

**ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE**

FAKULTA STROJNÍ



**DOMÁCÍ AUTOMATIZACE
S VYUŽITÍM RASPBERRY PI
A OPENSOURCE PRODUKTŮ
NA BÁZI PROTOKOLU MQTT**

**BAKALÁŘSKÁ
PRÁCE**

2019

**TOMÁŠ
BAŤKA**

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Bat'ka** Jméno: **Tomáš** Osobní číslo: **465334**
Fakulta/ústav: **Fakulta strojní**
Zadávací katedra/ústav: **Ústav přístrojové a řídicí techniky**
Studijní program: **Teoretický základ strojního inženýrství**
Studijní obor: **bez oboru**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Domácí automatizace s využitím Raspberry Pi a OpenSource produktů na bázi protokolu MQTT

Název bakalářské práce anglicky:

Home automation based on Raspberry Pi and MQTT protocol

Pokyny pro vypracování:

Popis protokolu MQTT a možnosti jeho implementace
Rešerše HW a SW opensource řešení pro Smart Houses využívajících MQTT
Návrh řešení řídicího a komunikačního hardware pro domácí automatizaci
Popis a připojení senzorů a akčních prvků k RPi
Přenos senzorických dat s využitím protokolu MQTT.

Seznam doporučené literatury:

- [1] MQTT V3.1 Protocol Specification. WWW: <http://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html>
- [2] Antonin Vojáček, IoT MQTT prakticky v automatizaci - 1.díl - úvod | Automatizace.HW.cz, led. 2017. WWW: <https://automatizace.hw.cz/iot-mqtt-prakticky-v-automatizaci-1dil-uvod.html>
- [3] S. Long, Raspbian Stretch has arrived for Raspberry Pi, en-GB, srp. 2017. WWW: <https://www.raspberrypi.org/blog/raspbian-stretch/>
- [4] Home Assistant. WWW: <https://www.home-assistant.io/>

Jméno a pracoviště vedoucí(ho) bakalářské práce:

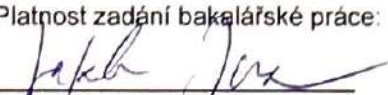
Ing. Mgr. Jakub Jura, Ph.D., U12110.3

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

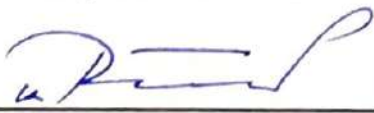
Datum zadání bakalářské práce: **26.04.2019**

Termín odevzdání bakalářské práce: **12.06.2019**

Platnost zadání bakalářské práce:


Ing. Mgr. Jakub Jura, Ph.D.
podpis vedoucí(ho) práce


podpis vedoucí(ho) ústavu/katedry

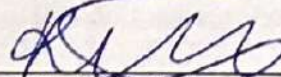

prof. Ing. Michael Valášek, DrSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

26-04-2019

Datum převzetí zadání



Podpis studenta

Poděkování

Děkuji všem lidem kolem mě, kteří mě podporovali při psaní této bakalářské práce. Osobně bych pak chtěl poděkovat mému vedoucímu Mgr. Ing. Jakobovi Jurovi Ph.D., za jeho čas a chuť mi radit a za jeho připomínky k mé práci.

Prohlášení

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a že jsem uvedl všechny použité prameny a literaturu, ze kterých jsem čerpal.

V Praze, 25. května 2019

Abstrakt

Cílem této práce je navrhnout strukturu datové komunikace řídicího softwaru chytrého domu s využitím Open-Source produktů využívající pro přenos dat protokol MQTT. V práci je nejdříve detailně popsán protokol MQTT. Následuje řešení Open-Source jednodeskových počítačů a automatizačního softwaru pro chytré domy. Hlavním bodem je pak vypracování schématického návrhu datové struktury s využitím vybraných řešeršovaných produktů. Tento model je poté zprovozněn a komunikace je otestována zapojením senzorů a WiFi spínačů. V poslední části jsou popsány možnosti vizualizace odesílaných dat.

Klíčová slova: MQTT, Chytrý dům, Open-Source, domácí automatizace

Vedoucí: Ing. Mgr. Jakub Jura, Ph.D.
Odbor Automatického řízení,
Fakulta strojní
České vysoké učení technické v Praze
Technická 2
160 00 Praha 6
Česká republika
jakub.jura@fs.cvut.cz

Abstract

The aim of the bachelor thesis is to design the structure of data communication of smart home control software using Open-Source products, that are using MQTT for data transfer. In the first section of thesis is focus laid on detailed description of the MQTT protocol. This part is followed by a review of Open-Source single board computers and smart home automation software. The main point of the thesis is then a schematic design of a data structure by using selected reviewed products. This model was afterwards put into operation and the communication was tested by connecting sensors and WiFi switches. At the end of the thesis are described the possibilities of visualization of the transferred data.

Keywords: MQTT, Smart Home, Open-Source, Home Automation

Title translation: Home automation based on Raspberry Pi and MQTT protocol

Obsah

1 Úvod	1	3.1 Hardware - jednodeskový počítač	11
1.1 Cíle práce	2	3.1.1 Raspberry pi	11
2 Protokol MQTT	3	3.1.2 Banana Pi	14
2.1 Základní topologie sítě komunikující pomocí MQTT	4	3.1.3 Orange Pi	15
2.2 Hierarchie zpráv - dělení témat	5	3.1.4 Porovnání technických parametrů	15
2.3 Obsah zpráv	6	3.2 Software	16
2.4 Nastavení odesílání zpráv	6	3.2.1 Home Assistant	16
2.4.1 Quality of Service	6	3.2.2 OpenHab	19
2.4.2 Retained Message	7	3.2.3 SmartThings	20
2.4.3 Last Will Testament	8	4 Návrh řídicího systému chytrého domu s využitím Open-Source	23
2.5 Zabezpečení přenosu zpráv	8	5 Připojení senzorů a akčních prvků	27
3 Domácí automatizace pomocí Open-Source	9	5.1 Akční prvky MQTT sítě a jejich připojení	27
3.0.1 Open-Source	10	5.1.1 Čipy Espressif	27
3.0.2 Historický vývoj Open-Source	10	5.1.2 Připojení ESP8266 do MQTT sítě včetně integrace do Home Assistanta	29
3.0.3 Definice Open-Source	10	5.2 Připojení senzorů a nastavení jejich datové komunikace	31

5.2.1 DHT11/DHT22 (Digital Humidity and Temperature)	32
5.2.2 Senzor dýmu	33
5.2.3 PIR senzor	34
5.2.4 WiFi Switch	35
6 Vizualizace dat	39
6.1 LovelaceUI	39
6.2 HADashboard	40
7 Závěr	43
Příloha	45
Literatura	47

Obrázky

2.1 Porovnání síťových modelů TCP/IP a ISO/OSI [1]	3
2.2 Zařazení MQTT do ISO/OSI modelu [2]	4
2.3 Základní schéma přenosu dat rozvržení sítě s využitím MQTT ...	5
2.4 Schéma přenosu dat sítě s využitím MQTT při rozvržení s dvěma brokery	5
2.5 Grafické znázornění Quality of Servis MQTT [3]	7
3.1 Zjednodušený historický vývoj "mini počítačů" Raspberry Pi [4] ...	12
3.2 Popis vstupních a výstupních periférií na Raspberry Pi 3B [5] ...	13
3.3 Schéma GPIO pinů pro Raspberry Pi 3B+ [6]	14
3.4 Vybrané jednodeskové počítače značky Banana Pi	14
3.5 Orange Pi 3 [7]	15
3.6 Logo Home Assistant [8]	16
3.7 Schéma struktury operačního systému HassOS a HASS.IO [9] ...	18
3.8 Logo OpenHAB [10]	19
3.9 Schéma konfigurace struktury komponentů v OpenHABu [11] ...	20
3.10 Logo SmartThings [12]	20
4.1 Detailní schéma komunikace řídicích a ovládacích prvků řídicího systému chytrého domu	24
5.1 Logo Espressif Systems [13]	27
5.2 Čip ESP8266EX [14]	28
5.3 Dev Kit NodeMCU ESP32 [15] .	29
5.4 Porovnání klíčových technických parametrů čipů ESP8266 a ESP32 [16]	29
5.5 Uživatelské rozhraní firmaru TASMOTA	31
5.6 Schéma zapojení senzoru DHT11/DHT22 k vývojovému kitu NodeMCU [17]	32
5.7 Schéma zapojení detektoru kouře k vývojovému kitu NodeMCU [17] ..	34
5.8 Schéma zapojení PIR čidla k vývojovému kitu NodeMCU [17] ..	34
5.9 Popis GPIO pinů WiFi spínače Shelly 1 [18]	35

5.10 Schéma zapojení WiFi spínače Shelly 1 s manuálním spínačem a žárovkou pro DC a AC [19]	36
6.1 Příklad uživatelského rozhraní pomocí LovelaceUI [20]	40
6.2 Příklad uživatelského rozhraní pomocí HADashboard [21]	41

Tabulky

3.1 Tabulka s porovnáním technických parametrů jednodeskových počítačů Raspberry Pi, Orange Pi a Banana Pi	16
--	----

Použité zkratky

BSON Binary JSON

GPI General Purpose Input/Output

HMI Human-Machine Interface

I/O Input/Output

IDE Interactive Development Environment

IoT Internet of Things

JSON JavaScript Object Notation

LAN Local Area Network

M2M Machine-to-Machine

MQTT MQ Telemetry Transport

OS Operční systém

OTA Over-the-Air

PCB Printed Circuit Board

PIR Passive Infrared

PLC Wide Area Network

RPI Raspberry Pi

SBC Single Board Computer

SoC System on Chip

WAN Wide Area Network



Kapitola 1

Úvod

V současné době je velice často skloňován výraz Smart Home nebo chcete-li česky chytrý dům. Díky stále dostupnějším bezdrátovým komunikačním rozhraním a tedy zlevňováním chytrých zařízení obecně zažívá celý obor Internet of Things (IoT) neboli internet věcí velký rozvoj.

Již není nutné pro řízení chytrého domu nainstalovat po celém domě metalické vedení pro komunikaci, které bylo jedinou opravdu funkční variantou řízení v domě ještě před 20 lety. Nyní se pro ovládání a automatizaci elektroniky v domě, ale i elektricky ovládaného vytápění, klimatizace, zabezpečení a dalších využívá bezdrátová komunikace. Technologií podporujících bezdrátovou komunikaci je celá řada a stále vznikají nová řešení. Nedá se určit, která a zda vůbec nějaká technologie bude výrazně dominovat. Spíš je pravděpodobnější varianta paralelně fungujících technologií, které budou díky postupné standardizaci a centrálnímu řídicímu bodu řídicích systému schopny spolu komunikovat.

Jednou z technologií, která byla navržena přímo pro internet věcí je protokol pro bezdrátovou komunikaci MQTT. Internet věcí můžeme zjednodušeně definovat jako síť fyzických zařízení vybavených elektronikou, softwarem, senzory a hlavně síťovou konektivitou, která umožňuje zařízením mezi sebou komunikovat a přeposílat si data. Z toho jednoznačně vyplývá, že internet věcí by měl v ideálním případě fungovat bez nutného zásahu člověka. Proto při realizaci IoT hledáme technologické řešení, které bude spolehlivé a zároveň dostatečně jednoduché, aby bylo provozuschopné na všech typech chytrých zařízeních. Jedním z těchto komunikačních řešení může být právě protokol MQTT.

■ 1.1 Cíle práce

Cílem této bakalářské práce je navrhnout řídicí systém chytrého domu, který bude využívat pro komunikaci MQTT protokol a open-source řešení jak pro hardwarovou tak i pro softwarovou část.

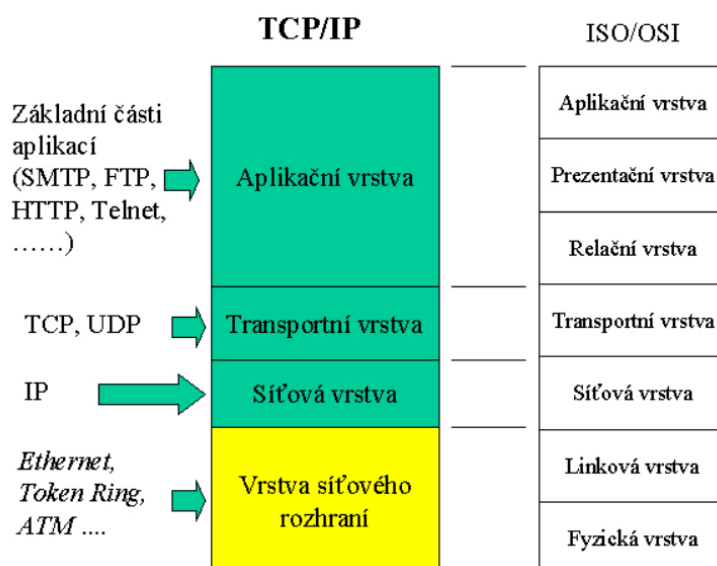
V teoretické části detailně popíši MQTT protokol. Vypracuji rešerši open-source jednodeskových počítačů, které je možné použít jako řídicí jednotku, a rešerši open-source řídicích systému pro chytrý dům, které jsou kompatibilní s předvybranými řídicími jednotkami.

V praktické části vypracuji návrh řídicího systému včetně detailního schématu. Celé navržené řešení zprovozním a pro ukázkou popíši připojení vybraných senzorů a akčních členů, které budou komunikovat s řídicím systémem pomocí protokolu MQTT. Data budou vhodně vizualizována ve vybraném řídicím systému.

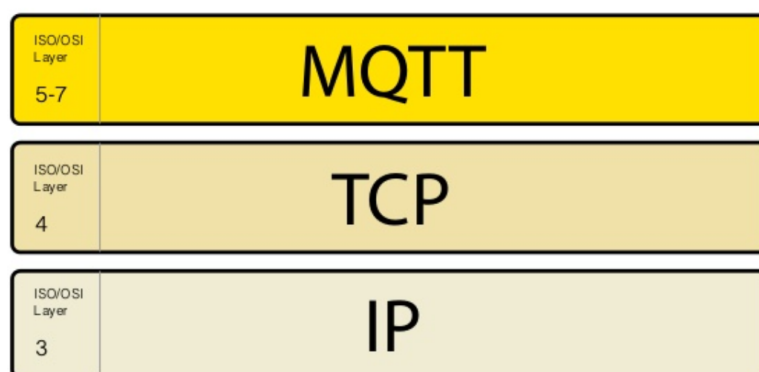
Kapitola 2

Protokol MQTT

MQTT (MQ telemetry transport) je jednoduchý a nenáročný protokol navržený pro přenos menšího obsahu dat mezi dvěma zařízeními (M2M - machine to machine), tedy primárně určen pro IoT aplikace. [3]. MQTT protokol popisuje pouze strukturu zpráv, ale nedefinuje způsob přenosu dat. K tomu využívá TCP/IP protokol. To znamená, že MQTT tvoří pouze aplikační hladinu TCP/IP modelu, [22] která je ekvivalentní pro poslední 3 vrstvy ISO/OSI modelu. Díky své jednoduchosti dokáže rychle přenášet data i na malých mikroprocesorech často implementovaných například do chytrých domácích spotřebičů.



Obrázek 2.1: Porovnání síťových modelů TCP/IP a ISO/OSI [1]



Obrázek 2.2: Zařazení MQTT do ISO/OSI modelu [2]

První verze protokolu vznikla v IBM v roce 1999 a byla navržena dvojicí Andy Stanford-Clark a Arlen Nipper. Následně stála za vývojem Eclipse Foundation, která v roce 2013 přistoupila na standardizaci MQTT pod OASIS. Tak vznikl standard MQTT verze 3.1.1 pod označením ISO/IEC 20922:2016.

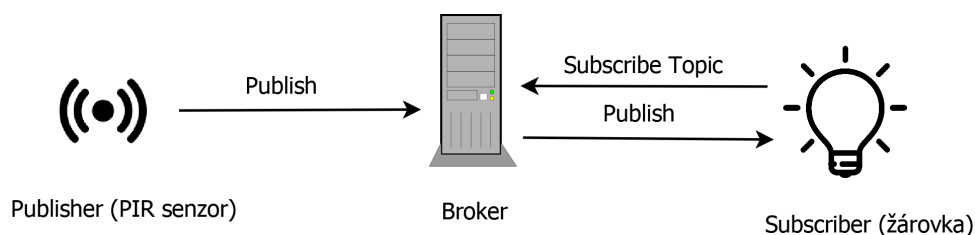
Dalším vývojem vznikla i verze s označením MQTT-SN, kde SN je zkratka pro *sensor network*. Hlavní rozdíl oproti významně více užívanému protokolu MQTT nalézáme v tom, že pro základ komunikace využívá protokol UDP na rozdíl od protokolu TCP/IP, což umožňuje používat různé sítě nevyužívající rodinu TCP/IP protokolů, jakou je třeba technologie ZigBee. [23]

I přes to, že se tyto verze od sebe v základních principech příliš neliší, se budu dále výhradně zabývat standardizovanou verzí MQTT 3.1.1.

2.1 Základní topologie sítě komunikující pomocí MQTT

MQTT protokol vychází z publish-subscribe modelu řízeného centrálním serverem, který se zde nazývá broker. Ten dostává zprávy od poskytovatelů zpráv (publisherů) a přeposílá je těm odběratelům (subscriberům), kteří se k odběru daných zpráv přihlásili.

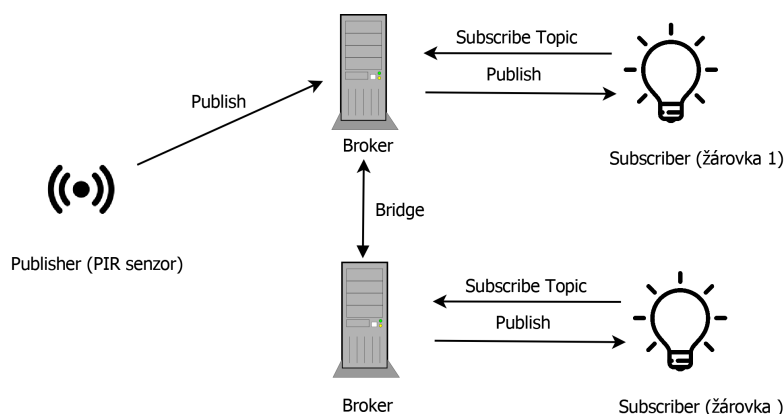
Jako praktický příklad můžeme uvést jednoduchou aplikaci, kdy snímač teploty odesílá aktuální naměřené hodnoty brokerovi, ten je přeposílá elektrickému pohonu okna, který posílá brokerovi požadavek, aby mu tyto zprávy odesílal. Na základě obdržené zprávy (příchozí hodnoty) se okno buď otevře, nebo zavře.



Obrázek 2.3: Základní schéma přenosu dat rozvržení sítě s využitím MQTT

Jednotliví klienti, ať už se jedná o publishery nebo subscribery, mohou odesílat i přijímat vícero zpráv současně a dokonce je možné, aby klient poskytoval zprávy o jednom tématu a zároveň poskytoval zprávy o jiném tématu, což znamená, že je současně v roli subscribera i publishera.

V topologie sítě je možné použít 2 i více brokerů. K tomuto účelu se používá tzv. bridge. Stačí nakonfigurovat alespoň jeden ze dvou spolu komunikujících brokerů. Při nastavení je možno sdílet pouze vybraná nebo všechna témata. Následně se mohou klienti přihlašovat ke všem sdíleným tématům nezávisle na tom, ke kterému brokeru jsou připojeni. [24]



Obrázek 2.4: Schéma přenosu dat sítě s využitím MQTT při rozvržení s dvěma brokery

2.2 Hierarchie zpráv - dělení témat

Zprávy jsou tříděny do tzv. témat (topic). Každá zpráva je přiřazena právě k jednomu tématu. Jednotlivá témata můžeme hierarchicky vnořovat do sebe a jednotlivé vrstvy oddělovat dopředným lomítkem /. Nejjednodušší zde bude uvést příklad, ze kterého bude strukturování hned jasné. Řekněme, že v místnosti 1 v 2. patře domu číslo 4 je umístěn senzor s označením číslo 2. Obsah dat, která senzor odesílá, tak můžeme přiřadit do tématu

„dům1/patro2/místnost1/senzor2“ nebo můžeme zvolit jiné rozdělení jako je třeba „senzory/senzor2/. Z příkladu je jasné, že hierarchie témat není přesně dána a záleží zcela na volbě projektanta, jak strukturu navrhne.

Pro přihlášení k odběru zpráv se postupuje podobně. Klient zavolá příkaz *subscribe* s názvem požadovaného tématu např.: „senzory/senzor2/ a při bezchybné komunikaci obdrží zprávu, kterou senzor2 poslal. Při přihlašování k odběru je možno použít tzv. zástupné znaky jako + nebo #. Znaménko + nahrazuje úroveň hierarchie témat. Takže pro téma „dům4/patro2/+světlo“ pošle všechny zprávy světel ve všech místnostech v druhém patře domu číslo 4. Zástupný znak # musí být vždy na konci volané struktury, ale může nahrazovat 1 i více úrovní. Takže například téma „dům4/patro2/# pošle všechny zprávy od všech přihlášených *publisherů* v druhém patře domu číslo 4.

Pro celistvost je dobré ještě doplnit, že znak \$ je vyhrazen pro systémová témata, která publikuje sám broker. Jako příklad můžeme uvést témata začínající \$SYS/.

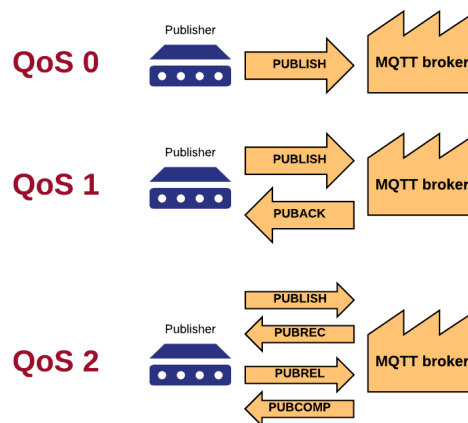
2.3 Obsah zpráv

Samotný obsah zpráv není přesně specifikován. Jednoduše řečeno taková zpráva, kterou dokáže odběratel přečíst, je doručena. Často se však používá struktura zpráv ve formátu JSON (JavaScript Object Notation), BSON (Binary JSON) nebo textové zprávy. Velikost zpráv nesmí překročit 256 MB. [22] Tato maximální hranice bývá zřídka omezující, protože při aplikaci v IoT se zpravidla přeposílají jednoduché zprávy nenáročné na datovou velikost.

2.4 Nastavení odesílání zpráv

2.4.1 Quality of Service

Pro snížení zatížení sítě definuje MQTT tři úrovně potvrzení zpráv (Quality of Service - QoS).



Obrázek 2.5: Grafické znázornění Quality of Servis MQTT [3]

- **Nejnižší úroveň QoS 0:** Zpráva je odeslána a nečeká na potvrzení. (at-most-once)
- **Střední úroveň QoS 1:** Zpráva je doručena alespoň jednou. (at-least-once)
- **Nejvyšší úroveň QoS 2:** Zpráva je doručena právě jednou. (axactly-once)

Klient nemusí podporovat všechny tři úrovně QoS, protože to protokol nenařizuje. V případě, že nastane situace, kdy klient nepodporuje nastavenou úroveň, automaticky použije takovou, která je dostupná. Logicky se bude jednat o nižší úroveň.

■ 2.4.2 Retained Message

U zpráv je možno nastavit tzv. retained flag. Pokud tuto vlastnost nastavíme na true, pak broker u daného tématu zprávu s tímto nastavením uloží a při přihlášení nového klienta k tomuto tématu mu ihned přepośle danou zprávu. Každé téma může mít pouze jednu *retained message*, takže broker má vždy uloženou pouze poslední zprávu s tímto nastavením. [25]

■ 2.4.3 Last Will Testament

Protokol myslí i na případ, kdy se publisher klient neočekávaně odpojí od sítě. Pokud je zařízení připojeno, neustále posílá v definovaných časových intervalech zprávu keep-alive, jestliže tuto zprávu broker neobdrží, myslí si, že spojení bylo přerušeno, a pošle tzv. závěť (Last Will Testament), což je zpráva, která definuje, co má zařízení vyslat do sítě při odpojení. Tato vlastnost je užitečná hlavně tehdy, pokud na datech vypnutého zařízení je závislá činnost ostatních klientů. [3]

■ 2.5 Zabezpečení přenosu zpráv

Ve výchozím nastavení je přenos zpráv nezabezpečený a nešifrovaný. Přihlášení klienta probíhá pomocí ClientID. Od verze 3.1 [26] je možné přidat do paketu uživatelské jméno a heslo jako určitý způsob zabezpečení. K šifrování přenosu zpráv je možné využít SSL/TLS protokol s využitím SSL certifikátu. [22]

Kapitola 3

Domácí automatizace pomocí Open-Source

Návrh řídicího systému chytrého domu (Smart Home) není jednoznačně určen a vždy se odvíjí od požadavků, které by měl splňovat. Pro komplexní řešení, kdy chceme chytrá zařízení řídit pomocí jednoho uživatelského rozhraní, musíme využít centrální server, kterému se říká Smart Home Hub. Hub označuje nejen hardwarovou část nýbrž se přeneseně používá tento termín i pro softwarovou část. Vhodnější je ale používat pro software řídicího systému domu označení Home Automation Software. Důležité kritérium, dle kterého bychom měli "centrální uzel" domácí automatizace vybírat, je primárně kompatibilita s různými zařízeními a technologiemi jako jsou například ZigBee, Zwave, stále se rozrůstající platforma Tuya nebo zařízení využívající pro svou komunikaci protokol MQTT. Další důležité kritérium, nad kterým bychom se měli zamyslet, je, kdo stojí za vývojem dané platformy. V ideálním případě je to velká stabilní firma s dlouhou historií, která ale bude primárně prosazovat své produktové portfólio, a nebo bychom se mohli poohlédnout po open-source řešení, za kterým stojí velká aktivní komunita lidí. Při dodržení těchto požadavků by nás vybraný Smart Home Hub neměl omezovat a zužovat náš výběr kompatibilních zařízení, která do systému připojíme, a měl by zaručit udržitelnost systému ve vývoji i v tak rychle proměnlivém oboru, jakou domácí automatizace v současnosti je.

3.0.1 Open-Source

V předchozím odstavci jsem se zmínil o otevřeném systému (open-source). Protože se budu dále primárně zaměřovat na projekty vyznávající tuto politiku distribuce, je vhodné si tento termín blíže popsat a definovat.

3.0.2 Historický vývoj Open-Source

Historicky se vývoj počítačů a programování nejdříve odehrával v akademickém prostředí, kde nově vzniklý software byl volně šiřitelný v souladu s akademickými zásadami sdílení znalostí. Postupem času v 70. a hlavně 80. letech začalo vznikat čím dál více komerčních firem vyvíjejících a prodávajících software (tzv. proprietární software). [27] V akademických sférách byl stále kladen důraz na otevřenost systému. Dokazuje to například Donald Knuth ze Stanfordovy univerzity, který roku 1978 publikoval sázecí software (ang. typesetting software) TeX, [27] z kterého vychází například i LaTeX, ve kterém je psaná i tato práce. [28] Důležitým milníkem byl vznik GNU volně šiřitelného operačního systému s názvem GNU Project, který roku 1983 publikoval Richard Stallman, chtěl tím poukázat na důležitost otevřených systémů. Při publikaci sepsal také dokument nebo lépe řečeno licenci, ve které popisuje zásady volného šíření programu. Na tomto základě vznikla GNU General Public Licence, [29] kterou například používá Linus Torvalds pro volnou distribuci Linuxu. [30] Celý projekt GNU stál za vznikem hnutí, které se označuje The Free Software Foundation [31] a které podporuje volnou distribuci softwaru. Ale termín open-source se začal formovat až po vydání práce Ericem S. Raymondem v roce 1997 v originále s názvem: The Cathedral and the Bazaar. [27] V dnešní době se používá termín open-source také pro projekty zabývající se vývojem hardwaru, které disponují volně dostupnou kompletní dokumentací vývoje daných produktů.

3.0.3 Definice Open-Source

Pro zajímavost zde uvádím volný překlad definice slovního spojení Open-Source z Oxford Dictionary of English:

“Označení softwaru, pro který je původní zdrojový kód volně dostupný a může být redistribuován a modifikován.” [32]

3.1 Hardware - jednodeskový počítač

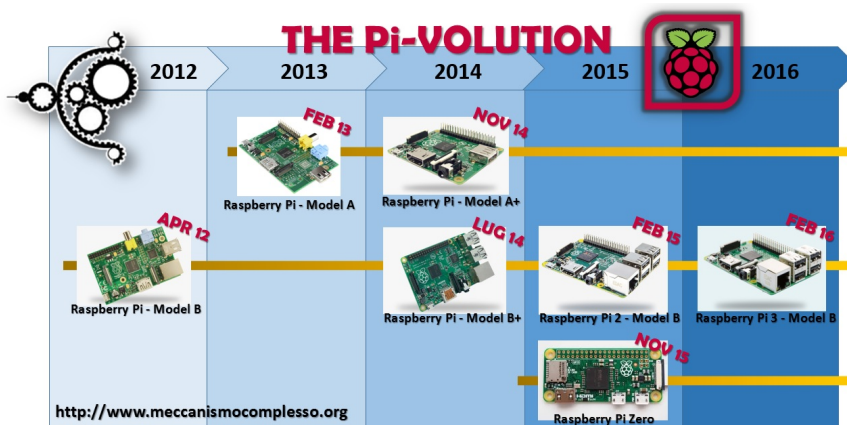
V úvodu této kapitoly jsem popsal základní prvky potřebné pro sestavení komplexního řídicího systému chytrého domu. Jedním z prvků je centrální jednotka, která potřebuje určitý výpočetní výkon, jenž dokáže zpracovávat všechny vstupní signály, vizualizovat je a na základě těchto vstupních signálů ovládat a regulovat výstupní zařízení a chytré spotřebiče v domě. Touto jednotkou může být například PLC (Programovatelný logický automat), který je spíše určen do průmyslového prostředí, kde díky své odolnosti a vysoké spolehlivosti splní náročné požadavky průmyslových aplikací. Právě kvůli vysokým požadavkům a nutné certifikaci jsou PLC drahou položkou v rozpočtu. V domácnosti tak vysoké nároky nemáme, a proto můžeme použít klasický počítač, který ale není optimalizovaný pro celodenní provoz a má tak příliš vysokou spotřebu a navíc má jiné vstupní a výstupní periferie, než jsou u centrální řídicí jednotky řídicího domu požadovány. Proto začali vznikat tzv. Smart Home Huby, které jsou určeny pro tyto účely. Huby jsou již hotová řešení často již s nainstalovaným proprietárním softwarem určité společnosti. Nebo můžeme zvolit svou vlastní cestu a za zlomek ceny si koupit jednodeskový počítač, na který si řídicí systém nainstalujeme sami. Tuto cestu jsem zvolil já pro praktickou část své práce. Proto v další podkapitole předložím rešerši na vybrané jednodeskové počítače vhodné pro použití řídicí jednotky chytrého domu.

3.1.1 Raspberry pi

Historie

Za zrodem tohoto často skloňovaného "mini počítače" stojí jeden ze zakladatelů Eben Upton, který po dosažení svého doktorského titulu na Cambridgeské univerzitě v roce 2006, začal sestavovat "mini počítač" využívající čip ATMEL Atmega 644, což je 8 bitový čip s AVR architekturou. Jednalo se tedy o velmi jednoduchý počítač, který bychom zařadili spíše do kategorie mikrokontrolerů. Po nástupu do společnosti Broadcom zabývající se vývojem elektroniky pochopil, že jeho navržený počítač je příliš málo výkonný pro naplnění požadavků i tehdejší doby. Ale díky svému prvnímu návrhu získal řadu kontaktů, které využil při svém druhém projektu. Další projekt začal pouze jako sestavení počítače podobného historicky známému britskému počítači BBC Mikro, ale v průběhu realizace přišel nápad na vytvoření cenově dostupného jednodeskového počítače pro výukové a vývojové účely. Tento nápad se později ukázal jako geniální myšlenka. [33] První oficiální představení Raspberry Pi se konalo

v únoru 2012, kdy byl představen model B odkazující svým názvem právě na počítač BBC Mikro. Již od počátku chtěla Raspberry Pi Foundation nabízet dvě verze, výkonnější model s označením B a cenou přibližně 35\$ a levnější model A se startovací cenou 25\$. Verze A byla nahrazena v roce 2015 ještě levnější variantou Zero, která se prodává již od ceny 4£. [34]



Obrázek 3.1: Zjednodušený historický vývoj "mini počítačů" Raspberry Pi [4]

Ke konci roku 2018 bylo prodáno přes 23 milionů počítačů s tímto úsměvným názvem. [35]

Jméno Raspberry vzniklo v raném stádiu vývoje, kdy jeden ze zakladatelů při mailové komunikaci pro pobavení svých kolegů napsal, zda počítači nebudou raději říkat raspberry (ang. malina). Toto označení se postupně vžilo a časem bylo doplněno slovo Pi odkazující na první testovací aplikace vyvíjené v jazyce Python. [33]

■ Operační systém

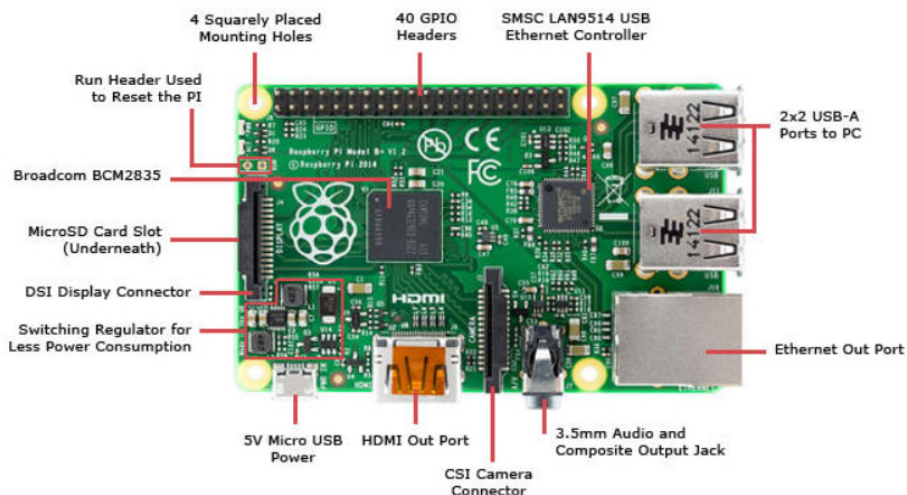
Rasbian je jediný oficiálně podporovaný operační systém, který Raspberry Pi Foundation poskytuje. Tento systém vznikl krátce po představení Raspberry Pi v červnu roku 2012. První verzi vytvořil Mike Thompson a Peter Green. [36] Nadace jej začala oficiálně podporovat až v roce 2015. Zjednodušeně řečeno je Raspbian odlehčená verze linuxové distribuce Debianu. [37]

Jedna z velkých výhod počítačů Raspberry Pi je široká škála kompatibilních operačních systémů pro různá využití. To znamená, že nemusíte zůstat u oficiálně podporované varianty OS Raspbianu, ale můžete využít OS třetích stran. Na některé dokonce odkazuje oficiální web raspberrypi.org. [38] Pro zajímavost bych uvedl například Windows 10 IoT Core pro řízení chytrých

zařízení využívající platformu Windows, LibreELEC s následným doinstalováním open-source aplikace KODI pro video entertainment platformu nebo Weather Station pro řízení domácí meteostanice.

■ Vstupní a výstupní periferie

Další výhodou, díky které se Raspberry Pi tak masově rozšířilo, je obsáhlý balíček vstupních a výstupních periferií, kterými disponuje. Proto je jej možné využít jako plnohodnotný stolní počítač se základními prvky jako jsou monitor, klávesnice a myš bez nutnosti použití rozšiřujících modulů. Proto jej řadíme mezi tzv. jednodeskové počítače anglicky označovaných jako Single Board Computers (SoC). [39]



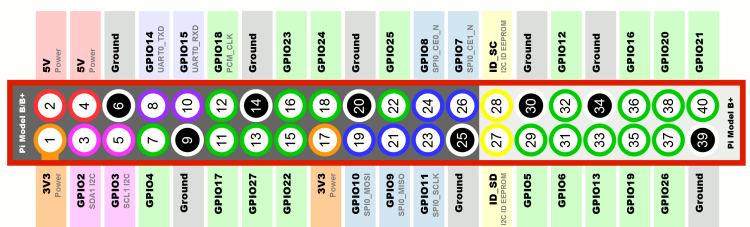
Obrázek 3.2: Popis vstupních a výstupních periferií na Raspberry Pi 3B [5]

Pro řízení a snímání digitálních signálů je deska osazena GPIO piny, jež daly dobrý základ nejen výukovým účelům, ale také jsou používány pro rychlý vývoj řídicích systémů menších projektů. Díky velké oblíbenosti se počet GPIO pinů rozrostl z původních 26 na současných 40.

Kvůli historickému vývoji jsou používána dvě číslování (knihovny) GPIO pinů. Dnes je ale mnohem používanější BCM (Broadcom SOC Channel) setup místo BOARD setupu.

- BOARD - pojmenování pinů jak jdou na desce číselně v řadě za sebou 1-40

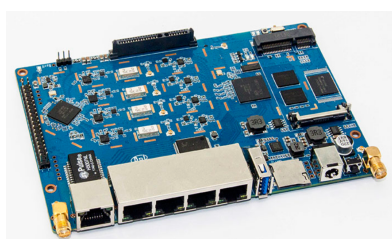
- BCM - pojmenování pinů se odvíjí podle toho jak je deska navržena a koresponduje s číslováním pinů podle procesoru Broadcom (číslování odpovídá číslům za označením GPIO) [40]



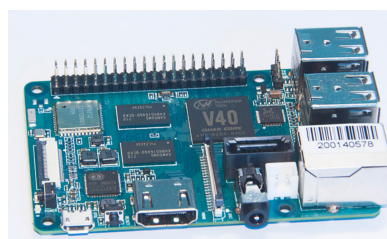
Obrázek 3.3: Schéma GPIO pinů pro Raspberry Pi 3B+ [6]

3.1.2 Banana Pi

Projekt Banana Pi začal vznikat na začátku roku 2013 jako reakce na rozrůstající se komunitu Raspberry Pi a na jí poptávanou vyšší výkonost. První desky Banana Pi šly do prodeje v dubnu roku 2014. [41] Za jejich vývojem a prodejem stojí čínská společnost Shenzhen SINOVOIP Co. Ltd. Firma ctí politiku open-source hardware. [42] K hlavním výhodám patří možnost výběru z široké modelové řady, což umožňuje vybrat vhodnou variantu pro specifické projekty. Velkou konkurenční výhodou je nabídka desek navržených pro aplikaci routerů, které většina ostatních výrobců postrádá. Například model BPI-R64 má 5 ethernetových portů a je jasně určen pro aplikaci routeru. [43] Jako přímého konkurenta nejvyšší řady současného Raspberry Pi můžeme označit model BPI-M2 Berry. [44]

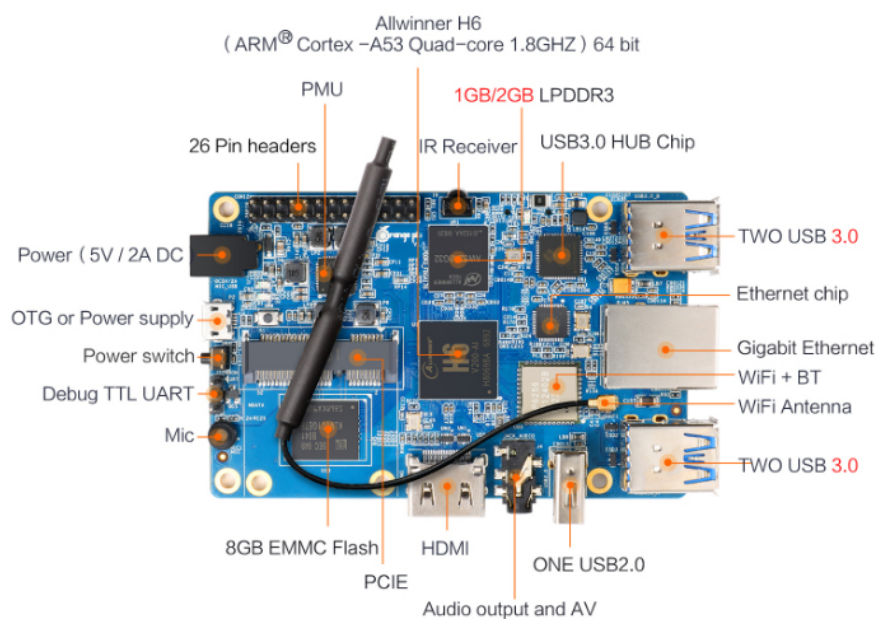


(a) : Banana Pi BPI-R64 [43]



(b) : Banana Pi BPI-M2 Berry [44]

Obrázek 3.4: Vybrané jednodeskové počítače značky Banana Pi



g

Obrázek 3.5: Orange Pi 3 [7]

3.1.3 Orange Pi

První jednodeskové počítače šly do prodeje ke konci roku 2015. Ve srovnání produktů s ekvivalentními výkonovými parametry obou výše zmíněných konkurentů jsou produkty Orange Pi cenově nejdostupnější, i když rozdíl je řádově v jednotkách dolarů. Za jeho otevřeným vývojem a prodejem stojí opět firma z Číny s názvem Shenzhen Xunlong Software CO., Limited. [45] Výraznější odlišnosti bychom hledali obtížně. Uvedl bych možnost konfigurace varianty s vestavěnou eMMC Flash pamětí vnitřního úložiště nebo použití sousého napájecího konektoru 5,5/2,1mm, který je primárně určen pro napájení na rozdíl od mikro-B USB, které využívají konkurenti. Ve většině článků je tento parametr uváděn jako výhoda, protože při připojování rozšiřujících modulů k hlavní desce není potřeba rozšiřující modul napájet separátním napájecím zdrojem. V podobné cenové relaci jako se pohybuje Raspberry Pi, nabízí Orange Pi model 3. [7]

3.1.4 Porovnání technických parametrů

V této části jsem zpracoval tabulku s porovnáním technických parametrů nejnovější výkonné verze Raspberry Pi s označením 3B+ s typově i cenově ekvivalentními modely výrobců Orange Pi a Banana Pi. V tabulce naleznete

i ceny jednotlivých zařízení, které byly brány ze zahraničních serverů, proto jsou uvedeny v dolarech.

Parametry	Raspberry Pi 3B+	Orange Pi 3	Banana Pi RPI-2M
Procesor	1.4 GHz 64-bit quad-core	1.8 GHz 64-bit quad-core	1.2 GHz 32-bit quad-core
Uložiště	mikro SD	8GB eMMC + mikro SD	mikro SD + SATA
RAM	1GB DDR2	2GB DDR3	1GB DDR3
GPU	Broadcom IV @400MHz	Mali T720 @650MHz	Mali 400 @500MHz
Ethernet	10/100/1000Mbps	10/100/1000Mbps	10/100/1000Mbps
WiFi	802.11 b/g/n/ac	a/b/g/n/ac	802.11 b/g/n
USB porty	4x USB 2.0	4x USB 3.0	4x USB 2.0
GPIO piny	40	26	40
IR receiver	NE	ANO	NE
Cena	35\$	35\$	36\$

Tabulka 3.1: Tabulka s porovnáním technických parametrů jednodeskových počítačů Raspberry Pi, Orange Pi a Banana Pi

3.2 Software

Dalším důležitým prvkem řídicího systému je samotný software, který umožňuje sdružovat chytrá zařízení a následně je všechna ovládat a automatizovat v jedné aplikaci. Často používané označení těchto softwarů je Home Automation Software česky volně přeloženo jako software domácí automatizace. V následující podkapitole bych rád detailně popsal dva nejrozšířenější otevřené systémy domácí automatizace a na závěr se zmíním o oblíbeném komerčním řešení.

3.2.1 Home Assistant

Zakladatelem otevřeného řídicího systému pro chytrý dům Home Assistant je holandský programátor Paulus Schoutsen. Svými slovy popisuje začátek projektu jako postupné zkoumání možností chytrých žárovek Philips Hue, kdy velice rychle narazil na hranice ovládání systému Philips, a tak si začal psát vlastní skripty v Pythonu 3, protože se chtěl právě v tomto jazyce zdokonalit.

[46]



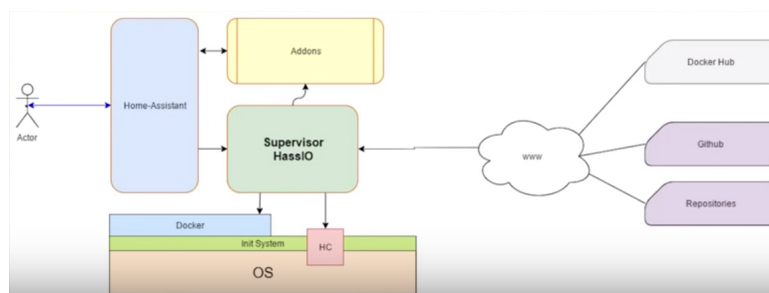
Obrázek 3.6: Logo Home Assistant [8]

Přibližně v září roku 2013 zveřejnil kód svého projektu na GitHubu. [47] Postupem času si začali Paulusova projektu všimati další vývojáři, testovali jej a navrhovali vylepšení. Velkou zásluhu na rozšíření komunity má sám zakladatel, jelikož se neustále snaží vylepšovat kanály komunikace mezi vývojáři a uživateli, aby tak mohl celý projekt růst rychleji a samovolně. [46]

■ Instalace Home Assistanta

Postupným vývojem a zjednodušováním pro koncové uživatele vzniklo několik variant instalace. Když přeskóčíme prvotní vývojářskou etapu, tak za zmínku stojí níže zmíněné instalační postupy.

- **Install Package/Dependencies** - Tato metoda funguje na většině operačních systémů stojících na Linuxovém jádře. Nutnou podmínkou pro instalaci na serverovém hostiteli je podpora Pythonu 3. Po úspěšné instalaci Pythonu a jeho instalačního balíčku PIP můžete zadat příkazy pro samotnou instalaci Home Assistanta. [48]
- **HASSbian** - jak již název napovídá, jedná se o upravený operační systém pro Raspberry Pi Raspbian, který je odlehčen o nepotřené komponenty vč. uživatelského rozhraní a má přeinstalovanou základní konfiguraci Home Assistanta. HASSbian je tedy primárně určen pro Raspberry Pi konkrétně pro modely Zero, 2,3 i 3B+. Instaluje se naboťováním diskového obrazu na mikro SD kartu, kterou následně vložíme do zařízení. [49]
- **HASS.IO** - nejjednodušší na instalaci a i následnou správu celého systému je vlastní operační systém vytvořený pro hostování Home Assistanta HassOS, který má nainstalovanou verzi Home Assistanta s názvem Hass.IO. [50] Operační systém byl vytvořen pomocí nástroje BuildRoot, [51] který je určen pro tvorbu vlastního linuxového systému, a využívá software Docker, [52] pro zjednodušení správy a nasazování nových balíčků a vnitřních aplikací. Systém se instaluje podobně jako HASSbian přes diskový obraz. Tento systém má již volně stažitelné diskové obrazy pro většinu modelů Raspberry Pi nebo i další jednodeskové počítače jako Asus ThinkerBoard, Odroid, OrangePi a další. [50]



Obrázek 3.7: Schéma struktury operačního systému HassOS a HASS.IO [9]

■ Základní struktura systému Home Assistant

Nejdůležitější soubor pro koncového uživatele, ve kterém se nastavují a přidávají nová zařízení/služby, se jmenuje `configuration.yaml`, nachází se ve složce `config`. Zde najdete další důležité soubory jako například `automations.yaml`, který slouží pro definici pravidel automatizačních procesů.

YAML (YAML Ain't Markup Language) je serializační formát pro ukládání dat. Strukturování je řešeno pomocí odsazování. Důležitá poznámka: Vnoření o jednu úroveň je řešeno 2 nebo 4 mezerami, použití tabulátoru je bráno jako chyba. [53]

Při správném nastavení komunikace zařízení v souboru `configuration.yaml` začne Home Assistant odebírat data odesílaná zařízením a ukládat je do tzv. entity, která slouží pro softwarové označení skupiny dat daného zařízení. Z toho je patrné, že pokud se jedná o multifunkční zařízení nebo senzor, tak může odesílat více skupin dat, která se zvláště budou ukládat pod jednotlivé entity. Seznam všech nastavených přístupných entit nalezneme v sekci Vývojářské nástroje > Stav. Pro vizualizaci dat se ve výchozím nastavení využívá uživatelské rozhraní LovelaceUI [54]. Ovládací obrazovky je možné vytvořit i pomocí rozšiřujícího modulu HA Dashboardu, [55] který je vhodný pro vizualizaci dat na HMI panelech. Při vizualizaci využíváme výše zmíněných entit pro propojení potřebných dat vytvořených ovládacích prvků.

Velkou výhodou Home Assistanta je široká škála tzv. komponentů, které slouží pro napojení chytrých spotřebičů nebo zařízení používajících různé technologie jako jsou třeba Z-wave, Tuya nebo MQTT protokol. [56]

Instalování rozšiřujících balíčků u verze HASSbian se provádí přes příkazový řádek pomocí jazyka Bash. [57] U varianty instalace Hass.IO příkazový řádek není k dispozici, ale je v nabídce stále se rozšiřující knihovna s tzv. Add-Ons,

[58] kde můžeme rozšíření nainstalovat jednoduše v uživatelském rozhraní. K nabídce rozšíření patří mimo jiné i MQTT broker Mosquitto.

Díky aktivní komunitě přicházejí systémové aktualizace přibližně jednou za 14 dní, což je velká výhoda oproti konkurenci, ale může vést k častému výskytu chyb při rané aktualizaci.

■ 3.2.2 OpenHab

Zakladatelem dalšího otevřeného řídicího systému pro chytrý dům OpenHAB je německý softwarový architekt Kei Kreuzer. Sám uvádí, že o domácí automatizaci se zajímá již od malička. Proto postupným rozvojem svých schopností a získáním pracovních zkušeností v oblasti IoT v roce 2010 založil open-source projekt OpenHAB. [59] Jeho název je zkratkou z anglických slov Open Home Automatin Bus. [60]

Celý systém je napsaný v jazyku JAVA s využitím modulární platformy OSGi. [60] Podobně jako Home Assistant nabízí i OpenHAB několik možných způsobů instalace. Od Linuxových balíčků (Packages) až k nejkompexnějšímu řešení pro koncové uživatele v podobě diskové obrazu upraveného Raspbianu s předinstalovaným OpenHABem, který se jmenuje OpenHABian a je opět primárně určen pro zařízení Raspberry Pi. Tuto variantu používá většina současných uživatelů. [61]



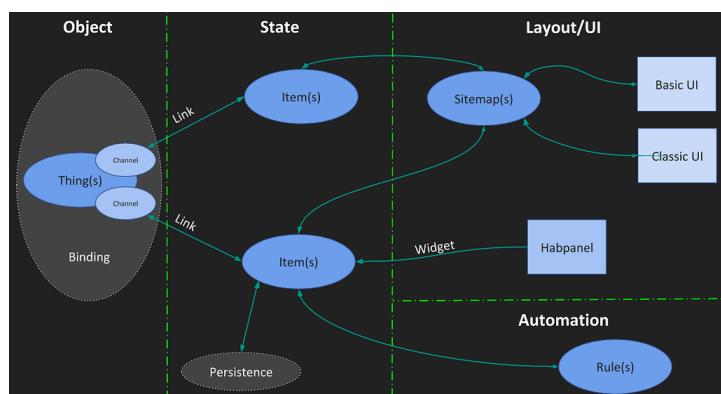
Obrázek 3.8:
Logo OpenHAB
[10]

■ Základní struktura systému OpenHAB

OpenHAB má přesně definovanou strukturu pro práci s daty. Jednotlivá připojená zařízení se nazývají Things. Pro připojení Things musíte nastavit Bindings, což je buď rozhraní komunikační technologie jako je například Z-wave nebo MQTT nebo to může být konektor pro určitou konkrétní značku a typ zařízení, jakými jsou například žárovky Philips Hue. Významově Bindings odpovídá Components u Home Assistanta. Každé Things má jedno a více Channels, které reprezentují ovládatelné vlastnosti připojených zařízení, což může být například barva světla žárovky a její intenzita svítivosti (zařízení tedy může mít jedno i více Channels). Pro další práci s daty a jejich vizualizaci nastavíme Items, to znamená, že napojíme jeden Channel nebo více Channels v závislosti na tom, zda se jedná o základní typ Item, nebo o Group Item.

Pro správné zobrazení dat ještě musíme definovat, o jaký typ Items se jedná. Tyto informace je možné vyčíst z oficiální dokumentace OpenHABu. [62] Po nastavení souboru default.sitemap se zobrazí již nadefinovaný přehled využívající Items pro napojení jednotlivých ovládacích prvků v sekci ClassicUI. Po doinstalování HABpanelu si můžeme vytvořit vlastní uživatelsky přívětivý ovládací panel, [63] který je dobře optimalizovaný pro tablety a může tak sloužit jako vhodné řešení pro HMI panely.

Na rozdíl od Home Assistanta vydává OpenHAB otestované aktualizace přibližně jednou za půl roku. Takže z tohoto hlediska jej můžeme považovat za stabilnější platformu, která ale nemá tak velkou knihovnu již hotových konektorů chytrých zařízení. Další výhodou může být právě výše zmíněný HABpanel, který je velice intuitivní a jednoduchý na používání.



Obrázek 3.9: Schéma konfigurace struktury komponentů v OpenHABu [11]

3.2.3 SmartThings

Celým názvem Samsung SmartThings je označení pro ekosystém chytrých zařízení od této jihokorejské firmy. [64]



Obrázek 3.10: Logo SmartThings [12]

Samsung v roce 2014 odkoupil úspěšnou americkou firmu SmartThings vyvíjející řídicí systém pro chytré domy [65], a tak rychleji vstoupil do světa IoT a domácí automatizace.

Nejedná se o open-source produkt. Otevřenost platformy je do jisté míry v tom, že nabízí dokumentaci pro tvorbu rozšiřujících modulů a konektorů zařízení třetích stran. [66] Díky tomu má SmartThings velkou podporu zařízení různých značek. [67] Celý systém je možné ale provozovat pouze s originálním Samsung Hubem.

Toto řešení jsem zde uvedl jednak pro srovnání a jednak proto, že Smart-Things patří k celosvětové nejprodávanějším komerčním Hubům pro chytrý dům [68] a vyhrál většinu aktuálních srovnávacích testů komerčních řešení. [69]

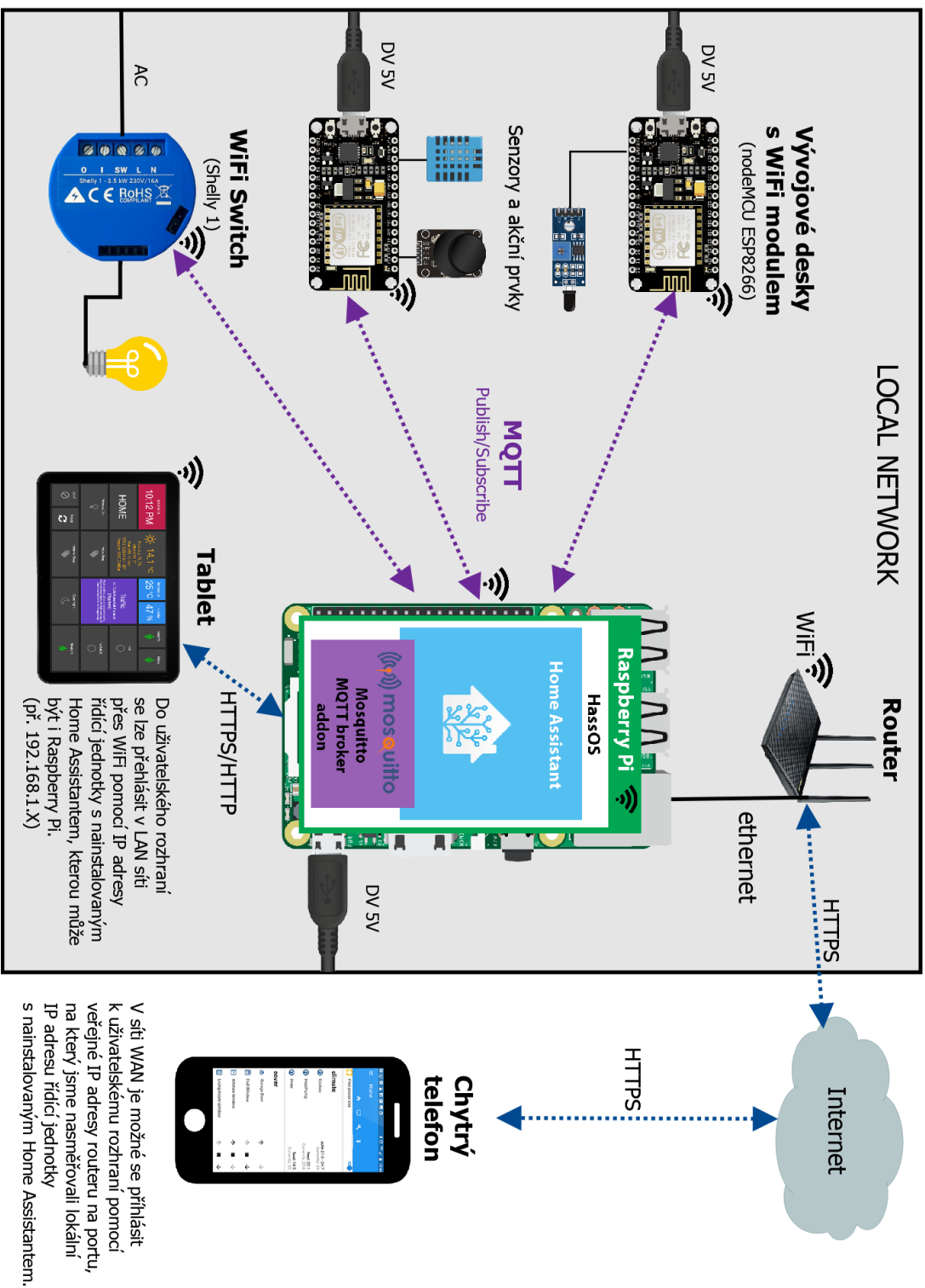


Kapitola 4

Návrh řídicího systému chytrého domu s využitím Open-Source

Po důkladné rešerši současných open-source řešení pro hardware i software, která je zpracovaná v předchozí kapitole, jsem pro praktickou část bakalářské práce zvolil jednodeskový počítač Raspberry Pi a software domácí automatizace Home Assistant (varianta HassOS) a to hlavně díky široké komunitě, která za oběma projekty stojí.

V následující kapitole zpracuji schématický návrh komunikace řídicích a ovládacích prvků v síti využívající MQTT protokol včetně detailního popisu jednotlivých částí.



Obrázek 4.1: Detailní schéma komunikace řídicích a ovládacích prvků řídicího systému chytrého domu

■ Řídicí jednotka

Za hlavní uzel sítě, kde je nainstalovaný operační systém HassOS s Home Assistantem vč. přidaného rozšíření (Add-on) Mosquitto Brokera, jsme zvolili Raspberry Pi 3B+, které je připojeno pomocí do sítě pomocí ethernetového vedení. Je možné využít variantu s WiFi, ale pro rychlejší komunikaci je doporučeno využít metalické vedení. Při nastavování komunikace dalších zařízení budeme odkazovat v síti na centrální jednotku, musíme ji tedy nastavit statickou IP adresu. Toto pravidlo platí pro všechna připojená zařízení, protože pro správnou integraci a vizualizace dat v Home Assistantovi budeme podobně odkazovat na zařízení, která data posílají. Proto potřebujeme zajistit stabilní IP adresu používaných zařízení.

■ Bezdrátové komunikační moduly

Abychom pro přenos dat mohli využít bezdrátovou MQTT komunikaci, musí mít každý síťový uzel procesor s WiFi modulem. A to ať už se jedná o jakýkoliv chytrý spotřebič nebo vývojový kit, který jsem použil ve schématu. K vývojovým kitům je možné pomocí propojovacích vodičů připojit senzory a akční prvky, které je následně možné bezdrátově ovládat vývojovým kitem, do něhož je potřeba nahrát potřebný firmware s nastavením WiFi komunikace včetně MQTT implementace a posíláním dat z GPIO pinů, které jsou na desce využity pro sbírání senzorických dat nebo akčních prvků. Já jsem použil oblíbený mikrokontroler s vestavěným WiFi čipem ESP8266.

Ve schématu jsem také znázornil WiFi switch (spínač), což je jednoduché zařízení pro bezdrátové spínání elektrických obvodů. Díky své ceně jsou rozšířené spínače značky Sonoff. Já jsem využil sofistikovanější variantu bulharské firmy Shelly. [\[70\]](#)

■ Ovládací prvky

Pokud se nacházíte v lokální síti, kde je umístěna i řídicí jednotka, připojíte se k uživatelskému rozhraní administrace pomocí IP adresy řídicí jednotky. Za tuto adresu je ještě nutno dodat číslo portu 8123 (psáno dvojtečka a číslo portu), který je ve výchozím nastavení určen pro přihlášení do administrace Home Assistanta.

Pokud chcete ovládat nebo sledovat aktuální data mimo lokální síť řídicího serveru je nutné v první řadě nastavit vzdálený přístup a navíc je vhodné využít šifrování pro bezpečný přenos. Šifrovaný přenos lze nejjednodušeji nastavit přes DuckDNS rozšíření, které využívá SSL protokol. Postup je následující. Založíme si zdarma účet na duckdns.org. [71] Po úspěšném přihlášení si zvolíme libovolnou volnou doménu 3. řádu například: mojedomana.duckdns.org, kterou nasměrujeme na veřejnou adresu našeho routeru. Zkopírujeme token účtu a ten vložíme do inicializačního kódu již nainstalovaného rozšíření v administraci Home Assistanta, kde také vložíme jméno celé domény. Údaje vkládáme vždy mezi uvozovky. Pro dokončení je nutné ještě upravit configuration.yaml soubor, kam musíme vložit následující kód.

```
http:
  base_url: https://my-domain.duckdns.org:8123
  ssl_certificate: /ssl/fullchain.pem
  ssl_key: /ssl/privkey.pem
```

Tím jsme nastavili šifrovanou komunikaci využívající protokol HTTPS místo HTTP. Od tohoto okamžiku tedy musíme zadávat https:// před IP adresu řídicí jednotky i pro přihlášení v lokální síti.

Pro vzdálený přístup je nutné ještě nastavit na routeru přesměrování portu 8123 na lokální IP adresu řídicí jednotky. Toto nastavení se dělá v administraci routeru, kterou má každý výrobce mírně odlišnou.

Po zadání adresy https://mojedomana.duckdns.org:8123 se dostaneme na přihlašovací plochu Home Assistanta. Stejně tak bude fungovat URL adresa s využitím veřejné IP adresy, kterou nám přidělil náš internetový poskytovatel https://vaše_veřejná_IP_adresa:8123. Pro fungování dalších služeb přes vzdálený přístup, tedy pokud jsme mimo lokální síť řídicí jednotky, je nutné nasměrovat další používané porty v nastavení routeru. Typicky se jedná o port 1883 pro MQTT broker, 3218 pro vestavěný editor souborů Configurator a další.

Kapitola 5

Připojení senzorů a akčních prvků

V této kapitole detailně popíši a sám otestuji připojení vybraných senzorů a vývojového kitu, který bude přeposílat sensorická data do centrální jednotky řídicího systému. Při realizaci budu vycházet z navrženého schématu vypracovaného v předchozí kapitole. [4]

5.1 Akční prvky MQTT sítě a jejich připojení

5.1.1 Čipy Espressif

V předchozí kapitole jsme se zmínili o čipu ESP8266. Nyní se podrobněji podíváme na čipy firmy Espressif Systems, která stojí za jejich vývojem a včetně jejich derivátů. Následně připravíme a nahrajeme firmware pro tyto čipy, abychom mohli nastavit bezdrátovou komunikaci s řídicím systémem využívajícím protokol MQTT.

Historie společnosti Espressif Systems se datuje od roku 2008. Teprve v roce 2010 založila své vlastní vývojové oddělení. [72] Svůj první čip vydala v prosinci 2013 s označením ESP8089. Jednalo se o mikrokontroler s vestavěnou WiFi určený pro tablety a set-top boxy. Hned v květnu roku 2014 byl představen další produkt



Obrázek 5.1:
Logo Espressif
Systems [13]

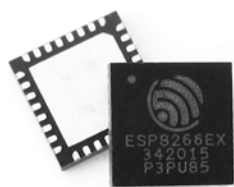
ESP8266 podobný předchozímu čipu s rozšířením o vstupy a výstupy a navrženým primárně pro IoT zařízení. [72] Tento čip se díky své ceně, malým rozměrům a integraci většiny potřebných komponent do jednoho čipu stal hotovým bestsellerem.

Vzhledem k tomu, že se jedná o open-source hardware, existuje velké množství čipů a vývojových kitů vycházejících ze stejné architektury od jiných výrobců (AI-Thinker, Wemos, Adafruit atd.).

Dnes čip ESP8266 nebo jeho nástupce ESP32 najdete ve velkém množství chytrých zařízení. I po představení čipu ESP32 je častěji používán čip ESP8266 díky své ceně a dostačujícím parametrům pro většinu jednoduchých spotřebičů. Jeho nástupce je používán spíše pro náročnější IoT aplikace.

Zajímavostí je, že firma Espressif Systems má po celém světě pouze 3 pobočky a jedna z nich sídlí od srpna 2017 v Brně. [72]

■ ESP8266EX



Obrázek 5.2:

Čip ESP8266EX

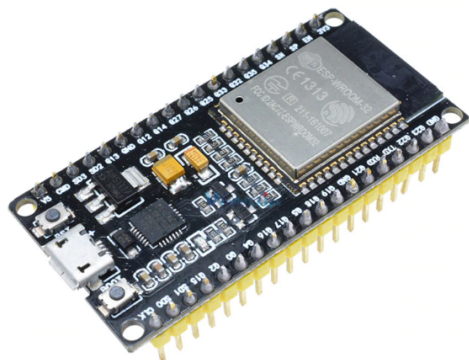
[14]

Pro vývoj, testování a výukové účely jsou spíše doporučeny vývojové kity využívající tento čip než použití čipu samotného. Nejrozšířenější je verze kitů NodeMCU, která má navíc integrovaný USB-to-TTL převodník, napájenou kolíkovou lištu, napěťový regulátor a další. Díky tomu je mnohem jednodušší programování čipu a připojení vstupních a výstupních prvků na GPIO piny. [73]

■ ESP32

V srpnu roku 2016 byl představen výkonnější model s označením ESP32. K hlavním výhodám patří např. dvoujádrový procesor, rychlejší WiFi, větší

počet GPIO pinů, Bluetooth, vestavěný senzor dotyku, senzor teploty, Hallův senzor. [74] Stejně jako u jeho předchůdce se staly velmi populární vývojové kity využívající tento čip.



Obrázek 5.3: Dev Kit NodeMCU ESP32 [15]

Specifications	ESP8266	ESP32
MCU	Xtensa® Single-Core 32-bit L106	Xtensa® Dual-Core 32-bit LX6 600 DMIPS
802.11 b/g/n Wi-Fi	Yes, HT20	Yes, HT40
Bluetooth	None	Bluetooth 4.2 and below
Typical Frequency	80 MHz	160 MHz
SRAM	160 kBytes	512 kBytes
Flash	SPI Flash , up to 16 MBytes	SPI Flash , up to 16 MBytes
GPIO	17	36
Hardware / Software PWM	None / 8 Channels	1 / 16 Channels
SPI / I2C / I2S / UART	2/1/2/2	4/2/2/2
ADC	10-bit	12-bit
CAN	None	1
Ethernet MAC Interface	None	1
Touch Sensor	None	Yes
Temperature Sensor	None	Yes
Working Temperature	- 40°C ~ 125°C	- 40°C ~ 125°C

Obrázek 5.4: Porovnání klíčových technických parametrů čipů ESP8266 a ESP32 [16]

5.1.2 Připojení ESP8266 do MQTT sítě včetně integrace do Home Assistanta

Aktivní člen sítě, který bude přeposílat data ze senzorů, máme již vybraný. Z výše zmíněných budu v praktické části využívat vývojový kit NodeMCU s čipem ESP8266. Aby datový přenos bylo vůbec možné zprovoznit, musíme nahrát firmware, který umožní komunikaci MQTT protokolu a využití GPIO pinů na vývojovém kitu. Tento proces nahrávání se často označuje anglickým výrazem tzv. flashování.

Zkompilovaný program, který budeme nahrávat, musí mít příponu `.bin` (soubor s daty v binární formě). Způsobů, jak soubor s `.bin` příponou vytvořit, je díky rozšířeným knihovnám od vývojových pracovníků Espressif systems a celé komunity hned několik.

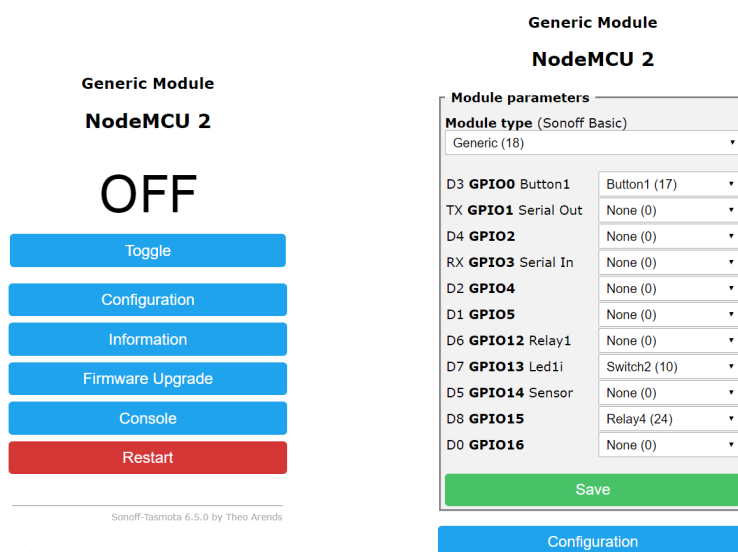
Pokud nepotřebujete naprogramovat specifické prvky pro svůj projekt, můžete využít již zkompilovaný volně stažitelný firmware, který můžete již rovnou "naflashovat" na jakýkoliv podporovaný procesor daného firmwaru. Pro tvorbu samotného programu se používá mezi kutily rozšířené vývojové prostředí ArduinoIDE [75], nebo Visual Studio Code s rozšiřujícím pluginem PlatformIO. [76] Tyto dva softwary umožňují po úspěšné kompilaci program jednoduše nahrát na čip pomocí USB portu na vývojovém kitu.

Mezi nadšenci a kutily v oblasti domácí automatizace se nejvíce rozšířil open-source firmware TASMOTA, [77] který je určen pro většinu zařízení využívajících čip ESP8266. TASMOTu si můžete stáhnout buď jako zip soubor obsahující potřebné knihovny a soubory nutné pro kompilaci v některém výše zmíněných vývojových prostředích, nebo si můžete vybrat z již zkompilovaných verzí a čip po zprovoznění firmwaru a přihlášení se k lokální síti naprogramovat pomocí over-the-air (OTA) programování ve webovém grafickém rozhraní, které je součástí firmwaru.

■ Nahrání firmwaru TASMOTA na vývojový kit NodeMCU

Pro samotné nahrání binárního kódu jsem použil software NodeMCU flasher. [78] Pro komunikaci vývojového kitu s počítačem je potřeba nainstalovat pro USB na TTL převodník systémové ovladače CP210X. [79] Následně pak v programu NodeMCU flasher vybereme USB port (značeno COM + číslo USB portu), ke kterému je připojeno zařízení s čipem ESP8266, vybereme místo uložení zkompilovaného `.bin` souboru a jednoduše spustíme nahrávání stisknutím tlačítka Flash. Po úspěšném dokončení procesu spustíme program Termite, [80] ve kterém nastavíme přístup k lokální síti a statickou IP adresu zařízení. Nakonec zařízení restartujeme. Pokud vše proběhlo v pořádku, vývojový kit se po restartování automaticky přihlásí k lokální síti.

Nyní můžeme využít webového grafického rozhraní s nastavením, které je přístupné na IP adrese, kterou jsme zadali v předchozím kroku. V nastavení zvolíme typ modulu, který je pro NodeMCU označen "Generic". V této sekci budeme následně přiřazovat funkce jednotlivým GPIO pinům v závislosti na připojených prvcích. Pro přenos dat ještě nastavíme údaje MQTT brokera (v našem případě IP adresu Raspberry Pi s nainstalovaným Home Assistantem



(a) : Úvodní obrazovka firmwaru Tasmota

(b) : Sekce nastavení GPIO pinů modulu Generic firmwaru Tasmota

Obrázek 5.5: Uživatelské rozhraní firmaru TASMOTA

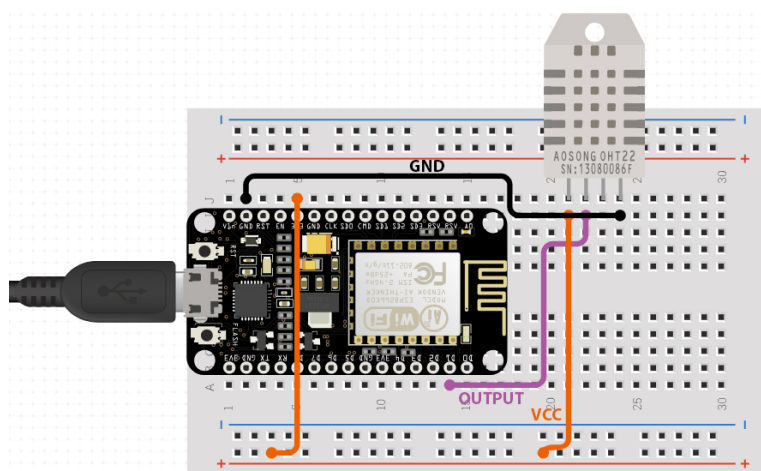
a případně uživatelské jméno a heslo, pokud jsme jej brokeru nastavili). Po každé úpravě nastavení musíme celý modul restartovat buď manuálně nebo přímo v grafického prostředí.

5.2 Připojení senzorů a nastavení jejich datové komunikace

Pro názornou ukázkou jsem použil senzory vhodné pro aplikaci chytrého domu z výukové sady 37 senzorů. [81] Pro každý prvek přikládám schéma zapojení k vývojovému kitu NodeMCU. Podobný postup je možné aplikovat na jakýkoliv řídicí člen s GPIO piny a je možné použít rozdílné číslování pinů při zachování vztahů pro napájecí, datové a zemnicí piny. Protože postup připojení jednotlivých prvků je velmi podobný, detailně jej popíši pouze u prvního senzoru a u dalších se budu spíše soustředit na schéma zapojení a implementační kód pro propojení komunikace s Home Assistantem nebo případně na definování automatizačních procesů.

5.2.1 DHT11/DHT22 (Digital Humidity and Temperature)

Senzory s označením DHT jsou určeny pro snímání teploty a vlhkosti. V současné době jsou v nabídce dvě řady: starší DHT11 a její nástupce DHT22. Rozdíl je pouze v přesnosti a rozsahu měření. Levnější DHT11 měří teplotu v rozsahu 0-50 stupňů Celsia s přesností ± 1 °C a vlhkost od 20% do 80% s odchylkou ± 4 %. Vyspělejší měřící modul DHT22 dokáže měřit teplotu v rozmezí -40 až +80 stupňů Celsia s přesností $\pm 0,5$ °C a vlhkost v plném rozsahu 0-100% s přesností ± 2 %. [82]



Obrázek 5.6: Schéma zapojení senzoru DHT11/DHT22 k vývojovému kitu NodeMCU [17]

Nastavení přenosu dat senzoru DHT11/DHT22

Po připojení senzoru nastavíme v uživatelském rozhraní TAMOSTy u pinu, ke kterému jsme připojili datový výstup senzoru z nabídky položku DHT11. Toto je právě velká výhoda firmwaru TASMOTA, neboť velké množství populárních modulů a používaných sběrnic má již plně integrované. Po restartování celého firmwaru by se na úvodní stránce grafického rozhraní měly již zobrazovat aktuální hodnoty dat měřených senzorem.

Pro nastavení komunikace mezi vývojovým kitem a řídicí jednotkou s Home Assistantem musíme zjistit MQTT téma v sekci Console ve firmwaru TASMOTA. MQTT témata jsou vytvářena samovolně, ale je možné je upravit a přejmenovat právě v této sekci Console s využitím příkazů dobře popsanych v dokumentaci programu. [83] Následně vložíme do souboru configuration.yaml v Home Assistantu následující kód pro nastavení komunikace mezi vývojovým kitem a řídicí jednotkou:

```

sensor :
- platform: mqtt
  state_topic: "tele/nodemcu1/SENSOR"
  name: "Temperature"
  unit_of_measurement : "C"
  value_template: "{{ value_json.DHT11.Temperature }}"
  retain: true
- platform: mqtt
  state_topic: "tele/nodemcu1/SENSOR"
  name: "Humidity"
  unit_of_measurement: "%"
  value_template: "{{ value_json.DHT11.Humidity }}"
  retain: true

```

U senzoru DHT je specifické to, že posílá v jedné zprávě hodnoty teploty i vlhkosti. Proto je nutné obsah zprávy ve formátu JSON rozklíčovat a vybrat potřebné proměnné, aby byl řídicí systém schopen zaslaná data přečíst. Pokud jsme vše nastavili správně, po restartování celého řídicího systému se zobrazí nadefinované entity včetně hodnot v administraci Home Assistanta ve vývojové sekci Stavů. Nyní je možné s entitami pracovat a tvořit tak grafická rozhraní nebo automatizační procesy.

Pozn. autora:

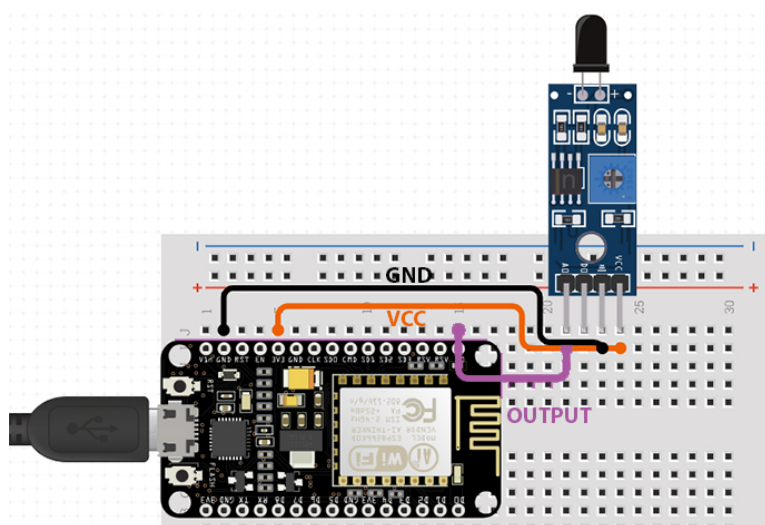
Aktuální hodnoty posílaných dat se začnou po restartování Home Assistanta zobrazovat s menší prodlevou, protože je nutné počkat na konfiguraci MQTT brokera a přihlášení odběratele zpráv k definovaným tématům. S tímto je nutno počítat.

5.2.2 Senzor dýmu

Mezi další senzory, které jsou vhodné pro aplikaci ve vnitřním prostředí budov, jsou detektory kouře.

Nastavení přenosu dat detektoru dýmu

Po připojení nastavíme v TASMOTě u datového GPIO pinu připojeného senzoru z nabídky položku Switch, jelikož se jedná o jednoduchý binární sensor. Po zjištění MQTT tématu odesílaných zpráv vložíme do souboru configuration.yaml následující kód:

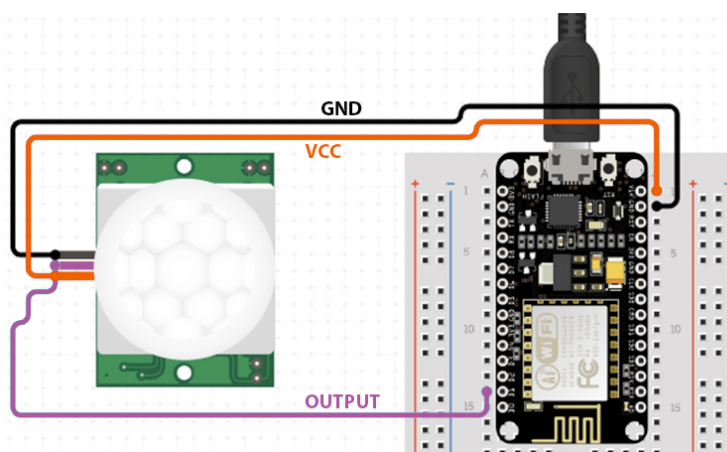


Obrázek 5.7: Schéma zapojení detektoru kouře k vývojovému kitu NodeMCU [17]

```
binary_sensor :
- platform: mqtt
  state_topic: "cmnd/nodemcuflame/POWER"
  name: "Fire Sensor"
```

5.2.3 PIR senzor

Pro řízení osvětlení se velice často používají pohybová čidla na bázi PIR (Passive InfraRed) senzoru. Čidlo se aktivuje při detekci pohybujícího se objektu s teplotou lidského těla. [84]



Obrázek 5.8: Schéma zapojení PIR čidla k vývojovému kitu NodeMCU [17]

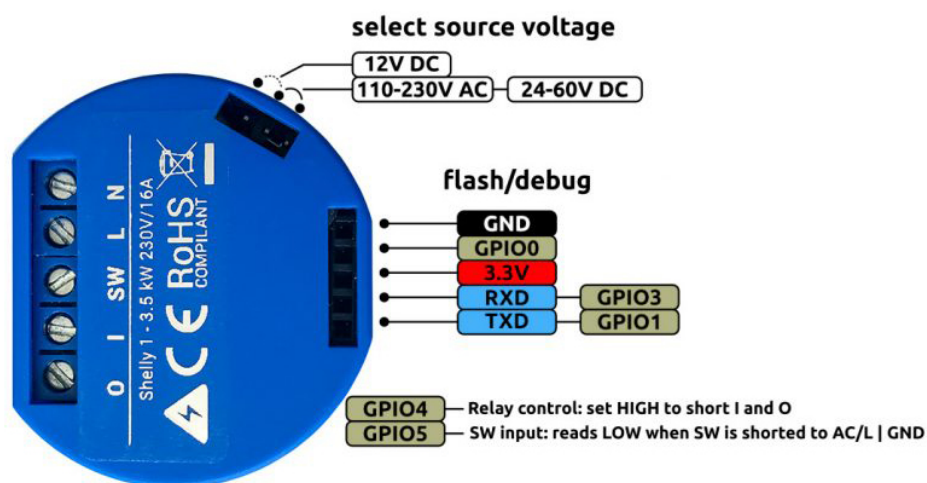
■ Nastavení přenosu dat PIR čidla

Po připojení a nastavení GPIO pinu senzoru opět s použitím položky Switch u datového pinu z PIR senzoru a po zjištění MQTT tématu přeposílaných zpráv vložíme do souboru configuration.yaml následující kód:

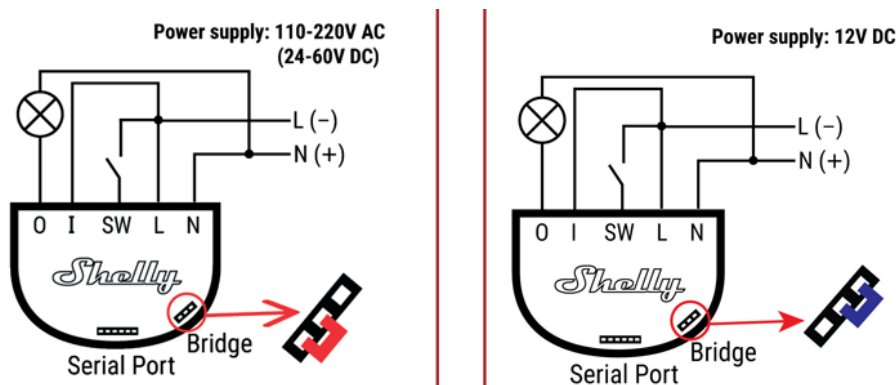
```
binary_sensor:
  - platform: mqtt
    state_topic: "cmnd/nodemcu2pir/POWER2"
    name: "PIR Sensor"
```

■ 5.2.4 WiFi Switch

Abychom PIR čidlem mohli něco spínat musíme implementovat do řídicího systému akční člen. Abych dodržel cíle práce a využil pro komunikaci opět protokol MQTT, zvolil jsem WiFi spínač. Značek a modelů je na trhu velmi mnoho. Většina z nich využívá výše zmíněný čip ESP8266, který obsahuje i mnou vybraný spínač Shelly 1. [18] Pro nastavení MQTT komunikace je možné využít originální firmware, nebo opět můžeme nahrát firmware TASMOTA. Vzhledem k tomu, že už se nejedná o vývojový kit, zařízení nedisponuje USB to TTL převodníkem. To znamená, že musíme použít externí převodník, abychom byli schopni firmware nahrát. Zjednodušení u tohoto konkrétního modelu spínače je v tom, že nemusíme pájet vodiče přímo k čipu ESP8266, protože spínač je již z výroby osazen GPIO dutinkovou lištou pro jednoduché zapojení vodivých konektorů. Já jsem upřednostnil v praktické ukázce opět použití firmwaru TASMOTA.



Obrázek 5.9: Popis GPIO pinů WiFi spínače Shelly 1 [18]



Obrázek 5.10: Schéma zapojení WiFi spínače Shelly 1 s manuálním spínačem a žárovkou pro DC a AC [19]

■ Nastavení přenosu dat WiFi spínače

Následně po připojení spínače do elektrického obvodu, který je schématicky naznačen na obrázku 5.10, nastavení MQTT komunikace a zjištění MQTT tématu zprávy o stavu spínače vložíme následující kód do souboru configuration.yaml:

```
switch:
  - platform: mqtt
    name: "shelly1_zkouska_muj_pokoj"
    command_topic: "cmd/shellytas1/POWER"
    state_topic: "stat/shellytas1/POWER"
    qos: 1
    payload_on: "ON"
    payload_off: "OFF"
    retain: true
```

■ Nastavení automatizačního procesu s využitím PIR senzoru a WiFi spínače

Automatizační proces se opět píše pomocí jazyka YAML. Je možné jej ukládat do separátního souboru, který je ve výchozím nastavení automaticky vytvořen a pojmenován automations.yaml, na který je odkazováno v hlavním souboru configuration.yaml, a nebo můžeme psát automatizační pravidla přímo do

tohoto hlavního souboru. Případně je možné si vytvořit jakýkoliv vlastní soubor uložený jako yaml soubor a na tento soubor odkazovat opět v hlavním souboru configuration.yaml.

```
automation:
  - alias: 'DI motion ON'
    trigger:
      - platform: state
        entity_id: binary_sensor.pir_sensor
        from: 'off'
        to: 'on'
    condition:
      - condition: time
        after: '18:00:00'
        before: '01:30:00'
      - condition: state
        entity_id: sun.sun
        state: below_horizon
    action:
      - service: switch.turn_on
        data:
          entity_id: switch.shelly_1_jidelna
```

Výše jsem na ukázkou uvedl jednoduchý automatizační proces. Pomocí klíčového slova Trigger je definováno, na základě jakého úkonu se proces spustí a pomocí slova Action je definováno, jaký úkon se vykoná. Ještě je možné přidat nepovinnou část Condition, pomocí které je možné upřesnit dodatečnou podmínku, která musí být splněna, aby se celý proces vůbec spustil.

Pokud bych tedy výše uvedený kód měl interpretovat svými slovy, popsal bych jej následovně. Spínač s názvem "Shelly 1 jidelna" se sepne, pokud PIR senzor detekuje pohyb a zároveň budou splněny podmínky času v rozmezí mezi 18:00 až 1:30 a současně bude již po západu slunce.

Kapitola 6

Vizualizace dat

V předchozí kapitole [5](#) jsem detailně popsal několik příkladů, jak lze zapojit vybrané senzory a nastavit komunikaci tak, aby data byla posílána do řídicího systému Home Assistant.

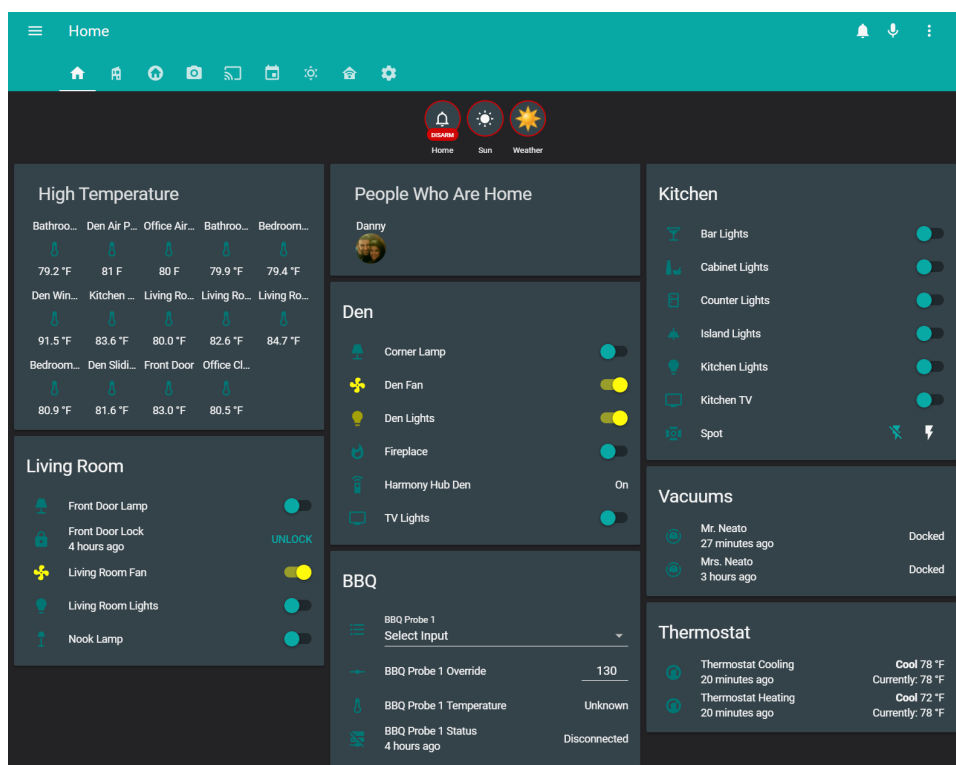
Data jsou tedy již odesílána do řídicího systému, ale stále se nezobrazují na úvodním ovládacím panelu. Za tímto účelem je nutné nastavit vizualizaci dat. Způsobů, jak toho docílit, nabízí Home Assistant velké množství. Já jsem v této kapitole popsal dvě nejpoužívanější řešení, které jsem měl možnost otestovat.

6.1 LovelaceUI

Základní nástroj určený pro vizualizaci dat a tvorbu grafických ovládacích panelů Home Assistant je integrovaný editor LovelaceUI. Ten využívá podobně jako většina inicializačních souborů softwaru Home Assistant (např. `configuration.yaml`) serializační jazyk YAML. Původně byl editor LovelaceUI rozšiřující plug-in, ale od verze 0.86 je již integrován jako výchozí editovatelné uživatelské rozhraní. [85](#)

Rozhraní má předdefinovaných 24 tzv. karet, což jsou bloky pro zobrazení určitých typů entit. Karty jsou přizpůsobitelné. Pro jejich grafické úpravy je možné využít některých stylů CSS. V novějších verzích editoru je již možné

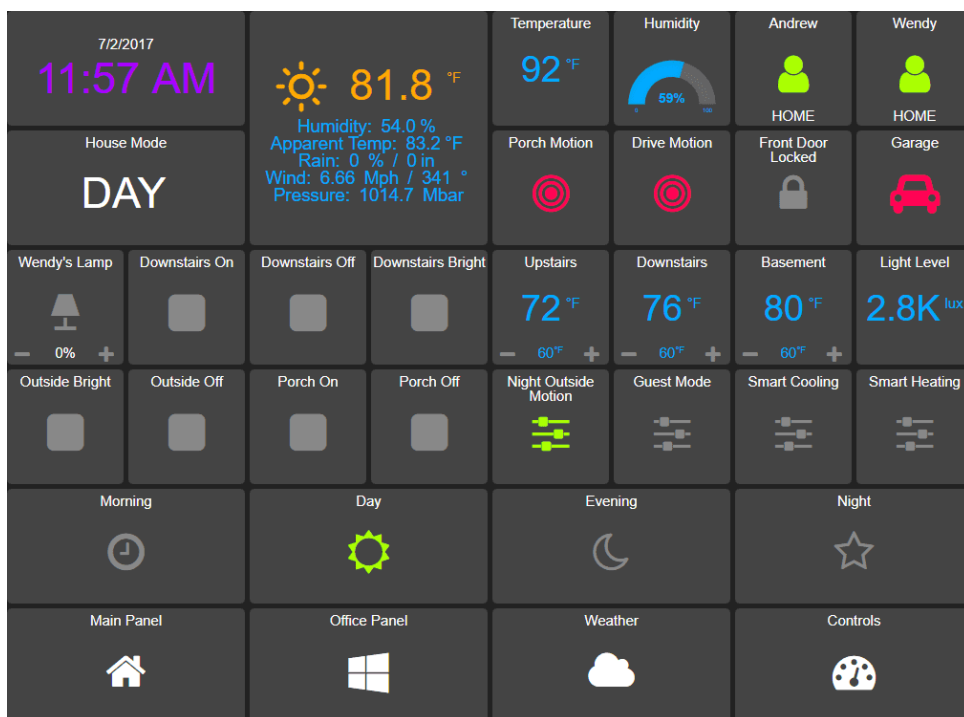
tvořit komplexní přehledy přímo z administrace Home Assistanta bez zásahu do YAML kódu.



Obrázek 6.1: Příklad uživatelského rozhraní pomocí LovelaceUI [20]

6.2 HADashboard

Další oblíbený nástroj určený pro tvorbu uživatelských rozhraní v Home Assistantovi se jmenuje HADashboard. Je vhodnější pro HMI panely a tablety. Pro jeho zprovoznění je nutné nainstalovat rozšíření AppDaemon. Po instalaci můžeme začít tvořit přehledy v nově vytvořené složce `config > appdaemon > dashboard`. Dokumentace včetně popisu konfigurace nových panelů je zpracována v dokumentaci rozšíření AppDaemon. [86] K rozhraní přistupujeme ve výchozím nastavení na portu 5050 IP adresy řídicí jednotky. Pro vzdálený přístup je opět nutné nastavit předání portu v nastavení routeru LAN.



Obrázek 6.2: Příklad uživatelského rozhraní pomocí HADashboard [21]

Kapitola 7

Závěr

Tato práce se zabývá řízením chytré domácnosti. Primárním cílem bylo vybrat jednotlivé komponenty řídicího systému chytrého domu a navrhnout komunikační strukturu s využitím protokolu MQTT. V neposlední řadě jsem měl za úkol celý systém zprovoznit a otestovat, zda navržená struktura funguje.

V úvodu práce jsem krátce nastínil problematiku domácí automatizace a popsal pojmy chytrý dům a internet věcí. Následně jsem detailně popsal protokol MQTT, vysvětlil strukturu základní sítě komunikující s využitím tohoto protokolu a definoval pojmy potřebné pro pochopení datového toku této sítě. V další kapitole jsem formou rešerše představil open-source jednodeskové počítače jako centrální řídicí jednotky a automatizační systémy pro chytrý dům, které jsou kompatibilní s předvybranými počítači. U každého jsem se snažil poukázat na jeho přednosti ve srovnání s konkurencí. Pro samotný návrh komunikační struktury a pro praktickou ukázkou jsem zvolil řídicí jednotku Raspberry Pi 3B+ a software domácí automatizace Home Assistant. V následující části jsem vypracoval vlastní návrh struktury řídicího systému formou schématu včetně detailního popisu.

Poznatky z teoretické části jsem využil pro praktické zprovoznění celého systému v našem domě při dodržení navrženého schématu. Pro otestování datové komunikace jsem použil vybrané senzory z výukové sady a WiFi spínač pro řízení žárovky v závislosti na posílaných datech ze sensorů. V poslední části jsem se zmínil o možnostech vizualizace dat a tvorby grafických rozhraní řídicího systému Home Assistant.

Vzhledem k úspěšnému otestování navržené strukturu řídicího systému lze tento systém považovat za funkční a je možné ho použít nebo se jím inspirovat pro další realizace. Velkou výhodou v tomto navrženém systému je široká kompatibilitnost s různými zařízeními a technologiemi, jako může být Z-wave, ZigBee, Tuya a další. Díky tomu je možné jej využít jako řídicí systém i pro komplexní náročnější aplikace s širokou škálou komunikačních rozhraní. Cíle této práce je tím pádem možné považovat za splněné.

Za sebe mohu říci, že mne několikaměsíční studium celé oblasti automatizace chytré domácnosti a fungování bezdrátové komunikace velice obohatilo a nadchlo natolik, že bych se v této oblasti i nadále rád vzdělával a minimálně pokračoval v rozšiřování řídicího systému v našem domě s využitím i dalších výše zmíněných komunikačních technologií.



Příloha

Obsah přiloženého CD je rozdělen do následujících adresářů:

- Adresář Řešení obsahuje otestovaný operační systém HassOS řídící jednotky s Home Assistantem pro školní Raspberry Pi model B včetně implementace rozšíření popsaných v této bakalářské práci a soubor sonoff.bin obsahující firmware TASMOTA pro nahrání na vývojový kit NodeMCU pro zprovoznění MQTT komunikace mezi tímto vývojovým kitem a řídící jednotkou. Navíc je v tomto adresáři i textový soubor read.txt, kde jsem sepsal základní informace pro práci se soubory z hotového řešení.
- Adresář text obsahuje text této práce se zadáním bakalářské práce a další adresář se zdrojovým kódem textu v LaTeX. Pro vypracování jsem použil šablonu dostupnou z [\[87\]](#).



Literatura

- [1] J. Peterka, “Sága rodů LAN a WAN,” *Softwarové noviny*, Jul. 1997. [Online]. Available: <http://www.earchiv.cz/a708s600/a708s684.php3>
- [2] C. Götz, “Getting started with MQTT - Virtual IoT Meetup presentation,” Jun. 2015. [Online]. Available: <https://www.slideshare.net/goetzchr/getting-started-with-mqtt-virtual-iot-meetup-presentation>
- [3] M. Malý, “Protokol MQTT: komunikační standard pro IoT,” Jun. 2016. [Online]. Available: <https://www.root.cz/clanky/protokol-mqtt-komunikacni-standard-pro-iot/>
- [4] Fabio Nelli, “Raspberry Pi 3 - The evolution goes on - Meccanismo Complesso,” Jul. 2016. [Online]. Available: <https://www.meccanismocomplesso.org/en/raspberry-pi-3-the-evolution-goes-on/>
- [5] “New Page 5.” [Online]. Available: <http://www.mpcug.net/raspberry%20Pi%203.htm>
- [6] M. Hawkins, “Simple Guide to the Raspberry Pi GPIO Header and Pins,” Jun. 2012. [Online]. Available: <https://www.raspberrypi-spy.co.uk/2012/06/simple-guide-to-the-rpi-gpio-header-and-pins/>
- [7] Xunlong Software CO., Limited, “Orange Pi 3 - Orangepi.” [Online]. Available: <http://www.orangepi.org/Orange%20Pi%203/>
- [8] Home Assistant Community, “Home Assistant.” [Online]. Available: <https://www.home-assistant.io/>
- [9] H. A. Community, “Hass.io Architecture · Home Assistant dev docs,” 2019. [Online]. Available: <https://developers.home-assistant.io/index.html>

- [10] K. Kreuzer, “New logo for openHAB,” Oct. 2017. [Online]. Available: <https://community.openhab.org/t/new-logo-for-openhab/35140>
- [11] B. Hobby, “openHAB Basics Tutorial - (Part 2/n) - Bindings & Things,” Feb. 2019. [Online]. Available: <https://community.openhab.org/t/openhab-basics-tutorial-part-2-n-bindings-things/66345>
- [12] SmartThings Inc., “SmartThings.” 2019. [Online]. Available: <https://www.smartthings.com/press-kit>
- [13] Espressif Systems, “Espressif Systems - Wi-Fi and Bluetooth chipsets and solutions,” 2019. [Online]. Available: <https://www.espressif.com/en>
- [14] P. Hobusch, “Hardware Planning.” [Online]. Available: <http://www.liot.io/hardware/planning/>
- [15] Redakce Root.cz, “NodeMCU a jeho verzie: doska s Wi-Fi čipom ESP8266,” Nov. 2017. [Online]. Available: <https://www.root.cz/clanky/nodemcu-a-jeho-verzie-doska-s-wi-fi-cipom-esp8266/>
- [16] “What are the difference between #ESP8266 and #ESP32? #Amica #IoTpic.twitter.com/uUb5zjw9xh,” Mar. 2016. [Online]. Available: https://twitter.com/Amica_IO/status/713271719230709761/photo/1
- [17] ROBOPLAN TECHNOLOGIES LTD, “Circuit Design App for Makers-circuito.io.” [Online]. Available: <https://www.circuito.io/>
- [18] Teracomunications, “Shelly 1 Open Source,” 2018. [Online]. Available: <https://shelly.cloud/shelly1-open-source/>
- [19] G. Dümmler, “Anschlussplan Shelly 1 v2,” Nov. 2018. [Online]. Available: <https://forum.creationx.de/forum/index.php?thread/1225-anschlussplan-shelly-1-v2/>
- [20] H. A. Comunity, “Lovelace UI,” 2019. [Online]. Available: <https://www.home-assistant.io/lovelace/>
- [21] D. Shokouhi, “Home Assistant settings example,” Apr. 2019, original-date: 2018-01-23T23:16:17Z. [Online]. Available: <https://github.com/dshokouhi/Home-AssistantConfig>
- [22] Antonín Vojáček, “IoT MQTT prakticky v automatizaci - 1.díl - úvod | Automatizace.HW.cz,” Jan. 2017. [Online]. Available: <https://automatizace.hw.cz/iot-mqtt-prakticky-v-automatizaci-1dil-uvod.html>
- [23] “MQTT,” Mar. 2019, page Version ID: 887547235. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=MQTT&oldid=887547235>

- [24] Ben Hardill, “protocols - Can two different MQTT brokers communicate with each other?” Jan. 2018. [Online]. Available: <https://stackoverflow.com/questions/48288285/can-two-different-mqtt-brokers-communicate-with-each-other>
- [25] T. H. Team, “MQTT Essentials Part 8: Retained Messages,” Mar. 2015. [Online]. Available: <https://www.hivemq.com/blog/mqtt-essentials-part-8-retained-messages/>
- [26] “MQTT V3.1 Protocol Specification.” [Online]. Available: <http://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html>
- [27] “History of free and open-source software,” Feb. 2019, page Version ID: 884223470. [Online]. Available: https://en.wikipedia.org/w/index.php?title=History_of_free_and_open-source_software&oldid=884223470
- [28] “Overleaf, Online LaTeX Editor.” [Online]. Available: <https://www.overleaf.com>
- [29] Free Software Foundation, Inc., “Licenses.” [Online]. Available: <https://www.gnu.org/licenses/licenses.html>
- [30] I. Free Software Foundation, “Why do you call the system we use gnu/linux and not linux?” [Online]. Available: <https://www.gnu.org/gnu/gnu-linux-faq.en.html#why>
- [31] Free Software Foundation, Inc., “Proprietární software.” [Online]. Available: <https://www.gnu.org/philosophy/categories.html#ProprietarySoftware>
- [32] A. Stevenson, *Oxford dictionary of English*. Oxford University Press, USA, 2010.
- [33] The Centre for Computing History, “Eben Upton - The Story of Raspberry Pi - Computing History,” Jun. 2016. [Online]. Available: <http://www.computinghistory.org.uk/det/41927/Eben-Upton-The-Story-of-Raspberry-Pi/>
- [34] Pi Swag Co Ltd., “Raspberry Pi Zero - Raspberry Pi Swag,” Feb. 2016. [Online]. Available: <https://web.archive.org/web/20160208235627/https://swag.raspberrypi.org/collections/pi-zero/products/pi-zero>
- [35] P. Torrone, “23 million + Raspberry Pi computers sold @Raspberry_pi #raspberrypi,” Dec. 2018. [Online]. Available: https://blog.adafruit.com/2018/12/21/23-million-raspberry-pi-computers-sold-raspberry_pi-raspberrypi/
- [36] “Raspbian,” Dec. 2018, page Version ID: 875141376. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=Raspbian&oldid=875141376>

- [37] S. Long, “Raspbian Stretch has arrived for Raspberry Pi,” Aug. 2017. [Online]. Available: <https://www.raspberrypi.org/blog/raspbian-stretch/>
- [38] the Raspberry Pi Foundation, “Raspberry Pi Downloads - Software for the Raspberry Pi.” [Online]. Available: <https://www.raspberrypi.org/downloads/>
- [39] C. Atwell, “10 Best Single-Board Computers for 2018,” Nov. 2018. [Online]. Available: https://www.eetimes.com/document.asp?doc_id=1333687
- [40] R. Pelayo, “Raspberry Pi GPIO Tutorial,” Jun. 2018. [Online]. Available: <https://www.teachmicro.com/raspberry-pi-gpio-tutorial/>
- [41] “Banana Pi,” Feb. 2019, page Version ID: 16954827. [Online]. Available: https://cs.wikipedia.org/w/index.php?title=Banana_Pi&oldid=16954827
- [42] Sinovoip, “About BPI - Banana Pi Wiki.” [Online]. Available: http://wiki.banana-pi.org/About_BPI
- [43] Sinovoip, “Banana Pi BPI-R64.” [Online]. Available: <http://www.banana-pi.org/r64.html>
- [44] Sinovoip, “BPI-M2 Berry Quad-core Development Board.” [Online]. Available: <http://www.banana-pi.org/m2ub.html>
- [45] Xunlong Software CO., Limited, “Orange Pi.” [Online]. Available: <http://www.orangepi.org/Docs/mainpage.html>
- [46] G. Pollack, “Interview with Paulus Schoutsen, founder of Home Assistant,” Mar. 2017. [Online]. Available: <http://codepop.com/open-sourcecraft/episodes/paulus-schoutsen/>
- [47] P. Schoutsen, “Home Assistant,” Sep. 2013. [Online]. Available: <https://github.com/home-assistant>
- [48] “Installation of Home Assistant.” [Online]. Available: <https://www.home-assistant.io/docs/installation/>
- [49] Home Assistant Community, “Installing Hassbian,” 2019. [Online]. Available: <https://www.home-assistant.io/docs/installation/hassbian/installation/>
- [50] H. Assistant, “Installing Hass.io,” 2019. [Online]. Available: <https://www.home-assistant.io/hassio/installation/>
- [51] B. Association, “Buildroot - Making Embedded Linux Easy.” [Online]. Available: <https://buildroot.org/>
- [52] D. Inc., “Enterprise Application Container Platform | Docker,” 2019. [Online]. Available: <https://www.docker.com/>

- [53] M. Malý, “YAML: Serializační formát pro ukládání dat,” Dec. 2009. [Online]. Available: <https://www.zdrojak.cz/clanky/yaml-serializacni-format-pro-ukladani-dat/>
- [54] Home Assistant Community, “Lovelace UI,” 2019. [Online]. Available: <https://www.home-assistant.io/lovelace/>
- [55] H. A. Community, “HADashboard,” 2018. [Online]. Available: <https://www.home-assistant.io/docs/ecosystem/hadashboard/>
- [56] Home Assistant Community, “Components.” [Online]. Available: <https://www.home-assistant.io/components/>
- [57] Free Software Foundation, Inc., “gnu.org,” Dec. 2017. [Online]. Available: <https://www.gnu.org/software/bash/>
- [58] H. Assistant, “Hass.io Add-ons,” 2018. [Online]. Available: <https://www.home-assistant.io/addons/>
- [59] K. Kreuzer, “About Me - Kai Kreuzer,” 2018. [Online]. Available: <http://www.kaikreuzer.de/about/>
- [60] OpenHAB Community and the openHAB Foundation e.V., “Introduction,” 2019. [Online]. Available: <https://www.openhab.org/docs/>
- [61] O. Community and the openHAB Foundation e.V., “openHABian,” 2019. [Online]. Available: <https://www.openhab.org/docs/installation/openhabian.html>
- [62] OpenHAB Community and the openHAB Foundation e.V., “Concepts,” 2019. [Online]. Available: <https://www.openhab.org/docs/concepts/>
- [63] O. Community and the openHAB Foundation e.V., “HABPanel,” 2019. [Online]. Available: <https://www.openhab.org/docs/configuration/habpanel.html>
- [64] Samsung, “Historie | Společnost | Samsung Česká republika,” 2019. [Online]. Available: <https://www.samsung.com/cz/aboutsamsung/company/history/>
- [65] A. Tilley, “Samsung Acquires SmartThings, A Fast-Growing Home Automation Startup,” Jul. 2014. [Online]. Available: <https://www.forbes.com/sites/aarontilley/2014/08/14/samsung-smartthings-acquisition-2/>
- [66] SmartThings, “SmartThings Classic Developer Documentation — SmartThings Classic Developer Documentation,” 2018. [Online]. Available: <https://docs.smartthings.com/en/latest/>
- [67] “SmartThings.” [Online]. Available: <https://www.smartthings.com/products>

- [68] “Amazon Best Sellers: Best Home Automation Hubs & Controllers.” [Online]. Available: <https://www.amazon.com/Best-Sellers-Home-Improvement-Automation-Hubs-Controllers/zgbs/hi/6478740011>
- [69] “The Best Smart Hub for 2019,” Dec. 2018. [Online]. Available: <https://www.reviews.com/smart-hub/>
- [70] Teracomunications, “Contact,” 2018. [Online]. Available: <https://shelly.cloud/contact/>
- [71] “Duck DNS.” [Online]. Available: <https://www.duckdns.org/>
- [72] Espressif Systems, “Milestones | Espressif Systems,” 2019. [Online]. Available: <https://www.espressif.com/en/company/about-us/milestones>
- [73] “ESP8266,” Apr. 2019, page Version ID: 890535598. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=ESP8266&oldid=890535598>
- [74] S. Santos, “ESP32 vs ESP8266 - Pros and Cons,” Jan. 2019. [Online]. Available: <https://makeradvisor.com/esp32-vs-esp8266/>
- [75] Arduino, “Arduino - Software,” 2019. [Online]. Available: <https://www.arduino.cc/en/main/software>
- [76] Microsoft, “Visual Studio Code - Code Editing. Redefined,” 2019. [Online]. Available: <https://code.visualstudio.com/>
- [77] T. Arends, “TASMOTA,” Apr. 2019, original-date: 2017-01-28T13:37:08Z. [Online]. Available: <https://github.com/arendst/Sonoff-Tasmota>
- [78] R. Huang, “NodeMCU-flasher,” Apr. 2019, original-date: 2014-12-05T07:33:43Z. [Online]. Available: <https://github.com/nodemcu/nodemcu-flasher>
- [79] Silicon Laboratories, “USB to UART Bridge VCP Drivers - Silicon Labs,” 2019. [Online]. Available: <https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers>
- [80] T. Riemersma, “Termite: a simple RS232 terminal,” May 2018. [Online]. Available: https://www.compuphase.com/software_termite.htm
- [81] LASKARDUINO.CZ, “LASKKIT 37, Sada senzorů 37 v 1.” [Online]. Available: <https://laskarduino.cz/laskkit-diy-a-starter-kity/180000-sada-senzoru-37-v-1-v-plastovem-boxu.html>
- [82] M. Luboš, “Teploměr a vlhkoměr DHT11 a DHT22 | Arduino návody,” Jul. 2016. [Online]. Available: <https://navody.arduino-shop.cz/navody-k-produktum/teplotni-senzor-dht11.html>

- [83] T. Arends, “Provide ESP8266 based itead Sonoff with Web, MQTT and OTA firmware using Arduino IDE or PlatformIO: arendst/Sonoff-Tasmota,” Apr. 2019, original-date: 2017-01-28T13:37:08Z. [Online]. Available: <https://github.com/arendst/Sonoff-Tasmota/wiki/Commands>
- [84] P. Honzík, “Jak funguje bezpečnostní pohybový detektor | Jak to funguje... | zabezpečovací-zařízení.cz,” May 2015. [Online]. Available: <https://www.zabezpecovaci-zarizeni.cz/pohybove-detektory/vnitri-pohybove-detektory/jak-funguje-pohybovy-detektor-%5Bb058%5D>
- [85] P. Schoutsen, “Lovelace UI released!” Jan. 2019. [Online]. Available: <https://www.home-assistant.io/blog/2019/01/23/lovelace-released/>
- [86] H. A. Community, “AppDaemon,” 2018. [Online]. Available: <https://www.home-assistant.io/docs/ecosystem/appdaemon/>
- [87] P. Olšák, “CTUstyle - Šablona pro sazbu závěrečných prací na ČVUT,” May 2019. [Online]. Available: <http://petr.olsak.net/ctustyle.html>