

Obsah

1. Úvod	1
2. Automatizace domácnosti a technologie dálkového přenosu	2
2.1 Automatizace domácnosti	2
2.2 Technologie bezdrátového přenosu	3
2.3 Technologie bezdrátového přenosu WiFi	6
2.4 Technologie bezdrátového přenosu Bluetooth	8
2.5 Technologie bezdrátového přenosu ZigBee	8
3. Vestavné systémy a vývojové prostředky	9
2.1 Jednodeskový počítač Raspberry Pi	11
2.2 Moduly ESP8266 a ESP32	11
2.3 Programovací jazyk Python	12
2.4 Python knihovny pro vývoj GUI	12
4. Zhodnocení současného stavu a plán práce	14
5. Realizace a testování	18
2.1 Volba prostředků	18
2.2 Aplikace pro Raspberry Pi	20
4.1 Volba a konfigurace veřejného serveru	28
4.2 Aplikace pro koncové moduly	30
6. Závěr	31
7. Seznam použité literatury	32

1. Úvod

Zde bude cca jedna strana úvod...

2. Automatizace domácnosti a technologie dálkového přenosu

Následující část má bezprostřední vztah k bakalářské práci. Neobsahuje tedy všechny informace o daných tématech, ale pouze ty, které jsou potřebné. Nejprve je zde popsána automatizace domácnosti, co to znamená a jaké jsou dnes možnosti jejího využití. Následuje část zabývající se technologiemi bezdrátového přenosu informací. Nejprve je pojednáno o tom, co tento typ komunikace obnáší, jaké má výhody a nevýhody. Dále je uveden popis některých principů tří často používaných standardů bezdrátové komunikace (WiFi, Bluetooth a ZigBee).

2.1 Automatizace domácnosti

Automatizace domácnosti spočívá v automatizování činností, které řídí domácnost, normálně vykonávané člověkem. V souvislosti s tím někdy hovoříme o inteligentní domácnosti. Může se jednat v nejjednodušším případě o prostý časovač, vypínající světlo [S, str.1], nebo i o komplexnější řešení, jako je ovládání domácnosti na dálku (ideálně mobilním telefonem, nebo tabletem), inteligentní řízení spotřebičů domácnosti (jako světla, větrák, žaluzie...) na základě nejrůznějších senzorů a podobně.

Automatizace v mnohém usnadňuje život a umožňuje provádění akcí, které by jinak byly prakticky nemožné (například zabezpečení domu). Jedním z cílů inteligentních domů je úspora energií, díky využití automatizace. Mezi typické aplikace automatizace domácnosti patří:

- Zmíněné zabezpečení
- Systém pro inteligentní vytápění a ventilaci (HVAC)
- Ovládání spotřebičů [T]
- Samo zavlažovací systémy [U]

Ovladače a časovače

Nejjednodušším prvkem automatizace domácnosti mohou být různé časovače. Na českém trhu jsou k dispozici především zásuvky a relé (na DIN lištu) s časovým spínačem. Jejich cena se pohybuje okolo 100–300 Kč.

Další volbou jsou pak různé WiFi spínače, jako například [V] či spínače/relé, ovládané RF ovládači pracující na některé z nelicencovaných frekvencích, například [G]. Cena je podobná.

Inteligentní ovládací prvky

Pro ještě komfortnější ovládání a automatizaci domácnosti je možné využít některé inteligentní systémy, jako jsou Amazon Alexa, nebo Google Assistant.

//Loxone?

//Home assistant a open hub

2.2 Technologie bezdrátového přenosu

Pro přenos dat či řídicích signálů je vždy potřeba zvolit vhodné médium, přes které se budou tyto informace přenášet. V některých situacích není pro přenos vhodné (a někdy dokonce ani možné) používat kabely (ať už metalické nebo optické). V těchto případech je potřeba přenášet informace bezdrátově, tj. za využití jiných médií, jako je vzduch.

Podobně jako je nutné u kabelového spojení využít vhodný způsob komunikace (například zvolit vhodnou sběrnici a nastavit ji správné parametry) je potřeba se způsobem komunikace zabývat rovněž u bezdrátového přenosu. Zde je nutné zejména zvolit vhodnou technologii (jako je Wifi, Bluetooth či ZigBee) a její parametry [A].

Výhody bezdrátového přenosu

Bezdrátová komunikace má oproti kabelové řadu výhod. Zejména se jedná o následující:

- Jednodušší připojení – zařízení není potřeba připojovat kabelem, a dokonce nemusí být ani vybaveno konektorem pro toto připojení (pozn. pro dálkový přenos prostřednictvím světla je však stále potřeba mít nějaký přijímací port). Z toho rovněž plyne, že není potřeba měnit strukturu sítě kvůli změnám v místnosti a rovněž není potřeba myslet na konkrétní strukturu sítě ještě před budováním.
- Větší spolehlivost – Častým zdrojem problémů s kabelovým připojením jsou chyby na straně kabelů – jejich poškození. Použitím bezdrátových technologií se lze vyhnout tomuto typu chyb.
- Snadná rozšiřitelnost sítě – U kabelového připojení je potřeba řešit způsob rozšíření sítě a v případě, že stávající struktura sítě rozšíření nepodporuje, tak je potřeba ji celou pozměnit. Bezdrátové sítě tento problém eliminují.
- Nižší cena – Použitím bezdrátových technologií se značně sníží pořizovací cena sítě – není potřeba kupovat drahou kabeláž. Rovněž instalace kabelů do starých budov může být velmi nákladná a problémová.

Nevýhody bezdrátového přenosu

Kromě množství výhod, které bezdrátová komunikace představuje jsou zde rovněž některé

nevýhody tohoto typu komunikace:

- Rušení signálu – zařízení, využívající bezdrátové technologie může způsobovat rušení ostatních zařízení a rovněž opačně – dané zařízení může být rušeno od ostatních zařízení, pracujících na podobném principu
- Bezpečnost – bezdrátová komunikace často vysílá (a přijímá) signály do relativně rozsáhlého otevřeného prostoru, tudíž jsou takto vysílaná data často daleko méně chráněná než u kabelového přenosu (kde je k získávání dat potřeba mít fyzické připojení k síti, ve které se data přenáší) [A] [Q, str. 5-6] [R, str. 406]. Je tedy nutné zabezpečit přenos dat.

Modulace

Signál, který je potřeba přenášet v bezdrátových sítích reprezentuje přenášená data. Ty jsou však třeba nějakým způsobem upravit, protože z dat vytvořený signál (obvykle reprezentovaný dvěma úrovněmi napětí na vodiči) není pro bezdrátový přenos příliš vhodný. Tato modifikace signálu se nazývá modulace. Jedná se o proces, při kterém signál v základním pásmu (např. video) moduluje tzv. nosný signál, který data přenáší a informace je zakódovaná právě tím, jak je původním signálem modulovaný.

Modulace přináší do bezdrátového přenosu mnoho výhod:

- Menší velikost antén
- Větší dosah
- Kvalitnější přenos informací [AI, str.1-2]

Způsob komunikace

V případě bezdrátových technologií se využívá některého pásma elektromagnetického vlnění. Rychlost šíření tohoto záření je ve vakuu rovno konstantě c (přibližně 3×10^8), v médiu jako je vzduch se pak šíří rychlostí c , podělenou indexem lomu (konkrétně pro vzduch je tento index blízký 1, takže můžeme uvažovat prakticky stejnou rychlost jako pro vakuum) [M] [N, str 2-3] [O, str.24] [P, kap 8.2].

V současnosti se na trhu s elektronikou zařízení, využívající především dva různé principy dálkového přenosu informací, první je založen na využití světla, druhý pak využívá rádiové vlny. [A]

Přenos informací pomocí světelného signálu

V případě světelného signálu se většinou využívá infračerveného záření, jelikož není lidským

okem viditelné, avšak je možné vyrobit přijímač, která tento signál detekuje (a to je vlastnost, která se zde vyžaduje).

Kromě těchto definovaných protokolů je pro komunikaci pomocí IR záření možno použít standardů IEEE 802.11 (Tyto standardy to tak definují). V praxi se však nikdy nic takového nedočkal rozšíření.

První princip, který je možný použít je využití infračerveného záření. Jelikož není obvykle v domácnosti mnoho zařízení, pracujících s IR, nebývají většinou zařízení navzájem příliš rušeny. Stále zde však existuje rušení od jiných zdrojů infračerveného záření, například ze slunečního záření, nebo fluorescenčního světla. Rušení od těchto zdrojů je však možné potlačit jistými principy. Prvním je vyhrazení určité vlnové délky, která se bude pro přenos informací používat a následným použitím filtru na přijímací diodě, který odfiltruje ostatní vlnové délky. Nepotlačené rušení (od zdrojů, které vyzařují v oné vyhrazené vlnové délce (problémem je tedy zejména Slunce) je možné dále potlačit tím, že bude přijímač reagovat pouze na nějakou modulovanou frekvenci, nepřítomnou v daném zdroji (tedy například ve slunečním záření). Systémy využívající IR záření se vyznačují tím, že je musejí splňovat podmínku přímé viditelnosti vysílače a přijímače. Není tedy možné (bez případných dodatečných, opakovacích zařízení) ovládat zařízení za rohem, pokud není přímo viditelné. Právě díky této vlastnosti je možné volně využívat zařízení, využívající tohoto principu, protože nedochází k žádnému rušení a není tak potřeba regulovat směrnici používání IR vysílání. Dosah IR vysílačů se obvykle udává v jednotkách, případně desítkách metrů.

Přenos informací pomocí rádiových vln

Kromě IR záření mohou zařízení k dálkovému přenosu informací využívat také rádiových vln na různých frekvencích. Zde však již existují jistá omezení. Rádiové vlny se totiž (na rozdíl od IR světla) šíří i skrze předměty. To je příčinou toho, že se mohou i relativně vzdálená zařízení komunikující na stejných vlnách vzájemně rušit. Aby se předešlo naprostému zarušení prostoru, je potřeba mít k vysílání na určitých frekvencích licenci. Je zřejmé, že si běžní uživatelé zařízení v domácnosti nemohou dovolit kupovat drahé licence kvůli každému bezdrátově ovládanému zařízení, které si koupí. Z tohoto důvodu bylo navrženo tzv. pásmo ISM.

V pásmu ISM jsou definovány frekvenční rozsahy, které je možné volně použít pro schválená

zařízení bez licence. To ovšem také znamená, že zařízení pracující v těchto rozsazích musejí tolerovat rušení od ostatních zařízení pracujících na stejných frekvencích. [C, s.66].

Dokument „ITU Radio Regulations“ toto pásmo vyhrazuje pro „Provoz vybavení nebo zařízení určených ke generování a využívání lokální vysokofrekvenční energie pro průmyslové, vědecké, lékařské, domácí nebo podobné účely, s výjimkou aplikací v oblasti telekomunikací“.

Nejčastěji se pro komunikaci v pásmu ISM používá frekvenční pásmo 2,4 GHz. To je dané historickým vývojem. Zejména u mikrovlnných trub bylo potřeba zvolit vhodné pásmo [X]. Zvolené pásmo 2,4 GHz bylo vybráno z několika důvodů, zejména však na základě empirického měření průniku a šíření tepla pro různé potraviny (při použití frekvencí tohoto pásma) a s ohledem na rozměry použitého magnetronu (součástky, která generuje mikrovlnné záření [W, kap. 6-20]).

IOT protokoly:

<https://www.postscapes.com/internet-of-things-protocols/>

<https://www.ubuntupit.com/top-15-standard-iot-protocols-that-you-must-know-about/>

sniffing:

<https://www.youtube.com/watch?v=pfG8uEDZj5g>

MQTT:

<https://www.youtube.com/watch?v=Elxdz-2rhLs>

https://www.youtube.com/watch?v=iNWsW_q4Fu0

PŘEDNÁŠKA!!!!:

<https://www.youtube.com/watch?v=s6ZtfLmvQMU>

2.3 Technologie bezdrátového přenosu WiFi

Wifi je technologie, využívající standardů z rodiny IEEE 802.11. První verze tohoto standardu byla organizací IEEE schválena v roce 1977 [A, s.6]. Od té doby vyšlo mnoho dalších verzí standardů. Jednotlivé verze se od sebe mohou odlišovat různými parametry, například frekvenčním pásmem, šířkou pásma jednotlivých kanálů, maximální rychlostí přenosu atd. Organizace Wi-Fi Alliance rozlišuje některé standardy IEEE 802.11 číslem generace WiFi, nejnovější je prozatím zatím 6. generace (založená na standardu 802.11ax).

IEEE 802.11 PHY Standards							
Release date	Standard	Frequency Band	Bandwidth	Transmission Scheme	Max Modulation	MIMO	Max Data Rate
1997	802.11	2.4 GHz	20 MHz	DSSS, FHSS	QPSK	N/A	2 Mb/s
1999	802.11b	2.4 GHz	20 MHz	DSSS	QPSK	N/A	11 Mb/s
1999	802.11a	5 GHz	20 MHz	OFDM	64QAM	N/A	54 Mb/s
2003	802.11g	2.4 GHz	20 MHz	DSSS, OFDM	64QAM	N/A	54 Mb/s
2009	802.11n	2.4 GHz 5 GHz	20 MHz 40 MHz	OFDM	64QAM	4x4	600 Mb/s
2013	802.11ac	5 GHz	20 MHz 40 MHz 80 MHz 160 MHz	OFDM	256QAM	8x8	6.93 Gb/s
2018	802.11ad	60 GHz	2160 MHz	SC, OFDM	256QAM	Beamforming	6.93 Gb/s



Tabulka 1 – některé důležité verze standardu IEEE 802.11 a jejich parametry¹

Wifi funguje na principu vysílání a přijímání rádiových vln. Organizace IEEE rozhodla využít pro technologii Wi-Fi frekvence z pásma ISM [B, str.2]. Wifi standardně využívá frekvencí 2,4Ghz a 5Ghz. Nejprve byla zařízení Wi-Fi schopná pracovat pouze v jednom z těchto dvou frekvenčních pásem, ale 4. generace (IEEE 802.11n) přidává možnost práce v obou zmíněných pásmech. Moderní zařízení s wifi si tak mohou vybrat (a dokonce během své činnosti měnit) frekvenci, na které budou spolu komunikovat.

Obě pásma mají svá pro i proti. Mezi výhody pásma 2.4Ghz patří zejména větší pokrytí signálu a rovněž větší kompatibilita (platí spíše pro starší zařízení). Na druhou stranu pásmo 5Ghz nabízí podstatně vyšší přenosové rychlosti a dále větší množství komunikačních kanálů [D].

Režim sítě

Wifi nachází uplatnění v (bezdrátových) lokálních sítích. V nich pak rozlišujeme 3 režimy na základě toho, jak se Wifi zařízení v síti mezi sebou navzájem spojují (jakou plní roli):

- Režim infrastruktury
- Ad hoc režim
- Smíšený režim

V režimu infrastruktury je v síti přítomen minimálně jeden centrální prvek (tzv. přístupový

¹ <https://www.grandmetric.com/2018/05/29/wi-fi-standards-evolution/>

bod), který zprostředkovává komunikaci mezi jednotlivými prvky (klienty) sítě, případně poskytuje připojení do jiné sítě přes distribuční systém (DS). V tomto režimu sítě je výhoda, že je snadné připojit do stávající infrastruktury nový prvek.

Ad hoc je režim bezdrátové sítě, ve které není přítomen žádný centrální prvek (přístupový bod) se kterým by prvky sítě komunikovali, ani zde není žádné spojení se pevnou sítí přes distribuční systém. Jedná se tedy o decentralizovanou síť. Jednotlivé prvky tedy mezi sebou navzájem komunikují přímo (toto spojení se někdy označuje jako tzv. peer-to-peer). V tomto režimu má síť rovněž SSID identifikátor, kterým je možné síť identifikovat. [A][B]

Bezpečnost wifi sítě

2.4 Technologie bezdrátového přenosu Bluetooth

Bluetooth je standard, definovaný v IEEE 802.15.1. Vytvořila jej firma Ericsson v roce 1994 a od té doby vyšlo několik nových verzí [A]. Podobně jako WiFi pracuje v ISM pásmu 2,4 GHz. Na rozdíl od Wi-Fi však není definován pouze na prvních dvou vrstvách ISO/OSI, ale definuje protokoly na všech sedmi vrstvách tohoto modelu. Na nejnižší úrovni, kde definuje způsob přenosu jednotlivých bitů využívá metodu FHSS, která zajišťuje, že při přenosu bitů vysílač přeskakuje mezi několika frekvencemi [AD].

Zařízení, které jej využívají, umožňuje vytvořit tzv. PAN (osobní síť). V těchto sítích má každé zařízení přiřazeno unikátní 48bitovou adresu BD_ADDR (Bluetooth Device Address) – jedná se o obdobu MAC adresy u ethernetu. Tu používá pro komunikaci s ostatními zařízeními. Jedno zařízení může být v roli master (řídící), slave (podřízená) nebo obojího [AB, str.4]. K jedné řídící stanici se připojuje jedno a více podřízených zařízení (používá se pouze adhoc komunikace mezi master a slave stanicí). Zde hovoříme o tzv. piconetu (pikosíti). Maximální počet zařízení v jedné pikosíti je 8 (jedna řídící stanice a až 7 podřízených). Stanice náležící do jedné pikosítě může zároveň patřit do jiné pikosítě. Jedná se tedy o rozšíření sítě mezi zařízeními. Takto vytvořenou síť nazýváme tzv. scatternet (rozprostřená síť). V každé rozprostřené síti má každá pikosíť unikátní identifikátor – je jím BD_ADDR její řídící stanice. Díky rozlišení jednotlivých pikosítí pak může každá tato síť využívat jiné skokové sekvence (frekvenčních kanálů na kterých se vysílají/přijímají data) [AC, str. 20].

2.5 Technologie bezdrátového přenosu ZigBee

Zigbee je bezdrátová technologie, založená na standardu IEEE 802.15.4. Je určena pro

vytváření sítě PAN (osobní síť) a pracuje v pásmu ISM 868 MHz, 902-928 MHz a 2,4 GHz [A].

Zařízení v ZigBee síti

ZigBee standard specifikuje 2 typy zařízení – FFL (Full Function Device) a RFD (Reduced Function Device). FFL zařízení je obvykle schopné mnoha funkcí a je stále aktivní, zatímco RFD se nachází většinu času v režimu spánku, ze kterého se občas probudí, například aby odeslalo hodnoty neměřené na nějakém senzoru.

V síti pak každé ze zařízení plní některou ze 3 funkcí:

- Koordinátor
- Koncové zařízení
- Směrovač

Topologie sítě

Na základě definovaných zařízení pak existují 3 možné topologie ZigBee sítě:

- Hvězda
- Strom
- Mesh síť [AG, str.5]

ZigBee Model

ZigBee podobně jako Bluetooth definuje komunikaci na všech úrovních modelu ISO/OSI, nekopíruje však přesně jednotlivé vrstvy. První 3 vrstvy modelů ISO/OSI a ZigBee si odpovídají, ale vrstvy L4-L7 jsou spojené do vrstev APS (Application Support) a ZDO (ZigBee Device Object). [AH, str. 42]

Thread, WeMo, ZigBee and Z-Wave (<https://www.tomsguide.com/us/smart-home-wireless-network-primer,news-21085.html>)

3. Vestavné systémy a vývojové prostředky

Vestavný systém můžeme definovat jako software spolu s počítačem, zabudovaným do nějakého zařízení takovým způsobem, že jej uživatel nevidí jako počítač [E, str.3]. Tento počítač je většinou jednoúčelový, určený pro předem navržené použití. Tím se liší od univerzálních počítačů, které mohou poskytovat různé funkce a jejichž uplatnění se může

měnit (například osobní počítač) [F, str. 3].

Historie

Po objevení polovodičů a zejména po vynálezu tranzistoru a integrovaného obvodu se začali vedle mechanických systémů objevovat systémy elektronické. Ty přinesli mnohá vylepšení. Oproti mechanickým systémům byli znatelně jednodušší na návrh, lehčí, menší a rovněž bylo odstraněno mechanické opotřebení, které bylo u mechanických systémů problémem. I přes výhody, které číslicové systémy přinášeli zde však byli některé vážné nedostatky. Například podobně jako u mechanických systémů, i zde je často nutné při přidání nějaké funkcionality navrhnout znovu celý systém (zejména celý návrh desky plošných spojů).

Velké změny v návrhu systémů způsobil rozvoj integrovaných obvodů, zejména mikroprocesorů. Dosud bylo totiž potřeba pro každou aplikaci vytvořit vlastní systém „na míru“. S programovatelnými paměťmi, které bylo možné připojit k mikroprocesorům, případně které byli součástí mikrokontrolerů však tento problém odpadl a samotný návrh systému se přesunul do jeho popisu. Systém totiž vzniká jeho programováním, což je svým způsobem samotný popis systému. Zde jde konečně mluvit o vestavěném systému, protože zde máme hardware (počítač) i software, které společně vytváří nějaký celek, který se navenek uživateli jeví jako jednoúčelové zařízení. Vestavěné systémy mají oproti číslicovým a mechanickým systémům mnoho výhod. Kromě již zmíněného usnadnění návrhu jsou tyto systémy rovněž snadno rozšiřitelné – stačí do systému pouze nahrát nový kód a není potřeba vytvářet úplně nový návrh systému, jako tomu bylo dříve. Hromadná výroba a rozsah použití těchto systémů také způsobil značný pokles ceny oproti dříve zmíněným systémům.

Základním prvkem vestavěných systémů je jedno, či více číslicové zařízení, které se chová jako mozek celého systému a řídí jeho činnost. Typicky se jedná o mikroprocesor, mikrokontroler nebo digitální signálový procesor [H, str. 1].

Architektura řídicího zařízení

Při návrhu systému se využívá jedna ze dvou architektur:

- Von Neumannova architektura
- Harvardská architektura

Hlavním rozdílem je způsob práce s pamětí. Ve Von Neumannově architektuře je paměť pro program i data spojená do jedné paměti, v Harvardské architektuře je pak rozdělena.

Mikroprocesor (CPU) je programovatelné elektronické výpočetní zařízení, určené pro všestranné použití. Jedná se o čip, obsahující 3 základní součásti:

- Aritmeticko-logickou jednotku
- Řídící jednotku
- Registry [I, str. 18–19]

Mikroprocesor sám o sobě je z hlediska vestavěných systémů relativně jednoduché zařízení, které pro funkci systému potřebuje připojit některé další součásti, jako jsou paměti (RAM a ROM), čítače, časovač a podobně. Návrhář tedy musí tyto součásti přidat externě, aby zařízení fungovalo správně. Systémy, zahrnující mikroprocesory jsou obvykle založeny na Von Neumannově architektuře [J].

Mikrokontrolér je zařízení, které na rozdíl od mikroprocesoru má již všechny součásti, potřebné pro svoji činnost v sobě. Obvykle využívá Harvardské architektury [J].

3.1 Jednodeskový počítač Raspberry Pi

Zařízení Raspberry Pi je levný univerzální počítač malých rozměrů. Poskytuje široké možnosti v oblasti multimédií a 3D grafiky, předpokládá se, že bude časem využíván i jako herní platforma.

Název Raspberry Pi vytvořila komise dozorčí rady. Slovo Raspberry je vzato jako název ovoce (malina), jak už je u počítačových systémů zvykem nazývat podle ovoce. Slovo “Pi” označuje zkráceně “Python” - programovací jazyk, který měl být původně jediným programovacím jazykem dostupným na platformě Raspberry Pi [AA].

Historie

Raspberry Pi vzniklo v r. 2006 za přispění studijního ředitele pro informatiku na Cambridgeské univerzitě za účelem lokálních potřeb. Měl to být nástroj, který by poskytl prvotní impuls studentů k nějakému z univerzitních kurzů.

3.2 Moduly ESP8266 a ESP32

Jedná se o levný mikročip, disponující Wi-Fi stackem, schopný provozu RTOS (realtime operačního systému). Je založen na 32bitovém procesoru s architekturou RISC [AE].

ESP32 je nástupce ESP8266. Kromě komunikace přes Wi-Fi umožňuje rovněž komunikaci pomocí Bluetooth, díky hybridnímu Wi-Fi/Bluetooth čipu [AF].

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

Tabulka 1 – srovnání specifikace modulů ESP8266 a ESP32 ²

3.3 Programovací jazyk Python

Python je objektově orientovaný skriptovací jazyk, s velkou oblastí využití. Mezi jeho přednosti patří jednoduchost, přehlednost, objektově orientovaný přístup. Jelikož je Python open-source, je možné jeho interpret stáhnout a používat zdarma [Z, str 3]. Často se využívá k programování GUI aplikací, aplikací využívající síťovou komunikaci a rovněž pro různé matematické programy či programy, využívající umělou inteligenci.

Python se rovněž často používá k programování Raspberry Pi. Právě druhá část jména vychází z názvu Python [AA, str 165].

3.4 Python knihovny pro vývoj GUI

K programování grafického uživatelského rozhraní existuje mnoho různých knihoven. Pro jazyk Python se nejčastěji používá některá z následujících knihoven:

- Kivy
- Tkinter
- Qt

Tkinter

Knihovna tkinter je základní knihovna pro tvorbu okenních aplikací. Na rozdíl od ostatních

² <https://www.cnx-software.com/2016/03/25/esp8266-and-esp32-differences-in-one-single-table/>

knihoven je již obsažená v samotném jazyce Python, nemusí se tedy dodatečně instalovat [AL]. Používání knihovny je tedy zdarma.

Doporučuje se zvláště začátečníkům pro svou jednoduchost. Pro plnohodnotné využití je možné ke knihovně stáhnout dodatečné moduly [AM].

Kivy

Knihovna Kivy patří mezi multiplatformní knihovny. Dle oficiálních stránek knihovny je možné vytvářet aplikace běžící na následujících zařízeních:

- Desktopové počítače: OS X, Linux a Windows
- IOS zařízení: iPad a iPhone
- Android zařízení: tablety a telefony
- Jakékoli jiné dotykové zařízení, podporující TUIO (Tangible User Interface Objects) [AJ]

Knihovna přináší mnoho různých aspektů, které dle autorů knihovny mají zjednodušit tvorbu grafického rozhraní, obsluhu událostí a podobně.

Ke knihovně rovněž patří zvláštní jazyk (nazvaný stejným jménem jako knihovna – jazyk Kivy). Jedním z jeho cílů je oddělení zodpovědnosti za prezentaci (zobrazení) a logiku programu [AK]. Zobrazení je dáno právě souborem kv (napsaném v jazyce Kivy) a logika souborem s Python kódem.

Podobně jako tkinter, je i tato knihovna nabízena zdarma. Musí se však na rozdíl od tkinteru instalovat dodatečně.

4. Zhodnocení současného stavu a plán práce

Smyslem je na základě zhodnocení současného stavu určit cíl práce a vytvořit tak vlastně „detailní zadání“, případně určit předpokládané parametry řešení, ale ne jeho způsob.

Osnova této části může být například: - Kritické zhodnocení dosavadního stavu (co je dobře, co je špatně, co případně není řešeno vůbec a případně cenové parametry, dostupnost řešení, potřebný výpočetní výkon apod.), - návrh, co by bylo vhodné vyřešit na základě znalostí dosavadního stavu a také osobních preferencí, zadání práce, požadavků z praxe apod., - podle možnosti i specifikace práce ve smyslu „co to má dělat“, „jaké to má mít parametry“, „jaké prostředky budou použity“, „jak se to bude vyhodnocovat“, „jak se pozná, že se to povedlo“. Je vhodné do této části práce napsat pravdivou úvahu, zejména v bodě „návrh“ tak, aby zbytek práce byl uvěřitelný a aby po přečtení této kapitoly a poté zbytku textu byl čtenář přesvědčen, že bylo uděláno, co se udělat mělo.

Současný stav

Na českém trhu jsou známy především dva komplexní systémy, disponující s displejem velikosti kolem 7". Jsou to Amazon Echo Show 8 (8" displej) a Google Home Hub (7" displej). Mají však některé nevýhody, které bych chtěl mojí prací vyřešit.

Oba systémy stojí okolo 3 500 Kč. Cena systému pak dále pochopitelně rychle narůstá s různými moduly a zařízeními, které je potřeba k těmto systémům dokoupit (pro plnohodnotnou automatizaci domácnosti) – jde zejména o různá čidla, ovládaná světla, zásuvky a další zařízení. Přitom cena takových přídatných zařízení někdy dosahuje téměř poloviční ceny zmíněných systémů. V mojí práci bych se tedy chtěl zabývat možnostmi, jak snížit cenu jak systému samotného, tak různých přídatných modulů.

Využití systémů kolísá, pokud je budeme chtít umístěny na stěně. Oba jsou totiž vybaveny stojánkem a nemají žádný úchyt ke zdi. Kromě toho jsou i jejich rozměry nevhodné k umístění na stěnu. Pokud budeme chtít vlastnit systém, který by na nejfrekventovanějších místech v domácnosti zobrazoval aktuální stav zařízení a senzorů a umožňoval je ovládat, pak bude vhodné mít nejen přenosná zařízení se stojánkem, ale budou se hodit na pevně nainstalované panely na zdech (zejména v kuchyni a obývacím pokoji). V mé práci bych tedy chtěl vytvořit systém, který nebude problém ani umístit na zeď, ani položit na podstavec. Takový systém pak získá opět trochu nový, „širší“ rozměr.

Další překážkou pak pro mnoho českých uživatelů může být absence české lokalizace systémů automatizace domácnosti. I tento nedostatek bych chtěl svojí prací vyřešit.

<https://www.alza.cz/chytra-domacnost-od-amazonu-a-google>

<https://www.alza.cz/jak-postavit-chytrou-domacnost>

Stanovení cílů/návrh

Cílem práce bude vytvořit systém pro automatizaci domácnosti. Tento systém bude sestávat z několika částí.

Základním prvkem bude centrální jednotka (tzv. hub), která bude řídit jednotlivé spotřebiče. Kromě přímého ovládání bude rovněž umožňovat tvorbu podmínek (na základě snímači naměřených hodnot), za kterých se provede jistá akce a rovněž uchovávat případné časovače nejrůznějších událostí. Centrální prvek bude moci být řízen buďto prostřednictvím připojeného displeje, ale pro pohodlnější přístup rovněž z jiných zařízení, jako je mobilní telefon, notebook a podobně. Displej bude kromě možnosti ovládat systém sloužit jako doplněk (například na zed') ukazující některé zajímavé informace (například hodnoty senzorů).

Dalším prvkem navrhovaného systému budou koncová zařízení, která se budou skládat z nějakého modulu a připojeného senzoru/relé/tranzistoru. Tento modul bude mít některé piny použity jako vstupní a některé jako výstupní. Díky tomu jej bude možné použít jako zařízení:

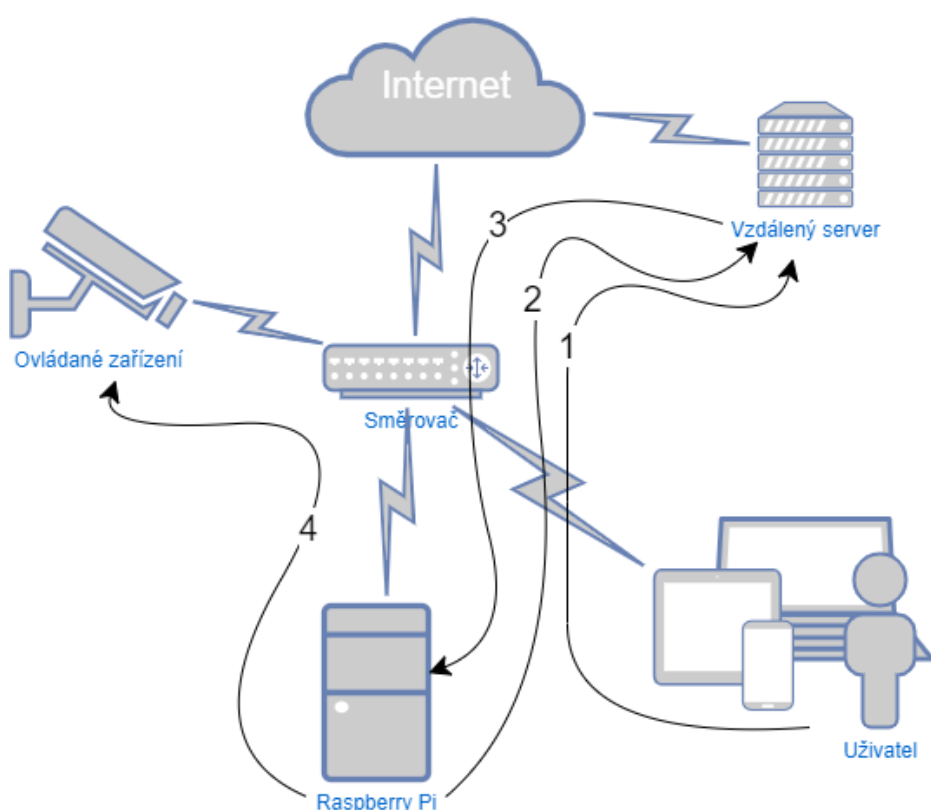
- Vstupní – pro snímání hodnoty senzoru
- Výstupní – pro spínání připojených zařízení, ať již půjde o prosté zapnutí/vypnutí zařízení, nebo ovládání výstupu pulzně šířkovou modulací (stmívání světel, nastavení úhlu natočení servo motorů...)
- Kombinované – jeden modul může obsahovat jak vstupní, tak výstupní prvky

Kombinované zařízení pak přinese velkou výhodu úspory, protože pro ovládání několika světel (a třeba i snímání teploty) pak bude potřeba jen jeden jediný modul, což znamená úsporu financí.

Posledním prvkem systému bude server, přijímající požadavky od zařízení ovládajících domácnost. Tento server bude určen k řízení domácnosti v globálním měřítku, to znamená, aby ji bylo možné ovládat i mimo lokální síť. Za tímto účelem bude obsahovat databázi,

obsahující informace o všech uživateli, o všech jejich připojených zařízeních, o jejich stavu a případné instrukce pro tato zařízení (jako časované spouštění). Každý uživatel se při přístupu k tomuto veřejnému serveru bude muset autentizovat pro zajištění bezpečnosti komunikace po síti (zejména jako ochrana neautorizovaného přístupu k nějakému zařízení a k informacím o domácnosti vůbec).

Celá architektura systému je zobrazena na obr. 3.



Obr.3 – Architektura systému pro automatizaci domácnosti³

Jako technologii bezdrátového přenosu jsem zvolil Wi-Fi. Perfektně se hodí jak pro lokální komunikaci, tak pro komunikaci prostřednictvím internetu (Což je u některých technologií, jako Bluetooth nemožné). V rámci lokální sítě je možné sice použít i jiné technologie, ale protože rozšíření stávající infrastruktury sítě WLAN je snadné, a navíc centrální jednotka (Raspberry Pi) stejně musí komunikovat se směrovačem (bránou do sítě internet) přes WiFi, rozhodl jsem se rozhodl i při komunikaci centrální jednotky s moduly využít Wi-Fi. Systém

³ Vlastní obrázek

sice nebude kvůli absenci některých přenosových technologií možné použít ve spojení s některými již existujícími prvky, které nevyužívají wifi (například chytré žárovky, komunikující přes zigbee), nicméně cílem práce je víceméně nahradit tyto prvky levnější variantou.

Další výhodou je rovněž to, že pro komunikaci prostřednictvím Wi-Fi sítě existuje větší podpora (existující knihovny pro komunikaci touto bezdrátovou technologií a množství modulů, které ji využívají).

Systém by měl být pro uživatele snadno pochopitelný a intuitivní. Celý projekt by mohl být následně uveden na trh jako startup. A právě díky cenové dostupnosti by bylo možné jej rozjet i s omezeným rozpočtem.

5. Realizace a testování

5.1 Volba prostředků

Na základě porovnání různých technologií, které je možné využít při budování systému pro automatizaci domácnosti, a které jsem shrnul v předchozích částech jsem se rozhodl pro následující technické vybavení:

Jako centrální prvek jsem zvolil využít Raspberry Pi verze 3. Tento počítač má dostatečný výkon na obsluhu displeje a již v sobě (ve zvolené verzi 3) obsahuje Wi-Fi přijímač.

K Raspberry Pi bude připojen displej o velikosti 7 palců. Je to dostatečná velikost pro základní přehledné uživatelské rozhraní a zároveň s relativně malou velikostí zamezí velkým nákladům na pořízení budovaného systému. Velikost je ideální jako pro ovládání, tak i pro „přehled“. Bude tak možné letmým pohledem z menší vzdálenosti zkontrolovat například teplotu v některé místnosti, nebo stav světel.

Koncové moduly budou využívat čipu ESP8266. K tomu jsem se rozhodl z toho důvodu, že jejich cena je oproti výkonnějším a novějším ESP32 asi o třetinu nižší. ESP8266 svým výkonem bohatě postačují k jejich účelu v systému. Konkrétně využiji moduly „ESP-01 ESP8266 WIFI“ a „Wemos D1“

Jak již bylo shrnuto v části „Automatizace domácnosti“, existuje mnoho různých systémů ovládání domácnosti, ať již se jedná o ta komerční, která obvykle zahrnují nějaký „chytrý“ centrální prvek, který je možné zakoupit jako plně funkční, prakticky nakonfigurované zařízení, nebo různé programové řešení (obvykle zdarma ke stažení s licencí open source) pro některou platformu jako je Raspberry Pi či BeagleBone. Přestože je některá z těchto řešení možné rozšiřovat, obvykle jsou spíše určeny pro ovládání různých komerčních zařízení, ovládaných přes Wi-Fi, jako jsou zásuvky, žárovky, rychlovarné konvice, ústřední vytápění a podobně. Záměrem mé práce bude se spíše zaměřit na ovládání koncových zařízení, které bude možné připojit k různým spotřebičům (například u modulu, který bude ovládat výstup pomocí relé). Tato koncová zařízení se budou skládat z již zmíněných modulů ESP8266 a dalších modulů/součástí, rozšiřujících jejich funkčnost.

Především půjde o tyto výstupní prvky:

- Relé
- MOSFET tranzistor pro PWM modulaci (jednak pro ovládání intenzity LED osvětlení, ale rovněž pro řízení servo motorů apod.)

Rovněž moduly osadím několika různými senzory, aby tak bylo ovládání domácnosti flexibilnější a uživatel mohl stanovit podmínky provedené různých akcí (jako například sepní topný ventilátor, pokud teplota klesne pod 15 °C, natoč žaluzie při žádané intenzitě slunečního záření a podobně). Za účelem těchto a dalších akcí jsem se rozhodl využít těchto typů senzorů:

- Senzor vlhkosti ovzduší a teploty
- Senzor vlhkosti půdy (pro samo-zavlažování)
- Senzor intenzity osvětlení
- Detektor pohybu (PIR senzor a mikrovlnný senzor doppler)
- Hallův senzor (pro detekci otevření dveří / oken)
- Senzor přítomnosti vody (pro detekci záplavy a podobně)
- Senzor pro detekci hořlavých plynů

Je zřejmé, že konkrétní aplikace senzorů (a jejich kombinace) pak již bude záviset pouze na požadavcích uživatele, jelikož jednotlivé senzory mohou mít různé využití. Například pohybové čidlo může sloužit jako plnohodnotný zabezpečovací systém, stejně dobře, jako spínač osvětlení v místnosti – případně může plnit funkci obojího současně. Ve chvíli, kdy bude místnost zakódovaná tak bude senzor sloužit pro ochranu před vloupáním, zatímco v době odkódování bude (v případě přítomnosti v místnosti) spínat světlo.

Programové vybavení

Jako programovací jazyk pro Raspberry Pi i android jsem se rozhodl použít Python (vzhledem k...??) a programovací prostředí Geany IDE, jelikož jej standardně Raspbian zvolené verze obsahoval a nebylo nutné instalovat další nástroje. Pro moduly ESP jsem zvolil jazyk Wiring a Arduino IDE, jelikož je jednoduché a přehledné a podporuje tyto moduly (bylo však nutné ve správě desek doinstalovat zvolený modul Wedos1-nebo jak že se to jmenuje...).

V této práci se chci zaměřit především na nízkou cenu výsledného systému, s možností komerčního prodeje. Z tohoto důvodu jsem při volbě GUI knihovny vyloučil využití Qt, která sice nabízí mnoho skvělých nástrojů (například GUI builder), avšak cena licence pro komerční využití je příliš vysoká. Tkinter je zase příliš omezená co se týče různých funkcí, které budou potřeba implementovat (například přenositelnost na android je problematická). Rozhodl jsem se tedy využít knihovnu Kivy, která obsahuje množství užitečných funkcí a aspektů (jako multiplatformní přístup, zabudované funkce pro notifikace na androidu atd).

5.2 Aplikace pro Raspberry Pi

Konfigurace Raspberry PI

Jak již bylo zmíněno, bude počítač Raspberry Pi v systému figurovat jako centrální jednotka. Nebude se však jednat o nějaké zařízení běžící někde v pozadí, ale bude vybaven displejem pro samotné ovládání domácnosti.

Za účelem ovládání je potřeba nakonfigurovat Raspberry Pi tak, aby se skutečně jevil jako vestavný systém a uživatel s ním neinteragoval jako s počítačem, ale jako s jednoúčelovým zařízením. Vyhneme se tak i tomu, že systém bude nestabilní/zranitelný. Například by uživatel vůbec neměl mít možnost vypnout hlavní aplikaci a dostat se na plochu zařízení (na němž poběží OS Raspbian). Za tímto účelem je tedy potřeba následující:

- Aplikace se musí (v grafickém módu) bez požadavku uživatele zapnout ihned po spuštění Raspberry Pi
- Aplikace musí běžet ve full-screen módu
- Veškerá potřebná nastavení systému (zejména jde o připojení k síti Wi-Fi) musí být prováděna přímo v aplikaci a nikoli v nastavení operačního systému
- Aplikace nesmí být snadno ukončitelná

Spuštění aplikace ihned po spuštění OS jde zajistit několika způsoby. Jeden z nich, který použiji spočívá v úpravě souboru „/etc/rc.local“. Otevřu jej v konzoli s právy administrátora pomocí editoru nano:

```
sudo nano /etc/rc.local
```

Následně na konec souboru připojím příkaz pro spuštění python skriptu (opět s právy administrátora). Nakonec ještě připojím ampersand, aby skript neblokoval činnost OS, ale byl spuštěn na pozadí:

```
sudo python /home/pi/sample.py &
```

Spouštění ve full-screenu zajistíme použitím framework Kivy pro tvorbu grafického uživatelského rozhraní, který aplikace standardně spouští ve full-screen módu a není potřeba nic dodatečně nastavovat (aktivace/deaktivace fullscreen módu je dále možná v konfiguračním souboru kivy)

Připojení k WiFi zajistím programově konkrétní implementací, viz dále v části **Realizace**.

Ukončení aplikace (bez vypnutí operačního systému) zajistím deaktivací USB portů Raspberry

Pi. Tuto deaktivaci je možné provést až před nasazením systému do reálného prostředí, protože Raspberry Pi již nebude reagovat na periferie jako klávesnici či myš, a nebude tedy možné provádět změny v systému.

Funkce aplikace

Před samotným návrhem grafického uživatelského rozhraní je potřeba stanovit veškeré funkce systému.

Rozhodl jsem se, že systém bude obsahovat všechny důležité funkce, které budou nutné pro plně funkční, rozšiřitelný systém ovládání (případně automatizace) domácnosti. Bude tedy obsahovat funkce:

- Nastavení „módu“ domácnosti (jako doma, v práci, film, večer...)
- Nastavení podmínek, za kterých dojde k určité akci (pokud je venku světlo, natoč žaluzie)
- Okamžité nastavení stavu některého ze zařízení (například okamžité zapnutí konkrétního světla)

Dále v aplikaci bude správa zmíněných módů (přidávání, úprava, mazání), stejně tak podmínek a jednotlivých zařízení.

Kromě těchto základních funkcí samozřejmě musí mít aplikace (pro komerční nasazení) funkční části, jako přihlašování a registrace uživatelů, správa domácnosti a jejích prvků a různá nastavení.

Rozvržení a grafický návrh aplikace

Aplikace bude rozdělena na několik samostatných částí (oken), mezi kterými bude uživatel přecházet. Jako úvodní obrazovka při prvním spuštění se zobrazí stránka, na které si uživatel zvolí Wi-Fi síť ke které se má Raspberry Pi připojit. Dalším oknem bude přihlašovací okno. Zde bude možnost se rovněž dostat k registraci. Po přihlášení se uživatel dostane do hlavního okna aplikace, s přehledem o jednotlivých místnostech a možnosti aktivovat/deaktivovat pravidla automatizace. Také zde bude menu, přes které se bude možné dostat do správy místností, správy zařízení, správy podmínek a nastavení aplikace.

Volba Wifi Sítě

Zde uživatel pouze zvolí SSID přístupového bodu, vypíše heslo a stiskne tlačítko k připojení. Pokud se již na daném Raspberry Pi uživatel někdy připojoval k Wi-Fi (například pokud již bylo zařízení využíváno k jiným účelům a uživatel tam aplikaci nainstaluje) a zvolí dané SSID,

doplní se automaticky heslo samo (z konfiguračního souboru wpa_supplicant.conf, obsahujícího hesla všech uložených Wifi sítí). Je důležité, aby měl uživatel možnost vybrat síť, i kdyby byl v tu chvíli již k nějaké připojený, protože automatizace bude fungovat v rámci jedné konkrétní Wifi sítě – a jelikož může mít uživatel v dosahu více dosažitelných sítí, je potřeba zvolit, ve které bude vytvořena síť ovládaných zařízení. Dále zde ještě bude tlačítko pro opětovné skenování sítě (pro případ, že by z nějakého důvodu jeho přístupový bod nebyl nalezen – například při slabším signálu). V případě že se nejedná o první spuštění, tak se program pokusí automaticky připojit k Wi-Fi síti, ke které byl připojen při posledním spuštění a tato úvodní obrazovka se uživateli vůbec neukáže. Toho je dosaženo tak, že bude uložený údaj o Wifi v konfiguračním souboru aplikace. Samozřejmě bude později v aplikaci změnit Wifi síť.

Přihlašování

Jak již bylo naznačeno, aplikace se bude připojovat ke vzdálenému serveru, který bude centrálně spravovat uživatele. Za tímto účelem bude mít každý uživatel svůj účet. Po úspěšném připojení k síti se tedy uživatel dostane na novou stránku s přihlášením do svého účtu. Zde vyplní uživatelské jméno a heslo. Případně zde bude tlačítko pro registraci nového uživatele. Jelikož obvykle není důvod, aby se uživatel z takové aplikace odhlašoval, zůstane trvale přihlášen, dokud se explicitně neodhlásí v menu aplikace a při příštím spuštění se (podobně jako v případě připojení k Wi-Fi síti) uživatel přihlásí do posledně přihlášeného účtu – nemusí tedy údaje zadávat po každém vypnutí zařízení (Někde zmínit že zařízení bude určeno k trvalému zapnutí – bude se vypínat jen výjimečně!!!). Nakonec zde bude tlačítko pro návrat zpět k volbě Wifi sítě.

Registrace

Na stránce registrace budou podobně jako na přihlašovací pouze 2 vstupní pole – zvolené přihlašovací jméno (email uživatele) a heslo. Po jejich vyplnění uživatel své údaje potvrdí tlačítkem a pokud nedojde k žádné chybě (jako zadání již existujícího emailu), dostane se opět na přihlašovací stránku. V opačném případě se mu ve vyskakovacím okně zobrazí potvrzovací upozornění o chybě. Opět zde bude rovněž tlačítko pro návrat zpět (k přihlášení)

Domovská obrazovka

Po přihlášení se uživatel dostane na domovskou stránku své domácnosti. Zde bude

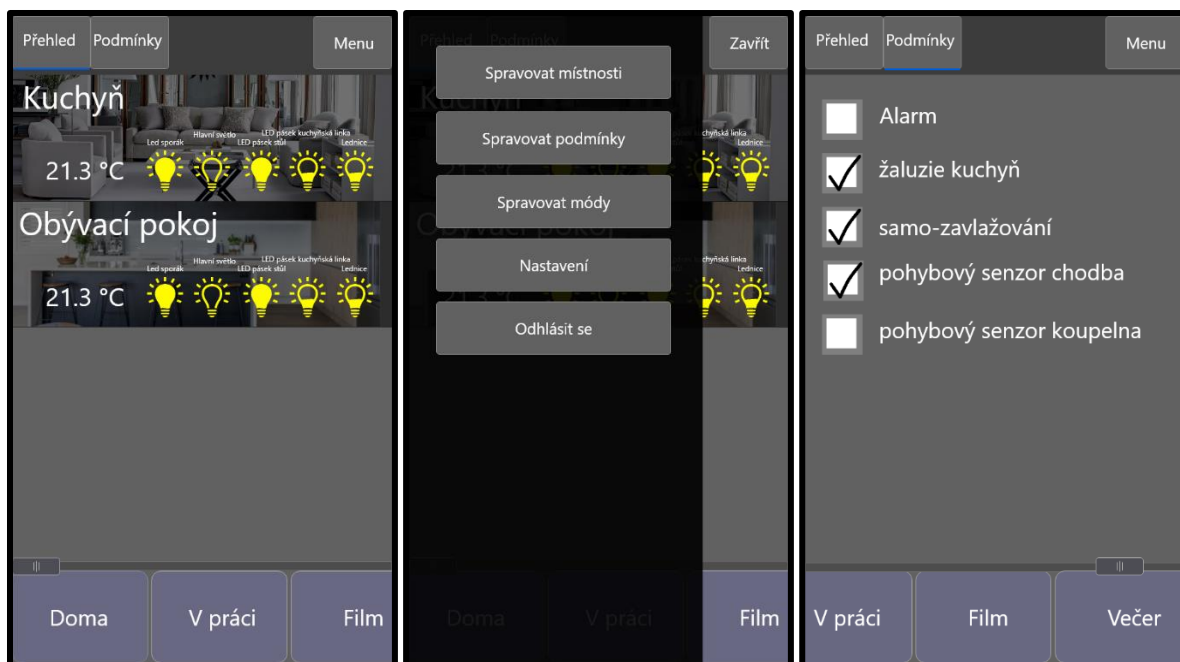
obrazovka rozdělena na 3 pomyslné části. V první budou vlevo dvě záložky („Přehled“ a „Podmínky“), kterými bude uživatel přepínat mezi přehledem místností a aktivací podmínek. Napravo pak bude tlačítko pro rozvinutí vysunovacího menu z levé strany (popisek tlačítka se při otevření menu změní na „Zavřít“ s odpovídající funkcí).

Pod touto první částí bude „hlavní“ část aplikace. Půjde v podstatě o jádro aplikace, protože se zde bude program nacházet nejčastěji. V případě že bude aktivována záložka „Přehled“, pak v této hlavní části bude přehled jednotlivých místností a zařízení jim přiřazeným. Jednotlivé místnosti budou zobrazené v řádcích pod sebou. Každá bude mít přiřazené pozadí (to si uživatel volí při vytváření místnosti) a název.

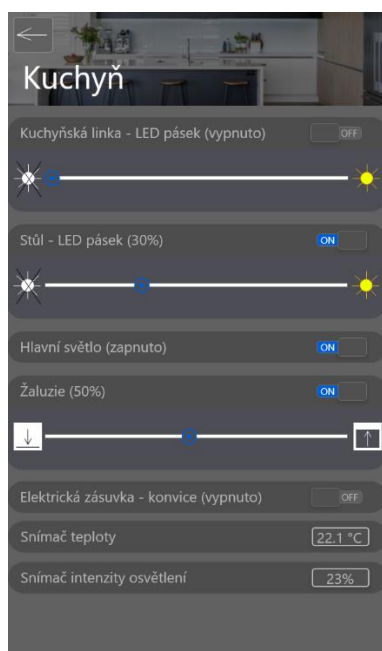
Dále bude pro každou místnost zobrazena teplota (bude-li mít přiřazeno zařízení se snímačem teploty), vlhkost (opět, bude-li v místnosti zařízení se snímačem vlhkosti) a nakonec seznam čidel a výstupů. Bude-li těchto zařízení méně než 5, budou v řadě vedle sebe s pevnou pozicí. Bude-li jich mít místnost přiřazených více, budou umístěny na posuvném panelu tak, že bude současně zobrazeno vždy 5 zařízení. Nad každým zařízením bude malým písmem název každého se zařízení (jako například „LED stůl“, „světlo sporák“, „žaluzie“ apod).

Pokud uživatel aktivuje záložku „Podmínky“, tak se hlavní část aplikace změní na seznam podmínek (opět v řádcích pod sebou, podobně jako u seznamu místností), definovaných uživatelem. Pro každou podmínku se zde bude nacházet zaškrtačací políčko pro volbu, zda je pravidlo aktivní (například někdy uživatel bude chtít, aby některé podmínky platily jen když je doma a podobně), vedle něj název pravidla, který uživatel zadává při vytváření (příp. editaci) pravidla a dále tlačítko pro otevření detailu pravidla. Tento detail bude sloužit pro zobrazení celé podmínky tak, jak je definovaná (v přehledu podmínek jsou kvůli přehlednosti pouze názvy), případně k úpravě či smazání.

Jak přehled místností, tak i přehled podmínek bude mít vyhrazený prostor tak, aby pod těmito přehledy zůstalo ještě místo na třetí část domovské stránky. V této menší části budou tlačítka s názvy módů, kterými uživatel může nastavit žádaný režim domácnosti. Ty se nebudou skládat v řádcích, ale naopak ve sloupcích. Pokud jich bude více než 5, budou opět na posuvném panelu.



Detail místnosti



Nastavení

V nastavení aplikace bude možné ovlivnit některé vlastnosti aplikace. Například půjde nastavit, aby se po přepnutí záložky na „Podmínky“ po minutě nečinnosti aplikace přepnula zpět na přehled místností (pro případ, když uživatel „zapomene“ přepnout záložku zpět, aby i tak až příště půjde okolo zařízení měl přehled).

Správa místností

Správa místností bude vypadat podobně, jako přehled místností s tím rozdílem, že nebude určena k ovládání, ale k jejich úpravě. Vedle každé již přidané místnosti budou dvě tlačítka pro posun nahoru/dolu. To proto, že při více místnostech nebudou některé (bez posunutí) vidět, tak aby uživatel mohl zvolit, o kterých chce mít primárně přehled i letmým pohledem na display bez nutnosti zásahu. Tlačítko nahoru/dolu bude sloužit o posun o jednu pozici. Na ploše jednotlivých místnosti (vymezené obrázkem na pozadí) bude seznam již přidaných zařízení. Na poslední pozici bude tlačítko pro přidání dalšího zařízení. Při kliknutí na některé zařízení se zobrazí vyskakovací okno, ve kterém bude možné spravovat zvolené zařízení. Bude možné změnit název zařízení, místnost, ve které se nachází (pomocí vysunovací nabídky – při změně se zařízení odebere ze stávající místnosti a přidá do nové), případně i ikonu zařízení (z předem definovaných).

Správa podmínek

...

Správa módů

...

Způsob ukládání dat

V systému ovládání domácnosti je třeba nějakým způsobem zajistit persistenci dat. Toho je možné dosáhnout buď ukládáním informací do textového/binárního souboru či využitím některého databázového systému. K implementaci jsem se rozhodl využít druhé možnosti, tedy databázový systém, a to zejména kvůli snazší správě a rychlejšímu zpracování dat.

Na výběr existuje mnoho různých databázových systémů. Buďto je možné vybírat mezi relačními nebo tzv. NoSQL databázemi.

Při návrhu systému jsem musel zvážit, kde budou data uložena. Nabízejí se totiž celkem 3 možné řešení:

- Na Raspberry Pi
- Na vzdáleném serveru
- Na obou zmíněných zařízeních

První řešení by představovalo problém kvůli neveřejným (privátním) IP adresám, které má přiřazena většina domácností. Aby bylo Raspberry Pi a jeho databáze přístupné globálně ze

sítě internet, bylo by zapotřebí, aby si uživatel u svého poskytovatele objednal veřejnou IP adresu. Ne každý poskytovatel internetového připojení ji nabízí, a i když ano, tak by se veřejná IP adresa stala dalším výdajem pro uživatele takového systému. A to nikoli jednorázovým, ale měsíčním. Takový systém by nepůsobil dobrým dojmem. Kromě toho i kdyby uživatel veřejnou IP adresu měl, tak by se muselo jednat o statickou adresu, jinak by tu byl další problém v podobě měnící se adresy, ke které by se nebylo možné spolehlivě připojovat. Nakonec i kdyby si uživatel připlatil (případně již měl) veřejnou, statickou IP adresu, tak je tu ještě jedna obtíž. Pro dosažení databáze v Raspberri Pi by bylo nutné v administraci směrovače nastavit přesměrování portu na Raspberry Pi. Systém však má být určený i pro uživatele bez technických dovedností a měl by fungovat ihned po prvním spuštění bez zbytečných konfigurací.

Druhé řešení je sice možné, ale s jistými komplikacemi. **Tou první je, že....** Nakonec je tu i bezpečnostní riziko, pokud budou uživatelé spravování centrálně.

Poslední řešení se snaží řešit některé nedostatky obou předchozích. Raspberry Pi by mělo svoji vlastní databázi (v lokálním úložišti pro plnění příkazů) se kterou by komunikovalo a tu by následně zrcadlilo do veřejné. Z této veřejné databáze by pak rovněž získávala aktualizovaná data (pro možnost globálního ovládání přes internet). Jelikož je tu Raspberry Pi v roli klienta, který se databáze na straně serveru dotazuje a rovněž mu posílá aktualizace, není tu žádný problém s potřebou vlastnit veřejnou (či statickou) IP adresu na straně uživatele. Tato duplicitní databáze má za výhodu i to, že systém bude fungovat i při jakýchkoli problémech s připojením k internetu (kolísání rychlosti, případná odpojení).

Tento způsob sice neřeší bezpečnost dat plně, ale vzhledem ke zvolené technologii (firebase databáze) se stará o zabezpečení dat společnost Google.-Smazat nebo upravit...až dále se zminuji o volbě firebase...možná to zmínit později jako další vyhodu firebase???

Z těchto důvodů je nutné, aby byla data na dvou místech současně (v lokálním úložišti Raspberry Pi pro plnění příkazů a zároveň na veřejném serveru pro možnost globálního ovládání přes internet). V případě ukládání/aktualizace dat je tedy třeba rozhodnout, která databáze bude aktuálnější:

-buďto veřejná a Raspberry Pi si z ní bude zrcadlit data do svojí lokální databáze

-nebo lokální s tím, že na server se budou data do databáze na veřejném severu zrcadlit s malým zpožděním

Jelikož je obvykle důležitější rychlá odezva v případě přítomnosti osoby v dané domácnosti než na dálku (například určitě je důležitější, aby po povelu k zapnutí nějakého zařízení se toto ihned zapnulo, kdežto při například zhasnutí světla, které bylo opomenuto při odchodu z domu nepotřebujeme obvykle "bleskovou" reakci), bude se systém snažit nejprve ukládat data na lokální disk a teprve po sléze je odesílat na veřejný server. Na to je třeba rovněž pamatovat v mobilní aplikaci, která musí nejprve zkontrolovat, zda se nalézá v lokální síti spolu s Raspberry Pi a pokud ano, tak provede zápis na lokální disk. V opačném případě data uloží do veřejné databáze, odkud si je Raspberry Pi bude snažit nahrát.

Někdy tedy budeme chtít, aby primárně platila data v Raspberry Pi a jindy na veřejném serveru. Tuto "dvojí" prioritu je potřeba vzít v potaz i při návrhu relační databáze a ukládat spolu s daty značku, která rozhodne o prioritě (aktuálnosti) dat. Z pochopitelných důvodů se bude jednat o časový údaj, kde pro snadné porovnávání využijí tzv unixový čas.

Pro funkci programu bude potřeba v databázi uchovávat několik relací, které budou podrobně vysvětleny dále:

- relace uživatelů
- relace domácností
- relace zařízení

Konfigurace kivy pro vyhovující GUI aplikace

Knihovna Kivy není standardní součástí jazyka Python, musel jsem ji tedy nainstalovat dodatečně, podle oficiálních stránek této knihovny⁴.

Na zvoleném displeji nepřijímala aplikace doteky jako vstup. Musel jsem tedy upravit konfigurační soubor knihovny kivy (/home/pi/.kivy/config.ini). Sekci [input] bylo třeba nahradit instrukcemi:

```
mouse = mouse
mtdev_(name)s = probesysfs,provider=mtdev
hid_(name)s = probesysfs,provider=hidinput
```

⁴ <https://kivy.org/doc/stable/installation/installation-rpi.html>

V tomto souboru bylo potřeba rovněž v sekci [graphics] nastavit, aby se aplikace spouštěla ve full-screen módu a

```
fullscreen = 1
```

Rovněž bylo potřeba v této sekci nastavit, aby se nezobrazoval kurzor myši (nepůsobí to totiž dobrým dojmem – uživatel by měl mít ze zařízení podobnou zkušenost, na jakou je zvyklý z mobilní aplikace):

```
show_cursor = 0
```

Nakonec jsem ještě v sekci [graphics] nastavil rotaci aplikace – rozhodl jsem se totiž, že bude aplikace běžet v „portrétovém“ módu:

```
rotation=90
```

Implementace

Detaily implementace....

5.3 Volba a konfigurace veřejného serveru

Jelikož jsem si vytkl za cíl vytvořit systém, který bude umět komunikovat nejen v lokální síti, ale i dálkově mimo ni, bude za potřebí vytvořit veřejnou část systému. Zde se mi nabízely v podstatě dvě možná řešení:

- Vytvořit program na virtuálním privátním serveru, který by v reálném čase zpracovával požadavky klientů a reagoval na ně
- Nebo vytvořit webovou stránku (případně jen databázi), která bude reagovat na uživatelské potřeby

První řešení by bylo sice z pohledu zpracování dat rychlejší (Raspberry Pi by mohlo udržovat trvalé spojení se serverem a v reálném čase by si vyměňovalo informace), ovšem za cenu vyšších provozních nákladů a složitějšího kódu. Cena pronájmu VPS od českých firem se pohybuje v řádu stovek korun. Případnou levnější variantou by bylo provozovat takový server přímo na Raspberry Pi. Potom by zcela odpadla nutnost řešit nějaký vzdálený server i přesto, že by byl zachován vzdálený přístup. Výhodou by pak bylo, že by nebylo nutné nijak zrcadlit databázi z Raspberry Pi (jak bylo uvedeno v části Raspberry Pi) a to by tak bylo jediným místem, kde by data byla uchovávaná. To by s sebou přinášelo i větší bezpečnost dat, jelikož by uživatelé nebyli nijak centrálně spravováni (jako v případě veřejného VPS nebo webové stránky s databází). Toto řešení má ovšem 2 potencionální problémy:

Ne každá domácnost má k dispozici veřejnou IP adresu (a někteří poskytovatelé internetového připojení ji dokonce ani nenabízejí) a její případné pořízení se pak stává dalším výdajem. Pokud pak domácnost veřejnou IP adresu má, tak by dalším problémem mohlo být, kdyby se jednalo o adresu dynamickou, což by komplikovalo vzdálené připojování.

Druhým problémem pak je, že by se takový systém víceméně nedal použít jako komerčně vyráběný pro laickou veřejnost kvůli potřebě složitého nastavení takového systému před prvním použitím (nutnost nastavovat v administraci směrovače sítě propouštění portů a další podrobnosti). Zajistě si každý koupí raději jednoduchý systém, kterému k práci stačí připojit napájení a funguje a ne takový, který by se spojoval s dalšími poplatky s instalací a montáží.

Na druhou stranu druhé řešení serveru v podobě webhostingu a databáze by bylo finančně mnohem úspornější, protože je nabízejí některé firmy i zdarma (obvykle s jistým omezením, jako vložená reklama na webu). Vzhledem k tomu, že je cílem mé bakalářské práce vytvořit finančně nenáročný systém, který by případně i bez velkých počátečních či provozních nákladů bylo možné vyrábět v rámci startup projektu, a je potřeba aby byl systém snadno konfigurovatelný i pro lidi bez technické zdatnosti, rozhodl jsem se, že zvolím druhou variantu, tedy databázi. Při volbě konkrétní služby jsem porovnával různé parametry, zejména kapacitu databáze a její cenu. Nakonec jsem se rozhodl použít real-time databázi Firebase database od společnosti Google. Tato služba je dostupná ve dvou různých plánech (licencích). Licence zdarma („Spark plan“) zahrnuje uložení pro 1GB dat, což je pro plánované využití více než dostatečné. Je zde však rovněž limit přenesených dat, který pro licenci zdarma činí 10GB/měsíc. Později při testování tedy bude nutné otestovat velikost dat pro „průměrnou“ domácnost (pro odhad na kolik domácností by takový systém stačil) a dále přibližnou velikost dat přenesených za měsíc při aktivním používání. Výhodou je, že je snadné přestoupit na placenou licenci („Blaze plan“), která je standardně zdarma a platí se jen to, u čeho uživatel databáze překročí limit neplacené licence (přičemž uživatel databáze je zde případná společnost, která by navrhovaný systém automatizace domácnosti prodávala, nikoli koncový uživatelé). U placené licence je rovněž možnost zálohy dat každý den.

Konfigurace serveru

Databáze

5.4 Aplikace pro koncové moduly

Koncové moduly budou v systému hrát úlohu ovládaných zařízení a současně snímat měřené veličiny. Přičemž jako (koncový) modul zde rozumím jednotlivé ESP moduly. Každý z těchto modulů může v sobě obsahovat více různých vstupních i výstupních zařízení, jak již bylo naznačeno dříve.

Koncové moduly mají celkem dvě žádané funkce:

- Obsluhovat uživatelské povely z Raspberry Pi – jako rozsvítit konkrétní světlo, natočit žaluzie apod.
- A odesílat data z připojených snímačů na Raspberry Pi

Je tedy zřejmé, že v prvním případě je ESP modul v roli serveru, který obsluhuje jediného klienta – Raspberry Pi.

Odesílat data na Raspberry Pi se budou v případě, kdy bude k modulu připojen nějaký senzor, jehož hodnotu bude Raspberry Pi potřebovat pro řízení domácnosti či k zobrazení kvůli přehledu (například teplota místnosti). Zde bude potřeba rozlišit dva různé modely odesílání dat, které jsem musel při návrhu aplikace pro ESP moduly zvážit:

- a. Odesílání dat v pravidelných intervalech
- b. Impulzivní odesílání dat

Odesílání dat v pravidelných intervalech využiji pro senzory, jejichž hodnota se v průběhu času plynule mění a je potřeba ji zaznamenávat. Například může jít o teplotní senzor, jehož hodnotu není potřeba číst po každé maličké změně (impulzivně), ale například každých 10 minut.

Impulzivní odesílání dat se naopak hodí například v případě různých událostí, na které je potřeba okamžitě reagovat. Půjde především o všechny bezpečnostní prvky, jako jsou snímače kouře, zatopení apod.

Pro ještě větší bezpečnost systému jsem se však rozhodl, že i prvky s impulzivním odesíláním dat je potřeba v pravidelných intervalech kontrolovat – pro případ, že by došlo k nějaké chybě a při vzniku dané události nebylo Raspberry Pi informováno.

ESP moduly mají relativně malou paměť, a tak jsem navrhl rozdělit co nejvíce zátěže na Raspberry Pi, které má výkon dostatečný. Proto nebudou tyto moduly obsahovat žádný

mechanismus odesílání dat na Raspberry Pi. Místo toho bude o data v daných časových intervalech žádat samotné Raspberry Pi, jak již bylo zmíněno dříve u návrhu aplikace pro Raspberry Pi.

Budu používat synchronní komunikaci-normálně by to vadilo, zde ne (jen jeden klient – mobil není klient).

6. Závěr

Cca 1 strana závěr

Náměty na pokračování práce?

Hlasové ovládání

Více domácností

Různá oprávnění u ovládání prvků (například ve firmě se dá využít, nebo jako rodičovský zámek)

Multifunkční stojan-bylo navrhnuo zařízení, ale je potřeba ještě dořešit umístění (na stěně i stole)

Přehrávání multimedií (přes Bluetooth??)

Použití sběrnic ESP – připojení různých senzorů

7. Seznam použité literatury

- [A] Wi-Fi, Bluetooth, ZigBee and WiMax
- [B] Bezdrátové sítě WiFi (Zandl Patrick)
- [C] ITU Radio Regulations (<https://www.itu.int/pub/R-REG-RR/en>)
- [D] <https://www.minim.co/blog/wifi-frequency-bands-2.4-ghz-and-5-ghz>
- [E] Designing Embedded Systems with PIC Microcontrollers: Principles and Applications (Tim Wilmshurst)
- [F] Embedded Systems: Architecture, Programming and Design (Raj Kamal)
- [G] <https://tfe.cz/ctyrkanalovy-prijimac-rfc4-rx.htm>
- [H] Embedded Systems Circuits and Programming (Julio Sanchez, Maria P. Canton)
- [I] Microprocessor and Interfaces (A.P.Godse, D.A.Godse)
- [J] <https://circuitdigest.com/article/what-is-the-difference-between-microprocessor-and-microcontroller>
- [K] <https://www.arduino.cc/en/guide/introduction>
- [L] Arduino Programming with .NET and Sketch (Agus Kurniawan)
- [M] http://coolcosmos.ipac.caltech.edu/cosmic_classroom/cosmic_reference/whatisir.html
- [N] Clinical Optics and Refraction (Andrew Keirl)
- [O] The World of Physics (John Avison)
- [P] Electronic, Magnetic, and Optical Materials, Second Edition (Pradeep Fulay, Jung-Kun Lee)
- [Q] Wireless Communications (T. L. Singal)
- [R] Guide to Computer Network Security (Joseph Migga Kizza)
- [S] Home Automation Made Easy: Do It Yourself Know How Using UPB, Insteon, X10 and Z-Wave (Dennis C Brewer)
- [T] <https://www-sciencedirect-com.ezproxy.lib.vutbr.cz/science/article/pii/S1084804517303533>
- [U] <https://www-sciencedirect-com.ezproxy.lib.vutbr.cz/science/article/pii/S0045790615000257>

- [V] <https://www.chytrevypinace.cz/Sonoff-Basic-d1.htm>
- [W] Microwave Engineering (V.S.Bagad)
- [X] <https://www.indiegogo.com/projects/why-2-4ghz-chasing-wireless-history#/>
- [Y] **Síťové aplikace a jejich architektura (Petr Matoušek)**
- [Z] Naučte se Python – pohotová příručka (David Ascher)
- [AA] Raspberry Pi – uživatelská příručka (Eben Upton)
- [AB] Bluetooth Application Developer's Guide (David Kammer, Gordon McNutt, Brian Senese, Jennifer Bray)
- [AC] Wireless Personal Communications: Simulation and Complexity (Mohsen A. M. El-Bendary)
- [AD] Getting started with Bluetooth (Madhushree Ganguli)
- [AE] <https://www.espressif.com/en/products/hardware/esp8266ex/overview>
- [AF] <https://www.espressif.com/en/products/hardware/esp32/overview>
- [AG] Hands-On ZigBee: Implementing 802.15.4 with Microcontrollers (Fred Eady)
- [AH] Zigbee Wireless Networking (Drew Gislason)
- [AI] Analog Communication (A.P.Godse U.A.Bakshi)
- [AJ] <https://kivy.org/doc/stable/gettingstarted/intro.html>
- [AK] <https://kivy.org/doc/stable/guide/lang.html>
- [AL] <http://tkinter.programujte.com/tkinter-whats-tkinter.htm>
- [AM] <https://www.py.cz/Tkinter>