



Zadání bakalářské práce

Název:	Prototyp chytrého zrcadla
Student:	Jakub Zahradník
Vedoucí:	Ing. Martin Daňhel, Ph.D.
Studijní program:	Informatika
Obor / specializace:	Počítačové inženýrství
Katedra:	Katedra číslicového návrhu
Platnost zadání:	do konce letního semestru 2021/2022

Pokyny pro vypracování

Seznamte se s trendy v oblasti chytrých zrcadel. Provedte analýzu dostupných open source projektů (např.: MagicMirror2, Home Assistant) zabývajících se chytrými zrcadly a senzory:

1. analyzujte vhodnost použitých skel
2. zaměřte se na možnosti ovládání - hlasové vs. dotykové
3. vyberte vhodný displej
4. analyzujte vhodný uživatelský systém

Na základě analýzy realizujte prototyp chytrého zrcadla:

1. zvolte vhodnou platformu chytrého zrcadla (Raspberry apod.)
2. vymyslete vhodnou konstrukci zrcadla s ohledem na budoucí použití (koupelna, garáž, pracovna apod.), vyberte vhodný rám
3. uvažujte možnosti budoucího rozšíření
4. uvažte možnost komunikace s jinými systémy např. telefonem

Realizovaný prototyp chytrého zrcadla bude umožňovat:

1. komunikaci se senzory v domácnosti na monitorování domácnosti (zabezpečení oken, řízení světel apod.)
2. zobrazování informací o počasí, dopravě atd. z internetu
3. zvolený způsob ovládání

Realizovaný prototyp otestujte v laboratorních podmínkách.

Elektronicky schválil/a doc. Ing. Hana Kubátová, CSc. dne 27. ledna 2021 v Praze.

Bakalářská práce

PROTOTYP CHYTRÉHO ZRCADLA

Jakub Zahradník

Fakulta informačních technologií ČVUT v Praze
Katedra číslicového návrhu
Vedoucí: Ing. Martin Daňhel, Ph.D.
13. května 2021

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2021 Jakub Zahradník. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bez uplatněných zákonných licencí nad rámec oprávnění uvedených v Prohlášení, je nezbytný souhlas autora.

Odkaz na tuto práci: Jakub Zahradník. *Prototyp chytrého zrcadla*. Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2021.

Obsah

Poděkování	vii
Prohlášení	viii
Abstrakt	ix
Seznam zkratek	x
1 Úvod	1
1.1 Motivace	1
1.2 Cíle	2
1.3 Aktuální stav	2
2 Analýza	3
2.1 Projekty chytrých zrcadel	3
2.1.1 DIY	4
2.1.2 Produkty dostupné na trhu	10
2.2 Skla	17
2.2.1 Technické specifikace	17
2.2.2 Porovnání	18
2.2.3 Časté otázky při výběru skel	18
2.2.4 Shrnutí	20
2.3 Displeje	21
2.3.1 Typy LCD panelů	21
2.3.2 OLED panely	22
2.3.3 Quantum Dots	23
2.3.4 MicroLED	25
2.3.5 Výběr vhodného displeje	25
2.3.6 Požadavky na monitor	26
2.3.7 Pořízení displeje	26
2.4 Platformy	27
2.4.1 MagicMirror ²	27
2.4.2 HomeAssistant	30
2.5 Možnosti ovládání	33
2.5.1 Dotykové ovládání	33
2.5.2 Hlasové ovládání	34
2.5.3 Alternativní ovládání	34
2.6 Volba hardware	36
2.6.1 Cena a zhodnocení výkonu	36
2.6.2 Požadavky na hardware	36
2.6.3 Kategorie	37

3	Návrh a realizace	41
3.1	Vhodné ovládání a konstrukce	41
3.1.1	Vhodné ovládání podle místností	41
3.1.2	Konstrukce	43
3.2	Instalace systému	48
3.2.1	Volba a natavení operačního systému	48
3.2.2	Instalace platformy HomeAssistant	50
3.2.3	Instalace platformy MagicMirror ²	52
3.2.4	Instalace neuronové sítě Kalliope	55
3.3	Konfigurace	57
3.3.1	Konfigurace HomeAssistant	57
3.3.2	Konfigurace MagicMirror ²	68
3.3.3	Konfigurace Kalliope	72
3.3.4	Aplikace na ovládání světla a samo-spouštěcí skript	76
3.4	Instalační příručka	79
4	Testování, rozšiřitelnost a náklady	81
4.1	Testování	81
4.1.1	Světelné podmínky	81
4.1.2	Rozpoznání klíčového slova	82
4.1.3	Zobrazované moduly	82
4.1.4	Systém jako celek	82
4.1.5	Výsledek	83
4.2	Rozšiřitelnost a náklady	83
5	Závěr	85
	Obsah přiloženého média	93

Seznam obrázků

2.1	Původní chytré zrcadlo od zakladatele platformy MagicMirror ²	5
2.2	Chytré zrcadlo ovládané hlasem přes platformu Jasper	7
2.3	Poškození skla po snaze o očištění od lesklých fleků	7
2.4	Dotykové chytré zrcadlo s FaceID a jeho autor	9
2.5	Uspořádání komponent chytrého zrcadla	9
2.6	Chytré zrcadlo Waveshare dostupné českém trhu	11
2.7	Bezrámečkové chytré zrcadlo <i>Frameless Smart Mirror</i> s elegantním designem	13
2.8	Kosmetické zrcadlo <i>HiMirror Mini Premium X 64GB</i>	15
2.9	Vodotěsné inteligentní zrcadlo (koupelňová televize)	16
2.10	Porovnání typů skel chytrých zrcadel	19
2.11	Porovnání průhlednosti zrcadlových skel	19
2.12	Vrstvy LCD displeje	23
2.13	Princip polarizace u LCD	24
2.14	Vrstvy TN panelu	24
2.15	Quantum dots u LCD a OLED	25
2.16	Možné způsoby instalace HomeAssistant	31
2.17	Architektura HomeAssistant s operačním systémem	32
3.1	Zadní strana 3D modelu chytrého zrcadla	44
3.2	Přední strana 3D modelu chytrého zrcadla	44
3.3	Rozebraný monitor	45
3.4	Uchycení monitoru k nosným profilům	45
3.5	Výsledek ohybu plechu pro horní panel konstrukce	46
3.6	Zkompletovaný horní panel konstrukce	46
3.7	Hotová konstrukce ze zadní strany, bez skla	47
3.8	Nastavení sítě v souboru <i>NetworkManager.conf</i>	49
3.9	Nastavení sítě pomocí <i>nmcli</i>	49
3.10	Integrace platformy do HomeAssistant	50
3.11	Úprava řádku v souboru <i>/etc/passwd</i>	52
3.12	Úprava řádku v souboru <i>/etc/group</i>	52
3.13	Úprava řádku v souboru <i>/etc/sudoers</i>	53
3.14	Úprava souboru <i>/etc/lightdm/lightdm.conf</i>	53
3.15	Úprava Makefile v <i>snowboy/swig/Python3</i>	55
3.16	Uživatelské rozhraní <i>Lovelace</i> platformy HomeAssistant	58
3.17	Integrace platformy do HomeAssistant ve formě „add-ons“	59
3.18	ESPHome „add-on“	60
3.19	Webové rozhraní ESPHome	61
3.20	Editor kódu zařízení platformy ESPHome	62
3.21	Fotografie zapojení obvodu s <i>NodeMCU v3</i>	63
3.22	Přidání nové karty s entitami v rozhraní <i>Lovelace</i>	64
3.23	Základní panel rozhraní <i>Lovelace</i> po přidání nové karty	65
3.24	Vytvoření autorizačního tokenu v HomeAssistant	66
3.25	Uživatelské rozhraní <i>Lovelace</i> v mobilní aplikaci HomeAssistant	67

3.26	Platforma ESPHome v mobilní aplikaci HomeAssistant	67
3.27	Ukázka sekce událostí na wikipedii, které se staly v aktuální den	71
3.28	Ukázka zdrojového kódu sekce událostí na wikipedii, které se staly v aktuální den	71
3.29	Schéma zapojení obvodu pro ovládání LED pásku	76
5.1	Vytvořený prototyp s autorem této práce	87

Seznam tabulek

2.1	Hodnoty optických vlastností skel – odrazivosti a průhlednosti	17
2.2	Výhody a nevýhody TN panelů	21
2.3	Výhody a nevýhody VA panelů	22
2.4	Výhody a nevýhody IPS panelů	22
2.5	Výhody a nevýhody OLED panelů	22
2.6	Výhody a nevýhody MicroLED panelů	25
2.7	Zařízení patřící do kategorie <i>low cost</i>	37
2.8	Zařízení patřící do kategorie <i>výkon</i> část 1.	39
2.9	Zařízení patřící do kategorie <i>výkon</i> část 2.	39
4.1	Orientační cena použitých komponent	83

Seznam výpisů kódu

3.1	Konfigurace modulu <i>MMM-kalliope</i> v <i>config.js</i>	69
3.2	Konfigurace modulu <i>MMM-HASS</i> v <i>config.js</i>	69
3.3	Úprava modulu <i>MMM-HASS</i> v <i>MMM-HASS.js</i>	70
3.4	Úprava modulu <i>MMM-HASS</i> v <i>node_helper.js</i>	70
3.5	Přidání češtiny do modulu <i>MMM-OnThisDay</i> v souboru <i>WikiCssSelectors.js</i>	71
3.6	Začátek souboru <i>brain.yml</i> v adresáři projektu <i>kalliope_starter_cs</i>	72
3.7	Synapse na odeslání notifikace pro zapnutí červené diody	73
3.8	Synapse na odeslání poslední „Text-To-Speech“ zprávy, aby se zobrazila na zrcadle	73
3.9	Synapse odpovědná za zapnutí přisvětlovacího světla u zrcadla	73
3.10	Synapse na odeslání notifikace pro přepnutí zobrazení na stránku 3	74
3.11	Neuron Kalliope pro ovládání GPIO portů na Raspberry Pi	75
3.12	Samo-spouštěcí skript pro automatické spuštění zrcadla	77
3.13	Aplikace pro ovládání přisvětlovacího světla u zrcadla	78

Chci upřímně poděkovat všem, kteří mi věnovali čas a poskytli pomoc při psaní této bakalářské práce. Především děkuji svému vedoucímu Ing. Martinovi Daňhelovi za odborné vedení, trpělivost a cenné rady, které mi pomohly práci zkompletovat. Dále chci srdečně poděkovat své rodině za podporu během celých mých studií. Také bych rád poděkoval Janovi Wijnhorstovi za diskuzi ohledně návrhu konstrukce a pomoc s jejím vytvořením, Dominikovi Reslovi za konzultace použitých platforem a všem mým kamarádům, kteří mě psychicky podporovali a motivovali.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací. Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 13. května 2021

.....

Abstrakt

Bakalářská práce se zabývá návrhem a realizací funkčního prototypu chytrého zrcadla. Analytická část práce se ve své první fázi věnuje detailnímu rozboru již hotových projektů, ať už ze strany technologických nadšenců či produktů dostupných na trhu. V této části jsou zhodnoceny možné přínosy či naopak negativa zkoumaných řešení, ať už z pohledu použitých materiálů, technologií či způsobu uživatelského ovládání. V další fázi analytické části uvažovány možné SW platformy a HW produkty, které by bylo možné pro realizaci chytrého zrcadla využít. Velká pozornost je věnována především výběru vhodných zobrazovacích komponent – jsou zde diskutovány různé typy skel a displejů. Praktická část této práce se zabývá tvorbou prototypu chytrého zrcadla. Konstrukce zrcadla je unikátní, bezrámečková a je navržena tak, aby uživateli nabídla co největší zobrazovací plochu pro informace a současně také moderní design s hlasovým ovládáním. Vytvořený prototyp je založený na platformě Paspberry Pi, jsou zde uvedeny rozšiřitelné moduly na ovládání světel. Následně jsou popsány kroky k instalaci jednotlivých softwarových platform HomeAsistant, MagicMirror² a hlasového asistenta Kalliope. Součástí práce je i postup, jak jednotlivé platformy konfigurovat, řešit případné problémy a je také naznačena cesta možného rozšíření či využití navrženého zrcadla v chytré domácnosti. Nakonec je prototyp důkladně otestován, aby byla zaručena funkční spolehlivost celého systému.

Klíčová slova chytré zrcadlo, inteligentní zrcadlo, IoT, internet věcí, chytrá domácnost, domácí automatizace, MagicMirror, HomeAssistant, Kalliope, Raspberry Pi

Abstract

This bachelor thesis deals with the design and implementation of a functional prototype of a smart mirror. The analytical part of this work is devoted to a detailed analysis of already completed projects, whether by technology enthusiasts or products available on the market. The possible benefits or negatives of the investigated solutions are evaluated in this analytical part of work, either from the point of view of the materials used, technologies or the way of user control. In the next phase of the analytical part, possible SW platforms and HW products are considered, which could be used for the implementation of a smart mirror. Great attention is paid especially to the selection of suitable display components - various types of glasses and displays are discussed here. The practical part of this work deals with the creation of a functional prototype of a smart mirror. The construction of the mirror is unique, frameless and is designed to offer the user the largest possible display area for information and at the same time a modern design with voice control. The created prototype is based on the Raspberry Pi platform, and also are listed here expandable modules for light control. The steps for installing the individual SW platforms HomeAssistant, MagicMirror² and the voice assistant Kalliope are described. Part of the work is also a procedure how to configure individual platforms, solve possible problems and also it is indicated the way of possible expansion or use of the designed mirror in a smart home. Finally, the prototype is thoroughly tested to ensure the functional reliability of the entire system.

Keywords smart mirror, IoT, Internet of Things, smart home, MagicMirror, HomeAssistant, Kalliope, home automation, Raspberry Pi

Seznam zkratek

API	Application Programming Interface
AR	Augmented Reality
BCM	Broadcom
CPU	Central Processing Unit
CSS	Cascading Style Sheets
DIY	Do It Yourself
DNS	Domain Name System
GPIO	General-Purpose Input/Output
GPU	Graphics Processing Unit
HD	High-Definition
HDMI	High-Definition Multimedia Interface
HDR	High Dynamic Range
HMI	Human Interface Machine
HTML	Hypertext Markup Language
HW	Hardware
IP	Internet Protocol
IP Code	International Protection Code
IPS	In-Plane Switching
IR	InfraRed
LCD	Liquid Crystal Display
LED	Light-Emitting Diode
OLED	Organic Light-Emitting Diode
OpenCV	Open Source Computer Vision Library
OS	Operační Systém
OTA	Over The Air
PC	Personal Computer
RAM	Random Access Memory
RGB	Red-Green-Blue
SD	Secure Digital
SSD	Solid-State Drive
SSID	Service Set Identifier
SW	Software
TF card	T-Flash card
TN	Twisted Nematic
TV	Televize
USB	Universal Serial Bus
VA	Vertical Alignment
Wi-Fi	Wireless Fidelity

Kapitola 1

Úvod

Co si představit pod pojmem chytré zrcadlo? Základní myšlenka je založena na displeji za polopropustným sklem. To využívá optických vlastností materiálů – odrazivosti a průhlednosti. Pokud je plocha za sklem tmavá, je vidět odraz. Pokud je za sklem nějaký zdroj světla, je vidět skrz. Aplikace zobrazují na displeji bílý obsah na černém pozadí, aby dosáhly co nejlepšího efektu zrcadla a zároveň byly informace dobře čitelné. Tento optický princip byl převzat z výslechových místností, odkud také pochází pojem špionážní sklo.

V této práci bude využita populární platforma pro chytrá zrcadla – MagicMirror², která bude propojena s platformou pro vytvoření vlastní chytré domácnosti – HomeAssistant. Výsledkem této integrace bude chytré zrcadlo, které se bude starat o chod chytré domácnosti a bude jejím centrálním prvkem.

Práce se dělí na tři hlavní kapitoly. První se zaměří na analýzu a teoretickou stránku chytrého zrcadla. To se skládá z dílčích komponent, které budou zkoumány z hlediska kvality, ceny a dostupnosti. Práce je zaměřena především na aktuální technologie v této oblasti. Druhá kapitola se bude zabývat návrhem konstrukce, instalací systému a platformem. Bude uveden způsob, jak propojit platformy HomeAssistant a MagicMirror² a neuronovou síť Kalliope. V práci bude vysvětleno, jak je nakonfigurovat. Bude ukázán jeden z možných způsobů nastavení, aby si je pak mohl každý upravit podle své libosti. V poslední kapitole bude vytvořený prototyp chytrého zrcadla podroben testování v laboratorních podmínkách. Testovat se bude přesnost a spolehlivost zvoleného způsobu ovládání a spolehlivost modulů. Dále bude zkoumána čitelnost zobrazených informací v různých světelných podmínkách.

1.1 Motivace

Vzhledem k tomu, že v dnešní době jsou vyvíjeny systémy pro ulehčení práce a zjednodušení života pro člověka na velkém vzestupu, chytré zrcadlo přináší mnoho výhod a usnadnění. Tento prvek dokáže komunikovat s chytrou domácností, nebo ji dokonce řídit a řadí se mezi IoT zařízení. Internet věcí je aktuálně stále vědní obor, který je na svých počátcích a má obrovský potenciál. Na internetu existuje mnoho návodů, jak si vytvořit vlastní chytré zrcadlo doma, ale většina z nich je v angličtině s použitím materiálů, které se, autorovi návodu, válely doma. Chybí v nich analýza komponent a ve většině případů se používá primitivní, ne příliš hezká, konstrukce.

Důvod proč se něčím takovým zabírat je prostý. Na světě je mnoho lidí, kteří tráví před zrcadlem hodně času. Ať už se jedná o lidi, kteří se na sebe rádi dívají, nebo o ženy, které se dlouho líčí. Což může mít za následky, že dotyčný nebude stíhat a přijde někam pozdě. Chytré zrcadlo nabízí jednoduché řešení, jak mít vše pod kontrolou. Potřebujete si najít spoj? Zjistit aktuální informace o dopravě? Vědět kolik je hodin? Přečíst si zprávy? Zjistit jaké je počasí? Mít přehled o aktivitách, které jste měli v kalendáři? To vše a mnoho dalšího může nabídnout chytré zrcadlo, díky kterému budete mít všechny informace, které jsou pro vás důležité, po ruce.

Komu by to hlavně mohlo pomoci a kdo všechno by z toho mohl něco mít? Lidé, kteří vstávají brzo ráno a spěchají. Lidé, kteří jsou časově zaneprázdnění, chtějí dobře vypadat na nějaké akci a nestíhají. To jsou ti, pro něž by chytré zrcadlo mělo největší výhody. Hlavní myšlenka je seskupit věci, které lidi zajímají a zabírají jim čas, do nějakého paralelismu. Ten by jim ten čas ušetřil. Při úpravě před zrcadlem je možné pohodlně přejet pohledem o pár centimetrů vedle hlavy a dozvědět se informace, které by bylo nutné zjišťovat jiným způsobem.

Chytré zrcadlo jsem chtěl vytvořit už delší dobu, ale nikdy jsem nenašel odhodlání k tomu, abych si zjistil potřebné informace a pustil se do toho. Nikdy jsem nepracoval s Raspberry Pi a tento projekt je vhodný pro to, abych se s ním naučil, zjistil, jak se s Raspberry Pi pracuje a co dokáže. Tato práce mi přinese nové zkušenosti a znalosti.

1.2 Cíle

Mým primárním cílem při tvorbě této práce je analyzovat komponenty, které jsou základními stavebními prvky pro vytvoření chytrého zrcadla a zamyslet se nad vhodnou konstrukcí. Následně vybrat vhodné produkty a vytvořit prototyp, který bude zobrazovat informace a dokáže komunikovat s prvky chytré domácnosti. To vše s cílem maximálně snížit cenu při zachování rozumné kvality. K dosažení tohoto cíle je potřeba zaměřit se na dílčí úkony.

Počátečním krokem je vypracování teoretické části práce, která se zabývá přehledem současných znalostí a poznatků k danému tématu. Vzhledem k tomu, že nejsem první s touto myšlenkou, je potřeba zjistit postup řešení z již existujících projektů, aby bylo možné se vyvarovat problémům, které již někdo vyřešil. Aby mělo smysl vytvářet vlastní zrcadlo, musí být jeho cena nižší než cena chytrých zrcadel na trhu. Z tohoto důvodu bude proveden průzkum současných produktů, které je možné koupit. Následně budou porovnány jejich vlastností a ceny s uvažovaným řešením.

V další části je potřeba prozkoumat současné technologie skel. Následně provedu rozbor polopropustných skel, které využívají optických vlastností skla a dokážou světlo jak propouštět, tak odrážet. Aby bylo co propouštět, je potřeba displej. Také budu muset provést analýzu, jaké vlastnosti nabízejí dnešní monitory a bude popsáno, jaké parametry jsou pro chytré zrcadlo klíčové. Ačkoliv bude v této práci použito zařízení Raspberry Pi 4 Model B, budou porovnány i jeho alternativy.

Jeden z cílů je vytvořit kvalitní a elegantní bezrámečkovou konstrukci. Nejprve je potřeba se zaměřit na materiál konstrukce, vytvořit model a nakonec zvolenou konstrukci sestavit. Konstrukce musí být navržena, aby působila přirozeně a bylo možné zrcadlo zavěsit na zeď. Ve výsledku bude vytvořen prototyp s touto konstrukcí, který bude podporovat hlasové ovládání.

Hlavní záměr této práce je propojit platformy HomeAssistant a MagicMirror tak, abych vzniklo chytré zrcadlo, které bude fungovat jako centrální prvek chytré domácnosti. Výsledný prototyp bude umožňovat zobrazovat informace a bude cenově dostupný. Jelikož **neexistuje pořádný tutoriál, jak dát tyto dvě platformy dohromady**, cílem této práce je jeden takový vytvořit v českém jazyce. Tak aby mohl kdokoliv, se zájmem o vytvoření chytré domácnosti a chytrého zrcadla, postupovat podle tohoto návodu a dopracovat se ke stejnému výsledku. Tuto práci lze využít jako kostru pro další řešení.

1.3 Aktuální stav

Projekt se aktuálně nachází ve stavu, kdy platformy – MagicMirror² a HomeAssistant – je možné zprovoznit samostatně. První zmíněná je postavena na operačním systému Raspbian OS, kdežto HomeAssistant v režimu supervised lze nainstalovat pouze na čistý Debian 10 (Buster). Bylo zde pár pokusů od komunity těchto dvou platforem se snahou je propojit, ale výsledné řešení není ani jednou platformou podporováno, protože doposud nikdo nedal k dispozici návod, jakým způsobem to udělat. Šlo pouze o pokusy a řešení technicky zdatných nadšenců, které často narazili na problém při nových verzích ať už jedné, nebo druhé platformy. Tato práce se bude zabývat propojením těchto platforem a vytvořením návodu, který k tomu chybí.

Kapitola 2

Analýza

V této kapitole budou podrobně rozebrány **projekty chytrých zrcadel**, které někdo vytvořil. Často se jedná o dokumentaci jejich postupu, která má sloužit jako průvodný návod, kdyby se chtěl někdo inspirovat a vytvořit si vlastní chytré zrcadlo. Následuje popis a zhodnocení **produktů**, které jsou na trhu k dispozici a může si je kdokoliv koupit.

Další část bude věnována **sklu**. Jedná se o hlavní komponentu, která určuje, jak bude chytré zrcadlo kvalitní a jestli vůbec bude použitelné. Budou analyzována nejčastěji použitá skla pro chytrá zrcadla a použití polopropustné odrazové folie.

Jedna sekce bude porovnávat **typy displejů**. Nejprve bude popsáno jak fungují LCD (liquid crystal display). Práce se zaměří na výhody a nevýhody jednotlivých LCD panelů, porovnání s panely OLED a microLED. Bude vysvětlena technologie *Quantum dots* a čeho jejím použitím lze dosáhnout. Nakonec na základě těchto informací bude vybrán nejlepší typ displeje pro tuto práci a budou shrnuty požadavky na monitor pro chytré zrcadlo.

Budou představeny způsoby interakce se zrcadlem. U jednotlivých **možností ovládní** budou popsány hlavní výhody a nevýhody. Dále bude vysvětleno jak funguje infračervená dotyková plocha, jeden z nejčastějších způsobů, jak přidat dotykové ovládní na displej, hlavně u chytrých zrcadel. U hlasového ovládní budou popsány potenciální problémy s bezpečností. Nakonec budou zmíněny alternativy pro hlasové a dotykové ovládní a co budoucnost ještě může v tomto ohledu nabídnout.

V poslední části budou srovnány vybrané jednodeskové počítače. Budou rozděleny do dvou cenových kategorií. Jedné s orientací na co nejnižší cenu, druhé pohybuující se na stejné hodnotě jako Raspberry Pi 4. U všech zařízení budou zmíněny parametry, kterými disponují. Podle požadavků na **hardware** bude vyhodnoceno, jaké produkty lze použít a jaké se pro tuto práci nehodí.

2.1 Projekty chytrých zrcadel

Následující část pojednává o projektech, které vytvořili nadšenci a produktech dostupných na trhu, které si může kdokoliv pořídit. Nejprve popisuje výtvořky od lidí, kteří si vytvořili chytré zrcadlo sami. Jejich projekty jsou pojaty jako návody, aby si ostatní mohli vytvořit vlastní chytré zrcadlo. Jedná se tedy o jejich konkrétní postup. Každý klade důraz na něco jiného. Někdo se zabývá výběrem kvalitních komponent, někdo použije komponenty, které se mu vají doma, aby minimalizoval cenu. Jiný zase použije jím naprogramované moduly do systému a upraví tak jeho funkčnost dle svých potřeb.

U produktů se práce zaměřuje na velikost zrcadla, cenu a jeho funkce. Zkoumá se design, provedení, zvolený způsob ovládní a umístění případných komunikačních zařízení. Jsou popsány hlavní výhody a nevýhody a přidávám vlastní posudek, jak na mě zrcadlo působí.

2.1.1 DIY

Zde bude výčet projektů tzv. DIY (Do It Yourself = udělej si sám). Většinou se jedná o projekty zachycující postup konkrétního tvůrce, který má sloužit jako návod pro případné zájemce, kteří by si chtěli vyrobit vlastní chytré zrcadlo. Je popsáno čím se autoři zabývali a jak postupovali. Ke každému projektu mám vlastní shrnutí, ve kterém řeším, co dotyční udělali dobře a čemu nevěnovali pozornost, či kde nezvolili vhodné řešení.

2.1.1.1 MagicMirror

Jeden z prvních projektů chytrého zrcadla, který je dílem Michaela Teeuwa. První zpracování projektu bylo uvedeno jako sada příspěvků na jeho blogu. Michael nečekal, že se jeho projekt stane tak populárním a že si z něj lidé budou brát příklad. Z tohoto důvodu později vytvořil open-source projekt, který pojmenoval MagicMirror² a dal tak prostor ostatním, aby se podíleli na vývoji systému pro chytré zrcadlo.

Jeho původní práce je rozdělena na šest příspěvků, kde vysvětluje, jak k samotnému nápadu přišel a postupně dal dohromady, co bylo potřeba. Při výběru skla zmiňuje, že by se mělo chovat jako zrcadlo, pokud je obrazovka za ním černá a jako sklo, pokud je zobrazena nějaká informace. Došel tak k závěru, že vhodné bude polopropustné sklo, někdy také známé jako špionážní zrcadlo. V angličtině tomu odpovídají pojmy one-way mirror, také jinak two-way mirror. Princip takového zrcadla je založen na optických jevech skla. Pokud by zrcadlo bylo mezi dvěma místnostmi, jedna by byla osvětlena, druhá ne, při pohledu ze světlejší místnosti do tmavé působí sklo jako zrcadlo, ale z temnější je do světlé vidět. [1]

Na základě svých měření volí monitor s úhlopříčkou 24", protože se jedná o poměrně levné monitory s nativním rozlišením 1080 P. Během hledání vhodného produktu zjistil, že vyvedení konektorů nesmí být na zadní straně, protože by to přidávalo na tloušťce zrcadla. Odstraněním přebytečných plastů a dalších nepotřebných částí minimalizoval velikost monitoru a rozhodl se orientovat displej na výšku. [2]

Na rám použil borové dřevo a na vyčnívající rámeček, který drží sklo a monitor „uvnitř“, podlahové lišty. Michael zmiňuje, že jeho výtvar dohromady váží 6,5 kg, je tedy důležité, aby konstrukce byla dost pevná na to, aby takovou tíhu vydržela. Do horní části vyvrtal otvory pro odvod teplého vzduchu, který monitor vydává. [3]

Pro samotný chod chytrého zrcadla zvolil Raspberry Pi. Poté, co si přichystal všechny kabely, zjistil, že by musel vést kabel na napájení Raspberry Pi a na monitor zvlášť. To se mu ale nelíbilo, a tak upravil adaptér pro napájení tak, aby ho mohl propojit s kabelem pro napájení monitoru. [4]

Poslední dvě kapitoly se věnují nastavení software. Použil webový server, aby mohl své zrcadlo zobrazit jako webovou stránku. Upravil konfiguraci Raspberry OS, tak aby vyhovovala jeho podmínkám – žádný spořič obrazovky, režim portréту, spuštění Chromia ihned po bootu zobrazující lokální webový server ve fullscreen. [5, 6]

2.1.1.2 Shrnutí

Jelikož se jedná o základní projekt v oblasti chytrých zrcadel, spousta věcí již byla v dnešní době vylepšena. Michael dostal nápad a chtěl ho zrealizovat, dal dohromady své myšlenky, ale nijak více se nad výběrem komponent nepozastavil. Oceňuji, že vymyslel způsob, jak vést pouze jeden kabel do zrcadla. Jedná se sice o způsob, který vyžaduje, aby byl proveden někým zkušenějším, ale je to rozhodně levnější než kupovat předražené adaptéry, nebo redukce. Upozornění na monitory, které mají konektory vyvedené dozadu bylo přínosné. Stejně tak zmínění otvorů pro odvod teplého vzduchu. Další věc, kterou Michael uvedl je váha. Ta bude sloužit jako orientační při volbě vhodné konstrukce pro tuto práci. Mezi hlavní nevýhody patří nemožnost jakkoliv ovládat zrcadlo, protože není implementované HMI (Human Machine Interface). Informace jsou zobrazené staticky a jedná se o čtyři základní údaje – komplement, počasí, hodiny s kalendářem a

novinky ze zpráv. Jelikož je vypnutý spořič obrazovky, monitor i Raspberry Pi jsou v neustálém provozu a celkové zařízení má vyšší spotřebu. Při nejvyšším jasu, který je u těchto zrcadel potřeba, je nutné brát nejvyšší možnou spotřebu monitoru jako provozní.



■ **Obrázek 2.1** Původní chytré zrcadlo od zakladatele platformy MagicMirror², Michaela Teeuwa, které zobrazuje čtyři základní údaje – kompliment, počasí, hodiny s kalendářem a novinky ze zpráv [7]

2.1.1.3 Voice-Controlled Raspberry Pi Smart Mirror with Jasper

Projekt začíná výčtem věcí, které jsou pro projekt potřeba. Mezi nimi se nachází obal na Raspberry Pi. Nicméně není potřeba používat kryt pro zařízení, které je schované v rámu a není vidět.

Zach popisuje, že hlasem ovládané chytré zrcadlo je tvořeno ze čtyř hlavních komponent:

- Polopropustné sklo – akrylové sklo by mělo přímo doléhat na monitor, díky tomu je možné vidět obraz na monitoru a zároveň mít zrcadlový efekt; doporučuje objednat 4,5 mm tlusté sklo, aby nedocházelo k prohýbání, které může způsobit deformovaný odraz
- Monitor – uvádí, že optimální je využít LED monitor kvůli nízké spotřebě a kvalitě obrazu, doporučuje volit nějaký se zabudovanými reproduktory
- Raspberry Pi – malý počítač o velikosti kreditní karty, na kterém poběží Magic Mirror interface a Jasper, systém pro hlasové ovládání
- Rám – ukrývá veškeré komponenty a je v něm zabudovaný mikrofon

Celková cena zahrnující dřevo, sklo, monitor a Raspberry Pi vychází na \$300–350 (6606–7707 Kč).

Veliká část jeho projektu se věnuje samotné tvorbě rámu. Ten je tvořen čtyřmi dřevěnými prkny, které tvoří obal, spojenými se čtyřmi dalšími, která tvoří přední rám. Opět jsou vytvořeny ventilační průduchy, aby se teplý vzduch nehromadil uvnitř. Po zhotovení rámu Zach varuje, že je důležité si dát pozor, aby při odlepování ochranné folie z povrchu skla nedošlo k jeho ohnutí. Je dobré si sklo nějakým způsobem uchytit a zbavit se folie opatrně, jinak hrozí deformace odrazu.

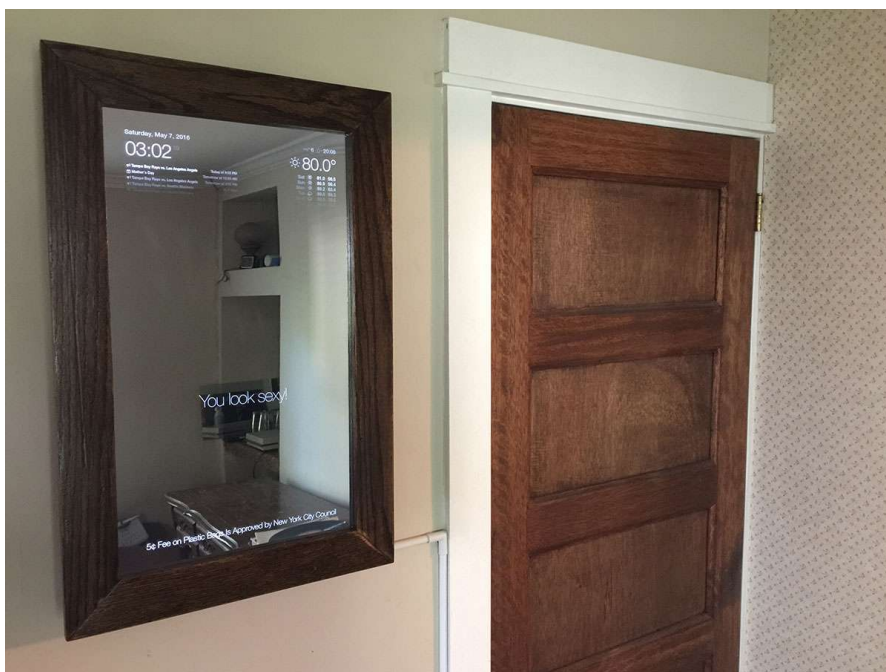
Díru pro umístění mikrofonu vyvrtal dole, z důvodu, že tam mikrofon nebude vidět. Zmiňuje, že ho nechtěl dávat zepředu, protože by to zkazilo vzhled zrcadla, i přesto, že by to byla lepší pozice. Původní myšlenka Zacha byla použít Bluetooth reproduktor. Bohužel si věci špatně změřil a zjistil, že se mu tam jeho reproduktor nevejde. Místo něj využil integrovaných reproduktorů v monitoru s tím, že do budoucna připojí reproduktor přes Bluetooth a dá ho vedle zrcadla.

Zach chtěl zkrášlit vzhled svého rámu. Použil lak na dřevo. Aby neupatlal sklo zrcadla, nalepil na něj maskovací pásku. Při odlepení vznikly na povrchu akrylového skla šmouhy, které chtěl odstranit. To se mu nepovedlo tak, sklo poškodil a musel objednat jiný kus.

Když bylo zrcadlo zkompletováno, zbývalo dodat software. Aby se dalo ovládat hlasem, využil open-source platformu Jasper. Udělal samostatný návod, jak takovou platformu integrovat do Raspberry Pi (Dostupné z [8]). Popisuje i rozdíl oproti hlasovému asistentovi Alexa od Amazon. Asi nejdůležitější je zmínit, že Jasper může běžet v offline módu a Alexa vyžaduje pro použití vývojářský klíč. Zach plánuje využití hlasové platformy Alexa, která se nedávno otevřela pro veřejné využití. Pro samotný systém využil platformu MagicMirror² od zakladatele projektu Michaela Teeuwa, jehož prvotní projekt je zmíněn výše. [9]

2.1.1.4 Shrnutí

Projekt je velmi přínosný díky chybám, kterých se autor dopustil. Upozorňuje na ně a radí čemu se vyvarovat. Rám je velmi mohutný a hluboký, na některá místa by se nemusel hodit. Cena tohoto projektu je poměrně vysoká. Využití platformy pro hlasové ovládání Jasper se zdá být atraktivní řešení, které nabízí možnost fungovat offline. Díky autorově chybě se ví, že na akrylové polopropustné sklo neměly lepit žádné pásky, může se tím poškodit sklo. Umístění mikrofonu není vhodné, ani použití reproduktorů zabudovaných v monitoru. Mezi výhody patří možnost ovládat zrcadlo hlasem a nemít tak pouze statické informace. Zach chtěl využít Bluetooth reproduktor, který nakonec nebyl součástí jeho projektu, ale jedná se o dobré řešení zvuku. Dále se mi líbí text-to-speech, díky kterému není třeba věnovat pohled na informace, ale můžeme si je poslechnout. Vybraná platforma MagicMirror² se zdá být nejvhodnější volba pro jakýkoliv projekt chytrého zrcadla.



■ **Obrázek 2.2** Chytré zrcadlo ovládané hlasem přes platformu Jasper posazené v robustní dřevěné konstrukci; velikostní porovnání s dveřmi [9]



■ **Obrázek 2.3** Celkové poškození skla odvíjející se od použití maskovací pásky při zakrytí skla během lakování dřevěného rámu; po odlepení došlo k vytvoření lesklých fleků na skle a když se je autor snažil odstranit, poškodil sklo zrcadla [9]

2.1.1.5 Smart Mirror Touchscreen with FaceID

Cílem tohoto návodu je vytvořit chytré zrcadlo, které slouží jako hlavní dotykové rozhraní pro ovládání chytré domácnosti. Poskytuje aktuální zprávy, zobrazuje vývoj na akciovém trhu, umí rozpoznat obličej apod. Projekt je založen na platformě MagicMirror² a je nazýván Smart Mirror AI (SMAI) (Videonávod dostupný z [10]).

Na začátku projektu se nachází výčet standardních komponent pro chytré zrcadlo. O chod systému se stará Raspberry Pi 4 Model B 2 GB. Cena není blíže určena, protože záleží na zvolené velikosti skla. Ukázkový model v tomto návodu je sestaven z 32" monitoru, na jehož povrch přiléhá polopropustné sklo. Na rámu displeje je přidělaný infračervený rámeček, který se stará o dotykovou interakci se zrcadlem. K Raspberry Pi je připojen přes USB 3.0. Některé infračervené rámečky také podporují multi-touch a gesta, to by se mohlo hodit do budoucna.

Rám je vytvořen ze dřeva tak, aby pojmul 32" monitor, nicméně Eben říká „fantazii se meze nekladou“. Na zadní části byl přidán popruh, který zajišťuje podporu rámu. Na monitor byly zezadu nalepeny LED pásky, které přidávají ambientní osvětlení. Použitý monitor má konektory umístěné vzadu, a tak tvůrce projektu zpětně doporučuje, aby byl použit monitor s konektory vyvedenými do strany.

V softwarové části se opět vyskytuje návod na zprovoznění Raspbian OS a instalaci platformy MagicMirror². Čím se ale tento projekt liší od ostatních, je integrace vlastních vytvořených modulů. Zejména modul *SmartTouch*, který umožňuje transformovat chytré zrcadlo na „normální“ zrcadlo. Další specifický modul pro tento projekt slouží na rozpoznání tváře. Je založený na knihovně *OpenCV*. Je responzivní a hodí se pro spouštění ostatních procesů. Pro funkčnost modulu jsou potřebné tři soubory:

1. SMAI-face-detection.py – python skript, který se neustále snaží rozpoznat osoby na základě poskytnutých obrázků; jediný program pro vývoj dalších *use-case*, který tento projekt neobsahuje
2. MMM-Face-Recognition-SMAI.js – zobrazuje obrázek identifikované osoby a její jméno
3. Sample.txt – dočasný buffer pro uložení jmen osob ze skriptu pro zobrazovací modulem

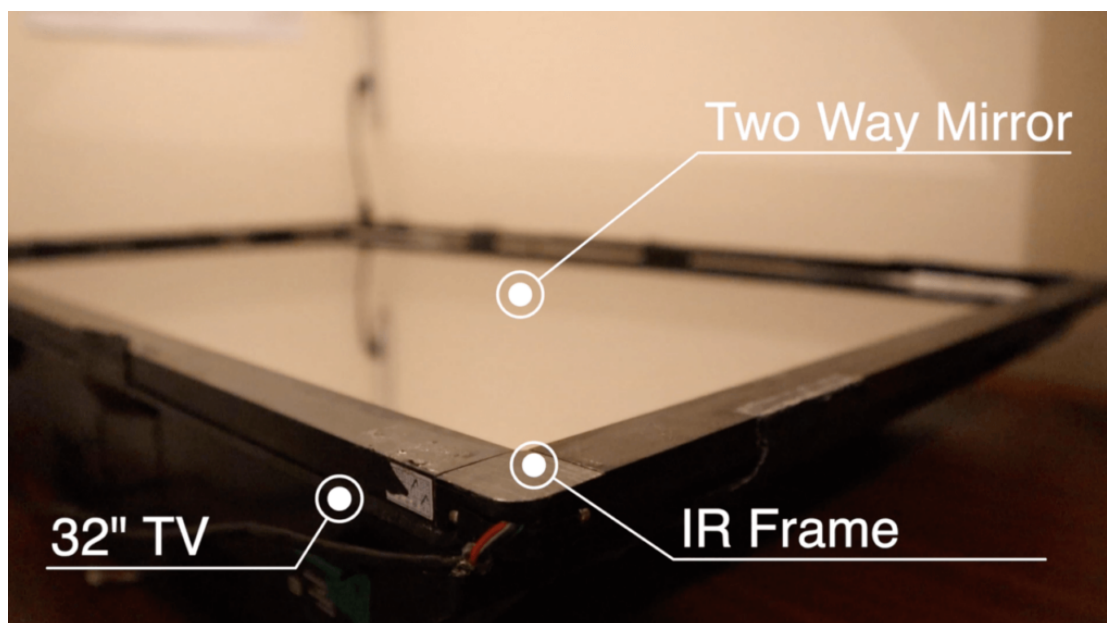
Změna profilu ještě není kompletní. Jedná se o beta verzi, která je omezená pouze na jednoho uživatele. [11]

2.1.1.6 Shrnutí

Jedná o jeden z pokročilejších projektů, který necílí na začátečníky, ale na zkušenější jedince a nadšence pro prvky chytré domácnosti. Není rozebráno, jakým způsobem byly vybrány komponenty a určitě by bylo vhodné zmínit, že pro správnou funkčnost dotykového ovládání je potřeba mít stejně velký infračervený rámeček, jako velikost displeje. Rám působí nedokončeně a celkový vzhled zrcadla není moc hezký. Je přínosné, že ukázal, jak funguje jeho vlastní modul a dal k dispozici návod, jak ho nainstalovat a uvedl, jak případně pracovat na jeho použitelnosti i za hranicemi jeho projektu. Jedná se o solidní základ pro zrcadlo s dotykovým ovládáním, které umí na základě rozpoznání tváře dělat nějaké akce. Dalo by se na to navázat v případě nějaké domácí automatizace. Personalizované rozložení a změna profilů je pro chytré zrcadlo dobrá vlastnost. Nejen kvůli komfortu v rámci rodiny, ale také kvůli bezpečnosti, aby se například nevhodná návštěva nedostala k řídicímu panelu chytré domácnosti, k čemuž by tento projekt mohl směřovat.



■ **Obrázek 2.4** Dotykové chytré zrcadlo s FaceID, které díky obrazu z kamery a využitím platformy OpenCV detekuje obličej a nastaví podle toho profil uživatele, či provede nějakou akci [11]



■ **Obrázek 2.5** Uspořádání komponent chytrého zrcadla – Televize/monitor, přes něj polopropustné sklo a infračervený rámeček umožňující dotykové ovládání zrcadla [11]

2.1.2 Produkty dostupné na trhu

V této části budou chytrá zrcadla, která jsou k dispozici a je možné si je koupit. Nejprve bude popsáno, co všechno zrcadlo nabízí a jaké má funkce. U každého produktu je mé vlastní shrnutí výhody a nevýhody.

2.1.2.1 Waveshare Magic mirror

Velikost zrcadla: 32,95 × 20,79 cm (685 cm²)

Cena: 5 499 Kč

Vlastnosti dostupné z [12]:

- 13,3" IPS obrazovka, 1920 × 1080 rozlišení
- Založeno na Raspberry Pi 3A+
- Jednosměrné zrcadlo, umí zobrazovat zprávy, počasí a čas
- Tvrzené sklo s kapacitně dotykovým panelem, tvrdost 6H, podporuje 10bodový dotyk
- Vestavěný feritový Hi-Fi reproduktor
- Vestavěný mikrofon a reproduktor, založený na Snowboy hotword detection engine
- Pro rozpoznání angličtiny používá Google Assistant
- 75 × 75 mm montážní otvory (M4) pro připevnění na zed
- Součástí je 75° stojánek

2.1.2.2 Shrnutí

Jedná se o zařízení vytvořené na stejné bázi jako tento projekt. Bude tedy sloužit jako referenční bod. Úhlopříčka zrcadla (13,3 palců) je poměrně malá. V této práci bude využita větší plocha. Hlavní cíl bude nepřesáhnout cenu tohoto chytrého zrcadla. Pokud se tak stane, cenový rozdíl by neměl být větší než pořízení většího displeje.

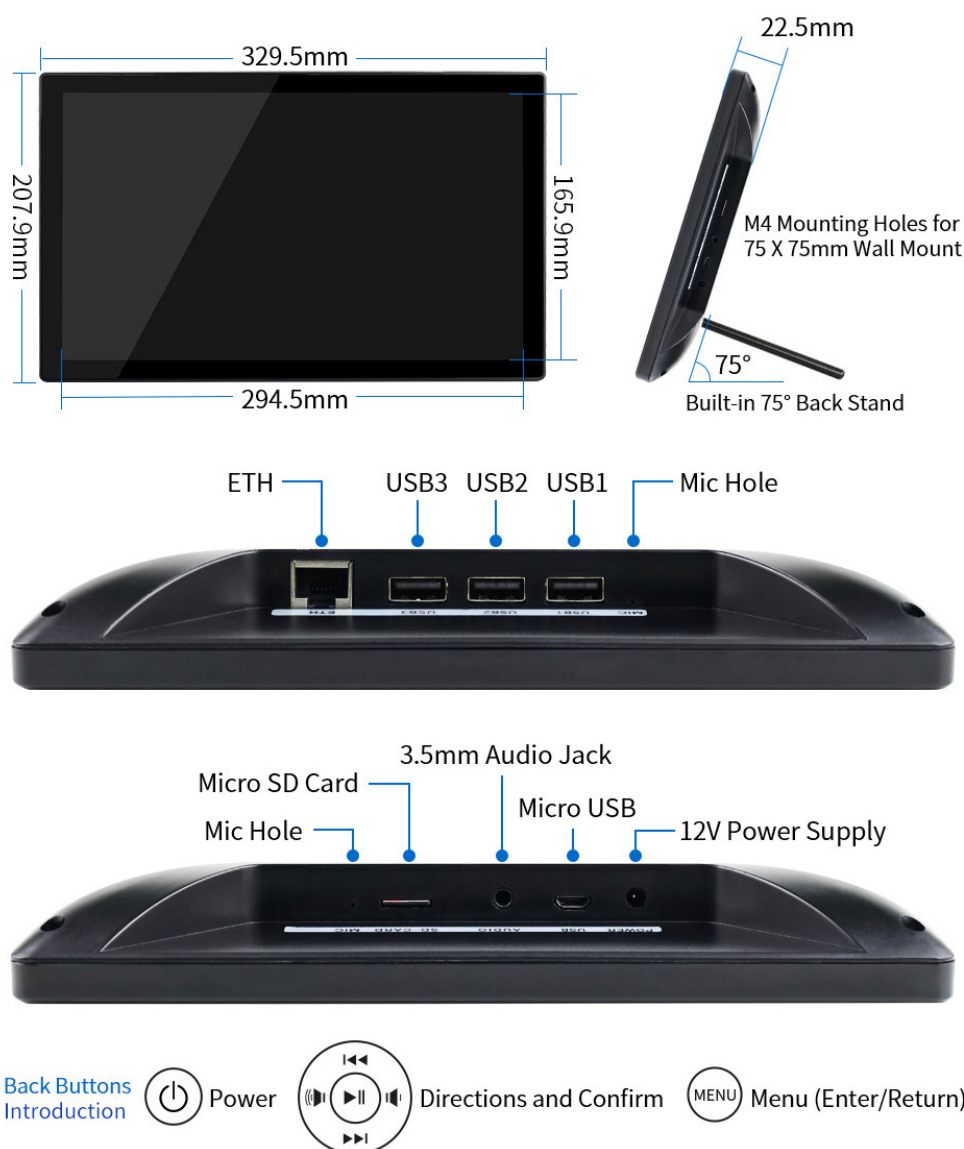
U Waveshare určitě není omezené zobrazování zpráv, počasí a času napevno. Pokud uživatel upraví nastavení, může z produktu vytěžit více. To je pouze pro nadšence, kteří tomu rozumí. Běžný uživatel se musí spokojit s tím, že má pouze základní výše zmíněné funkce.

2.1.2.3 Výhody

Je potřeba pochválit konstrukci a zpracování. Zrcadlo dle fotek vypadá, že má tenké rámečky a opravdu působí, že se jedná o zrcadlo. Díky své velikosti je snadno přenosné, a tak nemusí být jen pověšené na stěně. Opěrný stojánek umožňuje umístit zrcadlo téměř kamkoliv. Připojením klávesnice a myši pak uživatel získá plnohodnotný počítač se systémem MagicMirrorOS. Je potřeba ale počítat s tím, že Raspberry Pi 3 model A+ 512 MB RAM, které se stará o chod tohoto zařízení nenabídne velký výkon. Stačí to na jednoduché brouzdání internetem v podobě čtení článků. Na něco jako je například sledování videí na YouTube je možné rovnou zapomenout.

2.1.2.4 Nevýhody

Lehce nedomyšlená jsou komunikační zařízení, která jsou vestavěná vzadu za zrcadlem. Z teoretického hlediska nelze vyloučit problémy s hlasovým ovládáním jak při analýze řeči, tak při přehrávání zvuku. Jedná se hlavně o problém při pověšení na stěně.



■ **Obrázek 2.6** Chytré zrcadlo Waveshare prodávající se na českém trhu; rozměry, konstrukce, konektory pro připojení periferií, tlačítka, která jsou k dispozici ze zadní strany zrcadla [12]

2.1.2.5 Frameless Smart Mirror

Velikost zrcadla: 60 × 80 cm (4800 cm²)

Cena: 19 056 Kč (cena produktu £ 450 + poštovné £ 37,79 + clo a další poplatky £ 137,25)

Bezrámečkové chytré zrcadlo s elegantním a čistým designem. Zrcadlo je možné zavěsit na výšku v režimu portrétu. V režimu na šířku (režim krajina) se zrcadlo dodává pouze na objednávku. Tento produkt používá vysoce kvalitní 5 mm polopropustné sklo a zahrnuje 10" displej pro zobrazení informací. Jedná se o tablet s operačním systémem Android. Výrobce poskytuje aplikaci *Smart Mirror App*, která zobrazuje dashboard informací pro uživatele. Je možné využít poskytnutého Bluetooth ovládání ke spuštění jakékoliv aplikace pro Android (YouTube, Netflix,

RadioPlayer atd.). Zabudované hlasové ovládání s asistentem Alexa umožňuje komunikovat se zrcadlem a ptát se na informace, nebo řídit kompatibilní chytrá zařízení jako jsou zásuvky a světla.

Dostupné informace, které mohou být zobrazeny poskytovanou *Smart Mirror App*:

- Datum a čas
- Lokální počasí
- Webovou stránku dle vlastního výběru
- Informace o hustotě dopravy na silnicích
- Informace o vlakové dopravě
- Zobrazit (sdílet) Iphone, nebo android telefon na obrazovku
- Nadpisy v novinách
- Google kalendář
- Alexa Voice Control

MySmartMirror, aplikace s cloudovou synchronizací, umožňuje nastavit displej chytrého zrcadla z jiného Android zařízení, aniž by byla potřeba se ho dotýkat. Integrace asistenta *Alexa* dovoluje ovládat zrcadlo hlasem. Zrcadlo není vybaveno funkcí pro nepřetržitý poslech (always listening voice control). *Časová kontrola* – tato funkce nabízí možnost si vybrat, kdy se jaká informace bude zobrazovat. Například zobrazit vlakovou dopravu pouze ráno, nebo vypnout zrcadlo na noc. Zabudovaný reproduktor umožňuje poslouchat hudbu, televizi, nebo nějaké video. Poskytnutá mini Bluetooth klávesnice pak dává uživateli plnou kontrolu nad funkcemi Android zařízení. [13]

2.1.2.6 Shrnutí

Zrcadlo se ovládá hlasem a o jeho chytrou stránku se stará tablet. Ten zobrazuje aplikaci poskytnutou výrobcem. Vše běží na operačním systému Android. Nabízí se otázka jak jsou řešené „update“ operačního systému a co dělat když se tablet vypne. Není jasně specifikované uchytení tabletu. Nelze říci, jestli jde tablet od zrcadla oddělit. Pokud ano, v jakém je stavu, jestli ho lze používat samostatně. Výrobce ani neuvádí jeho parametry. Aplikace chytrého zrcadla je pravděpodobně na konkrétním tabletu testována a ošetřena proti zasekávání.

2.1.2.7 Výhody

Konstrukci se opět nedá nic vytknout. Jedná o prvotřídní ukázkou, toho, jak by chytré zrcadlo mělo vypadat – žádné viditelné rámečky. Díky využití tabletu, je zrcadlo tenké a nepůsobí dojmem, že něco skrývá. Naopak je elegantní a přirozené, nijak nevyčnívá.

2.1.2.8 Nevýhody

Zrcadlo samo o sobě stojí 13 718 Kč (£ 450). Když se k tomu připočítají náklady a poplatky spojené s dovozem ze Spojeného království, cena se vyšplhá vysoko a není přijatelná pro běžného uživatele. Za takovéto peníze dostane zrcadlo, které umí to, co dokáže jeho mobilní telefon a malou zobrazovací plochu, na které nebude možné zobrazovat větší množství informací. S možností ovládat zrcadlo pouze hlasem, by se hodilo mít někde lépe umístěný mikrofon, stejně tak i reproduktor.



■ **Obrázek 2.7** Bezrámečkové chytré zrcadlo s elegantním designem a tenkým provedením (takřka nerozeznatelné od běžného koupelnového zrcadla); o informace se stará tablet s OS Android a aplikací chytrého zrcadla, kterou poskytuje výrobce [13]

2.1.2.9 HiMirror Mini Premium X 64GB

Velikost zrcadla: 28,8 × 16,2 cm (466,58 cm²)

Cena: 8 808 Kč (cena produktu \$246 + poštovné \$73,18 + clo a další poplatky \$82,52)

„Nejvíce prémiový kosmetický zážitek“, takto se prezentuje produkt *HiMirror Premium X*. Informace dostupné z [14]. Jedná se o kosmetické zrcátko s možností otočit displej do režimu portrétu, nebo režimu krajina. Obsahuje bezdrátovou nabíjecí podložku, 64 GB paměti a podporuje až čtyři uživatelské účty. Zrcadlo se dodává společně se stylusem a čistícím hadříkem pro udržení skla bez otisků prstů a šmouh. Mezi přednosti toho zrcadla patří chytrá péče o pleť, která se skládá ze tří kroků:

1. Analýza pleti

HiMirror provede analýzu stavu pleti skrze fotku. Ty jsou uloženy stejně jako naměřená data tak, aby umožnily zobrazit vývoj v čase a informovaly uživatele na jaká místa je pleti je potřeba se zaměřit. Pokročilá technologie analýzy pleti – *Skin Analysis Engine X* – posoudí,

jestli se na pleti nachází vrásky, jemné linie (záhyby), tmavé kruhy, černé tečky, červené tečky, hrubá pokožka nebo póry.

2. Sledování péče o pleť

Tato část se zaměřuje na efektivitu produktů na pleť. Monitorováním pokožky po nějakou dobu lze zjistit jaké produkty opravdu pomohly.

3. Správa produktů péče o pleť

Prohlédnutí a sledování všech produktů péče o pleť ke zjištění, které přináší a které nepřináší kýžené výsledky. Zrcadlo dokonce upozorní, kdy jednotlivým produktům končí expirace. Systém se taktéž stará o personalizované doporučení výrobků z databází od Amazon, SkinSAFE a HiMirror.

Mezi další zajímavé vlastnosti tohoto zrcadla patří:

- Makeup studio – možnost okamžitě vyzkoušet nové makeupy s pomocí technologie rozšířené reality (AR), k dispozici jsou: rtěnka, podkladový krém (česky makeup), tvářenka, oční stíny, oční linky, obočí, řasy
- Makeup lights – pět přednastavených scénářů pro nastavení LED osvětlení, možnost si volně nastavit jas a teplotu
- HiTube – YouTube optimalizovaný pro HiMirror, možnost přesouvat si video volně po obrazovce, nebo minimalizovat a poslouchat pouze zvuk, sledování makeup návodů nikdy nebylo jednodušší a efektivnější
- Lupa – lze přiblížit až 3x, pro zpřesnění aplikace výrobků
- Režim krajiny – lze pohodlně sledovat líčící tutoriály, nebo jiná YouTube videa
- Hudba a sociální sítě – podpora aplikací YouTube, Facebook, Instagram, Spotify, nebo je možné si zavolat přes Google Duo
- Amazon Alexa – došel zvlhčovač? Stačí říct hlasovému asistentovi *Alexa*, aby ho objednala znovu, propojení s chytrou domácností, je možné si skrz zrcadlo zavolat Uber, nebo objednat pizzu
- Bezdrátové nabíjení – součástí zrcadla je podložka na bezdrátové nabíjení

2.1.2.10 Shrnutí

Ačkoliv se jedná o menší zrcátko, je nutné ho mezi chytrá zrcadla také zařadit. Funkce, které nabízí a samotná konstrukce (kosmetické zrcátko) cílí zejména na ženy. Jedná se o domácího asistenta pro péči o pleť, který kromě analýzy pleti a jejího monitoringu nabízí například možnost vést videohovor.

2.1.2.11 Výhody

Na stojánku se nachází bezdrátová nabíjecí podložka, to je určitě velké plus. Mobil chce mít, téměř každý, po ruce a když se během líčení bude ještě nabíjet, tak to ušetří další starosti. Mezi další výhody patří hlasové i dotykové ovládání a propojení s hlasovým asistentem *Alexa*. Reproduktor míří směrem k uživateli. Po stranách jsou LED světla, jejichž barvu lze ovládat. Podpora až čtyř uživatelských účtů a možnost mít zrcadlo jak na šířku, tak i na výšku. Stejně jako u ostatních produktů i HiMirror disponuje krásnou konstrukcí.

2.1.2.12 Nevýhody

Mezi slabší stránky patří kamera, která má pouze 8 MP, vysoká cena a slot na SD kartu nelze využít k rozšíření paměti. Vzhledem k typu zrcadla nelze říci, že velikost patří mezi nevýhody.



■ **Obrázek 2.8** Kosmetické chytré zrcadlo Hi Mirror, které je jako stvořené pro ženy; nabídne několik druhů kosmetických doplňků, které jsou schopné analyzovat pleť, objednat produkty, nebo se zaměřit, zda používané produkty opravdu fungují; zrcadlo je ale stvořené i pro zábavu, videohovory s přáteli a sledování kosmetických tutoriálů [14]

2.1.2.13 Vodotěsné inteligentní zrcadlo

Velikost zrcadla: 51 × 34,8 cm (1774,8 cm²)

Cena: 27 990 Kč

Specifikace dostupná z [15]:

- Poměr stran 16:10
- Velikost LED displeje 19"
- Rozlišení 1366 × 768
- Podpora DVB-T2
- Hmotnost 6 kg
- Android 9.0 OS
- Vodotěsný a prachotěsný stupeň IP66
- Ideální pozorovací vzdálenost 1,5–1,9 m
- Použití: koupelnová TV

2.1.2.14 Shrnutí

Produkt je označen jako inteligentní zrcadlo, nicméně se jedná o chytrou televizi s operačním systémem Android 9.0 OS, která je pouze za polopropustným sklem. Nabízí se tento produkt umístit spíše někam k vaně, kde bude sloužit jako koupelňová televize, která je přizpůsobená těmto podmínkám. Díky ochraně proti prachu a proti intenzivně tryskající vodě, není třeba se bát, že při pocákání vodou dojde k poškození produktu.

2.1.2.15 Výhody

Reproduktor je namířen dopředu, směrem na uživatele. Konstrukce nijak nenadchne, ale vypadá čistě, bez vyčnívajících rámečků. V popisu produktu není uvedené nic o způsobu ovládní. Jedná se ale o televizi, lze předpokládat, že součástí balení bude dálkový i ovladač.

2.1.2.16 Nevýhody

Je potřeba vytknout vysokou cenu. Nevidím přesně důvod, proč si pořizovat koupelňovou televizi schovanou za polopropustným sklem. Funkcionalita zrcadla zde je zbytečná, na toto použití se vůbec nehodí. S rozlišením pouze HD a velikostí displeje 19" je cena opravdu směšná. Když vezmu v úvahu, že za zlomek ceny tohoto produktu lze pořídit televizi s lepšími parametry, rozhodně není důvod si toto zařízení pořídit.



■ **Obrázek 2.9** Vodotěsné inteligentní zrcadlo prezentované jako koupelňová televize; garantuje stupeň ochrany IP66 a je výrazně dražší než kdejaká kvalitnější televize [15]

2.2 Skla

Tato sekce se bude zabývat analýzou skel pro chytrá zrcadla. Budou porovnány vlastnosti, typy skel a vhodnost jejich použití. Zároveň bude popsána i alternativní možnost – standardní sklo s polopropustnou fólií.

2.2.1 Technické specifikace

U zrcadlových skel se udávají dva hlavní parametry. Jedná se o optické vlastnosti materiálu – **odrazivost** a **průhlednost**. Obojí se udává v procentech a vyjadřuje to poměr, jak moc je přes zrcadlo vidět. 100 % odrazivosti je běžné zrcadlo. Průhlednost znamená kvalitu zobrazovaného obsahu, poněvadž doplněk průhlednosti vyjadřuje, kolik procházejícího světla skrz sklo je blokováno. Hodnoty pro základní typy skel pro chytrá zrcadla jsou v tabulce 2.1. Je nutné podotknout, že součet hodnot odrazivosti a průhlednosti nemusí nutně dávat 100 %. [16]

Typ skla	Odrazivost	Průhlednost
Polopropustné sklo	70 %	11 %
Akrylové polopropustné sklo	70 %	30 %
Sklo pro chytré zrcadlo	70 %	30 %
Dielektrické sklo	30 %	70 %

■ **Tabulka 2.1** Hodnoty optických vlastností skel – odrazivosti a průhlednosti [16]

■ Polopropustné sklo

Sklo, které je z jedné strany průhledné a z druhé odráží obraz. Tato skla se využívají například ve výslechových místnostech, pozorujících sklech a tzv. „infinity“ zrcadlech, kde jde o optickou iluzi – obraz se odráží od jednoho zrcadla do druhého a vzniká tak 3D efekt, který nikdy nekončí (dost často využívané jako designový prvek na principu LED diod). Vyrábí se vypalováním zrcadlového povlaku na skle s šedým odstínem a výsledkem je vysoce odolné sklo, které je odolné proti poškrábání.

Funkce skla závisí na tom, z jaké strany se na něj člověk dívá. S tmavším osvětlením na pozorovací straně je možné vidět skrz. Z druhé strany je vidět odraz. Sklo je možné temperovat, čímž se zvýší jeho odolnost a čistí se standardními prostředky na sklo a papírovými utěrkami. [17]

■ Akrylové polopropustné sklo

Toto sklo funguje na stejném principu a používá se stejně jako výše zmíněné polopropustné sklo. Nicméně je mnohem silnější a používá se například pro klinická pozorování, kde by se běžné sklo mohlo roztříštit. Oproti sklu má lepší průhlednost a velmi se podobá standardnímu zrcadlu. Je také mnohem jednodušší pro úpravu rozměrů (řezání, či vrtání). Protože se nejedná o běžné sklo, je lehké. Na čištění se používají speciální přípravky na akryl a čisticí hadříky. Na skle jsou vidět plastové hrany, proto se doporučuje kraje nějakým způsobem zakrýt, například rámem.

Nabízí se dvě hlavní tloušťky – 6 mm a 3 mm. Je dobré si dát pozor, protože akrylové sklo má tendenci se s většími rozměry prohýbat, a to v tomto případě znamená deformaci odrazu. Tloušťka 6 mm je lepší pro velikosti větší než 24" × 24". [18]

- Sklo pro chytré zrcadlo

Jedná se o sklo, které je nadřazené polopropustnému sklu se specializací přímo na chytrá zrcadla je upraveno, aby poskytovalo velikou odrazivost a zároveň dostatečnou průhlednost. Stejně jako ostatní skla se dá temperovat. To je vhodné především na místech, kde by mohlo dojít k jeho rozbití. Například na veřejnosti, nebo obecně venku. K čištění se používají běžné prostředky na sklo. [19]

- Dielektrické sklo

Někdy také známé jako televizní zrcadlo, je polopropustné sklo. Jeho vysoká průhlednost se využívá tam, kde je potřeba bezkonkurenční kvalita a nízké oslnění. Pro zajištění nejvyšší viditelnosti bez odstínu, je zrcadlový povlak vytvořen na skle s nízkým obsahem železa. Používá se pro skrytí televize. Pokud je televize vypnutá, působí jako běžné zrcadlo. Pokud je zapnutá, efekt zrcadla zmizí a je vidět obsah zobrazený v televizi. Tuto funkcionalitu využije někdo, kdo chce mít televizi schovanou a zároveň zrcadlo jako dekoraci. [20]

- Polopropustná zrcadlová fólie

Hlavní použití fólií je spíše na získání soukromí v domě tak, aby přes den nebylo vidět dovnitř. Některé fungují tak, že dovnitř není vidět ani v noci. Zrcadlová fólie vytváří z jedné strany efekt zrcadla a z druhé je vidět skrz, nicméně obraz podléhá jisté deformaci. Většinou je tmavší, někdy i s nádechem nějaké barvy. Využití mají především na oknech budov, například škol, v tanečních studiích, pro ochranku v obchodech, v chytrých a nekonečných zrcadlech, na policejních stanicích, ve vězení a pro kamerové show, kde se využívá skrytých kamer. Dodává se většinou v konkrétních rozměrech se stěrkou. Fólie často bývají samolepící, stačí tedy nacákat na povrch skla vodu a vytlačit bublinky ven. Lépe přilne na sklo, nicméně na akrylové sklo (plast) se dá přilepit taktéž. [21]

2.2.2 Porovnání

Na obrázcích 2.10 a 2.11 je simulované prostředí chytrého zrcadla pro test barev, zobrazování textu a grafů.

Dielektrické sklo je nejlepší, co se týče věrohodnosti a viditelnosti zobrazeného obsahu, nicméně nemá tak dobrou odrazivost a proto jím vytvářený odraz není z výše zmíněných skel nejbližší standardnímu zrcadlu.

Akrylové sklo má poněkud tmavší barvy, které z bílé dělají spíše šedivou, ale samotné sklo barevně nijak neovlivňuje zobrazovaný obsah.

Sklo chytrého zrcadla má o něco jasnější barvy než sklo akrylové a má i lepší odraz. Společně s polopropustným sklem mají nejvěrohodnější odraz, což z nich dělá skoro nerozeznatelná skla od standardního zrcadla.

Polopropustné sklo podává obsah velmi tmavě, protože má malou průhlednost. Vyznačuje se pouze dobrou odrazivostí. [16]

2.2.3 Časté otázky při výběru skel

Informace jsou dostupné z [16, 17, 18, 19].

- Mohu pro svůj projekt použít akrylové sklo?

Odpověď zní ano, pokud jsou splněny následující podmínky, jinak je to silně nedoporučené.

1. Sklo bude zasazené v rámu

Je to z toho důvodu, že u standardního skla se často zbrušují hrany, či dochází k jejich jiné povrchové úpravě. To se ale u akrylového skla nedělá a na krajích jsou vidět plastové hrany.

2. Nepoužijte se infračervený dotykový rámeček

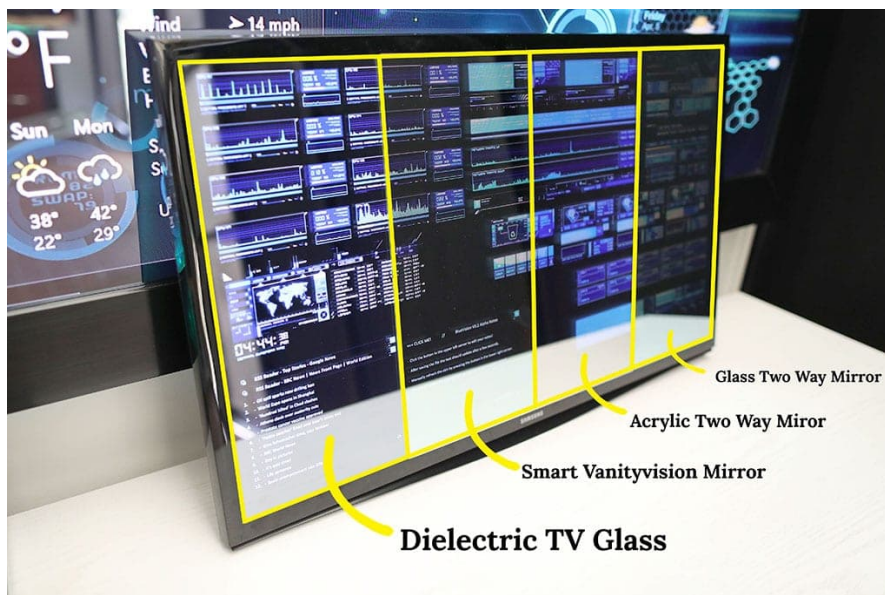
Je doporučeno se vyhnout kombinaci akrylového skla s infračerveným dotykovým rámečkem za každou cenu. Plast se o ohýbá a způsobuje rušení pro infračervené záření.

3. Zrcadlo nebude větší než 24" × 48"

Pro tloušťku 3 mm je doporučena maximální velikost 12" × 24". Pro silnější 6 mm sklo by velikost neměla přesahovat 24" × 28". Je to kvůli vlastnostem materiálu. Akryl s přibývajícím velikostí má větší tendenci se ohýbat.



■ **Obrázek 2.10** Porovnání zobrazení barevného kolečka a jednoduchého bílého nápisu zobrazených stejným displejem skrz čtyři různá skla. Zleva: dielektrické, akrylové, sklo pro chytré zrcadlo a polopropustné sklo [16]



■ **Obrázek 2.11** Porovnání zobrazení nějakého monitoring dashboardu a jednoduchého bílého textu zobrazených stejným displejem přes čtyři různá skla. Zleva: dielektrické, sklo pro chytré zrcadlo, akrylové a polopropustné sklo [19]

- Je lepší sklo, nebo akryl?
Sklo se dobře čistí, je odolné proti poškrábání a zpravidla bývá tenčí. Akryl je náchylný na stopy škrábanců, lehčí a bezpečnější. S většími rozměry se ale ohýbá.
- Může být zrcadlo větší než displej?
Ano, pokud se nepoužívá infračervený dotykový rámeček. Stačí vzít černou barvu na sklo a natřít plochu, kde nebude displej. Je doporučeno použít kvalitní maskovací pásku na plochu, kterou nechceme zabarvit, aby nedošlo k neúmyslným škodám.
- Je možné do skla vyvrtat díry?
Ano, je to možné u netemperovaného skla, nebo u akrylového skla
- Může být fólie aplikována na akryl, nebo polykarbonát?
Nejlépe přilne fólie na sklo, nicméně pokud někomu nevadí malé bublinky, nebo bude hodně pečlivý s jejich vytlačněním, může být fólie aplikována i na plast.
- Jaké lepidlo použít při lepení polopropustného skla?
Doporučuje se použít UV lepidlo, nebo černou pryskyřici na zrcadlo. Před použitím na celé zrcadlo je dobré si vzhled otestovat na vzorku, jak to bude vypadat. Ne každému se to může zamlouvat.
- Jak poznat stranu ze které to má vypadat jako zrcadlo?
Nejjednodušší je použít tzv. „fingernail test“. Pokud při dotyku zrcadla prstem je mezi ním a sklem mezera, jedná se o zrcadlovou stranu (je vidět špička prstu). Pokud tam žádná mezera není, jedná se o stranu se zrcadlovým povlakem.

2.2.4 Shrnutí

Dostupnost těchto typů skel je na tuzemském trhu špatná. Především z důvodu, že v angličtině existují pojmy, které přesně definují typ skla a najít jejich rozumný ekvivalentní překlad je poměrně náročné. Pokud by se mělo vybrat z výše zmíněných variant, přicházejí v úvahu dva typy – **sklo pro chytrá zrcadla** a **akrylové polopropustné sklo**. Oba zmíněné mají ideální poměr odrazivosti a průhlednosti (70 % a 30 %), působí nerozeznatelně od standardního zrcadla a informace jsou skrz ně dobře čitelné. Vzhledem k tomu, že cílem této práce je i minimalizovat cenu, zde se vyskytuje problém. Skla jsou převážně ze zahraničí a jedná se o materiál, který je křehký. Vyžaduje, aby s ním bylo zacházeno opatrně. To ale znamená vyšší cenu poštovního, která se někdy vyšplhá až na dvojnásobek ceny skla. To už je na zvážení zda se to vyplatí. Pokud si někdo potrpí na maximální kvalitě a netíží ho cena, pak se jedná o jasnou volbu. V tomto ohledu představuje výhodu fólie, která se dá na tuzemském trhu koupit a standardní sklo taktéž. Cenově vychází fólie nejlépe, avšak nedosahuje takových kvalit jako samotná skla.

2.3 Displeje

V předchozí části je možné vyzorovat, že projekty nejčastěji používají nějaké starší monitory. Důvodem je cena. Ne vždy se ale vyplatí šetřit, nebo sáhnout po prvním odloženém monitoru. V této práci bude uvedeno, na co je dobré se zaměřit a jaké jsou hlavní vlastnosti daných typů displejů.

LCD (Liquid Crystal Display) – hlavní princip je založen na vyzařování podsvícujícího bílého světla směrem do očí. Zbytek panelu se stará o to, aby bylo toto podsvícení změněno na jednotlivé pixely. LCD využívá jev zvaný *polarizace*. Jedná se o směr, ve kterém paprsek světla osciluje. Světlo vychází z podsvícení nepolarizované a prochází polarizátorem, který zaručuje, že veškeré světlo jde jedním směrem.

LED (Light-Emitting Diode) – dost často se vyskytuje u LCD i nápis LED. Jedná se o druh podsvícení, který je dnes zcela běžný oproti dříve používanému podsvícení studenou katodou.

Tekutý krystal (liquid crystal) je krystalová struktura, která dokáže změnit polarizaci, která jím projde. Klidová pozice krystalu je udělána tak, aby polarizaci neměnila. Ve chvíli, kdy světlo dosáhne druhého polarizátoru, veškeré světlo je blokováno. Přivedením nějakého napětí se „zapne“ krystal na nějakou konkrétní úroveň (napětí reguluje intenzitu). To změní procento polarizace procházejícího světla tak, že ho druhý polarizátor již neblokuje a umožní očím toto světlo vidět. O barvu se starají trojbarevné filtry, skládající se ze složek RGB (červená, zelená, modrá), které propouští každý pouze svoji barvu. Rozdíl mezi LCD panely tvoří způsob, jak tato krystalová část funguje.

[22]

2.3.1 Typy LCD panelů

Informace dostupné z [22].

■ TN panely

TN, neboli *Twisted Nematic*, je technologie z 80. let 20. století. Jakmile je světlo usměrněno, dorazí do krystalové části, kde na základě stavu krystalu může dojít ke stočení polarizace o 90°, tedy do takové podoby, že bude mít stejnou orientaci jako druhý polarizátor a ten nechá světlo projít. Nebo se může krystal natočit v souladu s prvním polarizátorem, pak ale druhý světlo zablokuje.

Výhody	Nevýhody
Nízká cena Skvělá odezva, panely dosahují až 240 Hz	Horší kontrast Horší podání barev Horší pozorovací úhly

■ **Tabulka 2.2** Výhody a nevýhody TN panelů

■ VA panely

VA, neboli *Vertical Alignment*, je technologie z 90. let 20. století. Tekuté krystaly jsou zarovnaný kolmo (vertikálně), nebo paralelně (horizontálně) vůči druhému polarizátoru. Ve vypnutém stavu jsou krystaly kolmo mezi polarizátory. V zapnutém stavu jsou zarovnaný horizontálně a mění polarizaci tak, aby světlo prošlo přes druhý polarizátor a bylo tak vidět.

Výhody	Nevýhody
Tato struktura umožňuje hlubší černou a lepší barvy než TN panely Lepší pozorovací úhly oproti TN Vysoký kontrast	Menší obnovovací frekvence a horší odezva než TN Vyšší cena oproti TN

■ **Tabulka 2.3** Výhody a nevýhody VA panelů

■ IPS panely

IPS, neboli *In-Plane Switching*, panely se objevily chvíli po TN panelech. Krystaly jsou vždy horizontálně mezi dvěma polarizátory a stočením o 90° se vypínají a zapínají. V tomto návrhu je nutné, aby dvě elektrody, které přivádí proud do tekutých krystalů, byly na stejném substrátu skla. Nikoliv jako u ostatních typů LCD, kde jsou na skleněných substrátech nad a pod krystalem. To ale zablokuje o něco více světla než TN i VA panely.

Výhody	Nevýhody
Skvělé barvy Skvělý kontrast Dobré pozorovací úhly Dobrá odezva, obnovovací frekvence až 144 Hz	Méně jasné Vyšší cena

■ **Tabulka 2.4** Výhody a nevýhody IPS panelů

2.3.2 OLED panely

OLED, také jinak *organic LED*, jsou panely, které se liší od LCD. Nepochází zde k polarizaci. Každý pixel (subpixel červené, zelené a modré) se rozsvítí podle napětí, které je přivedeno do komplexní molekuly zvané organic LED. Barva záleží na molekule a jas na přivedeném napětí. OLED panely mohou dosahovat HDR (High Dynamic Range) jasu, protože molekuly vyzářují správné barvy a světlo není ničím blokováno. Typické pro OLED je nepravidelné rozložení subpixelů. Za to může například nerovnoměrný rozklad molekul jednotlivých barev. [22]

Výhody	Nevýhody
Vysoký kontrast Výborné barvy, dokonalá černá Rychlé blikání a vypínání OLED snižuje odezvu Průběžné skenování – princip zapínání a vypínání bloků obrazovky shora dolů, který snižuje rozmazání Vysoká hustota pixelů	Nízká obnovovací frekvence Vysoká cena Molekuly se po čase rychle opotřebovávají, to vede ke snížení jasu Při dlouhodobém svícení konkrétních diod vznikají vypáleniny, které vytváří viditelný „otisk“

■ **Tabulka 2.5** Výhody a nevýhody OLED panelů [22]

Mezi další výhody patří, že OLED panely by měly mít nižší spotřebu. Zatím se molekuly OLED, které se zapínají nižším napětím neobjevily. Je to tedy otázka času, kdy k tomu teprve dojde. V základu ale platí, že OLED displeje mají vyšší spotřebu energie při zobrazování světlých pixelů (hodně zářících) a minimální na zobrazení tmavých barev. Černé pixely pak nespotřebovávají žádnou energii – to je věc, která je naprosto dokonalá pro projekt chytrého zrcadla, kde drtivá většina plochy displeje bude černá, aby fungovala jako zrcadlo. [22]

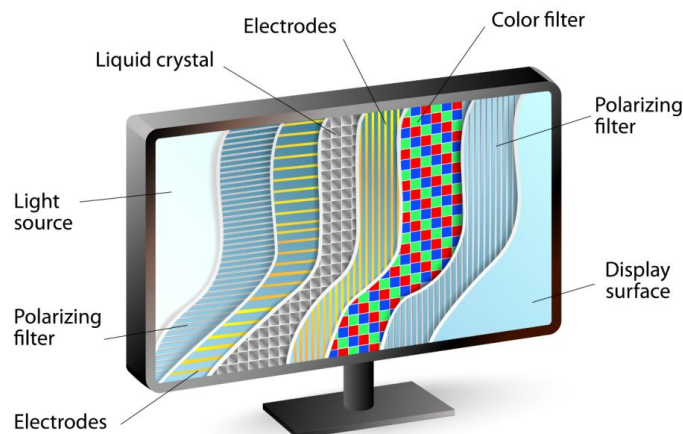
2.3.3 Quantum Dots

Tato technologie je založena na molekulách, které absorbují světlo a poté ho znovu vyzařují v barvě, která se po nich vyžadovala. Díky tomu jsou LCD panely schopné dosahovat HDR (High Dynamic Range = vysoký dynamický rozsah) jasu.

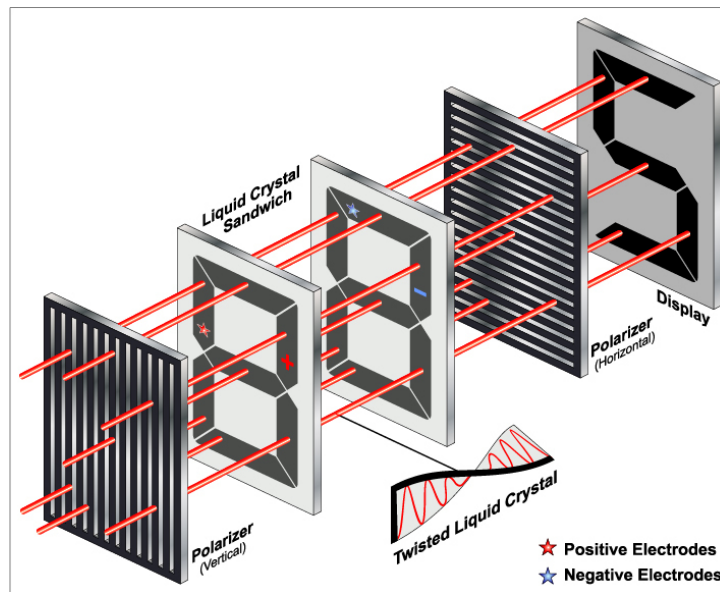
Vrstvy těchto *Quantum dots* jsou nejčastěji umístěny mezi modré podsvícení a polarizační část. Obvykle se využívají k vytvoření červené a zelené složky, aby více odpovídaly barevným filtrům, přes které pak projde více světla. Také dochází k redukci toho, že nějaké světlo projde špatným subpixelem. Výsledkem jsou pak lepší barvy.

Ačkoliv se stále zkouší nové návrhy *Quantum dots*, slibně vypadá hlavně kompletní nahrazení filtrů barev. Díky čemuž projde ještě více světla displeje. LCD podsvícení produkuje více světla než OLED panely, a tak jsou tyto panely nejjasnější. *Quantum dots* nemění obnovovací frekvenci, ale pouze **podání barev a jas**. [22]

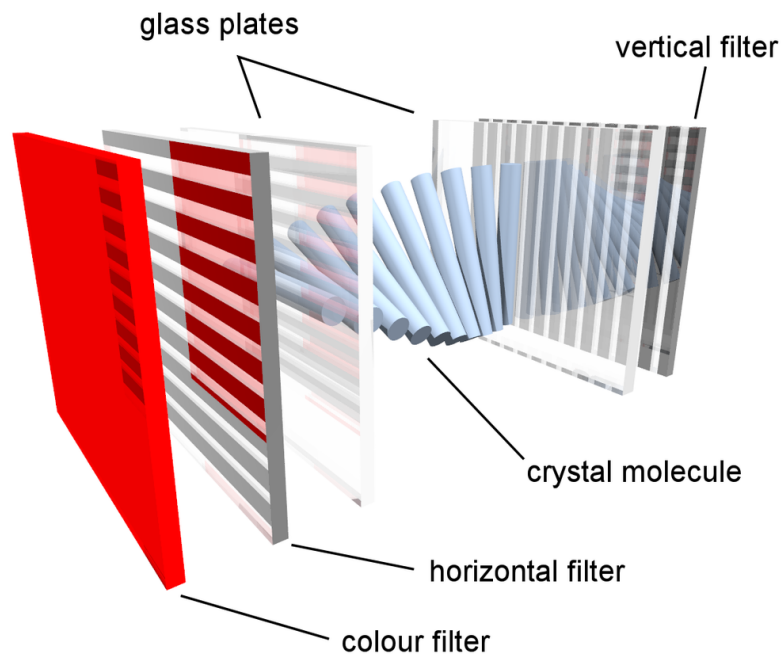
LIQUID CRYSTAL DISPLAY



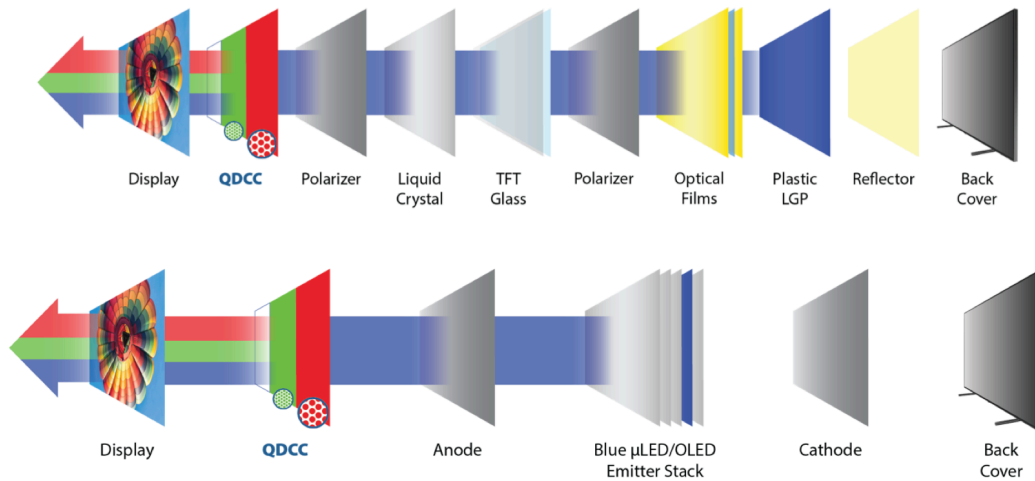
■ **Obrázek 2.12** Vrstvy LCD displeje od zadního podsvícení přes polarizační filtr, elektrody, složku tekutých krystalů, druhých elektrod, barevného filtru, druhého polarizačního filtru až po povrch displeje [22]



■ **Obrázek 2.13** Zachycení principu polarizace kde světlo nejprve projde prvním (vertikálním) polarizátorem, pak přes elektrody mezi nimiž se nachází vrstva tekutých krystalů, do druhého polarizátoru, který propustí správně orientované světlo, které se zobrazí na displeji [22]



■ **Obrázek 2.14** Vysvětlení principu TN panelu, kde světlo vycházející z podsvícení projde vertikálním polarizačním filtrem, přes elektrody do vrstvy tekutých krystalů, které jsou schopny stočit polarizaci o 90° , přes druhou vrstvu elektrod, horizontální polarizátor a nakonec přes barevný filtr, který pak vytváří zobrazení na displeji [22]



■ **Obrázek 2.15** Technologie Quantum dots, která zlepšuje grafické podání barev u LCD a OLED; ukázka přes jaké vrstvy světlo prochází; podsvícuje se modrým světlem, proto se u je u barevných filtrů modrá barva vynechána [23]

2.3.4 MicroLED

Budoucnost? Panely fungují podobně jako OLED, ale jsou navrženy tak, aby byly lepší. *MicroLED* jsou přibližně 100 krát menší než běžné LED a jeden pixel tvoří „soudiod“ tří diod – červené, zelené a modré (RGB). Nevyužívají organické látky, a tak nehrozí vypálení a mají delší životnost. Díky většímu počtu LED lze dosáhnout jemnějšího obrazu a vyššího jasů. Díky modulární konstrukci lze vytvářet obrazovky s libovolnou velikostí. Výroba LED je ale drahá a bude nějakou dobu trvat, než se tyto displeje začnou používat ve velké míře. Podle expertů je možné očekávat nástup této technologie v zařízeních s malými rozměry, jako jsou chytré hodinky. [22]

Výhody	Nevýhody
Vysoká hustota pixelů (jemnost) Vysoký kontrast Vysoký jas Velmi nízká spotřeba Dokonalá černá barva	Nedostupnost Vysoká cena

■ **Tabulka 2.6** Výhody a nevýhody MicroLED panelů [22]

2.3.5 Výběr vhodného displeje

Pro tento projekt je podle parametrů jednoznačný vítěz – OLED panel. Dokonalá černá barva a nulová spotřeba při vypnutých pixelech je ideální pro chytré zrcadlo. Nicméně parametry nejsou všechno a je potřeba vzít v úvahu cenu a dostupnost. Vhodný displej je LCD s LED podsvícením s typem panelu VA. Kombinuje lepší části TN i IPS a přitom zachovává rozumnou cenu.

2.3.6 Požadavky na monitor

Důležitý faktor je spotřeba. Jedná se o údaj, který nejvíce lidí přehlíží. Ušetření peněz za levnější monitor nemusí být nutně ušetření z dlouhodobého hlediska. Další, na co je nutné se zaměřit, je jas. Čím vyšší, tím lepší. Je potřeba prosvítit sklo tak, abych skrz něj byl vidět co nejlepší obraz. Monitory v chytrých zrcadlech mohou být postaveny na výšku, nebo na šířku, záleží na preferencích. Určitě je dobré myslet na obě varianty a mít dobré pozorovací úhly jak v horizontálním, tak ve vertikálním směru, aby bylo na zrcadlo něco vidět a obraz nebyl deformovaný při pohledu ze strany. Rozlišení takovou roli nehraje. Není potřeba mít dokonale jemný displej, nicméně jako spodní hranici lze brát 1600×900 (HD+), přestože dnes je typicky nejnižší hodnota 1920×1080 (Full HD). Nevhodné konektory se dají řešit redukcemi. Priorita je, aby měl monitor HDMI konektor, ale není to nutnost.

2.3.7 Pořízení displeje

Displeje dnes mají dvě hlavní zastoupení – monitor a televize. Monitory jsou zpravidla menší než televize a levnější. Neplatí to tak ale vždy. Sféra produktů, na které se tento projekt pohybuje, to tak má. Důvodů proč nebrat televizi je několik. Platí se navíc za TV tuner, mají vyšší spotřebu, horší pozorovací úhly (minimálně starší modely) a jsou větší, typicky od 30". Monitor je vhodná volba zejména kvůli tomu, že je připravený po zapojení kabelů okamžitě fungovat na rozdíl od samotného displeje. Někdy má navíc i vestavěné reproduktory, nebo USB hub, což jsou bonusy. Rozhodně se nevyplatí kupovat zvlášť displej a zvlášť tzv. *controller board*, tedy hardware, který se stará o zobrazování dat na displej. Nevyplatí se to kvůli ceně. Aby zrcadlo nebylo příliš drahé, dá se ušetřit právě na monitoru. Není nutné mít nejnovější technologie, které jsou dražší, ale naopak se vyplatí pořídit monitor z druhé ruky. Jejich cena je znatelně nižší a zároveň se nepřijde o takovou ztrátu kvality. Ovšem přichází s tím riziko, že se monitor může dříve porouchat, a tak je důležité pečlivě vybírat.

2.4 Platformy

Následující text se zabývá popisem platform zabývajících se chytrým zrcadlem a chytrou domácností. Open-source MagicMirror² se dnes běžně používá pro podomácku vyrobená chytrá zrcadla. Tato platforma je založená na modulech a právě díky tomu je hodně vyvíjena komunitou. Stačí vytvořit modul s nějakou funkcionalitou, který je pak přístupný ostatním lidem, pokud projde schválením. HomeAssistant je platforma pro chytrou domácnost. Jedná se o plně lokální software pro domácí automatizaci bez cloudu.

2.4.1 MagicMirror²

Je open-source modulární platforma pro chytré zrcadlo. MagicMirror² je založené na původním projektu Michaela Teeuwa, který dal k dispozici svůj projekt a vznikla skupina lidí, kteří si říkají tzv. „builders“ (stavitelé). Projekt kompletně předělali a upravili tak, aby se dal používat a rozvíjet do budoucna. Nyní je spravován velkou skupinou nadšenců, kteří do tohoto projektu přinášejí vlastní nápady a inovace. Platforma je založena na modulech, kterých je aktuálně velké množství a stále jich přibývá. Drtivá většina modulů je pro projekty tvořené v angličtině, nicméně se zde najdou moduly implementované pro konkrétní země a jazyky. Příklad takového modulu pro Českou republiku je *MMM-idos*, který zobrazuje odjezdy hromadné dopravy z konkrétní stanice v reálném čase. Platforma se zaměřuje na modulární systém pluginů a používá *Electron*¹ jako aplikační wrapper. [24]

2.4.1.1 Hardware a operační systém

MagicMirror² je navržen pro běh na Raspberry Pi. Není to pevně definované. Platforma je spustitelná i na jiných zařízeních, ale nové verze jsou testovány na pouze na Raspberry Pi. Použitý aplikační wrapper Electron podporuje verze Raspberry Pi 2 a novější. Je potřeba nainstalovat poslední plnohodnotnou verzi Raspbian OS, která nesmí být ve verzi lite a Electron potřebuje desktopové prostředí. [25]

2.4.1.2 Možnosti instalace

Platformu lze nainstalovat manuálně, nebo využitím instalačních skriptů. Na začátku roku 2020 však bylo rozhodnuto, že automatické instalátory budou přesunuty z hlavního repozitáře do komunitou spravovaného samostatného repozitáře. Jediná oficiální a podporovaná instalace je manuální. Mezi alternativní metody patří (dostupné z [25]):

1. Automatické instalační skripty – instalační a update skript, rozhraní v příkazové řádce na správu modulů
2. Docker Image – platforma spuštěna pouze v režimu serveru a nasazena v dockeru, více k režimu serveru v 2.4.1.3
3. Kubernetes Helm Chart – platforma spuštěna pouze v režimu serveru jako kubernetes cluster
4. MagicMirrorOS – kompletní operační systém založený na Raspbian OS. Tento operační systém lze stáhnout a nahrát na SD kartu a MagicMirror² rovnou funguje se základním nastavením. Za tímto způsobem se skrývá alternativní nastavení dockeru umožňující přímé zobrazení obsahu na obrazovku Raspberry Pi.

¹Electron.js je runtime framework, který umožňuje uživateli vytvářet desktopové aplikace s HTML5, CSS a JavaScript. Jedná se o open source projekt, který zahájil Cheng Zhao, inženýr společnosti GitHub.

2.4.1.3 Speciální případy

MagicMirror² umožňuje běh aplikace **pouze** v **server** režimu, tedy samotné okno (grafické rozhraní) není vidět. Tento režim lze spustit příkazem, nebo použitím dockeru. Samotný obsah chytrého zrcadla je možné zobrazit v libovolné prohlídce. Stejně tak je možné pustit aplikaci **pouze** jako **klienta**. To vyžaduje, aby server běžel někde vzdáleně a Raspberry Pi se na něj připojí. [25]

2.4.1.4 Konfigurace

Vlastnosti, které je možné nastavit, v závorkách jsou uvedené základní hodnoty (dostupné z [25]):

- port – číslo portu, na kterém MagicMirror² server poběží (8080)
- address – IP adresa rozhraní na které dochází k potvrzení připojení (localhost)
- ipWhitelist – IP adresy, které mají povolený přístup k MagicMirror² ([":::1", "127.0.0.1", "::ffff:127.0.0.1"])
- zoom – umožňuje škálovat zobrazený obsah přibližujícím faktorem (1.0)
- language – jazyk rozhraní, podporované jsou angličtina, nizozemština, ruština a francouzština (en)
- timeFormat – časový formát, který bude použit, možný je 12 a 24 hodin (24)
- units – jednotky, ve kterých budou měřeny hodnoty v modulu pro počasí, možná je metrická, nebo imperiální soustava (metric)
- modules – seznam modulů použitých v zrcadle a jejich konfigurace
- electronOptions – volitelé nastavení možností pro Electron
- customCss – cesta k definici vlastních CSS stylů (css/custom.css)

2.4.1.5 Rozhraní a moduly

Platforma je postavena na technologiích *HTML*, *CSS*, *JavaScript* a knihovnách jako je například *jQuery*. Dříve byla platforma spustitelná pouze v prohlídce, ale díky aplikačnímu wrapperu Electron se z ní stala aplikace pro desktop. Pro svůj běh potřebuje operační systém a aktuální verzi prostředí Node.js – nutné k fungování Electronu. Dalo by se říct, že se jedná o webovou stránku s černým pozadím, na které jsou nositeli informací moduly. Autor původní verze MagicMirror používal v základu čtyři moduly, později jich vytvořil více. Díky velikému rozšíření v Raspberry Pi komunitě vzrostl zájem o vlastní moduly. Michael Teeuw se rozhodl, že výtvoři přispěvatelů podpoří a vytvořil dokumentaci a pravidla, jak takové vlastní moduly vytvořit, aby měly jednotnou strukturu a byly kompatibilní pro všechny uživatele. Dnes existují stovky modulů různých kategorií, mezi které patří například finance, noviny, doprava, počasí hlasové ovládání apod.

Moduly jsou napsané v JavaScriptu a mohou používat vlastní CSS prvky pro lepší vzhled. Konfigurace modulů je součástí hlavního konfiguračního souboru uvedeného v 2.4.1.4. Možnosti, které lze konfigurovat (dostupné z [25]):

- module – název modulu, nebo podadresář, kde se modul nachází
- position – pozice na „stránce“, kde bude modul zobrazen, volitelný parametr; při více modulech se stejnou pozicí se moduly umísťují podle pořadí v konfiguračním souboru

- `classes` – volitelný seznam CSS tříd, které budou použity pro modul
- `header` – volitelný text zobrazený nad modulem
- `hiddenOnStartup` – volitelná možnost skrytí modulu při spuštění
- `disabled` – volitelná možnost, která přeskočí vytvoření modulu
- `config` – nastavení konkrétních parametrů nějakého modulu, volitelné, pokud to modul explicitně nevyžaduje

2.4.2 HomeAssistant

Je software pro chytrou domácnost. Jedná se o bezplatný open-source projekt, který dává lokální kontrolu a soukromí na první místo. HomeAssistant není pouze aplikace, je to celý vestavný systém, který umožňuje onboarding, konfiguraci a update skrze jednoduché uživatelské rozhraní. Je to alternativa k platformám jako je HomeBridge a SmartThings. Hlavní výhoda je, že se nejedná o platformu pro domácí automatizaci s cloudem. Tím se zbavuje závislosti na připojení k internetu, nebo vzdáleným serverům. Díky tomu je provoz rychlejší a spolehlivější s ohledem na to, že se o chod této platformy stará dostatečně výkonný hardware.

Existují i jiné alternativy, například *OpenHAB* nebo *WebThings* od Mozilly. Všechny jsou lokálně hostované. HomeAssistant má ale v tuto chvíli nejvíce integrací. Vzhledem k využití pythonu, což je jeden z nejpobulárnějších programovacích jazyků na světě, má HomeAssistant veliký počet přispěvatelů, kteří svojí prací rozšiřují funkcionalitu tohoto projektu. [26]

2.4.2.1 Proč jsou možnosti založené na cloudu špatné?

Mnoho chytrých zařízení je založeno na webové infrastruktuře. Je to kvůli komfortu. Při aktualizacích není nutné řešit stranu klienta, nastavení je jednodušší, ale přináší to s sebou nevýhody. Kontrolu nad systémem má výrobce, stejně tak jako data a statistiky. Může dojít k jejich zneužití. Data jsou v dnešní době cenné informace a mohou představovat vysokou hodnotu.

Některá zařízení dokážou fungovat offline, jiná ne. Pokud jde o automatizaci, ta je často zpracována externími servery. Jakékoliv zařízení v síti s přístupem k Wi-Fi představuje potenciální terč pro napadení různými viry a hacknutím, jako je sbírání a odesílání dat, přístupem do sítí, nebo vytvořením zadních vrátek pro jiné zákeřné úmysly. Většinou se jedná o nějaké mikrokontroléry a minipočítače, kde je v podstatě nemožné mít antivirový program. Bezpečnostní „díry“ ve firmwaru má na starost výrobce.

Ačkoliv bylo dříve náročné s platformou HomeAssistant pracovat, s každým vydáním se zlepšuje uživatelské rozhraní. Platformu se vyplatí použít, kvůli výkonu, který nabízí a kolik zařízení podporuje. Instalace samotného systému je jednoduchá, pouze se nahraje image HassOS na SD kartu a vloží se například do Raspberry Pi. Konfigurace a integrace základních zařízení, jako jsou světelné spínače, žárovky, Wi-Fi zástrčky, je velmi jednoduchá a dobře zdokumentovaná. Tam kde si uživatel neví rady, je možné zeptat se v komunitě, která pomůže s případnými dotazy. [26]

2.4.2.2 Co znamená HassOS, Hass, HomeAssistant Core, HomeAssistant Supervised a jak se liší?

Informace dostupné z [26].

- Hass

Jedná se pouze o zkratku HomeAssistant

- HomeAssistant Core

Je hlavní python *main* program pro domácí automatizaci. Je to jeho hlavní část – uživatelské rozhraní, jádro, komponenty, integrace, všechno. Dá se pustit jako python program nainstalovaný pomocí *pip*, nebo ve virtuální prostředí. Takto tento projekt začínal a běžel před pár lety.

- HomeAssistant (dříve Hass.IO)

Jedná se o Docker verzi HomeAssistant s hypervizorem (označení v informatice pro jednu z technik virtualizace hardwaru počítače, která umožňuje na jednom počítači spustit zároveň více operačních systémů) podobným Portaineru přizpůsobenému pro práci s Hassem. Přidává některé věci uživatelského rozhraní do HomeAssistant Core a umožňuje instalovat „add-ons“, aktualizovat HomeAssistant jedním kliknutím a další funkce.

- HomeAssistant Supervised

Jiný název pro HomeAssistant zmíněný výše. Používá se pro rozlišení mezi *HomeAssistant* a *HomeAssistant Core*.

- HassOS

Je minimální linuxová distribuce, která má vše potřebné na to, aby spustila platformu HomeAssistant. Uživatel má nad hostitelským systémem minimální kontrolu a nemá mnoho věcí, které se nacházejí v jiných obecných distribučních systémech.

2.4.2.3 Možnosti instalace

Home Assistant nabízí čtyři základní možnosti instalace (dostupné z [27]):

1. HomeAssistant Operating system – minimální operační systém, který obsahuje Supervisor pro správu HomeAssistant Core a předinstalované add-ons
2. HomeAssistant Container – instalace HomeAssistant Core například v Doceku
3. HomeAssistant Core – manuální instalace přes python virtuální prostředí (pro zkušenější uživatele)
4. HomeAssistant Supervised – manuální instalace Supervisor (pro zkušenější uživatele)

	OS	Container	Core	Supervised
Automations	✓	✓	✓	✓
Lovelace	✓	✓	✓	✓
Integrations	✓	✓	✓	✓
Blueprints	✓	✓	✓	✓
Uses container	✓	✓	✗	✓
Supervisor	✓	✗	✗	✓
Add-ons	✓	✗	✗	✓
Snapshots	✓	✗	✗	✓
Managed OS	✓	✗	✗	✗

- **Obrázek 2.16** Možné způsoby instalace HomeAssistant a jejich podpora funkcionalit [27]

Na obrázku 2.16 je znázorněna podpora funkcionalit HomeAssistant s konkrétním typem instalace. Vysvětlení jednotlivých položek popsane na [27]:

- Automations – automatizace, nastavení zařízení, služeb a jejich konfigurace
- Lovelace – rychlý, přizpůsobitelný a mocný dashboard pro ovládání a kontrolu
- Integrations – integrace s jinými platformami, aplikacemi a službami (aktuálně 1771 integrací)
- Blueprints – návrh konfigurace pro automatizaci s nastavitelnými částmi, více konfigurací může být založeno na jednom návrhu

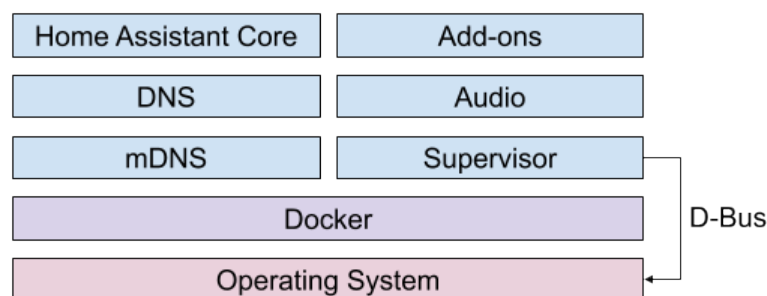
- Uses container – zda platforma využívá kontejnery
- Supervisor – umožňuje uživateli spravovat konkrétní instalaci a je zodpovědný za běh HomeAssistant Core, jeho update (případně roll back), vytváření a obnovu záloh, add-ons, sjednocený audio systém
- Add-ons – umožňují uživateli rozšířit funkcionalitu instalováním dodatečných aplikací
- Snapshots – záloha HomeAssistant a add-on dat a konfigurací
- Managed OS – správa operačního systému

2.4.2.4 Architektura

Obrázek 2.17 popisuje architekturu HomeAssistant s operačním systémem na kterém běží docker se základními stavebními prvky této platformy.

Význam jednotlivých bloků převzatý z [28]:

- HomeAssistant Core – více informací v části 2.4.2.2
- Add-ons – více informací v části 2.4.2.3
- DNS – umožňuje core a add-ons komunikovat mezi sebou
- Audio – umožňuje core a add-ons přehrávat audio
- mDNS – pomáhá objevovat a připojovat se na zařízení a služby v síti
- Supervisor – více informací v části 2.4.2.3
- Docker – služba pro nasazení aplikací jako přenosných kontejnerů
- Operating system – operační systém založená na linuxu
- D-Bus – komunikační systém na ovládání částí operačního systému



■ **Obrázek 2.17** Architektura HomeAssistant s operačním systémem na kterém běží docker se základními stavebními prvky této platformy [28]

2.4.2.5 Slabina

Při běhu HomeAssistant na Raspberry Pi je SD karta nejslabší stránka. Paměťové karty nejsou uzpůsobeny, aby zvládaly provoz při vysokém souběžném zatížení (mnoho I/O operací). HomeAssistant používá rozšířené logování a další operace, proto může dojít k poškození i vysoce kvalitních SD karet. Mohou být poškozeny ztrátou energie. Řešit by se to dalo externím SSD, ale to je zbytečné a je lepší se smířit s nějakým rizikem, které se ve většině případů neprojeví. [26]

2.5 Možnosti ovládání

Existují tři hlavní možnosti, jak lze ovládat chytré zrcadlo:

- Dotykové
- Hlasové
- Alternativní

U každého z nich je nejprve potřeba se zamyslet k jakým účelům se plánuje chytré zrcadlo využívat a v jakém prostředí bude zrcadlo umístěno. Tyto faktory silně ovlivňují volbu jednotlivých způsobů ovládání.

2.5.1 Dotykové ovládání

Dnes nejvíce používané pro pohyb po systému na nějakém zobrazovacím zařízení. Zejména díky chytrým telefonům, či tabletům, které má dnes téměř každý s výjimkou starší generace. Jedná o intuitivní způsob, jak komunikovat se zařízením. Konkurovat už může jen klávesnice s myší, tedy primární ovládací nástroj pro osobní počítač a jeho obdobu, nebo dálkové ovládání k televizi. To se v dnešní době řeší přidáním chytrých „ukazovátek“ a pro pohyb po systému se využívá namíření na danou komponentu na obrazovce. Pointer se přesouvá podle namíření ovladače a vzniká tak intuitivní, často ne tolik přesné řešení, které se v podstatě chová jako by se uživatel dotýkal prstem, jen je bezkontaktní.

Právě jednoduchost a rozšířenost je hlavní výhodou, při volbě dotykového ovládání. Ovšem je potřeba se zaměřit i na nevýhody, kterých není málo. První a asi nejzásadnější věc, která u chytrého zrcadla vadí, je upatlení a umaštění. Zkrátka stopy, které vzniknou po kontaktu s plochou zrcadla. Další je poškození nebo selhání dotykové plochy. Při volbě infračervené dotykové plochy mohou vadit nežádoucí falešné doteky. [29]

2.5.1.1 Infračervená dotyková plocha

Princip Tato zařízení podporují vstup na základě doteku, jako jakékoliv jiné dotykové zařízení. Způsob, jakým identifikují dotykové příkazy je kompletně unikátní. *Kapacitní* dotyková zařízení rozpoznávají dotek měřením změn kapacit kondenzátorů, oproti tomu *rezistivní* (odporová) dotyková zařízení zjišťují dotek zmáčknutím horní vrstvy ke spodní. Pro srovnání, *infračervené* dotykové plochy fungují na základě přerušení v jednotlivých paprscích světla.

Infračervené dotykové plochy mají do mřížky uspořádané pole LED diod a photodetektorů (senzorů) kolem okrajů daného zařízení. Tyto LED diody konstantně vyzařují, pro samotné oko neviditelné, světlo. Při dotyku povrchu vznikne narušení tohoto světla. Zařízení je pak schopné určit, kde k přerušení došlo a díky tomu můžeme přesně určit bod doteku. (převzato z [30])

Výhody Pro začátečníky nabízejí vynikající úroveň pevnosti a trvanlivosti. Kapacitní dotyková zařízení například vyžadují vzorkování na skle, což je následně dělá náchylnějšími k poškození. I při použití skla *Gorilla Glass* se kapacitní dotykové zařízení může při pádu na zem zlomit nebo rozbít. Infračervené dotykové plochy však nevyžadují vzorkování na skle, a proto jsou silnější a odolnější než jejich kapacitní protějšek.

Infračervená dotyková zařízení jsou navíc vysoce citlivá na dotykový vstup. U odporového zařízení je nutné dostatečně zatlačit, aby horní vrstva zatlačila do spodní vrstvy. Protože infračervené dotykové plochy identifikují dotyky hledáním rušení na světle, spustí dotykový příkaz i ten nejjemnější dotek. Díky tomu je vynikající volbou v aplikacích, kde je vyžadována citlivost. Navzdory tomu může ultracitlivá povaha infračervené technologie vyvolat chybný vstup, například falešný dotyk od nečistot nebo úlomků. (převzato z [30])

2.5.2 Hlasové ovládání

Je založeno na rozpoznání řeči (speech recognition). To převádí lidský hlas do dat tak, aby je mohl počítač zpracovat. Jedná se o vstup, který je možné využít, když není k dispozici například myš a klávesnice. V poslední době se rozrostly cloudové systémy na rozpoznání řeči, které umožnili integraci tzv. hlasových asistentů do mobilních zařízení a ovládacích systémů domácích sítí. Hlasový asistent dokáže využívat některé funkce systému, například napsat zprávu. Lze ho najít také v automobilovém odvětví u komunikace se systémem během řízení. Ukázalo se, že na bezpečnost u hlasových asistentů nebyl brán zřetel. Dříve když měl útočník zařízení fyzicky k dispozici, bylo těžké dostat se k soukromým informacím, pokud byl telefon uzamčen. Vývoj hlasových asistentů způsobuje zranitelná místa, protože je možné zadávat příkazy a přijímat soukromé informace i když je zařízení uzamčeno. Jedná se o ekvivalent malware v mobilním prostředí. [31]

Ačkoliv je hlasové ovládání dnes již v mnoha zařízeních, uživatelé ho téměř nepoužívají. Téměř každý smartphone má hlasového asistenta, ať už v operačním systému, či v konkrétních aplikacích. Nejvíce zajímavý je Google assistant, či Amazon Alexa. Propojení právě s těmito asistenty by mohlo vytvořit ovládací subsystém, který bude mít alespoň nějakou záruku spolehlivosti a funkčnosti, neboť se jedná o hlasové asistenty obrovských společností, které do jejich vývoje vkládají spoustu úsilí, peněz a času. Ovšem většina uživatelů dotykových mobilních telefonů funkci těchto asistentů nevyužije, či hlasové ovládání obecně. Je to jeden z novějších způsobů ovládání a pro převážně starší generaci se toto může jevit jako „hi-tech“, se kterým si myslí, že nebudou umět pracovat. Mnoho lidí ani neví, že má hlasové ovládání k dispozici. I přesto, že se jedná o do budoucna pravděpodobně běžný způsob ovládání, je stále ve vývoji. Lidé nejsou zvyklí na komunikaci se stroji, a tak je zcela přirozené, že nemají důvěru, že když budou na nějaké zařízení mluvit, že bude skutečně dělat to co má. [32]

Nabízí se otázka ochrany osobních údajů, bezpečnosti, či možnosti být špehován. Vzhledem k využití hotových asistentů od společností jako je Google nelze očekávat, že v licenčních podmínkách nebude napsáno, že mohou sbírat informace pro budoucí zlepšení. Zde je potřeba si říci, zda takovéto firmy skutečně jenom sbírají anonymní údaje, nebo si dělají záznamy a ukládají osobní informace. Z tohoto důvodu je hlasové ovládání bezpečnostní riziko pro osoby, které ho používají. Nejen, že může zachytávat informace k ovládání zrcadla, typicky nějaká klíčová slova, která se používají pro interakci se zrcadlem, ale může být v nějakém pohotovostním režimu, kde čeká na aktivaci klíčovým slovem. Jenže při tomto čekání sbírá data z mikrofonu, analyzuje, vyhodnocuje a posílá na nějaký vzdálený server právě vybrané společnosti.

2.5.3 Alternativní ovládání

Pod tímto pojmem se ukrývá ovládání pomocí nějakého externího joysticku, kapacitní klávesnice, trackpadu, touchpadu, myši a dalších podobných zařízení.

Všechny výše vyjmenované s sebou přináší komplikace. Pohyb po systému není intuitivní, mnohdy zvýšená citlivost ovládání může působit nepřesnost. Na druhou stranu velmi nízká citlivost těchto zařízení může vést ke ztrátě trpělivosti při komunikaci s chytrým zrcadlem. Může vzniknout problém, že pro obdržení výsledku nějakého jednoduchého úkolu, je nutné provést složité operace. To samozřejmě není účelem. Jde o usnadnění získání informací a o co největší komfort při interakci se systémem.

Mezi hlavní problémy patří přesnost, spolehlivost (jedná se o další kus hardwaru, který může selhat), umístění na zrcadle. Ovšem alternativní ovládání má i své výhody. Například joystick a trackpad nepodléhají vlivu prostředí, ve kterém jsou používány. Nedojde k upatlení zrcadlové plochy otisky prstů.

Stále ještě poněkud futuristické řešení, které je v neustálém vývoji a hodilo by se pro ovládání zrcadla je ovládání gesty. To by mohlo vyřešit problém s upatlením skla, a přitom nabídnout komfortní a intuitivní způsob ovládání. Druhá podobná možnost je ovládání na základě snímání

očí, tedy pokud se uživatel někam dívá dostatečně dlouho, dojde k nějaké aktivaci signálu apod. O něco pokročilejší by pak mohlo být ovládání pomocí myšlenek, ale pouze za předpokladu, že by nebylo nutné si na hlavu nasadit nějaký další hardware, který by mnohonásobně zvýšil cenu. Jak již bylo zmíněno, tyto technologie jsou ještě ve fázi vývoje a v době kdy je tvořena tato práce nejsou k dispozici. Výjimku tvoří pouze ovládání gesty, které se již v dnešní době dostává do různých zařízení.

2.6 Volba hardware

V této části bude vysvětleno na co se zaměřit při volbě zařízení, na kterém chytré zrcadlo poběží. Dále budou zmíněny požadavky pro tento konkrétní projekt. Zařízení budou rozdělena do dvou kategorií podle ceny. Jedna se snaží minimalizovat náklady a druhá nabízí dostatečný výkon.

2.6.1 Cena a zhodnocení výkonu

S ohledem na to, aby bylo zrcadlo do budoucna dostupné pro co největší množství lidí, je kladen důraz na cenu celého systému jako celku. Proto při volbě vhodného hardware je dobré se podívat na zařízení, která budou mít dostatečný výkon na běh systému a nějaké vykreslování grafiky na displej za sklem. Ovšem je zbytečné hledat produkty, které mají nadměrně vysoký výkon. Pro dostatečný výkon je nutné se poohlédnout po jednodeskových počítačích, které nabízejí vstupně/výstupní moduly, které je možné využít pro nějakou další konektivitu.

2.6.2 Požadavky na hardware

Z důvodu připojení k internetu pro získávání zpráv, notifikací apod. je vyžadováno, aby zařízení mělo Wi-Fi modul. V lepším případě lze najít Wi-Fi modul přímo na desce jednodeskového počítače, ale pokud tam není, lze ho přidat Wi-Fi adaptérem do USB. Samotný ethernetový port bude u tohoto projektu brán pouze jako volitelný, tedy není nutné, aby ho zařízení mělo. Pro stabilní připojení je lepší dát uživateli na výběr a stejně jako v případě chytrých televizí ho mít k dispozici.

Kvůli operačnímu systému Raspbian a běžící aplikaci chytrého zrcadla, využívající aplikační wrapper Electron je potřeba určit minimální velikost operační paměti. Z neoficiálních zdrojů (dostupných z [33]) lze vyčíst, že Raspbian s desktopem využívá přibližně 450 MB operační paměti. Stejným způsobem lze vyhledat i aplikace spuštěné s wrapperem *Electron*, kde jednoduchý *HelloWorld* příklad zabírá necelých 350 MB. Uživatelé vyvíjející v tomto frameworku tvrdí, že typická aplikace si žádá 500 MB operační paměti (informace dostupné z [34]). V součtu, je nutné počítat s absolutním minimem. Pro projekt bude potřeba alespoň 950 MB RAM. Je dobré se ale zamyslet nad funkcionalitou a modularitou zrcadla. V případě využití nějakého kamerového systému a streamování (zobrazení v reálném čase) jeho obsahu na displej chytrého zrcadla, by si aplikace mohla nárokovat více než 500 MB. Pro jistotu je dobré mít nějakou rezervu. Jako jednoznačné minimum se jeví 1 GB, ale pro správnou funkcionalitu a plynulý běh systému se budou zmíněna o zařízení alespoň 2 GB.

Při volbě hlasového ovládání je potřeba mikrofon pro vstup a reproduktory pro výstup. Zde jsou priority poněkud flexibilní. Není nutné se vázat pouze na konkrétní typy konektorů, jako je například 3,5 mm audio jack pro zvuková zařízení. Pro jednoduchost bude hlavní, aby byly dostupné dvě z následujících možností připojení zvukových zařízení:

- USB
- Audio jack 3,5 mm
- Bluetooth

Vše je možné řešit pomocí externích rozšiřujících karet, které jsou řízeny vstupně/výstupními porty, nebo USB redukce. Každá součástka navíc něco stojí a může zvýšit poruchovost celého systému.

2.6.3 Kategorie

Z nabízených produktů, které jsou dostupné na trhu lze vytvořit dvě hlavní kategorie pro tento projekt, a to:

- Low cost
- Výkon

K výsledné ceně zařízení se musí připočítat cena poštovního a balného, která se u produktů z cizích trhů může vyšplhat až na 430 Kč (\$20). Tím to ale nekončí, cenu zvedne i napájecí zdroj. V mnoha případech lze využít nějakou starší nabíječku, kterou má každý doma, pokud odpovídá její konektor. Ne vždy je to ale možné, a ne vždy tyto nabíječky dokážou poskytnout dostatečný proud pro zařízení. Pokud to nestačí, je nutné dokoupit napájecí zdroj, jehož cena se pohybuje okolo 250 Kč. Volitelná jsou pak pouzdra na zařízení, nebo například externí pasivní, či aktivní chladiče. V tomto projektu nebude využito ani jednoho.

2.6.3.1 Low cost

Zde budou zahrnuta zařízení s cenou okolo 430 Kč (\$20). Tato zařízení by se dala použít pro nějakou odlehčenou verzi chytrého zrcadla, kde by nevadilo, že přechody při zobrazení by nemusely být vždy plynulé. Pokud by někomu šlo především o co nejmenší cenu a byl by smířen s tím, že se obraz pravděpodobně bude sekát, přichází v úvahu zařízení z této kategorie.

Jednodeskový počítač	AML-S805X-AC (LA FRITE)	RASPBERRY PI ZERO WH	BANANA PI BPI-M2 ZERO	ORANGE PI PC
Výrobce	Libre Computer	Raspberry Pi	Sinovoip	Xunlong Software
Cena	430 Kč (\$20)	437 Kč	388 Kč (\$18)	345 Kč (\$16)
CPU	4 ARM Cortex-A53 @ 1.2 GHz	1 GHz single-core ARM1176JZF-S	H2+ Quad-core Cortex-A7	H3 Quad-core Cortex-A7
GPU	2G + 3P ARM Mali-450 @ 650 MHz	Broadcom VideoCore IV @ 250 MHz	Mali400MP2 @ 600 Mhz	Mali400MP2 @ 600 Mhz
RAM	1 GB	512 MB	512 MB	1 GB
USB	2 × USB 2.0 Type A	1 × micro-USB	1 × USB 2.0 OTG	3 × USB 2.0 HOST, 1 × USB 2.0 OTG
Network	100 Mb Fast Ethernet	802.11n Wi-Fi	802.11b/g/n Wi-Fi	100 Mb Ethernet
Video Out 3,5 mm Jack	HDMI 2.0	Mini-HDMI	Mini-HDMI	HDMI Ano
Storage	eMMC 5.x Interface	MicroSDHC	TF card	TF card/MMC card
IR	Receiver	—	—	Receiver
Další	U-Boot Button 40 Pin Low Speed Header (PWM, I2C, SPI, GPIO)	17x GPIO, HAT ID sběrnice	40 Pin Header, Power button, Reset button	40 Pin Header, Power button

■ **Tabulka 2.7** Zařízení patřící do kategorie *low cost* [35, 36, 37, 38]

2.6.3.2 Shrnutí

Důvod, proč jsou uvedena zařízení mající pouze 512 MB operační paměti RAM je, že nějací nadšenci již zkoušeli konkrétně na *Raspberry PI Zero WH* spustit MagicMirror² OS a jimi vytvořené chytré zrcadlo se jim podařilo zprovoznit. Byl problém pouze s plynulostí rozhraní. Samotný obraz se sekal a nedělalo to dobrý dojem. Nicméně pokud někomu sekání nevádí a chtěl by maximálně ušetřit, je to možné řešení.

U všech výše zmíněných jednodeskových počítačů je operační paměť sdílená s grafickým procesorem. Dále je důležité zmínit, že veškeré ceny jsou uvedeny pouze za zařízení samotné. Doprava není v ceně započtena a pohybuje se okolo \$4–20. To je nezanedbatelná částka. Stejně tak je možné říci, že všechny produkty, kromě těch od firmy Raspberry, se na českém trhu nevyskytují, a tak je nutno počítat s poštovním primárně z Číny. Doba dodání je pak různá, ale zpravidla pokud si někdo nechce připlatit za extra rychlé doručení, pohybuje se od jednoho, do dvou měsíců.

Jako nejvhodnější volba vychází *Orange PI PC*, a to i přesto, že nemá Wi-Fi. Stejně tak je na tom i jeho největší možný konkurent, *La Frite*. Ten ale nemá 3,5 mm audio jack a vzhledem k pravděpodobnému využití USB Wi-Fi adaptéru se musí počítat s tím, že oba USB porty budou obsazeny a kdyby si někdo chtěl připojit nějaké další extra zařízení přes USB port, neměl by kam ho připojit. *Orange PI PC* je na tom v tomto ohledu lépe. Při osazení dvou USB portů stále jeden zůstane k dispozici. Díky TF slotu je možné vložit paměťovou kartu s nahraným operačním systémem MagicMirror² OS. Pro využití nějakých dalších modulů se na zařízení vyskytuje 40 vstupně/výstupních pinů. Čtyřjádrový procesor nabízí dostačující výpočetní výkon, nicméně 1 GB operační paměti by mohl působit problémy. Na desce se taktéž nachází vestavěný mikrofon, který pro samotné použití zrcadla bude umístěn na špatném místě. Pro komunikaci se zrcadlem se zdá být nepoužitelný, ale pro vývoj a debugování je to bonus. S cenovkou \$16 + \$4 poštové se cena dostane na \$20, tedy krásných 430 Kč, to odpovídá ceně za samotné zařízení *La Frite*.

Na *Raspberry PI Zero WH* je sice možné zrcadlo zprovoznit, ale jeho funkčnost není dobrá (512 MB RAM je opravdu málo pro tento projekt). Dle předchozích údajů není dostačující a ani jeho obdoba *Banana PI BPI-M2 Zero* s vícejádrovým procesorem, lepším grafickým procesorem není vhodná volba pro tento projekt. Jediná výhoda, kterou *Raspberry PI Zero WH* má je, že je dostupný na českém trhu, má kvalitní manuál a dokumentaci. Díky popularitě produktů firmy Raspberry existuje mnoho návodů a detailně popsanych projektů s tímto zařízením.

2.6.3.3 Výkon

Zde budou uvedena zařízení s cenovkou přibližně \$40 (870 Kč). Tato zařízení mají dostatečný výkon, a proto není třeba řešit dražší zařízení. Je velmi důležité zmínit, že výrobci jednodeskových počítačů je v dnešní době tolik, že není možné vybírat ze všech produktů, které jsou reálně k dispozici, ale pouze z těch, které lze na internetu jednoduše najít. Pokud zde nějaké zařízení, které by bylo vhodné pro realizaci tohoto projektu, chybí, pak to není úmyslné.

Při hledání možných jednodeskových počítačů se vycházelo z hlavního trendu v dnešní době, který udává asi nejznámější společnost v tomto odvětví – *Raspberry*. Zařízení Raspberry PI 4 Model B s 2 GB RAM bude stačit a je označeno jako výchozí zařízení. Proto pro hledání ostatních zařízení bylo využito vyhledávání Google. Cílem bylo najít alternativy právě zmíněného zařízení. Výrobce *Libre Computer* působí solidně a nabízí více produktů, které se pohybují s cenovkou do \$50 a disponují dobrou konektivitou.

Jednodeskový počítač	ALL-H3-CC H5 (Tritium)	ROC-RK3328-CC (Renegade)	AML-S905X-CC (Le Potato)
Výrobce	Libre Computer	Libre Computer	Libre Computer
Cena	650 Kč (\$30)	866 Kč (\$40)	758 Kč (\$35)
CPU	Quad-core ARMv8 Cortex-A53	4 ARM Cortex-A53 @ 1.4 GHz	4 ARM Cortex-A53 @ 1.512 GHz
GPU	ARM Mali-450 MP4 @ 700 MHz	2G + 2P ARM Mali-450 @ 500 MHz	2G + 3P ARM Mali-450 @ 750 MHz
RAM	2 GB	2 GB	2 GB
USB	4 × USB 2.0 Type A	2 × USB 2.0 Type A 1 × USB 3.0 Type A	4 × USB 2.0 Type A
Network	100 Mb Fast Ethernet	Gigabit Ethernet	100 Mb Fast Ethernet
Bluetooth	—	—	—
Video Out	HDMI 1.4	HDMI 2.0	HDMI 2.0
3,5 mm Jack	Ano	Ano	Ano
Storage	Micro SD Card Slot, eMMC 4.x Interface	Micro SD Card Slot, eMMC 5.x Interface	Micro SD Card Slot, eMMC Interface
IR	Receiver	Receiver	Receiver
Other	U-Boot Button, 40 Pin Low Speed Header (PWM, I2C, SPI, GPIO), UART Header	U-Boot Button, 40 Pin Low Speed Header (PWM, I2C, SPI, GPIO), ADC Header	U-Boot Button, 40 Pin Low Speed Header (PWM, I2C, SPI, GPIO), UART Header, AUDIO Headers (I2S, ADC, SPDIF)

■ **Tabulka 2.8** Zařízení patřící do kategorie *výkon část 1*. [39, 40, 41]

Jednodeskový počítač	ROCK64 Rev2	BPI-M64	Pi 4 Model B
Výrobce	Pine64	Sinovoip	Raspberry Pi
Cena	866 Kč (\$40)	1194 Kč (\$55)	1059 Kč
CPU	Rockchip RK3328 Quad-Core ARM Cortex A53 @ 1,5 GHz	Allwinner Quad-core ARM Cortex A53 @ 1,5 GHz	Quad-core ARM Cortex A72 @ 1,5 GHz
GPU	ARM Mali-450 MP2 @ 500 MHz	Mali 400 MP2 @ 500 MHz	Broadcom VideoCore VI @ 500 MHz
RAM	2 GB	2 GB	2 GB
USB	2 × USB 2.0 Host, 1 × USB 3.0 Host	2 × USB 2.0, 1 × USB 2.0 OTG	2 × USB 2.0, 2 × USB 3.0
Network	Gigabit Ethernet	100 Mbit Ethernet, 802.11b/g/n Wi-Fi	Gigabit Ethernet, 2,4GHz a 5GHz IEEE 802.11.b/g/n/ac Wi-Fi
Bluetooth	—	Bluetooth 4.0	Bluetooth 5.0 (BLE)
Video Out	HDMI 2.0	HDMI 1.4	2 × MicroHDMI 2.0
3,5 mm Jack	Ano	Ano	Ano
Storage	Micro SD Card Slot, eMMC Interface, SPI Flash 128 Mbit	Micro SD Card Slot, 8G eMMC Flash onboard	Micro SD Card Slot
IR	Receiver	Receiver	—
Other	40 GPIO Pin Header (Pi2), 22 GPIO Pin Header (Pi PF5+), Power, Reset, Recovery buttons	28 GPIO Pin Header, UART, I2C, SPI, PWM	40 GPIO Pin Header

■ **Tabulka 2.9** Zařízení patřící do kategorie *výkon část 2*. [42, 43, 44]

2.6.3.4 Shrnutí

Z těchto šesti produktů vychází nejlépe jednoznačně *Raspberry PI 4 Model B*, který vyniká zejména konektivitou – nabízí, jak možnost připojit Ethernetový kabel, tak Wi-Fi a dokonce je možné připojit i zařízení s Bluetooth. Má 3,5 mm Jack pro připojení reproduktorů. Výkonu nabízí procesorové jednotky firmy Broadcom dostatek a 2 GB paměti RAM jsou taktéž dostačující. Na tomto počítači jsou k dispozici dohromady 4 USB porty, slot na Micro SD kartu a 40pinový GPIO header pro připojení senzorů a obdobných periférií.

Poněkud hůře si ve srovnání s ostatními vedou produkty společnosti Libre Computer, a to zejména kvůli absenci Wi-Fi, avšak díky nízké ceně se dá dokoupit externí USB adaptér. Nejlepší volba z výše zmíněných tří zařízení je *Le Potato*. Oproti *Renegade* má o něco vyšší výkon a o jedno USB navíc. Myšlenka, že je možné říkat, že chytré zrcadlo běží na bramboře (přeloženo), nebo že se o chod zrcadla stará brambora, zní velmi lákavě, ale pobavení na výkonu nepřidává.

ROCK64 Rev2 je specifikací velmi podobný produktům Libre Computer, oproti nim má navíc ještě další 22pinový GPIO header. BPI-M64 jakožto produkt nejvíce se blížící k PI 4 Model B by musel nabídnout nižší cenu, jinak není důvod volit toto zařízení, když je k dispozici lepší a výkonnější PI 4 Model B.

Jak již bylo výše zmíněno, k volbě vítěze jednoznačně přispěla dostupnost na českém trhu, kvalitní návody a dokumentace. Obrovská komunita nadšenců a fóra, kde je možné se v případě jakýchkoliv problémů někoho zeptat s garancí odpovědi v blízké době. Preferovaný MagicMirrorOS je navržen a otestován právě na zařízeních společnosti Raspberry PI, a tak i tato práce bude vyvíjena na aktuálním modelu PI 4 Model B.

Návrh a realizace

Tato kapitola se zabývá návrhem konstrukce a samotnou realizací zrcadla, s ohledem na cílové umístění. Prvky a technologie použité ke konstrukci zrcadla vychází především z předchozí kapitoly analýzy.

V první fázi této kapitoly je řešena samotná konstrukce nosných částí a vývoje rámu chytrého zrcadla. Další části jsou potom věnovány přípravě hardware platformy Raspberry Pi na kterou je provedena vlastní instalace HomeAssistenta a dalších podpůrných systémů Kalliope, pro zprovoznění požadovaných funkcí ovládání světel, hlasového ovládání atd. V závěru kapitoly se nachází instalační příručka a jsou zde popsány možné problémy, které mohou zprovoznění chytrého zrcadla či samotnou instalaci doprovázet.

3.1 Vhodné ovládání a konstrukce

V této sekci práce bude popsáno zvolení vhodného ovládání podle typu místností a postup při vytvoření konstrukce.

3.1.1 Vhodné ovládání podle místností

Tato část práce analyzuje místnosti, kde by se zrcadlo mohlo nacházet. Informuje o předpokládaném způsobu užití skla, rozboru vhodného ovládání a problémech spojených s typem místnosti.

3.1.1.1 Ložnice

V ložnici je kvůli posteli, dekám, závěsům a oblečení zvýšená prašnost. Nejsou v ní problémy s vlhkostí (záleží na umístění, například u oken může být zvýšená vlhkost).

Předpokládané užití Uvažuji situaci, že v domácnosti není šatna, tedy oblečení je uloženo v ložnici. Zde může zrcadlo sloužit pro kontrolu outfitu, tedy jak se člověk cítí v tom oblečení, které má právě na sobě. Jsou lidé, kteří si zkouší mnoho různých kusů oblečení, než si vyberou, v čem se cítí dobře. Tito lidé by využili služeb chytrého zrcadla, protože před ním stráví hodně času. Nemusí tak neustále koukat hodiny, či mobilní telefon, pro kontrolu, jestli ještě nejdou pozdě. Zrcadlo je může informovat, kdy jim jede spoj, či jaká je dopravní situace, počasí venku apod. S využitím chytrého zrcadla by měli vše pod kontrolou a mohli by si zjistit například jak dlouho jim potrvá cesta do práce.

Další z možných variant je zrcadlo na líčení. Ve chvíli, kdy se ženy líčí, nemají volné ruce a určitě by ocenily mít pod kontrolou čas, tak aby nemusely neustále odvracet zrak na samostatné hodiny.

Rozbor ovládnání Do ložnice bych volil hlasové ovládnání. Z výše zmíněných případů užití je zřejmé, že se očekává, že se nebude moci zrcadlo efektivně ovládat rukama. Díky klidným podmínkám jsou eliminovány problémy s analýzou řeči.

Problémy V ložnici se může nacházet televize, počítač a další zařízení, která dokážou vydávat zvuk. To může být potenciální problém pro hlasové ovládnání. Může se stát, že zrcadlo bude reagovat na zvuk přicházející z nějakého jiného zařízení. To může vést na problémy s rozpoznáním řeči. Uživatel během této doby nebude schopen jakkoli ovládat zrcadlo.

3.1.1.2 Pracovna

V pracovně bývá klid, vlhkost (výjimku mohou tvořit umístění vedle oken) ani prašnost zde nehraje roli.

Předpokládané užití Zde funkcionality samotného zrcadla takřka postrádá smysl. Nabízí se využít plochu zrcadla jako dashboard pro monitoring domácnosti, zobrazení aktuálního kalendáře s naplánovanými událostmi a dalších systémů podle preferencí. Zrcadlo je užitečné na kontrolu vzhledu před případným online videohovorem.

Rozbor ovládnání Pokud by někdo chtěl mít zrcadlo v pracovně, hlasové ovládnání je jasná volba. Nejsou potřeba ruce na interakci se zrcadlem a díky klidu nebude mít analýza řeči okolní ruch, čímž se zvýší přesnost rozpoznání slov.

Problémy Jediné, co by mohlo uživatelům vadit je bezpečnost, tedy ochrana osobních údajů, která představuje pro hlasové ovládnání problém.

3.1.1.3 Obývací pokoj

Standardně se v obývacím pokoji nachází televize. Je nutné provést pár experimentů, aby mohlo být rozhodnuto, zda zvuk z televize může být brán jako vstup pro hlasové ovládnání. Pokud ano, nastal by problém, že chování chytrého zrcadla by mohlo být nepredikovatelné. Nelze určit, jaká konkrétní slova by se z televize dala odchytnout a eventuálně provést odpovídající příkazy. Pokud ne, televize by nepředstavovala jiný problém než ruch. Docházelo by tak k zhoršení přesnosti rozpoznání klíčových slov nezbytných pro ovládnání systému v zrcadle.

Předpokládané užití Zrcadlo v obýváku se hodí jako dashboard na monitorování chytré domácnosti. Nebo například jako místo, kde lze mít přehledně zobrazené úkoly pro členy domácnosti (nějaké body ohledně rozdělení domácích prací, povinnosti apod.). Televizi to nenahradí, mobilní telefon také ne, ale obývací pokoj je často průchozí a zde se hodí funkcionality zrcadla. Pokaždé když někdo projde, může si zkontrolovat nejen to, jak zrovna vypadá.

Rozbor ovládnání Zde je opravdu na výběr, dokonce nejlepší možnost je zkombinovat jak dotykové ovládnání, tak hlasové. Hlasové nabízí při použití v obývacím pokoji mnoho výhod, hlavní je komfort. Není nutné být v přímém kontaktu se zrcadlem, nicméně s ohledem na zapnutou televizi je dobré mít k dispozici nějakou jinou variantu ovládnání. Dotykové ovládnání je jednoduché, přirozené a pro tuto místnost se hodí.

Problémy Ruch od televize bude zhoršovat analýzu řeči. Zvýšená prašnost této místnosti znamená časté čištění. V případě vystavení přímému slunci nebudou informace na displeji vidět zřetelně.

3.1.1.4 Pokoj

V pokoji je klid, pokud se nejedná o dětský pokoj. V této místnosti je nutné počítat se zvýšenou prašností, obzvláště pokud se v ní nachází záclony a poličky. V blízkosti oken může být zvýšená vlhkost.

Předpokládané užití Zrcadlo zde plní svoji funkci a chytré doplňky nahrazují nástěnné hodiny, papírový kalendář apod. Vznikne místo, kde má člověk vše pohromadě a nemusí mít tyto věci na stole. V pokoji se můžou nacházet i další prvky chytré domácnosti, které lze ovládat například hlasem.

Rozbor ovládání Stejně jako u předchozí místnosti, i zde se jeví kombinace hlasového a dotykového ovládání jako vhodná volba. V blízkosti například stolu by bylo lepší dotykové ovládání, ale hlasové nám umožňuje přístup z celé místnosti.

Problémy Největší problém představují malé děti, které jsou mnohdy velmi hlasité. Vytváří tak ruch, který brání správné analýze řeči. V případě že v pokoji nejsou malé děti, problém tvoří pouze zvýšená prašnost.

3.1.2 Konstrukce

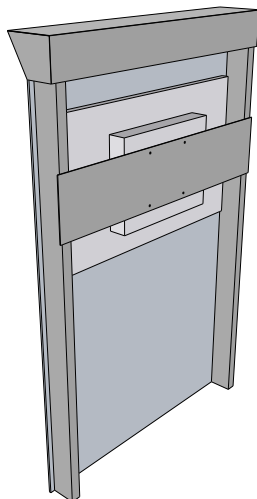
Pro vymyšlení rozměrů jsem se inspiroval zrcadlem, které mám v koupelně. To má rozměry 60×100 centimetrů a nahoře má přídavné osvětlení. Rozhodl jsem se použít monitor s úhlopříčkou 22", který má na šířku 51 cm. Aby zrcadlo co nejlépe využilo plochu nad rameny, kde je volný prostor, zvolil jsem orientaci monitoru na šířku. Mým záměrem bylo vytvořit unikátní bezrámečkovou konstrukci, pro kterou je nutné, aby sklo mělo větší šířku než monitor a zároveň mělo nějakou rezervu pro uchycení monitoru do konstrukce. Rozměry hledaného skla tak činí minimálně 56×80 centimetrů.

Samotnou konstrukci jsem probíral se svým kamarádem Janem Wijnhorstem, se kterým jsme po několika diskuzích a konzultacích vytvořili 3D model (obrázky 3.1, 3.2) chytrého zrcadla. Konstrukce se skládá ze dvou nosných profilů, které jsou zboku přidělané k monitoru pomocí šroubů. Mezi nosnými profily je ocelový plát, který s nimi vytváří hlavní nosnou konstrukci ve tvaru písmene „H“, na tento plát je pak přidělaná zadní část monitoru. Nahoře je panel (bedýnka), kde jsou umístěny reproduktory, LED pásek, Raspberry Pi a veškerá chytrá zařízení. Zepředu je sklo, které je přilepené k nosným profilům.

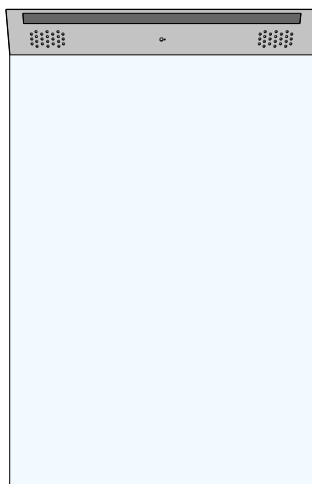
3.1.2.1 Postup vytvoření konstrukce

Nejprve je potřeba vybrat materiál, ze kterého bude konstrukce vytvořena. Zvolil jsem hliník, kvůli jeho stříbrnému vzhledu a ocel. Polopropustné sklo jsem si objednal a nechal na míru vyříznout od Sklenářství Praha 6 - Tomáš Vávra [45], na rozměry 56×80 centimetrů a nechal jsem zbrousit hrany. Po objednání hliníkového plechu, šroubků, pantů, profilů apod. je třeba rozvrhnout si práci a postupovat systematicky. Je dobré používat nástroje pro práci s kovem.

Nyní je potřeba připravit monitor, to znamená sundat zbytečný plast okolo něj (obrázek rozebraného monitoru 3.3). Poté přichází na řadu zkrácení hliníkových profilů a vyvrtání děr v místech, kde má monitor závity pro šroubky. Ve chvíli, kdy je možné přidělat profily k monitoru, zbývá naměřit a vyvrtat otvory pro uchycení ocelového plátu tak, aby konstrukce vytvořila tvar písmene „H“ (obrázek s uchycením monitoru k profilům 3.4).



■ **Obrázek 3.1** Zadní strana 3D modelu chytrého zrcadla



■ **Obrázek 3.2** Přední strana 3D modelu chytrého zrcadla



■ Obrázek 3.3 Rozebraný monitor bez okolního plastu

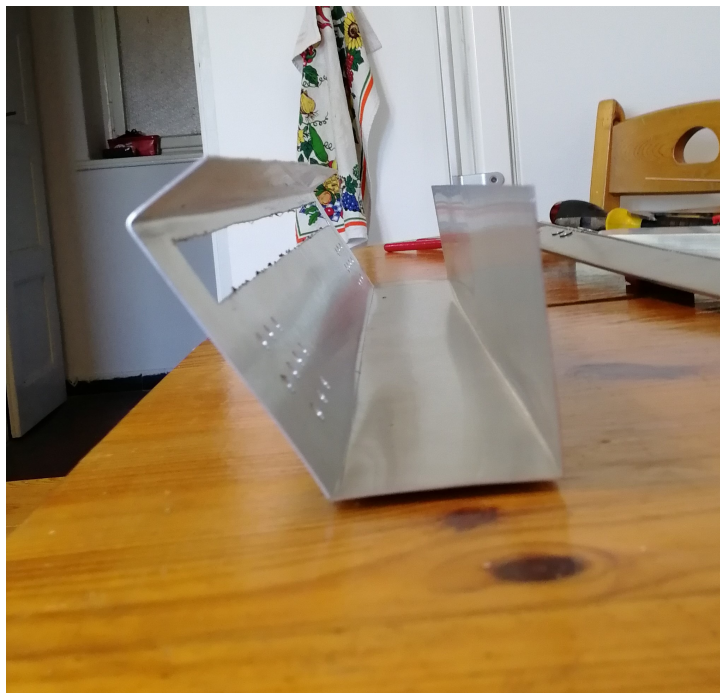


■ Obrázek 3.4 Uchycení monitoru k nosným profilům

Po vytvoření nosné konstrukce zbývá vytvořit horní panel, do kterého přijde téměř veškerá elektronika. Pro tuto část konstrukce jsem vybral hliníkový plát vyměřený tak, aby z něj bylo možné sestavit dostatečně velký panel, aby se do něj vešlo Raspberry Pi. Plát je nutné zmenšit na požadované rozměry a začít ohýbat tak, aby se vytvořila konstrukce ve tvaru „U“. Výsledek ohýbání lze vidět na obrázku 3.5. Poté je potřeba připravit a připevnit boční krytky a na panty shora přidělat servisní dvířka, aby bylo jednoduché přistupovat ke komponentám uvnitř, ale zároveň nebylo poznat, že tam je nějaká „díra“. Zkompletovaný horní panel vypadá takto 3.6. Ještě je potřeba vyvrtat díry na spodní stranu panelu pro odvod teplého vzduchu z monitoru. Výsledná konstrukce vypadá takto 3.7. Na spodní straně profilů jsou připevněné dva úhelníky zkrácené tak, aby nepřesahovaly sklo. Úhelníky slouží sklu jako podpora, pro případ, že by

lepící páska začala povolovat. A zbývá poslední věc – přilepit sklo. Pro přilepení skla jsem zvolil černou *3M pásku* širokou 19 mm, tlustou 1,1 mm, která pokryla celou plochu profilů a sklo je k nim přilepené opravdu pevně.

S tvořením celé konstrukce mi pomáhal můj kamarád Jan Wijnhorst, který mi poskytl vybavení a čas.



■ **Obrázek 3.5** Výsledek ohybu plechu připomínající tvar písmene „U“



■ **Obrázek 3.6** Zkompletovaný horní panel konstrukce



■ Obrázek 3.7 Hotová konstrukce při pohledu ze zadní strany, bez skla

3.2 Instalace systému

Tato část práce se zabývá výběrem a instalací operačního systému pro chytré zrcadlo a jeho základním nastavením, aby na něm mohla být spuštěna platforma MagicMirror², HomeAssistant a Kalliope. Dále je zmíněný postup na instalaci těchto platforem. Jejich nastavení vychází z dokumentací dostupných na [25, 46, 47].

3.2.1 Volba a natavení operačního systému

Platforma HomeAssistant podporuje pouze operačním systémem Debian 10 (Buster). Není vyloučeno, že by na ostatních derivativech nefungovala, ale není to oficiálně zaručeno. Byly zde nějaké pokusy od ostatních tvůrců, které na starších verzích dokázaly zprovoznit HomeAssistant na Raspbianu. Tento návod jsem zkusil aplikovat. Tento postup skončil chybou a platformu se nepodařilo spustit v režimu Supervised, který je potřeba, aby bylo možné přidat další platformu (ESPHome) jako tzv. „add-on“. MagicMirror² je testovaný a vyvíjený na operačním systému Raspbian, který disponuje grafickým rozhraním a mnoho potřebných systémových balíčků má již v sobě předinstalovaných. Pro běh chytrého zrcadla je tedy nutné mít operační systém s grafickým rozhraním. Kalliope, modulární always-on hlasem ovládaný osobní asistent navržený pro domácí automatizaci, podle dokumentace dostupné z [47] podporuje operační systémy Raspbian, Debian Jessie/Stretch a Ubuntu.

Po zhodnocení dostupných možností jsem vybral Debian 10 (Buster), kterému ručně nainstaluji grafické rozhraní, aby na něm bylo možné spustit MagicMirror². Asistent Kalliope je zároveň funkční na jiných derivativech Debianu, proto je veliký předpoklad, že Buster bude také fungovat.

Nejprve je potřeba stáhnout čistý Debian 10 (Buster) image z [48] a pomocí softwaru *balena-Etcher* nahrát image na paměťovou kartu. Já osobně jsem zvolil kartu o velikosti 32 GB, protože 8 GB by pro tento projekt mohlo být nedostatečné a cena 16 GB a 32 GB paměťových karet je u nejlevnějších produktů stejná. Po nahrání operačního systému se karta vloží do Raspberry Pi. Je důležité připojit kabel na monitor, klávesnici a napájení. Po spuštění je potřeba se přihlásit jako root (do *login*: napsat „root“ a pokračovat bez hesla). Pro prvotní nastavení je nutné připojit zařízení k internetu kabelem, bezdrátové připojení se nakonfiguruje později. Následuje sada příkazů:

1. Pro zajištění správných verzí balíčků

```
$ apt update
```

2. Nastavení správného času (je potřeba si někde zjistit aktuální unix timestamp a nahradit jím placeholder `<timestamp>`)

```
$ date +%s -s @<timestamp>
```

3. Nainstalovat nějaký konzolový editor (já zvolil *nano*) a balíček pro správu sítě

```
$ apt install nano network-manager
```

4. Smazat veškeré konfigurace síťových rozhraní (vyčistit)

```
$ rm /etc/network/interfaces.d/*
```

5. Otevřít v editoru *NetworkManager.conf*, změnit *managed* z „false“ na „true“

```
$ nano /etc/NetworkManager/NetworkManager.conf
```



```
[main]
plugins=ifupdown,keyfile

[ifupdown]
managed=false

[device]
wifi.scan-rand-mac-address=no
```

■ **Obrázek 3.8** Nastavení sítě v souboru *NetworkManager.conf*

6. Restartovat zařízení

```
$ reboot
```

7. Po opětovném spuštění systému je potřeba nastavit Wi-Fi pomocí *nmcli* (network manager command line interface)

Spuštění skenování dostupných sítí Wi-Fi a jejich výpis

```
$ nmcli device wifi rescan
$ nmcli device wifi list
```

Ukázka výpisu z příkazu `$ nmcli device wifi list` (hvězdička označuje aktivní připojení, v této fázi tam žádná nebude)

Pokud se ve výpisu nachází jméno sítě, ke které se chci připojit, mohu pokračovat, jinak je potřeba zajistit lepší propojení mezi přístupovým bodem a Raspberry Pi

```
mirror@rpi4-20210210:~$ nmcli device wifi list
IN-USE  SSID          MODE  CHAN  RATE      SIGNAL  BARS  SECURITY
*        NoConnection  Infra 11    65 Mbit/s  80      █████ WPA2
        SOMK         Infra 11    270 Mbit/s 73      █████ WPA1 WPA2
        INTERNETDOMA Infra 3     130 Mbit/s 65      █████ WPA1 WPA2
        UPC1605445   Infra 1     130 Mbit/s 59      █████ WPA1 WPA2
        Amálka      Infra 1     130 Mbit/s 55      █████ WPA1 WPA2
        WiFe        Infra 11    130 Mbit/s 49      █████ WPA1 WPA2
        UPC6F4447D   Infra 6     270 Mbit/s 44      █████ WPA1 WPA2
mirror@rpi4-20210210:~$
```

■ **Obrázek 3.9** Ukázka výpisu z příkazu `$ nmcli device wifi list` s aktivním připojením k síti *SOMK*

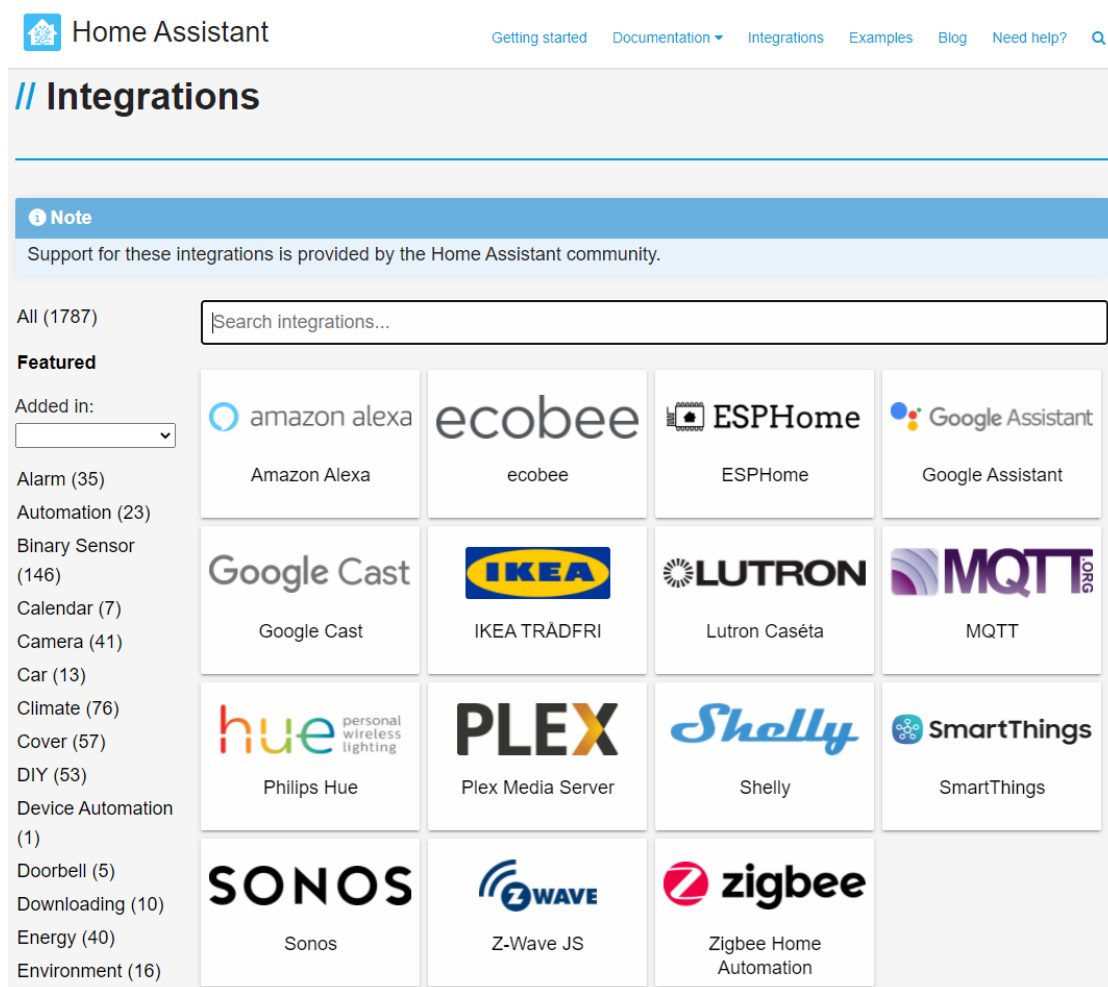
8. Připojení k síti, placeholder `<SSID>` je třeba nahradit skutečným SSID konkrétní sítě a `<password>` heslem k ní; tento postup funguje pouze pro jednoduché připojení k síti pomocí hesla, pokud je síť zabezpečena lépe, například nějakým dalším přihlašováním, je potřeba vyhledat v manuálu *nmcli*, jak se k takové síti připojit

```
$ nmcli device wifi connect <SSID> password <password>
```

Základní nastavení systému je hotové, zařízení je připojeno k síti pomocí Wi-Fi a je připraveno pro instalaci platformy HomeAssistant.

3.2.2 Instalace platformy HomeAssistant

Pro tento projekt je nutné nainstalovat platformu HomeAssistant v režimu Supervised, a to kvůli integraci platformy *ESPHome*, která se do HomeAssistant přidává jako tzv. „Add-on“. jedná se o formu rozšíření HomeAssistant o další platformy, některé z nich je možné vidět na obrázku 3.10. Tyto „Add-ons“ se vyskytují pouze v instalaci režimu Supervised a OS viz 2.16. Důvodem, proč chci integrovat platformu ESPHome, je, že mám doma vývojové desky *NodeMCU v2* a *NodeMCU v3*, které jsou osazené čipem *ESP8266* od firmy Espressif [49]. Tyto čipy v sobě mají integrovaný Wi-Fi modul, což umožní platformě HomeAssistant komunikovat s těmito zařízeními, kontrolovat je a monitorovat. Zároveň jsou vývojové desky cenově dostupné, stojí přibližně 70 Kč a tím se stávají ideálními prvky pro chytrou domácnost.



■ **Obrázek 3.10** Ukázka platform integrovaných do HomeAssistant ve formě rozšíření zvaných „Add-ons“ [50]

Postup pro instalaci platformy HomeAssistant:

1. Instalace *curl* (Client URL) sloužící ke stahování souborů přes počítačovou síť

```
$ apt install curl
```

2. Získání a uložení instalačního skriptu jako *get-docker.sh*

```
$ curl -Lo get-docker.sh https://get.docker.com
```

3. Instalace dockeru spuštěním staženého instalačního skriptu

```
$ sh get-docker.sh
```

4. Instalace *jq*, nástroje pro „parsování“ formátu JSON v příkazové řádce

```
$ apt install jq
```

5. Získání a uložení instalačního skriptu pro HomeAssistant Supervised jako *installer.sh*

```
$ curl -Lo installer.sh https://raw.githubusercontent.com/home-assistant/supervised-installer/master/installer.sh
```

6. Instalace HomeAssistant Supervised spuštěním staženého instalačního skriptu s parametrem *-m quemuarm-64*, který specifikuje typ zařízení; instalace chvíli trvá, je nutné počkat na její dokončení

Na všechny otázky v příkazové řádce s odpovědí typu „Y/N“ zmáčknout *Y* a potvrdit enterem

```
$ bash installer.sh -m quemuarm-64
```

Instalace platformy HomeAssistant je tímto hotová. Nyní je možné se připojit v rámci sítě (připojení na stejnou síť jako je připojené Raspberry Pi) v prohlížeči na stránku *<IP adresa zařízení>:8123*, kde je potřeba počkat, pokud stále probíhá instalace, poté si vytvořit účet a proklikat se uvítacími kroky.

3.2.3 Instalace platformy MagicMirror²

Před instalací platformy MagicMirror² je potřeba nainstalovat grafické rozhraní. Zvolil jsem jednoduché prostředí *blackbox* se správcem displejů *LightDM*. Dále je nutné nainstalovat další balíčky a upravit systém tak, aby bylo možné nespouštět všechny příkazy jako *root*.

Kroky nutné před instalací platformy:

1. Instalace balíčků potřebných pro grafické rozhraní, terminálu *xterm*, balíčků *xdotool* a *unclutter* pro pozdější konfiguraci kurzoru na obrazovce

```
$ apt install xserver-xorg xinit x11-xserver-utils blackbox xterm lightdm
xdotool unclutter
```

2. Nastavení časového pásma a zapnutí automatické synchronizace (aby se v aplikaci MagicMirror ukazoval správný čas)

```
$ timedatectl set-timezone Europe/Prague
$ systemctl start systemd-timesyncd
$ systemctl enable systemd-timesyncd
```

3. Vytvoření nového uživatele a nastavení hesla (po příkazu *\$ passwd mirror* je nutné zadat heslo)

```
$ useradd mirror
$ passwd mirror
```

4. Zkopírovat aktuální adresář *root* do domovského adresáře uživatele *mirror* a nastavit mu v domovském adresáři práva

```
$ cp -r /root /home/mirror
$ chown -R mirror:mirror /home/mirror
```

5. Upravit poslední řádek souboru */etc/passwd*, který začíná *mirror:x* tak, aby na konci místo *:/bin/sh* bylo *:/bin/bash* (výchozí shell bude *bash*); ukázka na obrázku 3.11

```
$ nano /etc/passwd
```




```
tmp:x:1001:1001:~/home/tmp:/bin/sh
/etc/passwd (END)
```

■ **Obrázek 3.11** Ukázka upravovaného řádku v souboru */etc/passwd* pro uživatele *tmp*

6. Instalace balíčku pro spouštění příkazů s vyššími právy, balíčku pro správu verzovacího systému *git*, interpretoru jazyka javascript – *nodejs* a správce balíčků pro node – *npm*

```
$ apt install sudo git nodejs npm
```

7. Upravit řádek v souboru */etc/group*, který začíná *sudo:x*, připsáním nakonec za dvojtečku *mirror* (aby uživatel *mirror* mohl spouštět příkazy jako *root*); ukázka na obrázku 3.12



```
sudo:x:27:mirror
```

■ **Obrázek 3.12** Ukázka řádku v souboru */etc/group* po upravení


```
# User privilege specification
root    ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo  ALL=(ALL:ALL) NOPASSWD: ALL

# See sudoers(5) for more information on "#include" directives:

#include /etc/sudoers.d

/etc/sudoers (END)
```

■ **Obrázek 3.13** Ukázka konce souboru `/etc/sudoers` po upravení

8. Upravit řádek v souboru `/etc/sudoers`, který začíná `%sudo`, před poslední `ALL` přidat `NOPASSWD:` (aby uživatel `mirror` nebyl při spuštění příkazů jako `root` dotazován na heslo); ukázka na obrázku 3.13
9. Upravit soubor `/etc/lightdm/lightdm.conf` (automatické přihlašování uživatele `mirror` do systému), na konec přidat řádky:

```
[SeatDefaults]
autologin-user=mirror
autologin-user-timeout=0
```

Na konci souboru `/etc/lightdm/lightdm.conf` by měl být text jako na obrázku 3.14

```
[SeatDefaults]
autologin-user=mirror
autologin-user-timeout=0
(END)
```

■ **Obrázek 3.14** Ukázka konce souboru `/etc/lightdm/lightdm.conf` po upravení

```
$ nano /etc/lightdm/lightdm.conf
```

10. Nastavení hostname (jak se zařízení bude jmenovat v síti), upravit soubor `/etc/hostname` – přepsat na `mirror`

```
$ nano /etc/hostname
```

11. Nastavení aliasu adresy `127.0.0.1` na `mirror`, upravit soubor `/etc/hosts` – připsat `127.0.0.1 mirror`

```
$ nano /etc/hosts
```

12. Restart

```
$ reboot
```

13. Instalace chromia (nezbytné pro MagicMirror²)

```
$ sudo apt install chromium
```

Tímto končí instalace prerekvizit a nyní je možné použít instalační skript pro platformu MagicMirror² na Raspberry. Následují kroky pro instalaci platformy:

1. Získání a uložení instalačního skriptu jako *magicmirror.sh*

```
$ curl -Lo magicmirror.sh https://raw.githubusercontent.com/sdetweil/MagicMirror_scripts/master/raspberry.sh
```

2. Instalace MagicMirror² spuštěním staženého instalačního skriptu pro Raspberry

```
$ bash magicmirror.sh
```

Nyní je platforma MagicMirror² nainstalována. Je možné aplikaci spustit, stačí být v adresáři MagicMirror (`$ cd ~/MagicMirror`) a spustit příkazem `$ DISPLAY=:0 npm start`. Nicméně je tam základní konfigurace a některé moduly nebudou fungovat, to je potřeba nastavit, a tím se budou zabývat následující sekce této kapitoly.

3.2.4 Instalace neuronové sítě Kalliope

Ačkoliv na Raspberry Pi běží Debian 10 (Buster), pro instalaci na Raspberry Pi bude využit skript pro Raspberry Pi OS (někdy také zvaný *Raspbian*). Skript provádí nastavení zvuku pro zařízení Raspberry Pi. Může se stát, že Kalliope po instalaci nebude fungovat, první, na co je nutné se zaměřit, je zvuk, o který se stará zvukový server *PulseAudio*. Pokud je zvuk v pořádku a stále Kalliope nefunguje, je potřeba projít znovu manuální instalaci, tentokrát ale pro Debian z oficiální dokumentace pro Kalliope.

Instalace Kalliope:

1. Získání a uložení instalačního skriptu jako *kalliope-install.sh*

```
$ curl -Lo kalliope-install.sh https://raw.githubusercontent.com/kalliope-project/kalliope/master/install/rpi_install_kalliope.sh
```

2. Instalace Kalliope spuštěním staženého instalačního skriptu pro Raspberry

```
$ bash kalliope-install.sh
```

3. Jelikož *Snowboy hotword detection engine* není v Kalliope nastaven pro architekturu aarch64 (pro Raspberry Pi 4 Model B), je potřeba vytvořit sdílený objekt, který se bude používat; nejprve stáhnout repozitář *Snowboy*

```
$ git clone https://github.com/Kitt-AI/snowboy.git
```

4. Po stažení je nutné se dostat adresáře *snowboy/swig/Python3*

```
$ cd snowboy/swig/Python3
```

5. Následně je důležité upravit *Makefile* tak, aby se vytvořil sdílený objekt pro architekturu aarch64; přepsat řádek:

SNOWBOYDETECTLIBFILE = \$(TOPDIR)/lib/ubuntu64/libsnowboy-detect.a tak, aby řádek odpovídal obrázku 3.15

```
SNOWBOYDETECTLIBFILE = $(TOPDIR)/lib/aarch64-ubuntu1604/libsnowboy-detect.a
```

- **Obrázek 3.15** Upravený řádek s architekturou aarch64 v *snowboy/swig/Python3*

```
$ nano Makefile
```

6. Spustit vytvoření nového sdíleného objektu *__snowboydetect.so*

```
$ make
```

7. Nyní je potřeba vytvořit adresářovou strukturu a do ní nakopírovat výše vytvořený soubor *__snowboydetect.so* tak, aby se nacházel na níže zmíněné cestě

```
/usr/local/lib/python3.7/dist-packages/kalliope-0.7.1-py3.7.egg/kalliope/trigger/snowboy/aarch64/python37/_snowboydetect.so
```

8. S velkou pravděpodobností bude problém se zvukem, proto je potřeba nastavit PulseAudio; v této části instalace jsou známé chyby se zvukem a dost často se do operačních systémů předinstalovávají zvukové systémy s nějakou konkrétní konfigurací, která je ve většině případů nefunkční; mně osobně pomohl následující příkaz, který přeinstaloval PulseAudio

```
$ sudo apt-get --purge --reinstall install pulseaudio
```

Nyní je Kalliope nainstalována a je možné spustit základní demo aplikaci. Je nutné být v adresáři *kalliope*, který je v adresáři, kam se nainstaloval *projekt kalliope* (`$ cd ~/kalliope-project/kalliope`). Stačí spustit příkaz `$ kalliope start`. Dalšímu nastavení se budou věnovat následující kapitoly.

3.3 Konfigurace

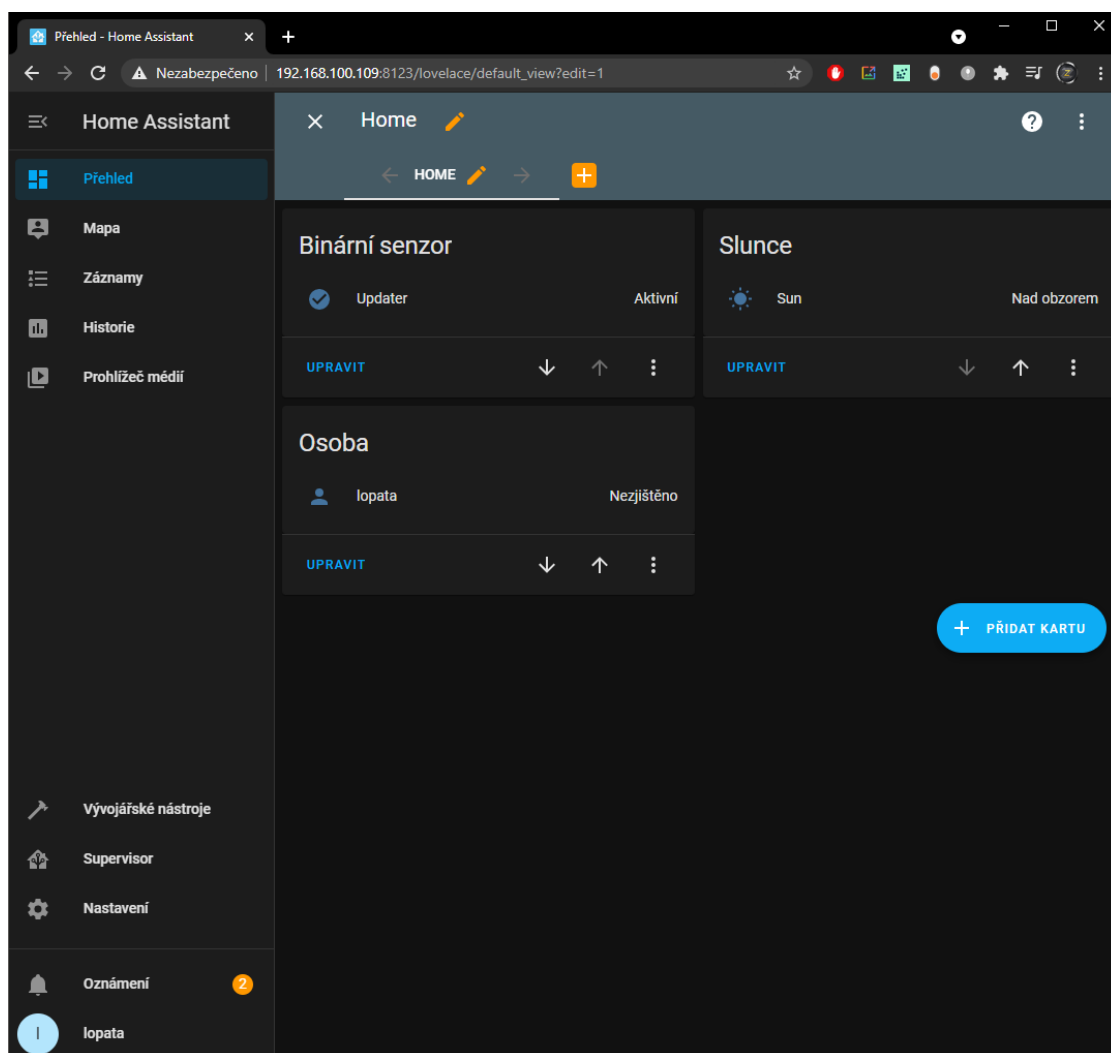
V této části bude popsáno, jakým způsobem nastavit platformy HomeAssistant, MagicMirror² a neuronovou síť Kalliope. Zároveň bude vidět, jak jsou platformy se sebou propojené.

3.3.1 Konfigurace HomeAssistant

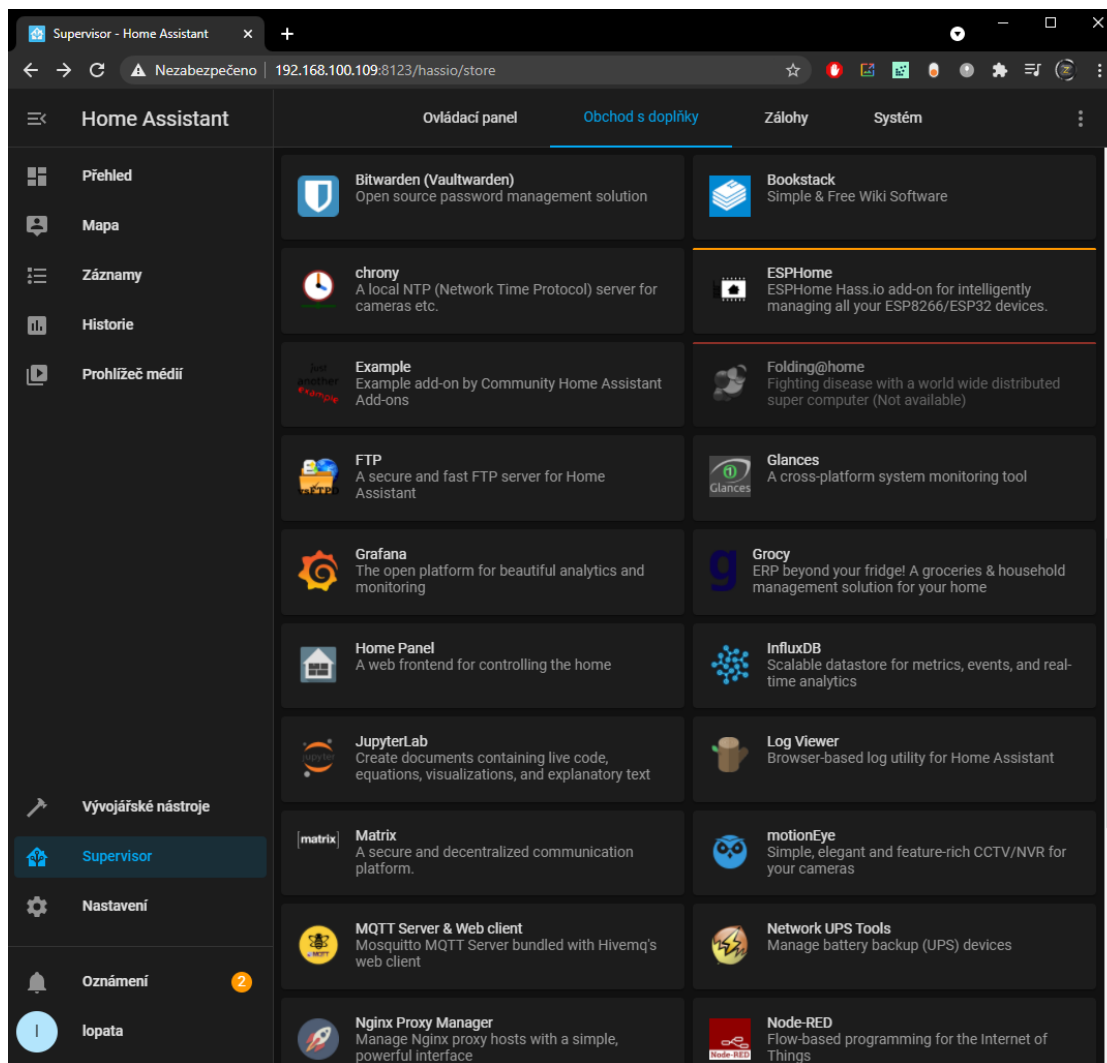
Po přihlášení do HomeAssistant ve webovém prohlížeči (<IP adresa zařízení>:8123), se zobrazí základní strana uživatelského rozhraní *Lovelace* (obrázek 3.16). Nyní je potřeba přidat platformu ESPHome. V levém menu je potřeba vybrat položku *Supervisor*, tím se otevře nabídka na kartě *Ovládací panel*. Dále je nutné na kartě *Obchod s doplňky* najít a kliknout na *ESPHome* (obrázek 3.17). Po otevření této „add-on“, je potřeba kliknout na *instalovat* a po dokončení instalace na *spustit*. Doporučuji ještě zapnout možnost *Zobrazit v postranním panelu*, tím se zjednoduší přístup k tomuto rozšíření (lze vidět na obrázku 3.18). Nyní je potřeba otevřít webové rozhraní pro *ESPHome* a nastavit první zařízení. Zobrazí se stránka podobná té na obrázku 3.19, ale nebude tam zařízení *test_node*. Vpravo nahoře je způsob nahrávání kódu do zařízení, výchozí je nastavení přes OTA (Over The Air), ale to funguje až po prvním nahrání kódu přes kabel. Vpravo dole je zelené kolečko *plus*, kterým se přidá zařízení. V dalším kroku se specifikují typ zařízení, jméno a heslo k síti apod. Po přidání zařízení je možné kód editovat viz obrázek 3.19. Po kliknutí na tlačítko *edit* se objeví kód pro zařízení. Pod tento kód je pak možné syntaxí z platformy ESPHome přidat vlastní kód. V mém případě kód pro desku NodeMCU v3 vypadá takto 3.20. Za placeholdery ve špičatých závorkách je nutné doplnit skutečné údaje. Nastavil jsem dva přepínače (switch), jeden pro červenou diodu na pinu *D1*, druhý pro zelenou diodu na pinu *D4* (obrázek zapojení 3.21). Po napsání kódu je potřeba ho nahrát do zařízení (upload). Po úspěšném nahrání kódu je nutné povolit, aby se nově přidané entity v ESPHome automaticky přidaly do HomeAssistant. Toho se docílí v *nastavení*, položka *Integrace*, políčko s *ESPHome*, tři tečky, vybrat *Více možností*, povolit a aktualizovat. Když jsou entity připravené, zbývá přidat ovládání do rozhraní *Lovelace*. V levém menu vybrat *Přehled* (viz obrázek 3.16), vpravo kliknout na tlačítko *Přidat kartu*, vybrat možnost *Entity*. Otevře se nastavení tohoto panelu (obrázek 3.22), kde je nutné vybrat konkrétní entity. V mém případě jsem přidal entity *GREEN LED* a *RED LED*. Po uložení vypadá základní rozhraní jako na obrázku 3.23 a je připravené ovládat LED diody připojené k vývojové desce NodeMCU v3. Pro propojení s platformou MagicMirror² je nutné si vytvořit token s dlouhou životností. Ten je možné vytvořit v profilu (v levém menu položka úplně dole, se jménem uživatele), v nabídce popojet co nejvíce dolů a token vytvořit, ideálně i někde uchovat (obrázek 3.24).

Tímto způsobem lze libovolně přidávat zařízení v ESPHome a ovládat je přes HomeAssistant. Zmíněné kódy jsou pouze ilustrativní, HomeAssistant je mnohem silnější nástroj, dovede pouštět skripty podle času, chovat se podle nastavených hodnot, hlídat hodnoty a mnoho dalšího, je jen na uživateli jak moc bude chtít svoji chytrou domácnost ovládat.

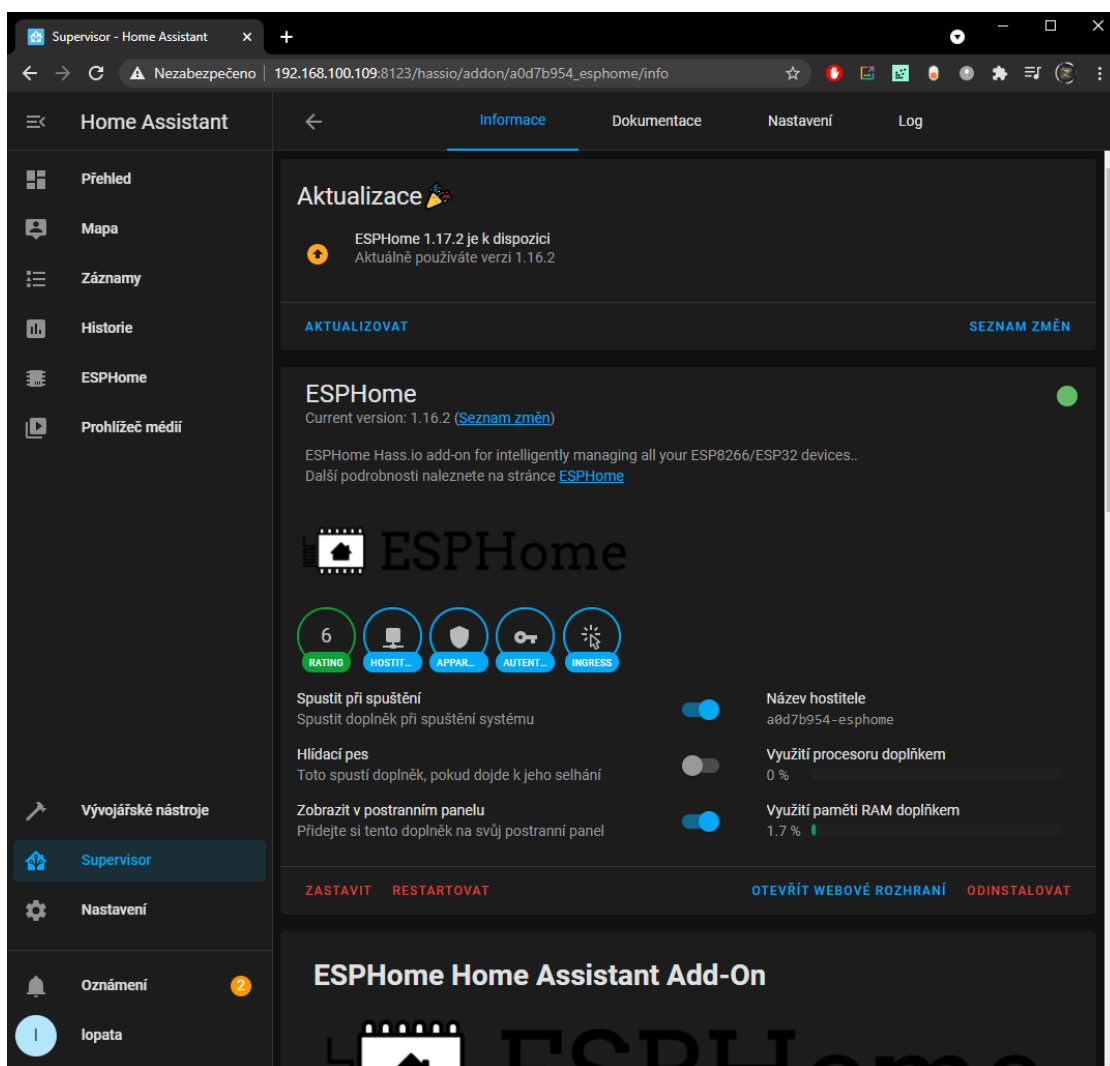
HomeAssistant má i svoji aplikaci pro mobilní telefony s operačním systémem Android i iOS. Do aplikace je nutné se přihlásit. Pokud aplikace běží lokálně, je nutné být telefonem připojený na stejnou síť. Rozhraní *Lovelace* zachovává definované objekty a dokonce je možné z telefonu konfigurovat zařízení v „add-on“ *ESPHome*. Jak aplikace na operačním systému Android vypadá je vidět na obrázku 3.25 a 3.26.



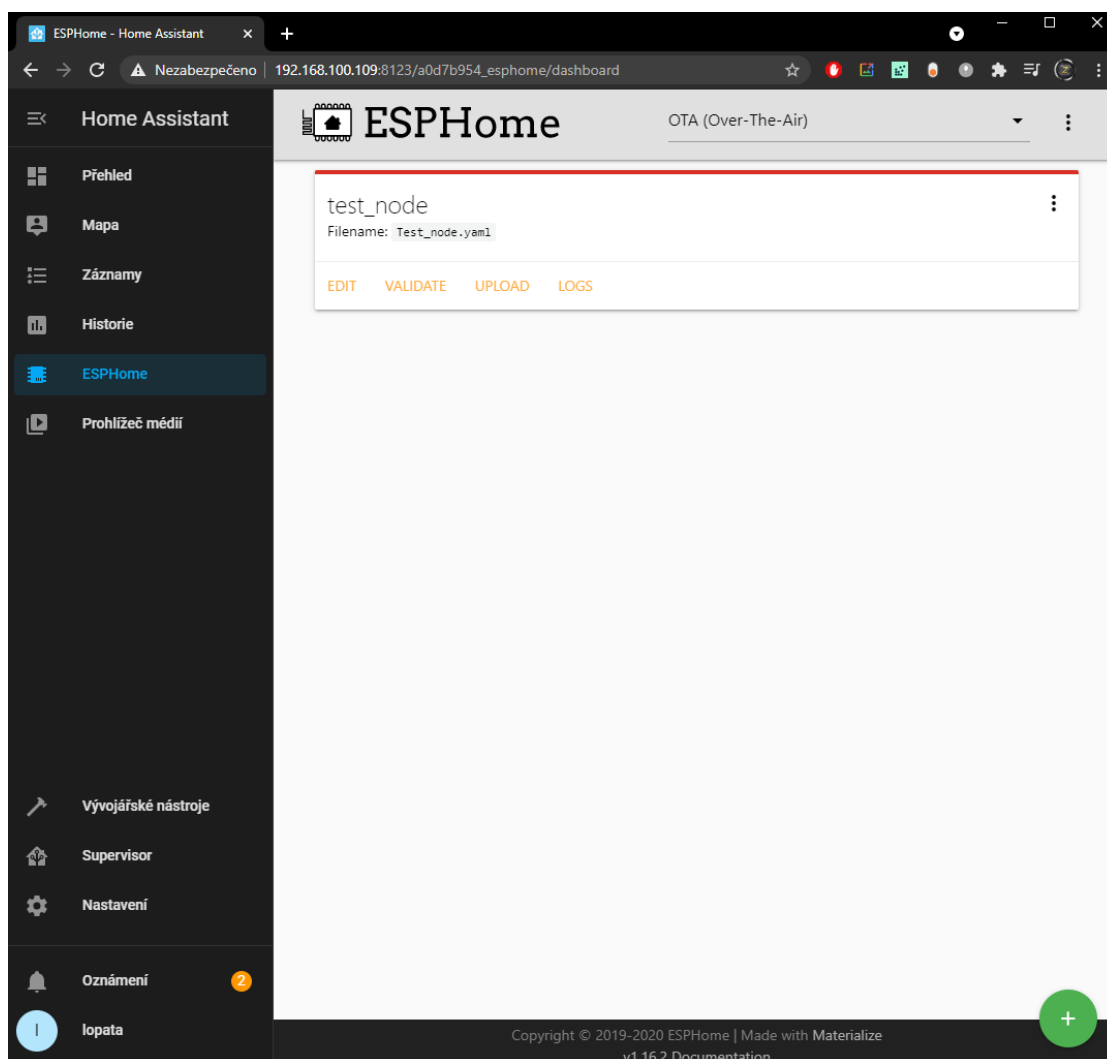
■ **Obrázek 3.16** Domovská stránka a uživatelské rozhraní *Lovelace* platformy HomeAssistant po instalaci



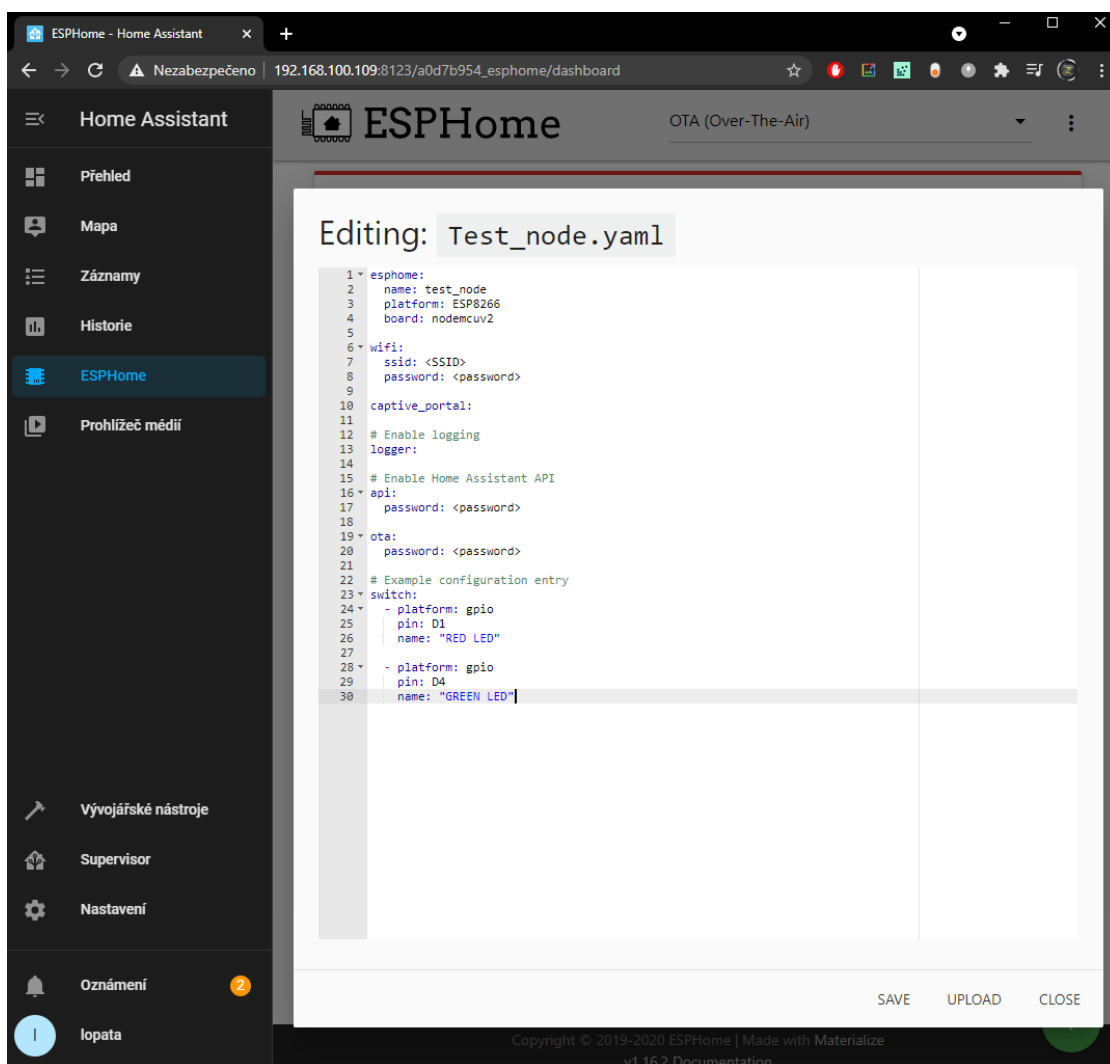
■ **Obrazek 3.17** Výpis integrovaných platform do HomeAssistant ve formě „add-ons“ na kartě *Supervisor*



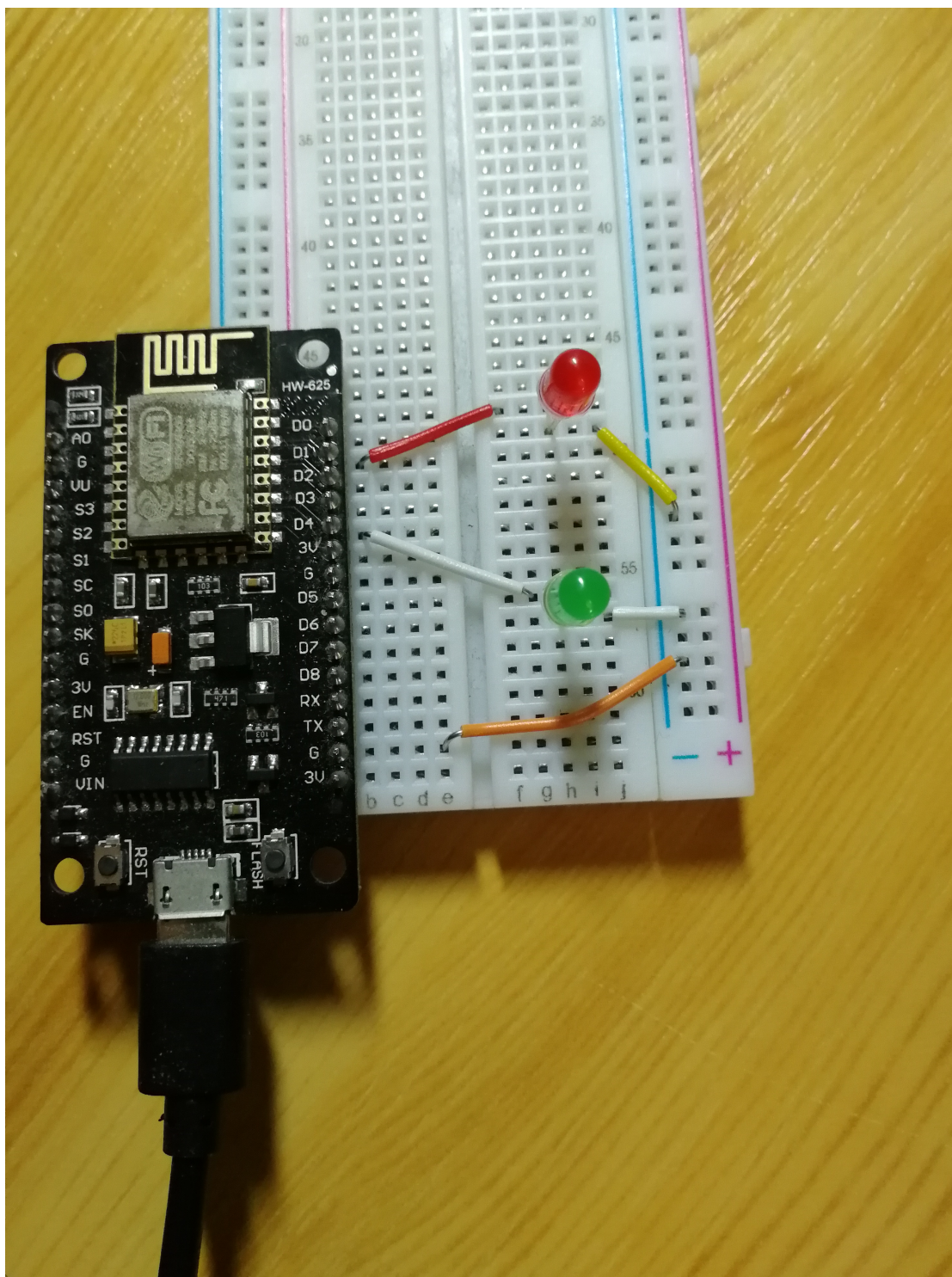
■ Obrázek 3.18 ESPHome „add-on“ pro přidání platformy ESPHome do HomeAssistant



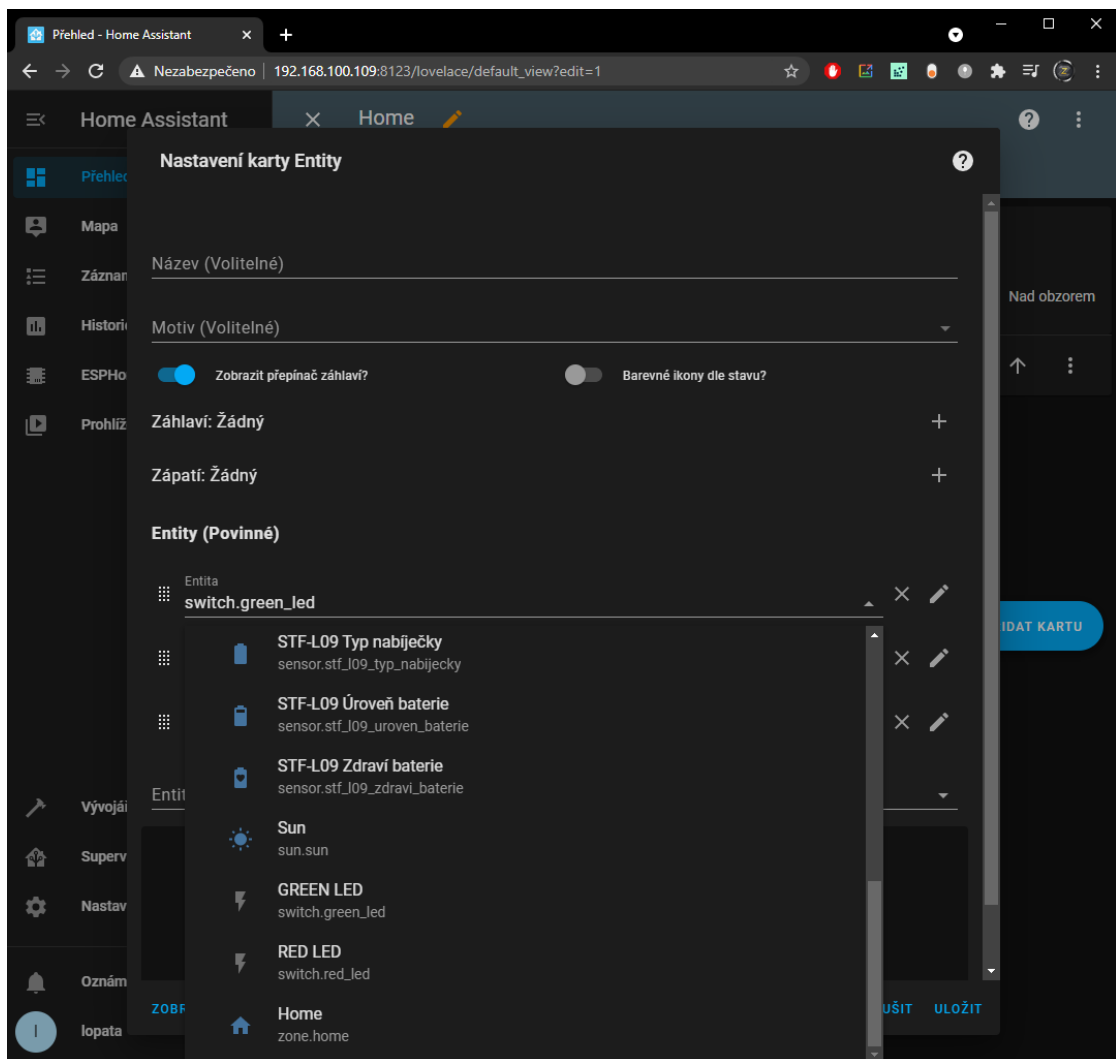
■ Obrázek 3.19 Základní stránka webové rozhraní „add-on“ ESPHome s jedním zařízením



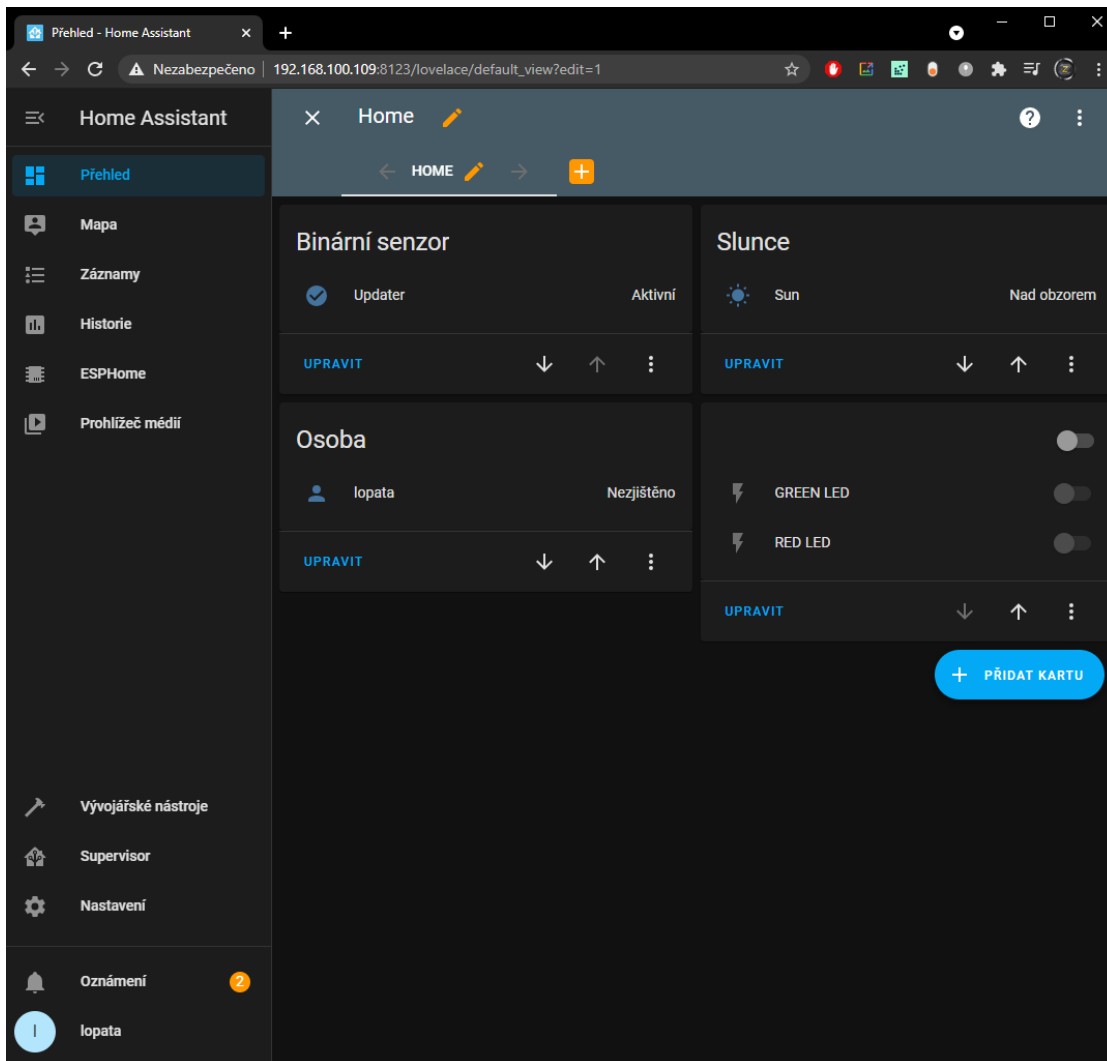
■ **Obrázek 3.20** Nástroj pro úpravu kódu zařízení platformy ESPHome; kód, nahraný do čipu ESP8266 na vývojové desce *NodeMCU v3*, který ovládá dvě LED diody – červenou na pinu D1 a zelenou na pinu D4



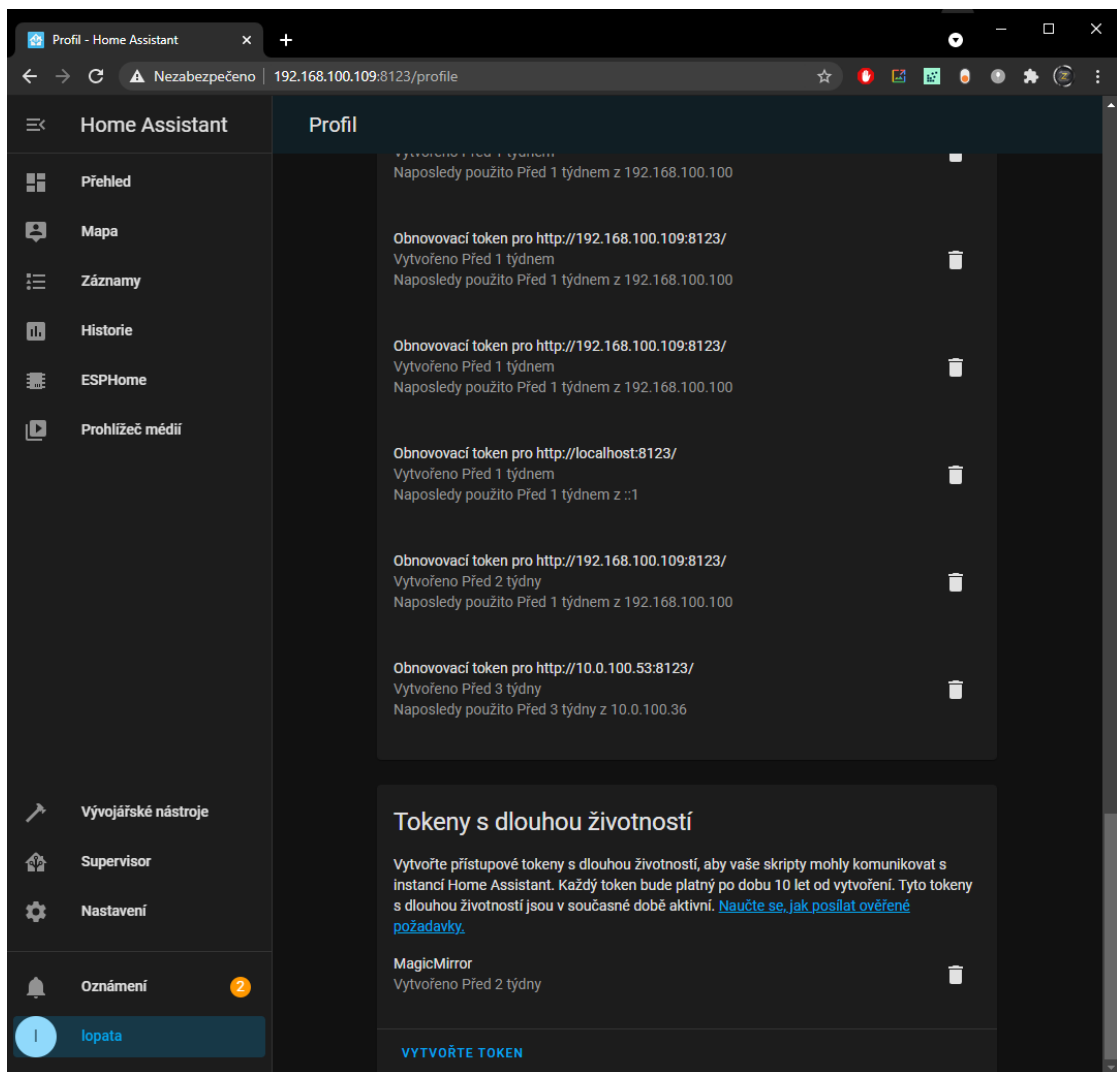
■ **Obrázek 3.21** Fotografie zapojení obvodu s vývojovou deskou *NodeMCU v3*, červená LED dioda je připojena k pinu D1 a zelená k pinu D4



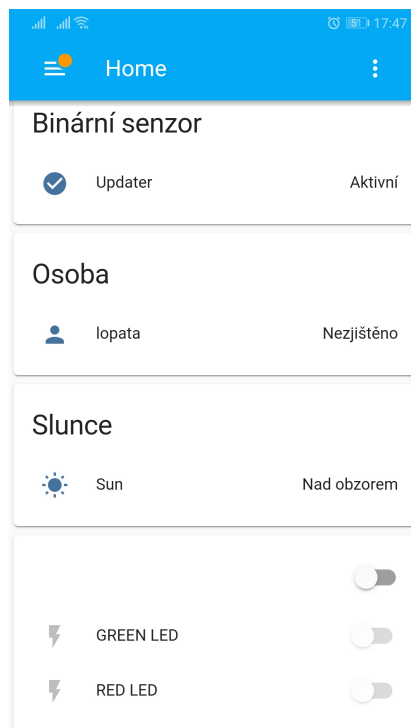
■ **Obrázek 3.22** Přidání nové karty s entitami v rozhraní *Lovelace*, nastavení konkrétních entit, které se mají zobrazit



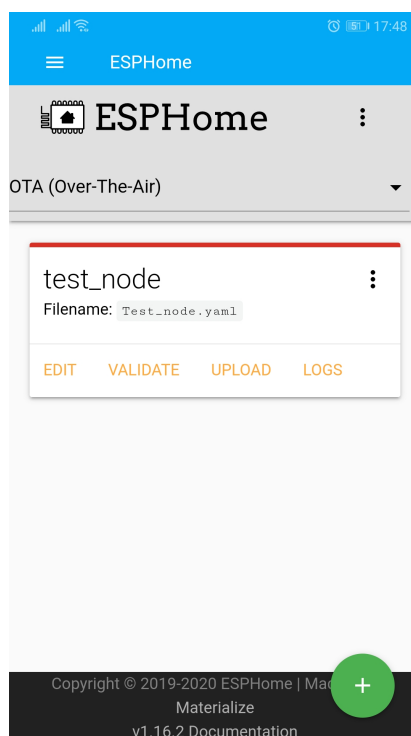
■ **Obrázek 3.23** Základní panel rozhraní *Lovelace* po přidání nové karty s entitami *GREEN LED* a *RED LED* pro ovládání zelené a červené diody připojené k vývojové desce NodeMCU v3



■ **Obrázek 3.24** Vytvoření autorizačního tokenu s dlouhou životností, potřebného pro propojení s MagicMirror², v HomeAssistant v nastavení profilu



■ **Obrázek 3.25** Uživatelské rozhraní *Lovelace* v mobilní aplikaci HomeAssistant s ovládáním LED diod



■ **Obrázek 3.26** Platforma ESPHome v mobilní aplikaci HomeAssistant

3.3.2 Konfigurace MagicMirror²

Platforma je založena na modulech třetí strany, které jsou vyvíjené komunitou. Autor platformy MagicMirror² tedy není za moduly zodpovědný a nemůže garantovat, že jsou moduly validní, ani že budou fungovat. Po nainstalování jsou dostupné pouze základní moduly platformy, zbytek je potřeba si doinstalovat. Většina modulů má v *readme* návod, jak je nainstalovat. Moduly se instalují do adresáře *MagicMirror/modules*. Když je modul nainstalovaný, je potřeba ho přidat do konfiguračního souboru, který je dostupný v *MagicMirror/config/config.js*.

Vybrané moduly:

- calendar – kalendář
- MMM-NameDay – zobrazení, kdo má dnes svátek
- MMM-HASS – propojení s platformou HomeAssistant
- MMM-OnThisDate – události z wikipedie, které se staly v tento den
- phone_notification – propojení s aplikací pro notifikace z mobilního telefonu *Pushbullet*
- MMM-AnyList – propojení s aplikací pro mobilní telefony *AnyList*
- MMM-EveryNews – zprávy z českých novin
- MMM-GmailFeed – propojení s gmail schránkou
- MMM-Instagram2020 – zobrazení fotek z osobního účtu aplikace Instagram
- MMM-NowPlayingOnSpotify – zobrazení, co právě hraje v hudební aplikaci Spotify
- MMM-pages – modul pro nastavení zobrazení do nezávislých stránek (vždy zobrazena pouze jedna strana)
- clock – hodiny
- MMM-3Day-Forecast – počasí na aktuální den a dva dny dopředu
- MMM-page-indicator – zobrazení na jaké straně se aktuálně chytře zrcadlo nachází
- MMM-kalliope – propojení s neuronovou sítí Kalliope

Všechny moduly lze dohledat na [51], existují k nim návody, jak je nainstalovat a zprovoznit. V práci budou dále zmíněny pouze *MMM-HASS* a *MMM-kalliope*, které jsou pro projekt stěžejní, zbytek je individuální a ne každý může chtít stejné moduly.

3.3.2.1 Úprava konfiguračního souboru

V konfiguračním souboru *MagicMirror/config/config.js* je možné si změnit jazyk na češtinu. Je potřeba najít řádky obsahující:

```
language: "en",
locale: "en-US",
```

a změnit je na:

```
language: "cs",
locale: "cs-CZ",
```

V konfiguračním souboru je taktéž seznam modulů, který je vypsán za "modules:". Pro propojení s Kalliope je potřeba stáhnout modul *MMM-kalliope* do *MagicMirror/modules*. Když je modul stažený, musí se zavést do konfiguračního souboru, tedy do seznamu s moduly přidat:

■ **Výpis kódu 3.1** Konfigurace modulu *MMM-kalliope* v *config.js*

```
{
  module: "MMM-kalliope",
  position: "top_center",
  config: {
    title: "Kalliope"
  }
},
```

Kalliope se bude zobrazovat nahoře uprostřed obrazovky a pokud se text objeví, bude mít nad ním uvedeno *Kalliope*, aby bylo jasné, odkud text pochází. Tímto je modul připravený, ale stále ještě nebude fungovat, protože je potřeba přidat do Kalliope neuron *MagicMirror*. Tomu se bude věnovat další sekce této kapitoly.

Modul *MMM-HASS* je nastaven, aby se na obrazovce zobrazoval vlevo nahoře. Jelikož běží lokálně, jako *host* byla použita adresa *127.0.0.1* s portem, na kterém běží HomeAssistant (8123). Pro autorizaci se využívá dříve vytvořený token a *https* se není nastaveno, tedy *false*. Stav zařízení se obnovuje každých 10 sekund. Samotná zařízení lze rozdělit do skupin (například místnosti v domácnosti), u kterých je potřeba nastavit seznam zařízení, které tam patří. Důležité je definovat senzor, v tomto případě červená a zelená LED dioda, jejichž *entity ID* lze zjistit v HomeAssistant. Parametr *icon* představuje ikonku. Já jsem použil jednoduchou ikonu žárovky a jako *suffix* jsem použil mé označení diod, aby je bylo možné rozeznat. Tato funkcionalita se ale lépe využije například u zobrazení teploty, kde je možné přidat například °C. Tento modul každých 10 sekund vyšle request na HomeAssistant API a zjistí stav definovaných senzorů. U přepínače (switche) se jedná o možnosti on/off (zapnuto/vypnuto).

■ **Výpis kódu 3.2** Konfigurace modulu *MMM-HASS* v *config.js*

```
{
  module: "MMM-HASS",
  position: "top_left",
  config: {
    host: "127.0.0.1",
    port: "8123",
    hassiotoken: true,
    https: false,
    updateInterval: 10 * 1000,
    devices: [
      { deviceLabel: "Exterior",
        deviceReadings: [
          { sensor: "switch.red_led",
            icon: "far fa-lightbulb",
            suffix: " červená,"},
          { sensor: "switch.green_led",
            icon: "far fa-lightbulb",
            suffix: " zelená"}
        ]
      }
    ]
  }
},
```

Konfigurace je nastavena, nyní je nutné upravit rozhraní tak, aby Kalliope mohla posílat notifikace do modulu, který když dostane tyto notifikace, pošle request na HomeAssistant API a provede zadanou operaci. V *MMM-HASS.js* je potřeba si definovat konkrétní notifikace. V této práci jsem zvolil notifikaci *HASS_RED* pro ovládání červené LED diody a *HASS_GREEN* pro ovládání zelené LED diody viz 3.3 (*HASS_1* je nevyužitý). V *node_helper.js* se pak definuje konkrétní request an HomeAssistant api. V 3.4 je uvedena funkce *socketNotificationReceived*, která má jako parametry *notification* – název notifikace a *payload* – data. Pro červenou diodu se zavolá funkce *sendHassEvent*, která si vyčte informace z *MagicMirror/config/config.js*, sestaví základ dotazované url, a podle HomeAssistant API pošle akci *turn_on/off* podle toho, jestli v parametru *payload* přijde hodnota *on/off*. V těle requestu je JSON obsahující *entity_id* červené diody. Obdobně to platí pro zelenou diodu.

■ **Výpis kódu 3.3** Úprava modulu *MMM-HASS* v *MMM-HASS.js*

```
notificationReceived(notification, payload) {
  if (notification === 'HASS_1') {
    this.sendSocketNotification('HASS_1', payload);
  } else if (notification === 'HASS_RED') {
    this.sendSocketNotification('HASS_RED', payload);
  } else if (notification === 'HASS_GREEN') {
    this.sendSocketNotification('HASS_GREEN', payload);
  }
},
```

■ **Výpis kódu 3.4** Úprava modulu *MMM-HASS* v *node_helper.js*

```
socketNotificationReceived: function(notification, payload) {
  if (notification === 'GETDATA') {
    var self = this;
    self.config = payload;
    var structuredData = {};
    var completed_devices = 0;
    this.getHassReadings(this.config, function(device) {
      completed_devices++;
      structuredData[device.label] = device;
      if (completed_devices == self.config.devices.length) {
        self.sendSocketNotification('DATARECEIVED', structuredData);
      }
    });
  } else if (notification === 'HASS_1') {
    var self = this;
    this.sendHassEvent(this.config, 'media_player', 'select_source', {
      'entity_id': 'media_player.menjador',
      'source': 'Tria asm'
    });
  } else if (notification === 'HASS_RED') {
    var self = this;
    this.sendHassEvent(this.config, 'switch', 'turn_' + payload, {
      'entity_id': 'switch.red_led'
    });
  } else if (notification === 'HASS_GREEN') {
    var self = this;
    this.sendHassEvent(this.config, 'switch', 'turn_' + payload, {
      'entity_id': 'switch.green_led'
    });
  }
}
```

3.3.2.2 Úpravy některých modulů

U některých modulů se stalo, že jejich základní nastavení nevyhovovalo a bylo potřeba udělat pár změn. U modulu *MMM-OnThisDate* chyběla čeština, nicméně modul se zdál být zajímavý a funkční. Rozhodl jsem se proto češtinu přidat. Do souboru *MagicMirror/modules/MMM-OnThisDay/WikiCssSelectors.js* jsem přidal text 3.5. Dále je ještě potřeba v hlavním souboru modulu *MagicMirror/modules/MMM-OnThisDay/MMM-OnThisDate.js* změnit jazyk na češtinu – `usedLanguage: 'cs', // Fallback`. Pro tento zásah bylo potřeba modul prozkoumat, podívat se odkud bere data a upravit tak, aby se používala česká wikipedie [52] a správné názvy HTML tříd. Data z wikipedie jsou na obrázku 3.27, odpovídající html kód na 3.28.

■ **Výpis kódu 3.5** Přidání češtiny do modulu *MMM-OnThisDay* v souboru *WikiCssSelectors.js*

```
cs: {
  title: '.mainpage-block.calendar-container .mainpage-headline',
  events: '.mainpage-block.calendar-container .mainpage-content > ul'
},
```

11. květen v minulosti

- 1871 – Zemřel astronom, přírodovědec a průkopník fotografie **John Herschel** (na obrázku).
- 1891 – **Carevič Mikuláš** byl během své návštěvy **Japonska** zraněn při **atentátu**, který na něj spáchal japonský policista. Zachránil ho synovec **princ Jiří**, který odrazil útok svojí vycházkovou holí.
- 1911 – Byl založen fotbalový klub **Valur Reykjavík**.
- 1981 – V Londýně měl premiéru muzikál **Kočky**.
- 1996 – **Let ValuJet 592** se deset minut po startu zřítíl do *Everglades* v důsledku požáru zapříčiněného špatně uloženým nebezpečným nákladem. Všech 110 osob na palubě zahynulo.
- 2001 – Zemřel spisovatel **Douglas Adams**, autor kultovního románu *Stopařův průvodce po Galaxii*.



Další výročí...

■ **Obrázek 3.27** Ukázka sekce událostí na wikipedii, které se staly v aktuální den[52]

```
191 <div class="mainpage-headline"><a href="/wiki/11._kv%C4%9Bten" title="11. květen">11. květen v minulosti<
192 <div class="mainpage-content"><span class="picture" style="margin-right: 1em; margin-left: 1em; float: ri
193 <ul><li><a href="/wiki/1871" title="1871">1871</a> - Zemřel astronom, přírodovědec a&#160;průkopník fotog
194 <li><a href="/wiki/1891" title="1891">1891</a> - <a href="/wiki/Mikul%C3%A1%C5%A1_II._Alexandrovi%C4%8D"
195 <li><a href="/wiki/1911" title="1911">1911</a> - Byl založen fotbalový klub <b><a href="/wiki/Valur" titl
196 <li><a href="/wiki/1981" title="1981">1981</a> - V&#160;<a href="/wiki/Lond%C3%BDn" title="Londýn">Londýn
197 <li><a href="/wiki/1996" title="1996">1996</a> - <b><a href="/w/index.php?title=Let_ValuJet_592&amp;actio
198 <li><a href="/wiki/2001" title="2001">2001</a> - Zemřel spisovatel <b><a href="/wiki/Douglas_Adams" title
199 </div>
```

■ **Obrázek 3.28** Ukázka zdrojového kódu HTML sekce událostí na wikipedii, které se staly v aktuální den, ze kterého bylo potřeba získat data pomocí CSS selektorů [52]

Aby správně fungoval modul pro počasí, který využívá překladů do jazyka definovaného konfiguračním souboru, je nutné přidat překlad pro slovo *pozítří*, které je v modulu *MMM-3Day-Forecast* definováno jako *DAYAFTER*. Ovšem překlady platformy MagicMirror² obsahují překlad pouze pro variantu *DAYAFTERTOMORROW*. Pro správné zobrazení stačí přidat překlad "DAYAFTER": "Pozítří" do *MagicMirror/translations/cs.json*.

3.3.3 Konfigurace Kalliope

Jako výchozí projekt pro Kalliope jsem zvolil *starter kit* v češtině, který je dostupný na stránkách [53]. Zmíněný projekt má již přednastavené klíčové slovo „kalliope“ a nějaké základní funkce, například dokáže říct „Ahoj“, nebo odpovědět na otázku „kolik je hodin“.

Aby bylo možné propojit Kalliope s MagicMirror², je nutné nainstalovat neuron *MagicMirror*. Pro to je nutné být v adresáři s projektem (`$ cd kalliope_starter_cs`), odkud se instalace provede příkazem:

```
$ kalliope install --git-url
https://github.com/kalliope-project/kalliope_neuron_magic_mirror.git.
```

Po úspěšné instalaci zbývá napsat vlastní mozek. Kalliope vyhledá hlavní *brain.yml* uložený v projektu, ve kterém je možné vložit další mozky a držet tak rozumné strukturování. Pro ovládání chytrého zrcadla jsem vytvořil mozek *mirror.yml* v adresáři *brains* a vložil ho do hlavního *brain.yml* viz 3.6.

Samotný *mirror.yml* obsahuje příkazy pro ovládání chytrého zrcadla. V 3.7 je vidět struktura mozku. Nejprve se definuje jméno synapse, v tomto případě *mm-zapnout-cervenou-diodu*, pak následuje výpis signálů, které synapsi dokáží spustit. Typ signálu *order* je textová podoba zvuku, který následuje po rozpoznání klíčového slova „kalliope“. Parametr *matching-type* určuje, jak přísně bude vyžadována shoda textu. Mnou zvolený *ordered-strict* je vhodná volba, pokud se někde jinde vyskytuje jiná synapse s podobnými slovy. Zabrání se tak tomu, že se na nějaká konkrétní slova spustí více synapsí, protože je nutné aby odpovídala všechna slova v parametru *text* v pořadí zleva doprava a žádné další slovo nebylo řečeno. Následují neurony, které se mají touto synapsí spustit. V tomto případě se spustí neuron *MagicMirror*, který má jako parametry *mm_url* (adresa zrcadla, kde poslouchá modul *MMM-Kalliope*), *notification* (název notifikace) a *payload* (data). Pro zapnutí červené diody se odesílá notifikace „HASS_RED“ a payload „on“.

Kalliope má vlastní paměť. Toho využívá například synapse *mm-say* (3.8, která slouží k zobrazení poslední „text-to-speech“ zprávy od Kalliope na zrcadlo.

Další, co je možné ovládat hlasem je modul *MMM-pages*, které mají definované rozhraní a fungují na základě notifikací. Ukázka takového ovládání, které řídí přepnutí na třetí stránku je vidět na 3.10 (číslo 3 odpovídá v číslování stránek číslu 2, protože se stránky číslovají od nuly, ale to pro ovládání v běžném životě není praktické). Notifikace *PAGE_CHANGED* odpovídá přepnutí na stranu definovanou v parametru *payload*.

Do této práce jsem zakomponoval i LED pásek, jako přisvětlení, které u zrcadel často bývá. Pro ovládání tohoto světla jsem do Kalliope přidal neuron *gpio*, který jsem ale musel upravit, protože na operačním systému Debian 10 (Buster) nefunguje. S nastavením a ovládáním GPIO portů na Raspberry Pi jsou známé oficiální chyby a jejich oprava zatím není ve stabilní verzi. Instalace neuronu se spouští z adresáře *kalliope_starter_cs* příkazem:

```
$ kalliope install --git-url https://github.com/corus87/gpio-neuron
```

Poté je potřeba upravit neuron (soubor *kalliope_starter_cs/resources/neurons/gpio/gpio.py*) tak, aby byl funkční. To znamená skoro všechno z něj odebrat a zachovat pouze kostru. Výsledný kód neuronu je na 3.11. Synapse, která se stará o zapnutí světla je na 3.9. Komunikace mezi neuronem a aplikací bude zmíněna níže.

■ **Výpis kódu 3.6** Začátek souboru *brain.yml*, s „include“ dalších mozků, v adresáři projektu *kalliope_starter_cs*

```
---
- includes:
  - brains/say.yml
  - brains/systemdate.yml
  - brains/neurotransmitter.yml
  - brains/uri.yml
  - brains/mirror.yml
```

■ **Výpis kódu 3.7** Synapse na odeslání notifikace pro zapnutí červené diody v *brains/mirror.yml* v adresáři projektu *kalliope_starter_cs* (bez diakritiky)

```
- name: "mm-zapnout-cervenou-diodu"
  signals:
    - order:
      text: "Zapni červenou ledku"
      matching-type: "ordered-strict"
    - order:
      text: "Zapnout červenou ledku"
      matching-type: "ordered-strict"
    - order:
      text: "Zapni červený světlo"
      matching-type: "ordered-strict"
    - order:
      text: "Zapnout červený světlo"
      matching-type: "ordered-strict"
    - order:
      text: "Zapni červené světlo"
      matching-type: "ordered-strict"
    - order:
      text: "Zapnout červené světlo"
      matching-type: "ordered-strict"
  neurons:
    - magic_mirror:
      mm_url: "http://127.0.0.1:8081/kalliope"
      notification: "HASS_RED"
      payload: "on"
```

■ **Výpis kódu 3.8** Synapse na odeslání poslední „Text-To-Speech“ zprávy, aby se zobrazila na zrcadle

```
- name: "mm-say"
  signals: []
  neurons:
    - magic_mirror:
      mm_url: "http://127.0.0.1:8081/kalliope"
      notification: "KALLIOPE"
      payload: "{{ kalliope_memory['kalliope_last_tts_message'] }}"
```

■ **Výpis kódu 3.9** Synapse odpovědná za zapnutí přísvětlovacího světla u zrcadla (bez diakritiky)

```
- name: "zapnout-svetlo"
  signals:
    - order:
      text: "Zapni světlo"
      matching-type: "ordered-strict"
    - order:
      text: "Zapnout světlo"
      matching-type: "ordered-strict"
    - order:
      text: "Rozsvit světlo"
      matching-type: "ordered-strict"
    - order:
      text: "Rozsvitit světlo"
      matching-type: "ordered-strict"
    - order:
      text: "Rozni světlo"
      matching-type: "ordered-strict"
    - order:
      text: "Roznout světlo"
      matching-type: "ordered-strict"
  neurons:
    - gpio:
      set_pin_high:
        - 7
```


■ Výpis kódu 3.10 Synapse na odeslání notifikace pro přepnutí zobrazení na stránku 3 (bez diakritiky)

```
- name: "mm-treti-strana"
  signals:
    - order:
      text: "Strana 3"
      matching-type: "ordered-strict"
    - order:
      text: "Strana cislo 3"
      matching-type: "ordered-strict"
    - order:
      text: "3. strana"
      matching-type: "ordered-strict"
    - order:
      text: "Stranka 3"
      matching-type: "ordered-strict"
    - order:
      text: "Stranka cislo 3"
      matching-type: "ordered-strict"
    - order:
      text: "3. stranka"
      matching-type: "ordered-strict"
    - order:
      text: "Zobraz 3. stranku"
      matching-type: "ordered-strict"
    - order:
      text: "Zobraz 3. stranu"
      matching-type: "ordered-strict"
    - order:
      text: "Zobrazit 3. stranku"
      matching-type: "ordered-strict"
    - order:
      text: "Zobrazit 3. stranu"
      matching-type: "ordered-strict"
  neurons:
    - magic_mirror:
      mm_url: "http://127.0.0.1:8081/kalliope"
      notification: "PAGE_CHANGED"
      payload: 2
```

■ Výpis kódu 3.11 Neuron Kalliope pro ovládání GPIO portů na Raspberry Pi

```
# -*- coding: iso-8859-1 -*-
import logging
from kalliope.core.NeuronModule import NeuronModule
import socket

logging.basicConfig()
logger = logging.getLogger("kalliope")

HOST = '127.0.0.1'
PORT = 65432

class Gpio(NeuronModule):
    def __init__(self, **kwargs):
        super(Gpio, self).__init__(**kwargs)
        # the args from the neuron configuration
        self.set_pin_high = kwargs.get('set_pin_high', None)
        self.set_pin_low = kwargs.get('set_pin_low', None)
        self.sensor = kwargs.get('sensor', None)
        self.fahrenheit = kwargs.get('fahrenheit', False)
        self.one_decimal_place = kwargs.get('one_decimal_place', False)

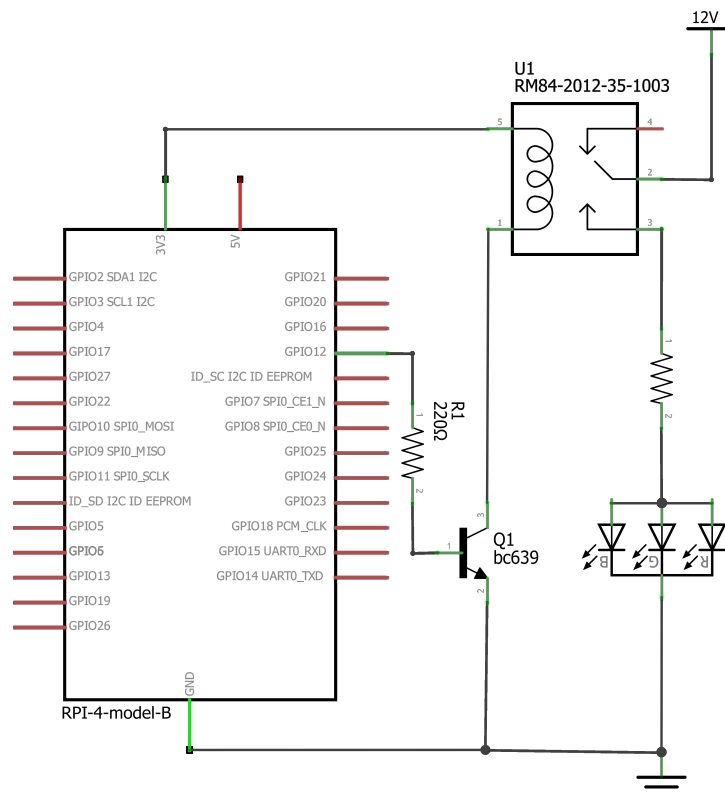
        if self.set_pin_high:
            logger.debug('[GPIO] Set pin %s to high' % self.set_pin_high)
            with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
                s.connect((HOST, PORT))
                s.sendall(b'on')

        if self.set_pin_low:
            logger.debug('[GPIO] Set pin %s to low' % self.set_pin_low)
            with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
                s.connect((HOST, PORT))
                s.sendall(b'off')
```

3.3.4 Aplikace na ovládání světla a samo-spouštěcí skript

Pro ovládání světla na portu GPIO jsem zvolil aplikaci v jazyce Python. Před spuštěním aplikace je nutné nainstalovat knihovnu pro práci s GPIO porty na Raspberry Pi *gpiozero*. Instalace probíhá příkazem `$ pip3 install gpiozero`. V kódu se nastavuje port GPIO 12 (podle číslování GPIO/BCM) jako LED, aby bylo možné volat funkce *on* a *off*. Aplikace funguje na principu tzv. „socketů“ a je nastavena tak, že poslouchá na adrese *127.0.0.1* a na portu *65432*. Ve chvíli, kdy přijde zpráva z neuronu GPIO, který se na tuto adresu a port připojí a pošle zprávu *on/off*, provede aplikace příslušnou operaci a světlo zapne, eventuálně vypne. Výsledný kód aplikace je na 3.13.

Pro zapojení obvodu, který se stará o ovládání LED pásky bylo potřeba přepínací relé ovládané 3,3 V. Pořídil jsem relé *RM84-2012-35-1003*, které má odpor cívky $22\ \Omega$. Z Ohmova zákona $I = \frac{U}{R}$, jsem vypočítal jaký proud je potřeba k přepnutí relé. Po dosazení do vzorečku ($\frac{3,3\text{ V}}{22\ \Omega} = 0,15\text{ A} = 150\text{ mA}$) vyjde, že k přepnutí relé je potřeba 150 mA. Takový proud ale Raspberry Pi na samotném GPIO pinu nemá. Piny jsou omezené na maximálně 16 mA. Proto je potřeba využít spínací tranzistor a ovládat jím napětí z napájecího pinu 3,3 V. Ten dokáže poskytnout až 50 mA, což stále není 150 mA, ale testováním bylo zjištěno, že to již k sepnutí relé stačí. Schéma zapojení je vidět na obrázku 3.29, kde LED diody představují LED pásek.



■ **Obrázek 3.29** Schéma zapojení obvodu připojeného k Raspberry Pi pro ovládání LED pásky

Aby se po zapnutí Raspberry Pi spustila aplikace MagicMirror, je nutné vytvořit samo-spouštěcí skript. Návod je převzatý z [54]. Toho se docílí pomocí správce procesů pro Javascript runtime *PM2*. Postup pro nastavení tohoto skriptu je následující:

1. Instalace správce procesů pro Javascript runtime *PM2*

```
$ npm install pm2
```

2. Nastavení, aby se skript spustil automaticky po startu

```
$ pm2 startup
```

3. Vytvoření skriptu v domovském adresáři

```
$ cd ~  
$ nano mm.sh
```

4. Kód skriptu dostupý na 3.12; na začátku se vypínají procesy, které odpovídají spuštěné Kalliope a aplikace pro ovládání světla, poté se smažou logy, do proměnné *HASSIO_TOKEN* se uloží dříve vygenerovaný token z HomeAssistant, spustí se aplikace pro ovládání světla, poté Kalliope, nastaví se číslo displeje, kam se má zobrazit MagicMirror, posune se kurzor myši do levého horního rohu a nastaví se, aby po pěti vteřinách nečinnosti zmizel, nakonec se spustí MagicMirror

■ **Výpis kódu 3.12** Samo-spouštěcí skript pro automatické spuštění zrcadla

```
to_kill=$(sudo lsof -t -i:5000)  
[ -z "$to_kill" ] || sudo kill -9 $to_kill  
to_kill=$(sudo lsof -t -i:65432)  
[ -z "$to_kill" ] || sudo kill -9 $to_kill  
  
# uncomment to have fresh logs  
sudo rm -f /var/log/mirror/*  
  
export HASSIO_TOKEN=<token generated from HomeAssistant>  
  
sudo python3 led_blink.py > /var/log/mirror/led_blink.log 2>&1 &  
cd ~/kalliope_starter_cs  
kalliope start > /var/log/mirror/kalliope.log 2>&1 &  
cd ~/MagicMirror  
export DISPLAY=:0  
xdotool mousemove 0 0  
unclutter -idle 5 -root &  
  
npm start > /var/log/mirror/mirror.log 2>&1
```

5. Udělat skript spustitelný

```
$ chmod +x mm.sh
```

6. Puštění a uložení skriptu

```
$ pm2 start mm.sh  
$ pm2 save
```

7. Restartovat zařízení (po následujícím spuštění již bude skript fungovat)

```
$ sudo reboot
```

■ Výpis kódu 3.13 Aplikace pro ovládání přisvětlovacího světla u zrcadla

```
from gpiozero import LED  
import socket  
  
HOST = '127.0.0.1'  
PORT = 65432  
  
red = LED(12)  
  
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:  
    s.bind((HOST, PORT))  
    s.listen()  
    while True:  
        conn, addr = s.accept()  
        while True:  
            data = conn.recv(1024)  
            if not data:  
                break  
            if data == b'on':  
                red.on()  
            elif data == b'off':  
                red.off()  
            print(data)
```

3.4 Instalační příručka

Následující kroky jsou důležité pro nastavení chytrého zrcadla:

1. Nahrát image, který je součástí přiloženého média, na SD kartu a vložit ji do Raspberry Pi 4 Model B
2. Připojit klávesnici a myš
3. Po zapnutí zařízení se spustí MagicMirror, aplikaci je nutné minimalizovat stiskem kombinace kláves „CTRL + M“
4. Pravým tlačítkem myši kliknout na plochu a spustit terminál *xterm*
5. Příkazem `$ nano mm.sh` upravit samo-spouštěcí skript, ve kterém je potřeba všechno zakomentovat
6. Otevřít MagicMirro – kliknout kolečkem myši na plochu, vybrat *Icons* a *MagicMirror*
7. Zavřít aplikaci MagicMirror stiskem kombinace kláves „CTRL + Q“
8. V terminálu *xterm* pomocí *nmcli* nastavit připojení k Wi-Fi
9. Upravit konfigurační soubor *MagicMirror/config/config.js*, aby odpovídala konfigurace modulů dle jejich specifikace a spustit MagicMirror ze stejnojmenného adresáře příkazem:
`$ npm start`
10. Libovolně měnit konfiguraci, vypínat a zapínat aplikaci MagicMirror, přidávat moduly apod.
11. Nakonec příkazem `$ nano mm.sh` upravit samo-spouštěcí skript, ve kterém je potřeba všechno odkomentovat
12. Restartovat zařízení příkazem `$ sudo reboot`

Nyní je zařízení nakonfigurované a připravené k používání. Pokud si chce uživatel debuggovat, je možné MagicMirror v průběhu nespouštět příkazy `$ npm start`, ale `$ npm start dev`. Otevře se webová konzole, kde je možné pozorovat logy.

Testování, rozšiřitelnost a náklady

Tato kapitola se zabývá testováním vytvořeného prototypu chytrého zrcadla, možnou budoucí rozšiřitelností a náklady, které tato práce stála. Testování je rozděleno do dílčích částí, ve kterých se zkoumá čitelnost zobrazovaných informací na skle zrcadla za různých světelných podmínek, spolehlivost rozpoznání klíčového slova nezbytného pro hlasové ovládání a spolehlivost zobrazovaných modulů platformy MagicMirror². Nakonec bude prototyp otestován v běžném denním provozu jako celek.

4.1 Testování

V této části je popsáno, jak zrcadlo funguje jako celek, jak dobře jsou informace čitelné za různých světelných podmínek, jak spolehlivé je rozpoznání klíčového slova a spolehlivost zobrazovaných modulů.

4.1.1 Světelné podmínky

Zrcadlo bylo testováno v následujících podmínkách:

- Venku svítilo sluncem a byly zatažené žaluzie
- Venku byla tma a svítilo se v pokoji světlem ze stropu
- Venku bylo šero
- Venku byla tma a svítilo se světlem směrem ke stropu

Test byl proveden spuštěním zrcadla a pozorováním, jak jsou jednotlivé informace dobře čitelné. Zdaleka nejhorší možná situace byla, pokud venku svítilo slunce. I přesto, že byly zatažené žaluzie, bylo v místnosti tolik světla, že zobrazované informace téměř nebyly vidět. Šero a tma s rozsvíceným světlem podaly stejné výsledky – informace byly dobře čitelné. Na základě provedených testů jsem přišel na to, že se zrcadlo nejvíce hodí do místností, kde je šero. Zpravidla garáž, nebo koupelna (místnosti, kde není okno, nebo je dostatečně malé, aby jím neprošlo tolik světla zvenku). Pro ideální použití zrcadla bych doporučil tmavé pozadí, aby informace zobrazené na monitoru bílou barvou byly v kontrastu s odrazem, a tedy byly lépe čitelné. Pro lepší viditelnost jsem zvětšil velikost písma zobrazovaného na zrcadle (v souboru *MagicMirror/css/main.css*).

4.1.2 Rozpoznání klíčového slova

Pro testování spolehlivosti klíčového slova jsem desetkrát za sebou řekl z různých částí místnosti klíčové slovo a po jeho rozpoznání se zeptal „kolik je hodin?“. V osmi případech bylo klíčové slovo rozpoznáno a na následující příkaz jsem vždy dostal odpověď. V rámci rozpoznávání klíčového slova jsem nechal Kalliope zapnutou a při online videohovoru s vedoucím práce jsem pozoroval, že se samovolně spustí i když klíčové slovo nezazní. To je pravděpodobně způsobeno tím, že engine na rozpoznání klíčového slova je vytrénovaný na sedmi vzorcích slova „Kalliopé“ s hlasem někoho jiného. Tento problém má řešení, a to vytrénovat si nová data pro rozpoznání klíčového slova na více vzorcích. To není předmětem této práce, a tak je potřeba postup vyhledat v dokumentaci enginu *Snowboy*. Výsledná spolehlivost rozpoznání klíčového slova v této práci se pohybuje okolo 80 %.

4.1.3 Zobrazované moduly

Po správném nastavení všech použitých modulů jsem testoval, jestli opravdu dělají to, k čemu jsou určeny. Zaměřil jsem se na propojení oznámení z mobilního telefonu pomocí *Pushbullet*. Z jiného zařízení jsem na propojený telefon posílal zprávy z aplikace *Messenger*, posílal email a zavolaal. Ne vždy byly notifikace na zrcadle vidět. Hovor se zobrazil vždy, ale zprávy z aplikace *Messenger* a emaily nikoliv. Tento problém byl nejspíše způsoben neaktivitou aplikace *Pushbullet* v mobilním telefonu, nikoliv modulem samotným. Nicméně výsledek tohoto selhání se projeví tak, že modul zrcadla nezobrazuje notifikace vždy správně. Problém se objevoval nahodile. Další modul, který nefungoval tak jak by měl byl *MMM-Instagram2020*, který dokáže zobrazovat pouze fotky. Toto je ale problém samotného modulu, který videa nepodporuje, přesto se snaží je zobrazit a výsledkem je nefunkční zobrazení na zrcadle. V průběhu testování došlo k poruše modulu poskytujícího informace o počasí. Chyba byla nejspíše na straně poskytovatele dat, protože modul se zničeho nic zasekl na hláске „načítání“, druhý den po zapnutí zrcadla se stejnou konfigurací opět fungoval. Zbytek modulů funguje spolehlivě a problémy nastávají pouze pokud je nefunkční internetové připojení, když dochází k pravidelné obnově dat.

4.1.4 Systém jako celek

Celé zrcadlo jsem otestoval pro reálné použití – řekl klíčové slovo a vyzkoušel více signálů pro spuštění konkrétních synapsí. Při testování se stalo, že jsem jedním příkazem („zapni červené světlo“) nechtěně spustil dvě různé synapse – zapnutí světla na zrcadle a zapnutí červené diody připojené k desce NodeMCU v3. Tento problém jsem vyřešil nastavením typu shody příkazu v mozku Kalliope tak, aby příkaz spustila slova, která jsou uvedena ve správném pořadí a žádné navíc není řečeno. Jelikož je Kalliope nastavena tak, že po rozpoznání klíčového slova se ozve, aby byl uživatel informován, že může zadávat příkazy, vzniká zde určitá prodleva mezi vyřčením klíčového slova „Kalliopé“ a následujícím příkazem. Tato doba se pohybuje v závislosti na tom, jak dlouhá je reakce Kalliope na rozpoznání klíčového slova. Pokud se jedná o slova kratší, například „Ano?“, je prodleva dostatečně malá a ovládání působí přirozeněji, než pokud je reakce dlouhá, například „Jak ti můžu pomoci?“. Pro správné rozpoznání příkazu je nutné vyčkat až Kalliope dokončí svoji reakci, pak počkat přibližně jednu sekundu a teprve pak může být vyřčen příkaz. Pokud to tak uživatel neudělá, Kalliope nerozpozná vstup. Jedná se o jistou limitaci, se kterou je nutné počítat a přizpůsobit se jí, alternativně lze jako reakce volit krátká slova.

4.1.5 Výsledek

Chytré zrcadlo funguje spolehlivě přibližně na 80 %, někdy vznikají problémy s rozpoznáním příkazů kvůli nedostatečné prodlevě. Při připojení k internetu moduly fungují, chyby se téměř nevyskytují, pokud ano, tak nahodile. Doporučuji zrcadlo umístit do místností kde je šero, nebo nemají okna a plocha odražená v zrcadle bude tmavších barev, aby byly informace co nejlépe vidět.

4.2 Rozšiřitelnost a náklady

Tato práce je ve stavu, kde jsou propojené platformy MagicMirror², HomeAssistant a neuronová síť Kalliope. Konfigurace platform je ilustrativní a slouží jako ukázka, jak může vypadat vstupní bod pro vytvoření vlastní chytré domácnosti. Vybrané moduly propojují chytré zrcadlo s mobilními aplikacemi. Samotná platforma HomeAssistant nabízí možnost ovládání skrze mobilní telefon. Do budoucna je možné práci rozšířit přidáním jiných/vlastních modulů do MagicMirror², přidáním dalších platform ve formě „add-on“ do HomeAssistant a přidáním/vytvořením neuronů do Kalliope. Zároveň lze jakkoliv jinak využít Raspberry Pi pro běh aplikací na pozadí.

Tento projekt vyšel na přibližně 4 700 Kč, ale použil jsem některé věci, které jsem již doma měl – monitor, reproduktory, webkamera s mikrofonom. Bližší informace o ceně jsou v tabulce 4.1. Cenu by bylo možné snížit, pokud by někdo doma měl některé věci, které může pro projekt použít. Výsledná cena je poměrně pevná a pokud si někdo bude chtít chytré zrcadlo podle tohoto návodu vytvořit, musí počítat s cenou okolo 6 500 Kč. V této částce nejsou započítané provozní náklady (náradí apod.), ani cena za čas strávený s vyráběním konstrukce a konfigurací platform.

Komponenty	Orientační cena v Kč
Monitor	1 000
Raspberry Pi 4 Model B 2 GB	1 200
Adaptér k Raspberry Pi	300
Webkamera s mikrofonom	500
Reproduktory	300
Polopropustné sklo	1 500
LED pásek a příslušenství	200
Materiál na konstrukci	1 000
Drobná mikroelektronika	100
Paměťová karta + čtečka do USB	400
Součet	6 500

■ **Tabulka 4.1** Orientační cena použitých komponent

Kapitola 5

Závěr

Cílem práce bylo seznámit se s trendy v oblasti chytrých zrcadel. Na základě analýzy zvolit vhodnou platformu, vymyslet vhodnou konstrukci, zvážit možnosti budoucího rozšíření a komunikaci s jinými systémy (například telefonem). Výsledkem práce měl být prototyp chytrého zrcadla, které bude umožňovat komunikaci se senzory v chytré domácnosti, zobrazovat údaje z internetu o počasí, dopravě apod. a bude integrovat zvolený způsob ovládání. Vzniklý výtvar měl být podroben testování v laboratorních podmínkách. Vzhledem k širokému rozsahu zadání jsem začal velmi podrobnou analýzou.

V analýze byly probrány již dostupné projekty chytrých zrcadel, typy skel vhodné pro chytrá zrcadla, dostupné typy displejů, platformy pro chytré zrcadlo a domácí automatizaci, možné způsoby ovládání a volba hardware. Poté jsem navrhl konstrukci, kterou jsem podle 3D modelu i vytvořil. Praktická část práce byla věnována instalaci a nastavení operačního systému Debian 10 (Buster), platform HomeAssistant, MagicMirror² a neuronové síti Kalliope. Poté jsem zjišťoval, jakým způsobem je možné platformy konfigurovat tak, aby si je mohl upravit každý podle svých preferencí. Následně jsem vytvořil instalační příručku, která dává dohromady, jaké dílčí úkony je potřeba vykonat pro nastavení systému propojení platform jako celku. Vytvořený prototyp chytrého zrcadla zobrazuje: čas, zprávy, notifikace z mobilního telefonu, inteligentní seznam, kdo má aktuální den svátek, jaké významné události se staly v aktuální den, emailovou schránku, kalendář, fotky z osobního Instagramového profilu, tří denní předpověď počasí, aktuálně přehrávanou hudbu v aplikaci Spotify a stav LED diod, ovládaných platformou HomeAssistant. Zrcadlo je možné ovládat hlasovými příkazy pro: zapnutí/vypnutí LED pásku (přisvětlovacího světla na zrcadle), zapnutí/vypnutí červené a zelené LED diody připojené k vývojové desce NodeMCU v3, přepínání stránek v aplikaci MagicMirror. Realizovaný prototyp jsem podrobil testování v domácím prostředí. Zkoumal jsem čitelnost informací v různých světelných podmínkách, spolehlivost rozpoznání klíčového slova, spolehlivost zobrazovaných modulů a systému jako celku v běžném provozu. Nakonec uvádím, jakým způsobem lze práci rozšířit a jaká je orientační cena hotového projektu. Dalším cílem bylo vytvořit návod pro propojení platform HomeAssistant a MagicMirror. To se mi také podařilo splnit. V práci jsem detailně popsal kroky, jak postupovat při výrobě prototypu.

Rozhodně bych rád zmínil, že systémy, které jsou v této práci porovnány, nejsou všechny, ale pouze ty, které vypadaly lépe, nebo byly jednoduše dohledatelné. Je možné, že existují i lepší, ale pokud je těžké se o nich dozvědět, pak asi nebudou tak kvalitní, nebo budou nové. Vždy je prostor ke zlepšení a nelze říci, jestli někdo nevyvine nový systém, nebo aplikaci. Stejně tak to platí pro zařízení, aktuálně bylo použito Raspberry PI 4 model B. Jedná se o nejnovější z řad PI. Do budoucna však přijdou jiná a novější zařízení.

Při pokusech o první spuštění Raspberry PI jsem narazil na problém. Myslel jsem si, že pro „flashnutí“ operačního systému, na paměťovou kartu, bude stačit připojit mobilní telefon

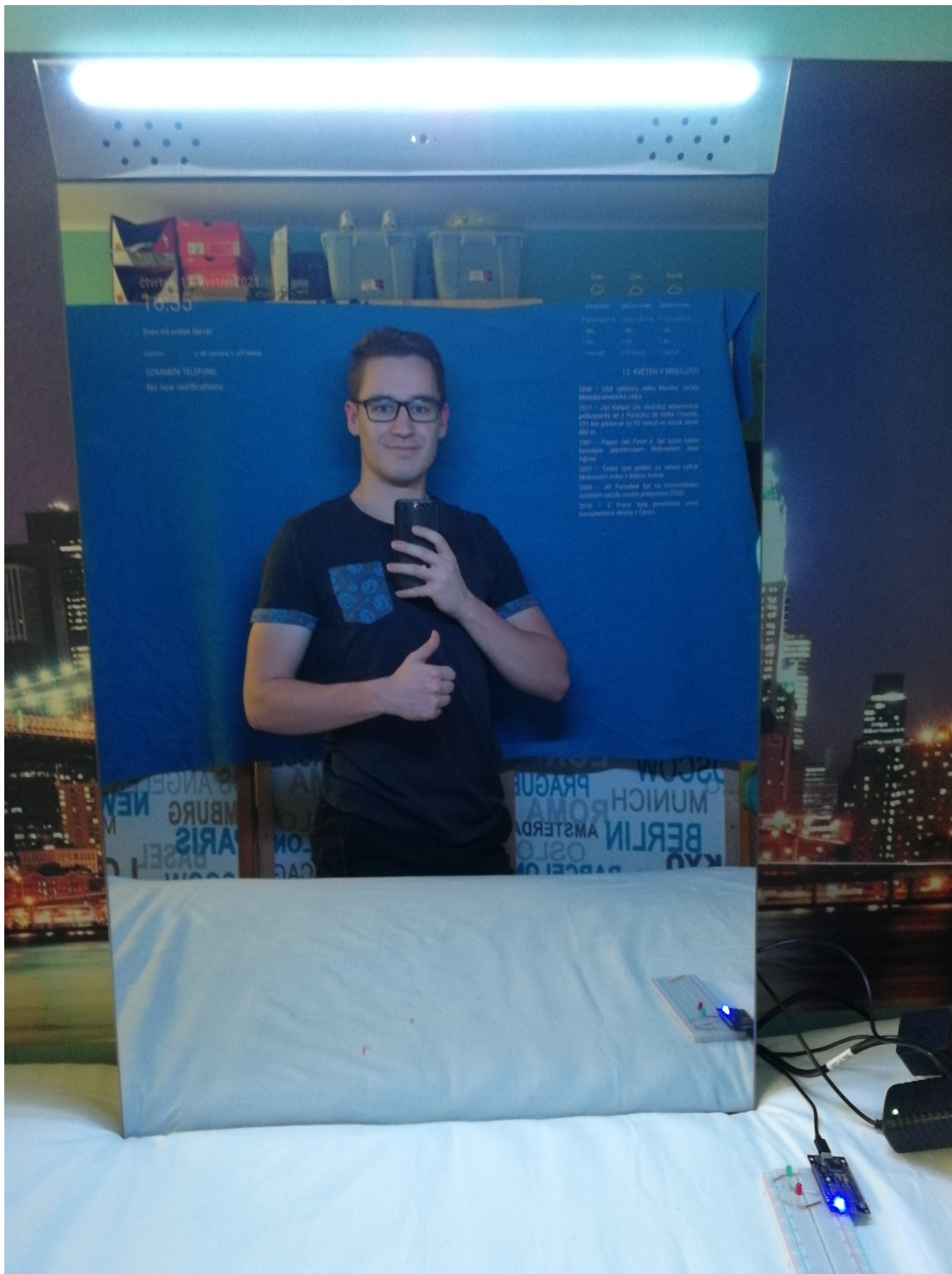
s pamětovou kartou uvnitř. Nicméně takhle to nefunguje. Počítač totiž zobrazuje takto připojenou pamětovou kartu jako součást úložiště uvnitř telefonu – *vyměnitelné úložiště*. Nástroj *balenaEtcher*, který se používá pro „flashnutí“ operačního systému na pamětovou kartu, toto úložiště nevidí. Vidí pouze úložiště připojené jako USB flash disk, tedy přes externí USB redukci pro pamětové karty. Nemůžu posoudit, jak je zobrazena pamětová karta, která je připojena přes interní konektor u notebooku, protože s tím nemám zkušenosti, ale tímto způsobem to určitě funguje taktéž. Vtipné je, že jsem při posuzování projektů ve výčtu použitých věcí viděl čtečku pamětových karet a původně jsem napsal, že mi to přijde zbytečné. Nicméně není potřeba ji kupovat, často ji má někdo z kamarádů doma na půjčení.

Jelikož jsem chtěl udělat zrcadlo co nejlépe, aby nebylo poznat, že se nejedná o klasické zrcadlo a zároveň jsem nechtěl být závislý na „zapuštění“ zrcadla do stěny, kde by muselo dojít k mechanickému upravení zdi, vytvořil jsem si vlastní unikátní bezrámečkovou konstrukci. Protože ji nikdo přede mnou nedělal, vytvořil jsem 3D model, který jsem upravoval, dokud jsem nepřišel na vhodné řešení. Se samotnou konstrukcí mi pomohl kamarád Jan Wijnhorst. Během jejího vytváření se objevovaly problémy, které na první pohled v modelu nebyly vidět. Například VESA držák na monitoru byl přidělaný do plastu a ne do kovové konstrukce. Naštěstí se podařilo na vzniklé problémy najít řešení, které bylo ve většině případů lepší, než to původní. Při představě, že výslednou konstrukci vytvořili dva studenti fakulty informačních technologií, je nutné říci, že se jedná o povedený kus. Kvůli hliníkové konstrukci funguje horní panel jako Faradayova klec, čímž se snižuje dosah bezdrátového připojení.

Při zprovoznění hlasového ovládání jsem narazil na problém, se kterým jsem nepočítal. Myslel jsem, že pouze nastavím modul hlasového ovládání a upravím příkazy tak, aby zrcadlo reagovalo podle toho, jak ho nastavím. Avšak moduly jsou udělané tak, že pouze spojí hlasového asistenta, nebo neuronovou síť s chytrým zrcadlem. Proto je potřeba mít někde samotného asistenta nainstalovaného. Téměř všechny moduly jsou starší než tři roky, a tak se dost často stane, že nějaká instalace selže, nebo se něco nepovede. Mnoho modulů využívalo k analýze řeči *Snowboy Hotword Detection engine*, který byl přesunut do jiného repozitáře. Instalační příručky a skripty přestaly fungovat a pro jejich správné nastavení je potřeba manuálně vyhledat a upravit potřebné soubory. Zprovoznit Amazon Alexa, nebo Google Assistant je náročné. Musí se založit developerské účty, zapnout účtování, které je na nějakou zkušební dobu zdarma. Pak je potřeba registrovat zařízení, vytvořit aplikaci a spárovat klíče. I když jsem postupoval podle návodů, nepodařilo se mi tyto asistenty zprovoznit. Proto jsem se nakonec rozhodl využít neuronovou síť Kalliope, která se liší tím, že má jednodušší instalaci.

Dost často se u modulů stane, že nefungují na první spuštění. To je způsobeno stářím modulů. Dost často chybí nějaký API klíč, který dříve nebyl potřeba, nebo neodpovídají verze. Například u některých hlasových asistentů a jejich integrace s chytrým zrcadlem byl využit python2, který již není oficiálně podporován. Jedna část (například modul pro MagicMirro²) byla přepsána do python3 a druhá (například samotný hlasový asistent) ne. Hlasového asistenta se proto nepodařilo zprovoznit.

Z důvodu rozsahu práce nebyl prostor se věnovat vytvoření dat pro rozpoznání klíčového slova *Snowboy Hotword Detection engine*. Pro rozpoznání slova „kalliope“ byla použita data vzniklé ze sedmi vzorků klíčového slova. To má za následek, že se poměrně často detekuje „falešné“ klíčové slovo například z komunikace mezi dvěma osobami.



■ **Obrázek 5.1** Vytvořený prototyp s autorem této práce

Bibliografie

1. TEEUW, Michael. *Magic Mirror: Part I – The Idea & The Mirror* [online]. 2014 [cit. 2021-04-25]. Dostupné z: <https://michaelteeuw.nl/post/80391333672/magic-mirror-part-i-the-idea-the-mirror>.
2. TEEUW, Michael. *Magic Mirror: Part II – The Monitor* [online]. 2014 [cit. 2021-04-25]. Dostupné z: <https://michaelteeuw.nl/post/81059936176/magic-mirror-part-ii-the-monitor>.
3. TEEUW, Michael. *Magic Mirror: Part III – The Casing* [online]. 2014 [cit. 2021-04-25]. Dostupné z: <https://michaelteeuw.nl/post/81784924322/magic-mirror-part-iii-the-casing>.
4. TEEUW, Michael. *Magic Mirror: Part IV – Install The Hardware* [online]. 2014 [cit. 2021-04-25]. Dostupné z: <https://michaelteeuw.nl/post/82565319113/magic-mirror-part-iv-install-the-hardware>.
5. TEEUW, Michael. *Magic Mirror: Part V – Installing the Raspberry Pi* [online]. 2014 [cit. 2021-04-25]. Dostupné z: <https://michaelteeuw.nl/post/83188136918/magic-mirror-part-v-installing-the-raspberry-pi>.
6. TEEUW, Michael. *Magic Mirror: Part VI – Production of the Interface* [online]. 2014 [cit. 2021-04-25]. Dostupné z: <https://michaelteeuw.nl/post/83916869600/production-of-interface>.
7. TEEUW, Michael. *[The end result of the Magic Mirror]* [online]. 2015 [cit. 2021-04-25]. Dostupné z: <https://michaelteeuw.nl/post/84026273526/and-there-it-is-the-end-result-of-the-magic>.
8. [ZACH]. *Add voice controls to your Raspberry Pi using Jasper* [online]. 2021-04-05 [cit. 2021-04-25]. Dostupné z: <https://howchoo.com/pi/add-voice-controls-to-your-raspberry-pi-using-jasper>.
9. [ZACH]. *Build a Voice-Controlled DIY Raspberry Pi Smart Mirror with Jasper* [online]. 2020-11-19 [cit. 2021-04-25]. Dostupné z: <https://howchoo.com/project/mzu3ndm2otu/building-a-voice-controlled-smart-mirror-with-raspberry-pi-and-jasper>.
10. KOUAO, Eben. *Smart Mirror Touchscreen (with Face ID) using Raspberry Pi 4 | Full Tutorial* [online]. 2020-05-06 [cit. 2021-04-25]. Dostupné z: https://www.youtube.com/watch?v=RWjvJq4Zabk&ab_channel=EbenKouao.
11. KOUAO, Eben. *Smart Mirror Touchscreen (with FaceID) Tutorial – Raspberry Pi 4* [online]. 2020-05 [cit. 2021-04-25]. Dostupné z: <https://smartbuilds.io/smart-mirror-touchscreen-raspberry-pi/>.

12. *Waveshare 13.3" Magic mirror, hlasový asistent, dotykové ovládání* [online] [cit. 2021-04-25]. Dostupné z: <https://rpishop.cz/kategorie/2414-waveshare-133-magic-mirror-hlasovy-asistent-dotykovove-ovladani.html>.
13. [DM-THOMAS]. *Alexa Smart Mirror – Mirror TV – X-Large Size 10"Display* [online]. 2021-02-22 [cit. 2021-04-25]. Dostupné z: <https://www.ebay.co.uk/itm/Alexa-Smart-Mirror-Mirror-TV-X-Large-Size-10-Display/184661946104>.
14. HIMIRROR. *HiMirror Mini Premium X* [online] [cit. 2021-04-25]. Dostupné z: <https://www.himirror.com/en-US/eshop/product/HiMirrorPremiumX>.
15. *Vodotěsné inteligentní zrcadlo – skryté LED TV Kouzelné sklo, postavený ve zdi televize* [online] [cit. 2021-04-25]. Dostupné z: <https://www.fruugo.cz/vodotsne-inteligentni-zrcadlo-skryte-led-tv-kouzelné-sklo-postaveny-ve-zdi-televize/p-51856536-104427194>.
16. TWO WAY MIRRORS. *DIY Smart Mirror: Step-By-Step Guide* [online]. [2018] [cit. 2021-04-25]. Dostupné z: <https://www.twowaymirrors.com/diy-smart-mirror/>.
17. TWO WAY MIRRORS. *Glass Two Way Mirror* [online] [cit. 2021-04-25]. Dostupné z: <https://www.twowaymirrors.com/glass/>.
18. TWO WAY MIRRORS. *Acrylic Two Way Mirror* [online] [cit. 2021-04-25]. Dostupné z: <https://www.twowaymirrors.com/acrylic/>.
19. TWO WAY MIRRORS. *Smart Mirror Product Information* [online] [cit. 2021-04-25]. Dostupné z: <https://www.twowaymirrors.com/smart-mirror/>.
20. TWO WAY MIRRORS. *Dielectric TV Mirror* [online] [cit. 2021-04-25]. Dostupné z: <https://www.twowaymirrors.com/tvmirror/>.
21. TWO WAY MIRRORS. *One Way Mirror Film* [online] [cit. 2021-04-25]. Dostupné z: <https://www.twowaymirrors.com/one-way-mirror-film/>.
22. THOMPSON, Sora. *LCD v. LED v. OLED: Understanding Different Display Panel Types* [online]. 2018-08-26 [cit. 2021-04-25]. Dostupné z: <https://www.tomshardware.com/reviews/lcd-led-led-oled-panel-difference,5394.html>.
23. LEE, Ernest; WANG, Chunming (Kevin); YUREK, Jeff; MA, Ruiqing. *A New Frontier for Quantum Dots in Displays* [online]. 2018-12-03 [cit. 2021-04-25]. Dostupné z: <https://www.nanosysinc.com/white-papers/2019/10/10/a-new-frontier-for-quantum-dots-in-displays>.
24. TEEUW, Michael. *MagicMirror²* [online]. [2016] [cit. 2021-04-25]. Dostupné z: <https://magicmirror.builders/>.
25. TEEUW, Michael. *Documentation* [online]. 2021-01-04 [cit. 2021-04-25]. Dostupné z: <https://docs.magicmirror.builders/>.
26. DAVYDOV, Michael. *HomeAssistant for newcomers: What it is, what is hassio, hassos, hassbian, 101 and cookies* [online]. 2019-06-22 [cit. 2021-04-25]. Dostupné z: <https://community.home-assistant.io/t/homeassistant-for-newcomers-what-it-is-what-is-hassio-hassos-hassbian-101-and-cookies/123004/1>.
27. HOMEASSISTANT. *Installation* [online] [cit. 2021-04-25]. Dostupné z: <https://www.home-assistant.io/installation/>.
28. HOMEASSISTANT. *Home Assistant Supervisor* [online] [cit. 2021-04-25]. Dostupné z: <https://developers.home-assistant.io/docs/supervisor/>.
29. DATONA. *Dotyková obrazovka* [online] [cit. 2021-04-25]. Dostupné z: https://www.datona.cz/content/dotyková_obrazovka.

30. MILLER, Nelson. *What is an Infrared Touchscreen and How Does It Work?* [Online]. 2017-11-08 [cit. 2021-04-25]. Dostupné z: <https://www.nelson-miller.com/infrared-touchscreen-work/>.
31. YOUNG, Park Joon; JIN, Jo Hyo; WOO, Samuel; LEE, Dong Hoon. BadVoice: Soundless voice-control replay attack on modern smartphones. In: *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*. 2016, s. 882–887. Dostupné z DOI: 10.1109/ICUFN.2016.7537163.
32. REVIEWS.COM STAFF. *The Best Voice Assistants* [online]. 2020-10-28 [cit. 2021-04-25]. Dostupné z: <https://www.reviews.com/home/smart-home/best-voice-assistant/>.
33. HALFACREE, Gareth. *Raspberry Pi 4 B: How Much RAM Do You Really Need?* [Online]. 2020 [cit. 2021-05-12]. Dostupné z: <https://www.tomshardware.com/news/raspberry-pi-4-how-much-ram>.
34. *Electron memory usage compared to other cross-platform frameworks* [online] [cit. 2021-05-12]. Dostupné z: <http://roryok.com/blog/2017/08/electron-memory-usage-compared-to-other-cross-platform-frameworks/>.
35. LIBRE COMPUTER. *AML-S805X-AC (La Frite)* [online] [cit. 2021-04-25]. Dostupné z: <https://libre.computer/products/boards/aml-s805x-ac/>.
36. RASPBERRY PI. *Raspberry Pi Zero W* [online] [cit. 2021-04-25]. Dostupné z: <https://www.raspberrypi.org/products/raspberry-pi-zero-w/>.
37. SINOVOIP. *BPI-M2 Zero* [online] [cit. 2021-04-25]. Dostupné z: <http://www.banana-pi.org/m2z.html>.
38. XUNLONG SOFTWARE. *[Orange Pi PC]* [online] [cit. 2021-04-25]. Dostupné z: <http://www.orangepi.org/orangepipc/>.
39. LIBRE COMPUTER. *ALL-H3-CC (Tritium)* [online] [cit. 2021-04-25]. Dostupné z: <https://libre.computer/products/boards/all-h3-cc/>.
40. LIBRE COMPUTER. *ROC-RK3328-CC (Renegade)* [online] [cit. 2021-04-25]. Dostupné z: <https://libre.computer/products/boards/roc-rk3328-cc/>.
41. LIBRE COMPUTER. *AML-S905X-CC (Le Potato)* [online] [cit. 2021-04-25]. Dostupné z: <https://libre.computer/products/boards/aml-s905x-cc/>.
42. PINE64. *[ROCK64 Rev2]* [online] [cit. 2021-04-25]. Dostupné z: <https://www.pine64.org/devices/single-board-computers/rock64/>.
43. SINOVOIP. *BPI-M64* [online] [cit. 2021-04-25]. Dostupné z: <http://www.banana-pi.org/m64.html>.
44. RASPBERRY PI. *Pi 4 Model B* [online] [cit. 2021-04-25]. Dostupné z: <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/>.
45. VÁVRA, Tomáš. *Sklenářství Praha 6 – Tomáš Vávra* [online]. 2021-05-09 [cit. 2021-05-12]. Dostupné z: <https://www.praha-sklenarstvi.cz/>.
46. HOMEASSISTANT. *Documentation* [online] [cit. 2021-05-12]. Dostupné z: <https://www.home-assistant.io/docs/>.
47. KALLIOPE. *Kalliope doc* [online] [cit. 2021-05-12]. Dostupné z: <https://kalliope-project.github.io/kalliope/>.
48. DEBIAN. *Installing Debian 10.9* [online]. 2021-01-28 [cit. 2021-05-12]. Dostupné z: <https://www.debian.org/releases/buster/debian-installer/>.
49. ESPRESSIF [online] [cit. 2021-05-12]. Dostupné z: <https://www.espressif.com/>.
50. HOMEASSISTANT. *Integrations* [online] [cit. 2021-05-12]. Dostupné z: <https://www.home-assistant.io/integrations/>.

51. TEEUW, Michael. *3rd Party Modules* [online]. 2021-05-09 [cit. 2021-05-12]. Dostupné z: <https://github.com/MichMich/MagicMirror/wiki/3rd-party-modules>.
52. KALLIOPE. *Hlavní strana* [online] [cit. 2021-05-12]. Dostupné z: https://cs.wikipedia.org/wiki/Hlavn%C3%AD_strana.
53. KALLIOPE. *Starter kits* [online] [cit. 2021-05-12]. Dostupné z: https://kalliope-project.github.io/starter_kit.html.
54. JON. *Complete Raspberry Pi Magic Mirror Tutorial (Newbie Friendly)* [online]. 2020-07-28 [cit. 2021-05-12]. Dostupné z: <https://www.magicmirrorcentral.com/complete-raspberry-pi-magic-mirror-tutorial/>.

Obsah přiloženého média

	readme.txt	stručný popis obsahu média
	image.img	image SD karty z Raspberry Pi se softwarovou částí práce
	src		
		config.js konfigurační soubor platformy MagicMirror ² bez citlivých údajů
		mirror.yml zdrojový soubor mozku Kalliope se synapsemi pro hlasové ovládání
	thesis.pdf	text práce ve formátu PDF
	images	adresář s fotkami z vytváření konstrukce