modulem a velikosti knihovny ovlaďovače čidla je v tomto projektu využitý jeden samostatný mikročip ESP8266, který obsluhuje pouze toto čidlo. Schéma zapojení čidla BME280, platformy NodeMCU a externí led diody je zobrazeno na Obr. 2.11.



OBRÁZEK 2.11: Schéma obvodu platformy NodeMCU, modulu BME280 a led diody na plošném spoji

Modul je k mikročipu ESP8266 připojen čtyřmi dráty - na výstup 3,3V pro napájení, GND pro uzemnění a dva digitální piny D5 a D6, které obstarávají přenos dat po sběrnici. Rozsah napájení desky s modulem BME280 je od 1,8 do 3,6 V. Senzor je umístěn na stole v rohu místnosti. Fotografie fyzické realizace senzoru je na Obr. 2.12.



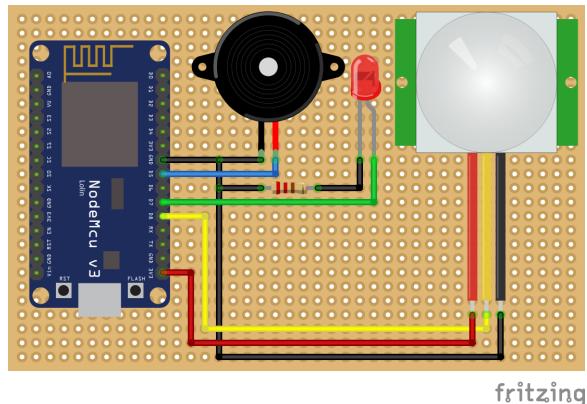
OBRÁZEK 2.12: Fyzická realizace senzoru s modulem BME280

### 2.2.5 Pohybové čidlo AM312

Modul pyroelektrické detekce pohybu AM312 slouží k monitorování pohybu osob v místnosti. Výstupem čidla je logická 1 v případě detekce pohybu a logická 0, pokud v místnosti není detekován pohyb. Tento modul je rozměrově velmi kompaktní a pohyb detekuje ve vzdálenosti do 5 metrů. Úhel detekce pohybu je do 100°. V případě místností v chytré domácnosti je tato detekční vzdálenost a úhel dosťažující.

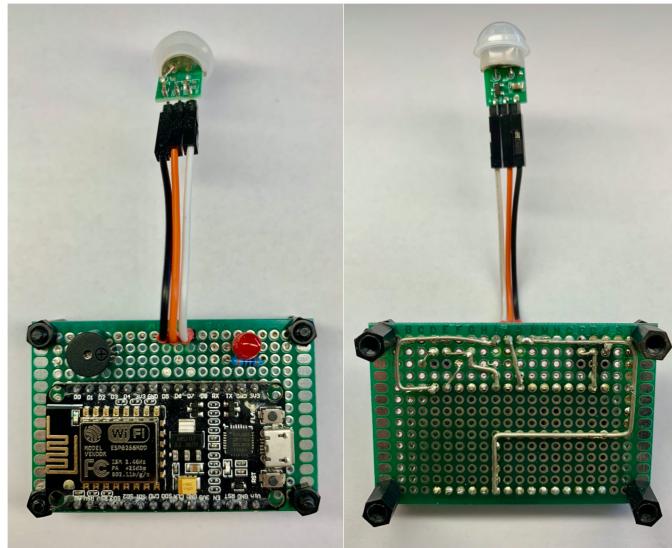
Modul AM312 se během několikaměsíčního provozu osvědčil pro svou spolehlivost. Pohyb byl detekován vždy, když jsem vstoupil do místnosti a zároveň modul neposílal falešné zprávy o pohybu v případě, že v domácnosti nikdo nebyl. Při experimentování s pohybovými čidly jsem testoval ještě modul HC-SR501, který má lepší specifikaci ve smyslu větší detekční vzdálenosti 7 m a úhlu do 120°, nicméně s tímto modulem byl problém právě s falešnými detekcemi.

Čidlo AM312 je konstruováno pro vstupní napětí od 2,7 do 12 V a pro prostřední s okolní teplotou od -20 °C do +60 °C. K mikrokontroléru ESP8266 je připojeno třemi dráty - jeden k vývodu 3,3 V pro napájení, jeden k GND výstupu pro uzemnění a jeden k digitálnímu pinu D8 pro přenos informace. Na plošném spoji tohoto senzoru je navíc přidán externí bzučák, který má využití jako alarm v případě detekce pohybu. Tento zvukový signál se hodil spíše pro prvotní ladění a testování na falešné poplachy, po odladění se jako vizuální signalizace detekce pohybu využívala externí led dioda, která se při zaznamenání pohybu rozsvítí na dobu 2 vteřin. Touto indikací jsem si ověřoval správnou funkčnost senzoru. Schéma zapojení pohybového čidla AM312, platformy NodeMCU, externí led diody a externího bzučáku na plošném spoji je na Obr. 2.13.



OBRÁZEK 2.13: Schéma obvodu platformy NodeMCU, modulu AM312, led diody a externího bzučáku na plošném spoji

Za účelem co nejlepší detekce je senzor umístěn v rohu místnosti naproti dveřím. Toto specifické umístění ovoluje využití mikročipu ESP8266 současně s jiným senzorem, ale v tomto případě by to ani nebylo žádané. Pohybový senzor je jedním ze dvou senzorů, které odesílají zprávy na základě vzniku události (více v Kap. 2.1). Tento přístup nelze efektivně kombinovat s periodickým odesíláním další veličiny. Fotografie fyzické realizace senzoru je na Obr. 2.14.

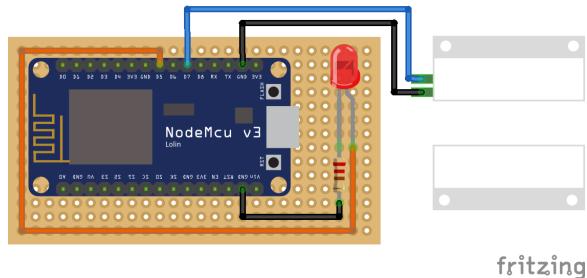


OBRÁZEK 2.14: Fyzická realizace senzoru s modulem AM312

### 2.2.6 Magnetické čidlo LS311B38

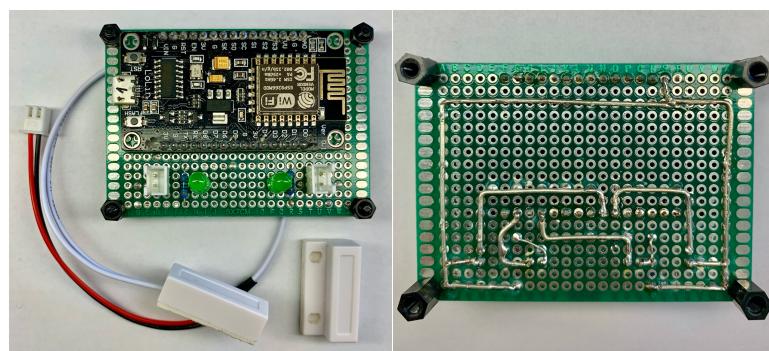
Magnetický jazýčkový kontakt LS311B38 slouží k detekci otevřeného okna nebo otevřených dveří. Toto čidlo je jedním z nejjednodušších čidel v projektu chytré domácnosti. Jedná se o 2 kusy magnetu, jedna část je umístěna na okně nebo dveřích a druhá část je připevněna k dveřnímu nebo **okennímu** rámu. Magnety musí být umístěny tak, aby v případě zavřeného okna nebo dveří, byly oba kusy magnetu blízko u sebe. Výstupem tohoto čidla je logická 1 v případě, že jsou magnety spojeny blízko u sebe (zavřené okno nebo zavřené dveře) nebo logická 0 v případě, kdy jsou magnetické jazýčky rozpojeny. V projektu chytré domácnosti je monitorován stav jednoho okna a jedněch dveří. Pro tuto realizaci je využitý jeden mikrokontrolér ESP8266, který obsluhuje oba magnetické kontakty. Na plošném spoji jsou dvě externí led diody, jedna slouží pro vizuální indikaci otevřených dveří, druhá pro indikaci otevřeného okna.

Spínací vzdálenost magnetů je od 15 mm do 25 mm. Princip zapojení magnetických jazýčku k mikročipu ESP8266 je velmi podobný zapojení tlačítka. Z magnetu vedou dva kabely, které jsou připájeny k výstupům GND pro uzemnění a digitálnímu pinu D7. Oba kabely vedou jen z jednoho kusu magnetu a mikročip čeká na přiblížení druhého magnetu, čímž se obvod spojí a signál je přes digitální pin přenesen do mikročipu. Schéma zapojení magnetického kontaktu LS311B38, platformy NodeMCU a externí led diody je zobrazen na Obr. 2.15.



OBRÁZEK 2.15: Schéma obvodu platforemy NodeMCU, magnetického kontaktu LS311B38 a led diody na plošném spoji

Vzhledem k jednoduchosti čidla lze předpokládat velmi dlouhou životnost a spolehlivost. Fotografie fyzické realizace senzoru je na Obr. 2.16.



OBRÁZEK 2.16: Fyzická realizace senzoru s magnetickým kontaktem LS311B38

## 2.3 Raspberry Pi

Raspberry Pi je jednočipový počítač s operačním systémem Raspbian. Raspbian OS je založený na Linuxu a po instalaci nabízí podporu několika programovacích jazyků a je přímo přizpůsoben na DIY projekty. Základní deska umožňuje připojit externí monitor pomocí microHDMI a externí komponenty jako je klávesnice a myš přes USB. K místní síti se připojuje klasickým LAN kabelem s koncovkou RJ45. Oproti mikrokontrolérům jako jsou čipy ESP8266 nebo Arduino nabízí kromě ovládání příslušenství pomocí GPIO kontaktů hlavně možnost programovat samotné aplikace. V projektu chytré domácnosti byla použita v současné době nejvýkonnější verze počítače Raspberry Pi 4 - Model B s 4 GB RAM, který byl uveden do prodeje v červnu 2019. Tento model je osazen 1,5 GHz čtyřjádrovým procesorem ARM Cortex-A72 a jako externí monitor lze použít displej s rozlišením 4K při 60 snímcích za vteřinu. Operační systém je nainstalován na externí microSD kartě s kapacitou 32 GB. Tato nejvýkonnější konfigurace byla zvolena z důvodu vytíženosti RPi v projektu chytré domácnosti.



OBRÁZEK 2.17: Jednočipový počítač Raspberry Pi 4 Model B

### Využití v chytré domácnosti

Počítač Raspberry Pi má v projektu chytré domácnosti několik využití.

- MQTT Broker
- Databázový server
- Trénování modelů a klasifikace
- Web server

Primárně slouží jako MQTT broker, který přijímá všechny příchozí zprávy ze senzorů (více v Kap. 3.1). Dále na RPi běží databázový server za účelem ukládání všech příchozích zpráv do databáze (více v Kap. 3.2). Současně jsou na RPi trénovány modely, na základě kterých je prováděna klasifikace jednotlivých příchozích dat (více v Kap. 4.2). Za účelem grafické vizualizace naměřených dat a aktuálních hodnot veličin běží na RPi webový server (více v Kap. 3.3).

Velkou výhodou RPi jsou minimální rozměry a hlavně malý odběr proudu. V těchto aplikacích je nutné, aby počítač byl neustále zapnutý. Klasické PC by se k těmto účelům nehodilo kvůli vysoké spotřebě a nepotřebně velkému výkonu. Náklady na provoz a pořízení počítače RPi jsou ve srovnání s klasickém PC serverem několikanásobně nižší.

Raspberry Pi 4 Model B má i v tomto případě, kdy na RPi probíhá několik procesů paralelně vedle sebe, dostatečný výkon. Doba počítání modelů a klasifikace zpráv je v rámci vteřin, narozdíl od starších modelů RPi. Nevýhodou tohoto přístupu, kdy jeden server slouží k několika účelům, je nebezpečí pádu celého systému, pokud server přestane pracovat. V případě, že z nějakého důvodu přestane pracovat RPi, přestanou se do databáze ukládat všechny příchozí zprávy, nově příchozí zprávy tudíž nebudou ani klasifikovány a webová vizualizace bude nedostupná. V případě nedostupnosti webové vizualizace pravděpodobně nejde o velký problém, ale pokud nejsou příchozí zprávy ukládány do databáze delší dobu, jde o závažnější problém a výsledné modely mohou být zkresleny. Tuto nevýhodu je vhodné vyřešit rozdělením procesů na více strojů - na více RPi. V této práci byl pro zálohu databáze využit nezávisle ještě jeden počítač RPi, který ale nakonec nebyl potřeba.

## Kapitola 3

# Síťová komunikace a databáze

Smyslem zkonstruovaných čidel je jejich snadná implementace do bytu či domu.

ESP8266 potřebuje jen napájení, celá komunikace mezi mikročipem a brokerem probíhá po WiFi

### 3.1 Protokol MQTT

Co je to protokol MQTT (vznik, využití)

Proč zrovna tento protokol

Princip použití protokolu MQTT (v síti je broker, který čeká na příchozí zprávy, ostatní zařízení jsou v módu publish - odesílají zprávy na broker, hierarchie zpráv, ...)

Zprovoznění komunikace přes MQTT

Sestavení struktury zpráv + identifikátorů čidel (popis jednotlivých atributů ve zprávě, proč jedno esp může posílat více zpráv do různých topiců, popis zvolené hierarchie topiců, ...)

ESP má aktuální čas pomocí NTP protokolu - díky tomu je možné přidávat do zpráv timestampy Popis logiky odesílání zpráv z ESP - ESP odešle zprávu periodicky vždy v polovině minuty (např.: 14:30:30, 14:31:30, ... ) - lehké nepřesnosti způsobené dobou načítání dat ze senzorů, ...

### 3.2 Ukládání dat do databáze

Základní princip databáze, proč jsme se rozhodli pro použití databáze, ...

Co je MongoDB (určení, využití, funkčnost, ... )

Ukládání hodnot měřených veličin do MongoDB

### 3.3 Webskript

Popis backendu engine.py Backend - z hlediska programování - jeden soubor engine.py, který zajišťuje serverování webové stránky, ukládání dat do databáze, sensor check, ...

## Kapitola 4

# Diagnostika a detekce anomálií

Proč řešit diagnostiku čidel v rámci chytré domácnosti, ...

Úvod do detekce anomálií, základní principy, smysl využití, ...

Diagnostika a detekce anomálií v projektu chytré domácnosti probíhá na 3 úrovních ...

### 4.1 Detekce chyb na úrovni ESP8266

První a nejnižší úroveň diagnostiky jednotlivých senzorů

Implementace detekce chyb na úrovni samotného microchipu

Popis základního principu - při načítání ze dat ze senzoru čidlo pozná, když je senzor nefunkční - nevrací žádné hodnoty a ESP pošle info o nefunkčním senzoru)

Robustnost - odolnost ESP vůči výpadkům senzorů - esp nespadne kvůli nefunkčnímu senzoru, jen změní status zprávy, ...

Nonstop provoz - senzory lze k ESP připojovat a odpojovat v reálném čase bez nutnosti vypínání nebo restartu microchipu, pokud senzor v jakémkoliv časovém okamžiku odpojím - změní se status zprávy na "error", ve chvíli kdy senzor zase připojím na desku - esp začne měřit a posílat, ...

### 4.2 Detekce anomálií na základě klasifikace

Implementace funkcí scikit-learn pro natrénovaní modelů

Nastavení modelu a natrénování modelů pro jednotlivé veličiny

Porovnání schémat natrénovaných modelů

Implementace do projektu - klasifikace jednotlivých příchozích zpráv (přidání atributu "classification" do každé zprávy) Implementace detekce anomálií přijatých zpráv na základě rozhodnutí klasifikace - klasifikace 1 nebo -1 (1 v případě, že přijatá zpráva svou hodnotou "odpovídá" natrénovanému modelu, -1 v případě, že se vychyluje od natrénovaného modelu)

Přenos této informace o klasifikaci do webového rozhraní

### 4.3 Diagnostika stavu čidel na serveru

Implementace detekce anomálií na úrovni serveru (brokeru)

Implementace funkce sensor-check()

Periodická kontrola odesílání zpráv jednotlivých čidel - pokud čidlo z neznámého důvodu neodešle zprávu nebo pokud se zpráva nepřenese k brokeru - informace o anomálii se přenese do webového rozhraní

Vysvětlení stavů čidel - 3 možné stavy čidla - ok, value-error, error

Závěr diagnostiky: kontrola stavu jednotlivých čidel + kontrola věrohodnosti posílaných hodnot + kontrola pravidelnosti odesílaných zpráv

## Kapitola 5

# Webové rozhraní

Podle toho jak to půjde.. Kapitola 5 Interakce systému z uživatelem - Webové rozhraní - Hlasové ovládání

Důvod využití webové vizualizace

Frontend - popis struktury webu - proč 3 záložky, popis Overview, Analytics, About, ...

Smysl Overwiev - nonstop zobrazení dat na monitoru, ...

Přijímání aktuálních zpráv z ESP bez nutnosti refresh stránky přes web-sockets - co je websockets, důvod využití této technologie, alternativa v podobě json souboru - nevýhody, ...

Zajištění integrity celého webu (nově přijaté zprávy se propisují na web, po znovunačení se ihned ukazují poslední přijaté hodnoty - má smysl hlavně u neperidicky posílaných veličin, ...)

# Kapitola 6

## Závěr

Conclusion and Discussion

Conclusion text...

### 6.1 Future Work

Outlook...

## Příloha A1

# Structure of the Workspace

