Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Systém pro bezpečnou aktualizaci firmware v MQTT zařízeních

Bakalářská práce

Autor: David Česák
Studijní obor: Aplikovaná Informatika

Vedoucí práce: Mgr. Daniela Ponce, Ph.D.

Hradec Králové Duben 2022

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 29.4.2022 David Česák

Poděkování:

Děkuji vedoucí bakalářské práce Mgr. Daniele Ponce, Ph.D. za metodické vedení práce.

Anotace

Tato bakalářská práce popisuje návrh a vývoj systému pro bezpečnou aktualizaci firmware v MQTT zařízeních v rámci sítě Internet. Bezpečnou aktualizací se rozumí proces, ve kterém je zajištěno, že firmware nelze neoprávněně podvrhnout třetí stranou. Bezpečně aktualizovaná MQTT zařízení slouží k měření různých veličin prostředí (např. teplota, vlhkost). Součástí systému je aplikace ukládající naměřená data do databáze. V teoretické části jsou popsány vybrané použité technologie a v praktické části je systém navržen, zkonstruován a otestován. Nejdříve jsou stanoveny požadavky, následně jsou charakterizovány dílčí prvky dle požadavků. Z uvedených dílčích prvků je sestaven celý systém, který je odolný vůči podvržení a zároveň je uživatelsky přívětivý. Následně je demonstrována funkčnost tohoto systému a v závěru jsou zhodnoceny výsledky a navržena možná vylepšení.

Annotation

Title: System for secure firmware update in MQTT devices

This bachelor thesis describes the design and development of a system for secure firmware updates in MQTT devices over the Internet. A secure update is defined as a process in which the firmware cannot be tampered with by a third party. Securely updated MQTT devices are used to measure various environmental variables (e.g., temperature, humidity). The system includes an application that stores the measured data in a database. In the theoretical part, the selected technologies used are described and in the practical part the system is designed, constructed, and then tested. First the requirements are determined, then the sub-elements are characterized according to the requirements. From these sub-elements a complete system is assembled that is both tamper-resistant and user-friendly. The functionality of this system is demonstrated and finally the results are evaluated, and possible improvements are proposed.

Obsah

1 Úvod

V rámci sítě Internet probíhají neustále kybernetické útoky na zařízení. [5] Otázka bezpečnosti těchto systémů není jen teoretická rovina, ale vyžaduje si ji přímo praxe. Proti těmto rizikům je nutné zařízení chránit, aby útoky na zařízení nebyly možné. Jedním druhem útoků může být neoprávněná vzdálená aktualizace zařízení, kterou tato práce řeší.

Bezpečnou aktualizaci firmware lze demonstrovat na následujícím příkladu. Existují zařízení rozmístěná po celé České republice, která měří teplotu a vlhkost ovzduší. Naměřené hodnoty se následně odesílají na server. Při zjištění chyby ve firmware by bylo nutné provést zásah v každém z těchto zařízení. To by však bylo logisticky velmi náročné a finančně nákladné. V této situaci se nabízí možnost vzdálené aktualizace, která by tuto nevýhodu odstranila. Nebezpečím vzdálené aktualizace je však riziko, že aktualizaci firmware provede neoprávněná osoba, která může firmware pozměnit (podvrhnout). Pro zamezení podvržení firmware je nutné implementovat společně několik různých technik. Tyto techniky spolu s funkčním řešením jsou popsány v této práci.

Obecně popsané technologie jsou rozvedeny v kapitole 4. Návrh a realizace systému jsou popsány v kapitolách 5 až 8. Funkčnost systému je následně demonstrována v kapitole 9.

Celý systém je složen z jednoho serveru a několika měřicích zařízeních. Server je umístěn v síti Internet a je dostupný odkudkoliv. Měřicí zařízení jsou malá přenosná zařízení, pracovně nazvaná CesDev1, která slouží ke sběru reálných dat ze senzorů. Tato zařízení se připojují k serveru za účelem provedení dvou akcí. Mezi tyto akce patří odesílání dat naměřených veličin ze senzorů, aktualizace vlastního firmware. Zařízení je složeno z firmware a hardware, kterým je zejména mikropočítač pro komunikaci přes Wi-Fi. Firmwarem se rozumí software v zařízení CesDev1, který je předmětem aktualizace.

2 Cíl práce

Cílem práce je návrh a realizace systému, který umožní bezpečnou aktualizaci firmware v zařízeních, která periodicky sbírají data a odesílají je prostřednictvím protokolu MQTT. Tato data budou dále zpracovávána a ukládána do databáze. Bezpečnou aktualizací se rozumí proces, při kterém nedojde k nahrání nevalidního (podvrženého) firmware.

3 Metodika zpracování

Metodika této práce se skládá z identifikace požadavků, následné charakterizace dílčích prvků, návrhu systému, jeho realizace a ověření funkčnosti za pomocí demonstrace.

4 Vybrané použité technologie

Tato kapitola obsahuje teoretický rozbor vybraných, použitých technologií, konkrétně se jedná o MQTT, hašování a šifrování. MQTT je důležité z toho důvodu, že se jedná o protokol, který se používá u zařízeních, která budou vzdáleně aktualizována. Hašování a šifrování. Ty jsou nutné k tomu, aby bylo možné zajistit ověření autenticity nového firmware, který je nutný k dosažení cíle této práce.

4.1 Message Query Transport Telemetry (MQTT)

Protokol MQTT (Message Query Transport Telemetry) využívá TCP spojení pro přenos jednoduchých dat v reálném čase. Tento protokol je zaveden jako standard v konsorciu OASIS. [1] MQTT je navržen jako jednoduchý, otevřený, a hlavně co nejméně náročný na implementaci. Díky těmto vlastnostem je vhodný pro různá IoT řešení včetně toho, který se používá v této práci.

Řešení využívající MQTT protokol fungují na architektuře klient-server. Jedná se o připojení klientů k jednomu centrálnímu serveru, kterému odesílají veškeré své požadavky. Serveru se v rámci MQTT říká broker. V MQTT se pro přenesená data používá termín zpráva. Zprávu nelze odeslat brokeru pouze s daty, ale je nutné znát i cíl. V MQTT je cíl nazýván kanálem a jedná se o jednoznačně identifikovatelný textový řetězec. Do každého takového kanálu je pak možné posílat různorodá data. Kanál také může mít různé úrovně, je tedy možné logicky vrstvit odesílané zprávy. Dá se to přirovnat k objektům a jeho vlastnostem (atributům), které umějí být znovu vnořené. Příkladem takového kanálu může být *esp_1/sht30/temperature.* V tomto kanálu jsou vidět tři úrovně. První úrovní je *esp_1*, což je v tomto případě identifikátor zařízení, ve druhé úrovni je *sht30*, což je senzor měřící několik veličin. Ve třetí a nejnižší úrovni je *temperature*, což v tomto případě značí naměřenou teplotu daného čidla.

Kanály mají také některá omezení. Nejdůležitější z nich jsou například:

- senzitivita na velikost písmen, (Banks a Gupta 2014)
- omezení minimální velikosti kanálu na 1 znak,
- omezení maximální velikosti kanálu na 65 535 bajtů.

Kompletní výčet lze nalézt v dokumentaci samotného MQTT protokolu. [1]
MQTT protokol funguje na principu zveřejni-odebírej (anglicky publish-subscribe). Pro klientskou stranu to znamená, že dostává pouze zprávy, které požaduje. K tomu, aby mohl klient začít přijímat zprávy, musí nejdříve brokeru zaslat zprávu o tom, které kanály bude odebírat a odkud mu budou zprávy posílány. Pro zjednodušení odebírání různých kanálů, existují v MQTT i takzvané zástupné znaky (wildcards). Tyto zástupné znaky fungují v podstatě jako běžné filtry v různých operačních systémech pro vyhledávání souborů (např. znak * jako libovolný text). V MQTT jsou dva druhy. [1]

- jednoúrovňový,
- víceúrovňový.

Jednoúrovňový zástupný znak se v názvu kanálu značí znakem plus (+). Jak již z názvu vyplývá, funguje pouze v jedné úrovni. Jako příklad lze uvést *esp_1/sht30/+*. V tomto případě klient, který odebírá tento kanál, bude dostávat všechny zprávy na této úrovni. Příkladem může být *esp_1/sht30/temperature* nebo *esp_1/sht30/humidity*. Ale například zprávy z kanálu *esp_1/sht30/sub/test* už nedostane.
Víceúrovňový zástupný znak je značen mřížkou (#). Funguje obdobně, jako jednoúrovňový, s tím rozdílem, že funguje i na všechny úrovně pod ním. V případě předchozího příkladu, u kterého upravíme poslouchání kanálů na *esp_1/sht30/#*, dostaneme všechny zprávy jako předtím, a to i včetně *esp_1/sht30/sub/test*.

4.2 Hašování

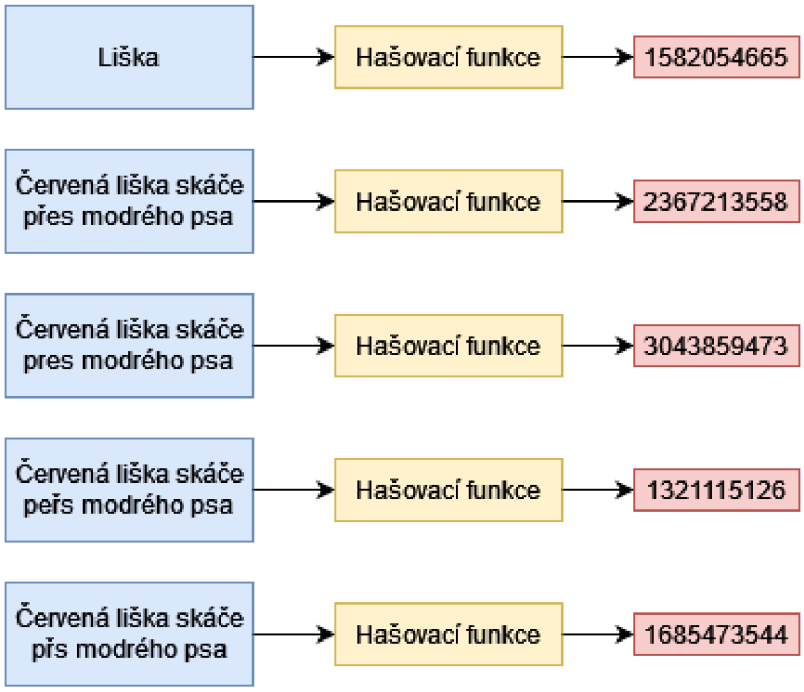
Hašování je předpis, jak přeměnit různorodá data (bity) na jinou reprezentaci tak, aby nebylo možné získat původní data zpět. Algoritmy k tomu určené se nazývají hašovací funkce. Navrženy jsou tak, aby při stejných vstupech generovaly stále stejné výstupy. Ale zároveň aby při rozdílném vstupu i o jeden bit byl výstup velmi rozdílný. Tedy aby nebylo jednoduché uhodnout původní vstup pomocí opatrné manipulace vstupních dat.
Výstup hašovacích funkcí má vždy fixní délku, tato délka závisí na využité funkci. Vzhledem k fixním délkám zde existuje omezené množství výstupů, díky čemuž se

vyskytuje možnost, že dva rozdílné vstupy generují stejný výstup. Takovéto výjimce se říká kolize. Tento jev je velice vzácný, ale možný u funkcí s malým výstupním blokem, nebo špatně navržených hašovacích funkcí.

Vlastností těchto hašovacích funkcí se využívá například k výpočtu kontrolního součtu pro soubory, ukládání hesel v databázi nebo jako klíč v hašovací tabulce.

Ačkoliv jsou hašovací funkce schopné zajistit integritu souboru, není možné ověřit jejich autentičnost (třetí strana může změnit obsah souboru a k tomu jednoduše vypočítá novou hodnotu hašovací funkce). K tomuto účelu je nutné hašovací funkce komplementovat s funkcemi šifrovacími, kde už není tak jednoduché upravit výsledný haš.

Mezi známé hašovací funkce patří například MD5, SHA-1, SHA-2 (SHA-256), Argon2.



Obrázek 1. Vizualizace hašovací funkce (zdroj: [4])

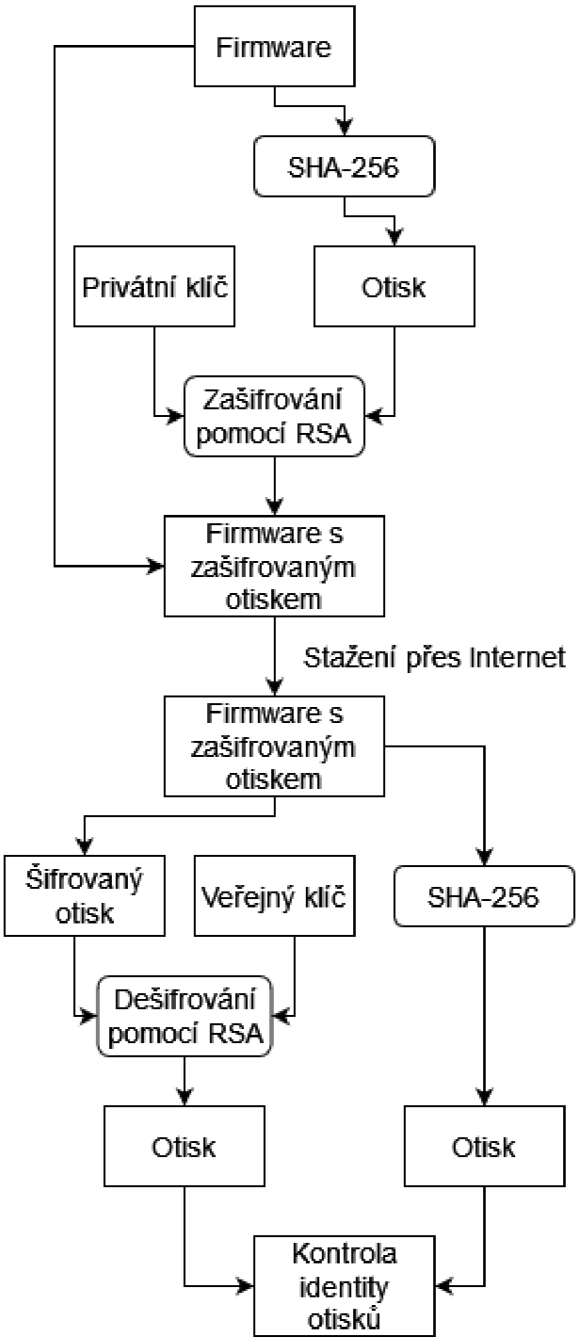
4.3 Šifrování

Šifrování je postup, jak upravit vstupní data do nečitelné formy a zároveň aby je bylo možné vrátit do původní podoby. Každá šifra má nějaké tajemství, pomocí kterého data zašifruje. Zároveň má stejné nebo odlišené tajemství pro dešifrování dat. Tento způsob transformace dat se využívá v případě, že nechceme, aby někdo přečetl obsah zprávy bez našeho vědomí.

Dnešní moderní kryptografie se dělí na dva druhy. Symetrická a asymetrická. Symetrická funguje s jedním klíčem pro šifrování a dešifrování. Asymetrická funguje se dvěma klíči – jeden (veřejný) pro zašifrování a druhý (privátní) pro dešifrování. Výhodou asymetrické kryptografie je ta, že je možné sdílet veřejný klíč s kýmkoliv a není tím kompromitována bezpečnost zašifrovaného obsahu oproti kryptografii symetrické.

V případě využití asymetrické kryptografie pro podepisování souborů se role klíčů zamění. Z privátního klíče se stane klíč podepisovací a z klíče veřejného se stane klíč ověřovací. V případě, že máme data, která chceme přenést a nevadí nám, že si je bude moci třetí strana přečíst a zároveň chceme zajistit autentičnost, dá se využít jednoduchá kombinace s použitím hašovacích funkcí.

Postup pro ověření autentičnosti je jednoduchý. Stačí vypočítat haš souboru u kterého chceme zajistit autentičnost. Poté využijeme privátní klíč, pomocí kterého podepíšeme haš. Tento haš se následně připojí k souboru na jeho konec. Následně se takto upravený soubor odešle a druhá strana si ho může pomocí veřejného klíče poskytnutého od podepisovací entity ověřit.



Obrázek 2. Bezpečný postup aktualizace a ověření firmware (zdroj: [8])

5 Identifikace požadavků pro systém měření

Požadavky jsou rozděleny do dvou částí. V první části jsou požadavky na samotné měřicí zařízení a ve druhé na vlastní server (cloud). Každý požadavek má své unikátní značení ve tvaru *REQ-ZAŘÍZENÍ-ČÍSLO*.

5.1 Požadavky na měřicí zařízení

Jako povinné požadavky jsou vyhodnoceny: připojení do vlastního cloudu (REQ-IOT-11) a do sítě Internet (REQ-IOT-21). Tyto požadavky jsou tak klíčové, že bez nich by nebylo možné dosáhnout cíle této práce. Dalším povinným požadavkem je schopnost měřit veličiny (REQ-IOT-31). Tato podmínka je zde čistě z demonstračního hlediska, aby bylo možné prokázat, že se odesílají reálná data. Programovatelný procesor je také povinný požadavek (REQ-IOT-41), protože to zajistí modularitu a případnou možnost zařízení vylepšit v budoucnu. Posledním povinným požadavkem je nízká pořizovací cena. Tato cena byla porovnána se zařízeními ESP8266, ESP32, Raspberry Pi 4 (REQ-IOT-51). Cenově nejvýhodněji vychází platforma ESP8266. Cenové porovnání lze nalézt v příloze této práce.
Z hlediska bezpečné aktualizace je zde několik dalších povinných požadavků. První (REQ-IOT-61) z nich je, že musí umět nahrát, a tedy i přepsat svůj vlastní firmware. Dalším požadavkem (REQ-IOT-71) je ten, že je nutné, aby zařízení bylo schopné rozeznat, zdali nově stáhnutý firmware je opravdu nahrán osobou s oprávněním. V případě, že by to zařízení neumělo splnit, cíl bakalářské práce by byl taktéž nesplnitelný a vynaložená námaha by byla naprosto zbytečná. Pro zaručení bezpečné aktualizace patří další požadavek (REQ-IOT-81). Ten říká, že zařízení musí umět pracovat s asymetrickou kryptografií. Díky tomuto požadavku je možné velice bezpečně ověřovat validitu nového firmwaru. Další požadavek k bezpečné aktualizaci říká, že je nutné, aby zařízení umělo hašovat pomocí bezpečné funkce. (REQ-IOT-91) Důvod pro tento požadavek je ten, že zařízení je schopné ověřit otisk nového firmware. Ačkoliv haš nelze brát za kryptografické ověření, tak díky němu se dá zjistit, jestli byl firmware při stahování poškozen. Spolu s asymetrickou kryptografií je to velice účinný způsob, jak omezit a v ideálním případě i zamezit podvrhnutí firmware.

Dále jsou zde doporučené požadavky. Mezi ně patří velikost a hmotnost zařízení (REQ-IOT-101), (REQ-IOT-111). Tyto vlastnosti nejsou klíčové, ale jelikož se jedná o IoT zařízení měřicí veličiny, bylo by dobré tyto vlastnosti udržet na co nejmenší hodnotě tak, aby bylo zařízení co nejvíce přenosné. Poslední doporučenou vlastností je napájení pomocí USB konektoru. To je zde uvedeno, protože USB konektor je dnes masově rozšířen a díky tomu by nebyl problém napájet CesDev1 zařízení téměř kdekoliv (REQ-IOT-121).

V tabulce č. 1 je uveden seznam všech požadavků na měřicí zařízení, kde sloupeček Typ označuje, jestli se jedná o povinný nebo doporučený požadavek.

Tabulka 1. Požadavky na měřicí zařízení CesDev1 (zdroj: autor)

ID požadavku	Text požadavku	Typ (P = povinný) (D = doporučený)
REQ-IOT-11	Zařízení musí umět posílat data do vlastního cloudu, který je pod kontrolou uživatele.	P
REQ-IOT-21	Zařízení se musí umět připojit k síti Internet.	P
REQ-IOT-31	Zařízení musí umět měřit teplotu a vlhkost.	P
REQ-IOT-41	Zařízení musí obsahovat programovatelný mikropočítač.	P
REQ-IOT-51	Zařízení musí mít nízkou pořizovací cenu.	P
REQ-IOT-61	Zařízení musí umět nahrát nový firmware.	P
REQ-IOT-71	Zařízení musí umět ověřit validitu nového firmware.	P
REQ-IOT-81	Zařízení musí umět pracovat s asymetrickou kryptografií.	P
REQ-IOT-91	Zařízení musí umět vypočítat libovolnou bezpečnou hašovací funkci.	P
REQ-IOT-101	Velikost zařízení musí být do 200 cm^3.	D
REQ-IOT-111	Hmotnost zařízení musí být do 150 g.	D
REQ-IOT-121	Napájení zařízení musí být pomocí USB konektoru.	D

5.2 Požadavky na vlastní server (cloud)

Tyto požadavky jsou sestavené pro použití na operačním systému založeném na GNU/Linux (REQ-SRV-62). Důvodem je, že tyto operační systémy zpravidla bývají bez licenčních poplatků a zdarma ke stažení. Také jsou méně náročné na hardware oproti konkurenčnímu OS Windows. Tím je myšlena menší režie ze strany operačního systému. Mezi povinnými požadavky je veřejná IPv4 adresa (REQ-SRV-12), která umožnuje připojení zařízení bez podpory IPv6. Další povinné požadavky jsou hardwarové parametry samotného serveru (REQ-SRV-22 až REQ-SRV-52). Ty zajistí plynulý chod celého systému. Samotný systém může vyžadovat vyšší systémové parametry, zejména uložiště. Záleží na počtu zařízení, četnosti a množství ukládaných dat do cloudu.

Vzhledem k tomu, že se na cloud budou připojovat zařízení s MQTT, tak je nutné, aby na serveru běžel MQTT server, také nazývaný MQTT broker (REQ-SRV-72). MQTT broker ve výchozím režimu posílá zprávy bez šifrování podobně jako protokol HTTP. Toto není žádoucí a naštěstí MQTT broker podporuje šifrování pomocí asymetrické kryptografie. Tento požadavek je nutný, protože nechceme, aby kdokoli mohl odposlouchávat data přenášená mezi MQTT brokerem a klienty (REQ-SRV-82). Vzhledem k tomu, že MQTT broker neumí data ukládat pro zpětnou analýzu nebo pro zobrazení výstupu ve formě grafu, je nutné tento požadavek splnit jiným způsobem (REQ-SRV-92). Řešení tohoto požadavku je rozebráno dále v práci. Dalším požadavkem je webové rozhraní pro správu nahraných firmware (REQ-SRV-102). Důvodem zavedení tohoto požadavku je, aby bylo možné jednoduše aktualizovat firmware. Webové rozhraní umožní nahrávání z jakéhokoli zařízení, které má v sobě webový prohlížeč.

V tabulce č. 2 jsou požadavky na vlastní cloud. Sloupeček Typ označuje, jestli se jedná o povinný nebo doporučený požadavek.

Tabulka 2. Požadavky na vlastní cloud (zdroj: autor)

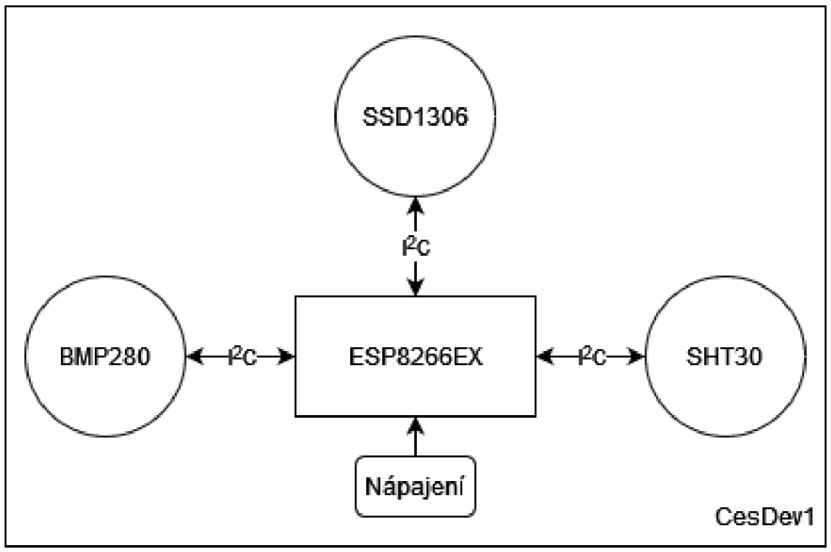
ID požadavku	Text požadavku	Typ (P = povinný) (D = doporučený)
REQ-SRV-12	Cloud musí mít veřejnou IPv4 adresu.	P
REQ-SRV-22	Cloud musí mít jedno či více jader s instrukční sadou AMD64 nebo x86-64	P
REQ-SRV-32	Rychlost připojení cloudu musí být 10 Mbit/s a více.	P
REQ-SRV-42	Cloud musí mít kapacitu uložiště nejméně 20 GiB.	P
REQ-SRV-52	Operační paměť cloudu musí být nejméně 2 GiB.	P
REQ-SRV-62	Operační systém cloudu musí být založený na GNU/Linux.	D
REQ-SRV-72	Cloud bude mít nainstalován MQTT broker.	P
REQ-SRV-82	MQTT broker bude zabezpečen.	P
REQ-SRV-92	Cloud bude ukládat data získaná z MQTT brokeru do uložiště pro možnost zobrazení historie.	P
REQ-SRV-102	Cloud bude mít webové rozhraní pro správu nahraných firmware.	P

6 Charakterizace dílčích prvků systému podle požadavků

Na základě dříve zjištěných požadavků se výsledný systém skládá ze dvou základních prvků: CesDev1 a vlastního cloudu. Samotné CesDev1 zařízení je sestaveno autorem této práce. Jedná se tedy o vlastní výrobek, který byl sestrojen pro potřeby firmy CESAK s.r.o.

6.1 Charakterizace CesDev1

Zařízení CesDev1 je možné charakterizovat do dvou částí: hardware a software. Hardware je na základě požadavků složen z několika částí, kde základní část tvoří procesorový modul. Ten zajišťuje komunikaci prostřednictvím sítě Internet. Další hardwarovou částí jsou dva senzory, které přenáší data prostřednictvím sběrnice procesorového modulu.



Obrázek 3. Blokové schéma CesDev1 (zdroj: autor)

6.1.1 Procesorový modul založený na platformě ESP8266

Platforma ESP8266, konkrétně modul ESP8266EX, je systém na čipu s integrovanou Wi-Fi podporou. Tento modul obsahuje 32bitový procesor Tensilica L106 s RISC architekturou. Jeho pracovní frekvenci je možné nastavit na 80 MHz nebo 160 MHz. [3]

Procesor využívá integrovanou paměť SRAM pro uložení dat po dobu napájení. K uložení dat bez napájení je nutné připojit externí FLASH paměť. Modul ESP8266 podporuje velikost až 16 MiB. [3] Tato externí paměť je součástí zakoupeného modulu.

Pro komunikaci s okolím je modul vybaven vstupními / výstupními piny, které může samotný procesor obsluhovat. Mezi podporované rozhraní patří 1x softwarově řešené I^2C, 2x UART, 2x SPI, 1x analogovo-digitální převodník a pulzně šířková modulace na výstupních špičkách. [3] Pro podporované rozhraní na konkrétním pinu je nutné si otevřít dokumentaci konkrétního modulu.

K bezdrátové komunikaci je modul vybaven Wi-Fi rozhraním s integrovanou anténou. Rozhraní Wi-Fi v modulu ESP8266EX umožnuje jeden ze tří módů [3]:

- Wi-Fi klient, ve kterém se modul ESP8266 připojuje na přístupový bod,
- Wi-Fi přístupový bod vytvořený modulem ESP8266, na který se připojují jiná koncová zařízení,
- Wi-Fi klient a přístupový bod, který kombinuje oba výše uvedené módy.

Modul ESP8266EX podporuje tři Wi-Fi standardy. Jedná se o standardy fungující v ISM pásmu 2,4 GHz. Podporované standardy jsou vyznačeny v tabulce 3.

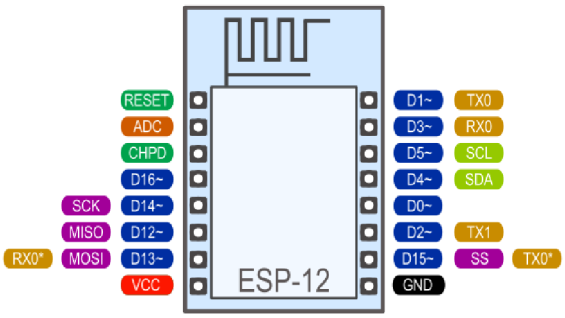
Tabulka 3. Podporované Wi-Fi standardy modulem ESP8266EX (zdroj: [3])

Standard	Frekvenční pásmo	Maximální rychlost
IEEE 802.11 b	2,4 GHz	11 Mbit/s
IEEE 802.11 g	2,4 GHz	54 Mbit/s
IEEE 802.11 n	2,4 GHz	600 Mbit/s

Firmware pro modul je možné navrhnout několika způsoby. Od tvůrců ESP8266 modulů Espressif jsou dvě možnosti. První je RTOS neboli Real Time Operating System (česky operační systém reálného času), druhou možností je využití ESP-IDF SDK pro programování bez operačního systému. Ačkoliv jsou tyto možnosti oficiální, tak neobsahují ideální prostředí pro vývoj z důvodu vysoké náročnosti na znalost konkrétního hardware. Obě tyto možnosti zahrnují programování v jazyce C.

Další, jednodušší možností je využití jazyka Python pro programování. Speciální verze pro vestavěná zařízení se nazývá MicroPython. Výhodou je velmi jednoduché programování, protože na zařízení se nahraje interpret tohoto jazyka a pak stačí psát kód v Pythonu. Ovšem velikou nevýhodou tohoto řešení je možnost si jeho zdrojový kód přečíst, jelikož tento jazyk není překládán do binární podoby. Další nevýhodou je rychlost běhu programu, jelikož interpretované jazyky jsou pomalejší než jazyky kompilované.

Třetí možností je využití platformy Arduino. Ačkoliv tato platforma v základu nepodporuje tento modul, existuje zde komunitní projekt, který tuto podporu přidává. Výhodou je velké množství knihoven pro různé senzory, aktuátory, displeje. Tyto knihovny lze ve většině případů využít i pro ESP8266 modul, ačkoliv původně byly vytvořeny pro jiná zařízení. V tomto prostředí se programuje v jazyce C nebo C++.



Obrázek 4. ESP8266EX modul v pouzdře ESP-12 (zdroj: [6])

6.1.2 Čidlo BMP280

BME280 je čidlo společnosti Bosch Sensortec. Toto čidlo je schopné měřit tři veličiny: teplota, vlhkost a atmosférický tlak. Čidlo bylo vybráno z důvodu nízkých pořizovacích nákladů a malých rozměrů. Jeho fyzické rozměry jsou: šířka a hloubka každá 2,5 mm a výška do 1 mm. Pro komunikaci s okolím je senzor vybaven sběrnicí I^2C.

Obrázek 5. Součástka BMP280 (zdroj: [2])

6.1.3 Čidlo SHT30

SHT30 je další čidlo umožňující měřit veličiny prostředí. Konkrétně se jedná o teplotu a vlhkost vzduchu. SHT30 je jedním z čidel řady SHT3x vyráběné společností Sensirion. Čidla z této řady mají identické rozhraní pro komunikace. Jsou vzájemně zaměnitelná a jediné, v čem se liší, je jejich možná chybovost naměřených hodnot. Vlastnostmi se jedná o podobné čidlo jako BME280. Jediným významným rozdílem je ten, že neumí měřit atmosférický tlak. Pro komunikaci s okolím také používá sběrnici I^2C.

Obrázek 6. Součástka SHT30 (zdroj: [7])

6.1.4 OLED displej

Ačkoliv v požadavcích není nikde nutnost žádné vizualizace, je v CesDev1 instalován i malý OLED displej s názvem SSD1306 o uhlopříčce 0,96“ a s rozlišením 128 x 64 pixelů. Důvod pro instalaci je znázornění demonstrace funkčnosti vzdálené aktualizace. Ta se projeví změnou vykresleného obsahu zobrazeného na displeji.

Obrázek 7. Čelní pohled na OLED displej (zdroj: autor)

6.2 Charakterizace vlastního cloudu

Na základě požadavků je vlastní cloud sestaven z jednoho serveru. Veřejnou IPv4 adresu je možné získat od poskytovatele internetového připojení, který však adresu nabízí za vyšší cenu než za kterou lze pořídit virtuální server včetně IPv4 adresy. Díky veřejné IP adrese je možné se na daný server připojit bez potřeby obcházet NAT a nastavovat přesměrování portů.

Samotný cloud bude poskytovat několik služeb:

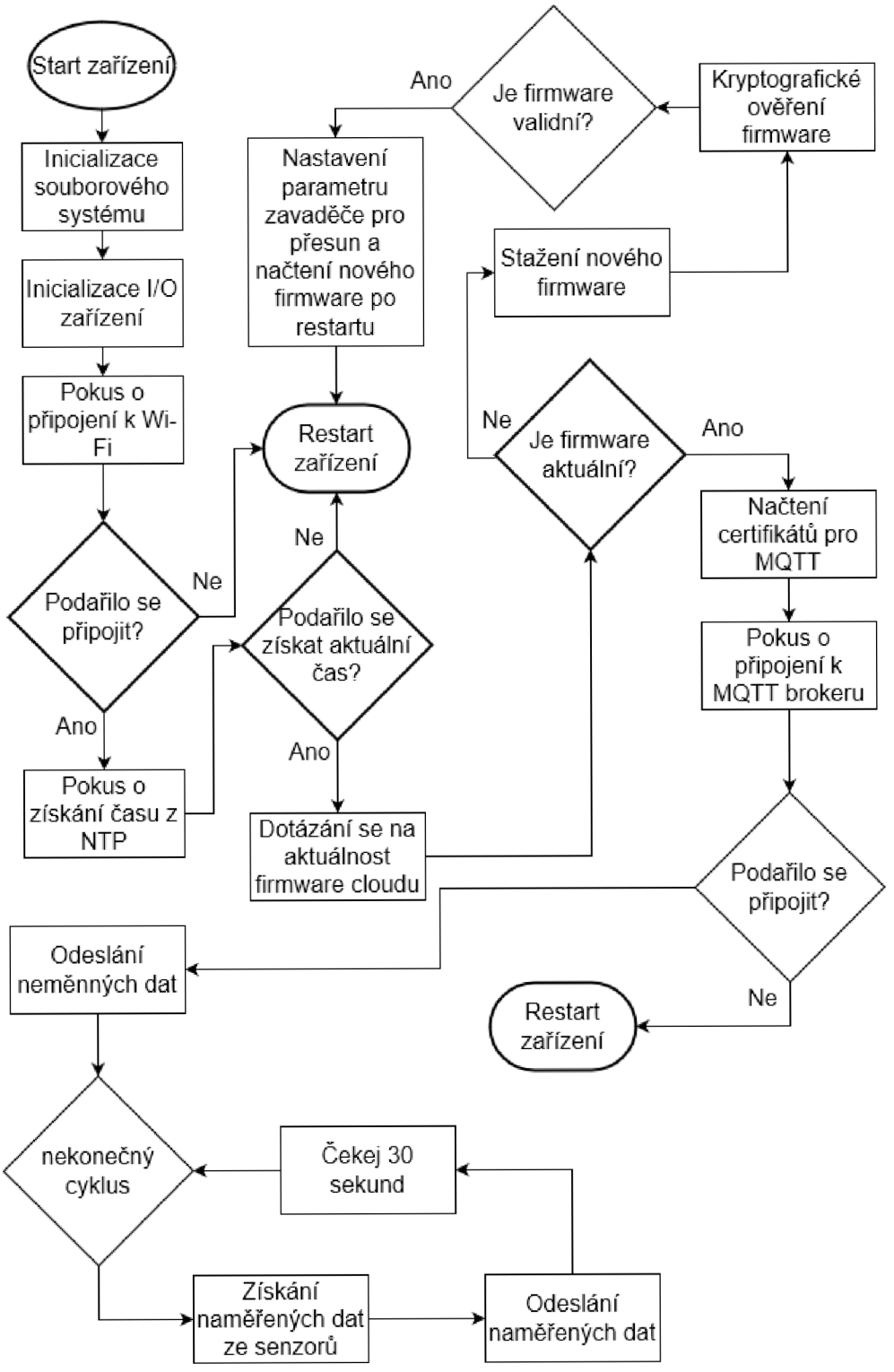
- MQTT broker,
- webový server,
- služba ukládající MQTT data do databáze,
- databázový server.

7 Návrh technického řešení

V této kapitole je návrh technického řešení dle dílčích prvků, které byly definované v předchozí kapitole, rozebrány do větších detailů. Zároveň je z těchto dílčích prvků navržen funkční celek s vhodně zvolenými parametry, který tvoří celý systém.

7.1 Základní parametry firmware

Programátor CesDev1 má zdrojový kód. Ten se pomocí kompilace převede do binární formy jednoho souboru. Jeho velikost se pohybuje od 100 KiB do 1019 KiB. Tento soubor lze použít a nahrát ho přímo do zařízení prostřednictvím nástroje *esptool*. Takové řešení ale není vhodné pro vzdálenou správu a je tedy nutné využít proces aktualizace na dálku. Výrobce nenabízí možnost aktualizovat firmware na dálku. Aktuálně vyvinutý firmware bude mít následující průběh znázorněn vývojovým diagramem níže.



Obrázek 8. Vývojový diagram průběhu programu v CesDev1 (zdroj: autor)

7.2 Rozložení obsahu Flash paměti

Flash paměť má kapacitu 4 MiB. Flash paměť se rozděluje na dvě části – část obrazová, ve které je uložen kód pro běh a část druhá, která slouží k ukládání podpůrných dat. V této části se nachází souborový systém, ze kterého lze libovolně číst a případně do něj i zapisovat. Na začátku paměti (oblast A) je vždy obraz, který je po spuštění zařízení postupně načítán a spouštěn. Tuto paměť lze rozdělit několika způsoby podle toho, jak veliký je požadován souborový systém. V tomto případě je zvolena konfigurace zapsaná v tabulce níže.

Tabulka 4. Rozdělení Flash paměti (zdroj: autor)

Oblast	Offset	Velikost	Význam
A	0 MiB	1 MiB	První slot pro kód aktuálního firmware
B	1 MiB	1 MiB	Nevyužité místo
C	2 MiB	1 MiB	Rezervované místo pro aktualizaci
D	3 MiB	1 MiB	Souborový systém

7.3 Dostupné postupy aktualizace firmware

7.3.1 Základní způsob aktualizace

Aby bylo možné firmware vzdáleně aktualizovat pomocí sítě Internet, je nutné splnit tyto požadavky:

- modul ESP8266EX,
- Wi-Fi připojení do sítě Internet,
- firmware schopný vzdálené aktualizace,
- webový server obsahující binární soubor.

Pro takto jednoduchý příklad postačí tento soubor jednoduše stáhnout za pomocí ESP8266EX a přepsat Flash paměť v oblasti A jeho obsahem.

Nedostatky této metody jsou:

- při nedokončení zápisu se stává zařízení nefunkční,
- nově nahrávaný firmware může být během přenosu nahrazen firmwarem podvrhnutým.

7.3.2 Ochrana proti přerušení aktualizace

První nedostatek metody v kapitole 7.3.1 je velmi kritický, protože by se k poškozenému zařízení musel připojit fyzicky programovací kabel. Až toto manuální přeprogramování by ho dovedlo opět zprovoznit. Proto je nutné této možné katastrofické události předejít. Jako možné řešení se zde nabízí možnost zápisu do volné oblasti Flash paměti, konkrétně do oblasti C. K zapnutí této možnosti stačí přidat kompilátoru kompilační vlajku navíc. A tou je ATOMIC_FS_UPDATE. Díky tomu dojde při aktualizaci k zápisu do oblasti C a po restartu zařízení se tato oblast přesune za pomocí zavaděče do oblasti A, a tím pádem se spustí nově nahraný firmware. Pokud ovšem není dostatek místa ve Flash paměti v dané oblasti, tato aktualizační metoda selže. Je tedy nutné zajistit, aby nebyla překročena maximální velikost firmware. Zařízení je proti tomuto nedostatku chráněno jedním způsobem. Tento způsob spočívá v přečtení HTTP hlavičky Content-Length při stahování nového firmware. V případě, že je větší než velikost oblasti, aktualizace se přeruší.

7.3.3 Ochrana proti podvrhnutí aktualizačního serveru

Před stažením nového firmware si CesDev1 ověří řetěz certifikátů. Cloud má certifikát generovaný od autority Let's Encrypt. Zařízení CesDev1 si ověří Common Name certifikátu, který od serveru obdrží. Zařízení to provádí automaticky zabudovaná knihovna starající se o šifrování.

7.3.4 Ochrana proti podvrhnutí firmware

Aby nedošlo k podvrhnutí firmware, je nutné zajistit jeho konzistenci při přenosu. Toto lze docílit pomocí asymetrické kryptografie. Nejdříve se vygeneruje privátní klíč, který bude sloužit k podepisování firmware. Dále k němu bude vygenerovaný veřejný klíč, pomocí kterého se dá zjistit, zdali konkrétní firmware byl tím privátním klíčem podepsán a tím pádem zůstal nepodvržený.

Na straně CesDev1 je klíč veřejný. Na straně serveru při nahrávání firmware budou nutné dvě věci: nově zkompilovaný firmware a privátní klíč k podepsání. Server daný firmware následně podepíše privátním klíčem a na serveru zůstává pouze binární soubor jím podepsaný. Formát konečného binárního souboru je firmware a kryptografický podepsaný haš spojený v jeden.
CesDev1 se po zapnutí a připojení k Wi-Fi snaží kontaktovat aktualizační server. CesDev1 posílá požadavek s aktuálním hašem svého firmware. Tento haš porovná cloud s aktuálně nahraným, a pokud nesouhlasí, odpoví mu zasláním nového firmware. V případě, že tyto haše souhlasí, CesDev1 pokračuje s připojením k MQTT. Pokud došlo ke stažení nového firmware CesDev1, vzhledem k omezené velikosti RAM, si ho uloží do paměti Flash v oblasti C. Po tom, co se celý firmware stáhne, si CesDev1 ověří validitu firmware pomocí uloženého veřejného klíče. Pokud je firmware validní, zařízení se restartuje a přesune tento firmware na začátek paměti Flash. V případě, že firmware byl podvrhnut, tak tento krok neprovede a pokračuje jako by nebyla žádná aktualizace dostupná.

7.3.5 Návrhy na zlepšení

V průběhu návrhu byla nalezena možnost vylepšení ochrany proti podvržení firmware:

- lepší ochrana velikosti firmware,
- po nahrání podvrhnutého firmware by mělo následovat jeho smazání.

7.4 Uložení veřejného klíče do CesDev1

CesDev1 má již při kompilaci firmware zabudovaný, předem vygenerovaný veřejný klíč k ověření kryptografického podpisu.

7.5 Cloud

Veškeré služby lze nainstalovat na jeden virtuální server, což pro vlastní cloud bohatě postačí. Pro velký systém by bylo vhodné server rozdělit na několik dílčích částí. Příkladem může být: MQTT broker, databázový server, vizualizační server.

7.5.1 MQTT Broker

Každé CesDev1 zařízení se musí připojit přes MQTT k tzv. MQTT brokeru (serverová část). Jedná se o centrální místo, ke kterému bude každé CesDev1 připojené, aby mohlo posílat naměřená data. Komunikace mezi CesDev1 a MQTT broker je šifrovaná.

7.5.2 Webový server

Webový server poskytuje službu, která uživateli umožňuje vzdálenou aktualizaci. Toto zajišťuje naprogramovaná interaktivní webová aplikace. Druhou vlastností je uložiště všech nahraných firmwarů. Pro zamezení platformní závislosti je aplikace navržena pro běh v jakémkoliv moderním prohlížeči. Podrobněji je webová aplikace rozepsána v kapitole 7.6.

7.5.3 Služba ukládající MQTT data do databáze

MQTT broker nikdy nebyl navržen k tomu, aby byla data, která dostává, ukládána. Tedy tuto funkci neposkytuje. V případě této aplikace se jedná o konfigurovatelného prostředníka mezi databázovým serverem a MQTT brokerem. Pracovní název této aplikace je MMB (zkratka pro anglické MQTT MySQL/MariaDB Bridge).
Tato aplikace nebude mít žádné grafické rozhraní, pouze výstup ve formě standardního výstupu. Jedná se tedy o konzolovou aplikaci. Ke konfiguraci je použit lidsky jednoduše srozumitelný serializační formát YAML. Jako nutné konfigurační parametry je potřeba uvést tyto:

- MQTT certifikáty (klientské a serverové),
- údaje pro připojení k MQTT brokeru,
- údaje pro připojení k MariaDB serveru,
- parametry zařízení, pro které jsou data ukládána.
    - název zařízení,
    - tabulka v databázi,
    - minimální časový interval uložení,
    - MQTT topic,
    - korespondující sloupec v databázi.

Příkladem validní konfigurace pro uložení dat ze zařízení ESP_1 založené na CesDev1:

```yaml
ServerCertificate: ca.crt
ClientCertificate: client_no_ca.pfx
MySql:
  Username: 'sensors'
  Password: '123456'
  Server: 'server.com'
  Port: 3306
  Database: 'sensors'
Mqtt:
  Username: 'mqttClient'
  Password: 'esp%1'
  UniqueIdentifier: 'MYSQLBridge'
  Server: 'server.com'
  Port: 8883
DeviceSubscriptions:
  - Table: 'esp_1'
    Interval: '00:00:30'
    UseUpTime: true
    Properties:
      - Topic: '%table%/sht30/temperature'
        Column: 'sht30_temperature'
      - Topic: '%table%/sht30/humidity'
        Column: 'sht30_humidity'
      - Topic: '%table%/bmp280/temperature'
        Column: 'bmp280_temperature'
      - Topic: '%table%/bmp280/pressure'
        Column: 'bmp280_pressure'
```

Obrázek 9. Validní konfigurace MMB aplikace (zdroj: autor)

7.5.4 Databázový server

Pro celý systém jsou navrženy dvě databáze běžící na jednom databázovém serveru. Databázových serverů existuje velké množství, ať už se jedná o relační nebo nerelační, komerční nebo nekomerční. Zde je z důvodu předem jasných dat navržena nekomerční relační databáze MariaDB.

Systém se bude skládat ze dvou databází. Role první databáze je poskytování a ukládání dat z webového serveru. Tato data jsou informace o uživatelích a zařízeních, která jsou zavedená v systému. Základní strukturu této databáze tvoří dvě tabulky. Informace o uživatelích a informace o zařízeních.

Tabulka 5. Struktura databázové tabulky zařízení (devices) je určená k ukládání dat o zařízeních, která se aktualizují vzdáleně. Obsahuje pět sloupců. První sloupec zařízení (device) uchovává název zařízení v systému. Jedná se o primární klíč, a tedy

o unikátní název. Jeho délka může být až dvacet pět znaků. Druhým sloupcem je verze (version). Jedná se o verzi firmware, se kterým se zařízení s konkrétním názvem reportuje k webové službě. Toto číslo se při každé kompilaci zvedá o jedničku. Třetí sloupec soubor (has_file) říká, jestli je k danému zařízení nahrán na webu firmware. Další sloupec (signed) toto doplňuje o vlastnost firmwaru, tedy jestli je digitálně podepsán. Pátý sloupec (last_update) říká, kdy byl u zařízení naposledy aktualizován firmware. Poslední sloupec (update_status) určuje, v jakém stavu se nahraný firmware nachází. Mohou nastat tři stavy:

1. čeká na stažení – firmware byl nahrán a zařízení si ho ještě nestáhlo,
2. stažen – firmware byl stažen, ale zařízení se s tímto firmware ještě webové aplikaci neohlásilo,
3. aktualizován – firmware byl úspěšně zaveden a zařízení se ohlásilo webové aplikaci.

Tabulka 5. Struktura databázové tabulky zařízení (devices); (zdroj: autor)

Název sloupce	Datový typ	Délka	Výchozí hodnota
device	VARCHAR	25	AUTO_INCREMENT
version	INT	11	1
has_file	TINYINT	1	0
signed	TINYINT	1	0
last_update	TIMESTAMP	4	
update_status	TINYINT	2	0

Tabulka uživatelů (users) uchovává informace o uživatelích systému. Skládá se z pěti sloupců. První sloupec je unikátní identifikátor (ID), který je automaticky generovaný. Druhý sloupec (username) obsahuje unikátní přihlašovací jméno uživatele. Třetí sloupec email ukládá unikátní emailovou adresu uživatele. Obě tyto hodnoty lze použít k přihlášení. Čtvrtý sloupec ukládá haš uživatelova hesla (password). Poslední sloupec označuje roli uživatele v systému (role). Aktuálně se v systému nachází dvě role:

- admin – plnohodnotný administrátor, může spravovat zařízení a uživatele v systému,
- user – slouží pouze k nahlížení do stavu zařízení, nemá žádné jiné pravomoci.

Tabulka 6. Struktura databázové tabulky uživatelů (users); (zdroj: autor)

Název sloupce	Datový typ	Délka	Výchozí hodnota
id	INT	10	AUTO_INCREMENT
username	VARCHAR	50	-
email	VARCHAR	50	-
password	VARCHAR	100	-
role	VARCHAR	50	user

Druhá databáze ukládá data z MQTT pro zobrazení historie měření. Obsahuje jednu tabulku pro každé zařízení, kterému jsou ukládána naměřená data v pravidelných intervalech. Pro demonstraci je v systému jedno zařízení – esp_1.

Tabulka s naměřenými hodnotami z esp_1 zařízení má sedm sloupců. První sloupec (id) je unikátní identifikátor záznamu. Druhý sloupec (datetime) je časová známka určující datum a čas pořízení měření. Třetí až šestý sloupec zachycuje všechny naměřené veličiny v číslech s posuvnou desetinnou čárkou. Poslední sedmý sloupec (uptime) je doba běhu zařízení v minutách od posledního zapnutí.

Tabulka 7. Struktura databázové tabulky měření (esp_1); (zdroj: autor)

Název sloupce	Datový typ	Délka	Výchozí hodnota
id	INT	10	AUTO_INCREMENT
datetime	DATETIME	8	current_timestamp()
sht30_temperature	FLOAT	4	-
sht30_humidity	FLOAT	4	-
bmp280_temperature	FLOAT	4	-
bmp280_pressure	FLOAT	4	-
uptime	INT	10	-

Pro určení průměrné velikosti jednoho záznamu je možné využít funkce databázového systému MySQL, který ukládá statistické informace o všech tabulkách

v databázi *information_schema*. Na základě následujícího dotazu byla vypočtena velikost jednoho záznamu v tabulce. Ta činí 101 B (zaokrouhleno na jednotky) pro jeden záznam při počtu záznamů převyšující hodnotu deset tisíc.

```
SELECT (DATA_LENGTH / (SELECT COUNT(*) FROM sensors.esp_1)) AS
"BytesUsedPerRow" FROM information_schema.TABLES WHERE TABLE_SCHEMA =
"sensors"  AND TABLE_NAME = "esp_1"
```

Za předpokladu, že se do databáze ukládají data dvakrát za minutu, tak velikost uložených dat za delší časové období znázorňuje následující tabulka.

Tabulka 8. Využití kapacity disku v databázi; interval 30 s (zdroj: autor)

Období běhu systému	Využité místo v uložišti
1 týden	1,94 MiB
1 měsíc	8,32 MiB
1 kvartál	24,97 MiB
1 rok	99,87 MiB

Z tabulky lze odečíst velikost uložených dat za delší časové období pro jedno zařízení. V případě jednoho kalendářního roku (365 dní) se využije během provozu zhruba 100 MiB místa na disku. Velikost uložených dat je lineárně závislá na periodě ukládání dat.

7.6 Webová aplikace pro vzdálenou správu aktualizací

Pro zajištění samotné aplikace je využit programovací jazyk PHP spolu s českým Model-View-Presenter (MVP) frameworkem Nette. Konkrétně kostra aplikace vychází ze šablony *nette/web-project*. Díky použití tohoto MVP frameworku je možné logicky rozdělit kód a zajistit jednoduchost pro budoucí rozšíření aplikace.

Aplikace se aktuálně skládá ze třech stránek. Ostatní interakce se řeší pomocí tzv. modálních oken. Modalová okna jsou speciální otevírací okna určená pro interakci s uživatelem. Nahrazují klasické přesměrování uživatele na jinou, speciální stránku k zadání dat.

Webová aplikace má několik funkcí, se kterými bude možné spravovat firmware jednotlivých zařízení. Mezi tyto funkce patří:

- přihlašovací stránka,
- stránka pro správu zařízení,
- stránka pro správu uživatelů.

7.6.1 Přihlašovací stránka

Webová aplikace, která má zajistit bezpečnou aktualizaci firmware v zařízení, musí být sama zabezpečena. Základním způsobem zabezpečení jsou uživatelské účty. Tyto uživatelské účty mají omezené přístupy, které jsou definovány tzv. Access Control List – ACL. Jedná se o velice modulární přístup udělování pravomocí dle uživatelských rolí různým zdrojům. Ve webové aplikaci se tím upravuje přístup k jednotlivým stránkám a jejich obsah. Jednotlivé role jsou detailněji rozepsány v kapitole 7.6.3.

7.6.2 Stránka pro správu zařízení

Stránka pro správu zařízení obsahuje tabulku všech zavedených zařízení v systému. Systém v aktuální podobně nerozděluje zařízení např. různým uživatelům nebo skupinám, nýbrž zobrazuje všechny. Z této stránky je možné nahrát a případně i digitálně podepsat firmware, které si CesDev1 zařízení po restartu samo aktualizuje. Privátní klíče se ve webové aplikaci z důvodu zabezpečení neukládají. Jsou na serveru uložena dočasně jen a pouze při nahrávání a podepsání firmware. Zároveň je zobrazen vizuálně stav aktualizace pomocí emotikonu ze sady UTF-8.

7.6.3 Stránka pro správu uživatelů

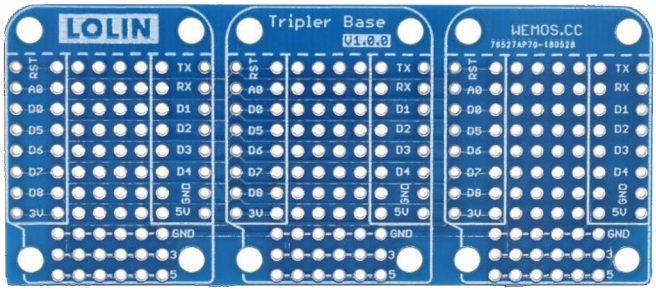
Stránka pro správu zařízení obsahuje tabulku všech uživatelů v systému. V systému jsou zavedeny dvě role – admin a user. Konkrétní pravomoci, které mají jsou uvedeny v kapitole 7.5.4. Uživatele lze jako admin libovolně přidávat, mazat a upravovat. Jedinou výjimkou je nemožnost smazat aktuálně přihlášeného uživatele. Tato stránka je přístupná pouze pro uživatele s rolí admin.

8 Postup realizace navrženého technického řešení

8.1 Výroba zařízení CesDev1

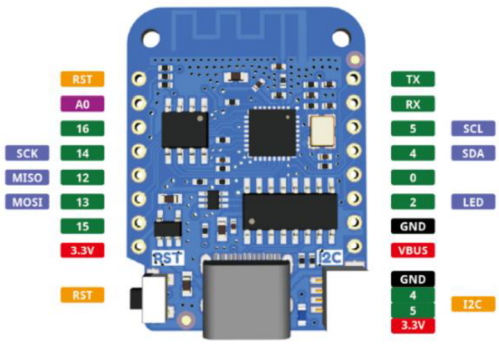
8.1.1 Hardware zařízení CesDev1

Aby bylo možné zařízení programovat, je nejdříve nutné ho sestrojit. Pro prototypovou desku je využita platforma Wemos. Jedná se o Wemos Tripler Base. Je to deska o šířce třech slotů, do které lze modulárně zasouvat moduly společnosti Wemos do výšky. Tedy maximální množství modulů není omezené na tři.



Obrázek 10. Wemos Tripler Base (zdroj: [10])

K připojení ESP8266EX modulu k této desce je využit modul Wemos D1 mini. Tento modul obsahuje navíc i tlačítko pro hardwarový reset ESP8266EX. Dále také obsahuje USB řadič a USB port pro snadné připojení k PC a následné programování.



Obrázek 11. Identifikace pinů modulu Wemos D1 mini pro ESP8266EX (zdroj: [9])

Finální zařízení je uzavřeno v praktické krabičce s výřezy pro senzory, pro OLED displej a pro tlačítko pro hardwarový reset. Dále tam jsou dvě indikační LED diody.

Jedna pro zobrazení stavu napájení a druhá indikující přenos MQTT. Výsledná podoba zařízení je vidět na obrázku níže s potiskem pro školní demonstraci.

Obrázek 12. Výsledná podoba CesDev1 zařízení - esp_1 (zdroj: autor)

8.1.2 Software pro vývoj CesDev1

Aby bylo možné vyvinout tento systém, je potřeba nainstalovat na počítač několik komponentů. Prvním a základním komponentem je vývojové studio Arduino IDE, jedná se o vývojové prostředí používané pro programování Arduino vývojových desek, ale je zde možné přidat podporu pro desky třetích stran. Tím lze přidat podporu pro právě používaný ESP8266EX modul. Návod, jak ho přidat lze nalézt na stránkách GitHubu pod projektem ESP8266 dostupný na adrese https://github.com/esp8266/Arduino#installing-with-boards-manager. Bohužel Arduino IDE není moc intuitivní vývojové prostředí, protože nenabízí v podstatě žádnou správu souborů, žádné našeptávání. Nabízí se zde možnost využít vývojového prostředí Visual Studio, které je mnohonásobně lepší pro vývoj. To ovšem nativně nepodporuje vývoj na Arduino platformě, ale je možné si zakoupit rozšíření Visual Micro, které tuto podporu přidává. Díky této kombinaci je možné ušetřit spoustu času při vývoji.

Samotná podpora ESP8266 desek však nestačí pro to, aby se šlo s CesDev1 připojit pohodlně k MQTT brokeru. K tomu je nutné doinstalovat několik knihoven.

K zprovoznění základní funkcionality je nutné nainstalovat tyto knihovny:

- PubSubClient – MQTT klientská knihovna,
- Adafruit_BMP280 – knihovna pro obsluhu BMP280 čidla,
- WEMOS_SHT3X – knihovna pro obsluhu SHT30 čidla,
- Adafruit_SSD1306 – knihovna pro obsluhu OLED displeje.

8.2 CesDev1 firmware

8.2.1 Celkový nadhled

Pro vývoj softwaru je využit programovací jazyk C++. Ten umožňuje objektový návrh a díky jeho snadné znuvupoužitelnosti byl i zvolen. Samotný projekt se skládá, jako každý Arduino projekt, ze souboru s příponou ino. V něm je zahrnuta veškerá logika programu.

8.2.2 Třída WiFiConnection

Třída, která modulu umožňuje připojit se k Wi-Fi. Dále také synchronizuje čas pomocí NTP protokolu k tomu, aby bylo možné ověřit validitu certifikátů.

```
void WifiConnection::get_time_from_ntp()
{
    configTime("CET-1CEST-2,M3.5.0/02:00:00,M10.5.0/03:00:00",
        "pool.ntp.org", "tik.cesnet.cz");
    auto clock_tries = 0;
    CustomSerial::print("NTP", "Waiting for time sync");
    auto now = static_cast<time_t>(0);
    now = time(nullptr);
    while (now < 57600)
    {
        now = time(nullptr);
        clock_tries++;
        if (clock_tries >= 200)
        {
            CustomSerial::println("NTP",
                "Failed to get time. Device will now restart");
            WiFi.disconnect(true);
            ESP.restart();
        }
        if (clock_tries % 40 == 0)
        {
            Serial.print(".");
        }
        delay(100);
    }
```

Obrázek 13. Úryvek kódu metody pro získání času z NTP (zdroj: autor)

8.2.3 Třída OTAChecker

Tato třída se stará o několik věcí týkajících se aktualizace „vzduchem“. Nejdříve načte všechny potřebné podpisy, tedy veřejný podepisovací klíč, kořenový a prostřední certifikát od certifikační autority Let's Encrypt. Poté zařízení kontaktuje aktualizační server, aby zjistilo, jestli je dostupná aktualizace. Konkrétní postup aktualizace je popsán v kapitole 7.3.

```
char buffer[12]{};
itoa(static_cast<int>(version), buffer, 10);
CustomSerial::println("OTA", "Checking for update");
abstract_display->setHeaderText("OTA");
const auto ret = ESPhttpUpdate.update(*secure_client,
    "https://hydra.cesak.com/update/download",
    device_name, buffer);
switch (ret)
{
case HTTP_UPDATE_FAILED:
    CustomSerial::printfln("OTA", "Couldn't update firmware. Reason: %s",
        ESPhttpUpdate.getLastErrorString().c_str());
    break;
case HTTP_UPDATE_NO_UPDATES:
    CustomSerial::println("OTA", "Device firmware is up to date");
    break;
}
```

Obrázek 14. Úryvek kódu obstarávající vzdálenou aktualizaci (zdroj: autor)

8.2.4 Třída MQTTClient

MQTTClient je třída zapouzdřující MQTT komunikaci. Stará se o správné načtení certifikátů pro šifrovanou komunikaci s brokerem. Také se stará o připojení a udržování spojení. Nejdůležitějším prvkem, které zprostředkovává tato třída je odesílání a přijímání zpráv.

```
void MQTTClient::mqtt_response(const char* topic,
    const uint8_t* payload, const unsigned int length)
{
    last_tx_time = millis();

    if (abstract_display_ != nullptr)
    {
        abstract_display_->setRX(true);
    }

    string payload_string(reinterpret_cast<const char*>(payload));
    payload_string = payload_string.substr(0, length);
    const string topic_string(topic);


    CustomSerial::printfln("MQTT",
        "RECEIVED DATA Topic: %s Value: '%s'",
        topic_string.c_str(), payload_string.c_str());

    for (const auto& subscribed_method : subscribed_methods_)
    {
        if (get<1>(subscribed_method) == topic_string)
        {
            get<2>(subscribed_method)(topic_string, payload_string);
        }
    }
}
```

Obrázek 15. Úryvek kódu obsluhující příchozí zprávy z MQTT (zdroj: autor)

8.2.5 Třída Sensor

Abstraktní třída, která unifikuje rozhraní pro veškeré senzory. Díky tomu je velmi jednoduché obsluhovat celou řadu senzorů nezávisle na jejich implementaci. Nejedná se o rozhraní, jelikož rozhraní jako takové v jazyce C++ neexistuje.

```
class Sensor
{
private:
    const char* sensorName;

protected:
    MQTTClient* mqttClient;

public:
    virtual void init() = 0;
    virtual void measureAndSend() = 0;
    virtual ~Sensor() = 0;
    Sensor(const char* sensorName, MQTTClient* mqtt);
    const char* getSensorName();
};
```

Obrázek 16. Předpis abstraktní třídy Sensor (zdroj: autor)

8.2.6 Podepisovací certifikáty

Jedním z nejdůležitějších komponentů bezpečné aktualizace je pár veřejného a privátního klíče asymetrického šifrovacího algoritmu RSA. Veřejný klíč zůstane v CesDev1 firmware a privátní klíč zůstane uschován na místě, kde bude nejmenší riziko ho odcizit a zneužít pro tvorbu alternativního firmware.

Jako typ šifrování je využit algoritmus RSA. K vygenerování je použita svobodná knihovna OpenSSL.

K vygenerování klíče o délce 2048 bitů se používá tento příkaz:

```
openssl genrsa -out "D:\cesdev1.key" 2048
```

Tento klíč je určen k podepisování a neměl by se dostat k někomu, kdo by této skutečnosti mohl zneužít. Pro vygenerování veřejného klíče se používá tento příkaz:

```
openssl rsa -in "D:\cesdev1.key" -outform PEM -pubout -out
"D:\cesdev1.crt"
```

Následný klíč je pak uložen ve statické paměti CesDev1 zařízení určené pouze ke čtení a není možné ho jednoduše změnit.

8.3 Vlastní cloud

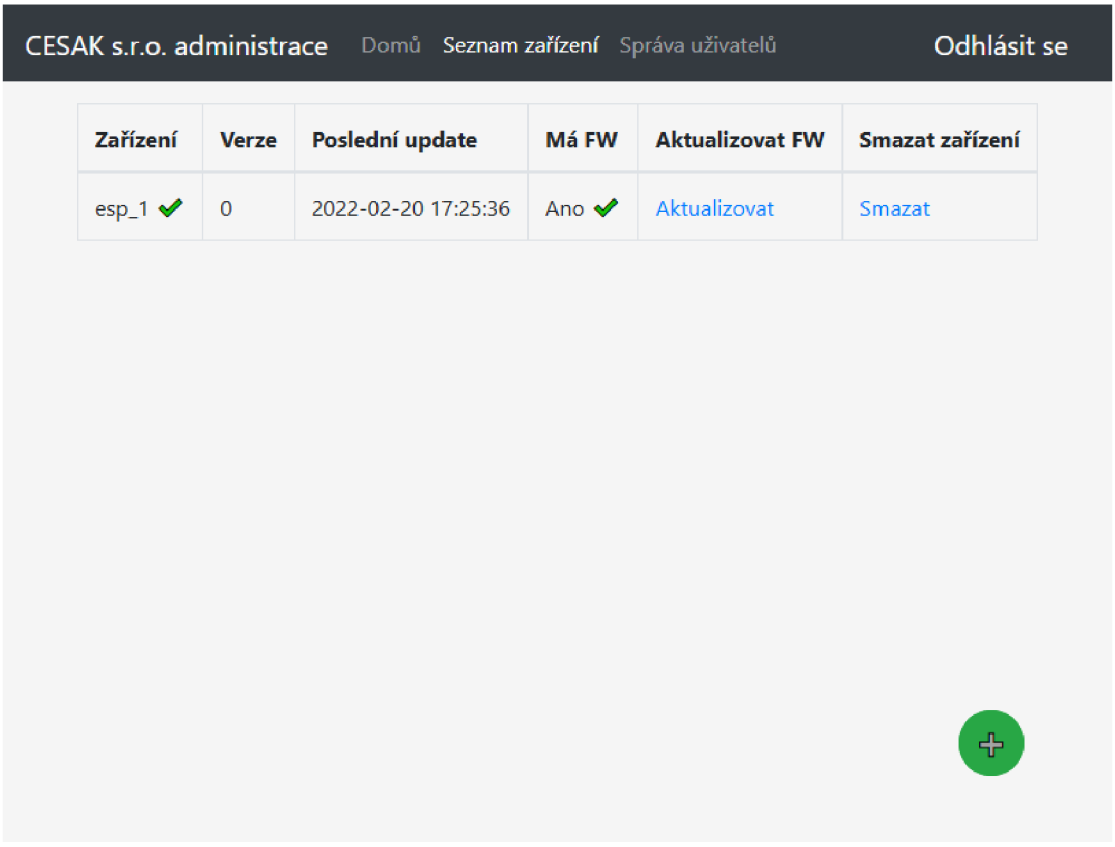
8.3.1 Tvorba webového rozhraní

K instalaci šablony *nette/web-project* je nejjednodušší mít nainstalovaný správce balíčků pro PHP – Composer. Poté ji stačí stáhnout v příkazové řádce pomocí příkazu:

```
composer create-project nette/web-project cesak/device-updater
```

Tím se vytvoří šablona v Nette Frameworku. Jelikož je Nette kompatibilní s aktuálně nejnovější verzi PHP – 8.1, je zároveň i tato verze použita. Pro běh samotné PHP aplikace je použit webový server Apache HTTP Server. Aplikace je logicky rozvržena dle MVP architektury.

Ukázku z webového rozhraní je možné vidět na obrázku níže, kde je vidět stránka pro správu zařízení.



Obrázek 17. Ukázka webové aplikace - správa zařízení (zdroj: autor)

8.3.2 Tvorba MMB aplikace

K uložení dat z MQTT brokeru do libovolné databáze neexistuje nativní řešení, které by bylo specifikováno a implementováno přímo v brokeru. Tedy pro tuto činnost je naprogramována velmi jednoduchá aplikace v multiplatformním jazyce C#. Vzhledem k tomu, že databáze, do které se ukládá, je MariaDB a ta není nativně podporována v jazyce C#, bude nutné doinstalovat obslužnou knihovnu. Tento samý problém má i implementace MQTT protokolu pro klienta i serializační formát s názvem YAML.

Pro vývoj je použito vývojové prostředí Visual Studio. Aplikace běží na nejnovější verzi běhového prostředí .NET ve verzi 6 (jedná o stabilní verzi s dlouhou podporou). Pro přidání podpory jsou pomocí správce balíčků NuGet nainstalovány tyto knihovny.

Tabulka 9. Použité balíčky v MMB aplikaci (zdroj: autor)

Název balíčku	Popis
CommandLineParser	Podpora pro zpracování parametrů zadaných přes příkazovou řádku
MQTTnet	Podpora připojení k MQTT brokeru
MySql.Data	Podpora připojení k MariaDB / MySQL databázi
YamlDotNet	Podpora serializačního formátu YAML

K věcem zmíněných v kapitole 7.5.3 je do aplikace přidána i možnost takzvaného „verbose“ výstupu. Ve výchozím režimu tato aplikace do konzole nic nevypisuje, ale je možné pomocí vstupních parametrů zadaných přes příkazovou řádku zapnout ladicí výstup, který následně vypisuje statusy při pokusu o ukládání do databáze (úspěch/neúspěch). Tento výstup je aktivován pomocí parametru `-v` nebo `–verbose`. V tomto případě se výstup povolí na všechna zařízení zkonfigurovaná v konfiguračním souboru. V případě nutnosti lze toto nastavení omezit na konkrétní zařízení přidáním jejich názvu za parametr. V případě nutnosti lze přidat další zařízení oddělené mezerou.

8.3.3 Instalace potřebného software

Pro získání vlastního cloudu je vhodné si pořídit virtuální server (VPS) s dostatečnými požadavky, které jsou specifikované výše v práci, viz. Tabulka 2. Požadavky na vlastní cloud. Jako operační systém byl zvolen GNU/Linux Debian v aktuální verzi 11.2. Pro splnění všech požadavků je nutné doinstalovat několik balíčků do operačního systému. Tyto balíčky jsou uvedeny v tabulce Nutné balíčky pro běh celého systému.

Tabulka 10. Nutné balíčky pro běh celého systému (zdroj: autor)

Název software	Balíček	Popisek
Eclipse Mosquitto	mqtt	MQTT broker
Apache HTTP server	apache2	Webový server
PHP 8.1	php8.1, php8.1-fpm, php8.1-bz2, php8.1-curl, php8.1-fileinfo, php8.1-mbstring, php8.1-mysqli	Programovací jazyk potřebný pro běh webového managementu
MariaDB	mysql	Relační databázový server
.NET 6.0 runtime	dotnet-runtime-6.0	Běhové prostředí pro .NET 6, které využívá MMB aplikace

9 Demonstrace funkčnosti systému

K úspěšné demonstraci tohoto systému je nutné splnit jeho požadavky. Pro tento konkrétní případ jsou zvoleny následující parametry.

9.1 Sestava pro demonstrování

Testovací systém se skládá z jednoho serveru (cloudu) a jednoho testovacího zařízení CesDev1 s názvem esp_1.

9.1.1 Sestava serveru

Všechny služby běží na jednom serveru, což ulehčuje administraci. Server má parametry, které jsou popsány v tabulce níže.

Tabulka 11. Parametry testovacího serveru (zdroj: autor)

Parametr	Hodnota
Počet jader	1
Velikost operační paměti	2048 MiB
Velikost uložiště	80 GiB
Počet veřejných IPv4 adres	1
Operační systém	GNU/Linux Debian 11.2

9.1.2 Sestava CesDev1 zařízení s názvem esp_1

Samotné zařízení se skládá z několika vzájemně propojených modulů, jejich výčet je následující:

- jedno CesDev1 zařízení s názvem esp_1,
- dvě čidla připojená k esp_1,
  - BMP280,
  - SHT30.
- jeden OLED display SSD1306 s rozlišením 128 x 64 pixelů a úhlopříčkou 0.96“ připojený k esp_1,
- esp_1 periodický měřící v intervalu 30 sekund.

9.2 Přehled splnění požadavků

K úspěšné demonstraci je nutné splnit alespoň všechny povinné požadavky. Proto byla sestavena tabulka, která shrnuje splnění požadavků. První tabulka je pro samotné měřící zařízení, druhá je pro cloud. Texty požadavků k jednotlivým číslům jsou uvedeny v kapitolách 5.1 a 5.2.

Tabulka 12. Splnění požadavků na měřicí zařízení (zdroj: autor)

Číslo požadavku	Splněno	Doplňující informace
REQ-IOT-11	Ano	
REQ-IOT-21	Ano	
REQ-IOT-31	Ano	
REQ-IOT-41	Ano	
REQ-IOT-51	Ano	
REQ-IOT-61	Ano	
REQ-IOT-71	Ano	
REQ-IOT-81	Ano	
REQ-IOT-91	Ano	
REQ-IOT-101	Ne	Zařízení je větší, konkrétně ~ 270 cm^3.
REQ-IOT-111	Ano	
REQ-IOT-121	Ne	Zařízení je napájeno pomocí zdroje s napětím 9 V.

Požadavky na měřicí zařízení byly splněny s drobnou odchylkou od rozměru a napájecího napětí. Vzhledem k tomu, že se nejedná o povinné požadavky, nelze to považovat za chybu. Požadavky na cloud byly splněny všechny.

Tabulka 13. Splnění požadavků na cloudu (zdroj: autor)

Číslo požadavku	Splněno	Doplňující informace
REQ-SRV-12	Ano	
REQ-SRV-22	Ano	
REQ-SRV-32	Ano	
REQ-SRV-42	Ano	
REQ-SRV-52	Ano	
REQ-SRV-62	Ano	
REQ-SRV-72	Ano	
REQ-SRV-82	Ano	Broker používá přihlašovací údaje i asymetrickou kryptografii pro přenos dat.
REQ-SRV-92	Ano	
REQ-SRV-102	Ano	

9.3 Postup aktualizace

Po instalaci a úspěšném zprovoznění systému je možné začít využívat jeho výhod. K tomu, aby byla aktualizace úspěšná je nutné provést několik kroků.

- připojení esp_1 zařízení k Wi-Fi s přístupem k internetu,
- upravení stávajícího firmware na základě nových požadavků / oprava chyb,
- kompilace nového firmware,
- zavedení nového zařízení / úprava stávajícího zařízení ve webové aplikaci,
- nahrání nového firmware spolu s privátním podepisovacím klíčem,
- restartování CesDev1 zařízení.

Výsledná velikost nového firmware je s digitálním podpisem 579 kB. V případě, že vše proběhlo v pořádku, bude zařízení vypisovat následující výstup do sériové konzole po jeho restartu. Vzhledem k limitaci není možné mít současně připojení k MQTT a zároveň bezpečně ověřovat dostupnost nové verze. Důvodem je nedostatek paměti pro uložení dat ze dvou šifrovaných připojení současně. Pro to je nutné zařízení restartovat.

```
[0.076] INFO: CESAK s.r.o. sensor chip. ID: esp_1 FW Version: 439
Core version: 3.0.2 Build date: Mar 20 2022 14:40:29
[0.083] FILE SYSTEM: Successfully initialized
[0.088] DISPLAY: SSD1306 OK
[0.671] SHT30: Sensor is CONNECTED
[0.782] BMP280: Sensor is CONNECTED
[0.879] Wi-Fi: Attempting to connect to SSID: 'espap'
[4.996] Wi-Fi: Connection established to espap on channel 1
[5.617] Wi-Fi: Got data from DHCP. IP address: 192.168.0.128 Default
Gateway: 192.168.0.1 Netmask: 255.255.255.0 DNS server: 1.1.1.1
[5.620] NTP: Waiting for time sync
[7.621] NTP: Successfully got time from NTP
[7.622] NTP: The time is: Sun Mar 20 19:20:15 2022
[8.831] OTA: Checking for update
[11.055] OTA: Device is outdated. Starting update.
[11.867] OTA: Downloading update... 1%
[12.332] OTA: Downloading update... 2%
(několik řádků vynecháno)
[55.702] OTA: Downloading update... 99%
[56.091] OTA: Downloading update... 100%
[56.618] OTA: Update finished. Device will now restart.
```

Zařízení se po úspěšné aktualizaci restartuje, identifikuje se s novou verzí a do sériové konzole vypíše následující:

```
[0.078] INFO: CESAK s.r.o. sensor chip. ID: esp_1 FW Version: 440
Core version: 3.0.2 Build date: Mar 20 2022 14:40:29
```

Tím se potvrdila úspěšná aktualizace firmware a zařízení nyní bude pracovat podle kódu obsaženého v jeho nové verzi. V případě nalezení další chyby je možné proces opakovat a zařízení bude zase úspěšně aktualizováno za předpokladu, že bude digitálně podepsáno správným klíčem.

Aktuálně je systém provozován po dobu dvou měsíců s různě dlouhými výpadky, které nastaly z důvodu práce na hardwarové části systému (CesDev1). K dnešnímu dni 10. 4. 2022 byly provedeny desítky úspěšných aktualizací a samotný systém naměřil a uložil do databáze desítky tisíc záznamů.

9.4 Test podvržení firmware

Aby bylo možné zjistit, že se nenahraje nepodepsaný nebo podvrhnutý firmware a celý systém je tedy zabezpečený, je nutné provést tři testy:

1. podvrhnutí webového serveru,
2. stáhnutí firmware bez digitálního podpisu,
3. stáhnutí firmware s podvrženým digitálním podpisem.

9.4.1 Podvrhnutí webového serveru

Aby bylo potvrzené, že CesDev1 zařízení odolá útoku vůči změně DNS záznamů, byl z cloudu dočasně odebrán validní Let's Encrypt certifikát se správným doménovým názvem. Byl nahrazen certifikátem z jiné webové stránky, také vydaný autoritou Let's Encrypt. Výsledkem je neúspěšné kontaktování aktualizačního serveru, protože se CesDev1 odmítne připojit. Tento pokus o záměnu cloudu se projeví v konzoli následujícím řádkem.

```
[5.966] OTA: Couldn't update firmware. Reason: HTTP error: connection
failed
```

9.4.2 Stáhnutí firmware bez digitálního podpisu

Druhým testem je pokus o nahrání nového firmware bez digitálního podpisu. V tomto případě CesDev1 takový firmware odmítne spustit. Pouze si ho stáhne a po zjištění, že není nijak digitálně podepsáno jej ignoruje a přechází do dalšího kroku inicializace jako v případě, že by žádná aktualizace nebyla dostupná. Tato chyba se pozná pomocí řádku, který CesDev1 vypíše do konzole po stáhnutí nové verze.

```
[46.697] OTA: Couldn't update firmware. Reason: Update error: ERROR[12]:
Signature verification failed
```

9.4.3 Stáhnutí firmware s podvrženým digitálním podpisem

Třetím testem je nahrání nového firmware s digitálním podpisem se špatným privátním klíčem, Nejdříve je vytvořen alternativní privátní klíč a následně je nový firmware podepsán tímto klíčem. Zařízení se zachová téměř identicky, jako o bod výše. Jediný rozdíl je ten, že sice zjistí, že je firmware podepsaný, ale digitální podpis

mu nesouhlasí. Zařízení bude pokračovat dál v inicializaci a aktualizaci neprovede stejně, jako v předchozím testu. CesDev1 vypíše identickou chybovou hlášku, jako v případě nepodepsaného firmware.

```
[47.501] OTA: Couldn't update firmware. Reason: Update error: ERROR[12]:
Signature verification failed
```

9.4.4 Vyhodnocení pokusů o podvržení firmware

Všechny tři testy dopadly s výsledkem, jak je vyžadují požadavky. Z toho vyplývá, že systém je zabezpečen a není možné podvrhnout firmware pomocí vzdálené aktualizace bez původního privátního klíče a patřičného veřejného certifikátu. Zároveň také není možné podvrhnout certifikát cloudu, kam se CesDev1 dotazuje na dostupnost nové verze firmware.

10 Závěr

Systém pro bezpečnou aktualizaci firmware v MQTT zařízeních se podařilo navrhnout a implementovat. Testování proběhlo na jednom virtuálním serveru sloužící jako vlastní cloud a MQTT zařízení CesDev1 s názvem esp_1. CesDev1 je kompaktní a zároveň i lehce rozšiřitelné pomocí dostupných sběrnic. Zároveň byly provedeny testy, které zjistily, že není možné podvrhnout firmware přes vzdálenou aktualizaci bez znalosti původního privátní klíče. Jediným omezením CesDev1 je jeho závislost na dostupnosti Wi-Fi sítě s připojením do sítě Internet.

Další část systému, aplikace dovolující nahrát nový firmware, je vytvořena jako webová. Je tedy možné do ní nahrát novou verzi firmware z téměř libovolného zařízení. Tato aplikace zároveň umožňuje přiložit privátní klíč k digitálnímu podepsání právě nahrávaného firmware. Zároveň se jako součást systému podařilo vyvinout podpůrnou aplikaci, která ukládá naměřená data z MQTT brokeru do databáze pro historizaci naměřených dat získaných a odeslaných z CesDev1.

11 Shrnutí a doporučení

Navržený systém je plně funkční. Při používání bylo identifikováno několik podnětů pro další vývoj.

Jedním z nich je omezení ve webové aplikaci. V aktuálním stavu tato aplikace nepodporuje mít zařízení rozdělená jednotlivým uživatelům, což by se mohlo využít v případě, že by tento systém chtělo využívat více nezávislých uživatelů z jednoho společného serveru. Další limitací webového rozhraní je ta, že aplikace neukládá starší verze firmware, tedy je vždy dostupná jen a pouze jeho nejnovější verze.

Z hlediska samotného CesDev1 zařízení má taky několik nedostatků. Zařízení není možné bezpečně zjistit velikost stahovaného firmware. Pokud by přesáhlo velikost bloku, tak by zapisovalo do části se souborovým systémem a ten by byl následně poškozen. Další limitací je ta, že zařízení nemá dostatek paměti s náhodným přístupem pro udržení dvou šifrovaných připojení současně. Z toho vyplývá, že CesDev1 neumí být současně připojené k MQTT brokeru a zároveň ověřit dostupnost aktualizace.

12 Seznam použité literatury

[1] BANKS, Andrew a Rahul GUPTA. *MQTT Version 3.1.1* [online]. B.m.: OASIS Standard. 29. říjen 2014 [vid. 2022-02-13]. Dostupné z: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.pdf

[2] BOSCH. *Data sheet BMP280 Digital Pressure Sensor* [online]. 2015 [vid. 2022-03-26]. Dostupné z: https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bmp280-ds001.pdf

[3] ESPRESSIF SYSTEMS. *ESP8266EX Datasheet* [online]. 2020 [vid. 2021-12-11]. Dostupné z: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf

[4] EUROPEAN DIGITAL ASSETS EXCHANGE. *Blockchain Transactions: HASH Functions* [online]. 2019 [vid. 2022-03-26]. Dostupné z: https://edsx.ch/blog-news/blockchain-transactions-hash-functions

[5] NÚKIB. *Zpráva o stavu kybernetické bezpečnosti České republiky za rok 2020* [online]. 26. červenec 2021 [vid. 2022-04-10]. Dostupné z: https://www.nukib.cz/download/publikace/zpravy_o_stavu/Zprava_o_stavu_KB_2020.pdf

[6] OOSTENVELD, Robert. *ESP-12F diagram* [online]. 2016 [vid. 2022-03-26]. Dostupné z: https://robertoostenveld.nl/esp-12-bootloader-modes/

[7] SENSIRION. *Datasheet SHT3x-DIS* [online]. 2016 [vid. 2022-03-26]. Dostupné z: https://www.mouser.com/datasheet/2/682/Sensirion_Humidity_Sensors_SHT3x_Datasheet_digital-971521.pdf

[8] SYNCRONESS. *Secure Over-the-Air Firmware Updates for IoT Systems* [online]. 2016 [vid. 2022-03-26]. Dostupné z: https://www.syncroness.com/2016-8-29-secure-over-the-air-firmware-updates-for-iot-systems/

[9] WEMOS.CC. *Wemos D1 Mini* [online]. 2021 [vid. 2022-01-25]. Dostupné z: https://www.wemos.cc/en/latest/_static/boards/d1_mini_v4.0.0_5_16x9.png

[10] WEMOS.CC. *Wemos Tripler Base* [online]. 2021 [vid. 2022-03-26]. Dostupné z: https://www.wemos.cc/en/latest/d1_mini_shield/tripler_base.html

Seznam obrázků

Seznam tabulek

13 Přílohy

1) Zdrojové kódy CesDev1
2) Zdrojové kódy webové aplikace pro vzdálenou aktualizaci
3) Zdrojové kódy podpůrné aplikace pro ukládání dat do databáze
4) Excel soubor s cenami dostupných a vhodných zařízení pro systém

Univerzita Hradec Králové
Fakulta informatiky a managementu

Zadání bakalářské práce

Autor:	**David Česák**
Studium:	I1900163
Studijní program:	B1802 Aplikovaná informatika
Studijní obor:	Aplikovaná informatika
Název bakalářské práce:	**Systém pro bezpečnou aktualizaci firmware v MQTT zařízeních**
Název bakalářské práce AJ:	System for secure firmware update in MQTT devices

Cíl, metody, literatura, předpoklady:

Cíl práce

Cílem práce je návrh a realizace systému, který umožní bezpečnou aktualizaci firmware v zařízeních, která periodicky sbírají data a odesílají je prostřednictvím protokolu MQTT. Tato data budou dále zpracovávána a ukládána do databáze. Bezpečnou aktualizací se rozumí proces, při kterém nedojde k nahrání nevalidního (podvrženého) firmware.

Osnova

1. Úvod
2. Cíl práce
3. Metodika zpracování
4. Identifikace požadavků pro systém měření Návrh systému pro bezpečnou aktualizaci firmware
5. Charakterizace dílčích prvků systému podle požadavků
6. Návrh technického řešení
7. Postup realizace navrženého technického řešení
8. Demonstrace funkčnosti systému
9. Závěr
10. Shrnutí a doporučení

- ESPRESSIF SYSTEMS. ESP8266EX Datasheet [online]. 2020. Dostupné z: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf
- KOLBAN, Neil. *Kolban's Book on the ESP32 {\& ESP8266 [online]. 1. 2016. ESP8266. Dostupné také z: https://leanpub.com/ESP8266_ESP32*
- *HUNKELER, Urs; TRUONG, Hong Linh; STANFORD-CLARK, Andy. MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks. In: 2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08) . IEEE, 2008. p. 791-798*

Garantující pracoviště:	Katedra informačních technologií, Fakulta informatiky a managementu
Vedoucí práce:	Mgr. Daniela Ponce, Ph.D.
Datum zadání závěrečné práce:	21.1.2021