

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií



Diplomová práce

Inteligentní domácnost – chytré zrcadlo

Bc. Jakub Zíta

© 2019 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Jakub Zíta

Informatika

Název práce

Inteligentní domácnost – chytré zrcadlo

Název anglicky

Intelligent household – smart mirror

Cíle práce

Diplomová práce je tematicky zaměřena na problematiku vlastního chytrého zrcadla v domácnosti. Hlavním cílem práce je analýza, komparace a vyhodnocení variant SW a HW řešení chytrých zrcadel s následným návrhem implementace. Dílčí cíle práce jsou:

- vypracování přehledu možných řešení pro konstrukci smart mirror,
- vypracování přehledu vývoje v oblasti IoT.

Metodika

Metodika řešené problematiky je založena na studiu a analýze odborných informančních zdrojů. Vlastní práce spočívá v analýze a na základě zvolených kritérií objektivní komparaci vybraných komponent pro řešení problematiky smart mirror s následným návrhem realizace. Na základě syntézy teoretických poznatků a výsledků praktické části budou formulovány závěry diplomové práce.

Doporučený rozsah práce

60 – 80 stran textu.

Klíčová slova

Internet věcí, IoT, inteligentní domácnost, raspberry pi, senzor, infrastruktura, chytré zrcadlo, bezdrátová síť, mikropočítač

Doporučené zdroje informací

GREENGARD Samuel. The Internet of Things (The MIT Press Essential Knowledge series). The MIT Press. ISBN: 978-0262527736

LEA Perry. Internet of Things for Architects: Architecting IoT solutions by implementing sensors, communication infrastructure, edge computing, analytics, and security. Packt Publishing. ISBN: 978-1788470599

NORRIS Donald. The Internet of Things: Do-It-Yourself at Home Projects for Arduino, Raspberry Pi and BeagleBone Black. McGraw-Hill Education TAB. ISBN: 978-0071835206

SHOVIC John C. Raspberry Pi IoT Projects: Prototyping Experiments for Makers. Apress; 1st ed. edition. ISBN: 978-1484213780

Předběžný termín obhajoby

2018/19 LS – PEF

Vedoucí práce

Ing. Pavel Šimek, Ph.D.

Garantující pracoviště

Katedra informačních technologií

Elektronicky schváleno dne 11. 9. 2018

Ing. Jiří Vaněk, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 19. 10. 2018

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 04. 03. 2019

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Inteligentní domácnost – chytré zrcadlo" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 17. března 2019

Poděkování

Rád bych touto cestou poděkoval Ing. Pavlu Šimkovi PhD. za jeho rady a pomoc při tvorbě diplomové práce.

Inteligentní domácnost – chytré zrcadlo

Abstrakt

Společně se šířením technologie internetu věcí, což je v informatice označení pro síť zařízení, vozidel, domácích spotřebičů a dalších zařízení, roste zároveň i popularita jednoho z odvětví této oblasti, konkrétně chytrých domácností. Kromě komerčních zařízení, které nabízí mnoho již zavedených firem se, ale do popředí dostávají i vlastní řešení, která kombinují především nízké náklady a možnost vytvořit zařízení přesně podle představ uživatele. Na trhu existuje široký výběr komponent, které je možné pro realizaci využít a každý si tak může s určitým množstvím technických znalostí vytvořit nejen jedno chytré zařízení, ale celý chytrý domácí systém.

Práce samotná se zabývá konkrétně možnostmi chytrého zrcadla, to samo není na trhu v komerční podobě zatím téměř zastoupeno a otevírají se tedy široké možnosti, jak k danému konceptu přistoupit. Na základě anonymního dotazníku tedy byly zjišťovány názory potencionálních uživatelů, jak by podle nich mělo ideálně chytré zrcadlo vypadat a fungovat, z těchto poznatků bylo několik variant definováno a ty byly vyhodnoceny pomocí vícekriteriální analýzy. Pro vybrané varianty byl následně vytvořen návrh realizace jak hardwaru, tak i softwaru. Součástí těchto návrhů bylo využití virtuálního asistenta Google Home, mikropočítače Raspberry Pi a programovacího jazyka Python.

Klíčová slova: internet věcí, inteligentní domácnost, chytré zrcadlo, Google Home, Raspberry Pi, senzor, DS18B20, DHT11, HC-SR501, Python, vícekriteriální analýza variant

Intelligent household – smart mirror

Abstract

Together with the expansion of the Internet of Things Technology, which is in the informatics label for the network of devices, vehicles, household appliances and other devices, the popularity of one of the sector's branches, namely smart households, is growing. In addition to the commercial devices offered by many already established companies, they also come to the forefront self-made solutions that combine mainly low costs and the ability to create devices exactly according to the user preferences. There is a wide selection of components on the market that can be used for implementation and everyone can create not only a clever device but a whole smart home system with a certain amount of technical knowledge.

The work itself deals specifically with the possibilities of a smart mirror, because it is not yet almost represented in the commercial form in the market, and thus opens wide possibilities of approaching the concept. Based on an anonymous questionnaire, therefore, the views of potential users were identified, as they would have the ideally smart mirror look and function, several variants of these findings were defined, and these were evaluated using a multi-criteria analysis. For the selected variants, the design of both hardware and software was subsequently created. These designs included the use of Google Home Virtual Assistant, Raspberry Pi microcomputers, and Python programming language.

Keywords: Internet of things, smart home, smart mirror, Google Home, Raspberry Pi, sensor, DS18B20, DHT11, HC-SR501, Python, multi-criteria analysis of variants

Obsah

1	Úvod	12
2	Cíl práce a metodika	13
2.1	Cíl práce	13
2.2	Metodika	13
3	Teoretická východiska	14
3.1	Internet věcí (IoT)	14
3.2	Chytrá domácnost.....	17
3.2.1	Historický vývoj a budoucnost chytrých domácností	18
3.2.2	Základní prvky chytré domácnosti.....	21
3.2.2.1	Energie.....	21
3.2.2.2	Zabezpečení.....	21
3.2.2.3	Pohodlí.....	22
3.2.2.4	Atmosféra	22
3.2.2.5	Zábava	23
3.2.3	Komunikační protokoly	23
3.2.4	Ovládání a řízení chytré domácnosti.....	26
3.2.4.1	Virtuální asistenti	27
3.2.4.2	Vývoj virtuálních asistentů.....	28
3.2.4.3	Princip fungování virtuálních asistentů.....	30
3.2.4.4	Google Home	32
3.2.5	Bezpečnost chytré domácnosti.....	33
3.3	Vlastní řešení v chytré domácnosti	34
3.4	Jednodeskový počítač.....	34
3.4.1	Raspberry Pi.....	35
3.4.1.1	Vývoj Raspberry Pi.....	36
3.4.1.2	Raspberry Pi Foundation	37
3.4.1.3	Architektura Rapsberry Pi.....	38
3.4.2	Komponenty pro Raspberry Pi.....	42
3.4.2.1	Teplotní čidlo DS18B20.....	42
3.4.2.2	Čidlo vlhkosti DHT11	43
3.4.2.3	Pohybové čidlo HC-SR501	43
3.5	Windows Subsystem for Linux	44
3.6	Python	45

4	Vlastní práce	47
4.1	Výběr řešení chytrého zrcadla.....	47
4.1.1	Definice variant.....	47
4.1.1.1	Vybrané varianty	52
4.1.2	Definice hodnotících kritérií	53
4.1.2.1	Určení vah kritérií	55
4.1.2.2	Vícekritériální analýza	55
4.2	Technický popis řešení varianty A.....	59
4.3	Technický popis řešení varianty B.....	60
4.4	Hardwarová realizace.....	64
4.5	Softwarová realizace	65
4.5.1	Instalace a konfigurace Raspbian OS.....	65
4.5.2	Vývojové prostředí.....	68
4.5.3	zrcadlo.py.....	69
4.5.3.1	Konfigurační soubor config.ini	70
4.5.3.2	Inicializace a načtení modulů	71
4.5.3.3	Hodiny	72
4.5.3.4	Zprávy	74
4.5.3.5	Počasí.....	76
4.5.3.6	Kalendář	79
4.5.3.7	Celoobrazkový režim	81
4.6	Finální podoba.....	82
5	Závěr	83
6	Seznam použitých zdrojů.....	85

Seznam obrázků

Obrázek 1 - Základní princip fungování Internetu věcí (Carritech, 2018)	14
Obrázek 2 - Počet připojených IoT zařízení v letech (Statista, 2018)	16
Obrázek 3 - Vzorový koncept chytré domácnosti (Antik Technology, 2018).....	17
Obrázek 4 - Příklad ovládání chytré domácnosti (Fearing's, 2018).....	27
Obrázek 5 - "Hype" křivky pro rok 2017 a 2018 (Gartner Inc., 2018).....	28
Obrázek 6 - Přesnost odpovědí virtuálních asistentů (Dunn, 2017)	30
Obrázek 7 - Reproduktory Google Home (Gebhart, 2018)	32
Obrázek 8 - Deska Raspberry Pi 3 B (Wikipedia.org, 2018).....	38
Obrázek 9 - Rozložení GPIO pinů Raspberry Pi 3 B (Raspberry Pi Spy, 2012).....	39
Obrázek 10 - Dvě verze čidla DS18B20 (GM Electronic, 2019)	42
Obrázek 11 - Čidlo DHT 11 (GM Electronic, 2019).....	43
Obrázek 12 - Pohybové čidlo HC-SR501 (GM Electronic, 2019)	44
Obrázek 13 – Srovnání jazyka Python s jinými (Robinson, 2017).....	46
Obrázek 14 - Dotazník: Princip fungování (Vlastní zpracování)	49
Obrázek 15 - Dotazník: Interakce s uživatelem a ovládání (Vlastní zpracování).....	50
Obrázek 16 - Dotazník: Umístění a způsob pořízení (Vlastní zpracování)	51
Obrázek 17 - Myšlenková mapa konceptů (Vlastní zpracování).....	52
Obrázek 18 - Návrh konstrukce pro variantu A (Vlastní zpracování)	59
Obrázek 19 - Princip fungování polopropustného skla (How it works, 2015)	61
Obrázek 20 - Návrh konstrukce pro variantu B (Vlastní zpracování)	62
Obrázek 21 - Architektura varianty B (Vlastní zpracování).....	63
Obrázek 22 - Návrh a reálné zapojení čidel v nepájivém poli (Vlastní zpracování)	65
Obrázek 23 - Instalace Raspbian OS (Vlastní zpracování).....	65
Obrázek 24 - Úvodní obrazovka Raspbian OS (Vlastní zpracování)	66
Obrázek 25 - Konfigurace Raspberry Pi (Vlastní zpracování)	67
Obrázek 26 - Zapnutí "Windows Subsystem for Linux" (Vlastní zpracování)	68
Obrázek 27 - Využití programu Xming (Vlastní zpracování)	69
Obrázek 28 - Konfigurační soubor.....	70
Obrázek 29 - Zdrojový kód: Úvodní nastavení programu (Vlastní zpracování)	71
Obrázek 30 - Stažení lokalizačního balíčku (Vlastní zpracování).....	72
Obrázek 31 - Zdrojový kód: Třída "Hodiny" (Vlastní zpracování).....	73
Obrázek 32 - Zdrojový kód: Funkce "aktualizaceCasu" (Vlastní zpracování).....	73
Obrázek 33 - Zdrojový kód: Funkce "aktualizaceSvatku" (Vlastní zpracování).....	74
Obrázek 34 - Zdrojový kód: Třída "Zpravy" (Vlastní zpracování).....	75
Obrázek 35 - Zdrojový kód: Třída "TitulkyZprav" (Vlastní zpracování).....	75
Obrázek 36 - Zdrojový kód: Funkce "aktualizaceZprav" (Vlastní zpracování)	76
Obrázek 37 - Zdrojový kód: Třída "Pocasi" (Vlastní zpracování).....	77
Obrázek 38 - Zdrojový kód: Funkce "aktualizacePocasi" (Vlastní zpracování).....	78
Obrázek 39 - Zdrojový kód: doplňkové funkce pro počasí (Vlastní zpracování).....	79
Obrázek 40 - Zdrojový kód: Třída "Kalendar" (Vlastní zpracování)	80
Obrázek 41 - Zdrojový kód: Třída "FullscreenOkno" (Vlastní zpracování).....	81
Obrázek 42 - Finální podoba varianty B (Vlastní zpracování)	82

Seznam tabulek

Tabulka 1 - Hodnocení kritérií H1-10 (Vlastní zpracování).....	55
Tabulka 2 - Hodnocení kritérií H11-20 (Vlastní zpracování).....	55
Tabulka 3 - Vícekriteriální hodnocení variant (Vlastní zpracování)	56
Tabulka 4 - Cena varianty A (Vlastní zpracování)	60

Tabulka 5 - Cena varianty B (Vlastní zpracování)	63
--	----

1 Úvod

S rozšířením a zdokonalením technologie internetu věcí roste zároveň popularita i jednoho specifického odvětví této oblasti, konkrétně chytrých domácností. Mnoho společností se začalo specializovat právě na realizaci služeb, které lidem ušetří každodenní starosti, zpříjemní život, zvýší bezpečnost v objektech a které v neposlední řadě přispějí ke snížení spotřeby energií. Do tohoto odvětví se mimo jiné pustil již i Švédský nábytkový gigant IKEA, což jen dokazuje, jaký je o toto odvětví zájem.

I přesto, že poslední dobou cena zařízení pro chytré domácnosti klesá, stále se nejedná o záležitost, kterou si může dovolit každý. Přitom cena samotných hardwarových prvků, ze kterých se komerční produkty skládají, je relativně nízká. K prodávaným sensorům, mikrokontrolerům a dalším prvkům je k dispozici velké množství dokumentace a návodů, a tak si každý šikovný kutil či technologický nadšenec může svojí chytrou domácnost vlastními silami. Náklady se pak oproti existujícím produktům stanou minimální. Možnost rozšiřovat zařízení či připojovat jiná a vytvářet tak celý funkční ekosystém je omezena pouze fantazií tvůrce systému, nikoliv nabídkou jedné řady komerčních produktů.

Cílem této práce je ukázat, jak může postup při realizaci takového zařízení vypadat. Konkrétně se zabývá chytrým zrcadlem, která jsou na trhu zatím téměř neznámým produktem. Je tedy nejprve třeba zjistit, jakým způsobem by zrcadlo mělo být „chytré“. Odpověď na tuto otázku je nejvhodnější získat přímo od potencionálních uživatelů a na základě těchto podkladů si definovat, jak by finální řešení mohlo vypadat. Až poté je možné přistoupit k samotné realizaci, která již vyžaduje technické znalosti jak v oblasti elektroniky, tak programování.

Závěrem práce je představení finálního funkčního řešení, které má potenciál fungovat nejen v takovém stavu v jakém bylo zkonstruováno, ale především také nabídnout možnosti následného rozšíření o další funkční prvky či jiná zařízení. To vše také s přihlédnutím na ekonomickou stránku věci.

2 Cíl práce a metodika

2.1 Cíl práce

Diplomová práce je tematicky zaměřena na realizaci vlastního chytrého zrcadla v domácnosti. Hlavním cílem práce je analýza, definice, komparace a vyhodnocení možných variant softwarového a hardwarového řešení chytrých zrcadel na základě objektivních kritérií s následným návrhem implementace vybraných variant.

Dílčími cíli práce je vypracování přehledu několika možných řešení pro konstrukci chytrého zrcadla a vypracování přehledu vývoje, využívaných technologií a blízké budoucnosti technologie internetu věcí se zaměřením na konkrétní oblast chytrých domácností.

2.2 Metodika

Úvodní teoretická část práce bude založena na studiu odborné literatury, na základě, které bude představena technologie internetu věcí, pod kterou spadají i chytré domácnosti. Pro ty bude definována samotná myšlenka a princip fungování, historický vývoj i s pohledem do blízké budoucnosti. Následně budou představeny i vybrané softwarové a hardwarové technologie využívané při konstrukci vlastních řešení pro chytré domácnosti. Ty budou následně využity i v druhé části práce.

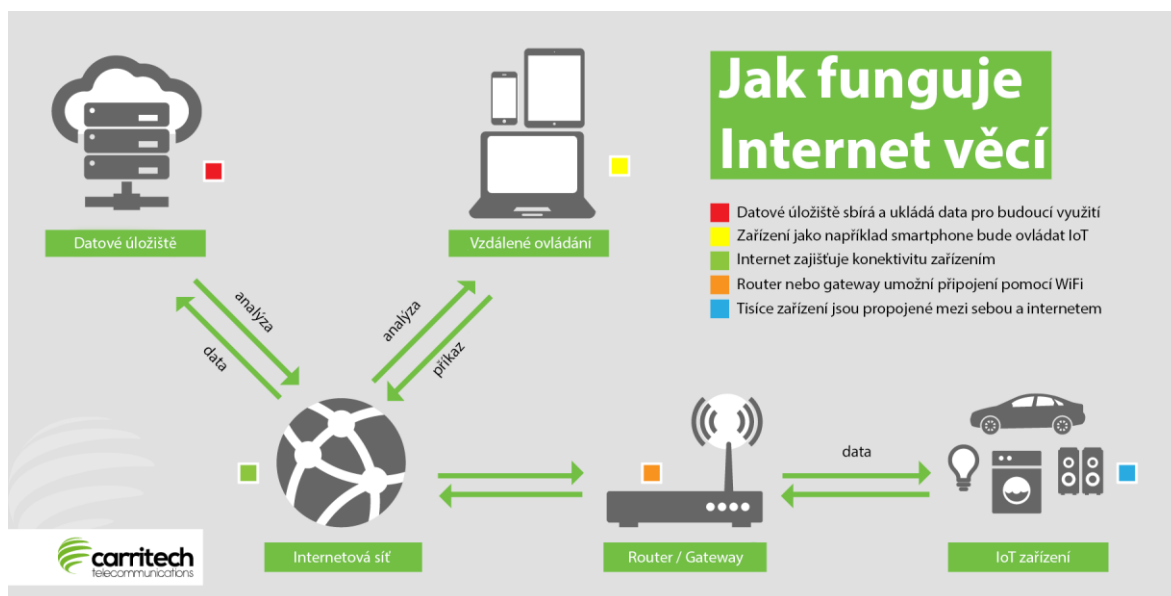
Praktická část bude začínat dotazníkovým šetřením, kde budou zjišťovány preference uživatelů, pro koncept chytrého zrcadla. Na základě získaných poznatků bude vydefinováno několik variant, které budou následně objektivně hodnoceny využitím vícekritériální analýzy, přičemž pro stanovení vah jednotlivých kritérií bude využita Metfesselova alokace sta bodů, kdy respondenti v rámci dotazníku přiřadí body jednotlivým kritériím tak, jak jsou pro ně při rozhodování důležitá. Dle výsledků analýzy budou pro vybrané varianty vypracovány návrhy realizace, přičemž bude využit z hardwarových prvků zejména mikropočítač Raspberry Pi a pro softwarovou část programovací jazyk Python.

3 Teoretická východiska

3.1 Internet věcí (IoT)

Internet věcí je v dnešní době stále ještě možné považovat za relativně nový fenomén, který se teprve rozrůstá a dostává do povědomí běžných uživatelů. Jeho vznik, vývoj a postupné šíření lze považovat za jednu z dalších internetových revolucí (nebo spíše evolucí) – hned po vzniku samotného internetu (World Wide Web) v devadesátých letech 20. stolní a následně rozmachu mobilního internetu začátkem nového milénia. (Sundmaeker, Guillemin, Friess, & Woelfflé, 2010)

Jedná se o termín, jehož definice není dosud plně ustálena. Vznik samotného anglického termínu „Internet of Things“ (z toho vychází i často využívaná zkratka IoT nebo český překlad „internet věcí“) je datován do roku 1999, kdy ho údajně poprvé použil Kevin Ashton. Podle jeho vlastních slov tento výraz poprvé použil v té době ještě jako zaměstnanec firmy Procter & Gamble jako název jedné ze svých prezentací. V článku z roku 2009 uvedl: „Mohu se mýlit, ale jsem si poměrně jistý, že fráze „Internet of things“ započala svoji existenci jako název jedné z prezentací, které jsem vytvořil v Procter & Gamble v roce 1999.“ (Ashton, 2009)



Obrázek 1 - Základní princip fungování Internetu věcí (Carritech, 2018)

Jak lze vidět z obrázku výše, obecně lze říci, že se jedná o síť zařízení, která mezi sebou dokáží komunikovat bez zásahu člověka. Jedná se především o různé senzory, chytré spotřebiče, chytrá fyzická zařízení či například automobily vybavené chytrou elektronikou. V současnosti zatím neexistuje jednotná oficiální definice, která by přesně vyjasňovala pojem

internet věcí. Nejbližší k této definici je ale pravděpodobně popis, který svojí vzájemnou spoluprací vytvořily dvě mezinárodní iniciativy zabývající se danou problematikou: IERC (European Research Cluster on the Internet of Things) a ITU (Internet of Things Global Standards Initiative). Definice, kterou společně představili zní: *„Jedná se o dynamickou globální síťovou infrastrukturu s možností vlastního nastavení, která je založena na standardních a interoperabilních komunikačních protokolech, kde fyzické i virtuální „věci“ mají své identity, fyzické atributy a virtuální osobnosti, využívají inteligentních rozhraní a jsou bezproblémově integrovány do informační sítě“* (European Research Cluster on the Internet of Things, 2018)

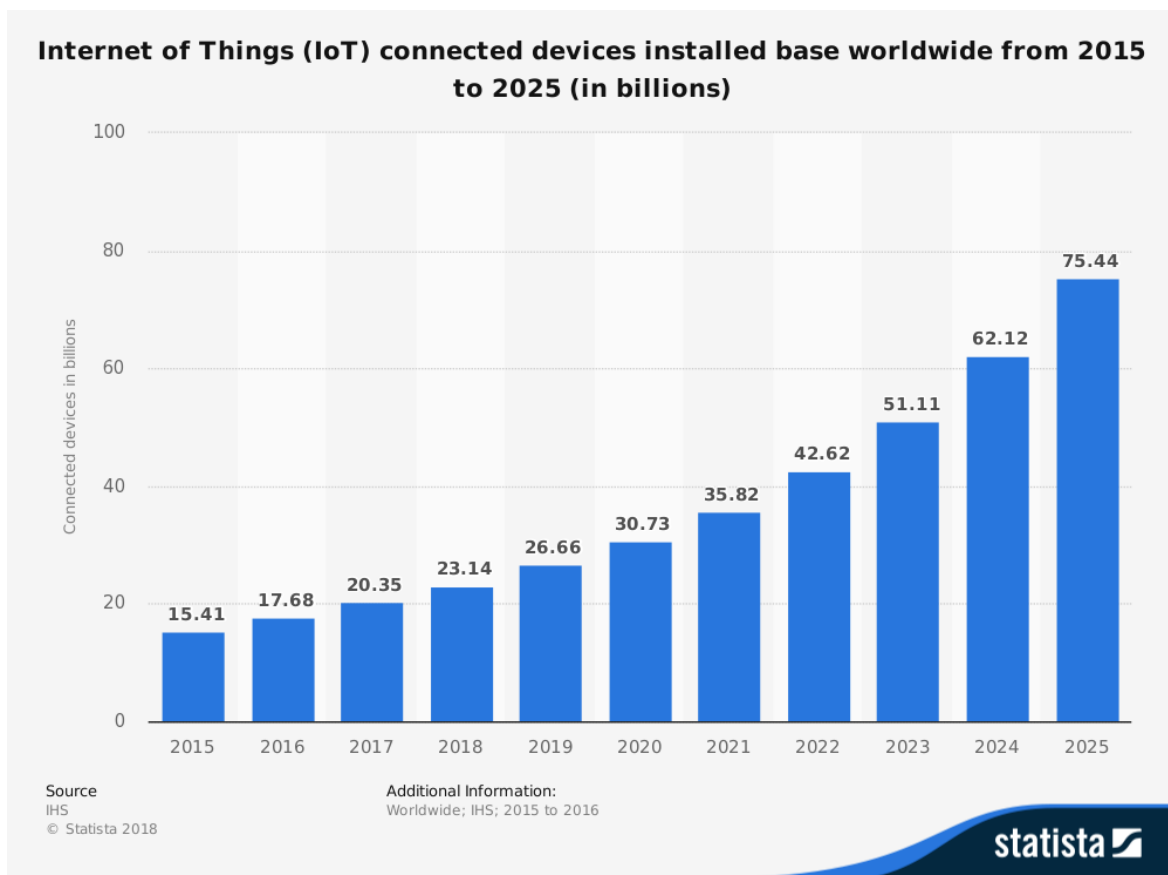
O definování tohoto termínu se pokusily i známé společnosti IT průmyslu. Jak je vidět z následujících definic, každá z těchto společností definici pojala odlišným způsobem, které především vychází z odlišných oblastí, ve kterých se tyto společnosti pohybují.

Společnost SAP (zabývá se vývojem ERP systémů) definuje IoT jako: *„Internet věcí je síť fyzických objektů – vozů, strojů, domácích zařízení a mnoha dalších – které využívají senzory a API k propojení, komunikaci a vzájemné výměně za využitím internetu.“* (SAP, 2018)

Společnost CISCO na IoT nahlíží jako na: *„Spojení sítí senzorů, připojených k objektům a komunikačním zařízením, poskytující data, která mohou být analyzována a využívána pro zahájení automatizovaných akcí“.* (Cisco, 2012)

Americká společnost pro výzkum a poradenství v oblasti IS/ICT, Gartner Inc. Prezentuje IoT jako: *„Síť fyzických objektů, které obsahují zabudované technologie umožňující komunikovat, vnímat nebo interagovat s jejich vnitřními stavy nebo s vnějším prostředím.“* (Gartner Inc., 2018)

V roce 2011 již zmíněná společnost Cisco ve svém průzkumu předpověděla, že v roce 2020 očekává přibližně 50 miliard připojených IoT zařízení. (Evans, 2011)



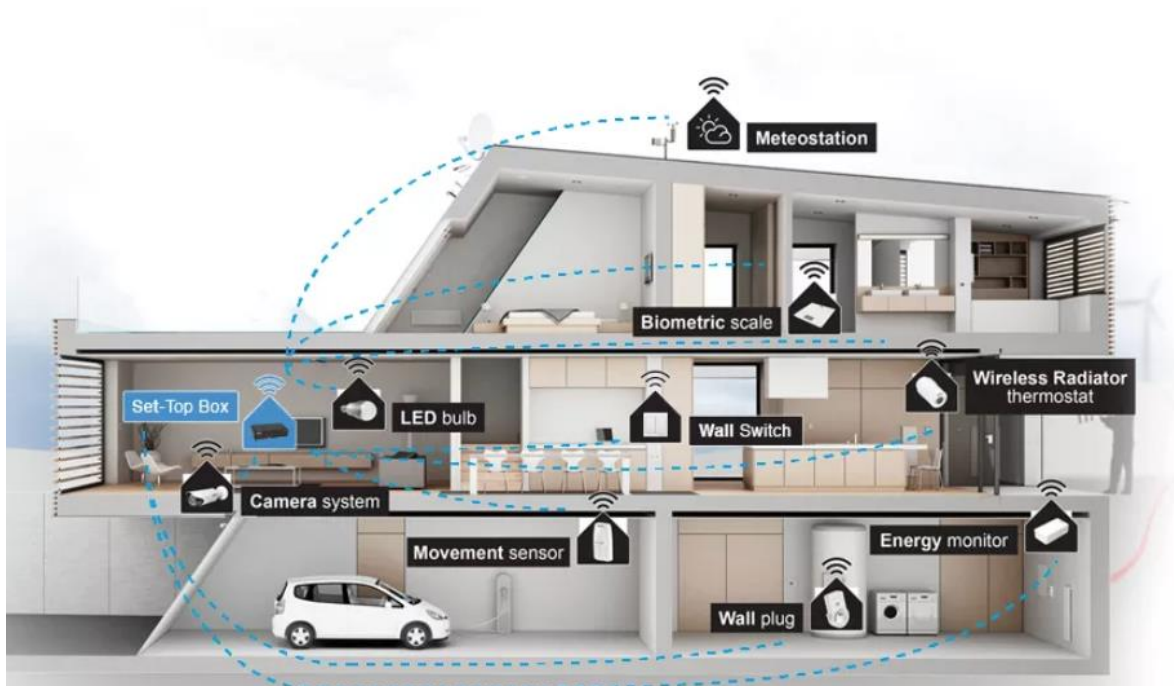
Obrázek 2 - Počet připojených IoT zařízení v letech (Statista, 2018)

Pokud se podíváme na aktuálnější data z portálu Statista, je vidět, že vzestup není tak rapidní, jak Cisco předpovídalo, ale i tak je meziroční nárůst zhruba 15 %. Dle aktuálních dat je tedy možné překročení hranice 50 miliard zařízení očekávat zhruba v roce 2023. Pro zajímavost, počet IoT zařízení převýšil počet žijících lidí na planetě již mezi roky 2011 a 2012. (Statista, 2018)

Součástí těchto předpokladů je zároveň i nárůst hodnoty trhu s IoT prvky, očekává se, že by jeho hodnota mohla zhruba do roku 2023 narůst až na 1,7 bilionů amerických dolarů. Pro představu, orientační hodnota trhu s chytrými telefony byla v roce 2017 necelých 500 miliard amerických dolarů. Je tedy na první pohled zřejmé, že se od internetu věcí mnoho očekává. (IoT Online store, 2018)

3.2 Chytrá domácnost

Jednou z oblastí, které pod IoT spadají jsou zařízení pro chytrou domácnost. S tímto pojmem je možné se někdy setkat pod různými názvy, jako například „inteligentní domácnost“, „inteligentní dům“ nebo také „digitální domácnost“ principiálně se ale vždy jedná v podstatě o totéž. Stručně řečeno se jedná o dům či byt, který je vybaven zařízeními a spotřebiči, které spolu dokáží za pomoci moderních technologií vzájemně komunikovat, interagovat a umožňují uživateli jejich automatizaci a vzdálené ovládní, čímž tato zařízení naopak ulehčují a zpříjemňují každodenní činnosti obyvatel. Definice pojmu může být předkládána různými způsoby.



Obrázek 3 - Vzorový koncept chytré domácnosti (Antik Technology, 2018)

Jak lze vidět z konceptu na obrázku, chytrá domácnost, je v základu dům, ve kterém jsou všechny (nebo alespoň většina) elektronické systémy (například osvětlení, vytápění a klimatizace, zabezpečení, zábava) propojené a mohou společně komunikovat a spolupracovat a uživatel má možnost je jednoduše ovládat nejlépe jedním centrálním ovladačem. Chytré domy bývají většinou napojeny na internet, takže je možnost svoji domácnost spravovat i když se zrovna nacházíme tisíce kilometrů daleko. (Clauser, Kelly, Roth, & Sesnovich, 2015)

Profesor Richard Harper z Anglické University of Lancaster definuje chytré domácnosti ve své knize *Inside the Smart Home* takto: „Chytrý dům může být definován jako obydlí vybavené výpočetními a informačními technologiemi, které předvídá a reaguje na potřeby

obyvatelů, snaží se zlepšovat jejich komfort, pohodlí, bezpečnost a zábavu pomocí správy technologií v domě a propojení s vnějším světem.“ (Harper, 2003)

U českých autorů můžeme najít definici v knize *Inteligentní dům* od Miroslava Kareše, který uvádí jeho pohled na inteligentní dům takto: „*Inteligentní dům v nejširším možném smyslu slova je budova vybavená počítačovou a komunikační technikou, která předvídá a reaguje na potřeby obyvatel s cílem zvýšit jejich komfort, pohodlí, snížit spotřebu energií, poskytnout jim bezpečí a zábavu pomocí řízení všech technologií v domě a jejich interakcí s vnějším světem.*“ (Valeš, 2006)

ICT expert Michal Hozák ze společnosti T-Mobile řekl o možnostech chytrých domácností: „*Stále více domácností i obcí nasazuje nějaká chytrá řešení. Pokud se tyto dva zdánlivě samostatné světy dokážou spojit a sdílet si vzájemně informace, mohou z toho značně profitovat*“ (Málek & Kovářová, 2018) z čehož vyplývá, že chytré technologie není nutné limitovat pouze na domy či byty, ale postupem času bude možné vytvářet celé chytré vesnice, či města.

Technologie chytré domácnosti je jedno z nejvíce rozvíjejících technologických odvětví dnešních dnů. Trvalo ale poměrně dlouho, než se o něj běžní uživatelé začali opravdu aktivně zajímat. Snaha o automatizaci domácnosti je tu již téměř 20 let, ale pro většinu lidí se tento trend stal zajímavý až během několika posledních roků. Pro to je několik vysvětlení, ale jako jedno z hlavních může být i rozvoj finančně nenáročných „udělej si sám“ řešení, které udělaly chytrá domácí řešení mnohem dostupnější pro každého.

3.2.1 Historický vývoj a budoucnost chytrých domácností

Chytré domy, stejně jako většina inovací, měly svůj teoretický základ již dlouho předtím, než se staly realitou. Po celá desetiletí si lidé hráli s myšlenkou automatizace domácnosti. Tato vize neunikla ani pozornosti mnoha literárních autorů, jakým byl například Ray Bradbury, který námět samostatně fungujících domů zpracoval v jedné ze svých povídek zhruba v polovině 20. století. (Hendricks, 2014)

20. století představilo dramatickou revoluci i evoluci domácích technologií, které kulminovaly koncem století právě příchodem inteligentních domů. Na začátku 20. století by mohla být většina domácích technologií snadno rozpoznána a používána i lidmi o sto let dříve. Do konce 20. století se však tyto technologie od základů změnily. První velký impuls pro změnu přineslo zavedení elektřiny do domácností v prvním čtvrtletí tohoto století. To poskytlo nový zdroj čistého a pohodlného pohonu pro spotřebiče a podnítilo představení a zavedení nových zařízení. Druhým velkým podnětem bylo rozšíření informačních technologií v poslední čtvrtině

století. To otevřelo nové možnosti pro výměnu informací mezi lidmi, zařízeními, systémy a sítí v domácnosti i mimo ni a mnoho dalších ještě čeká na své objevení. (Harper, 2003)

- **1901-1960**

- Během počátku století se rozvíjely střední třídy obyvatelstva a brzy začaly pociťovat nedostatek domácího služebnictva. V návaznosti na tento nedostatek pracovní síly byly poprvé představeny různé stroje, jako například první vysavače poháněné motorem v roce 1901 a během následujících desetiletí pokračovaly s vynálezem lednice, sušičky prádla, pračky, elektrické žehličky a dalších. Reklama byla cílena tak, že s pomocí těchto technologií je jedna osoba schopna zvládnout všechny domácí práce, a ještě bude mít dostatek volného času pro sebe.

- **1966-1967**

- ECHO IV bylo první inteligentní zařízení pro chytrý dům, které dokázalo řídit teplotu a zapnout a vypnout spotřebiče. Kuchyňský počítač, který se vyvinul o rok později, dokázal ukládat recepty. Obě zařízení tak zvládla předběhnout svou dobu. Velké množství dalších, čas šetřících zařízení se stalo běžnou součástí domácností jako například varné konvice, toasty, holicí strojky nebo například centrální vytápění a termostaty

- **1980-1990**

- Počátkem osmdesátých let měly například v Anglii již tři čtvrtiny domácností barevnou televizi a koncem osmdesátých let měla polovina i videorekordéry. Nové technologie jako například bezdrátové telefony začaly být dostupné a další technologie, především přinášející zábavu do domácností, jako například kabelová televize začaly prorážet na trhu víc a víc. Konečně příchod internetu začal lidem umožňovat lidem zvládat z pohodlí domova mnoho činností, u kterých doposud na takový servis zvyklí nebyli.

- **1990-2000**

- Inteligentní domy a inteligentní domácnosti začaly na počátku devadesátých let vzrůstat a stávat se finančně dostupnějšími pro běžné uživatele. Objevovaly se nejrůznější nové technologie, domácí sítě a další produkty.

- **Současnost**

- V dnešní době je hlavní myšlenkou inteligentních domů především zjednodušení každodenního života a zajištění bezpečnosti. Z průzkumu společnosti Berg Insight, který byl proveden v roce 2017, se přibližně 10,9 milionů domácností v Evropě dá považovat za chytré. Během příštích pěti let se očekává každoroční nárůst o 57 % až do roku 2021, kdy analytici předpovídají na stejném území celkem 80,6 milionů chytrých domácností. Pro porovnání, na konci roku 2014 jich bylo pouze 3,3 milionů, od té doby se jejich počet více jak ztrojnásobil. Větší zastoupení ale mají chytré systémy v Severní Americe, kde je podstatně znatelnější náskok, co se zralosti i penetrace trhu chytrých domácností týče. Z celkového počtu severoamerických domácností má již 16,7 % instalováno chytré řešení, ať už dílčí nebo komplexní. Jedním z dalších teritorií, kterému jsou přisuzovány velké příležitosti jsou země východní Asie, vysoký nárůst je očekáván v zemích jako je Čína, Jižní Korea nebo Japonsko. Na základě údajů z portálu Statista.com je Čína dokonce druhou zemí s nejvyšším obratem v oblasti chytrých domácností ihned po vedoucím USA, na třetím místě je poté Německo. (Asociace chytrého bydlení, 2017)

- **Budoucnost**

- Mnoho designéru v dnešní době přichází s různými vylepšeními, která rozhodně mají svůj potenciál. Nejdříve je ale potřeba vyřešit několik základních a zásadních problémů, pokud mají mít chytré domácnosti v budoucnu úspěch. Pravděpodobně nejmarkantnější problém je potřeba vytvořit standardy napříč celým odvětvím, kterými se budou moci výrobci jednotně řídit. Ty by měly umožnit komunikaci a sdílení dat napříč přístroji různých výrobců. Aktuálně většina technologií komunikuje skrze proprietární protokoly, a to zamezuje bezproblémové komunikaci zařízení v chytrém domě. (Harper, 2003)
- Nedílnou součástí, týkající se tohoto odvětví, je bezpečnost. Díky těmto technologiím sdílíme naši fyzickou stopu, aniž by si to někteří uživatelé reálně uvědomovali. Není to otázkou, zda, ale kdy budou tyto systémy ohroženy a důsledky by mohly být mnohem závažnější, než jen vykradený byt. Řešení bezpečnosti a soukromí se stane základním problémem, který toto odvětví vytvoří. (Forbes Technology Council, 2018)
- Důležitou roli v chytrých domácnostech bude bezpochyby hrát umělá inteligence. Technologie se stane mnohem efektivnějším a bude možné ovládat

téměř vše, co se v běžných domácnostech nachází. S rychlostí a úspěchem, jak se umělá inteligence vyvíjí, nakonec nebude třeba nic ručně ovládat, protože se tato zařízení automaticky přizpůsobí našim preferencím. (Forbes Technology Council, 2018)

3.2.2 Základní prvky chytré domácnosti

I přes to, že možnosti systémů chytrých domácností jsou rozmanité a v dnešní době již téměř neomezené, existuje několik základních kategorií, ze kterých bude celý systém složen. Těmi kategoriemi jsou: energie, zabezpečení, atmosféra, pohodlí a zábava. Tyto kategorie představují různá odvětví domácích zařízení, která jsou schopna těžit s technologií chytrých domácností. Aby bylo možné vytvořit perfektní inteligentní dům, je potřeba do něj provázaně zakomponovat prvky ze všech zmíněných oblastí.

3.2.2.1 Energie

Téměř každá domácnost spotřebovává elektřinu, plyn, vodu a další energie na denní bázi. Majitelé o tom mají velice dobré povědomí především díky účtům, které za využívání výše zmíněných musí platit. Snahou každého běžného člověka je spotřebu energií co nejvíce snížit, ale zároveň výrazně neomezovat své pohodlí a potřeby.

Inteligentní domy jsou perfektní příležitostí, jak toho dosáhnout díky pokročilým technologiím, které jsou velice energeticky efektivní. Sledují chování uživatelů a jsou schopny tomu přizpůsobit veškerý chod domácnosti, například na základě času odchodů a příchodu obyvatel domů upravují vytápění nebo automatické osvětlení.

Prvky, které spadají do této kategorie jsou především termostaty, inteligentní osvětlení, inteligentní elektrické zásuvky či měřiče výkonu a spotřeby. (Carlson, 2016)

3.2.2.2 Zabezpečení

Domov je centrem našich životů. Ať se jedná o farmu na venkově či podkrovní byt ve městě, vždy se jedná o místo, kde bychom se měli především cítit bezpečně.

Chytrý dům by toto měl být schopen zajistit po dobu 24 hodin, 7 dní v týdnu, tak aby jeho obyvatel nemusel pomyslet na žádné nebezpečí. Díky detektorům pohybu či kouře, alarmům a bezpečnostním kamerám jsou schopné tyto domácnosti zajistit bezpečí všech

obyvatel a v případě ohrožení, například požárem, mohou automaticky kontaktovat hasičský záchranný sbor nebo jiné příslušné autority.

Jak již bylo zmíněno, prvky spadající do této kategorie představují bezpečnostní zámky například na otisk prstu, automatické alarmy, reagující na pohyb, bezpečnostní kamery i s nočním viděním, senzor rozbití oken, detektory pohybu, kouře či jiných nebezpečných plynů nebo i dětské chůvičky. (Carlson, 2016)

3.2.2.3 Pohodlí

Zde se jedná konečně o část, která dělá z obyčejného domu dům budoucnosti. Energetická efektivita a bezpečnost jsou bezpochyby velice důležitými prvky inteligentního bydlení, ale existuje nepřeberné množství každodenních aktivit, které mohou být zefektivněny či dokonce automatizovány.

Díky inteligentním domům by se mělo stát provádění těchto činností pohodlnější a příjemnější, měly by především ušetřit čas a energii, kterou poté může uživatel vynaložit efektivněji.

Každý má jiný názor na to, co by mělo být inteligentní a automatizované, a ještě silnější názor na to, co by naopak chytré být nemělo. Dostupné produkty jsou stejně rozmanité jako tyto názory, a proto si tak může každý vybrat od chytré lednice, která uživatele vždy včas upozorní například pomocí notifikace na mobilu, že již dochází mléko, až po inteligentní toaletu, která je schopna analyzovat moč a upozornit uživatele v případě zdravotních potíží.

Jako další mohou být do této kategorie zařazeny především různé spotřebiče, od kávovarů, přes pračky, sušičky až po automatické vysavače, ale například i multifunkční sedačky či sprchové kouty. (Carlson, 2016)

3.2.2.4 Atmosféra

Inteligentní domy by měly být schopné poskytnout příjemnou atmosféru, přizpůsobenou na míru preferencím uživatele. Měly by být schopny poznat, kdo se v domě zrovna nachází, co mají rádi a tomu náležitě přizpůsobit atmosféru, která v domě panuje.

Atmosféra udává to, jak se v domácnosti cítíme, a proto je důležité, aby vždy byla dle představ uživatele a přizpůsobovala se situaci. Domácnost by měla být schopna se přizpůsobit náladě obyvatel a také podle toho reagovat, například podle hudby, která se zrovna poslouchá může být příslušně upravována barva a intenzita barev. Má každý z obyvatel rád jinou teplotu a

vlhkost vzduchu ve svém pokoji při spaní? I tomu by měl být inteligentní dům schopen se přizpůsobit.

Mezi prostředky, kterými je možné dosáhnout té správné atmosféry se řadí například hudební systémy, světla s možností nastavení intenzity a barvy světla, termostaty či zvlhčovače vzduchu, aroma difuzéry a další. (Carlson, 2016)

3.2.2.5 Zábava

Je důležité, aby inteligentní dům splňoval všechny výše zmíněné faktory, ale byla by škoda, mít energeticky úsporný, automatizovaný a zabezpečený dům, ve kterém se uživatel bude nudit. Zábava je důležitou částí života každého člověka a každý se bude rád bavit i doma. Jakým zábavním systémem dům vybavíme poté závisí pouze na preferencích uživatele. Často se může jednat o systém domácího kina s kvalitním obrazovým i zvukovým systémem nebo pokud se například jedná o milovníky domácích mazlíčků, je možné pořídit automatizované dávkovače potravy a čímž odpadá jedna ze starostí a o zvířata je postaráno i když je majitel například v práci. Tyto prvky mohou být propojeny s částmi, které se starají o navození vhodné atmosféry a společně tak vedou k tomu, že se obyvatelé budou v chytré domácnosti cítit vždy skvěle.

Do této kategorie lze zařadit množství zařízení od různých herních zařízení a příslušenství k nim až například po cvičební trenažery, které nám vždy řeknou, jak jsme si při našem výkonu vedli. (Carlson, 2016)

3.2.3 Komunikační protokoly

V bezpečném propojeném světě musí být chytré produkty schopné komunikovat, vyměňovat data a informace s mnoha jinými zařízeními ať už pomocí drátového propojení či bezdrátově. Aby to bylo možné, musí komunikovat stejnou řečí neboli pomocí komunikačních protokolů.

Protokoly definují, jak signály odesílané z jednoho zařízení na jiné, tak, aby došlo ke spuštění požadované akce, například zapnutí či vypnutí světel. Porozumění rozdílům mezi protokoly je důležité, jelikož je jedním ze základních pilířů při výběru zařízení pro automatizovaný chytrý ekosystém, jelikož každý má nějaká svá pro i proti. Existuje i možnost, kdy systém chytré domácnosti bude pracovat s různými protokoly, jelikož ne vždy je možné například pořídit veškerá vyžadovaná zařízení od jednoho výrobce, tak aby bylo vše řešeno pomocí jednoho komunikačního protokolu. Je ale důležité myslet na to, že většina těchto protokolů není mezi sebou přímo kompatibilní a je tedy potřeba využívat prostředníka, který

bude sloužit jako překladatel jednotlivých protokolů. Tím se může stát například některý z virtuálních asistentů, jelikož ti většinou v základu podporují větší množství protokolů nebo existují možnosti, jak jejich podporu rozšířit. Příkladem může být systém žárovek Philips Hue, který je schopen komunikovat s asistentem Amazon Echo pomocí speciálního doplňku Philips Bridge.

Komunikační protokoly jsou vyvíjeny organizacemi, konsorciemi či asociacemi, jejichž členi probírají a odsouhlasují detaily vývoje, popisují je a často i vytváří hardwarové prototypy sloužící pro demonstraci nově vytvořených protokolů.

Tyto skupiny mohou být otevřené, tedy každý může daný protokol využívat. Do této skupiny spadají například KNX a Z-wave. Druhou možností je, že protokoly jsou vyvíjeny soukromě, a tedy pouze výrobce může daný protokol využívat, případně ho poskytne i jiným výrobcům, ale až poté, co zakoupí od tvůrce licenci. Příkladem může být protokol Zigbee. (Dekra, 2017)

Mezi nejpoužívanější protokoly patří v dnešní době:

Infračervené záření

Jeden z nejjednodušších, ale zároveň nejspolehlivějších protokolů, nabízí jednosměrnou komunikaci. Nejčastějším využitím jsou dálkové ovladače, například pro TV. (Dekra, 2017)

Ethernet

Rychlá a spolehlivá drátová komunikace, která je v základu schopna pracovat na vzdálenost více než 100 metrů a má malou náchylnost k elektromagnetickému rušení.

WiFi

Je využíván především pro bezdrátové připojení do sítě Internet i v rámci rozsáhlejších lokalit. Je zde typická relativně vysoká přenosová rychlost a dosah až 100 metrů, to ale může být značně ovlivněno například zarušením oblasti či fyzickými překážkami. (Labioud, et al., 2007)

Bluetooth

Bezdrátový protokol, pracující na krátkou vzdálenost, který je často využíván v telefonech, sluchátkách či reproduktorech. Jeho adaptivní systém umožňuje detekovat stávající signály podobně jako WiFi a je schopen vybírat vhodný kanál tak, aby minimalizoval rušení. Protokol je energetickou náročností mezi WiFi a ZigBee a mezi jeho hlavní výhody

patří především cena komunikačních prvků a celková jednoduchost systému. (Labiód, et al., 2007)

Thread

Skupina společností zahrnující Nest, Samsung, Qualcomm a Osram vyvinula v červenci 2014 bezdrátový protokol, který byl designován tak, aby zařízení, která ho využívají, byla schopna komunikovat i v situaci, kdy dojde k výpadku WiFi sítě. Do sítě, využívající Thread, může být připojeno více než 250 zařízení a jelikož většina prvků, které s tímto protokolem pracují, jsou napájeny bateriemi, jedná se o energeticky velice úsporné řešení. (EH Contributor, 2016)

Zigbee

Zigbee je opět bezdrátový protokol, pracující v sítích typu mesh. To znamená, že využívá zařízení v síti k přenosu signálu ke vzdálenějším zařízením, čímž se dosah sítě posiluje a zvětšuje. Tento protokol může být využíván u stmívačů, dveřních zámků, termostatů a dalších. (Dekra, 2017). Nižší přenosová rychlost zároveň přináší vyšší odolnost proti rušení, díky čemuž je vhodné Zigbee například i pro využití v průmyslu, jelikož umožňuje připojení více jak 65 000 zařízení. (Labiód, et al., 2007) Největším problémem je i přesto, že se jedná o standardizovaný protokol s označením 802.15.4, občasná nekompatibilita zařízení různých výrobců, které Zigbee využívají. Zároveň k tomu ještě existují různé verze ZigBee, které mezi sebou také nejsou plně kompatibilní. Proto je tento protokol vhodný především v případech, kdy využíváme kompletní systém pouze od jednoho výrobce. (EH Contributor, 2016)

Z-Wave

Jedná se o protokol podobný Zigbee, je to open source protokol, který také pracuje v mesh sítích. Z technického hlediska je hlavní rozdíl v přenosové rychlosti těchto dvou. Z-Wave je přibližně 6krát pomalejší jak Zigbee. Tuto skutečnost ale částečně kompenzuje tím, že spotřebuje mnohem méně energie k pokrytí stejné oblasti ve srovnání se Zigbee. Velikou výhodou oproti jiným technologiím je zároveň i interoperabilita. Všechna zařízení Z-Wave komunikují se všemi ostatními zařízeními Z-Wave bez ohledu na typ, verzi či značku, zároveň i samotný protokol je kompatibilní napříč všemi jeho verzemi, maximálně s určitými omezeními funkčnosti. (EH Contributor, 2016)

KNX

Technologie je založena na decentralizované topologii, což znamená, že systém nepracuje se žádnou centrální řídicí jednotkou, ale každé zařízení, které je připojené do KNX ekosystému, je chytré samo o sobě a není závislé na funkčnosti ostatních prvků v síti. To přináší velkou výhodu, v případě výpadku některé z částí systému, je zbytek schopný bez problémů dále pracovat. Komfort a spolehlivost, která přichází s KNX, jsou zároveň spojena i se zvýšenou bezpečností, protože výrobky certifikované podle KNX musí splňovat normu EN-50941, která zaručuje spolehlivost v různých oblastech, například: elektrická bezpečnost, funkční bezpečnost či spolehlivost a výkon. (Dekra, 2017)

3.2.4 Ovládání a řízení chytré domácnosti

Ovládání chytrých domácnosti lze obecně rozdělit na dva způsoby, decentralizované a centralizované.

Základem prvně zmíněného, decentralizovaného řízení, je fakt, že každý senzor či zařízení v chytré domácnosti má integrovanou svou vlastní řídicí jednotku a komunikace mezi jednotlivými prvky probíhá přes sdílenou sběrnici. Komunikace na sběrnici je řízena předem stanoveným protokolem, kterým je umožněno zařízením dorozumívat se mezi sebou.

Pod pojmem zařízení si lze v tomto případě představit každý prvek připojený ke sdílené sběrnici. Decentralizovaně řízený systém je vhodný zejména pro velké budovy, kde není bezpečné, aby celý systém spoléhal na jeden řídicí bod.

Druhý způsob ovládání možností je řízení centralizované. Domácnost, která je řízena centrálně má jednu jednotku, která zpracovává a vyhodnocuje informace od všech připojených zařízení a také je zároveň ovládá. Toto řešení není vhodné pro velké domácnosti či budovy, kde je velice komplikované, ne-li nemožné zapojit všechny zařízení do jedné řídicí jednotky.

Pro pohodlné ovládání uživateli bývá veškeré elektronické vybavení v chytré domácnosti sloučené a synchronizované do jednoho ovládacího prvku. Uživatel tak může ovládat jednotlivé prvky přímo v místě, kde se dané zařízení nachází, nebo dálkově pomocí vzdáleného přístupu. Například je možné spustit vytápění před příchodem domů.



Obrázek 4 - Příklad ovládání chytré domácnosti (Fearing's, 2018)

Nejvíce používaným zařízením pro toto centralizované ovládání bývá v dnešní době především chytrý mobilní telefon, případně tablet.

Do popředí se ale velice razantně prosazují i virtuální asistenti, kteří umožňují ovládat chytrou domácnost pomocí příkazů v běžné řeči a samotný provoz domácnosti se tak stává pohodlnější a jednodušší. S vlastními asistenty přichází většina velkých firem, které se pohybují v IT odvětví, jako například Apple, Microsoft nebo Amazon a každý se snaží přesvědčit své zákazníky, proč by zrovna jejich zařízení mělo být to nejlepší. (Harper, 2012)

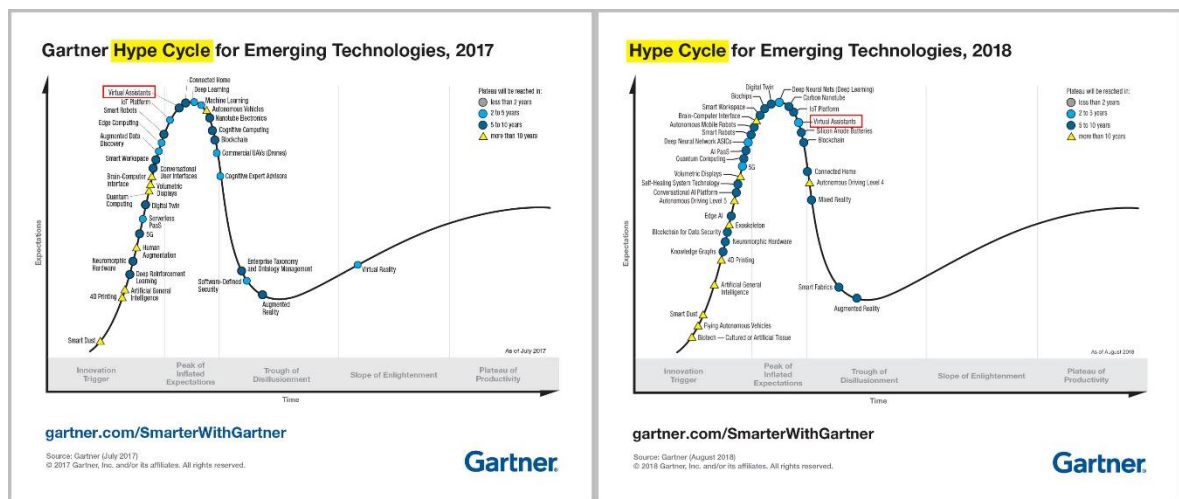
3.2.4.1 Virtuální asistenti

Základem hlasových nebo virtuálních asistentů je software, který je schopný interpretovat lidskou řeč a odpovídat pomocí syntetizovaného hlasu. Někdy se můžeme setkat i s termínem "chatbot", který se používá k odkazu na virtuální asistenty obecně nebo konkrétně k těm, které jsou přístupné v podobě online chatu.

Siri od společnosti Apple, Alexu Amazonu a Cortana od Microsoftu nebo Google Assistant jsou aktuálně nejpoblárnějšími v této oblasti a jsou součástí chytrých telefonů nebo dedikovaných domácích reproduktorů. Uživatel se může asistenta ptát na různé otázky stejně jako by se ptal jiného člověka a asistent mu je schopen odpovědět, případně se zařídit podle uživatelského požadavku. Další z možností, které tito asistenti nabízejí, je ovládání chytré

domácnosti, tento prvek tak může sloužit jako centralizovaný bod, přes který je obyvatel domácnosti schopen řídit jak osvětlení místností, tak například zábavní systém pouhými hlasovými příkazy. (Hoy, 2018)

Od roku 2017 se možnosti využití virtuálních asistentů rychle rozšiřují, vstupují na trh nové produkty a do popředí se dostává silný důraz na hlasové uživatelské rozhraní. Online anketa v květnu 2017 zjistila, že nejvíce používané v USA jsou Apple Siri (34 %), Asistent Google (19 %), Amazon Alexa (6 %) a Microsoft Cortana (4 %). Apple a Google mají rozsáhlé základny uživatelů na smartphonech, společnost Microsoft ovládá především osobní počítače se systémem Windows a Alexa převládá především v oblasti inteligentních reproduktorů. (auto.home.techs, 2018)



Obrázek 5 - "Hype" křivky pro rok 2017 a 2018 (Gartner Inc., 2018)

Jak je patrné na takzvané „hype“ křivce od společnosti Gartner, která každý rok ukazuje, o jaké technologie je mezi lidmi zájem, chytrí asistenti se již od roku 2017 drží na špičce technologií, od kterých mají uživatelé největší očekávání.

3.2.4.2 Vývoj virtuálních asistentů

Tím nejdůležitějším, aby virtuální hlasový asistent mohl vůbec fungovat, je zejména schopnost dobře rozpoznávat řeč, což znamená převést mluvenou řeč uživatele do formy, kterou může následně počítač uvnitř zpracovávat a na základě příkazu dodat uživateli relevantní výsledek nebo splnit určitý požadavek. Průkopníkem v technologiích pro rozpoznávání řeči bylo IBM, když začalo v roce 1961 vyvíjet a následně i představili první rozpoznávač – IBM Shoebox. Ten toho v porovnání s dnešními přístroji mnoho neuměl, rozpoznal pouze 16 slov a číslice. Byl to ale první důležitý krok, který otevřel dveře následnému vývoji této technologie.

A vývoj opravdu přišel a přinesl s sebou razantní pokrok. První nástroj, který byl dostupný i pro širokou veřejnost a mohl si ho tak zakoupit každý, kdo měl zájem, a především dostatek peněz, byl od společnosti Dragon. Nesl název Dragon Dictate a jeho cena se pohybovala v té době okolo 6000 amerických dolarů. Pro představu, při dnešním kurzu to je zhruba 131 tisíc českých korun. Jeho hlavní funkcí bylo převádění mluveného (anglického) slova do textové podoby. Mezi jeho hlavní výhodu patřilo bezpochyby množství rozpoznatelných slov, nicméně pořád se jednalo o rané pokusy a vše mělo i své stinné stránky. Například, aby bylo slovo korektně rozpoznáno, musela se celá věta „kouskovat“ po slovech.

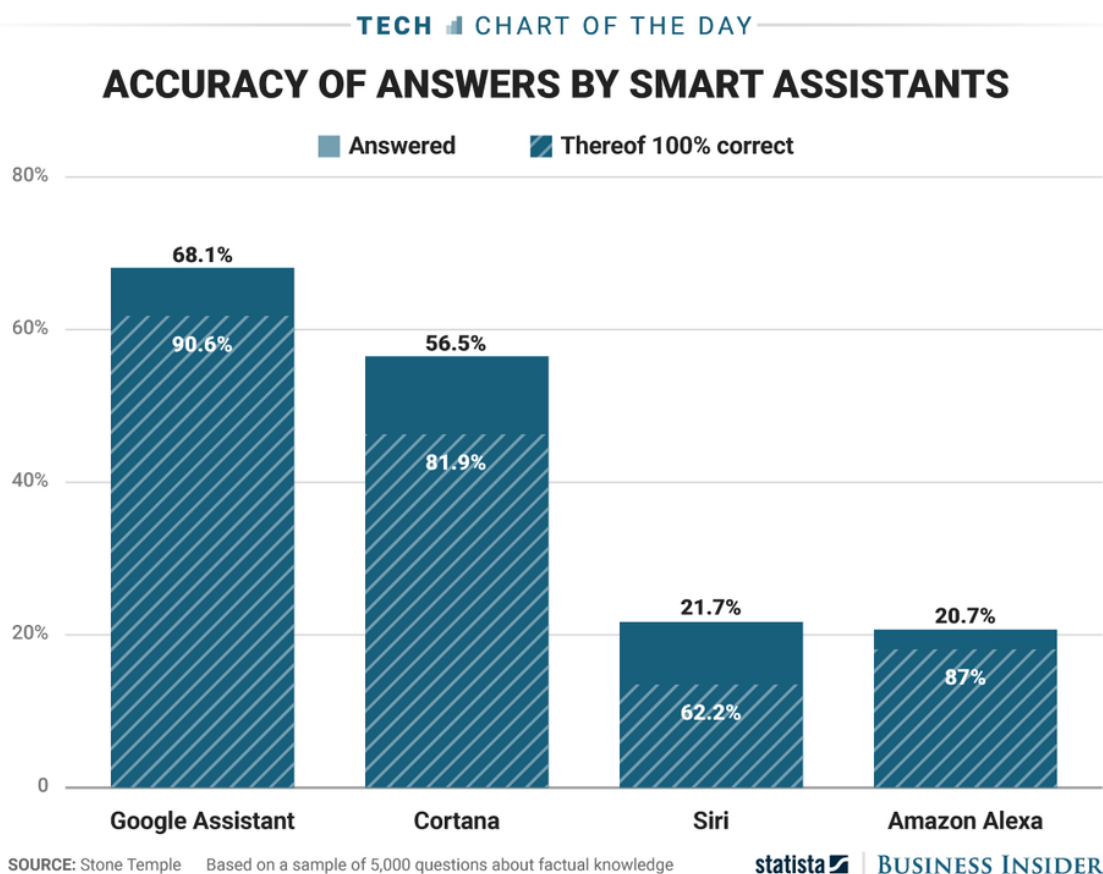
Dalším velkým a nechvalně známým krokem ve vývoji této technologie byl asistent Clippy, který byl dostupný uživatelům v aplikacích Microsoft Word od verze 97 do 2003. Jeho úkolem bylo sledovat vše, co jsme v dokumentu udělali a občas uživatele doslova zaplavoval různými radami a triky, které ne vždy byly užitečné a především požadované.

I když tato interaktivní sponka občas dokázala být trochu otravná, ukázala důležitou část dnešních asistentů. Reagovala na určitý typ dotazu, případně rozpoznala určitou formu psaného dokumentu a snažila se dodat co nejrelevantnější výsledek. Tento princip byl v podobné formě používán donedávna. Někteří hlasoví asistenti dokázali v počátku reagovat pouze na předem definované fráze, které byly pro uživatele předem určeny. Až dnešní inteligentnější asistenti nepracují jen s klíčovými slovy, ale snaží se rozpoznat smysl uživatelské věty tak jako normální člověk za využití umělé inteligence.

První opravdoví asistenti přišli až v tomto desetiletí, konkrétně v roce 2011. Tehdy Apple spojil své technologie pro rozpoznávání řeči, výsledky hledání a hlasovou syntézu a představil první hlasovou asistentku s názvem Siri. I před představením bylo možné ovládat mobilní telefon hlasem, nicméně se jednalo jen o prosté využití klíčových slov typu: zavolej někomu, přehraj něco. Siri překvapivě není od počátku projekt Applu, jak by si někteří uživatelé mohli myslet. Siri byla nejprve aplikace třetí strany, která šla běžně stáhnout z App Storu a byla původně vyvíjená SRI International, centrem, které se zabývá umělou inteligencí. Apple se ovšem v roce 2010 rozhodl projekt koupit a zapracovat přímo do systému.

Další velké softwarové firmy si nenechaly utéct potenciální velkou příležitost. Google tak následně představil svou službu Google Now o rok později, tedy v roce 2012, dva roky od uvedení Siri přivedl na svět svoji asistentku i Microsoft pod názvem Cortana. Kromě mobilních telefonů byly následně asistenti představeni i v podobě chytrých reproduktorů. Tu první, Alexu, představil Amazon v roce 2014 s reproduktorem Amazon Echo. Tím se opět nechal inspirovat i Google a později uvedl Google Home, tedy svoji verzi s Google Assistant. (Kurka, 2017)

Google zároveň, dle průzkumu společnosti Business Insider ve spojení s portálem Statista z roku 2017, drží důležité prvenství tím, že jeho Google Assistant byl schopen odpovědět na největší procento z celkových 5000 položených dotazů a zároveň jeho odpovědi byly nejrelevantnější.



Obrázek 6 - Přesnost odpovědí virtuálních asistentů (Dunn, 2017)

Oproti tomu Siri, která je na trhu nejvyužívanější byla schopna odpovědět na méně jak čtvrtinu dotazů a z toho ještě téměř polovina byla zodpovězena špatně (Dunn, 2017)

3.2.4.3 Princip fungování virtuálních asistentů

Základním bodem fungování moderních asistentů je umělá inteligence a strojové učení, aby nebyli závislí pouze na předpřipravených klíčových frázích. První věcí, co program stojící za hlasovým asistentem udělá, je převod hlasu na text. Software tak očistí nahraný zvuk od hluku, následně mluvenou řeč převede na text a s tím už je software schopný pracovat. V dnešní době je převod hlasu na text možný hned v několika světových jazycích. Například Google má k převodu řeči na text JavaScriptové API (zkratka pro Application Programming Interface,

označuje v informatice rozhraní pro programování aplikací), které využívá ve svých službách Google Now a Google Assistant.

Důležitější a zároveň i zajímavější část tvoří převod textu na smysluplný dotaz. Veškeré zpracování textu dotazu probíhá na serverech poskytovatele služby a od samotného zařízení tak není vyžadována téměř žádná výpočetní síla, všechnu těžkou práci totiž odvedou výkonné servery. Umělá inteligence stojící za asistenty se totiž nesnaží pochopit slovo od slova, tak jako to dělaly staré technologie, ale snaží se pochopit význam celé věty z pohledu přirozeného jazyka, tak jak by to dělal běžný člověk.

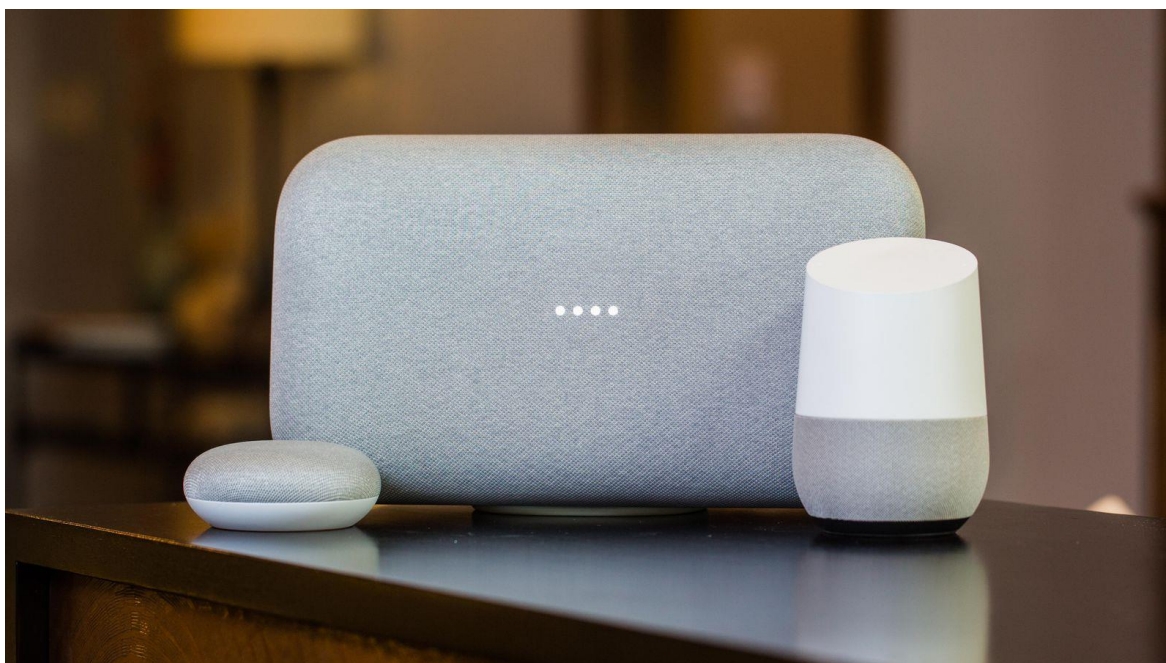
Díky vyhledávání klíčových slov je celá operace rychlejší. Díky tomu je stejně rychlé vyhledání dotazu „Siri, najdi mi prosím, jaké bude zítra počasí v Praze“ a „zítrejší počasí v Praze“. V tomto konkrétním případě je zřejmé, že se ptáme na počasí (konkrétní určení věci), zítřejší den (konkrétní určení času) a Prahu (konkrétní určení místa). Důvod je prostý: každý z nás používá jiný způsob, jak se vyjádřit. Kdyby se měl každý asistent učit jen fráze místo hledání kontextu ve větě, nikdy by se neposunul dále.

Informace, které asistent doručí, nepocházejí z jeho vlastní databáze. Pro jejich získání oslovuje služby třetích stran přes API, tím si značně rozšiřuje své možnosti. Pokud se tedy zeptáte na otázku ze světa matematiky, spojí se například se službou Wolfram Alpha (či jinou službou podobného charakteru), pokud dotaz bude směřovat například na počasí, asistent využije služby poskytující meteorologické informace, například Foreca či AccuWeather. (Kurka, 2017)

3.2.4.4 Google Home

Google Home je značkou inteligentních reproduktorů, vyvinutých společností Google. První zařízení bylo představeno v květnu 2016 a vydáno ve Spojených státech v listopadu 2016 s následnými globálními vydáními v roce 2017.

Reproduktory Google Home umožňují uživatelům využívat hlasové povely a komunikovat se službami prostřednictvím inteligentního osobního asistenta Google nazvaného Asistent Google. Velký počet služeb, a to jak z vlastní nabídky, tak i od třetích stran, je integrován, což umožňuje uživatelům poslouchat hudbu, ovládat přehrávání videí či fotografií nebo přijímat aktualizace zpráv pouze hlasem. Zařízení Google Home mají také integrovanou podporu pro domácí automatizaci a umožňují uživatelům řídit inteligentní domácí spotřebiče. Více reproduktorů Google Home lze umístit do různých místností v domě pro synchronizované přehrávání hudby či jiné aktivity. Aktualizace v dubnu 2017 přinesla podporu pro více uživatelů, což zařízení umožňuje rozlišit až šest osob dle hlasu.



Obrázek 7 - Reproduktory Google Home (Gebhart, 2018)

Originální produkt má válcový tvar s barevnými stavovými LED diodami na horním okraji pro vizuální znázornění jeho stavu a kryt nad základnou je modulární, s různými možnostmi barev nabízenými prostřednictvím služby Google Store, které umožní zařízení splynout s okolním prostředím. V říjnu roku 2017 Google oznámil dva přírůstky do produktové řady, Google Home Mini a Google Home Max. (Gebhart, 2018)

V roce 2018 Google oznámil, že se chystá v blízké době rozšířit jazykové schopnosti Google Home/Assistant, včetně toho, že zařízení bude možné ovládat více jazyky současně. Pravděpodobně nejzásadnějším přidaným jazykem bude čeština a Google Home se tedy stane prvním asistentem, který tento jazyk podporuje. Skutečnost, že v České republice sice virtuální asistenty lze využívat, ale s nutností využívat pro komunikaci například jazyk anglický odrazovala velké množství potencionálních zákazníků. (Huffman, 2018)

3.2.5 **Bezpečnost chytré domácnosti**

Koncept inteligentního domova, jak je již definován, je vybaven řadou propojených senzorů (optický, termální, pohyb, vlhkost, tlak atd.), systémy (topení, osvětlení, bezpečnost atd.) a zařízení, které jsou dostupné prostřednictvím počítače, chytrého telefonu nebo virtuálního asistenta, a to i uvnitř domova nebo vzdáleně přes internet. To přináší mnoho bezpečnostních rizik.

V poslední době technologické společnosti výrazně zdokonalují své hlasové asistenty, technologie může být v mnoha případech prospěšná, ale má velké bezpečnostní trhliny. Počítačovní experti upozorňují, že mohou být ovládáni skrze skryté příkazy například v hudbě nebo prostřednictvím nahrávek s mluveným slovem.

Hlasoví asistenti, jako je Siri, Alexa, nebo Google Assistant mohou být podle expertů z Kalifornské univerzity v Berkeley snadno zneužití pomocí příkazů, které člověk nemusí být schopen rozeznat. Hlasového asistenta mohou hackeři následně ovládnou a zneužít pro online peněžní transakce, nákup zboží na internetu, a dokonce i pro odemknutí dveří, což představuje skutečně vysoké bezpečnostní riziko. Útok hackerů je přitom velice těžko odhalitelný, protože asistenti mohou reagovat na skryté příkazy, které člověk neslyší.

Dřívější uspořádání inteligentních domů předpokládalo soudržný a jednotný model, ve kterém by jediný řídicí systém koordinoval všechny senzory a zařízení, většinu z nich poskytoval jediný dodavatel. Aktuální přístupy však zahrnují složitější prostředí, ve kterém se různé zařízení a senzory, poskytované řadou výrobců, koordinují mnoha odlišnými protokoly a připojují se a opouštějí síť podle vlastního uvážení. Tento typ dynamické sítě se řídí různými modely a protokoly, řízenými centrálním nebo distribuovanými ovládacími prvky. Tím opět vznikají zranitelná místa, jelikož je možné využít nejslabšího prvku sítě k ovládnutí jiných. Největší výzvou v chytrých domácnostech je omezená paměť a výkon zařízení a senzorů, což ztěžuje zavedení ověřených bezpečnostních funkcí.

Výchozí nastavení zabezpečení jednotlivých zařízení bývají také zpravidla nedostatečná. Většinou jsou nastavena tak, aby jejich zprovoznění a nastavení bylo pro

uživatelé co nejjednodušší. Nastavení bývá hotové během několika minut, což je sice pohodlné pro majitele zařízení, avšak výměnou za nízkou a v některých případech i žádnou bezpečnost. Způsob, jakým svá zařízení nastavíme a co vše jim povolíme dělat za nás, je přitom zásadní. Mnohé uživatele ani nenapadne změnit si výchozí nastavení, protože si pravděpodobně nejsou vědomi možných rizik, anebo zkrátka věří, že je zařízení bezpečné. (Anwar, et al., 2017)

3.3 Vlastní řešení v chytré domácnosti

Řešení pro chytré domácnosti jsou aktuálně extrémně populární. Výrobci těchto zařízení je na trhu nepřeberné množství. Ale i tak, jelikož se stále jedná relativně o mladé odvětví, mohou být pro někoho cenově nedostupná. Právě zde přichází ke slovu možnost vytvořit si chytré přístroje vlastní silou přesně na míru dle svých potřeb. Nic nebrání vytvořit systém, který bude kombinovat komerční a uživatelsky vytvářená zařízení.

S příchodem jednodeskových mikropočítačů typu Raspberry Pi nebo Arduino se otevřely naprosto nové obzory, jak k chytrým zařízením přistupovat. Uživatelé mají možnosti využít otevřenou technologii jak na poli softwarovém, tak i hardwarovém. Tato zařízení kombinují sílu počítače, komunikační a multimediální vlastnosti webu a skladnost mobilního zařízení. To vše v kombinaci s množstvím cenově dostupných čidel, senzorů, motorků a jiných prvků, kterými je možné zařízení rozšířit, přináší rozmanité možnosti zručným kutilům. Ti si mohou postavit zařízení z komponent, vyhovujících přesně jejich potřebám a naprogramovat pro ně software v jazyce, který jim nejvíce vyhovuje. A pro ty méně zručné je možné využít jednu z mnoha knih či webových stránek, které se touto problematikou zabývají a nabízejí detailní návody. (Dennis, 2015)

3.4 Jednodeskový počítač

Jednodeskový počítač, jak již název vypovídá, představuje kompletní počítačový systém, fungující na jedné malé desce. Deska obsahuje procesor, RAM, I/O a většinou i síťový port. Na rozdíl od tradičních počítačů nejsou tyto jednoduše modulární a jejich hardware nemůže být z pravidla vyměněn, jelikož je integrován do desky.

Jednodeskové počítače jsou dnes široce využívány u specializovaných řešení a mnoho jedinců i organizací vydali své produkty postavené právě na těchto systémech.

Jako hlavní průkopník odvětví je v dnešní době brán stále nejpopulárnější model Raspberry Pi, který se od roku 2016 prodává již ve třetí verzi. (Pajankar, 2015)

3.4.1 Raspberry Pi

Jedná se o miniaturní jednodeskový počítač o velikosti běžné platební karty. Jako operační systém se nejčastěji využívá distribuce Linuxu, především kvůli jeho upravitelnosti a je srovnatelný se stolním počítačem nižšího výkonu. Obsahuje nezbytné konektory jako je USB, Ethernet a HDMI. Jeho využití je hlavně v domácích projektech, jako levná a prostorově úsporná náhrada klasických počítačů. Je velmi oblíbený nejen pro běžné využití, ale i jako nástroj pro výuku programování, vývoj aplikací či jako např. multimediální centrum. Existuje již několik generací tohoto minipočítače s různým výkonem a vylepšeními.

Jádrem systému Raspberry Pi je multimediální procesor typu SoC (system on chip). To znamená, že většina systémových komponent, včetně hlavního a grafického procesoru spolu se zvukovým a komunikačním hardwarem, je integrována do jedné součástky, ukryté pod paměťovým čipem uprostřed základní desky.

Procesor se od těch, kterými jsou vybaveny klasické stolní počítače nebo notebooky však neliší jen svým návrhem typu SoC. Používá také jinou architekturu instrukční sady, která se označuje jako ARM.

Architektura ARM, kterou již v 80. letech vyvinula společnost Acorn Computers, se v počítačích třídy PC uplatňuje poměrně zřídka. Vyniká však v mobilních zařízeních. Vzhledem ke kombinaci jednoduché architektury s redukovanou instrukční sadou (RISC – reduced instruction set) a nízké spotřeby energie představuje v mobilních zařízeních mnohem výhodnější volbu než procesory stolních počítačů, které se vyznačují vysokými nároky na napájení a architekturou s komplexní instrukční sadou (CISC – complex instruction set).

Procesor třídy ARM vysvětluje, jak je možné, že počítač Raspberry Pi dokáže fungovat se zdrojem napájení s napětím jen 5 V a proudem 1 A, který poskytuje integrovaný port mikro USB. Ze stejného důvodu také na zařízení nenajdeme žádné tepelné jímky, díky nízké spotřebě energie procesor produkuje velmi málo odpadního tepla, dokonce i během komplikovaných výpočetních operací.

Zároveň to však znamená, že počítač Raspberry Pi není kompatibilní s tradičním softwarem pro PC. Většina programů pro stolní počítače a notebooky odpovídá architektuře s instrukční sadou x86, na které jsou založeny procesory značek AMD a Intel. Z toho důvodu tyto programy v počítači Raspberry Pi s procesorem typu ARM nemohou fungovat. (Upton & Halfacree, 2013)

Raspberry Pi se setkala s úspěchem ve školství a odtud se rozšířilo i pro domácí použití, kde může sloužit např. jako multimediální centrum, řídicí jednotka automatizované domácnosti,

domácí server či meteorologická stanice a mnoho dalších. Díky své ceně, rozměrům, variabilitě a možnostem vývoje našlo svoje uplatnění i v průmyslu. (Moorhead, 2012)

3.4.1.1 Vývoj Raspberry Pi

Na miniaturním, levném a jednoduchém počítači začal jeho autor, Eben Upton, pracovat zhruba v roce 2006 během svého působení jako studijní ředitel pro informatiku na Cambridgeské univerzitě. Nejprve studoval v univerzitní počítačové laboratoři a poté tam při navazující práci na svém doktorátu zároveň i vyučoval. Během tohoto období si všiml postupného patrného poklesu dovedností mladých lidí, kteří se hlásili ke studiu informatiky v laboratoři. V roce 2005 zájemci o studium měli trochu zkušeností s jazykem HTML – a když měli štěstí, občas jim něco říkaly i zkratky PHP a CSS, oproti tomu v 90. letech byly znalosti začínajících žáků mnohem vyšší, většina již při nástupu na školu uměla alespoň základy některých pokročilých programovacích jazyků a zároveň měli i povědomí o funkčnosti jednotlivého hardwaru.

Projekt Raspberry Pi byl tedy původně inspirován lokálními potřebami a měl jen jednoduchý a nepřilíš ambiciózní cíl: bylo potřeba vytvořit nástroj, který by malému počtu uchazečů o jeden z mnoha univerzitních kurzů poskytl počáteční impulz k většímu zájmu o počítačovou techniku.

Jakmile se však práce na projektu rozeběhla naplno, začal si autor uvědomovat skrytý potenciál svého nápadu, takový malý a levný počítač by toho mohl dokázat mnohem více. První prototypy příliš nepřipomínaly počítač Raspberry Pi, tak jak vypadá dnes. Jak říká sam Upton ve své knize: *„Začal jsem na mém kuchyňském stole pájet nejdelší zkušební model obvodu, který se prodává v obchodech Maplin, s čipem Atmel. První hrubé prototypy přímo ovládaly televizor se standardním rozlišením pomocí levných jednočipových počítačů.“* Tyto základní prototypy s pouhými 512 kB paměti RAM a několika MIPS (Million Instructions Per Second) neboli milion instrukcí za sekundu výpočetního výkonu se svými možnostmi velmi podobaly původním osmibitovým mikropočítačům.

Asi po pěti letech práce vznikl poměrně atraktivní prototyp základní desky velký asi jako flash disk. Na horní části desky se nacházel modul fotoaparátu, aby se ukázalo, jaké periferie by bylo možné snadno doplňovat. Prototyp byl představen oddělení výzkumu a vývoje BBC Microcomputer, kde si publicista Rory Cellan-Jones pořídil záznam prototypu a umístil ho na svůj blog. Tento záznam se následně virálním způsobem rozšířil.

Následně si prototyp Raspberry Pi prošel ještě několika úpravami a v roce 2011 se blížilo plánované únorové datum uvedení na trh. Začalo být zřejmé, že se vše vyvíjí rychleji a poptávka je mnohem vyšší, než bylo možné vůbec zvládnout. (Upton & Halfacree, 2013)

Jako první vznikl Raspberry Pi Model B v roce 2012, byl vybaven čipem Broadcom BCM2835 s 700 Mhz ARM1176JZF-S procesorem, VideoCore IV grafickým jádrem a 512 MB RAM. Tento model je dnes překonaný výkonnějšími generacemi, ale ve své druhé revizi se stále prodává a je používán. Rok 2013 přinesl zjednodušenou a levnější verzi, nazvanou Model A, který obsahoval pouze jeden USB 2.0 port a Ethernetové rozhraní nebylo integrováno. Výhodou však byla především nižší pořizovací cena a své využití našel například v robotice či v zabudovaných projektech, díky své skladnosti a nízké spotřebě a váze.

V roce 2014 byly představeny vylepšené modely Raspberry Pi 1 A+ a B+. Současně byl představen i model Raspberry Pi Compute Module, určený pro průmyslové využití. Od této doby obsahuje Raspberry Pi 40 GPIO pinů a místo klasické SD karty je použita microSD.

Na začátku roku 2015 byla na trh uvedena nová generace Raspberry Pi 2 Model B s výkonnějším čtyřjádrovým procesorem Cortex-A7 900 Mhz a 1 GB paměti RAM, která byla sdílená i pro grafické jádro. Tento rok se na trhu objevil model Raspberry Pi Zero, který byl dokonce o polovinu menší, než model A+, ale zároveň dvakrát tak výkonný. Pro svou extrémně nízkou cenu, skladnost a nízkou spotřebu se stal ideálním pro použití v IoT.

Nejnovějším produktem z hlavní série je Raspberry Pi 3 Model B, který opět přinesl výkonnější procesor, Wi-Fi 802.11n a Bluetooth 4.1, díky čemuž se oprostil od nutnosti využívání kabelového připojení. Na jaře roku 2018 byl představen vylepšený model B+, který především zlepšil kvalitu konektivity navýšením rychlosti ethernetového portu na 300 Mbit/s, pro Wi-Fi přinesl protokol 802.11ac, která již pracuje ve dvou pásmech 2,4 a 5 Ghz a novou verzi Bluetooth 4.2, jako bonus bylo navýšení rychlosti procesoru na 1,4 Ghz. (Hitaltech, 2018)

3.4.1.2 Raspberry Pi Foundation

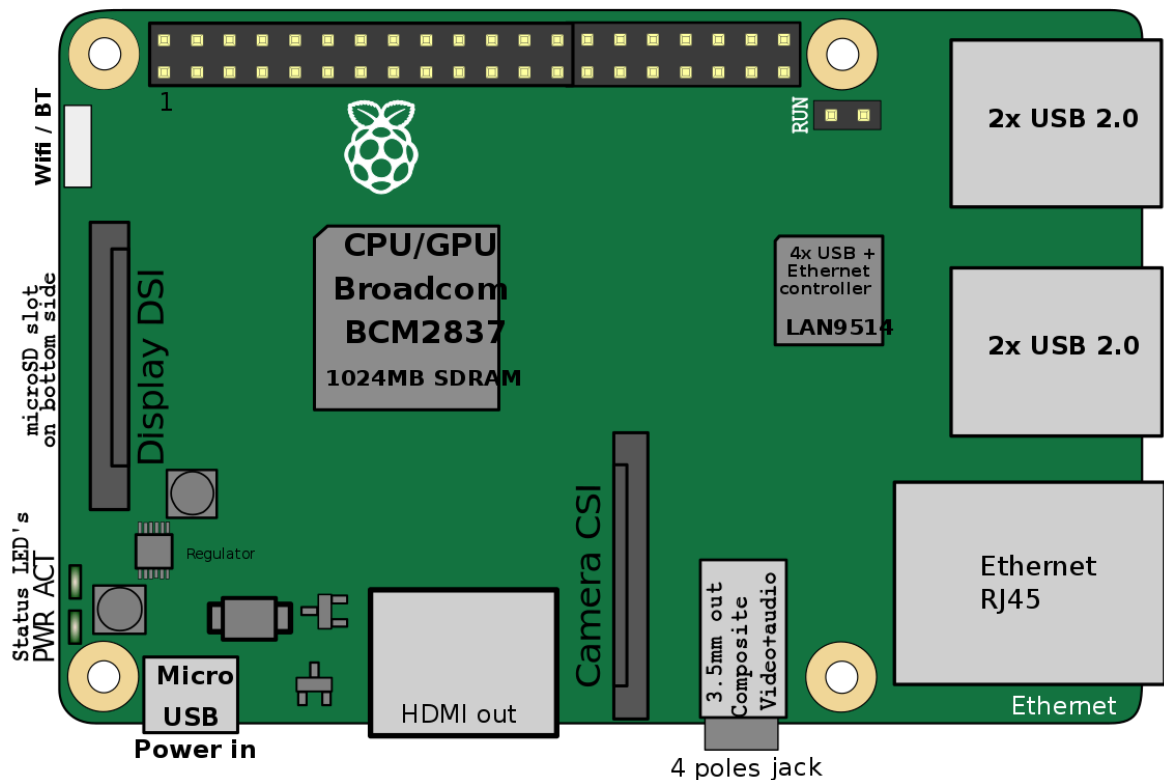
Společně s kolegy dr. Robem Mullinsem, profesorem Alanem Mycroftem, Jackem Langem, Petem Lomasem a Davidem Brabenem založil Eben Upton malou nadaci s názvem „*Raspberry Pi Foundation*“. (Upton & Halfacree, 2013)

Jejím cílem je především vývoj základních desek mikropočítačů Raspberry, ale i to, aby uživatelé po celém světě porozuměli počítačům a v dnešní době digitalizace byli schopni usnadnit si řešení svých potřeb právě využitím elektroniky a počítačů. Poskytují tedy levné, ale zároveň výkonné počítače, které mohou nejen uživatelé, ale například i školy používat ke

vzdělávání i zábavě. Vše, co nadace vyvíjí, je volně dostupné tak, aby se lidé mohli sami učit a seznamovat se s počítači. (Raspberry Pi Foundation, 2018)

3.4.1.3 Architektura Rapsberry Pi

Zařízení Raspberry Pi vypadá na první pohled jako malá základní deska s připojenými čipy a porty. Ve skutečnosti ale má vše, co potřebujete pro vstup, výstup, ukládání dat i provádění výpočtů. Hardware tohoto mikropočítače, který ve spojení se správným softwarem dokáže nahradit svojí funkcí i běžný základní počítač, se od jeho prvního prototypu razantně změnil a vyvíjí se nadále, stejně tak jako vznikají nové systémy, které je možné na tomto zařízení spouštět.



Obrázek 8 - Deska Raspberry Pi 3 B (Wikipedia.org, 2018)

3.4.1.3.1 Hardware

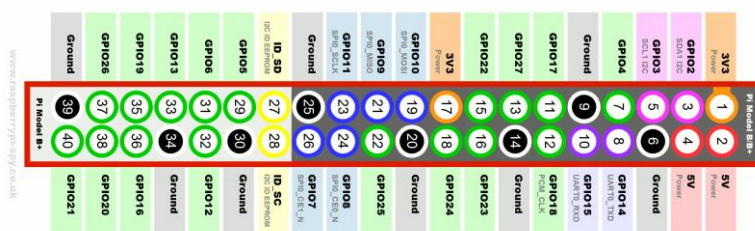
Hardwarové vybavení se s vývojem jednodušších modelů lišilo, měnily se využité čipy, konektory i další vybavení, tento vývoj bude pravděpodobně pokračovat i s nastupujícími modely. Aktuálně nejnovější model Raspberry Pi 3, který je v této práci využit, byl představen v roce 2016 a v roce 2018 vyšla jeho vylepšená verze, která ale v základu obsahuje podobné vybavení.

Základem je čip Broadcom BCM2837B0 založený na čtyřech 64bitových jádrech ARM Cortex-A53. Zde se nachází v nejnovější revizi B0, což umožňuje běh na taktu 1,4 Ghz, z důvodu lepšího odvodu tepla s nárůstem výkonu byl opatřen rozvaděčem tepla. Základem konektivity je LAN konektor, kde je využit čip LAN7515 s podporou gigabitového Ethernetu (reálně 300 Mbit/s kvůli jeho napojení přes USB 2.0) a navíc i s funkcí Power over Ethernet (PoE, napájení po datovém síťovém kabelu, bez nutnosti přivést napájecí napětí k přístroji dalším samostatným kabelem). Bezdrátová konektivita je řešena čipem Cypress CYW43455 s anténou umístěnou v plošném spoji a s podporou dvoupásmové Wi-Fi.

Výše zmíněné části procházely vývojem nejčastěji, jelikož se jednalo o komponenty, které mají nejmarkantnější vliv na výkon. Zbytek hardwaru, kterým je deska Raspberry Pi osazena, se také samozřejmě měnil a zdokonaloval, ale základy zůstávají podobné. Napájení celé desky je realizováno pomocí konektoru Micro USB a díky nenáročnosti ARM procesoru stačí běžně dostupné napájení stejné jako například od mobilních telefonů 5 V a 1-2 A. Jako obrazový a zvukový výstup slouží u nejnovějšího modelu plnohodnotný HDMI konektor a jako doplňující možnost se zde nachází 3,5 mm JACK konektor pro výstup zvuku a vstup mikrofону nebo výstup kompozitního videa, poslední možností je DSI konektor pro připojení například oficiálního Raspberry displeje. Pro připojení dalších periférií je deska vybavena několika porty USB, bohužel stále v již poměrně zastaralé verzi 2.0. Jako úložiště Raspberry Pi využívá microSD paměťové karty. Poslední konektor, který se na desce nachází, je CSI vstup pro kameru. (Vítek, 2018)

3.4.1.3.2 GPIO

Raspberry Pi obsahuje navíc i speciální vstupně/výstupní piny GPIO (General-purpose input/output), které je následně možné ovládat pomocí mnoha knihoven v programovacích jazycích, jako je například Python nebo Java. Prakticky se jedná o dvě řady pinů, u prvních modelů jich bylo celkem 26, ale od modelu Pi1 B+ byl počet pinů navýšen na 40. Na obrázku níže je možné vidět aktuální rozložení pinů na desce Raspberry Pi



Obrázek 9 - Rozložení GPIO pinů Raspberry Pi 3 B (Raspberry Pi Spy, 2012)

Raspberry Pi interně funguje s logikou napájení 3,3 V. V případě, že bychom chtěli vytvořit obvod, který bude s deskou komunikovat prostřednictvím právě portů GPIO, je potřeba využít komponenty, které budou kompatibilní právě s logikou 3,3 V nebo které před propojením s počítačem obcházejí obvod přes regulátor napětí. V případě, že by došlo k připojení zdroje napájení 5 V (kromě dvou 5 V pinů, které jsou speciálně pro toto napětí připravené) je pravděpodobné, že dojde k poškození počítače, jelikož piny jsou přímo propojeny s piny procesoru.

Kromě univerzálních pinů má port GPIO i piny, které jsou vyhrazeny pro speciální sběrnice. Konkrétně se jedná o:

- **Sériová sběrnice UART**
 - Poskytuje jednoduché sériové rozhraní se dvěma vodiči, pokud se tato sběrnice připojí k zařízení, které umí zprávy zobrazovat, je možné sledovat zprávy jádra systému Linux, jedná se o praktický diagnostický nástroj.
- **Sběrnice I²C**
 - Jak vyplývá z jejího názvu (Inter-Integrated Circuit), má zajistit komunikaci mezi více integrovanými obvody. Jedná se opět o sériovou sběrnici, která se využívá k připojení nízko rychlostních periférií. Připojená zařízení se dělí na řídicí a řízené.
- **Sběrnice SPI**
 - Synchronní sériová sběrnice, která je určena zejména pro interní systémové programování mikrokontrolerů a jiných zařízení. Na rozdíl od zbylých dvou sběrnic má tato čtyři vodiče s více linkami výběru čipu, které jí dovolují komunikovat s více než jedním cílovým zařízením. (Upton & Halfacree, 2013)

Dva GPIO porty jsou ještě navíc schopné PWM (Pulse Width Modulation), což znamená, že vysílají krátké pulzy, pomocí kterých je kontrolováno, jak silně jsou napájené, tím je možné kontrolovat například výkon motorů nebo sílu svítivosti LED. (Buckley, 2018)

3.4.1.3.3 Software

Téměř většina běžných stolních počítačů či notebooků, které se v současnosti prodávají, pracuje s jedním ze dvou operačních systémů: Microsoft Windows nebo Apple OS X. Obě platformy patří do kategorie „*closed source*“ a jejich vývoj probíhá v nepřístupném prostředí s použitím proprietárních metod. Důvodem, proč se takto tyto systémy označují, je nepřístupnost ke zdrojovému kódu, což je popis v počítačovém jazyce, který systému sděluje, co má dělat. V tomto případě zůstává zdrojový kód pečlivě hlídaným tajemstvím společnosti, která se o vývoj systému stará. Uživatelé tak získají hotový software, ale nemají možnost se podívat, jak je vytvořen nebo se na jeho následném vývoji podílet.

U Raspberry Pi se oproti tomu předpokládá, že na něm bude fungovat operační systém s názvem GNU/Linux. Na rozdíl od výše zmíněných systémů je Linux typu open source: je možné si stáhnout zdrojový kód celého operačního systému a změnit či přidat cokoli, co uživatele napadne. Díky tomuto stylu vývoje open source bylo možné standardní systém Linux rychle přizpůsobit ke spuštění v počítači Raspberry Pi. V dnešní době bylo pro počítače Raspberry Pi portováno již několik verzí systému Linux, které se nazývají distribuce. Patří k nim Debian, Fedora Remix a Arch Linux. Jednotlivé distribuce se zaměřují na odlišné uživatelské požadavky, ale všem je společné to, že se jedná o projekty open source. Vesměs jsou také vzájemně kompatibilní. Systém Linux není omezen jen na počítače Raspberry Pi. K dispozici jsou stovky různých distribucí určených pro stolní počítače, notebooky, a dokonce mobilní zařízení. Z jádra Linux vychází i oblíbená platforma Android společnosti Google.

Stejně jako byl výše popsán rozdíl mezi procesory ARM a x86 je i zde potřeba zmínit důležitý fakt, týkající se praktické odlišnosti systémů Windows, OS X a Linux, software určený pro systém Windows či OS X nebude v systému Linux fungovat. Pro naprostou většinu běžných softwarových produktů však naštěstí existuje mnoho odpovídajících alternativ. Většinu z nich lze navíc používat zdarma a patří do kategorie open source stejně jako samotný operační systém. (Upton & Halfacree, 2013)

Oficiální distribucí, která je doporučována jako primární pro Raspberry Pi je Raspbian, vyvíjený samotnou Raspberry Pi Foundation. Jedná se v podstatě o Linuxovou distribuci Debian speciálně upravenou pro hardware Raspberry Pi, od toho se také odvíjí název, který spojuje Raspberry a Debian. Byl vytvořen Mikem Thompsonem a Peterem Greenem v roce 2012 a i nadále je v aktivním vývoji. Jako hlavní desktopové prostředí využívá PIXEL (Pi Improved Xwindows Environment, Lightweight), který umožňuje plynulý chod i na slabším

hardwaru. Existuje i odlehčená verze, která je o grafické rozhraní oproštěna a funguje tak pouze v terminálu. (Raspberry Pi Foundation, 2018)

Existuje i speciální verze Windows od společnosti Microsoft s názvem Windows 10 IoT Core, která je navržena právě pro práci na minipočítačích typu Raspberry Pi, oproti běžným Windows je ale systém razantně pozměněn a zatím se stále vyvíjí, do širšího povědomí mezi uživateli Raspberry Pi se prozatím nedostal. (Bechynský, 2015)

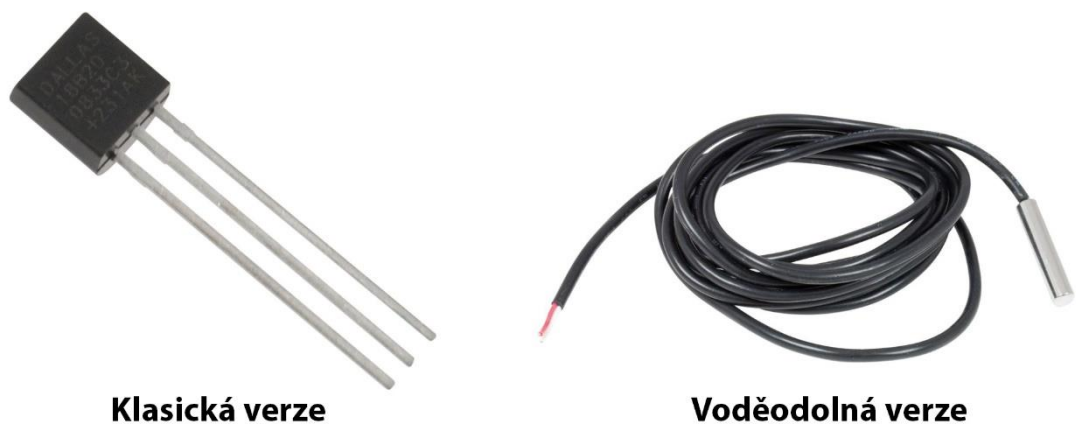
3.4.2 Komponenty pro Raspberry Pi

Jednou z mnoha výhod, které nabízí Raspberry Pi, je skutečnost, že je možné připojit obrovské množství různých senzorů a komponent k GPIO pinům a tím znatelně rozšířit funkčnost tohoto zařízení. Jedná se například o různá čidla pro zpracování meteorologických hodnot, přes čidla detekující pohyb či světlo, gyroskopy, kompas, až po různé motorky či dotykové obrazovky.

Hodnoty, které z těchto rozšiřujících komponent získáme, je následně možné zpracovávat a vyhodnocovat programy již existujícími nebo vytvořenými pro vlastní speciální potřeby. Toto příslušenství lze použít v projektech, jako je například meteorologická stanice, robot nebo jako komplexní řešení chytré domácnosti. (Raspberry Pi Tutorials, 2019)

3.4.2.1 Teplotní čidlo DS18B20

Jedná se o teplotní senzor (nebo také čidlo), vyrobený firmou Maxim (dříve Dallas), který je v komunitě uživatelů Raspberry a jiných mikropočítačů velice oblíbený. Za velmi dobrou cenu umožňuje měřit teplotu v rozsahu -55 až +125 stupňů Celsia, přičemž v rozsahu -10 až +85 stupňů Celsia má garantovanou přesnost $\pm 0,5$ °C.



Obrázek 10 - Dvě verze čidla DS18B20 (GM Electronic, 2019)

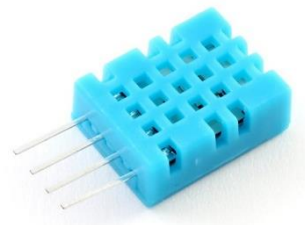
Další výhodou je také možnost zakoupit tento teplotní senzor jak v pouzdře TO-92, které se velikostí podobá obyčejným tranzistorům, tak i vodotěsnou variantu, kdy je senzor zataven v nerezové tyčince. Pro komunikaci s deskou je u čidla DS18B20 využita sběrnice OneWire, která využívá pouze jeden komunikační pin a toto čidlo také podporuje takzvaný parazitní režim, kdy pro spojení čidla s deskou stačí využít pouze 2 dráty. (Arduino návody, 2016)

Pro správné a funkční připojení čidla k desce Raspberry Pi je nejprve nutné doplnit samotné čidlo o odpor 4k7 nebo 10k. Tento odpor je potřeba připojit mezi napájení a datový konektor čidla. Následně už je možné čidlo samotné připojit na GPIO piny. (Arduino návody, 2016)

3.4.2.2 Čidlo vlhkosti DHT11

Vlhkoměr DHT11 je snadno použitelný modul, který obsahuje vše potřebné již ve svém pouzdře. Je tedy pouze potřeba ho připojit třemi vodiči k GPIO pinům desky Raspberry Pi a dále už jen vyčítat nové informace o okolí, které se obnovují přibližně každé dvě vteřiny.

DHT11 měří teplotu v rozsahu 0–50 stupňů Celsia s přesností ± 1 °C a vlhkost v rozmezí 20–90 % s přesností ± 4 %. (Arduino návody, 2016)



Obrázek 11 - Čidlo DHT 11 (GM Electronic, 2019)

3.4.2.3 Pohybové čidlo HC-SR501

Toto čidlo obsahuje pyroelektrický senzor, který snímá své okolí pod úhlem přibližně 120 stupňů. Pokud v tomto snímaném prostoru proběhne větší změna teplot, tedy například projde člověk, je tato změna zaznamenána pomocnou elektronikou na modulu pohybového čidla a zařízení, které senzor využívá, se o této skutečnosti dozví díky změně výstupního napětí na datovém pinu. Na senzoru se také nachází dva trimry. Pomocí prvního lze nastavit citlivost

snímání okolí, druhým trimrem se nastavuje časová prodleva, respektive délka sepnutého výstupu po detekci pohybu. (Arduino návody, 2016)



Obrázek 12 - Pohybové čidlo HC-SR501 (GM Electronic, 2019)

3.5 Windows Subsystem for Linux

„*Windows Subsystem for Linux*“, neboli v často využívané zkratce „*WSL*“, je speciální vrstva kompatibility, která byla poprvé představena ještě pod názvem „*Bash on Windows*“ jako beta funkce v roce 2016 v Insider programu pro systém Windows 10. Od té doby si již prošla značným vývojem a nyní je již dostupná i pro širokou veřejnost a kromě Windows 10 je dostupná již i pro Windows Server.

Jak název napovídá, díky WSL je možné instalovat a spouštět různé Linuxové distribuce během několika minut v prostředí systému Microsoft Windows 10 bez nutnosti vytváření virtuálního stroje. Jediné, co je potřeba pro funkčnost WSL udělat, je tuto funkci povolit a následně si již jen vybrat požadovanou distribuci v Microsoft Store.

WSL poskytuje rozhraní jádra, které je kompatibilní s Linuxem a umožňuje spouštět systémy GNU jako je například Ubuntu, openSUSE nebo Debian. Základním rozhraním pro ovládání systému se stává terminál dané distribuce, kterým je možno za pomoci Linuxových příkazů procházet i souborový systém Windows, případně i spouštět Windows aplikace. WSL umožňuje spustit i Linuxové aplikace s vlastním grafickým rozhraním, využitím protokolu X11 a patřičné aplikace v hostitelském systému Windows 10. (Hesse, 2018)

Tato funkce je určena především pro vývojáře a vývojářské nástroje, jelikož umožňuje jednoduše vyvíjet či testovat na obou systémech téměř zároveň a oproti virtuálním strojům umožňuje využívat silné stránky obou systému a jejich propojené možnosti. Například díky WSL může webový vývojář psát kód v Linuxovém prostředí, ale zároveň monitorovat a testovat přesně tu samou stránku v prohlížeči v systému Windows bez nutnosti placení za další software. Zároveň díky tomu, že WSL běží jako proces ve Windows, je možné s ním také nástroji Windows pracovat, například tedy můžeme ukončit Linuxový proces pomocí správce úloh, když dojde k chybě.

Nevýhodou WSL je, že není schopen spustit veškeré Linuxové aplikace, například ty, které vyžadují speciální služby kernelu, které doposud nebyly ve WSL implementovány. To samé platí i pro aplikace s grafickým rozhraním, které je sice možné využívat díky protokolu X11, ale jen v omezené míře, nepodporuje například přenos zvuku a ani hardwarovou akceleraci, čímž může docházet k problémům u aplikací s náročným grafickým rozhraním. (Microsoft, 2016)

3.6 Python

Programovací jazyk Python přitahuje nové uživatele z různých důvodů. Jedná se o jazyk pro více platforem, funguje stejně dobře na systémech Windows, Linux / UNIX nebo Macintosh, stejně tak na superpočítačích nebo mobilních telefonech. Lze ho využít pro vývoj malých aplikací a rychlých prototypů, avšak je vhodný i pro vývoj velkých a komplexních aplikací. Dodává se s výkonným a snadno použitelným grafickým rozhraním (GUI), knihovnamí pro webové programování a je zdarma.

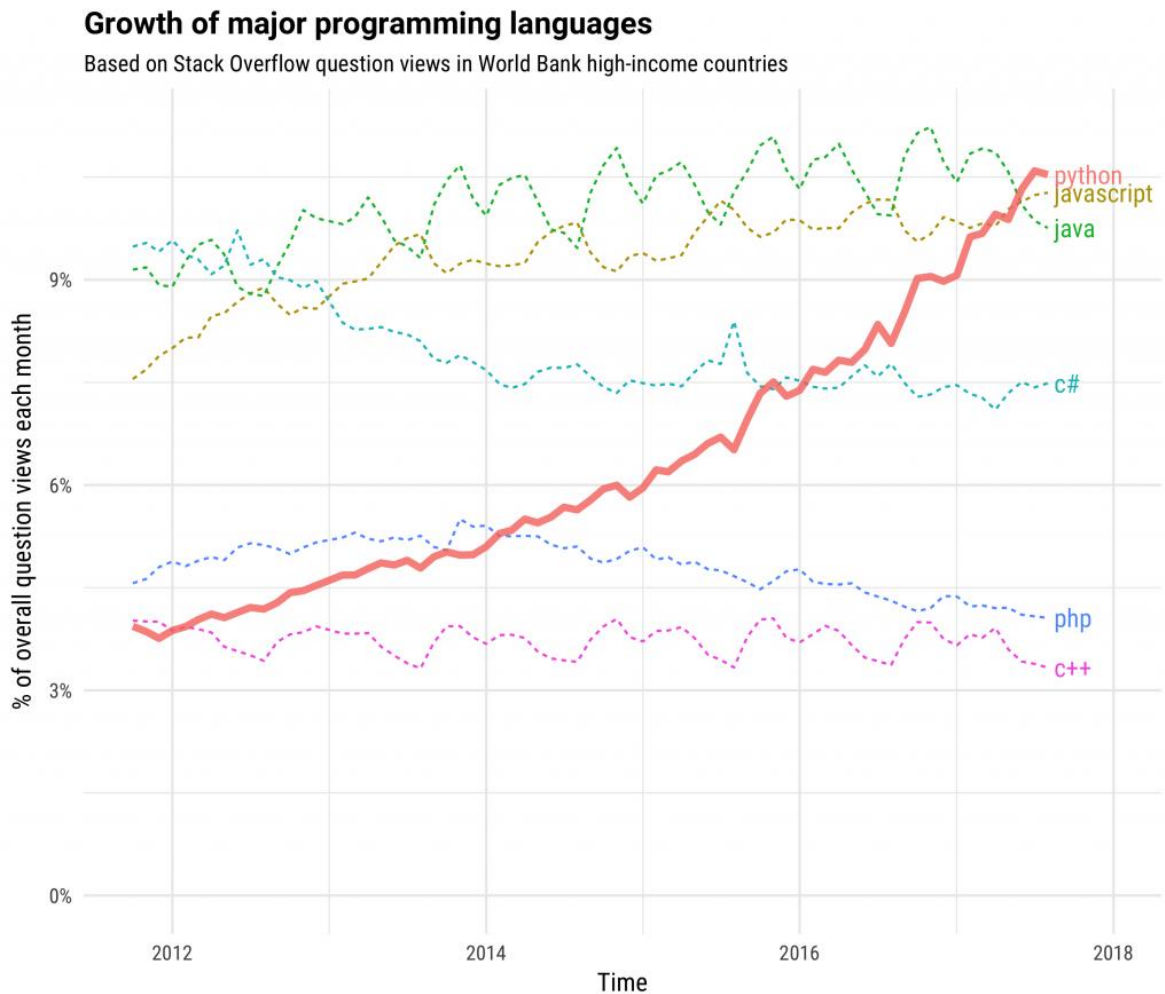
Python je interpretovaný, objektově orientovaný, vysokoúrovňový programovací jazyk, který v roce 1991 navrhl Guido van Rossum. Nabízí dynamickou kontrolu datových typů a podporuje různá programovací paradigmaty, včetně objektově orientovaného, imperativního, procedurálního nebo funkcionálního. I když Python není perfektní pro každou aplikaci, jeho síla a univerzalita ho dělá dobrou volbou pro velké množství situací.

Jedná se o open source projekt a je dostupný pro většinu běžných platforem, přičemž ve většině Linuxových distribucí je Python součástí základní instalace. V Pythonu je kladen důraz na snadno naučitelnou syntaxi, čitelnost a přehlednost kódu, čehož je dosaženo například využitím bílých znaků (whitespace) k vymezení jednotlivých bloků kódu místo různých druhů závorek. Tento jazyk má obsáhlé vyjadřovací schopnosti a kód programu bývá kratší, než kdyby byl napsán v jazycích jako je C++ nebo Java. Python umožňuje psát programy jak v malém, tak velkém měřítku.

V současné době (2019) se používají dvě nekompatibilní verze Pythonu 2.x a 3.x. Python 3.x byl vydán v roce 2008 a odstraňuje řadu nedostatků a chybných návrhů jazyka, které nešlo odstranit bez narušení kompatibility. Python 3.x má proto ucelenější návrh.

Ale i po vydání 3.0 byl Python 2.x stále vyvíjen a dostávaly se do něj nové vlastnosti z 3. řady. Přejít na novou verzi, portace knihoven třetích stran a následně programů tak byl a stále je nad očekávání pomalý a není ukončen dodnes. Například v Linuxových distribucích jsou běžně nainstalované obě verze Pythonu. V roce 2017 bylo ohlášeno, že Python 2.x přestane

být podporován ze strany vývojářů v roce 2020. Lze očekávat, že jeho útlum nabere rychlejší spád. (Ceder, 2018)



Obrázek 13 – Srovnání jazyka Python s jinými (Robinson, 2017)

Jak je možné vidět z průzkumu portálu StackOverflow, popularita jazyka Python v posledních letech extrémně roste. Může za to pravděpodobně jeho relativní jednoduchost, ale určitě i jistá univerzalita, která tento jazyk staví na přední příčky mnoha vývojářů, při rozhodování, v jakém jazyce jejich budoucí projekt bude vznikat. Jeho všestrannost ale není bez hranic a je tedy vždy nutné detailně prozkoumat odlišnosti oproti jiným jazykům, jeho výhody a nevýhody v závislosti na daném projektu. (Robinson, 2017)

4 Vlastní práce

4.1 Výběr řešení chytrého zrcadla

Skrytou silou, kterou chytrá zařízení nabízí v dnešní době je především možnost vytvořit si své vlastní. Tento postup se často popisuje anglickou zkratkou DIY neboli „*Do It Yourself*“, přeloženo do českého jazyka „*Navrhni (postav) si sám*“. Zruční uživatelé tak mají díky rozmanitému množství dostupných technologií, co se týče hardwarové i softwarové stránky, možnost sestavit si zařízení přesně na míru jejich potřebám, a to poté mohou napojit i na některý z komerčních chytrých ekosystémů.

Tato kapitola se konkrétně zabývá vlastní realizací chytrého zrcadla. Chytrým se v dnešní době může stát již téměř každé zařízení a mnoho se jich také již na trhu v komerční podobě nachází. Zrcadlo bylo vybráno právě proto, že je to prozatím poměrně nedotčená oblast, dostupných funkčních řešení je málo a je tedy možné využít vlastních nápadů a v případě úspěšné realizace i využít příležitosti a zaplnit díru na trhu. Koncept není založen pouze na subjektivních dojmech autora, ale i na podnětech potencionálních uživatelů.

4.1.1 Definice variant

Proto, aby bylo možné vybrat vhodnou variantu implementace chytrého zrcadla v domácnosti, je nejprve nutné si definovat, jaké možnosti do výběru zahrneme. Aby nebyly založeny pouze na subjektivním názoru autora práce, je potřeba získat podklady od potencionálních uživatelů, kteří mohou nezájatě popsat, jak si oni sami představují ideální využití chytrého zrcadla. Sběr těchto informací byl proveden pomocí anonymního internetového dotazníku (Google Formuláře), který vyplnilo 20 respondentů. Pro účely práce je tento vzorek dostačující, jelikož již z tohoto množství odpovědí bylo možné stanovit dostatečné množství variant, mezi kterými je možno vybírat a počet možných způsobů technické realizace by se pravděpodobně ani s větším vzorkem účastníku nijak zásadně nezměnil.

Pro účely průzkumu bylo definováno několik otázek s možností otevřených odpovědí. Tyto otázky se snaží od respondentů zjistit, co by od zrcadla očekávali, jakým způsobem by s ním chtěli interagovat a také, zda by byli ochotní pokusit se o konstrukci vlastního řešení, podobně jako je to předvedeno v této práci. Konkrétně se jedná o otázky:

- *Co si představíte pod pojmem „chytré zrcadlo“, jak by vám mohlo zjednodušit, zpříjemnit či zefektivnit každodenní činnosti?*

- *Kde by podle Vás bylo nejlepší chytré zrcadlo v domácnosti umístit?*
- *Jakým způsobem byste od zařízení chtěli získávat informace nebo jinou zpětnou vazbu?*
- *V případě že by bylo možné zrcadlo určitým způsobem ovládat, jaký způsob by pro Vás byl nejpohodlnější?*
- *Pokud by byly dostupné detailní podklady, pokusili byste se o konstrukci takového zařízení vlastní silou nebo byste dali přednost koupi již hotového řešení?*

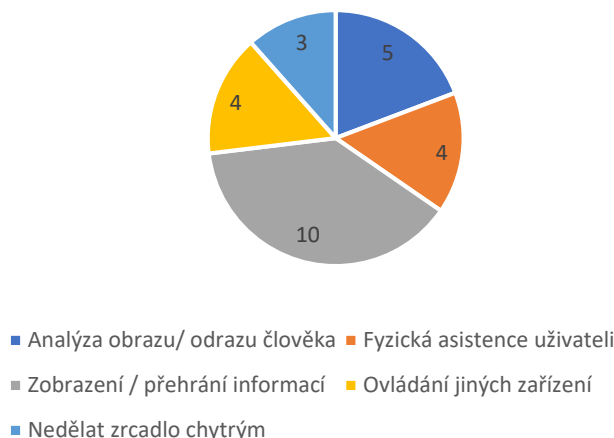
Dotazník byl uveden stručným popisem, co chytrá zařízení v domácnosti jsou, aby i člověk bez informací o dané oblasti měl alespoň základní přehled o tom, na co je dotazován. Popis zněl takto: „*Chytrý dům nebo chytrá zařízení mohou být definovány jako obydlí či zařízení vybavené výpočetními a informačními technologiemi, které předvídají a reaguje na potřeby obyvatelů, snaží se zlepšovat jejich komfort, pohodlí, bezpečnost a zábavu právě využitím zmíněných technologií a propojení s vnějším světem. Jejich cílem je usnadnění každodenních činností a zefektivnění situací, ve kterých se člověk každodenně nachází. Například je možné automaticky řídit osvětlení dle denní doby, uživatelů přítomných v obydlí či jiných atributů anebo můžeme mít chytrý zubní kartáček, který nám pokaždé řekne, zda jsme si správně vyčistili zuby, to vše lze považovat za inteligentní zařízení. Nejznámějším chytrým zařízením, který většina z nás používá je především mobilní telefon, umožňuje nám nejen telefonovat, ale i nepřehledné množství dalších činností. Cílem tohoto dotazníku je zjistit, jak by si potenciální uživatel představoval chytré zrcadlo, jaké výhody by mu mělo přinést a jakým způsobem by s ním chtěl pracovat“*

Samotné odpovědi jsou v příloze práce, pokud ale vybereme klíčová slova z většiny odpovědí, je možné definovat aspekty, které se napříč prolínají a na jejich základě poté vybrat několik konstrukčních variant, se kterými budeme dále pracovat.

Jak je vidět, z první otázky je možné vyčíst, jakým způsobem by chytré zrcadlo mělo fungovat, aby pro respondenty mělo opravdu nějakou „chytrou“ přidanou hodnotu. To je klíčovým bodem, ze kterého je možné vycházet při tvorbě konceptů pro realizaci.

Některé odpovědi zahrnovaly u některých otázek i více možností nebo nebylo zřejmé, jaký je přesně respondentův názor, proto v některých případech počet odpovědí neodpovídá počtu respondentů.

Princip fungování



Obrázek 14 - Dotazník: Princip fungování (Vlastní zpracování)

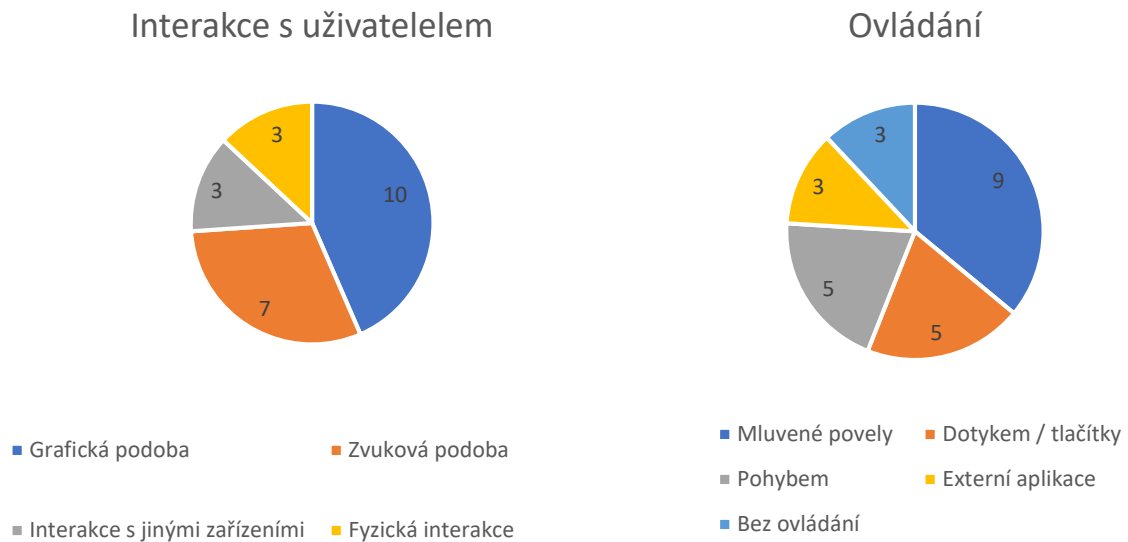
Ze tří odpovědí, je možné vyčíst, že by zrcadlo mělo buď zůstat klasickým pasivním zrcadlem. Jako důvod je zmiňováno především to, že zrcadlo by mělo především sloužit k odrazu reality a není potřeba jeho schopnosti rozšiřovat, druhým důvodem je také strach o soukromí, kdy by zrcadlo například mohlo sbírat informace bez vědomí uživatele.

Čtyři respondenti by ocenili, kdyby jim zrcadlo mohlo pomoci fyzicky, například vyčistit zuby. Taková technologie je aktuálně asi nejvzdálenější realitě. Vyžadovala by extrémně přesný a složitý hardware, kde by musely být sestrojeny mechanické paže, které by byly schopné jemných pohybů (například čištění zubů, holení atd.) a ty by byly kontrolovány pomocí komplexního softwaru.

V dalších čtyřech případech by mohlo zrcadlo sloužit jako centrální kontrolní bod pro ovládání jiných zařízení v chytré domácnosti. Například uživatel si může z koupelny zapnout rychlovarnou konvici. Takové řešení by nemuselo být komplikované, v případě že by tomu byl samozřejmě uzpůsoben i zbytek domácnosti. K samotnému ovládání by mohl být využit například některý z virtuálních asistentů.

V pěti odpovědích si respondenti zrcadlo představují jako něco, co bude analyzovat jejich vzhled a na základě toho například doporučovat co na sebe nebo radit s problémy pleti. Takový systém by vyžadoval kvalitní snímací i zobrazovací zařízení, ale především špičkový software, založený na umělé inteligenci, který by obraz zpracovával a vyhodnocoval.

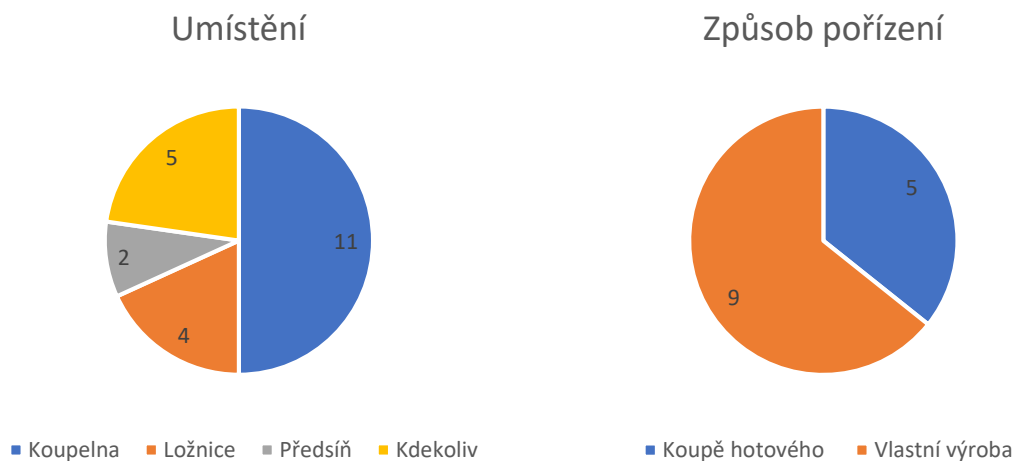
Nejvíce, tedy 10 odpovědí se shodovalo v tom, že potenciál chytrého zrcadla by mohl být v zobrazování nebo přehrávání různých informací získaných z internetu, jiných zařízení v domácnosti či různých senzorů. Zde by záleželo, jaké informace by chtěl uživatel zobrazit a také jakým způsobem.



Obrázek 15 - Dotazník: Interakce s uživatelem a ovládání (Vlastní zpracování)

Další dva dotazy sloužily ke zjištění, jakým způsobem by respondenti chtěli se zrcadlem interagovat. Nejvíce odpovědí zmiňovalo možnost, kdy by jim zrcadlo mohlo zobrazovat informace v grafické podobě, tedy text, videa, obrázky atd. případně jako druhá nejzmiňovanější možnost bylo přehrávání ve zvukové podobě. Na druhou stranu, pokud by uživatel měl zrcadlo nějakým způsobem ovládat, jednoznačně převládala možnost slovních příkazů, s nimi už uživatelé v dnešní době přichází do styku například i u mobilních telefonů, a proto není překvapivé, že i zde vede. V kombinaci se zrcadlem jde o logickou volbu.

Z těchto poznatků je tedy možné vydedukovat, že ideální chytré zrcadlo by mělo zobrazovat informace, které získá z internetu případně jiných částí chytré domácnosti, nejlépe přímo na ploše zrcadla nebo na externím displeji a ovládání by bylo řešeno pomocí běžné řeči, tak jako například u virtuálních asistentů.



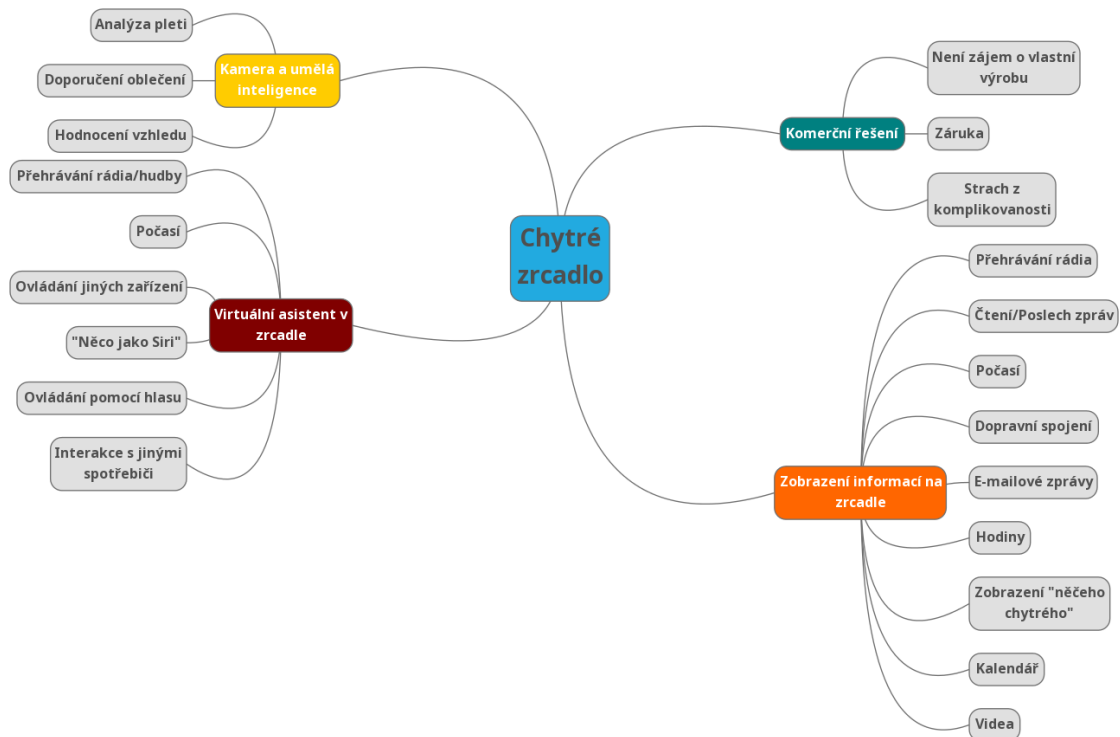
Obrázek 16 - Dotazník: Umístění a způsob pořízení (Vlastní zpracování)

Zbylé dvě otázky měly především doplňující roli. V první se zjišťovalo, kde by mělo být zrcadlo ideálně umístěno. Nejčastěji volenou možností byla koupelna, jelikož se ve většině případů jedná o místo, kde člověk u zrcadla tráví nejvíce času. V tomto případě nám to říká, že by například bylo vhodné zajistit odolnost zařízení vůči vlhkosti, která je v koupelně přítomna.

Poslední otázka zjišťovala, zda má tato práce reálnou šanci být přínosem. Cílem je dokázat, že potenciál je ve vlastní výrobě chytrých zařízení téměř na míru každého uživatele. Pokud by se ale uživatelé báli do výroby vlastní silou pustit a raději by dávali přednost komerčním řešením, bylo by zbytečné snažit se vytvářet nějaké podklady. Naštěstí z odpovědí je vidět, že většina respondentů by se v případě, že by měli k dispozici kvalitní dokumentaci a návod, nebála do výroby vlastní silou pustit.

4.1.1.1 Vybrané varianty

Na základě poznatků z předchozí kapitoly a konkrétních odpovědí z dotazníku je možné vytvořit takzvanou myšlenkovou mapu, ve které jsou představeny varianty, které splňují požadavky a představy respondentů a mezi kterými se bude následně vybírat podle definovaných kritérií. Pro vybrané varianty budou v poslední části práce realizovány návrhy implementace.



Obrázek 17 - Myšlenková mapa konceptů (Vlastní zpracování)

První z možností, v mapě nazvaná jako „*Kamera a umělá inteligence*“ je kombinace kvalitní kamery a obrazovky. Zde bylo cíleno na uživatele, kteří od zrcadla vyžadují analýzu obrazu. Efektu zrcadla by bylo docíleno tak, že kamera snímá uživatele a obraz je promítán na obrazovce. Tento obraz by byl softwarově analyzován a rozšířen o doplňující informace. Nezbytnou součástí tohoto systému je také relativně výkonný počítač, který by byl schopný obraz prezentovat, ale především zpracovávat pomocí patřičného softwaru, ten by byl založen na umělé inteligenci. Případně by výkonný počítač mohl být zastoupen některým cloudovým řešením. Například Microsoft se svojí cloud platformou Azure nabízí i API právě pro analýzu lidských obličejů s názvem Face API. (Microsoft, 2019)

Druhou variantou, „*Virtuální asistent v zrcadle*“, je zrcadlo relativně pasivní, které je pouze rozšířeno o hlasové a zvukové prvky pomocí integrace jednoho z virtuálních asistentů. Zde se bral ohled především na respondenty, kteří by chtěli interagovat pouze zvukově a také

by zrcadlo chtěli využít především jako ovladač pro jiná zařízení. Tato možnost není z technického pohledu nikterak náročná, uživatel by pouze vybral pro něj nejvhodnějšího hlasového asistenta v pohodě reproduktoru, které se aktuálně již prodávají i v „mini“ verzích, které by pro tento případ byly ideální a ten „ukryl“ za plochu běžného zrcadla. Při výběru asistenta by měl být brán zřetel především na jiná chytrá zařízení v domácnosti, aby mohl být využit plný potenciál virtuálního asistenta, kde by zrcadlo mohlo sloužit, kromě jiného, jako centrální prvek pro ovládání celého systému

Následující možností, která je v myšlenkové mapě uvedena jako „*Zobrazení informací na zrcadle*“, je využití obrazovky (PC monitoru) společně se zařízením, které bude produkovat zobrazované informace. V tomto případě by se mohlo jednat o některý z mikropočítačů. Monitor s mikropočítačem by byl umístěn za zrcadlem a v kombinaci s patřičným softwarem by na ploše zrcadla v určitých místech zobrazoval vybrané informace, které by mohl sbírat jak z internetu, tak například z jiných zařízení či senzorů v domácnosti. Tím by byl splněn požadavek, který prezentovalo nejvíce uživatelů, aby zrcadlo zobrazovalo informace na své ploše. Díky využití mikropočítače a vlastního softwaru by bylo možné rozšíření i o zvukové prvky. Nutností by byla úprava i samotného zrcadla, aby umožnilo skrz něj zobrazovat obraz monitoru.

Poslední variantu, pravděpodobněji nejjednodušší, je možné použít již některé komerční řešení. Těch ale překvapivě není mnoho, a proto je výběr velice omezený. Pro účely práce je vybráno chytré zrcadlo od americké společnosti Kohler s názvem „The Verdera“. Zrcadlo disponuje zabudovaným virtuálním asistentem Amazon Alexa a nabízí tak všechny možnosti, které tato technologie přináší. Veškeré informace jsou uživateli podávány zvukovou formou, stejně tak uživatel zrcadlo ovládá hlasem, případně je možné se připojit pomocí mobilní aplikace. (Kohler, 2019)

4.1.2 Definice hodnotících kritérií

Poté co byly definovány varianty, mezi kterými je možné se při plánování realizace chytrého zrcadla rozhodovat, je potřeba definovat také rozhodovací kritéria.

Pro účely práce byly zvoleny tyto kritéria:

- Pořizovací cena
- Náklady na užívání
- Rozšiřitelnost
- Znovu využitelnost
- Složitost

Pořizovací cena je jeden z nejdůležitějších aspektů, na které bude každý potenciální uživatel hledět. Pokud se pořizuje již hotové řešení, je potřeba počítat s určitou marží, kterou si výrobce účtuje. V tom vyniká veliká výhoda vlastních konstrukcí, jelikož je možné značně ušetřit právě vlastním výběrem a koupí jednotlivých komponent. Zároveň také uživatel kupuje jen to, co opravdu potřebuje a využije. Při hodnocení se zohledňují pouze komponenty, které přidávají chytré prvky, a ne případná cena zrcadla samotného nebo částí, které budou fungovat jako běžné pasivní zrcadlo.

Náklady na užívání navazují na předchozí část. Jelikož se jedná o aktivní zařízení, které ke svému chodu potřebuje elektrickou energii, je jasné, že tím budou vznikat i další náklady právě za energii, případně i za průběžnou údržbu. Je tedy důležité, aby tyto dva aspekty nepřevážily míru užítku, kterou uživatel bude mít. Cílem chytrých zařízení v domácnosti je nejen usnadnění každodenního života, ale zároveň i snížení nákladů. Proto by náklady na provoz chytrého zrcadla a jiné poplatky s užíváním spojené, měly být co nejnižší.

Další podstatnou vlastností chytrých zařízení je jejich **Rozšiřitelnost**, smyslem chytrých zařízení není užívat každé zařízení jako samostatný prvek, cílem je vytvořit funkční ekosystém, kde každé zařízení bude využívat pozitiv ostatních zařízení v jeho dosahu. Proto je důležité i v případě chytrého zrcadla hledět na to, zda je možné určitým způsobem rozšířit funkčnost zařízení samotného, případně ho propojit s jinými zařízeními, které již v domácnosti máme či plánujeme mít.

Znovu využitelnost, s tímto kritériem se dostáváme již mezi specifika vlastnoručně vyráběných zařízení. Velkou výhodou v tomto případě je situace, kdy můžeme komponenty, ze kterých je zařízení zkonstruováno, využít i v jiných projektech. Pokud bude zařízení, v tomto případě chytré zrcadlo, složeno z jednoúčelových komponent, dojde v případě, kdy uživatel například nebude chtít zařízení již využívat, k jejich znehodnocení. V opačném případě je možné zařízení pouze rozebrat a využít pro jiné projekty. Je proto důležité při plánování myslet i do budoucna.

Složitost, opět se jedná o vlastnost související se zařízeními, které si uživatel vytváří sám. Je důležité brát v potaz, zda je schopen konstrukci takového zařízení zvládnout například i amatér pouze s kvalitní dokumentací nebo zda je vyžadována odborná znalost, případně specializované vybavení jak pro konstrukci, tak pro následné využívání.

4.1.2.1 Určení vah kritérií

Pro určení vah jednotlivých kritérií byla využita Metfesselova alokace, kdy hodnotitelé rozdělí 100 bodů mezi jednotlivá kritéria, přičemž čím vyšší počet bodů udělí, tím je pro ně dané kritérium důležitější.

Součástí dotazníku, ve kterém se zjišťovaly podklady pro definici variant byla také část, kde respondenti měli ohodnotit kritéria představená v minulé kapitole. Kritéria jim byla představena ve stejné podobě jako zde v práci a bylo jim vysvětleno, aby je ohodnotili tak, jak by pro ně samotné byla jednotlivá kritéria důležitá při vlastním potenciálním zájmu o pořízení či realizaci chytrého zrcadla.

Výsledky hodnocení byly sloučeny do následujících tabulek. Byl vypočítán aritmetický průměr vah jednotlivých kritérií (K_i) a jeho normalizovaná hodnota (V_i), která je využita jako váha jednotlivých kritérií v následné vícekritériální analýze.

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10
Pořizovací cena	25	20	30	20	40	20	20	35	40	40
Náklady na užívání	25	20	25	20	40	20	20	25	15	30
Rozšiřitelnost	10	20	5	10	0	10	20	5	5	10
Znovu využitelnost	10	20	15	20	0	15	20	15	35	10
Složitost	30	20	25	30	20	35	20	20	5	10

Tabulka 1 - Hodnocení kritérií H1-10 (Vlastní zpracování)

	H11	H12	H13	H14	H15	H16	H17	H18	H19	H20	K_i	V_i
Pořizovací cena	20	50	33	20	25	30	45	35	20	31	29,95	0,2995
Náklady na užívání	20	50	28	20	25	30	35	30	20	29	26,35	0,2635
Rozšiřitelnost	10	0	8	15	15	25	0	10	20	13	10,55	0,1055
Znovu využitelnost	25	0	12	5	10	5	10	5	20	13	13,25	0,1325
Složitost	25	0	19	40	25	10	10	20	20	14	19,9	0,199

Tabulka 2 - Hodnocení kritérií H11-20 (Vlastní zpracování)

4.1.2.2 Vícekritériální analýza

Na základě poznatků z předchozích kapitol je možné pomocí vícekritériální analýzy zvolit nejvhodnější variantu pro konstrukci chytrého zrcadla. Jednotlivé varianty byly v daných kritériích ohodnoceny body 1 až 4, podle toho, jak dobře se varianta v dané kategorii umísťuje, přičemž čím vyšší bodové ohodnocení, tím lépe. Hodnocení jednotlivých možností je vidět

v tabulce níže. Poslední sloupec udává celkové hodnocení pro danou variantu, kde jsou pro jednotlivá kritéria započítané i jejich váhy.

	Pořizovací cena	Náklady na užívání	Rozšiřitelnost	Znovu využitelnost	Složitost	Vážené body
Kamera + umělá inteligence	2	1	3	3	1	1,7755
Zrcadlo + chytrý asistent	4	4	2	2	3	3,325
Monitor + mikropočítač	3	3	4	4	2	3,039
The Verdera - Kohler	1	2	2	1	4	1,966

Tabulka 3 - Vícekriteriální hodnocení variant (Vlastní zpracování)

U kritéria **pořizovací cena** se jednoznačně nejlépe umísťuje varianta, kde je pouze integrován hlasový asistent. Pokud vezmeme například reproduktor Google Home Mini, který využívá Google Asistenta, na internetovém obchodě Alza.cz, jeho cena je aktuálně (Leden 2019) 1 189 Kč, i cena reproduktoru od Amazonu, Echo Dot se pohybuje podobně, konkrétně 1 280 Kč. Toto je jediná investice, která je potřeba aby zrcadla získalo „chytré“ prvky. To už poté stačí jen doplnit zrcadlem s hlubším rámem a celé řešení je kompletní. Na druhém místě je varianta, která kombinuje mikropočítač a monitor, zde je zvoleno Raspberry Pi, jelikož se jedná o nejpobulárnější variantu v dané kategorii. Na typu monitoru v tomto případě příliš nezáleží, zobrazované informace budou pravděpodobně v jednoduché formě a není tedy nutné využívat drahý monitor například s vyšším rozlišením než Full HD. Zde je pořizovací cena tedy 1 081 Kč za nejnovější verzi Raspberry Pi a 2 490 Kč za nejlevnější, 24palcový monitor aktuálně prodávaný na Alza.cz, jedná se o model Acer G246HYLbd. Dohromady 3 571 Kč. Cena poté může sice růst při nákupu doplňkových komponent, ale to jsou již pouze volitelné části, základní funkční řešení vyžaduje především tyto dvě zmíněné komponenty. Předposlední varianta, tedy kombinace kamery, která snímá uživatele, obraz promítá na monitor skrze počítač a je softwarově obohacen o další prvky vyžaduje zakoupení minimálně kvalitní kamery a monitoru, jako vzorek je vybrána kamera Logitech Pro Stream Webcam C922 PRO s cenou 2 571 Kč a monitor Philips 243V7QJABF s cenou 2 555 Kč. Již tyto dvě komponenty cenou přesahují předchozí řešení, a to není započten ještě počítač nebo předplatné za cloudové řešení, které by se staralo o samotný běh softwaru. Jako nejdražší varianta je jednoznačně řešení od firmy Kohler, s pořizovací cenou 33 159 Kč, které se významně odlišuje od předchozích řešení. (Alza, 2019) (Kohler, 2019)

V **nákladech na užívání** opět vede řešení s hlasovým asistentem. Pokud zvolíme variantu s Google Home Mini, který v má průměrnou spotřebu pouhé 2 watty (Snell, 2017) je roční poplatek za elektřinu u tohoto řešení při průměrné ceně 4,1 Kč/kWh přibližně 72 Kč. Na druhém místě je varianta mikropočítač + monitor. Pokud je počítáno se stejnými komponentami jako u prvního kritéria, tak průměrná spotřeba Raspberry Pi se pohybuje mezi 2 až 5 watty (Geerling, 2019), tedy 72 až 180 Kč ročně. K tomu je ještě ale potřeba připočítat spotřebu monitoru, ten je sice možné softwarově uspávat, aby jeho spotřeba byla co nejmenší, ale i tak je nutné tuto hodnotu započítat. Dle technické specifikace monitoru je udávaná spotřeba 0,5 až 31 wattů (Acer, 2019), tedy maximálně 1 077 Kč, pokud by celý rok běžel na plný výkon, k čemuž by rozhodně nemělo dojít a spotřeba by se měla pohybovat o poznání níže. Na třetím místě se umístilo zrcadlo The Verdera, u kterého nebylo možné dohledat přesné hodnoty o spotřebě (což je již samo o sobě negativum), ale orientačně je potřeba počítat se samotným odběrem zrcadlo, které je vybaveno silným osvětlením a k tomu ještě přidat zařízení, které poskytuje hlasového asistenta Alexa, u toho se orientační spotřeba pohybuje kolem 4 wattů (Snell, 2017). Poslední místo zaujímá řešení s kamerou a monitorem, zde je opět potřeba počítat s odběrem monitoru, který pravděpodobně bude na plný výkon fungovat častěji než u druhé varianty. Konkrétně pro vybraný model se dle technické specifikace spotřeba pohybuje průměrně kolem 14 wattů (Philips, 2019). Největší spotřebu ale v tomto případě bude mít zařízení, které toto řešení bude pohánět, výkon Raspberry Pi pravděpodobně nebude dostatečný a bude potřeba výkonnější zařízení a s ním roste i samotná spotřeba. Pro příklad je možné vzít například Intel NUC 8i3BEH, jehož spotřeba je průměrně 30 wattů (Alza, 2019), touto kombinací se varianta řadí na poslední místo žebříčku. Z celkového pohledu je vidět, že ať zvolíme jakoukoliv variantu, neměla by spotřeba celého řešení být nijak závratně vysoká.

Na poli **rozšiřitelnosti** vyhrává poměrně jednoznačně řešení s mikropočítačem. Pokud se opět jako příklad zvolí Raspberry Pi, je možné jeho funkčnost rozšířit jak po stránce softwarové, kde nejsou kladena téměř žádná omezení na zvolený operační systém či programovací jazyk, omezení je pouze na straně výkonu a schopností uživatele, ale i po hardwarové, kdy je možné připojit další zařízení jak pomocí Bluetooth či USB, ale především pomocí GPIO pinů, pro které existuje nespočet různých senzorů (Raspberry Pi Tutorials, 2019). Na druhém místě je řešení s kamerou, které je možné rozšířit především po softwarové stránce, opět záleží na schopnostech uživatele a dostupném výkonu, v tomto případě ale bude vývoj softwaru pravděpodobně o něco komplikovanější. Samozřejmě je možné rozšířit i samotný počítač o další komponenty a ty využít. Na třetím místě se shodně umístila zbylá dvě zařízení,

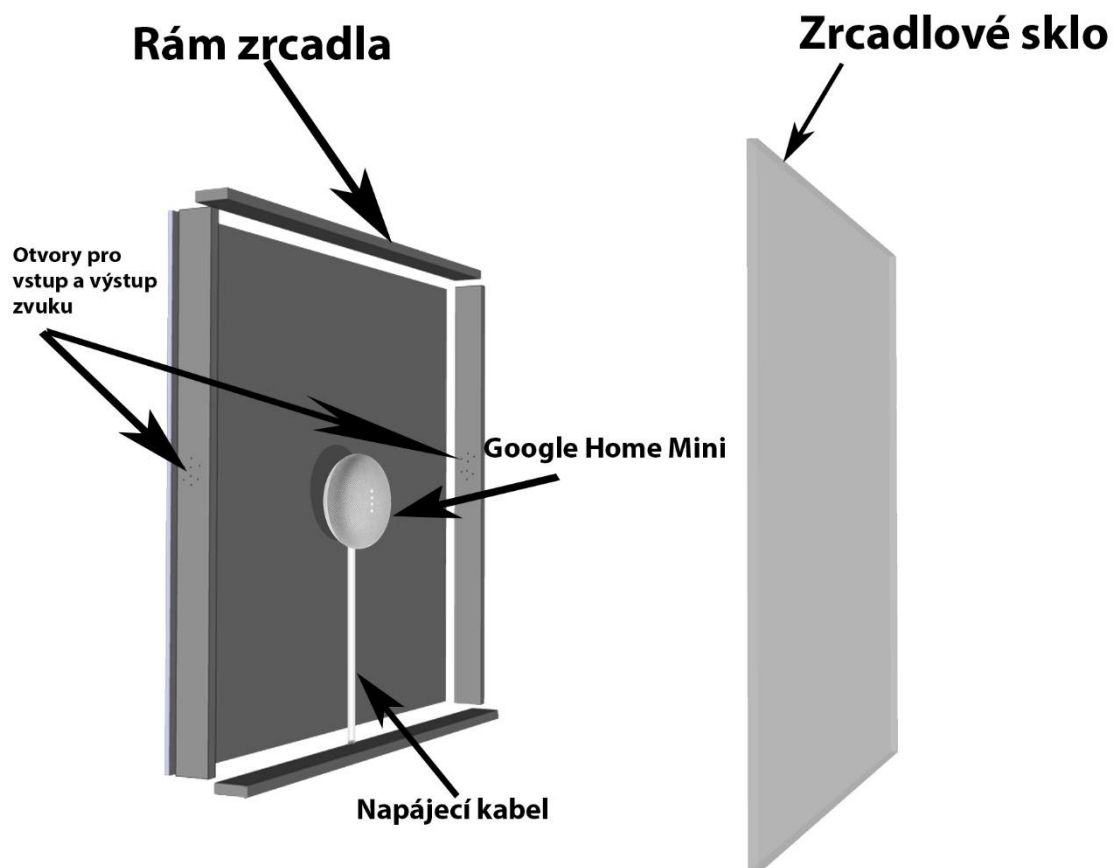
jelikož obě využívají služeb chytrého asistenta a možnosti pro rozšíření jsou tedy omezena právě schopnostmi daného asistenta.

Znovu využitelnosti lze opět nejlépe dosáhnout s Raspberry Pi a monitorem. V případě že již uživatel nebude chtít využívat chytré zrcadlo, nabízí se pro Raspberry Pi nepřehledné množství využití ať už v domácnosti nebo mimo ni, od jednoduchých řešení pro kompaktní meteostanici, zábavní systém k televizi postavený na aplikaci Kodi, až po pokročilé řízení zahradního skleníku (Upton & Halfacree, 2013). Monitor, který tvoří druhou část můžeme kdykoliv využít například u běžného počítače. Na druhém místě je systém složený z kamery, monitoru a počítače. Každou z těchto komponent je možné používat i samostatně, takže pokud uživatel zruší chytré zrcadlo, tyto komponenty rozhodně neprijdou vniveč. Třetí místo obsazuje zrcadlo s chytrým reproduktorem, v tomto případě můžeme sice reproduktor ven vyjmout, ale i venku poslouží ke stejným účelům jako uvnitř zrcadla. Poslední místo patří řešení od firmy Kohler, zde se jedná o uzavřené zařízení, které výrobce prodává tak, aby se využívalo jako zrcadlo a ne jinak, pokud by se tedy uživatel pokusil ho rozebrat a části využít jinak, ztratil by tím zároveň i záruku.

Posledním důležitým kritériem je **složitost**, ta bude mnoho uživatelů motivovat nebo naopak odrazovat od možnosti pokusit se zařízení vyrobit vlastními silami. Na prvním místě je zde logicky hotové řešení The Verdera, které pouze stačí zakoupit, nainstalovat na místě užití a používat. Druhé místo zaujímá řešení s chytrým asistentem, kde je opět potřeba jen zakoupit zařízení (reproduktor) podporující vybraného asistenta, a to ukrýt pravděpodobně za zrcadlo. Jako předposlední je zrcadlo s mikropočítačem, zde je již potřeba znalosti jak po hardwarové stránce, tak po softwarové, ale pokud budeme opět brát jako příklad Raspberry Pi, tak na internetu je již nezměrné množství návodů, jak vyvíjet software pro toto zařízení téměř ve všech programovacích jazycích a stejně tak existují i návody, jak připojit jiná zařízení k Raspberry. Například na populárním portále pro vývojáře StackExchange je 1680 stran dotazů a rad věnující se pouze Raspberry Pi (StackExchange, 2019). Jako nejsložitější byla zvolena varianta s kamerou, zde je největší komplikací extrémně složitý software využívající umělé inteligence, jak již bylo zmíněno, existují dostupné API, které pracují s podobnými úlohami, ale i tak by bylo potřeba software na míru upravit. Jednou ze značně pokročilých společností, které se touto problematikou zabývají je Systemnegar Saina s jejich softwarem DeepFace, který je zaměřen právě na analýzu lidských tváří. Jejich software ale není dostupný zdarma a v případě použití této technologie by bylo potřeba počítat s nárůstem pořizovacích nákladů (Systemnegar Saina, 2019).

Z výsledků, které nám analýza poskytla je vidět, že na celkovém prvním místě se s nepatrným odstupem umístilo řešení chytrého zrcadla s integrovaným virtuálním asistentem. Jedná se o konstrukčně jednoduché, levné a úsporné řešení, které plní účel chytrého zrcadla. V následující části práce bude představen technický návrh, jak toto řešení realizovat včetně cenové kalkulace, bude se jednat o variantu A. Zároveň ale bude představen i návrh realizace varianty, která se umístila jako druhá, tedy chytré zrcadlo využívající mikropočítače pro zobrazení informací na jeho ploše, jelikož na ní je možné lépe demonstrovat technologické možnosti, které jsou uživatelům dostupné při snaze o realizaci vlastních chytrých zařízení. Toto bude varianta B.

4.2 Technický popis řešení varianty A



Obrázek 18 - Návrh konstrukce pro variantu A (Vlastní zpracování)

Na obrázku výše je možné vidět návrh konstrukce řešení s využitím zmiňovaného Google Home Mini, tento virtuální asistent je zvolen především díky kvalitě jeho umělé inteligence při zpracování pokládaných otázek. Zároveň je také schopen spolupracovat se zařízeními jiných výrobců jako například Wemo, TP-LINK či Samsung Smart Things. (Google, 2019)

Po realizaci konstrukce je potřeba pouze provést úvodní nastavení (připojení k WiFi atd.) pomocí aplikace Google Home a celé zařízení je připraveno k použití. Následně už záleží pouze na preferencích uživatele, k čemu bude chytrého asistenta využívat a jaké možnosti naskýtá daná domácnost.

Jedná se o opravdu jednoduchý, ale funkční a úsporný koncept, který by měl být schopen sestavit každý potencionální uživatel, jelikož nevyžaduje žádné specifické znalosti, nutnost programování nebo speciální nástroje.

Orientační celkové náklady na dané řešení jsou vidět v tabulce níže. Jednotlivé části mohou být samozřejmě odlišné u každého uživatele, především zvolený asistent se může lišit. V případě že již například někdo používá další zařízení od firmy Apple bude pravděpodobně chtít do zrcadla integrovat hlasového asistenta Siri, který bude s danými zařízeními komunikovat a pracovat lépe.

Předmět	Cena s DPH	Zdroj
Google Home Mini	1 120 Kč	Alza.cz
Zrcadlo s hlubokým rámem	579 Kč	OBI
Celkem	1 699 Kč	

Tabulka 4 - Cena varianty A (Vlastní zpracování)

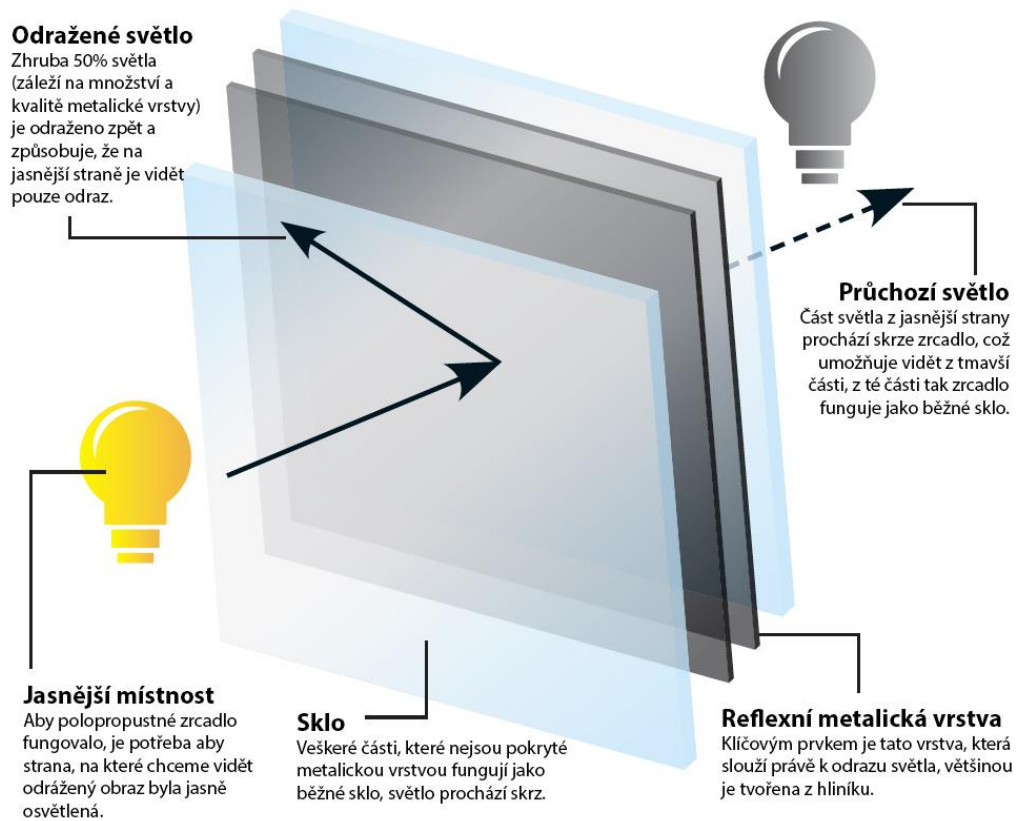
4.3 Technický popis řešení varianty B

Na základě předchozí kapitoly se řešení bude skládat především z mikropočítače, který bude vše řídit. Zvoleno bylo Raspberry Pi v nejnovější verzi 3 B+, které již bylo v předchozí části zmíněno. Doplnuje ho běžný 24palcový monitor s Full HD rozlišením, konkrétně model GW2470HE od společnosti BenQ. K GPIO pinům Raspberry Pi budou ještě připojeny sensory DHT11, DS18B20 a HC-SR501, aby byla dokázána rozšiřitelnost tohoto konceptu.

Aby zmíněná zařízení mohla zobrazovat informace na ploše zrcadla, je využita technologie dvoucestného zrcadla, známého například z policejních výslechových místností. Z jedné strany plocha funguje jako běžné zrcadlo, z druhé strany je ale plocha průhledná jako běžné sklo. Tohoto efektu je dosaženo jednoduchou prací se světlem.

Samotný princip zrcadla je založen na polopropustné fólii, která je nalepená na skleněnou desku. Polopropustná zrcadlová fólie funguje tak, že z jedné strany, která je světlejší, odráží světlo, tudíž funguje jako zrcadlo. To ale vyžaduje, aby druhá strana, za zrcadlem byla,

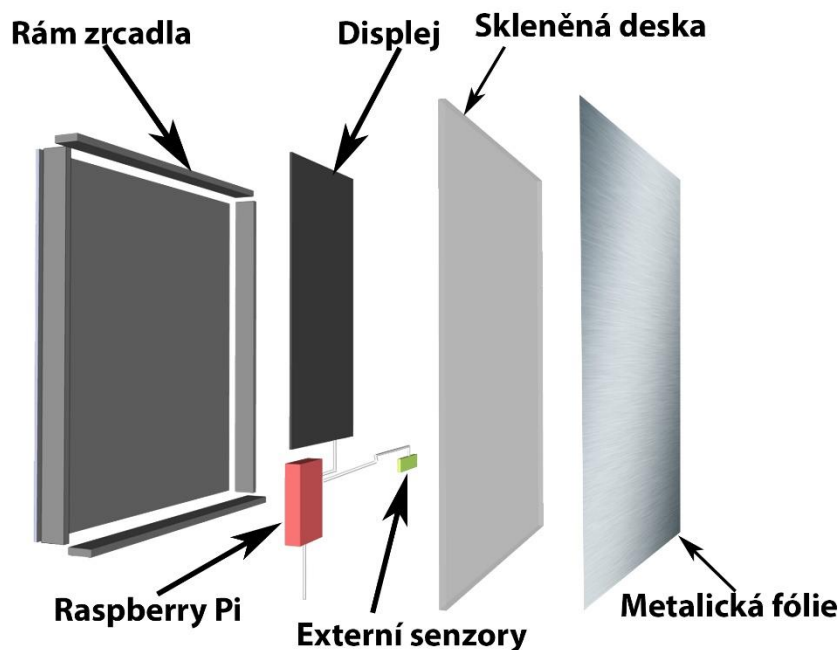
pokud možno co nejtmavší, pokud bude část za zrcadlem také osvětlena, efekt zrcadla se ztratí a sklo bude průhledné (Stavební sklo s.r.o., 2019).



Obrázek 19 - Princip fungování polopropustného skla (How it works, 2015)

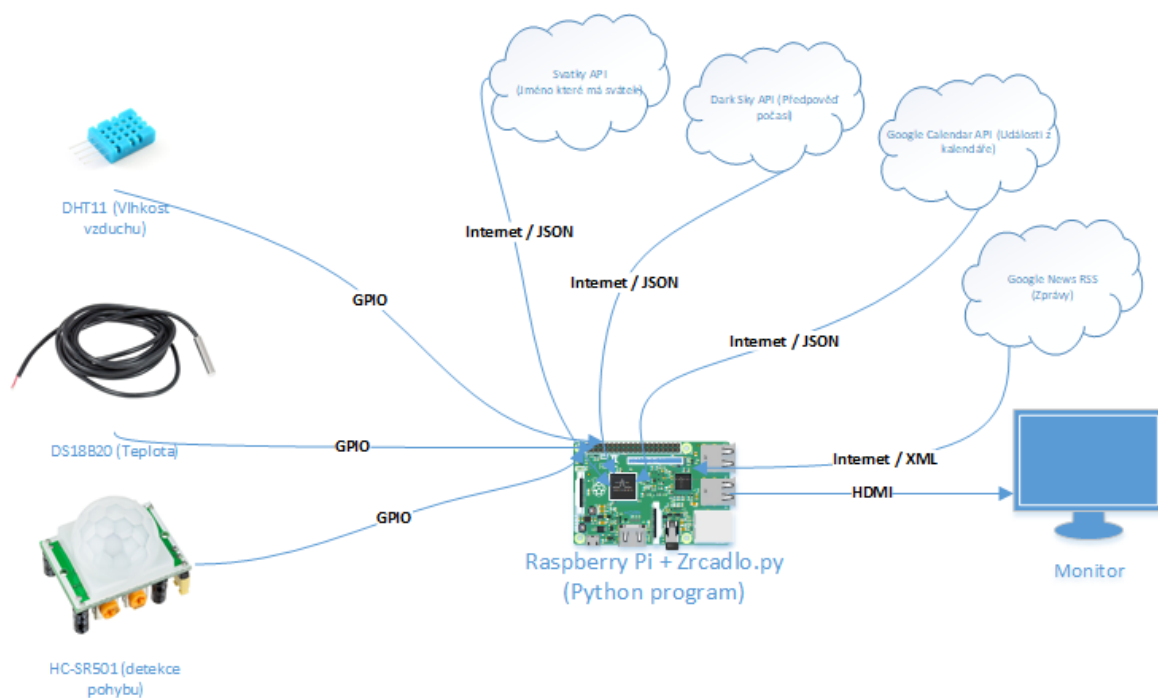
Tohoto efektu využijeme právě tím, že za zrcadlem bude instalován monitor se spuštěným programem, který bude mít z větší části tmavé pozadí, tedy bude fungovat jako tmavá strana a zrcadlo bude reflektovat reálný obraz tak jak má. Informace, které ale budeme chtít zobrazit, budou prezentovány jasným bílým fontem, tudíž budou prosvěcovat skrze sklo a zrcadlo tak bude mít společný efekt, kdy většina plochy bude odrážet světlo, a zároveň i obraz, ale v určitých místech budou vidět zobrazované informace z monitoru. Pro lepší znázornění fungování polopropustného zrcadla slouží obrázek výše.

Kompletní koncept zrcadla bude tedy sestávat ze skleněné polopropustné desky, která bude zasazená do dřevěného rámu, to bude vytvářet dojem běžného zrcadla, které uživatel může umístit kamkoliv po bytě. Uvnitř rámu bude dále skryt displej (PC monitor), který bude sloužit jako zobrazovací zařízení pro informace, které do něj bude zasílat Raspberry Pi. Deska mikropočítače bude také skryta uvnitř rámu. Ven bude vyveden pouze napájecí kabel pro Raspberry Pi a také doplňující senzory připojené na GPIO piny. Grafické zobrazení návrhu konstrukce je k vidění níže.



Obrázek 20 - Návrh konstrukce pro variantu B (Vlastní zpracování)

Předtím, než je možné se do samotné realizace pustit, je vhodné si udělat i návrh toho, co bude zrcadlo zobrazovat a jak bude tyto informace získávat. Pro tento účel skvěle poslouží aplikace od společnosti Microsoft s názvem Visio. Na obrázku dále je vidět návrh architektury tohoto projektu. Program běžící na Raspberry Pi tedy bude komunikovat jak s hardwarovými senzory, tak i s veřejně dostupnými API pomocí internetového připojení. Grafický výstup poté bude přenášen do monitoru.



Obrázek 21 - Architektura varianty B (Vlastní zpracování)

Celkové náklady budou společně s nároky na uživatele o něco vyšší než u varianty A, v tomto případě je již vyžadována určitá znalost jak s prací s hardwarem, tak i základy programování v některém z programovacích jazyků, které Raspberry Pi podporuje. Možnosti, které tedy tato varianta poskytuje jsou omezeny právě znalostmi uživatele, ale i tato skutečnost je kompenzována především množstvím různých návodů či diskuzních fór, zabývajících se podobnou problematikou, dostupných na internetu. Pořizovací náklady na koncept, který je dále v práci prezentován, jsou k vidění v tabulce níže.

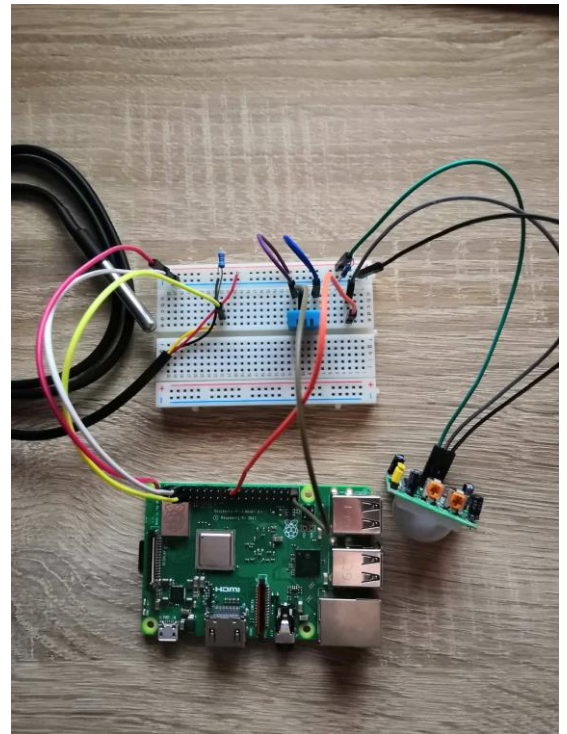
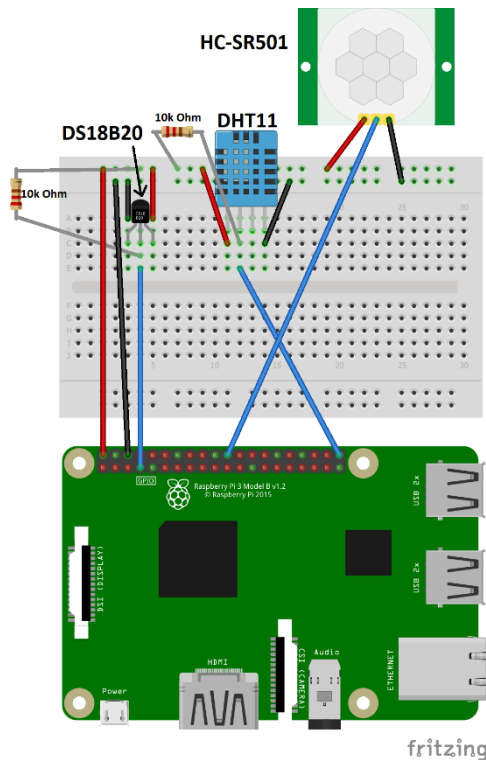
Předmět	Cena s DPH	Zdroj
Raspberry Pi 3B+	1 120 Kč	Alza.cz
MicroSD karta Kingston	99 Kč	Datart
Digitální teploměr a vlhkoměr DHT11	69 Kč	GME
Digitální teplotní čidlo s DS18B20	99 Kč	GME
PIR modul HC-SR501	110 Kč	GME
Skleněná deska	100 Kč	Dantik
Polopropustná zrcadlová fólie	349 Kč	Hornbach
Dřevěný rám	150 Kč	OBI
Celkem	2 096 Kč	

Tabulka 5 - Cena varianty B (Vlastní zpracování)

4.4 Hardwarová realizace

Z hardwarové stránky věci je nejsložitější částí správné zapojení jednotlivých čidel k GPIO pinům Raspberry PI, zbytek obnáší pouze připojení monitoru pomocí HDMI kabelu a připojení napájení jak pro monitor, tak pro Raspberry Pi.

Před samotnou realizací je vhodné si předem vytvořit grafický podklad, ze kterého je možné vycházet při reálném zapojení komponent. Pro vytvoření schématu zapojení čidel k GPIO pinům skvěle slouží program Fritzing (Fritzing, 2019), ten je možný využít pro vytváření schémat zapojení elektronických zařízení nejen v tomto případě, ale například i pro Arduino projekty a obsahuje rozsáhlou databázi čidel, elektronických desek a součástek. Výsledkem je schéma níže. Ve schématu je vidět zapojení s pomocí nepájivého pole. Jedná se o opakovaně použitelnou pomůcku k navrhování prototypů elektrických obvodů bez letování součástek. Nepájivé pole je zhotoveno z plastové izolační desky s otvory a uchycení součástek a jejich propojení zajišťují kontaktní hřebínky s pružinami zasunuté zesponu. Vedle je možné vidět reálné zapojení, to se mírně odlišuje použitými konektory na nepájivém poli, ale funkčnost zůstává zachována stejná jako u návrhu. Je vidět zapojení jednotlivých čidel, kde na grafickém znázornění modrý kabel znázorňuje datový přes, černý zem a červený napájení. U čidel DHT11 a DS18B20 je možné vidět i zapojení 10k Ohmový odpor, který propojuje datový a napájecí vodič.

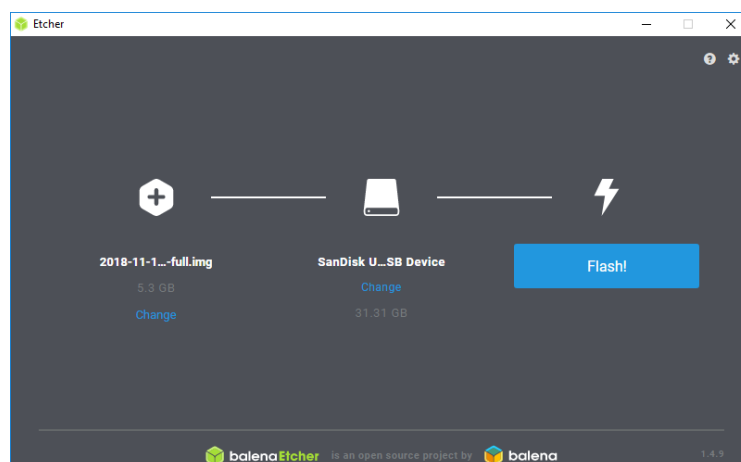


Obrázek 22 - Návrh a reálné zapojení čidel v nepájivém poli (Vlastní zpracování)

4.5 Softwarová realizace

4.5.1 Instalace a konfigurace Raspbian OS

Jako operační systém, běžící na Raspberry Pi je vybrán Raspbian ve verzi z listopadu 2018, jedná se o operační systém, který je vyvíjen samotnou Raspberry Foundation a jeho využití na Raspberry Pi je jedno z nejjednodušších. Instalaci je možné provést pomocí dvou způsobů. Každý z těchto způsobů vyžaduje PC, kde je možné nainstalovat potřebný software a připojit microSD kartu.

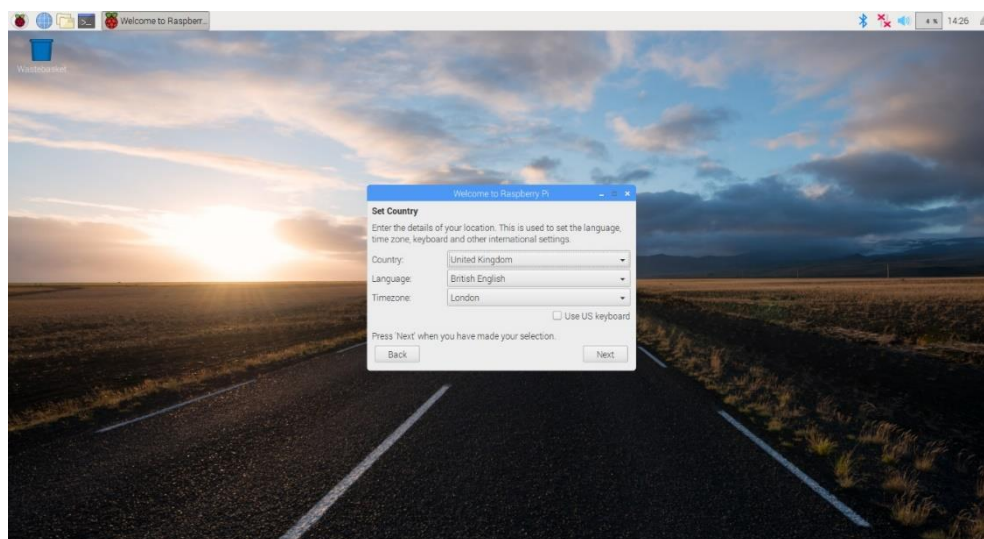


Obrázek 23 - Instalace Raspbian OS (Vlastní zpracování)

Prvním je využití NOOBS, což je software vytvořený především pro začínající uživatele Raspberry Pi, který je instalací bezpečně provede.

Druhou možností je stažení samotného ISO souboru Raspbianu, který se poté rozbalí na microSD kartu, která slouží jako úložiště v Raspberry Pi, pomocí programu Etcher. Zde je potřeba pouze vybrat .iso soubor obrazu Raspbianu, který je možný stáhnout z oficiálních webových stránek <https://www.raspberrypi.org/downloads/raspbian/> a microSD kartu, na kterou chceme systém nahrát. Prostředí zmíněného programu je vidět na obrázku výše.

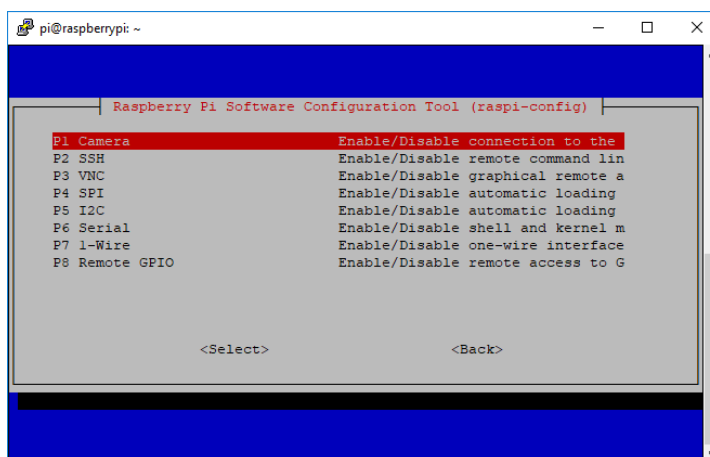
Po připravení microSD karty s operačním systémem je již možné ji připojit do Raspberry Pi, připojit k němu monitor nebo jiné externí zobrazovací zařízení a připojit napájení, zařízení se ihned automaticky spustí a zahájí načítání operačního systému.



Obrázek 24 - Úvodní obrazovka Raspbian OS (Vlastní zpracování)

Po prvním spuštění je zobrazena úvodní obrazovka, která uživatele provede základním nastavením, kde si může například změnit lokalizaci systému a vybrat připojení k síti WiFi. Poté je již zařízení plně připraveno k dalšímu použití.

Následně je potřeba již specifické nastavení pro případ tohoto projektu. Pomocí terminálového příkazu „*sudo raspi-config*“ se zobrazí speciální nastavení pro Raspberry Pi, zde je potřeba v záložce „*Interfacing options*“ povolit jednotlivé komunikační protokoly, využívané na GPIO pinech a zároveň je vhodné povolit i SSH přístup pro vzdálenou správu zařízení pomocí terminálu.



Obrázek 25 - Konfigurace Raspberry Pi (Vlastní zpracování)

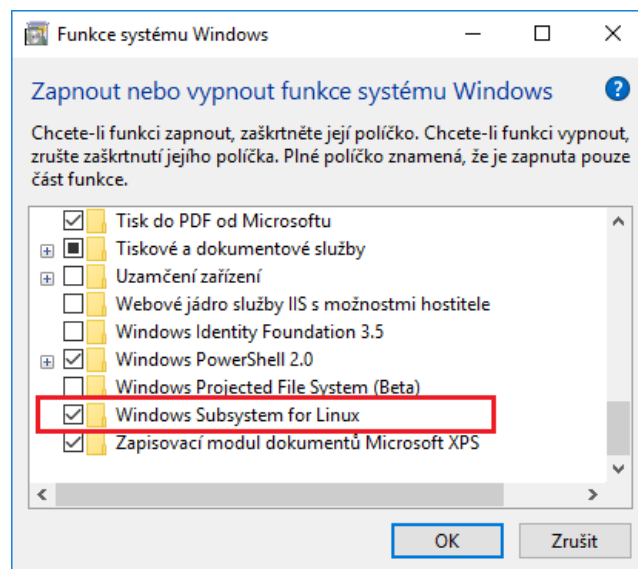
Poslední částí, kterou je potřeba v systému připravit, je doinstalování veškerého podpůrného softwaru, který bude běžící aplikace vyžadovat. Především se jedná o Python moduly, které jsou následně využívány hlavní aplikací.

Jako první se doinstalují moduly, které využívá Google API pro přístup ke kalendáři pomocí příkazu v terminálu „*sudo pip install --upgrade google-api-python-client google-auth-httplib2 google-auth-oauthlib*“ následně ještě „*sudo pip install requests feedparser Pillow*“, funkce těchto modulů jsou blíže představeny v kapitole věnující se aplikaci, která na Raspberry Pi poběží. Poslední nutností je stažení knihoven pro práci se senzorem DHT11, ty je možné získat z GIT repositáře pomocí příkazu „*git clone https://github.com/adafruit/Adafruit_Python_DHT.git*“. Jedná se o knihovny vytvořené přímo firmou Adafruit, která stojí i za výrobou samotného senzoru, pro práci s DHT11 na Raspberry Pi. Ve staženém repositáři je potřeba již jen spustit soubor „*setup.py*“, který se postará o veškerou instalaci. Toho dosáhneme pomocí příkazu „*sudo python setup.py install*“. Tímto je již zařízení plně připravené a stačí pouze spustit program, který bude prezentovat informace na zrcadle.

4.5.2 Vývojové prostředí

Kvůli omezením, které by se mohly vyskytnout, kdyby probíhal vývoj samotné aplikace přímo na Raspberry Pi, kdy by se musely připojit periferie přímo k zařízení, nebo by se k němu přistupovalo pomocí vzdáleného přístupu, byla využita relativně nová funkce Windows 10 a to konkrétně „*Windows subsystem for Linux*“, která je blíže popisována v teoretické části práce. Ta umožní simulovat Linuxové prostředí přímo v operačním systému Windows. Autor tak může vyvíjet a testovat v Linuxovém prostředí, které simuluje Raspberry Pi přímo na svém hlavním pracovním zařízení, kde reálně běží systém Windows 10.

Pro zprovoznění „*Windows subsystem for Linux*“ je nejprve potřeba tuto možnost aktivovat v Ovládacích panelech Windows 10, kde se nachází volba „*Zapnout nebo vypnout funkce systému Windows*“

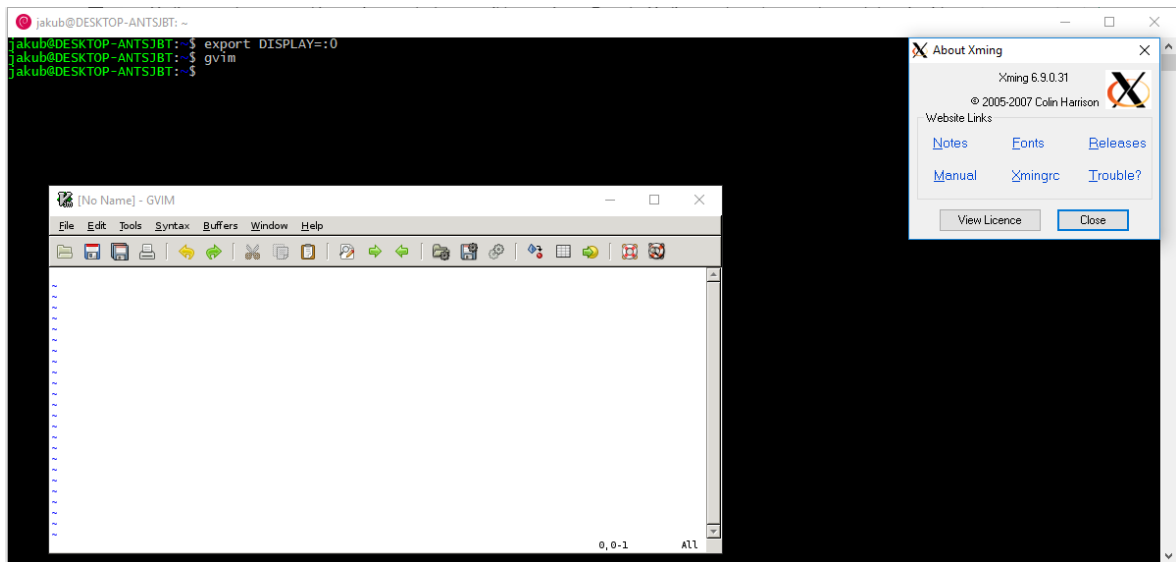


Obrázek 26 - Zapnutí "Windows Subsystem for Linux" (Vlastní zpracování)

Po následujícím restartu je již možné z Microsoft Store stahovat a instalovat různé, volně dostupné distribuce systému Linux. V případě této práce byl zvolen systém Debian GNU/Linux, jelikož je z něj odvozen i Raspbian, který běží na mikropočítači Raspberry Pi.

„*Windows subsystem for Linux*“ Je navržen především pro práci s terminálem, není tedy možné v základu zobrazovat aplikace s vlastním grafickým rozhraním. Aplikace v této práci ale bude okenní aplikací a je tedy potřeba toto omezení obejít, a to pomocí takzvaného „X serveru“. Jeho účelem bylo původně zobrazování grafických rozhraní při vzdáleném přístupu k Linuxovým systémům, ale stejně dobře poslouží i v tomto případě.

V systému Windows je potřeba nainstalovat aplikaci, která bude fungovat jako X server, v tomto případě se jedná o „Xming“. Po jeho spuštění už pouze nastavíme v Linuxovém prostředí proměnnou DISPLAY, která systému definuje, kam se mají aplikace s grafickým rozhraním zobrazovat, pomocí příkazu „*export DISPLAY=:0*“ zvolíme právě X server. Nyní je již možné spouštět i aplikace s grafickým rozhraním.



Obrázek 27 - Využití programu Xming (Vlastní zpracování)

Následujícím krokem je již instalace prostředků skriptovacího jazyka Python ve verzi 3, ve kterém bude vyvíjena samotná aplikace. Pomocí příkazu „*sudo apt install python3 python3-pip*“ se nainstaluje vše, co bude pro začátek vývoje potřeba. Pokud rozebereme jednotlivé parametry, tak „*python3*“ obsahuje veškeré náležitosti pro jazyk Python, „*python3-pip*“ slouží jako instalační program pro další doplňky jazyka Python.

4.5.3 zrcadlo.py

Hlavním programem, který na Raspberry Pi poběží, bude chytre_zrcadlo.py, jedná se o program napsaný v jazyce Python a jeho nejdůležitějším základem bude využití modulu (knihovny) Tkinter, který umožňuje vytvářet grafická okna. Tento program bude prezentovat vybrané informace na obrazovce, připojené k Raspberry Pi a bude tak tím hlavním prvkem, který rozšíří běžné zrcadlo o chytré funkce.

Program bude rozdělen do několika hlavních částí, které budou každá plnit jinou funkci, konkrétně prezentovat jinou informaci na displeji. Zbylá část programu se bude starat na pozadí o získávání aktuálních dat. V tomto prototypovém případě jsou názvy proměnných a jiných prvků ve zdrojovém kódu většinou pojmenovány, aby bylo výstižné, co prezentují. V reálném využití by ale bylo vhodné využít názvy stručnější, aby byl kód co nejjednodušší.

Hlavní části programu budou konkrétně tyto:

- **Hodiny** – Zobrazení času ve 24hodinovém formátu včetně vteřin, zároveň bude zobrazeno dnešní datum a patřičný den v týdnu.
- **Svátek** – Ihned pod hodinami bude zobrazeno jméno, které daný den slaví svátek v ČR. Zobrazení bude ve formátu „Dnes má svátek ...“.
- **Zprávy** – Vypsání titulků zpráv z agregátoru Google Zprávy, uživatel si bude moci konfigurovat množství a jazyk (lokalitu) pro který budou zprávy získávány.
- **Počasí** – Tato část bude ukazovat aktuální teplotu a vlhkost vzduchu. Tyto data bude získávat z hardwarových čidel přímo připojených k RaspberryPi. Druhou částí bude i krátkodobá předpověď na následující den, která bude získávána z veřejné Dark Sky API.
- **Kalendář** – Výpis konfigurovatelného množství nejbližších akcí z uživatelova Google kalendáře.
- **Detekce pohybu** – Z důvodu úspory elektrické energie se bude díky využití pohybové senzoru HC-SR501 přepínat monitor do úsporného režimu v případě, že se před zrcadlem nebude nikdo nacházet. Monitor je nejvíce energeticky náročnou částí konceptu, a proto je snaha o úsporu směřována především na tuto komponentu.

4.5.3.1 Konfigurační soubor config.ini

Aby uživatel nemusel zasahovat přímo do zdrojového kódu (což je vhodné i z důvodu bezpečnosti, aby nedošlo k nevědomé úpravě kódu) je vytvořen konfigurační soubor, ve kterém jsou uvedeny veškeré parametry, které může uživatel ovlivnit.

Ty jsou poté již programem načítány a dále se s nimi pracuje. Aktuálně uživatel musí vyplnit hodnoty vlastní vygenerovaných API klíčů pro služby DarkSky a ipstack a dále může ovlivnit jazyk zobrazovaných zpráv a jejich počet.

```
config.ini
1 [weather]
2 darksky_api = 04dc69d2ac7ebf799b084f8e3298c674
3 ip_stack_api = 32c993f3828bcf9b9b3d24227c054e04
4
5 [news]
6 jazyk_zprav = cs
7 pocet_zprav = 5
```

Obrázek 28 - Konfigurační soubor

4.5.3.2 Inicializace a načtení modulů

V úvodu samotného souboru je v prvních dvou řádcích definován interpret jazyka python a kódování, se kterým se v programu pracuje. Na následujících řádcích již jsou postupně importovány veškeré moduly, se kterými následně pracuje. Python sám o sobě v základu nepodporuje velké množství funkcí, které každému mohou připadat jako základní, například funkce pro práci s časem. Právě importováním odpovídajících modulů se k daným funkcím dostaneme. Na tom je založena jednoduchost Pythonu, uživatel pracuje opravdu jen s tím, co jeho program potřebuje.

Funkce jednotlivých modulů je popsána komentáři přímo v kódu a konkrétní využití funkcí z nich je popsáno v patřičných částech kódu programu.

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

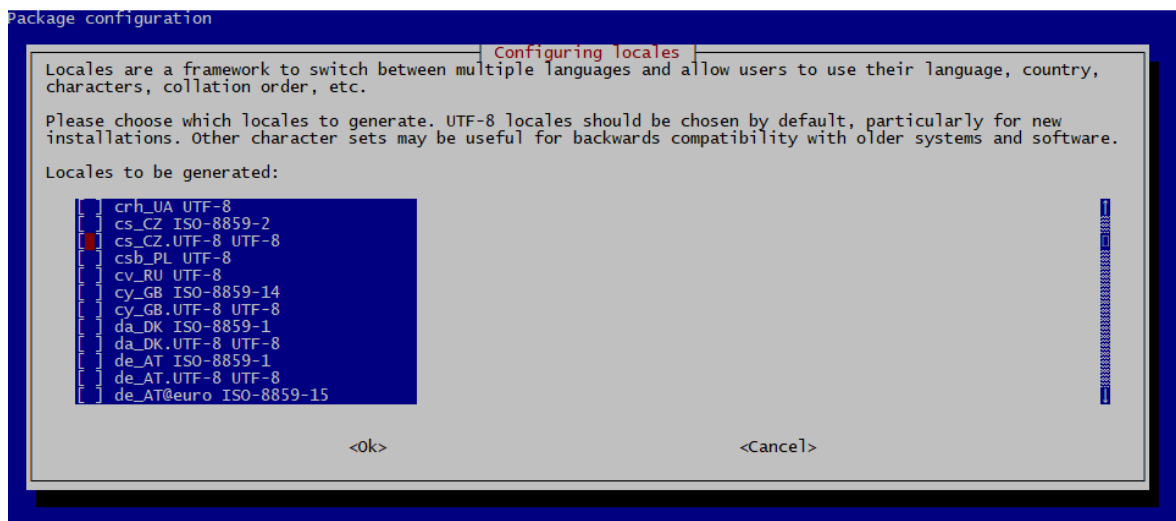
#ModulTkinter pro možnost tvorby okenních aplikací
from Tkinter import *
#Moduly Google API pro práci s kalendářem
from googleapiclient.discovery import build
from google_auth_oauthlib.flow import InstalledAppFlow
from google.auth.transport.requests import Request
import pickle
import os
#Modul pro práci s mezinárodními standardy, v případě tohoto programu se jedná o české dny a měsíce
import locale
#Modul pro práci s vlákny
import threading
#Modul umožňující práci s časem
import time
#Modul umožňující zaslání HTTP požadavků
import requests
#Modul pro zpracování .json souborů
import json
#Modul pro práci se zprávami o běhu programu
import traceback
#Modul pro parsování RSS kanálů
import feedparser
#Modul pro práci s datem
import datetime
#Modul pro práci s Unixovými cestami
import glob
#Modul pro matematické operace
import math
#Modul od výrobce Adafruit pro senzor DHT11
import Adafruit_DHT
#Modul Pillow, umožňuje práci s obrázky
from PIL import Image, ImageTk
#Modul pro práci s GPIO piny Raspberry Pi
import RPi.GPIO as GPIO
#Modul pro správu zdrojů
from contextlib import contextmanager
#Modul pro parsování konfiguračních souborů
from six.moves import configparser
#Modul pro volání terminálových příkazů v prostředí Linux
from subprocess import call
```

Obrázek 29 - Zdrojový kód: Úvodní nastavení programu (Vlastní zpracování)

Moduly, se kterými je možné pracovat, jsou popsány v oficiální dokumentaci jazyka Python, ale je také možné si vytvořit modul vlastní, kde si definujeme své univerzální funkce a ty je poté možné importovat do jiných projektů (Python Software Foundation, 2019).

4.5.3.3 Hodiny

Předpokladem je, že modul hodin nebude zobrazovat pouze čas, ale zároveň i datum a den v týdnu. Jelikož je celý systém tvořen pro užití v České republice, je také vyžadována česká lokalizace systému. Aby toho bylo reálné dosáhnout a bylo možné zobrazovat systémové datum i se dnem v týdnu je potřeba doinstalovat do Linuxového systému lokalizační balík pro český jazyk. Toho dosáhneme pomocí příkazu „*sudo dpkg-reconfigure*“. Otevře se následující obrazovka, kde je pouze potřeba vybrat požadovaný jazyk a nainstalovat ho.



Obrázek 30 - Stažení lokalizačního balíčku (Vlastní zpracování)

Poté je již možné vytvořit pomocí klíčového slova „*class*“ třídu „*Hodiny*“, ve které je v první části definováno, jak bude rozhraní hodin graficky vypadat. Pracuje se zde s widgetem zvaným „*Label*“, který je základním způsobem, jak v knihovně Tkinter zobrazovat text, či jiné grafické prvky na obrazovce. Rozhraní hodin je tedy rozděleno do tří částí pro čas, datum a svátek. Každá z těchto částí má svůj label, ve kterém je definován font, barva popředí a pozadí a také pozici, kde bude daná část umístěna. V poslední části jsou volány funkce, které se starají o aktualizaci zobrazovaných hodnot.


```

class Hodiny(Frame):
    def __init__(self, parent, *args, **kwargs):
        Frame.__init__(self, parent, bg='black')
        # Inicializace (typ a velikost fontu, barva popředí (textu) a barva pozadí) labelu pro zobrazení času
        self.casLabel = Label(self, font=('Helvetica', 48), fg="white", bg="black")
        # Určení pozice labelu pro zobrazení času, side určuje pozici celého labelu, anchor zarovnání textu
        self.casLabel.pack(side=TOP, anchor=W)
        # Inicializace labelu pro zobrazení datumu
        self.datumLabel = Label(self, font=('Helvetica', 18), fg="white", bg="black")
        self.datumLabel.pack(side=TOP, anchor=W)
        # Inicializace labelu pro zobrazení dnešního svátku
        self.svatekLabel = Label(self, font=('Helvetica', 18), fg="white", bg="black")
        self.svatekLabel.pack(side=TOP, anchor=W)
        # Nastavení počáteční hodnoty proměnných
        self.cas1, self.datum1, self.svatek1 = '', '', ''
        # Volání funkce aktualizaceSvatku, která se stará o aktualizaci aktuálního svátku
        self.aktualizaceSvatku()
        # Volání funkce aktualizaceCasu, která se stará o aktualizaci času a datumu
        self.aktualizaceCasu()

```

Obrázek 31 - Zdrojový kód: Třída "Hodiny" (Vlastní zpracování)

Po definici rozmístění a vzhledu jednotlivých prvků je na řadě funkce, která se stará o získávání a aktualizaci dat, která jsou následně zobrazována ve výše definovaných labelech. První z nich je „*aktualizaceCasu*“, v té je nejprve nastaven český lokalizační balíček, který byl nainstalován v první části této kapitoly, aby bylo možné v datu zobrazovat české názvy dnů a měsíců.

```

def aktualizaceCasu(self):
    with setlocale('cs_CZ.UTF-8'): #Nastavení české lokalizace pro práci s datem
        cas2 = time.strftime('%I:%M:%S') #Nastavení 24hodinového formátu se sekundami, pomocí knihovny "time"
        datum2 = time.strftime("%A %d. %B %Y") #Nastavení formátu pro zobrazení data, pomocí knihovny "time"
        #Pokud se čas od poslední kontroly změnil, updatuje se
        if cas2 != self.cas1:
            self.cas1 = cas2
            self.casLabel.config(text=cas2)
        #Pokud se datum od poslední kontroly změnilo, updatuje se
        if datum2 != self.datum1:
            self.datum1 = datum2
            self.datumLabel.config(text=datum2)
        #Funkce volá sama sebe každých 200 milisekund pro aktualizaci a překreslení času/datumu
        self.casLabel.after(200, self.aktualizaceCasu)

```

Obrázek 32 - Zdrojový kód: Funkce "aktualizaceCasu" (Vlastní zpracování)

Následně se již každých 200 milisekund, pomocí zacykleného volání funkce, porovnává aktuální hodnota času, která je zobrazena, s nově naplněnou proměnnou, která bere aktuální systémový čas pomocí importovaného modulu „*Time*“, ten umožňuje zvolení výstupního formátu, v jakém se čas i datum budou zobrazovat pomocí zástupných znaků, které jsou definovány v dokumentaci daného modulu. (Python Software Foundation, 2019)

Pokud je nová hodnota odlišná, je nastavena hodnota atributu „text“ pro odpovídající label na novou hodnotu. Tím se docílí vypsaní aktuálního času. Stejný proces funguje i pro zobrazení data.

Další funkcí, která je součástí třídy Hodiny je „aktualizaceSvatku“ ta funguje podobně jako předchozí funkce, ale využívá volání veřejně dostupného API na adrese „<http://svatky.adresa.info/json>“.

```
def aktualizaceSvatku(self):
    try:
        #Volání API a uložení do proměnné
        svatky_api_json = json.loads((requests.get("http://svatky.adresa.info/json")).text)
        #Výběr požadovaného atributu
        svatek2 = (svatky_api_json[0]['name']).encode('utf-8')
        #kontrola aktuálně zobrazované hodnoty oproti nově získané
        if svatek2 != self.svatek1:
            self.svatek1 = svatek2
            #Nastavení atributu "text" pro odpovídající label
            self.svatekLabel.config(text="Dnes má svátek " + svatek2)
        #Opakované volání funkce každých 60 sekund
        self.svatekLabel.after(60000, self.aktualizaceSvatku)
        #Odchycení výjimky za využití modulu Traceback
    except Exception as e:
        traceback.print_exc()
        return "Error: %s. Nelze získat data." % e
```

Obrázek 33 - Zdrojový kód: Funkce "aktualizaceSvatku" (Vlastní zpracování)

To vrací odpověď ve formátu json (způsob zápisu dat (datový formát) nezávislý na počítačové platformě, určený pro přenos dat, která mohou být organizována v polích nebo agregována v objektech (Json.org, 2019)), ve které je uvedeno datum a jméno, které slaví daný den svátek.

Tento formát odpovědi je možné zpracovat pomocí modulu pro Python, jednoduše nazvaného „*Json*“. Nejdříve je server zavolán a odpověď je načtena v textové podobě. Následně se prochází pole prvků, které je v odpovědi obsaženo a proměnné „svatek2“ je uložena hodnota, která v json odpovědi odpovídá parametru „name“. Funkce je opět volána sama sebou po určité době, konkrétně po 60 sekundách a kontroluje, zda je zobrazena správná hodnota.

V případě chyby je zde jednoduché ošetření pro odchycení a zobrazení chybové hlášky, o kterou se stará další importovaný modul, konkrétně „*Traceback*“.

4.5.3.4 Zprávy

Druhou třídou programu jsou „Zprávy“, ta se stará o získávání zpráv z portálu Zprávy Google, který funguje jako agregátor zpravodajských článků z různých serverů. RSS kanál, ve kterém jsou zprávy na portálu dostupné, je načítán pomocí modulu „*Feedparser*“, který

existuje právě pro tento účel. Uživatel může v konfiguračním souboru zvolit, v jakém jazyce (pro jaký stát) budou zprávy stahovány a kolik jich bude zobrazeno.

Základem této části, je podobně jako u hodin, třída s názvem „Zprávy“. Zde jsou opět definovány labely knihovny Tkinter. Jsou načteny proměnné z konfiguračního souboru pomocí modulu „ConfigParser“ a na závěr je volána funkce pro získání a aktualizaci zpráv, které jsou zobrazovány ve vytvořených labelech.

```
class Zpravy(Frame):
    def __init__(self, parent, *args, **kwargs):
        # Inicializace části okna pro zobrazení zpráv
        Frame.__init__(self, parent, bg='black')
        # Inicializace (titulek, typ a velikost fontu, barva popředí (textu) a barva pozadí) labelu pro zobrazení zpráv
        self.zpravyLabel = Label(self, text="Zprávy", font=('Helvetica', 28, 'underline'), fg="white", bg="black")
        # Určení pozice labelu
        self.zpravyLabel.pack(side=TOP, anchor=W)
        # Inicializace labelu, ve kterém budou zobrazeny jednotlivé titulky zpráv
        self.titulkyLabel = Frame(self, bg="black")
        self.titulkyLabel.pack(side=TOP)
        # Uložení jazyka zpráv z konfiguračního souboru do proměnné
        self.jazyk_zprav = parser.get('news', 'jazyk_zprav')
        # Uložení počtu zobrazených zpráv z konfiguračního souboru do proměnné v podobě integeru
        self.pocet_zprav = int(parser.get('news', 'pocet_zprav'))
        # Volání funkce pro načtení zpráv
        self.aktualizaceZprav()
```

Obrázek 34 - Zdrojový kód: Třída "Zpravy" (Vlastní zpracování)

Následně jsou definovány labely i pro jednotlivé titulky zpráv, je k nim přiřazena grafická ikona, a také je nastaveno zarovnání a zalamování textu, aby blok zpráv nezasahoval moc daleko do plochy zrcadla. Pro práci s obrázkem, tedy v tomto případě ikonou zprávy, je využit modul s názvem „Pillow“

```
class TitulkyZprav(Frame):
    def __init__(self, parent, titulek_zpravy=""):
        # Inicializace labelů pro jednotlivé zprávy
        Frame.__init__(self, parent, bg='black')
        # Uprava novinové ikony pomocí modulu Pillow
        ikona = Image.open("grafika/Noviny.png")
        ikona = ikona.resize((25, 25), Image.ANTIALIAS)
        ikona = ikona.convert('RGB')
        ikona_upravena = ImageTk.PhotoImage(ikona)
        # Definice labelu pro ikonu, umístění a přiřazení ikony
        self.ikonaLabel = Label(self, bg='black', image=ikona_upravena)
        self.ikonaLabel.image = ikona_upravena
        self.ikonaLabel.pack(side=LEFT, anchor=N)
        # Definice labelu pro text titulu zprávy
        self.titulkyZprav = titulek_zpravy
        self.titulkyZpravLabel = Label(self, text=self.titulkyZprav, font=('Helvetica', 18), fg="white",
        bg="black", wraplength= 900, justify='left')
        self.titulkyZpravLabel.pack(side=LEFT, anchor=N)
```

Obrázek 35 - Zdrojový kód: Třída "TitulkyZprav" (Vlastní zpracování)

Poslední částí Zpráv je funkce „*aktualizaceZprav*“. Ta se stará o samotné získávání zpráv ze serveru Google Zprávy. Funkce se volá každých 10 minut a na základě uživatelského nastavení, které si zvolí v konfiguračním souboru se z portálu načte a zobrazí daný počet zpráv ve zvoleném jazyce. Definice funkce i s komentáři je k vidění níže.

```
def aktualizaceZprav(self):
    try:
        # Odstranění všech existujících prvků (widgetů) v labelu "titulkyLabel"
        for existujici_titulek in self.titulkyLabel.wininfo_children():
            existujici_titulek.destroy()
        # URL RSS čtečky Google Zpráv odkud jsou získávány titulky zpráv
        # Podle nastavení v konfiguračním souboru se stahují zprávy pro daný jazyk
        url_zprav = "https://news.google.cz/news?ned=%s&output=rss" % self.jazyk_zprav
        # Zpracování RSS feedu pomocí modulu feedparser
        feed = feedparser.parse(url_zprav)
        # Vytvoření titulků zpráv dle počtu definovaného v konfiguračním souboru
        for zprava in feed.entries[0:self.pocet_zprav]:
            # Titulky zpráv se vytváří za použití třídy "TitulkyZprav"
            # Následně je umístěn a zarovnán jeho text
            TitulkyZprav(self.titulkyLabel, zprava.title).pack(side=TOP, anchor=W)

        # Odchycení výjimky za využití modulu Traceback
    except Exception as e:
        traceback.print_exc()
        print "Error: %s. Nelze získat zprávy." % e
    # Opakované volání funkce každých 600 sekund
    self.after(600000, self.aktualizaceZprav)
```

Obrázek 36 - Zdrojový kód: Funkce "aktualizaceZprav" (Vlastní zpracování)

4.5.3.5 Počasí

Třetí třída programu se nazývá „*Počasí*“ a je první, která získává data nejen z internetu, ale i z připojených čidel, konkrétně tedy teplotu a vlhkost vzduchu. Je tedy potřeba umístit čidla tam, kde chceme dané hodnoty zjišťovat. Tyto dvě hodnoty jsou zároveň ještě doplněny aktuální slovní předpovědí pro danou lokalitu, ta se získává pomocí API z webu „<https://darksky.net>“ na základě lokality veřejné IP adresy, která je také získávána ze stránek „<https://ipstack.com>“. Tato webová stránka funguje jako geolokační služba, která je schopna na základě veřejné IP adresy vrátit zeměpisnou šířku a délku, tyto hodnoty jsou následně využity pro získání předpovědi pro danou lokalitu. Pro využití služeb DarkSky a ipstack je potřeba se nejdříve na těchto stránkách zdarma registrovat a nechat si vygenerovat soukromé API klíče, které je následně potřeba uvést v konfiguračním souboru.

V první části je opět definice třídy, kde se pracuje především s labely modulu Tkinter, ty jsou konfigurovány pro každou hodnotu zvlášť. Zároveň jsou uvedeny počáteční hodnoty

jednotlivých proměnných a je zavolána funkce, která má na starost získávání jednotlivých měření veličin.

```
class Pocasi(Frame):
    def __init__(self, parent, *args, **kwargs):
        #Inicalizace části pro modul "Pocasi"
        Frame.__init__(self, parent, bg='black')
        #Nastavení počáteční hodnoty proměnných
        self.teplota, self.lokalita, self.aktualniPredpoved, self.vlhkost = '', '', '', ''
        # Inicalizace (typ a velikost fontu, barva popředí (textu) a barva pozadí) labelu pro zobrazení teploty
        self.teplotaLabel = Label(self, font=('Helvetica', 94), fg="white", bg="black")
        # Určení pozice labelu
        self.teplotaLabel.pack(side=TOP, anchor=W)
        #Inicalizace labelu pro zobrazení aktuální předpovědi
        self.aktualniPredpovedLabel = Label(self, font=('Helvetica', 28), fg="white", bg="black")
        self.aktualniPredpovedLabel.pack(side=TOP, anchor=W)
        #Inicalizace labelu pro zobrazení vlhkosti vzduchu
        self.vlhkostLabel = Label(self, font=('Helvetica', 28), fg="white", bg="black")
        self.vlhkostLabel.pack(side=TOP, anchor=W)
        #Inicalizace labelu pro zobrazení lokality, na kterou se vztahuje předpověď
        self.lokalitaLabel = Label(self, font=('Helvetica', 18), fg="white", bg="black")
        self.lokalitaLabel.pack(side=TOP, anchor=W)
        #Volání funkce aktualizacePocasi, která se stará o aktualizaci zobrazovaných hodnot
        self.aktualizacePocasi()
```

Obrázek 37 - Zdrojový kód: Třída "Pocasi" (Vlastní zpracování)

Volaná funkce „*aktualizacePocasi*“ má na starost především volání dalších podpůrných funkcí a sama nejprve postupně volá jednotlivá API, která vedou k získání aktuální textové předpovědi počasí. Konkrétně se jedná o ipstack API, která vrací na základě veřejné IP adresy zeměpisnou délku a šířku. Ty jsou následně využity při volání DarkSky API, ze kterého získává aktuální slovní předpověď počasí, zde je potřeba ve volané adrese především určit v jakém jazyce má být předpověď stažena, aktuálně služba podporuje 12 jazyků včetně češtiny. Následně se funkce pouze stará o aktualizování zobrazovaných hodnot každých 10 minut.

```

def aktualizacePocasi(self):
    try:
        #Uložení API webové adresy, která slouží k získání lokality dané IP adresy,
        #do proměnné, která má jako parametry veřejnou IP adresu,
        #která se získá pomocí funkce "ziskejIP" a unikátní API klíč uvedený v konfiguračním souboru
        lokalitaApiUrl = "http://api.ipstack.com/%s?access_key=%s" % (self.ziskejIP(), ip_stack_api)
        #Načtení Json souboru z API
        lokalitaJson = json.loads((requests.get(lokalitaApiUrl)).text)
        #Uložení zeměpisné šířky z načteného Jsonu do proměnné
        lat = lokalitaJson['latitude']
        #Uložení zeměpisné délky z načteného Jsonu do proměnné
        lon = lokalitaJson['longitude']
        #Uložení názvu lokality z načteného Jsonu do proměnné
        lokalita2 = "%s" % (lokalitaJson['region_name'])

        #Webová adresa API, která slouží pro získání předpovědi počasí, jako parametry jsou uvedené
        #unikátní API klíč z konfiguračního souboru a zeměpisná šířka a délka
        pocasiApiUrl = "https://api.darksy.net/forecast/%s/%s,%s?lang=cs&units=si" % (darksy_api, lat,lon)
        #Načtení Json souboru z API
        pocasiJson = json.loads((requests.get(pocasiApiUrl)).text)
        #Uložení teploty získané funkcí "ziskejTeplotu" do proměnné společně se znakem "°C"
        teplota2 = "%s°C" % (round(self.ziskejTeplotu(),1), u'\N{DEGREE SIGN}')
        #Uložení vlhkosti získané funkcí "ziskejVlhkost" do proměnné
        vlhkost2 = "%s%" % (self.ziskejVlhkost())
        #Uložení aktuální předpovědi získané z načteného Jsonu do proměnné
        aktualniPredpoved2 = pocasiJson['currently']['summary']

        #Kontrola jednotlivých hodnot, zda se od posledního přepsání změnily, pokud ano, jsou opět přepsány na
        #aktuální
        if self.aktualniPredpoved != aktualniPredpoved2:
            self.aktualniPredpoved = aktualniPredpoved2
            self.aktualniPredpovedLabel.config(text=aktualniPredpoved2)
        if self.vlhkost != vlhkost2:
            self.vlhkost = vlhkost2
            self.vlhkostLabel.config(text=vlhkost2)
        if self.teplota != teplota2:
            self.teplota = teplota2
            self.teplotaLabel.config(text=teplota2)
        if self.lokalita != lokalita2:
            if lokalita2 == ", ":
                self.lokalita = "Nelze získat lokalitu"
                self.lokalitaLabel.config(text="Nelze získat lokalitu")
            else:
                self.lokalita = lokalita2
                self.lokalitaLabel.config(text=lokalita2)

        #Odchycení výjimky využitím modulu Traceback
        except Exception as e:
            traceback.print_exc()
            print "Error: %s. Nelze získat počasí." % e

        #Zacyklené volání funkce každých 10 minut pro aktualizaci hodnot
        self.after(600000, self.aktualizacePocasi)

```

Obrázek 38 - Zdrojový kód: Funkce "aktualizacePocasi" (Vlastní zpracování)

Jako doplňující funkce jsou zde konkrétně „*ziskejTeplotu*“, ta slouží k načtení teploty z čidla DS18B20 připojeného pomocí sběrnice 1-wire, které funguje tak, že teplotu neustále zapisuje do specifického souboru, z kterého ho funkce čte, upravuje na stupně Celsia a danou hodnotu vrací. Následně funkce „*ziskejVlhkost*“, ta získává hodnotu vlhkosti vzduchu z čidla DHT11 pomocí modulu přímo od výrobce, ten z čidla získává jak teplotu, tak i vlhkost, ale funkce vrací jen hodnotu vlhkosti. Poslední funkcí je „*ziskejIP*“, ta využívá veřejně dostupného API ipify.com pro načtení veřejné IP adresy zařízení ve formátu Json. Na základě této IP adresy je poté v hlavní funkci získávána lokalita.

```

#Funkce pro získání teploty z čidla DS18B20
def ziskejTeplotu(self):
    #Načtení souboru, do kterého čidlo zapisuje hodnoty
    f = open(souborTeplot, 'r')
    #přečtení jednotlivých řádků a uložení do proměnné
    radky = f.readlines()
    f.close()
    #nalezení hodnoty teploty
    nactenaHodnota = radky[1].find('t=')
    #pokud teplota není chybová hodnota
    if nactenaHodnota != -1:
        #přečte se naměřená hodnota
        teplota = radky[1][nactenaHodnota+2:]
        #Vydělením 1000 se získá teplota ve stupních Celsia
        teplotaCelsium = float(teplota) / 1000.0
        #Vrácení hodnoty aktuální naměřené teploty
        return teplotaCelsium

#Funkce pro získání vlhkosti z čidla DHT11
def ziskejVlhkost(self):
    #Získání vlhkosti vzduchu a teploty z čidla DHT11 pomocí modulu pro Python od výrobce
    vlhkost, teplota = Adafruit_DHT.read_retry(11, 21)
    #Vrácení pouze hodnoty vlhkosti
    return vlhkost

#Funkce pro získání veřejné IP adresy
def ziskejIP(self):
    try:
        #Načtení Json souboru z veřejné API
        ip_json = json.loads((requests.get("https://api.ipify.org/?format=json")).text)
        #Vrácení hodnoty odpovídající IP adrese
        return ip_json['ip']
    #Odchycení výjimky využitím modulu Traceback
    except Exception as e:
        traceback.print_exc()
        return "Error: %s. Nelze získat IP adresu." % e

```

Obrázek 39 - Zdrojový kód: doplňkové funkce pro počasí (Vlastní zpracování)

4.5.3.6 Kalendář

Poslední třídou programu je „*Kalendar*“ ta zobrazuje 5 nejbližších událostí z uživatelského Google kalendáře. Pro zprovoznění API, které Google pro tuto službu poskytuje je nejprve nutné doinstalovat speciální moduly, které byly zmíněny v úvodu této kapitoly a poté ještě na webové stránce „<https://developers.google.com/calendar/quickstart/python>“ povolit využití Google Calendar API pro daný uživatelský účet, tím se vygeneruje soubor `credentials.json`, který je potřeba stáhnout a umístit do složky se souborem programu, pomocí něho je možné aby program byl schopen přistupovat ke kalendáři. Samotná třída je poté definována podobně jako třídy předchozí, opět je definováno rozmístění a vzhled jednotlivých labelů, které jsou následně plněny volanou funkcí „*aktualizaceUdalosti*“, tato funkce je založena na příkladu, který Google prezentuje na výše zmíněné webové adrese a je pouze

upravena pro účely tohoto programu, aby vracela záznamy v požadované podobě. Funkce je opět volána každých 10 minut.

```
class Kalendar(Frame):
    def __init__(self, parent, *args, **kwargs):
        Frame.__init__(self, parent, *args, **kwargs)
        self.config(bg='black')
        self.kalendarLabel = Label(self, text='Kalendář', font=('Helvetica', 28, 'underline'), fg="white",
                                   bg="black")
        self.kalendarLabel.pack(side=TOP, anchor=W)
        self.udalostiLabel = Frame(self, bg="black")
        self.udalostiLabel.pack(side=TOP)
        self.aktualizaceUdalosti()

    def aktualizaceUdalosti(self):
        # Odstranění všech existujících prvků (widgetů) v labelu "udalostiLabel"
        try:
            for udalost in self.udalostiLabel.wininfo_children():
                udalost.destroy()
            #Přihlašování do Google Calendar API
            creds = None

            if os.path.exists('token.pickle'):
                with open('token.pickle', 'rb') as token:
                    creds = pickle.load(token)
            #V případě že neexistuje přihlašovací soubor, je uživatel vyzván k přihlášení skrze webovou adresu
            if not creds or not creds.valid:
                if creds and creds.expired and creds.refresh_token:
                    creds.refresh(Request())
                else:
                    flow = InstalledAppFlow.from_client_secrets_file(
                        'credentials.json', SCOPES)
                    creds = flow.run_local_server()
            #Uložení přihlašovacích údajů pro příští využití
            with open('token.pickle', 'wb') as token:
                pickle.dump(creds, token)

            service = build('calendar', 'v3', credentials=creds)

            #Volání Google Calendar API a získávání 5 nejbližších událostí
            now = datetime.datetime.utcnow().isoformat() + 'Z' # 'Z' indicates UTC time
            events_result = service.events().list(calendarId='primary', timeMin=now,
                                                  maxResults=5, singleEvents=True,
                                                  orderBy='startTime').execute()
            udalosti = events_result.get('items', [])

            for udalost in udalosti:
                #Načtení času události do proměnné
                start = (udalost['start'].get('dateTime', udalost['start'].get('date')))[0:19]
                #Převod na požadovaný formát
                oldStart = datetime.datetime.strptime(start, '%Y-%m-%dT%H:%M:%S')
                startFormat = oldStart.strftime('%-d. %-m. %Y v %-H:%-M')
                #Nastavení atributu text pro odpovídající label
                udalostTitulek = UdalostiKalendare(self.udalostiLabel, startFormat + ' - ' +
                                                  udalost['summary'])
                #Umístění labelu
                udalostTitulek.pack(side=TOP, anchor=W)
        except Exception as e:
            traceback.print_exc()
            print "Error: %s. Nelze získat události z kalendáře." % e

        self.after(600000, self.aktualizaceUdalosti)

class UdalostiKalendare(Frame):
    def __init__(self, parent, udalost_kalendare=""):
        #Inicizace labelů pro jednotlivé události z kalendáře
        Frame.__init__(self, parent, bg='black')
        #Úprava ikony kalendáře pomocí modulu Pillow
        image = Image.open("grafika/Calendar.png")
        image = image.resize((25, 25), Image.ANTIALIAS)
        image = image.convert('RGB')
        photo = ImageTk.PhotoImage(image)
        #Definice labelu pro ikonu, umístění a přiřazení ikony
        self.ikonaLabel = Label(self, bg='black', image=photo)
        self.ikonaLabel.image = photo
        self.ikonaLabel.pack(side=LEFT, anchor=N)
        #Definice labelu pro text titulků událostí
        self.udalostKalendare = udalost_kalendare
        self.udalostKalendareLabel = Label(self, text=self.udalostKalendare, font=('Helvetica', 18), fg="white",
                                           bg="black", wraplength= 500, justify='left')
        self.udalostKalendareLabel.pack(side=LEFT, anchor=N)
```

Obrázek 40 - Zdrojový kód: Třída "Kalendar" (Vlastní zpracování)

4.5.3.7 Celobrazovkový režim

Poslední částí programu je funkce, která se stará o snížení spotřeby tím, že každých 5 sekund kontroluje pomocí detekce pohybu, k čemuž využívá senzor HC-SR501, zda se před zrcadlem někdo nachází, pokud ne, uspí pomocí Linuxového příkazu HDMI konektor, a tedy i k němu připojený monitor.

Zároveň se zde nachází i poslední třída s názvem „*FullscreenOkno*“ kde je definováno rozmístění jednotlivých modulů programu, které byly definovány výše a zároveň jsou zde i dvě funkce pro zapnutí a vypnutí celobrazovkového režimu, ty jsou namapovány na klávesu „*Enter*“ pro spuštění a „*Escape*“ pro ukončení.

```
def uspaniObrazovky(self):
    #Nastavení práce s GPIO piny
    GPIO.setwarnings(False)
    GPIO.setmode(GPIO.BOARD)
    GPIO.setup(21, GPIO.IN)
    #Načtení hodnoty z piny 21
    i = GPIO.input(21)
    #Pokud je hodnota 0 (Bez napětí) monitor (HDMI) se uspí
    if i == 0:
        #Linux terminálový příkaz pro uspaní HDMI
        call("vcgencmd display_power 0")
    #Pokud je hodnota 1, tedy je detekován pohyb, monitor se probudí
    else:
        call("vcgencmd display_power 1")
    #Opakovaná kontrola každou minutu
    self.after(60000, self.uspaniObrazovky)

class FullscreenOkno:

    def __init__(self):
        #Inicializace hlavního okna s černým pozadím
        self.tk = Tk()
        self.tk.configure(background='black')
        self.topFrame = Frame(self.tk, background = 'black')
        self.bottomFrame = Frame(self.tk, background = 'black')
        self.topFrame.pack(side = TOP, fill=BOTH, expand = YES)
        self.bottomFrame.pack(side = BOTTOM, fill=BOTH, expand = YES)
        self.state = False
        #Přiřazení kláves pro spuštění a ukončení fullscreen (celobrazovkového) režimu
        self.tk.bind("<Return>", self.zapniFullscreen)
        self.tk.bind("<Escape>", self.ukonciFullscreen)
        #Umístění části hodin a nastavení ohraničení
        self.hodiny = Hodiny(self.topFrame)
        self.hodiny.pack(side=RIGHT, anchor=N, padx=100, pady=60)
        #Umístění počasí
        self.pocasi = Pocasi(self.topFrame)
        self.pocasi.pack(side=LEFT, anchor=N, padx=100, pady=60)
        #Umístění zpráv
        self.zpravy = Zpravy(self.bottomFrame)
        self.zpravy.pack(side=LEFT, anchor=S, padx=100, pady=60)
        #Umístění kalendáře
        self.kalendar = Kalendar(self.bottomFrame)
        self.kalendar.pack(side=RIGHT, anchor=S, padx=100, pady=60)

        #Funkce pro aktivaci fullscreen režimu
        def zapniFullscreen(self, event=None):
            self.state = not self.state # Just toggling the boolean
            self.tk.attributes("-fullscreen", self.state)
            return "break"

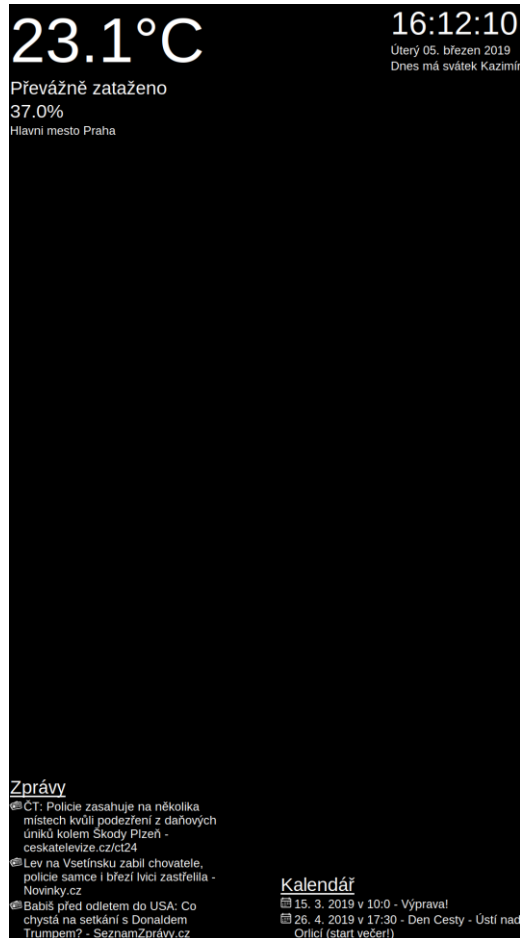
        #Funkce pro ukončení fullscreen režimu
        def ukonciFullscreen(self, event=None):
            self.state = False
            self.tk.attributes("-fullscreen", False)
            return "break"

    #Úvodní spuštění celého grafického rozhraní, tak jak je definováno v jednotlivých třídách
    if __name__ == '__main__':
        w = FullscreenOkno()
        w.tk.mainloop()
```

Obrázek 41 - Zdrojový kód: Třída "FullscreenOkno" (Vlastní zpracování)

4.6 Finální podoba

Finální podoba grafického rozhraní a zároveň i celého konceptu po osazení monitoru do dřevěného rámu se zrcadlovou plochou je k vidění na obrázku níže.



Obrázek 42 - Finální podoba varianty B (Vlastní zpracování)

Grafické rozložení jednotlivých prvků je nutné pomocí odsazení upravit po osazení do rámu tak, aby byl text dobře vidět a zrcadlo je následně připravené k užívání. Jedná se o modelový případ, proto jsou všechny senzory prozatím ukryty společně s Raspberry Pi za plochou zrcadla, při reálném nasazení by bylo samozřejmě nutné umístit senzory tam, kde uživatel chce dané veličiny měřit a následně data vést pomocí například pomocí běžných UTP kabelů k Raspberry Pi. Tato varianta již vyžaduje jistou dávku znalostí, ale má především velký potenciál, co se dalšího rozšíření týče.

5 Závěr

Diplomová práce se zaměřuje na oblast inteligentních domácností, které spadají pod technologii internetu věcí, se zaměřením na možnosti vlastní realizace chytrých zařízení. V tomto případě bylo daným zařízením chytré zrcadlo. Cílem bylo nejprve teoreticky představit danou oblast, popsat vývoj, využívané technologie a případný budoucí vývoj, následně na těchto základech pokračovala část praktická.

V té bylo cílem nejprve vydefinovat, jakým způsobem by mohlo být chytré zrcadlo realizováno, toho bylo dosaženo pomocí anonymního dotazníku, ve kterém 20 respondentů odpovědělo na důležité dotazy, jak by si oni sami představovali koncept chytrého zrcadla a práci s ním. Na základě vyhodnocení jednotlivých preferencí a díky využití grafické myšlenkové mapy byly ve výsledku vybrány čtyři varianty pro možnou realizaci. Konkrétně se jednalo o možnost využití kombinace snímání uživatele pomocí kamery a následného využití umělé inteligence pro analýzu obrazu, další možností byla integrace virtuálního asistenta do běžného zrcadla, třetí varianta představovala zobrazování vybraných informací na ploše zrcadla a poslední variantou bylo využití některého z mála existujících komerčních řešení. Tyto varianty byly následně porovnány využitím vícekritériální analýzy, kde byly hodnoceny v kritériích pořizovací náklady, náklady na užívání, rozšiřitelnost, znovu využitelnost a složitost. Kritéria byla vybrána tak, aby vystihovala oblast výroby takovýchto zařízení vlastními silami. Pro stanovení vah jednotlivých kritérií byl využit také již zmíněný dotazník, kde respondenti využitím Metfesselovy alokace sta bodů přiřadili jednotlivým kritériím body tak, jak by pro ně samotné byly při rozhodování důležité.

Výsledkem této analýzy bylo umístění konceptu s integrovaným virtuálním asistentem na prvním místě, především díky skutečnosti, že v kritériích s nejvyšší vahou, tedy pořizovací cena a náklady na užívání se umístil nejlépe. Jedná se o ekonomicky nenáročné a konceptuálně jednoduché řešení. Pro toto řešení byl tedy v další části vytvořen návrh realizace za využití asistenta Google Home, který byl vybrán především díky tomu, že by měl v nejbližší době jako první podporovat český jazyk. Tento koncept má silný potenciál především v domácnostech, kde jsou využívány i jiné inteligentní přístroje a byla by tedy možnost provázat funkčnost Google Home i s nimi.

Na druhém místě se umístilo řešení, které by zobrazovalo informace na ploše zrcadla. I pro tento koncept byl vytvořen návrh, jelikož je zde možné mnohem lépe demonstrovat technologie, které jsou aktuálně dostupné přesně pro situace jako je tato. Konkrétně tedy byl využit mikropočítač Raspberry Pi, jelikož se jedná o aktuálně nejpoblábnější zařízení v dané

oblasti, rozšiřující senzory pro měření teploty, vlhkosti vzduchu a detekci pohybu. To vše bylo zastřešeno program psaným v jazyce Python. Výsledkem je tedy chytré zrcadlo, které zobrazuje naměřené hodnoty ze senzorů, tedy teplotu a vlhkost, to doplňuje textovou předpovědí získávanou z webové služby. Dále také zobrazuje aktuální čas společně se jménem slavící daný den svátek, titulky zpráv získané z portálu Google Zprávy a také nejbližší události z uživatelova Google kalendáře. Toto řešení je otevřené i pro další rozvoj, na základě požadavků uživatel by mohly být například připojeny další senzory, které by měřily vybrané veličiny, či interagovaly s jinými zařízeními v domácnosti, stejně tak by mohly být rozšířeny informace získávané z internetu. V aktuálním stavu zrcadlo zobrazuje informace bez interakce od uživatele, to by bylo možné také změnit, například přidat možnost ovládat zrcadlo hlasem a zobrazovat informace tak, jak si uživatel zvolí běžnou mluvou. Zároveň bylo také možné vytvořit jednoduchý webový formulář pro úpravu konfiguračního souboru, aby uživatel nebyl nucen přistupovat přímo na úložiště Raspberry Pi. V neposlední řadě by bylo možné díky využití Raspberry Pi integrovat i virtuálního asistenta, například opět Google Asistenta, který je součástí i výše zmíněného Google Home a pro, kterého společnost Google nabízí veřejné API.

Navržená řešení mají potenciál nejen při tvorbě pro soukromé využití, ale jelikož komerčních řešení poskytujících tyto možnosti není na trhu mnoho, spíše téměř žádná, naskýtá se zde možnost i využití těchto konceptů jako základ pro obchodní model. Díky nízkým nákladům, široké možnosti rozšiřitelnosti a přizpůsobitelnosti požadavkům zákazníka by bylo možné pokrýt relativně rozsáhlou oblast trhu.

Internet věcí a jeho zařízení se budou i nadále integrovat do života běžných lidí a za inteligentní domácnost bude již brzy standardem, k tomu přispějí nejen komerční výrobci, ale zároveň i řešení podobná těm, která tato práce představuje.

6 Seznam použitých zdrojů

1. **Forbes Technology Council.** 14 Predictions For The Future Of Smart Home Technology. [Online] 2018. [Citace: 25. Říjen 2018.] <https://www.forbes.com/sites/forbestechcouncil/2018/01/12/14-predictions-for-the-future-of-smart-home-technology>.
2. **Dekra.** Smart Home Protocols – Thread, Zigbee, Z-Wave, KNX and More. [Online] 2017. [Citace: 5. Listopad 2018.] <https://www.iotforall.com/smart-home-protocols/>.
3. **Gebhart, Andrew.** Everything you need to know about Google Home. [Online] 2018. [Citace: 1. Prosinec 2018.] <https://www.cnet.com/how-to/everything-you-need-to-know-about-google-home/>.
4. **Huffman, Scott.** The future of the Google Assistant: Helping you get things done to give you time back. [Online] 2018. [Citace: 2. Prosinec 2018.] <https://www.blog.google/products/assistant/io18/>.
5. **Anwar, Malik Nadeem, Nazir, Mohammad a Mustafa, Khurram.** Security threats taxonomy: Smart-home perspective. [Online] 1. Září 2017. [Citace: 5. Prosinec 2018.] https://www.researchgate.net/publication/324727021_Security_threats_taxonomy_Smart-home_perspective.
6. **Dennis, K. Andrew.** *Raspberry Pi Home Automation with Arduino - Second Edition.* 2. Birmingham : Packt Publishing, 2015.
7. **Raspberry Pi Tutorials.** 50 of the most important Raspberry Pi Sensors and Components. [Online] 2019. [Citace: 1. Leden 2019.] <https://tutorials-raspberrypi.com/raspberry-pi-sensors-overview-50-important-components/>.
8. **Arduino návody.** Teplotní senzor DS18B20. *Arduino návody.* [Online] 28. Červen 2016. [Citace: 2. Leden 2019.] <https://navody.arduino-shop.cz/navody-k-produktum/teplotni-senzor-ds18b20.html>.
9. —. Pohybové čidlo HC-SR501. [Online] 2016. [Citace: 2. Leden 2019.] <https://navody.arduino-shop.cz/navody-k-produktum/pohybove-cidlo-hc-sr501.html>.
10. **Microsoft.** Face. [Online] 2019. [Citace: 15. Leden 2019.] <https://azure.microsoft.com/en-us/services/cognitive-services/face/>.
11. **Kohler.** Verdera® Medicine Cabinets and Mirrors. [Online] 2019. [Citace: 25. Leden 2019.] <https://www.us.kohler.com/us/Verdera-Medicine-Cabinets/article/CNT122900002.htm>.
12. **Alza.** Alza.cz . [Online] 2019. [Citace: 30. Leden 2019.] <https://www.alza.cz>.
13. **Kohler.** Verdera®. [Online] 2019. [Citace: 30. Leden 2019.] <https://www.us.kohler.com/us/verdera-lighted-medicine-cabinet-40-w-30-h/productDetail/medicine-cabinets/1276430.htm>.
14. **Snell, Essie.** OK Google, How Much Energy Does Alexa Consume? [Online] 2017. [Citace: 30. Leden 2019.] <https://www.esource.com/es-blog-2-17-17-voice-control/ok-google-how-much-energy-does-alexa-consume>.
15. **Geerling, Jeff.** Power Consumption Benchmarks. [Online] 2019. [Citace: 30. Leden 2019.] <https://www.pidramble.com/wiki/benchmarks/power-consumption>.
16. **Acer.** G246HYL. [Online] 2019. [Citace: 30. Leden 2019.] <https://www.acer.com/ac/cs/CZ/content/model/UM.QG6EE.001>.
17. **Philips.** Philips 243V7QJABF. [Online] 2019. [Citace: 30. Leden 2019.] https://www.philips.cz/c-p/243V7QJABF_00/full-hd-lcd-monitor/specifications.
18. **Upton, Eben a Halfacree, Gareth.** *Raspberry Pi: Uživatelská příručka.* 1. Brno : Computer Press, 2013.
19. **StackExchange.** Raspberry Pi. [Online] 2019. [Citace: 30. Leden 2019.] <https://raspberrypi.stackexchange.com/questions>.

20. **Systemnegar Saina.** DeepFace. [Online] 2019. [Citace: 30. Leden 2019.] <https://deepface.ir>.
21. **Google.** Google Home Mini Partners. [Online] 2019. [Citace: 20. Leden 2019.] https://store.google.com/us/product/google_home_mini_partners?hl=en-US.
22. **Stavební sklo s.r.o.** Polopropustná zrcadla. [Online] 2019. [Citace: 30. Leden 2019.] <http://www.stavebni-sklo.cz/polopropustna-zrcadla.php>.
23. **Fritzing.** About. [Online] 2019. [Citace: 30. Leden 2019.] <http://fritzing.org/about/>.
24. **Python Software Foundation.** Modules. [Online] 2019. [Citace: 30. Leden 2019.] <https://docs.python.org/3/tutorial/modules.html>.
25. **Json.org.** Introducing JSON. [Online] 2019. [Citace: 30. Leden 2019.] <https://www.json.org>.
26. **Sundmaeker, Harald, a další.** *Vision and Challenges for Realising the Internet of Things.* Luxembourg : Publications Office of the European Union, 2010.
27. **DiClerico, Dan.** Why 'Do It For Me' is Defining the Smart Home Market. *HomeAdvisor.* [Online] 13. Červenec 2017. https://www.homeadvisor.com/r/why-do-it-for-me-is-defining-the-smart-home-market/?entry_point_id=32454654&m=conengage.
28. **Hesse, Brendan.** How to Get Started With the Windows Subsystem for Linux. *Lifehacker.* [Online] 11. Zář 2018. <https://lifehacker.com/how-to-get-started-with-the-windows-subsystem-for-linux-1828952698>.
29. **Hoy, Matthew B.** Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants. [Online] 2018. [Citace: 5. Únor 2019.] <https://www.tandfonline.com/doi/abs/10.1080/02763869.2018.1404391?journalCode=wmrs20>.
30. **Labioud, Houda, Afifi, Hossam a de Santis, Costantino.** *Wi-Fi™, Bluetooth™, Zigbee™ and WiMax™.* 1. Amsterdam : Springer Netherlands, 2007.
31. **Python Software Foundation.** time — Time access and conversions. [Online] 2019. [Citace: 10. Leden 2019.] <https://docs.python.org/2/library/time.html>.
32. **Robinson, David.** The Incredible Growth of Python. [Online] 2017. [Citace: 15. Leden 2019.] <https://stackoverflow.blog/2017/09/06/incredible-growth-python/>.
33. **Raspberry Pi Spy.** Simple Guide to the Raspberry Pi GPIO Header and Pins. [Online] 2012. [Citace: 30. Prosinec 2018.] <https://www.raspberrypi-spy.co.uk/2012/06/simple-guide-to-the-rpi-gpio-header-and-pins/>.
34. **GM Electronic.** GM Electronic. [Online] 2019. [Citace: 1. Leden 2019.] <https://www.gme.cz>.
35. **How it works.** How do two-way mirrors work? [Online] 2015. [Citace: 20. Leden 2019.] <https://www.howitworksdaily.com/how-do-two-way-mirrors-work/>.
36. **Ashton, Kevin.** That 'Internet of Things' Thing. *RFID Journal.* [Online] 22. Červen 2009. [Citace: 15. Zář 2018.] <https://www.rfidjournal.com/articles/view?4986>.
37. **Carritech.** Internet of Things: Explained. *Carritech telecommunications.* [Online] 11. Leden 2018. [Citace: 15. Zář 2018.] <http://www.carritech.com/news/internet-of-things/>.
38. **European Research Cluster on the Internet of Things.** Internet of Things. *European Research Cluster on the Internet of Things.* [Online] 11. Leden 2018. [Citace: 15. Zář 2018.] http://www.internet-of-things-research.eu/about_iiot.htm.
39. **SAP.** The Internet of Things definition. *SAP.* [Online] 11. 1 2018. [Citace: 20. Zář 2018.] <https://www.sap.com/trends/internet-of-things.html>.
40. **Cisco.** Cisco. *The Internet of Things - The Internet Protocol Journal, Volume 15, No. 3.* [Online] Zář 2012. [Citace: 20. Zář 2018.] <https://www.cisco.com/c/en/us/about/press/internet-protocol-journal/back-issues/table-contents-57/153-internet.html>.

41. **Gartner Inc.** Internet of Things. *Gartner*. [Online] 11. Leden 2018. [Citace: 20. Zář 2018.] <https://www.gartner.com/it-glossary/internet-of-things/>.
42. **Evans, Dave.** The Internet of Things: How the Next Evolution of the Internet Is Changing Everything. [Online] 1. Duben 2011. [Citace: 3. Říjen 2018.] https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf.
43. **Statista.** Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025 (in billions). [Online] 2018. [Citace: 3. Říjen 2018.] <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/>.
44. **IoT Online store.** The Internet of Things: What it is and why you should care. [Online] 2018. [Citace: 12. Říjen 2018.] <http://www.iotonlinestore.com>.
45. **Harper, Richard.** *Inside the Smart Home*. 1. London : Springer-Verlag London, 2003.
46. **Clouser, Grant, a další.** *The Complete Guide to DIY Smart Home Systems*. 1. Filadelfie : BookBaby, 2015.
47. **Valeš, Miroslav.** *Inteligentní dům*. 1. Praha : Era - vydavatelství, 2006.
48. **Antik Technology.** ANTIK Smart Home Systems. [Online] 2018. [Citace: 15. Říjen 2018.] <http://www.antiktech.com/en/digital-signagebr--smart-home/smart-home>.
49. **Málek, Michael a Kovářová, Petra.** Chytrá domácnost jako součást chytrého města. *ICT Revue*. [Online] 14. Červen 2018. [Citace: 15. Říjen 2018.] https://ictrevue.ihned.cz/c3-66168830-0ICT00_d-66168830-chytra-domacnost-jako-soucast-chytreho-mesta.
50. **Hendricks, Drew.** The History of Smart Homes. *IoT Evolution*. [Online] 22. Duben 2014. [Citace: 16. Říjen 2018.] <https://www.iotevolutionworld.com/m2m/articles/376816-history-smart-homes.htm>.
51. **Harper, Richard.** *The Connected Home: The Future of Domestic Life*. 1. Londýn : Springer, 2012.
52. **Asociace chytrého bydlení.** Počet chytrých domácností v Evropě do roku 2021 vzroste sedminásobně! *Asociace chytrého bydlení*. [Online] 9. Srpen 2017. [Citace: 20. Říjen 2018.] <http://www.achb.cz/2017/08/vyvoj-smart-home-2017/>.
53. **Carlson, Troy.** The 5 elements of smart homes. [Online] 2016. [Citace: 1. Listopad 2018.] <http://www.modernsmarthome.com/the-5-elements-of-smart-homes/>.
54. **EH Contributor.** Home Automation Protocols: A Round-Up. [Online] 2016. [Citace: 10. Listopad 2018.] <https://www.electronicshouse.com/smart-home/home-automation-protocols-what-technology-is-right-for-you/>.
55. **Fearing's.** Remote home control system on a smart phone. [Online] 2018. [Citace: 15. Listopad 2018.] <https://www.fearings.com/blog/wisconsin-smart-homes/remote-home-control-system-on-a-smart-phone/>.
56. **auto.home.techs.** Siri vs Google vs Alexa vs Cortana – Which one is best voice assistant. [Online] 2018. [Citace: 20. Listopad 2018.] <https://homeautotechs.com/siri-vs-google-vs-alexa-vs-cortana-which-one-is-best-voice-assistant/>.
57. **Gartner Inc.** 5 Trends Emerge in the Gartner Hype Cycle for Emerging Technologies, 2018. [Online] 2018. [Citace: 20. Listopad 2018.] <https://www.gartner.com/smarterwithgartner/5-trends-emerge-in-gartner-hype-cycle-for-emerging-technologies-2018/>.
58. **Kurka, Milan.** Hlasoví asistenti v kostce: jaký byl jejich vývoj a jak fungují? [Online] 2017. [Citace: 25. Listopad 2018.] <https://casopis.fit.cvut.cz/technologie/hlasovi-asistenti-kostce-jaky-vyvoj-funguji/>.
59. **Dunn, Jeff.** It looks like Apple has some work to do if it wants Siri to be as smart as Google Assistant. [Online] 2017. [Citace: 25. Listopad 2018.] <https://www.businessinsider.com/siri-vs-google-assistant-cortana-alexa-knowledge-study-chart-2017-6>.

60. **Pajankar, Ashwin.** *Raspberry Pi Computer Vision Programming*. 1. Birmingham : Packt Publishing, 2015.
61. **Moorhead, Joanna.** Raspberry Pi device will 'reboot computing in schools'. *The Guardian*. [Online] 9. Leden 2012. [Citace: 20. Prosinec 2018.] <https://www.theguardian.com/education/2012/jan/09/raspberry-pi-computer-revolutionise-computing-schools>.
62. **Hitaltech.** The History of the Raspberry Pi. *Hitaltech*. [Online] 7. Září 2018. [Citace: 25. Prosinec 2018.] <https://hitaltech.co.uk/the-history-of-the-raspberry-pi/>.
63. **Raspberry Pi Foundation.** Raspberry Pi. *About Us*. [Online] 17. Leden 2018. [Citace: 26. Prosinec 2018.] <https://www.raspberrypi.org/about/>.
64. **Vítek, Jan.** V den čísla Pi přichází na svět nový Raspberry Pi 3 Model B+. *Svět hardware*. [Online] 14. Březen 2018. [Citace: 26. Prosinec 2018.] <https://www.svethardware.cz/v-den-cisla-pi-prichazi-na-svet-novy-raspberry-pi-3-model-b/46248>.
65. **Buckley, Ian.** Everything you need to know about Raspberry Pi GPIO pins. *Make Use Of*. [Online] 21. Květen 2018. [Citace: 30. Prosinec 2018.] <https://www.makeuseof.com/tag/raspberry-pi-gpio-pins-guide/>.
66. **Raspberry Pi Foundation.** Raspbian. *Raspberry Pi*. [Online] 17. Leden 2018. [Citace: 30. Prosinec 2018.] <https://www.raspberrypi.org/downloads/raspbian/>.
67. **Bechynský, Štěpán.** Raspberry Pi 2 a Windows 10 IoT Core. *Zdroják*. [Online] 11. Květen 2015. [Citace: 30. Prosinec 2018.] <https://www.zdrojak.cz/clanky/raspberry-pi-2-windows-10-iot-core/>.
68. **GM Electronic.** PIR modul HC-SR501. [Online] 2019. [Citace: 1. Leden 2019.] <https://www.gme.cz/pir-modul-hc-sr501>.
69. **Arduino návody.** Teploměr a vlhkoměr DHT11 a DHT22. *Arduino návody*. [Online] 12. Červenec 2016. [Citace: 2. Leden 2019.] <https://navody.arduino-shop.cz/navody-k-produktum/teplotni-senzor-dht11.html>.
70. **Microsoft.** Windows Subsystem for Linux Documentation. *Microsoft Docs*. [Online] 11. Červenec 2016. [Citace: 15. Leden 2019.] <https://docs.microsoft.com/en-us/windows/wsl/about>.
71. **Ceder, Naomi.** *The Quick Python book*. 3. New York : Manning Publications, 2018.