

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



Dokumentace k projektu do předmětu SIN  
Systém chytré domácnosti - individuální zadání

## Obsah

<b>1 Úvod</b>	<b>1</b>
<b>2 Zadání projektu</b>	<b>1</b>
2.1 Změny zadání . . . . .	2
<b>3 Návrh systému</b>	<b>2</b>
3.1 Ovládací kód . . . . .	2
<b>4 Implementace</b>	<b>4</b>
4.1 Periferní moduly . . . . .	4
4.2 Dashboard ve frameworku Django . . . . .	5
4.3 Skill pro platformu Mycroft . . . . .	5
4.4 Komunikace přes MQTT . . . . .	6
4.5 Způsob komunikace jednotlivých komponent . . . . .	6
<b>5 Instalace</b>	<b>6</b>
<b>6 Struktura odevzdaného archivu</b>	<b>7</b>
<b>7 Nedokončené a změněné body zadání</b>	<b>7</b>
<b>8 Závěr a plány do budoucna</b>	<b>7</b>

## 1 Úvod

Tato dokumentace se zabývá projektem do předmětu SIN, kde budou v jednotlivých sekcích popsány mj. zadání, instalace, návrh a realizace projektu. Dále zde budou zmíněny části, které byly oproti původnímu zadání neimplementovány a v závěru bude shrnutý mj. i plán pro rozšíření projektu.

## 2 Zadání projektu

Dle dohody s garantem předmětu bylo cílem projektu implementovat reálný (sub)systém chytré domácnosti, který by umožňoval kontrolovat současné vybavení domácnosti, které ovšem původně nebylo navrženo jako „chytré“ a bude třeba vytvořit moduly, které umožní připojení k internetu věcí. Systém se bude skládat z těchto součástí:

- Centrální počítač, kde poběží všechny „centrální“ služby jako například MQTT broker, logika systému, databáze a webový server.
- Jednotlivé moduly, realizované na platformě Wemos D1 mini<sup>1</sup>. tyto moduly psolu následně komunikují pomocí protokolu MQTT.

<sup>1</sup>[https://docs.wemos.cc/en/latest/d1/d1\\_mini.html](https://docs.wemos.cc/en/latest/d1/d1_mini.html)

## 2.1 Změny zadání

Jelikož projekt není zamýšlen jako „jednorázový“, ale autorovi má skutečně sloužit i mimo rozsah projektu a předmětu, v zadání proběhly některé změny. Bylo například přidáno ovládání hlasem pomocí open-source hlasového asistenta Mycroft<sup>2</sup>, umožňující hlasem ovládat světla v bytě.

## 3 Návrh systému

V následující sekci se budu podrobněji věnovat návrhu systému, jeho jednotlivých částí a pokusím se nastínit i fyzickou realizaci. Jak již bylo řečeno v předchozí sekci, systém se bude skládat z centrálního počítače, kde poběží většina logiky a modulů, které se budou starat o sběr údajů, případně o ovládání.

Centrální počítač je postaven na platformě Raspberry Pi 4 s OS Raspbian v „headless“ módu, tedy bez grafického uživatelského prostředí, kvůli lepšímu výkonu. Ovládání bude probíhat bezdrátově pomocí ssh. Kvůli hlasovému asistentu ?? bude také připojen reproduktor a USB mikrofon. Raspberry Pi implicitně bootuje z SD karty, jelikož ale SD karty nejsou stavěny na trvalý provoz a provoz operačního systému, bude upraven firmware Raspberry tak, aby umožňoval bootování z pevného disku, připojenému k USB. K tomuto účelu poslouží SSD disk.

Na centrálním počítači bude nainstalována většina logiky systému. V následujících odstavcích se pokusím krátce shrnout jejich účel.

### 3.1 Ovládací kód

Bude se jednat o několik skriptů v jazyce python3, které budou mít za cíl všechny další technologie spojit do funkčního celku.

Databáze InfluxDB<sup>3</sup>

Databáze bude sloužit pro ukládání dat z teplotní stanice. Tato databáze byla zvolena z toho důvodu, že slouží primárně k ukládání dat, která mají charakteristiku časových řad. Jedná se například o data ze senzorů, které byly použité v teplotní stanici. Samotná databáze bude obsahovat celkem tři měření - číselné řady - samostatně pro každý údaj z teplotní stanice. Konkrétně se jedná o teplotu, vlhkost a hodnotu z kouřového senzoru.

MQTT broker

MQTT je protokol určený pro komunikaci s malým datovým tokem. V jedné topologii se vždy vyskytuje jeden centrální prvek, nazývaný „broker“. Komunikující zařízení přitom neznají adresu sebe navzájem, ale pouze adresu brokera, což umožňuje velmi jednoduché přidávání a odebírání jednotlivých klientů.

Princip fungování protokolu MQTT je velmi jednoduchý. jednotlivá zařízení - klienti - buď odebírají nebo publikují do tzv. „topiců“ neboli témat. Struktura topicu (český ekvivalent se prakticky nepoužívá) je hierarchická a podobá se adresářové struktuře bez kořenového adresáře. Příkladem topicu budiž například `home/livingroom/dimmerlight`. Pokud se klient rozhodne nějaký topic odebírat, znamená to, že mu broker přepoše jakoukoli zprávu, která byla publikována. Naopak, pokud klient nějakou zprávu publikuje, broker se postará o to, aby byla doručena všem ostatním klientům, kteří odebírají daný topic.

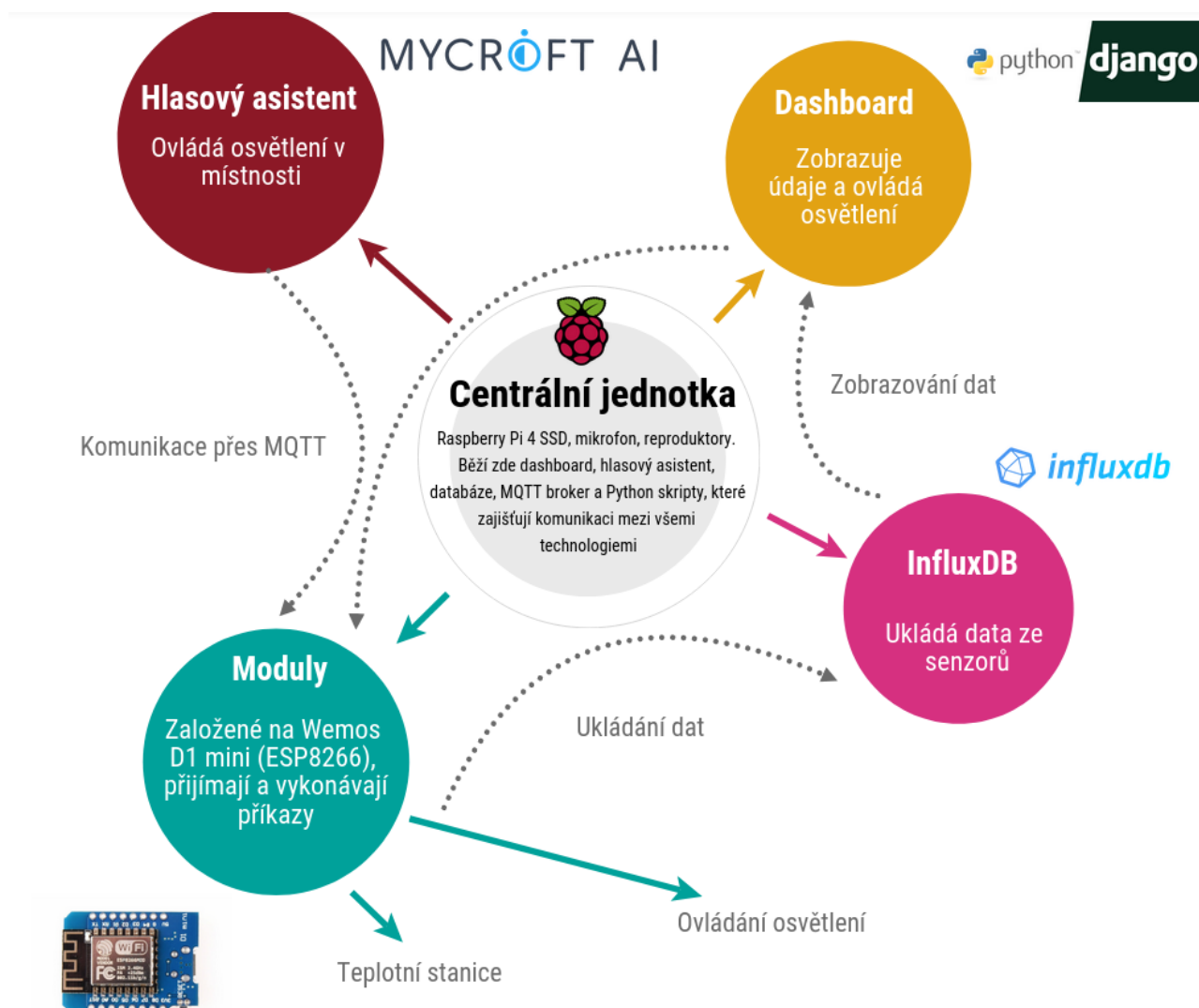
Hlasový asistent Mycroft

I když původní zadání s hlasovým asistentem nepočítalo, autorovi přišlo zajímavé a žádoucí ho do celého systému začlenit, jelikož mj. usnadňuje ovládání osvětlení a v budoucnu by měl umožňovat i další interakci se systémem. V neposlední řadě se jedná o zajímavou technologii a alternativu k majoritním službám a produktům typu Google Home.

---

<sup>2</sup><https://mycroft.ai/>

<sup>3</sup><https://www.influxdata.com/>



Obrázek 1: Grafické znázornění systému

Mycroft je open-source hlasový asistent po vzoru Google Asistenta, Amazonu echo a dalších. Jeho výhodou je již zmíněný otevřený kód a možnost jednoduše vytvářet další rozšíření - „skills“ - o kterých bude řeč dále. Další výhodou je i možnost lokální transformace řeči na text a zpět, což Google ani Amazon neumožňují. Celkově dbá Mycroft na soukromí uživatele a není třeba se bát, že by byla náhodně nahrávky využívala mateřská firma třeba i bez vědomí uživatele<sup>4</sup>.

Webový server, na kterém běží „dashboard“ pro získávání informací a ovládání

Jako webový server bude použit framework Django. Jedná se o backendový framework v jazyce Python3, bude tedy zjednodušené propojování jednotlivých technologií, jelikož v tomto jazyce budou napsány i všechny propojovací skripty.

## 4 Implementace

Sekce návrh poskytla obecný pohled na systém. Následující sekce podrobněji probere technické aspekty systému a některé podstatné detaily implementace. Jelikož je systém poměrně rozsáhlý a bohužel ho není možné jednoduše replikovat (například pro účely kontroly při bodování aj.), je tato sekce psaná pokud možno tak, aby si čtenář mohl udělat celkový pohled na systém. Důležité aspekty jsou podrobně vysvětleny, nedůležité aspekty poté jen zmíněny.

### 4.1 Periferní moduly

Jak již bylo zmíněno, základem každého modulu je prototypovací deska Wemos D1 mini, jejímž základem je MCU ESP8266. V rámci tohoto projektu byly vytvořeny dva moduly; první pro ovládání osvětlení, druhý pro měření teploty. Oba budou probírány níže.

Modul pro měření teploty obsahuje dva senzory: kombinovaný senzor teploty a vlhkosti DHT-11<sup>5</sup> a senzor pro měření plynů v ovzduší MQ-2<sup>6</sup>. Druhý jmenovaný senzor je primárně používán pro detekci hořlavých plynů jako alkohol nebo metan, ale obstojně funguje i jako detektor kouře. Senzor DHT11 je ovládán pomocí knihovny `dht.h`, senzor MQ-2 je poté ovládán přímo přes signální vodič.

Princip činnosti tohoto modulu je jednoduchý. Nejprve jsou inicializovány všechny potřebné objekty (klient pro WiFi a MQTT komunikaci, objekt pro senzor DHT11, seriová linka) a poté začne MQTT klient čekat na zprávu na topicu `home/livingroom/tempstation`. Pokud obdrží zprávu s textem `temperature`, `humidity` nebo `gas`, změří požadovanou veličinu a publikuje ji napříslušný topic (`home/livingroom/<velicina>`).

Druhý modul ovládá dvě LED žárovky. První z nich je RGB LED žárovka značky Livarno, která podporuje ovládání přes IR ovladač. Jelikož ale samotný způsob komunikace je neveřejný, bylo třeba provést „reverzní inženýrství“ a s pomocí Arduina a IR přijímače rozkódovat způsob, jakým se žárovka ovládá. Kombinací reverzního inženýrství a Github repozitáře uživatele `ikke-t`<sup>7</sup> se nakonec úspěšně podařilo zjistit ovládací kódy pro základní funkce žárovky - zapínání, vypínání, úprava jasu a nastavení základních barev.

Princip ovládání této žárovky tedy tkví v simulaci příkazů od původního IR ovladače a modul zde funguje jako adaptér mezi MQTT a IR protokolem NEC, který je pro IR komunikaci v tomto případě používán.

Druhá LED žárovka je ovládána pomocí stmívače RobotDyn<sup>8</sup> pomocí knihovny `RBDdimmer.h`. Přes MQTT obdrží modul číselnou hodnotu od 0 do 100, která vyjadřuje procentuální „svítivost“ žárovky.

<sup>4</sup><https://www.lupa.cz/aktuality/nahravky-z-google-home-poslouchaji-lidsti-experti-slysi-i-to-co-se-nahravat-nemelo/>

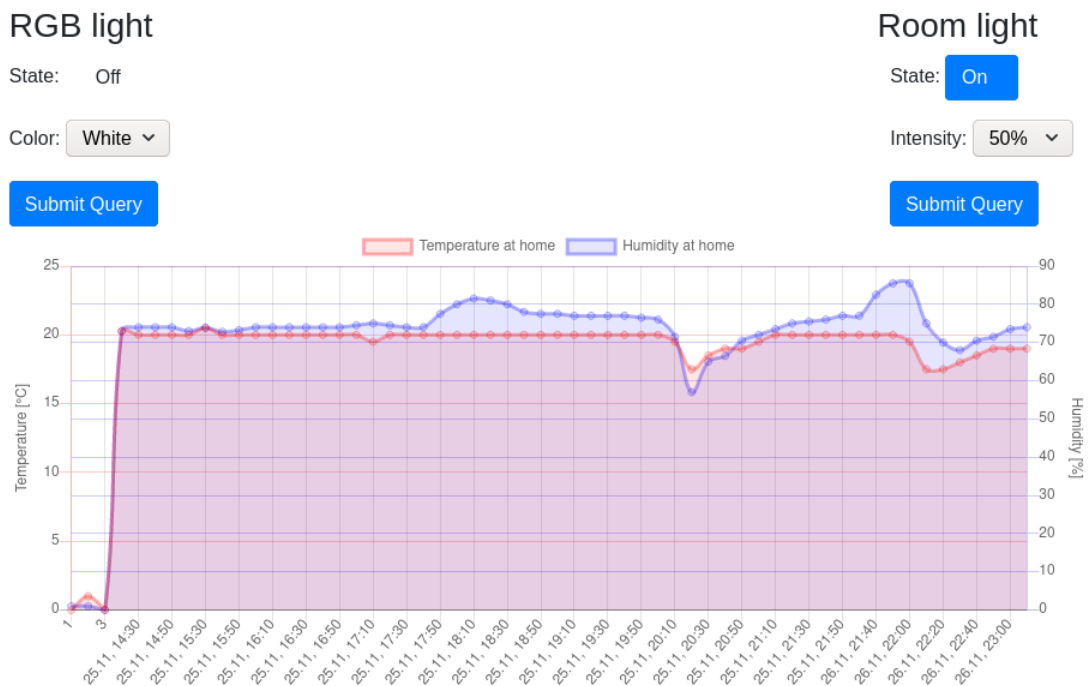
<sup>5</sup><https://www.hadex.cz/m441k-teplotni-cidlo-a-vlhkometer-dht11/?gclid=EAIaIQobChMI2IuNkNql7QIVDJ53Ch2Ibg3zEAQYAIAABwE>

<sup>6</sup><https://www.gme.cz/modul-mq2-s-detektorem-horlavych-plynu>

<sup>7</sup>Dostupný zde: <https://github.com/ikke-t/livarnoluxball-remote>

<sup>8</sup><https://robotdyn.com/ac-light-dimmer-module-1-channel-3-3v-5v-logic-ac-50-60hz-220v-110v.html>

## SIN project



Obrázek 2: Ukázka finálního dashboardu

Princip činnosti je opět podobný jako u prvního modulu. MQTT klient čeká na zprávu na konkrétním topicu (`home/livingroom/rgbled` pro RGB zářivku a `home/livingroom/dimmer` pro stmívatelnou zářivku) a podle obsahu zprávy buď nastaví požadovanou úroveň svítivosti nebo vyšle příslušný IR signál.

## 4.2 Dashboard ve frameworku Django

Dashboard umožňuje ovládání osvětlení a zároveň zobrazuje naměřené údaje z čidel teploty a vlhkosti. K tvorbě byl použit framework Django a několik Javascriptových skriptů (především `Charts.js`<sup>9</sup>, sloužících především k zobrazování dat.

Pomocí protokolu Websocket<sup>10</sup> je poté řešena aktualizace dat bez nutnosti aktualizovat stránku. Na straně frameworku je jeden otevřený socket, sloužící pro komunikaci frontendu a backendu a také frontendu a databáze (přes třídu `Updater`).

## 4.3 Skill pro platformu Mycroft

Hlasový asistent Mycroft, díky skutečnosti, že je open-source, umožňuje velmi jednoduché vytváření tzv. „skillů“, dovedností, které rozšiřují jeho funkcionalitu. Vytvořený skill `led_controller_skill` umožňuje hlasem ovládat obě LED světla, i když poněkud zjednodušeně. Pro stmívatelné světlo umožňuje jeho zapnutí

<sup>9</sup><https://www.chartjs.org/>

<sup>10</sup><https://www.websocket.org/>

a vypnutí, pro RGB světlo dále umožňuje změnu barvy na jeden ze základních odstínů - modrá, červená, zelená.

Níže bude popsáno, jak samotný skill funguje (skill samotný se nachází ve složce `skill`). Ve složce `vocab` se nachází jednotlivé soubory pro klíčová slova dané charakteristiky (použití bude vysvětleno níže), tedy například stav světla může být zapnutý nebo vypnutý. Ve složce `dialog` se poté nachází odpovědi, které Mycroft odpoví v reakci na pronesený příkaz. Všechna logika se poté nachází v souboru `__init__.py`. Skill je schopný provádět tzv. „intents“, tedy záměry uživatele. Tyto záměry jsou poté obsluhovány příslušnou funkcí. Záměr je tvořen jednotlivými klíčovými slovy, na jejichž základě může funkce měnit své chování.

Konkrétně skill `led_controller_skill` umožňuje splnit dva záměry; změnit stav LED RGB světla nebo zapnout, popřípadě vypnout stmívatelné světlo. Při rozpoznání správného záměru je zjištěna požadovaná akce (v závislosti na klíčových slovech) a přes MQTT odeslána příslušná zpráva na daný topic. Mycroft nakonec odpoví, že splnil příkaz a v jakém stavu se momentálně dané světlo nachází.

## 4.4 Komunikace přes MQTT

Tuto komunikaci zapouzdřuje několik tříd, především `LEDPublisher` pro komunikaci s RGB LED zářivkou a stmívatelnou žárovkou a třídy `TempstationPublisher` a `TempstationSubscriber`, které zajišťují komunikaci s teplotní stanicí.

## 4.5 Způsob komunikace jednotlivých komponent

Pokud potřebuje některá z částí systému publikovat nebo odebrat nějaké MQTT zprávy, vytvoří si lokální instanci tříd popsaných v předchozí subsekcí. Pro komunikaci s databází InfluxDB slouží třída `InfluxBridge`. Tato třída pouze zapouzdřuje jednotlivé příkazy pro nahrávání dat do databáze a případné dotazy na ni. Posledním článkem je třída `Updater`, která periodicky kontroluje, zda se nezměnila data v databázi. Pokud nějaká nová data přibyla, jsou pomocí protokolu WebSocket odeslána do frontendu dashboardu, kde jsou zobrazena.

# 5 Instalace

Instalace systému je poměrně komplikovaná a vzhledem k tomu, že bez konkrétního hardware (periferní moduly) je prakticky nemožná, nebude zde uveden postup zprovoznění kompletního systému, ale pouze přehledový způsob instalace jednotlivých součástí. V následujícím seznamu budou uvedeny základní kroky instalace jednotlivých součástí. Jelikož systém vyžaduje i funkční připojení k internetu a konkrétní IP například MQTT brokeru nebo Influx databáze, i po případné úspěšné instalaci na novém stroji systém nebude fungovat<sup>11</sup>.

Následující kroky platí pro OS na bázi linuxu.

1. Pro nahrávání kódu do periferních modulů je třeba program Arduino IDE, dostupný například z <https://www.arduino.cc/en/software>. Po jeho instalaci je nutné přidat podporu pro desky ESP8266. Přes File-Preferences se do kolonky „Additional Boards Manager URLs“ přidá následující odkaz: [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json). Následně přes Tools-Board-Boards Manager vyhledáme „ESP8266“ a nainstalujeme příslušnou knihovnu.
2. Instalace frameworku Django probíhá přes utilitu pip příkazem `pip install Django`. Jelikož je ovšem Django v základu synchronním frameworkem, je nutné nainstalovat i balík `channels` (`pip install channels`).

<sup>11</sup>Ze stejného důvodu byly z kódu odstraněny i všechny výskyty hesla k wifi autora projektu s komentářem na místě odstranění

3. Instalace platformy hlasového asistenta Mycroft obecně velmi záleží na konkrétním operačním systému a hardware daného stroje, jelikož je známo, že s některými reproduktory a mikrofony nemusí fungovat a u některých se mohou vyskytovat náhodné chyby. Obecným postupem je ovšem stažení oficiálního Github repozitáře<sup>12</sup> a následně spuštění instalačního skriptu pomocí `bash dev_setup.sh`. Tento interaktivní skript provede uživatele instalací krok za krokem. Posledním krokem je přidání skillu `led_controller_skill`, který lze stáhnout z <https://github.com/Honza0297/led-controller-skill>. Mycroft vyžaduje, aby byla složka se skillem zároveň gitovým repozitářem, jinak mívá problémy s jeho správným načtením.
4. Instalace databáze Influx se liší systémem od systému, podrobné instalační kroky pro všechny linuxové systémy lze nalézt zde: <https://docs.influxdata.com/influxdb/v1.7/introduction/installation/>.

## 6 Struktura odevzdaného archivu

V následující sekci popisují, jakou má strukturu odevzdávaný archiv. Jelikož se celý projekt sestává z mnoha různých částí, nemusí být struktura na první pohled jasná.

- `xberan43`: kořenový adresář
  - `sketches` obsahují Arduino sketche pro jednotlivé moduly (včetně modulu pro ovládání okenních žaluzií, který nakonec nebyl uveden do „ostrého“ provozu).
  - `led-controller-skill` obsahuje všechny soubory nutné pro funkčnost skillu pro platformu Mycroft.
  - `scripts` obsahuje všechny skripty v jazyce python3.
  - `website` obsahuje kompletní Django projekt, nutný pro případně zprovoznění webových stránek. Samotný kód, který přímo utváří dashboard, se nachází na cestě `website/sinapp` a jedná se především o soubory `views.py`, `consumers.py` a `templates/home.html`.

## 7 Nedokončené a změněné body zadání

Bohužel, kvůli časovému presu a problémům při jeho instalaci není modul pro ovládání žaluzií provozuschopný. Ovládací kód je sice napsaný, ale při instalaci se ukázalo, že v reálném světě nefunguje dobře. Problémy byly zejména s realizací zastavení a s kombinací ovládání pomocí motoru a manuálního ovládání.

Jelikož bez tohoto modulu postrádalo smysl implementovat část pro tvorbu pravidel (fakticky by nešla vytvořit žádná smysluplná pravidla), nedošlo ani k implementaci této části zadání.

Na druhou stranu byl projekt oproti původní dohodě rozšířen o hlasového asistenta Mycroft, který umožňuje hlasové ovládání světel.

## 8 Závěr a plány do budoucna

Zadání projektu bylo částečně splněno. Byly vytvořeny dva ze tří modulů (oba jsou reálně používány), centrální jednotka a většina logiky systému, stejně jako „dashboard“ ve formě webové stránky, umožňující ovládat světla z jakéhokoli zařízení, které se umí připojit na internet. Nad rámec zadání byl poté do systému včleněn hlasový asistent Mycroft, který umožňuje ovládat světla i prostřednictvím hlasu.

---

<sup>12</sup><https://github.com/MycroftAI/mycroft-core>



Systém je stále ve vývoji. Pro nejbližší budoucnost se autor plánuje zaměřit na rozhraní mezi InfluxDB a webovým frameworkem Django, které je v současné chvíli sice funkční, ale není ideální. Jako zajímavé rozšíření se jeví automatizace rozsvěcování vzhledem k denní době a přítomnosti členů domácnosti v málo frekventovaných místech (například na chodbě). Dalším plánem je včlenění ozvučení do celého systému, které by se následně stalo ovladatelným přes webový dashboard. Ostatní rozšíření jsou již velmi konkrétní; například úprava stolu IKEA Skarsta, který v současné chvíli umožňuje manuální změnu výšky, tak, aby bylo možné výšku nastavovat pomocí elektronicky.