

**Autor:** Bc. Pavel Koupy

**Abstrakt:**

Průzkum dostupných open-source nástrojů a frameworků pro realizaci monitorování, řízení IoT systému. Součástí projektu je demonstrace jednoho z vybraných nástrojů. Demonstrace bude zahrnovat vizualizaci za pomoci dashboardu, dále databázi a jednoduché řízení prvků založených na mikrokontrolérech ESP32 a ESP8266. Prvky v demu budou komunikovat pomocí protokolu MQTT. Demonstrace se bude skládat z jednoduchých senzorů typu teploměr či vlhkoměr a relé modulů jako akčních prvků.

**Průzkum technologií:**

Průzkum vychází ze srovnání webových portálů, které poukázaly na oblíbený software v oblasti open-source komunity IoT vývoje. Dalším výchozím bodem nebo spíše motivací byla zkušenost s bare-metal řešením[1], které bylo vytvořeno bez použití nástrojů pro správu a monitoring IoT zařízení. Z tohoto hlediska se vždy zdá lepším východiskem použít některý volně dostupných řešení, protože tyto technologie jsou aktuálním trendem.

Důležitými faktory, z hlediska konečné konstrukce jakéhokoli IoT zařízení, jsou dostupné komunikační protokoly, množství podporovaných technologií třetích stran zahrnující hardwarová i softwarová řešení a rychlost integrace s uvedením do provozu. Z uživatelského hlediska je důležitým prvkem řešení hierarchických vztahů modelujících reálnou situaci použití služby, tedy vztah řešící vztahy jednotlivých zařízení vůči jednotlivým jednotlivým uživatelům a rolím v informačních též middleware pro IoT.

**Frameworky a middleware**

Obvykle jsou nástroje pro IoT architektonicky řešeny formou monolitického systému obsahujícího všechny podstatné součásti pro extrakci, analýzu a ovládání nebo případně vizualizaci. Druhou novější strukturou je řešení pomocí microservice architektury, kdy jsou jednotlivé součásti IoT frameworku či middlewaru děleny do samostatných volně spolupracujících procesů. Tento model umožňuje postupné budování a rozšiřování aplikace bez zasahování do kompletní aplikace. Vývoj je tak lépe škálovatelný a nevyžaduje úpravy celého monolitického systému pro integraci nové součásti.

Dále v textu jsou převzaty obrázky z dokumentací konkrétních frameworků.

## Thingsboard

Thingsboard [3] poskytuje jak informační systém pro správu a řízení, tak i framework ve formě gateway aplikace, která umožňuje především kolekci dat z technologií třetích stran a jejich předání pro zpracování v informačním systému.

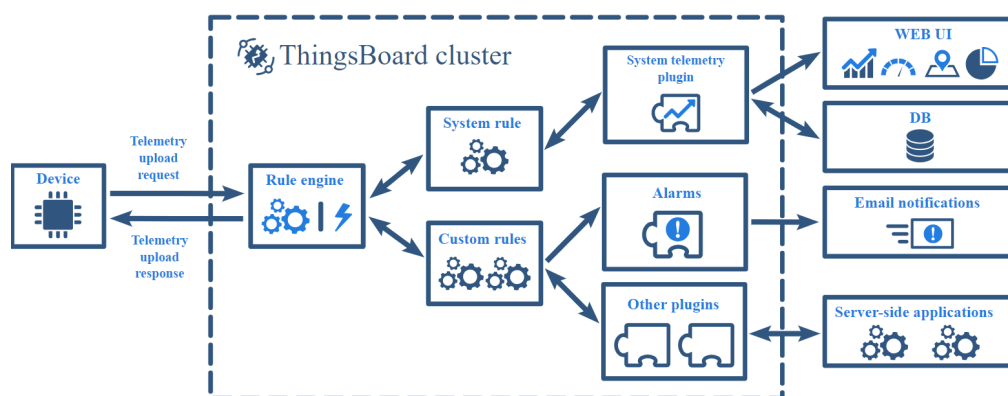
Toto řešení bylo vybráno pro demonstraci a bude tedy dále popsáno podrobněji. A popsané principy jsou poplatné i u dalších popsaných nástrojů.

Ohledně nasazení tohoto řešení. Je možná manuální instalace nebo instalace pomocí dockeru<sup>1</sup>.

### Sběr dat

Pro komunikaci na aplikační vrstvě je možné použít MQTT, CoAP<sup>2</sup> nebo HTTP protokolu. Komunikace přes MQTT, která je předmětem demonstrace používá JSON formátu pro zasílání zpráv :

- a) Přímým předáním telemetrie
- b) Přes gateway



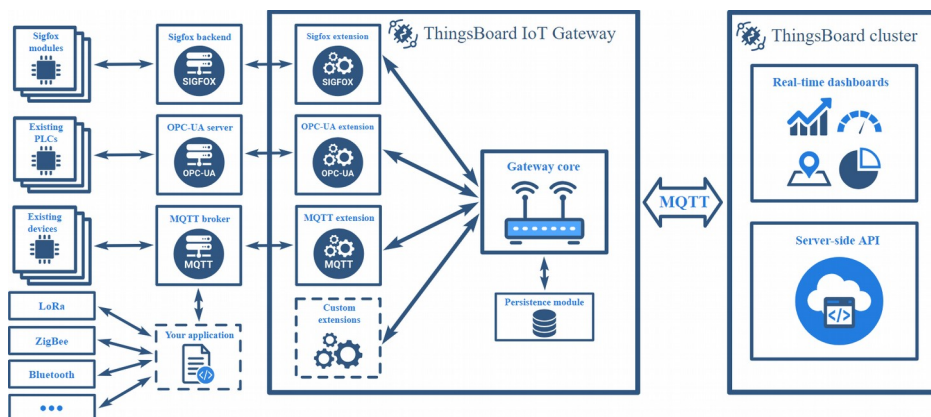
*Obrázek 1: Předání telemetrie přímo instanci*

Ve firmwaru IoT zařízení, je pro komunikace nutné integrovat MQTT klienta a callback funkcím k telemetrie requestům, které jsou taktéž ve formátu JSON. Data vstupují v aplikaci do Rule Engine , který slouží jako flow-control logika pro vstupní data a na základě analýzy generuje odpovědi ve formě requestů, emailů, alarmů. Telemetrie je ukládána do DB a je možné je vizualizovat pomocí dostupných widgetů ve formě budíků, spínačů, grafů, apod..

Druhým způsob je určen pro integraci technologií třetích stran jako například použití externího MQTT brokera. Gateway je poskytována jako instalační balíček pro Linux, Windows nebo Raspberry Pi. Tyto informace aplikace zasílá informačnímu systému pomocí MQTT. Gateway umožňuje integraci dalších komunikačních protokolů ZigBee, LoRaWan apod..

<sup>1</sup><https://www.docker.com/>

<sup>2</sup> <http://coap.technology/>

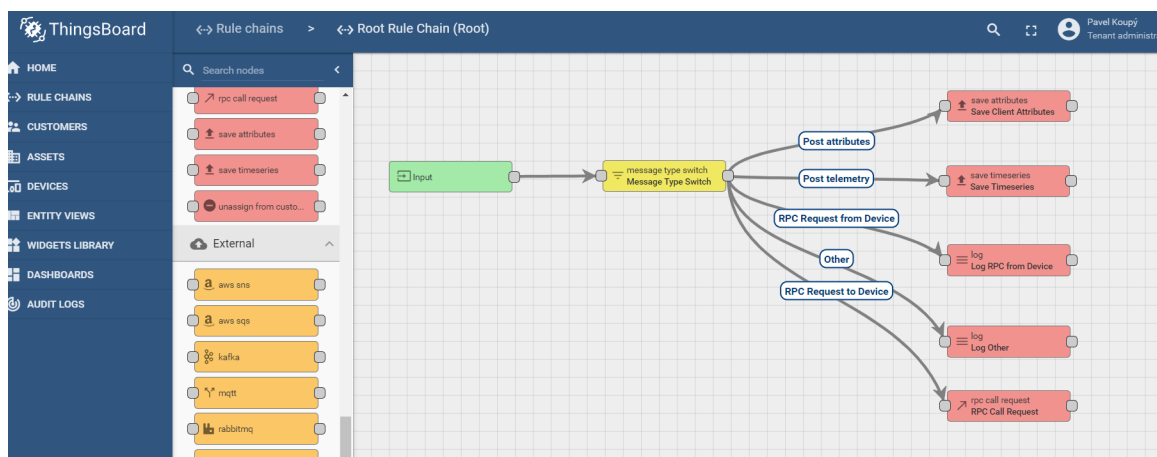


Obrázek 2: Předání dat přes gateway

## Uživatelské rozhraní

Aplikace informačního systému umožňuje zařídění jednotlivých zařízení do větších celků - budov, a tyto celky dále přiřazovat místům a jednotlivým klientům. Z kvalitativního hlediska je tato hierarchie třídění zařízení příjemná a blízká realitě.

Systém umožňuje nastavit kontrolní logiku tzv. Rule Chains pomocí grafového rozhraní, které řídí chování v případě příchozí zprávy. Základní nastavení obsahuje switch pro čtyři druhy žádostí. Specificky pro zasílání telemetrie a volání vzdálených procedur (callback<sup>3</sup>).



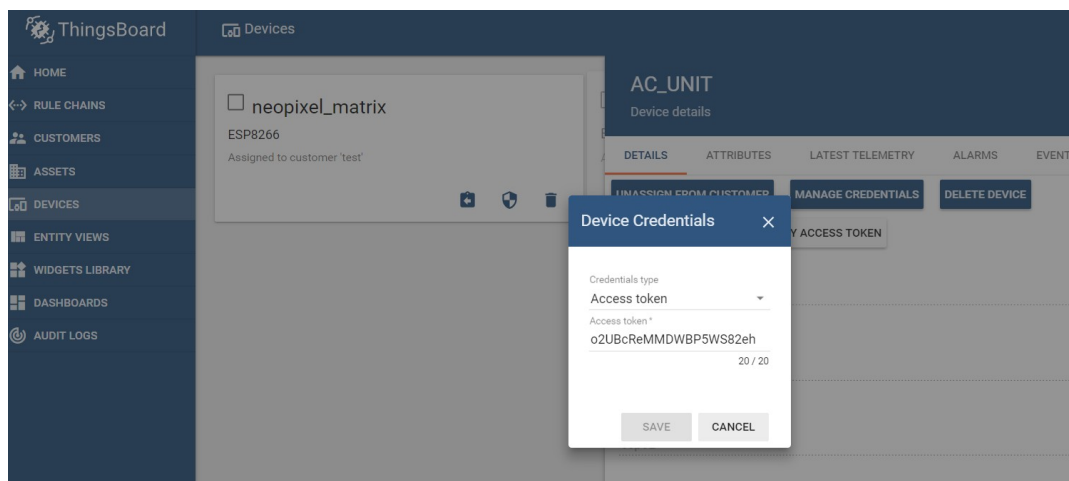
Obrázek 3: Rule Chains

Systém poskytuje multitenancy<sup>4</sup> funkcionalitu. Jednu instanci aplikace sdílí několik oddělených korporátních jednotek, firem nebo institucí. Je tedy třeba oddělit tyto skupiny, případně jejich správce. V systému má nejvyšší práva superadmin, který může vytvářet správce korp. jednotek. Tito správci mohou přidávat zařízení sdružovat zařízení, klienty, vytvářet dashboardy apod.. Klienti pak v systému mají jen roli, která umožňuje vyčítání dashboardů a zařízení klienta.

3 <https://thingsboard.io/docs/user-guide/rpc/>

4 <https://en.wikipedia.org/wiki/Multitenancy>

Zařízení jsou identifikována pomocí access tokenu nebo X.59 certifikátu. Každému zařízení je možné nastavit alarmy tedy určité reakce na události, sledovat poslední došlou telemetrii a atributy zařízení.



Obrázek 4: Zařízení - access token

## Vizualizace

Vizualizace telemetrie a atributů zařízení je možná pomocí přednastavených widgetů, na které je možné jednotlivé informace namapovat a zařízení tak monitorovat nebo ovládat.

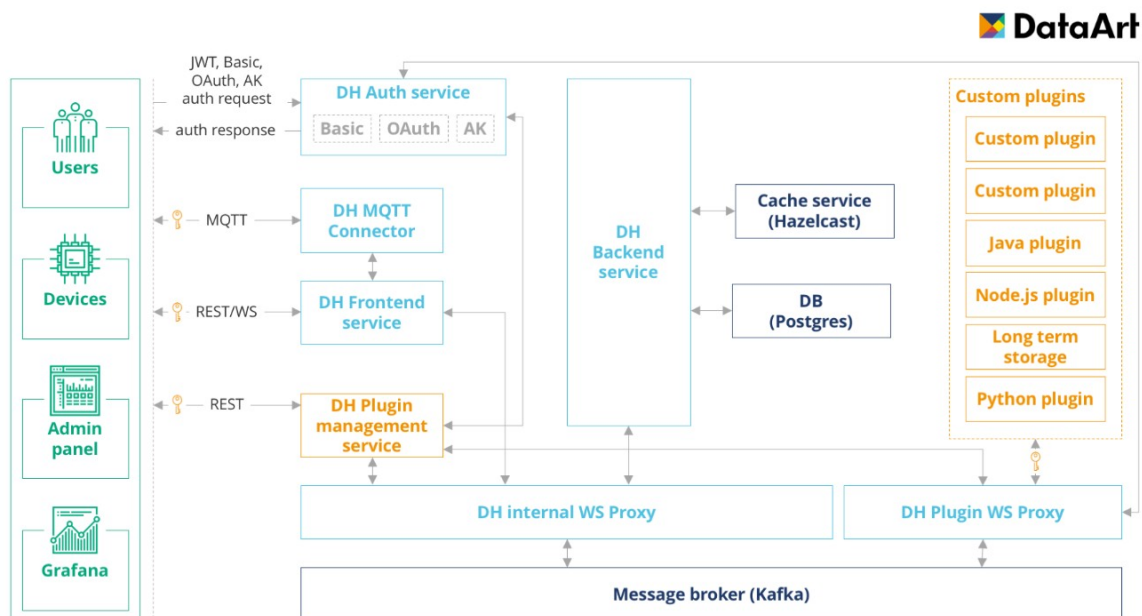


Obrázek 5: Dostupné widgety pro vizualizaci

V základu obsahuje aplikace různé druhy budíků, kontrolních prvků, GPIO ovladačů nebo také grafů a map. Tyto prvky je možné sjednocovat do dashboardů. Je možné vytvářet i vlastní widgety a vše opět z kvalitativního pohledu dobrou možností úprav i na úrovni Javascriptu.

Ke konkrétní implementaci v demu, instalace proběhla pomocí instalátoru a jako prerekvizitu vyžadovala instalaci DB serveru v tomto případě PostgreSQL.

## DeviceHive



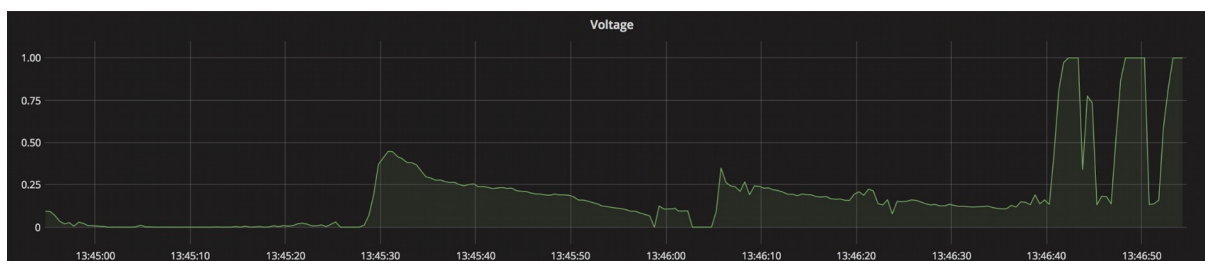
Obrázek 6: Architektura DeviceHive

Systém založený na microservice architektuře s podporou pluginů. Pluginy jsou spravovány pomocí Swagger<sup>5</sup>. Komunikace probíhá za pomoci zpráv ve formátu JSON konkrétně JSON Web Tokens<sup>6</sup>. Tento framework neposkytuje middleware jako Thingsboard. Využívá též PostgreSQL databázi.

Specificky pro tento projekt pro sběr dat a připojení ESP8266 do DeviceHive cloudu existuje firmware od vývojářů<sup>7</sup>. Pro přenos zpráv slouží WebScket Kafka Proxy microservice.

## Vizualizace

Vizualizace pomocí pluginu Grafana.



Obrázek 7: Příklad grafu generovaného pluginem Grafana

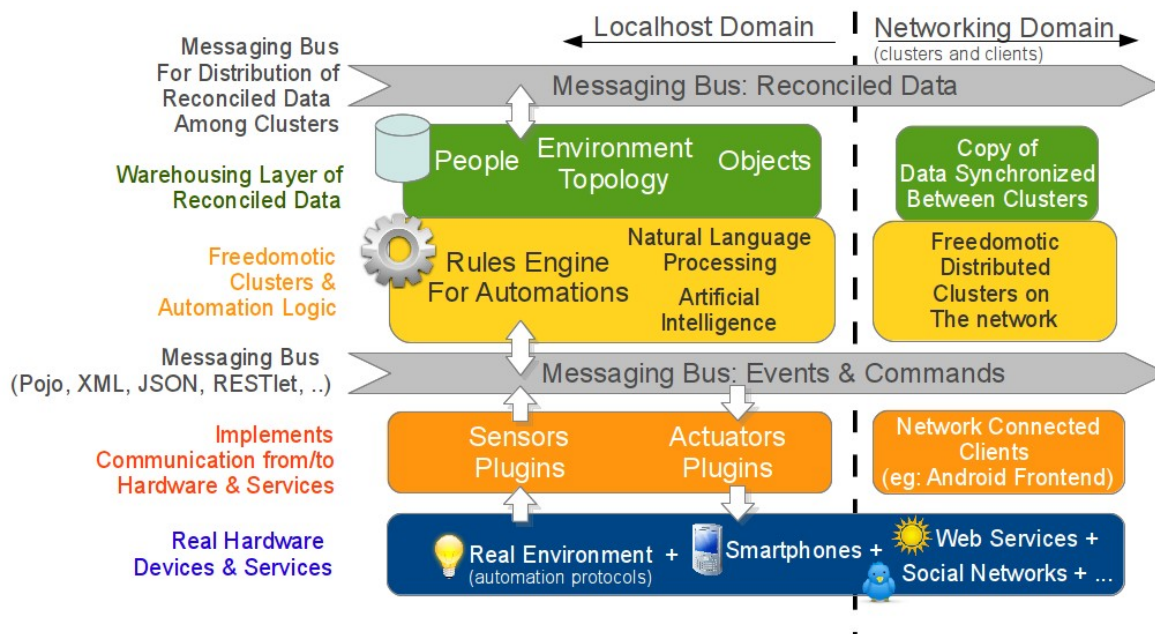
5 <https://swagger.io>

6 <https://jwt.io/>

7 <https://github.com/devicehive/esp8266-firmware>

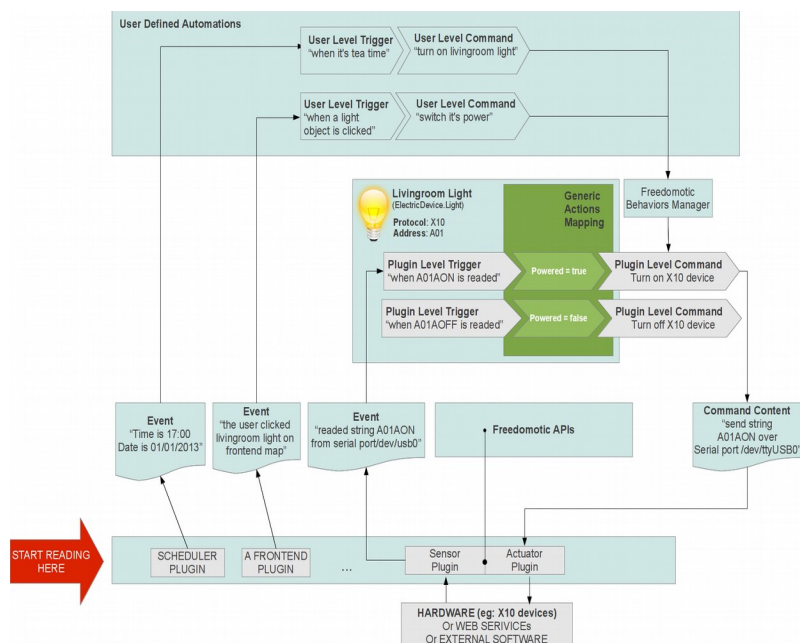
## Freedomotic

Zajímavý zástupce IoT frameworků zaměřených na vysoko úroňové příkazy v přirozeném jazyce. Framework využívá mapy prostředí objektů a lidí v něm. Zprávy v systému jsou typu „zapni světlo v kuchyni“ tyto zprávy zpracovává procesor přirozeného jazyka. Systém je ve fázi pokročilé beta verze.



Obrázek 8: Architektura Freedomotic

Pravidla chování systému spolu s procesorem přirozeného jazyka umožňuje vytvářet automatizaci za pomoci vět přirozeného jazyka např. „Pokud je venku tma, zapni světlo v pokoji“. V následujícím obrázku je znázorněna interakce mezi eventy, pluginy, triggerem a dalšími součástmi systému. Device pluginy umožňují rozšiřovat o nové technologie jako ThingSpeak, MQTT broker/klienta, mailového agenta apod. Grafický front-end je třeba jako DeviceHive dodat od třetí strany.

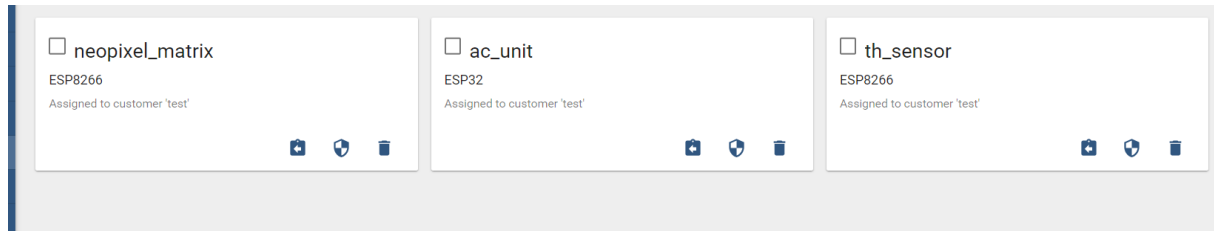


Obrázek 9: Spolupráce systémových komponent



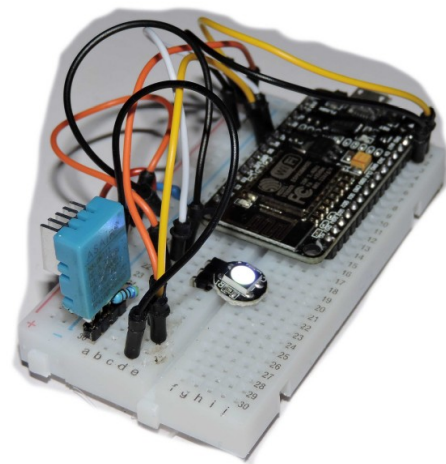
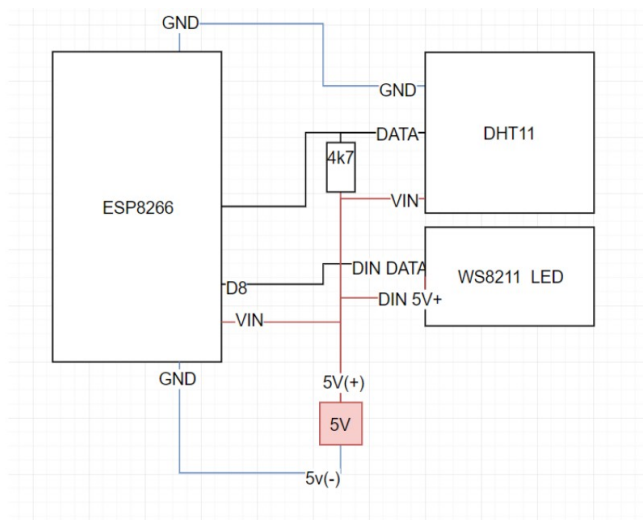
## Demostrace – Thingsboard

Pro demonstraci jednoho z frameworků jsem vyrobil několik obvodů zastupujících senzory a akční prvky založené na SoC ESP32 a ESP8266.



Obrázek 10: Zařízení - WEB UI

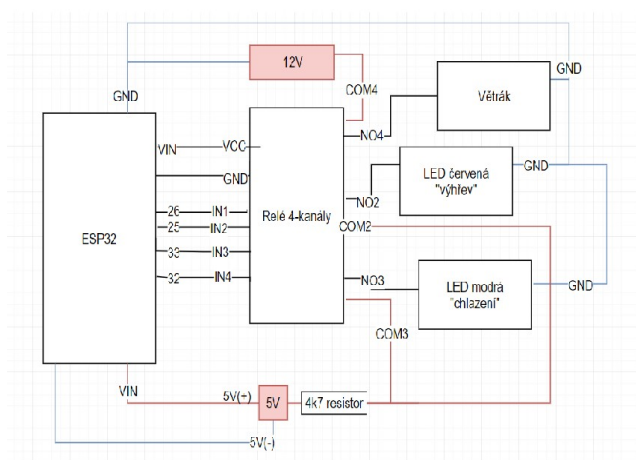
## Simulace - teploměru/vlhkoměru



Obrázek 11: Zapojení a fotografie obvodu DHT11

Pro demonstraci senzoru je použit DHT11, který je připojen k ESP8266. Firmware pro zasílání teploty se skládá z inicializace DHT11, PubSubClienta pro komunikaci přes MQTT. Každých 1,5 sekundy je snímána teplota a vlhkost a je zasílána do topicu "v1/devices/me/telemetry". Spolu s příslušným access tokenem tvoří jedinečnou identifikaci a všechny zařízení tak publikují do stejného topicu. Je zde použit LED modul WS2811 ale jen jako známka připojení k Wi-Fi a MQTT brokerovi.

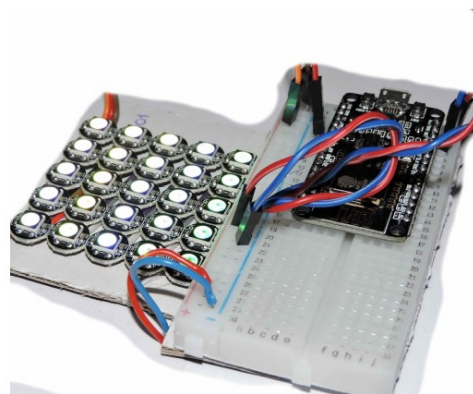
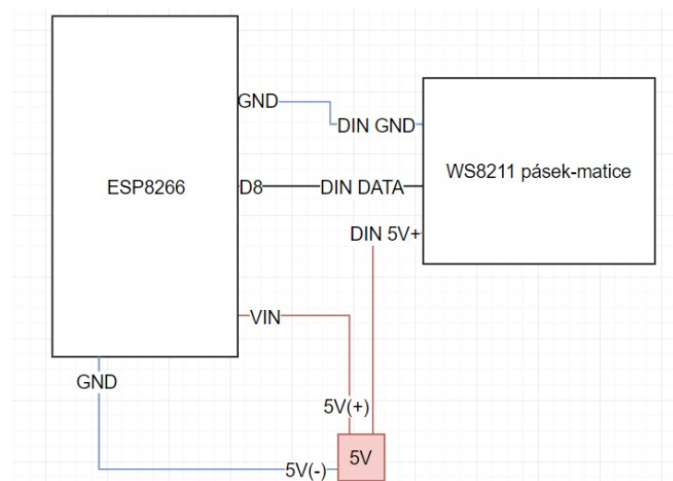
## Simulace - klimatizační jednotka



Obrázek 12: Zapojení a fotografie klim. jednotky

Demonstrace akčního prvku za pomoci ESP32 a 4 kanálového relé. Firmware se skládá callback funkce pro subscribe topiců na protokolu MQTT. Demo nemá žádné možnosti chladit nebo vyhřívát, tuto funkci simuluje červené a modré LED podsvícení větráku.

## Simulace - LED světlo/pásek



Obrázek 13: Zapojení a fotografie LED matice

Demonstrace ovládání LED pásku s moduly WS8211. Používá jednoduchý ThingsBoard wrapper nad knihovnou PubSubClient. Led pásek je ovládán pomocí PWM signálu knihovnou FastLed. Přepínání pomocí volby programu na otočném knoflíku způsobuje zasílání čísla RPC callbacku ve firmwaru ESP8266, které si zapisuje toto číslo do globální proměnné a použito jako index mění vestavěné palety barev.



## Ovládací panel



Obrázek 14: Raspberry Pi s dotykovým displejem

Raspberry Pi v kombinaci s dotykovým displejem poskytuje ovládání a demonstraci jednotlivých prvků dashboardu a poskytuje i dostatečnou výpočetní sílu pro zařízení iot-gateway zařízení pro sběr dat pomocí technologií a nástrojů třetích stran, nicméně není implementováno. Použitý operační systém je pro touchscreen upravená verze Raspbianu Stretch. Jednotlivé prvky dashboardu je možné zvětšit z webového rozhraní



Obrázek 15: Výsledný dashboard

## Závěr

V průběhu práce nenastalo žádné větší odchýlení od abstraktu. Jednou změnou v průběhu byl hostitel ThingsBoard aplikace. Původně měla běžet aplikace na Raspberry Pi, ale nakonec je použito systému s OS Windows. Demo je funkční, průzkum vyžaduje revizi, ale bylo vybráno zajímavých zástupců frameworku s middlewarem v podobě informačního systému a grafického programování, čistého IoT frameworku a zajímavého frameworku se zpracováním přirozeného jazyka. Skript INSTALL není součástí, protože mnohé části je třeba udělat za pomoci grafického rozhraní a firmware pro ESP moduly není plně automatizovaný. Skript RUN spouští Thingsboard aplikaci na systému Windows. Skript COMPILE také není součástí, protože kompilace je nutná pouze v případě firmware pro ESP moduly.

## Zdroje

- [1] Pavel Koupý , Inteligentní senzory 2018/2019: Senzor pro měření hladiny vody
- [2] <https://freedomotic-user-manual.readthedocs.io>
- [3] <https://thingsboard.io>
- [4] <https://docs.devicehive.com/>