VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ
DEPARTMENT OF COMPUTER SYSTEMS

INTELIGENTNÍ PŘÍSTUPOVÝ SYSTÉM PRO VĚTŠÍ OBJEKTY

INTELLIGENT ACCESS CONTROL SYSTEM FOR LARGE FACILITIES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **JAN TRUHLÁŘ**
AUTHOR

VEDOUCÍ PRÁCE **Ing. ZDENĚK VAŠÍČEK, Ph.D.**
SUPERVISOR

BRNO 2017

Zadání bakalářské práce/19542/2016/xtruhl05

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačových systémů Akademický rok 2016/2017

Zadání bakalářské práce

Řešitel: **Truhlář Jan**
Obor: Informační technologie
Téma: **Inteligentní přístupový systém pro větší objekty**
Inteligent Access Control System for Large Objects
Kategorie: Vestavěné systémy

Pokyny:

1. Seznamte se problematikou identifikace osob pomocí RFID a dále platformou ESP8266, jejími možnostmi a dostupným programovým vybavením. Zaměřte se zejména na podporu MESH sítí umožňujících spojit jednotlivé body do jedné sítě.
2. Navrhněte autonomní vestavěné zařízení na bázi ESP8266, které bude umožňovat řízení přístupu do určité části objektu. Předpokládejte vždy přítomnost elektronicky ovládaného zámku a zdroje střídavého napětí 12/24V. Celý systém navrhněte tak, aby bylo možné jednotlivá zařízení spojit do MESH sítě a přístupová práva tak spravovat z kterékoliv místa.
3. Zpracujte studii na výše uvedené téma.
4. Navržený systém implementujte formou prototypu. Správce nechť interaguje se systémem skrze WIFI rozhraní a mobilní telefon.
5. Vyhodnoťte a diskutujte parametry navrženého řešení. Použijte systém složený minimálně ze tří zařízení.

Literatura:

- Dle pokynů vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Splnění bodu 1 a 2 zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Vašíček Zdeněk, Ing., Ph.D.**, UPSY FIT VUT
Datum zadání: 1. listopadu 2016
Datum odevzdání: 17. května 2017

L.S.

prof. Ing. Lukáš Sekanina, Ph.D.
vedoucí ústavu

Abstrakt

Cílem této práce je vytvořit autonomní vestavěné zařízení umožnující řízení přístupu pomocí bezkontaktní identifikace (RFID). Zařízení má dále umožnit konfiguraci a přístup k záznamům o průchodech skrz webové rozhraní a získávání aktualizací databáze přístupových karet pomocí sítě se smíšenou topologií (mesh). K realizaci je využit vývojový kit na bázi Espressif ESP8266, který je naprogramován pomocí otevřené elektronické platformy Arduino. Vytvořené řešení, díky podpoře MESH sítí, umožňuje nasazení i v místech, která nejsou přímo pokryta bezdrátovou, či metalickou sítí.

Abstract

Aim of this work is to create autonomous embedded device which allows access control with radio frequency identification (RFID). Device is configurable by web interface and offers access history. Updates of RFID cards database are realised by mesh network. For realisation of this work is used developement kit based on Espressif Systems ESP8266 chip, which is programmed with open-source platform Arduino. Created device can be used in locations without direct network covrage.

Klíčová slova

Přístupový systém, ESP8266, NodeMCU, Wi-Fi, mesh, smíšená topologie

Keywords

Access control system, ESP8266, NodeMCU, Wi-Fi, mesh

Citace

TRUHLÁŘ, Jan. *Inteligentní přístupový systém pro větší objekty.* Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Vašíček Zdeněk.

Inteligentní přístupový systém pro větší objekty

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Zdenka Vašíčka, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jan Truhlář
15. května 2017

Poděkování

Tímto bych chtěl poděkovat Ing. Zdeňku Vašíčkovi, Ph.D. za jeho rady, trpělivost a velmi přátelskou atmosféru při vedení této práce.

Obsah

Kapitola 1

Úvod

Běžně používané přístupové systémy se síťovým připojením využívajícím metalickou kabeláž, či bezdrátové sítě v režimu infrastruktury, nejsou u již existujících objektů s nižší kvalitou síťového pokrytí ideálním řešením. Nasazení těchto přístupových systémů může znamenat nákladné úpravy síťové infrastruktury a v případě metalické kabeláže i rozsáhlé stavební úpravy.

S příchodem čipu ESP8266 a zejména možnosti jeho plné programové kontroly se objevily nové možnosti, jak realizovat bezdrátové přístupové systémy. Cílem této práce je vytvořit pomocí čipu ESP8266 autonomní vestavěné zařízení pro řízení přístupu, které bude pro komunikaci využívat bezdrátovou technologii Wi-Fi se smíšenou topologií (mesh). V tomto druhu bezdrátové topologie je každé zařízení zároveň klientem i přístupovým bodem, což umožňuje šíření dosahu bezdrátové sítě i do míst, která nejsou pokryta původní síťovou infrastrukturou.

Tato práce je členěna následovně. V první kapitole je analyzována problematika přístupových systémů, bezkontaktní identifikace, mesh sítí a podpůrných technologií, používaných k realizaci těchto systémů i této práce.

Druhá kapitola je zaměřená na analýzu platformy ESP8266. V kapitole jsou uvedeny základní parametry této platformy, dostupné softwarové vybavení a popis platformy Arduino.

Třetí kapitola se věnuje výběru hardware, návrhu zapojení a fyzické realizaci prototypu zařízení. Kapitola rovněž obsahuje informace o vybraných periferních zařízeních a knihovnách platformy Arduino, které jsou použity pro realizaci tohoto zařízení.

Poslední kapitola popisuje celkový koncept a proces implementace software. První podkapitola se věnuje celkovému konceptu přístupového systému. Dále následují části zabývající se realizací datového úložiště a přístupového systému. Poté následuje dvojice podkapitol, které se zaměřují na realizaci webového rozhraní správce s captive portálem, jeho optimalizaci a návrhu grafického uživatelského rozhraní. Následující podkapitola se věnuje vývoji mesh sítě a její optimalizaci. Poslední z podkapitol obsahuje celkové zhodnocení řešení, jeho výhody, nevýhody a navržená vylepšení.

Kapitola 2

Přístupové systémy na bázi bezkontaktní identifikace

Hlavním cílem této kapitoly je vytvořit ucelený teoretický základ zabývající se přístupovými systémy, bezkontaktní identifikací a sítěmi se smíšenou topologií. Dále jsou zde popsány používané technologie, protokoly a nastíněna problematika zabezpečení.

2.1 Elektronické přístupové systémy

Pojem přístupový systém lze definovat jako množinu bezpečnostních prvků, které společně umožňují zabezpečení určitých prostor proti přístupu neoprávněných osob.

Přístupové systémy[8] jsou založeny na různých technologiích, příkladem mohou být mechanické zabezpečovací prvky (zámky a klíče), či elektronické (radiofrekvenční identifikace, biometrické senzory). V reálných systémech jsou obvykle využívány kombinované bezpečnostní prvky, kde primární roli hraje elektronické zabezpečení, ale v případě poruchy nebo výpadku napájení je možné použití klíčů.

Elektronické přístupové systémy (viz obrázek 2.1) mají oproti mechanickým systémům značné výhody. Mezi největší z nich patří centralizovanost, jednoduchost změn přístupových oprávnění a automatické vytváření záznamů o průchodech s jednoznačnou identifikací osob.

Obrázek 2.1: Fotografie přístupového systému s RFID tagy.[1]

Tyto systémy jsou obvykle založeny na bázi mikrokontrolerů, či mikropočítačů, elektronicky ovládaných zámků a zařízení umožňujících jednoznačnou identifikaci osob.

[1] Převzato z: https://www.az-technik.cz/RFID-pristupovy-system-5ks-cip-d122.htm

V současné době jsou k identifikaci nejčastěji používány RFID čipy s unikátním identifikátorem, ale lze se setkat i s jinými způsoby ověření totožnosti. Příkladem může být identifikace pomocí biometrických systémů (otisky prstů, sken oční duhovky). [6]

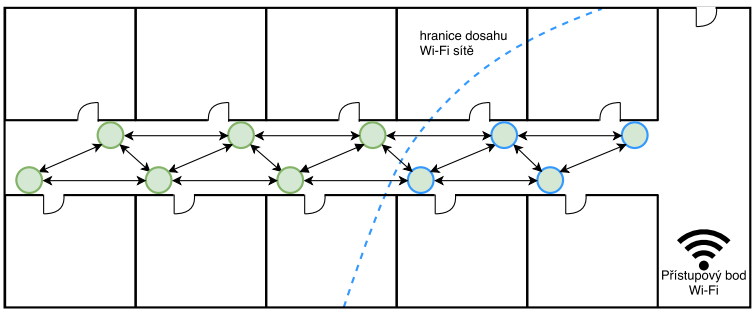
2.2 Přístupové systémy z pohledu síťového připojení

Běžně používané typy přístupových systémů je možné rozdělit do kategorií podle druhu síťového připojení, které používají. Rozdělení přístupových systémů podle tohoto kritéria je velmi důležité, jelikož typ připojení do značné míry určuje oblast použití tohoto systému a celkové náklady na jeho nasazení.

- **Systémy s metalickým připojením** – jedním z nejčastěji používaných typů jsou přístupové systémy využívající metalické síťové kabely[14], které jsou vhodné zejména v objektech se silným zarušením (průmyslové budovy) nebo v nově stavěných budovách. Výhodou je také možnost přímého napájení přístupového terminálu pomocí datového síťového kabelu (PoE), což snižuje náročnost nasazení. Nevýhodou jsou však značné náklady na vybudování vhodné síťové infrastruktury. Z tohoto důvodu je v mnoha případech výhodnější nasazení přístupového systému na bázi bezdrátové technologie Wi-Fi.

- **Bezdrátové systémy** – přístupové systémy založené na bezdrátové technologii Wi-Fi umožňují rychlé a levné nasazení bez nutnosti rozsáhlých stavebních úprav[1]. Nevýhodou je nutnost kvalitního síťového pokrytí všech částí objektu, ve kterých je tento systém nasazen a značné zatížení přístupových bodů sítě. Tato nevýhoda však může být odstraněna použitím mesh sítě.

- **Systémy s podporou mesh sítí** – mesh síť (viz kapitola 2.4.2) vytvořená z přístupových terminálů dokáže díky svým vlastnostem zprostředkovat (viz obrázek 2.2) přístup k mateřské síti i uzlům bez přímého pokrytí mateřskou Wi-Fi sítí. Tento přístup přináší značné výhody díky své jednoduchosti nasazení a rozšiřitelnosti, přináší však i nevýhody v podobě značně složitějších přístupových terminálů a nutnosti jejich rozmístění v takové vzdálenosti, aby mohli vzájemně komunikovat.



Obrázek 2.2: Ukázka šíření připojení pomocí mesh sítě. Uzly s modrým ohraničením jsou přímo pokryté a můžou sloužit jako síťové brány. Zeleně ohraničené uzly jsou pokryté nepřímo.

2.3 Bezkontaktní identifikace

S příchodem levných bezkontaktních čipů pro identifikaci zboží začalo být zřejmé, že bezkontaktní identifikace (RFID) má velký potenciál ke zrychlení nejen identifikace zboží, ale i osob a majetku. S rostoucím zájmem o tyto čipy docházelo k šíření bezkontaktní identifikace do mnoha oblastí, ve kterých napomáhá k vyšší efektivitě a bezpečnosti. [6, 16, 10]

2.3.1 Hardwarové vlastnosti

RFID čipy jsou vyráběny ve velkém množství variant, které se liší použitými technologiemi, technickými parametry i provedením čipu.

- **Aktivní RFID čipy** – jedná se o dražší variantu čipů, které obsahují vlastní napájecí zdroj. Díky vlastnímu napájení mohou být čipy identifikovány na velkou vzdálenost a mohou provádět i složitější operace s interní pamětí.

  Jejich využití spočívá převážně v aktivní lokalizaci na větší vzdálenosti. Příkladem může být využití ve sportu a mýtných branách [5, 20].

- **Pasivní RFID čipy** – tato varianta je značně levnější a rozšířenější. Pasivní čipy jsou napájeny elektromagnetickým polem vytvořeným čtecím zařízením.

  Nejčastější variantou čipů, se kterou se můžeme v běžném životě setkat, je forma nálepek, které se využívají k zabezpečení zboží v prodejnách. Druhou nejčastější variantou jsou čipové karty a klíčenky využívané k identifikaci osob.

Vzhledem k velkému množství typů RFID čipů a jejich celosvětovému používání vzniklo mnoho standardů, které definují jejich vlastnosti.

Standardní nosné frekvence: 125 kHz, 134 kHz a 13.56 MHz

Méně časté nosné frekvence: 915 MHz USA, 868 MHz Evropa

Délka unikátního identifikátoru: 96 bitů, 64 bitů, 32 bitů (MIFARE Classic)

Kapacita interní paměti: až desítky kB[10, s. 28]

Šifrovací protokoly: CRYPTO-1, 3DES, AES, PKE

2.3.2 Principy identifikace a autentizace

Pro identifikaci je nejčastěji využíván UID, což je unikátní číslo definované při výrobě čipu. Toto číslo by mělo být unikátní a neměnné, přesto se ale můžeme setkat s množstvím čipů, které umožňují změnu UID a tím umožňují vznik klonů RFID čipů. Tyto kopie představují bezpečnostní riziko, jelikož mohou být vytvořeny bez vědomí oprávněného držitele a to ze vzdálenosti několika desítek centimetrů. Aby bylo možné zabránit vytváření kopií RFID čipu, bylo zavedeno několik technik, které zabraňují jejich vytváření a umožňují jejich odhalení.

Za tímto účelem je nejčastěji používána interní paměť, která je spolu s UID součástí většiny RFID čipů, které jsou dnes využívány k identifikaci osob. Obsah této paměti je možné uživatelsky měnit a zabezpečit jej proti neoprávněnému přečtení. Jednotlivé bloky této paměti mohou být chráněny 6 bytovým přístupovým kódem[11], bez kterého není

umožněno čtení paměťového bloku. Tato ochrana zabraňuje úplnému klonování karty, jelikož není možné získat data uložená v chráněném sektoru bez znalosti jeho přístupového kódu. Dále také poskytuje možnost ověření pravosti karty pomocí odemčení sektoru a porovnání uložených dat.

2.3.3 Autorizace

Autorizace u jednoduchých přístupových systémů spočívá ve vyhledání přečteného UID v databázi karet a rozhodnutí, zda je držitel karty s tímto identifikátorem oprávněn ke vstupu, či nikoli. Tento způsob ověření však trpí značnými bezpečnostními riziky, viz kapitola 2.3.4.

Přístupové systémy s vyšší úrovní zabezpečení využívají ke komunikaci šifrované spojení mezi čtecím zařízením a RFID čipem, pomocí kterého ověřují platnost UID a následně i autenticitu karty pomocí dat uložených v její zabezpečené interní paměti. Šifrování je zde obzvlášť důležité, jelikož zamezuje odposlechnutí komunikace.

Třetí variantou[6] je použití interní paměti karty jako úložiště oprávnění pro přístup. Paměťové bloky v tomto případě obsahují data s přístupovými právy, která jsou šifrována symetrickou kryptografickou funkcí. Klíč pro dešifrování dat je v tomto systému uložen v každém z terminálů, aby bylo možné data dešifrovat a ověřit oprávnění. Výhodou tohoto systému je to, že nepotřebuje přístup k síti ani aktualizace.

2.3.4 Bezpečnost RFID

Mezi největší bezpečností rizika spojená s používáním RFID čipů patří možnost jejich neoprávněného klonování a tím i překonání zabezpečení přístupového systému. [10, s. 216]

Dalším možným vektorem útoku je hádání unikátních identifikátorů metodou hrubé síly (brute force), které může být realizováno pomocí specializovaného RFID modulu, který napodobuje chování karty a opakovaně odesílá nové UID. Tyto skutečnosti představují značná bezpečnostní rizika a je nezbytné je při implementaci přístupových systémů zohlednit.

2.4 Technologie používané pro realizaci přístupových systémů

Přístupové systémy používají velké množství podpůrných technologií, které umožňují jejich snadnou a efektivní realizaci. Mezi ně patří zejména technologie pro realizaci sítí, webových aplikací, komunikační protokoly a formáty. V této sekci jsou uvedeny a popsány technologie použité k realizaci této práce.

2.4.1 Bezdrátové připojení Wi-Fi

Jedná se o technologii bezdrátového přenosu dat[19] využívající frekvenční pásma 2.4 a 5 GHz. Tento typ připojení je definován standardem IEEE 802.11[13] a množstvím dodatků, které specifikují použité frekvence, kódování, zabezpečení a další klíčové parametry. Mezi největší výhody Wi-Fi sítí patří jednoduchost jejich nasazení, bezpečnost a vysoká přenosová rychlost.

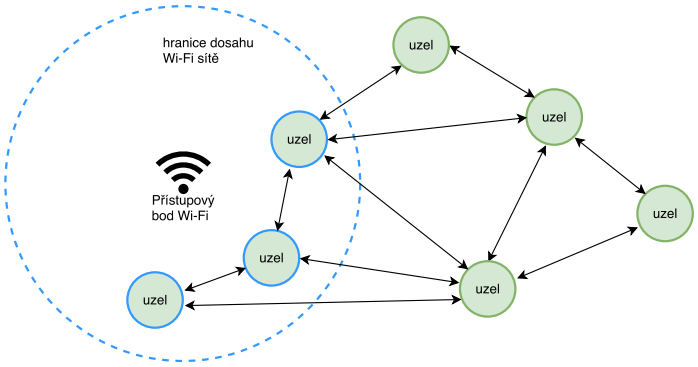
Další z důležitých vlastností je podpora ad-hoc režimu, který umožňuje vytvoření sítě nezávislé na existující infrastruktuře. To je dosaženo vzájemným propojením mezi jednot-

livými uzly, které se podílejí na směrování v rámci sítě. Tento režim je do značné míry podobný mesh sítím, ale není podporován všemi typy zařízení.

2.4.2 Smíšená topologie

Smíšená topologie[12, 7] (mesh) je druhem síťové topologie, ve které mohou být uzly přímo propojeny s více než jedním jiným uzlem v síti (viz obrázek 2.3).

Díky vícenásobným spojům je tento typ sítě odolný proti výpadkům jednotlivých uzlů a dokáže zprostředkovat připojení k síťové bráně i uzlům, které nejsou přímo pokryty.



Obrázek 2.3: Ukázka možné mesh sítě. Uzly s modrým ohraničením jsou přímo pokryté a mohou sloužit jako síťové brány. Zeleně ohraničené uzly jsou pokryté nepřímo.

K vytvoření Wi-Fi mesh sítě lze využít režimu ad-hoc, který umožňuje vzájemné propojení zařízení v decentralizované síti za účelem směrování a předávání dat. Existují však i jiná řešení fungující v odlišných pásmech rádiových vln. Příkladem může být systém ZigBee[2] pracující na frekvencích[18] 784 MHz (Čína), 868 MHz (Evropa), 915 MHz (USA a Austrálie).

Velmi důležitou vlastností mesh sítě je její schopnost rekonfigurace v případě výpadku některého z uzlů. Rekonfigurace zajistí dosažitelnost všech zbývajících uzlů, ke kterým vede alternativní trasa. Pokud k dané části sítě nevede žádná alternativní trasa (viz obrázek 2.4), stává se tato část nedostupnou. Proto je vhodné umístit jednotlivé uzly tak, aby existovala vždy alespoň jedna alternativní trasa.

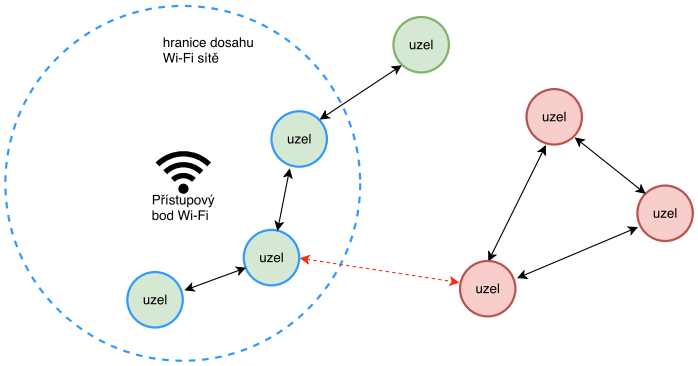
Předávání zpráv v rámci sítě může být realizováno pomocí dvou technik.

- **Záplavová technika (flooding)** – záplavová technika spoléhá na rozesílání obdržené zprávy na všechny uzly, ke kterým vede přímé připojení, kromě zdrojového uzlu. Tato technika šíření zpráv značně zatěžuje síť, ale její implementace je velice jednoduchá.
- **Směrování** – tato technika před odesláním vybere první uzel na trase k cíli, který po obdržení zprávy opakuje stejný postup. Tímto způsobem je zpráva prostupně předávána, až dorazí ke svému cíli. Tato technika je náročnější na implementaci, přináší však značné snížení zátěže sítě.

2.4.3 Webová aplikace

Webová aplikace je druh software založený na komunikaci typu klient-server, kde klientem je obvykle webový prohlížeč zobrazující grafické uživatelské rozhraní aplikace. Aplikace

[2] http://www.zigbee.org



Obrázek 2.4: Ukázka problematické mesh sítě. Červená přerušovaná čára naznačuje nefunkční spojení. Uzly s červeným ohraničením nejsou dostupné.

je obvykle založena na kostře ve formátu HTML, který doplňují další technologie jako kaskádové styly (CSS), dynamické skripty v jazyce JavaScript a mnohé další.

- **HTTP** – základem každé webové aplikace je server, který poskytuje její součásti skrz protokol HTTP nebo zabezpečenou variantu HTTPS. Standardně jsou za tímto účelem využívány porty TCP/80 pro HTTP a TCP/443 pro HTTPS. Server je obvykle konkurenční, dokáže tedy obsluhovat více klientů současně.

  Zpráva protokolu HTTP je rozdělena do dvou částí, kde první částí je hlavička obsahující tříciferný kód definující typ zprávy a další klíčové parametry. Druhou volitelnou částí jsou potom přenášená data.

- **HTML (HyperText Markup Language)** – je značkovací jazyk používaný pro tvorbu webových aplikací. Jeho využití spočívá ve vytváření koster webových aplikací a v případě statických stránek i k uložení obsahu. U dynamických webových aplikací je obsah doplněn až v okamžiku načítání požadované stránky klientem.

- **JSON** – neboli JavaScriptová objektová notace, je přenosový datový formát, který umožňuje serializaci polí a struktur. Výstupem serializace je textový řetězec, který může být snadno přenesen a v cílové aplikaci opět převeden na datové struktury.

  Tento formát není uzpůsoben pro přenos binárních dat, což vede k nutnosti použít některé z vhodných kódování pro převod binárních dat na posloupnost tisknutelných znaků.

- **Base64** – Tento typ kódování slouží k převodu binárních dat na tisknutelné znaky. To umožňuje jejich přenos skrz kanály, které umožňují pouze textovou komunikaci.

  Přináší však nevýhody v podobě většího objemu kódovaných dat, který je obvykle o 33% větší, než u originálu. Další nevýhodou je zatížení způsobené jejich kódováním a dekódováním.

- **Captive portál** – Je typ serveru, který přesměrovává všechny DNS dotazy na libovolné doménové jméno na svůj vlastní webový server. Díky tomuto chování je docíleno automatické zobrazení webové aplikace, což může být vhodné, pokud neznáme IP adresu nebo doménové jméno tohoto serveru.

- **Mezipaměť webového prohlížeče** Většina moderních prohlížečů obsahuje mezipaměť, která slouží k ukládání obsahu, který může být znovu použit k zobrazení stránky bez nutnosti opětovného načítání stejných dat. Za tímto účelem se používá široká škála položek HTTP hlavičky, které mohou určovat verzi souboru, čas expirace a mnohé jiné vlastnosti.

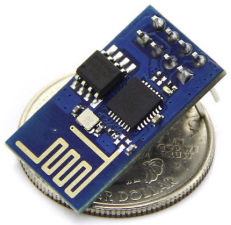
  Jedním z nejčastěji používaných způsobů identifikace souborů v mezipaměti je použití speciálních značek, takzvaných Etagů, které slouží jako unikátní identifikátor souboru. Etag je při prvním načtení získán od webového serveru a při opakovaných dotazech je tento tag odesílán serveru, který určí, zda je platný, či nikoli. Tímto způsobem je značně redukováno množství dotazů i přenesených dat.

Kapitola 3

Systém na čipu ESP8266

ESP8266[3, 4] je mikročip vyvinutý společností Espressif Systems, který byl primárně určen pro zprostředkování komunikace v bezdrátových sítích pracujících v pásmu 2.4 GHz. Jedná se de-facto o systém na čipu integrující 32 bitový RISC procesor s pracovní frekvencí 80 MHz, 64 KB instrukční paměti RAM a 96 KB datové paměti RAM. Detailní specifikace tohoto čipu jsou uvedeny v tabulce 3.1.

Na trhu se můžeme setkat s touto platformou většinou ve formě modulů (viz obrázek 3.1, které integrují čip ESP8266 a externí flash paměť. Kapacita flash paměti a další hardwarové vlastnosti se liší dle verze modulu a varianty.



Obrázek 3.1: Fotografie srovnávající velikost modulu ESP-01 a dolaru. [1]

V roce 2014 byla společností Espressif Systems uvolněna sada vývojových nástrojů (SDK) umožňující programování tohoto čipu a tím i otevřena možnost využít jej nejen jako modul pro připojení k Wi-Fi síti, ale rovněž jako plnohodnotný mikrokontrolér. To vedlo ke vzniku rozsáhlé komunity uživatelů, vývoji velkého množství knihoven, podpůrných programů a vývojových desek.

3.1 Periferní rozhraní

ESP8266 disponuje rozhraním, které zahrnuje 16 GPIO pinů, SPI rozhraní, sběrnici I^2C, I^2S s možností přímého přístupu do paměti, rozhraní UART s dedikovanými vývody a integrovaným 10 bitovým analogově-digitálním převodníkem.

- **GPIO** – periferní rozhraní tohoto čipu je tvořeno šestnácti GPIO vývody (General-purpose input/output), které mohou být nakonfigurovány jako vstupní, nebo výstupní.

[1] Převzato z: http://xanadu.khnet.info/esp8266.php

Kategorie	Vlastnost	Parametry
Wi-Fi	Standardy	FCC/CE/TELEC/SRRC
	Protokoly	802.11 b/g/n/e/i
	Frekvenční rozsah	2.4 – 2.5 GHz
	TX výkon	802.11 b:+20dBm
		802.11 g: +17 dBm
		802.11 n: +14 dBm
Hardware	CPU	Tensilica L106 32-bit micro controller
	Periferní rozhraní	UART/SDIO/SPI/I2C/I2S/IR ovládání
		GPIO/ADC/PWM/LED
	Digitální výstupy	16x GPIO
	Operační napětí	2.5 – 3.6 V
	Operační proud	Průměrně: 80 mA
	Provozní teplota	-40 °C – +125 °C
	Rozměry	QFN32-pin (5 mm x 5 mm)
Software	Wi-Fi režimy	Stanice, přístupový bod, stanice a přístupový bod
	Zabezpečení	WPA/WPA2
	Šifrování	WEP/TKIP/AES
	Sítové protokoly	IPv4, TCP/UDP/HTTP/FTP

Tabulka 3.1: Tabulka specifikací čipu ESP8266EX.

Nastavením GPIO pinů a jejich následnou softwarovou obsluhou je možné přímo komunikovat s jednoduchými periferními zařízeními pomocí změny logické úrovně, nebo softwarově emulovat sběrnice, jako je například I^2C.

- **SPI** – sériové rozhraní pro periferní zařízení je tvořeno čtyřmi vodiči s různým účelem, jeden z nich slouží jako rozvod hodinového signálu (SCK), dva slouží k výměně dat (MISO, MOSI) a poslední z nich k výběru periferního zařízení (SS).

  Tento čip disponuje dvěma SPI rozhraními, které při aktivaci využívají GPIO 6 (SCLK), dále pak GPIO 7, 8, 9, 10 jako datové vývody (MISO / MOSI), GPIO 0 a 1 pro výběr aktivního zařízení (SS). Frekvence hodinového signálu u tohoto rozhraní je maximálně 80 MHz.

- **UART** – jedná se o asynchronní vstupně výstupní sériové komunikační rozhraní s dedikovanými vývody. Rozhraní je nejčastěji používáno k nahrávání programu a dat do paměti zařízení, nebo jako sériová linka umožňující komunikaci. UART rozhraní tohoto čipu umožňuje provoz s maximální přenosovou rychlostí 460 800 Bd.

- **ADC** – analogově-digitální převodník (ADC) je druh vstupního rozhraní, které umožňuje převod analogové hodnoty napětí na binární reprezentaci. Rozhraní převodníku je tvořeno jedním vstupním pinem a umožňuje čtení hodnot v rozsahu 0–1 V.

- **I^2C** – jedná se o softwarově řízenou sériovou synchronní sběrnici využívanou k připojení nízkorychlostních periferií. Tento čip má pouze jedno I^2C rozhraní, které může být nastaveno do módu master, nebo slave.

  K provozu tohoto rozhraní jsou využity GPIO 14 (SCL) a 02 (SDA). Maximální podporovaná frekvence hodinového signálu u tohoto rozhraní je 100 kHz.

3.2 Softwarová podpora

Softwarová podpora ze strany společnosti Espressif Systems zahrnuje vývoj dvou oddělených větví specializovaného firmware, které poskytují základní funkce a rozhraní pro práci s touto platformou. Druhou část softwarové podpory pak tvoří specializované vývojové nástroje, které umožňují sestavení tohoto firmware, jeho nahrání do zařízení a další podpůrné operace.

3.2.1 SDK

Sada vývojových nástrojů (Software Development Kit), umožňuje sestavení programu využívaného pro řízení tohoto čipu. Každá z větví je založena na jiném technologickém principu a je tudíž vhodná k realizaci odlišných typů zařízení.

- **RTOS** RTOS je operační systém reálného času založený na systému FreeRTOS[2], který zaručuje, že budou požadované operace dokončeny ve stanoveném časovém limitu. Tento operační systém umožňuje vytváření pseudoparalelně běžících vláken s různou prioritou (celkem 14 úrovní, kde úroveň 0 značí nejnižší prioritu). Díky podpoře paralelismu je tato větev SDK vhodná zejména pro aplikace vyžadující souběžné provádění většího množství operací.
- **Non-OS** Non-OS je SDK nevyužívající operační systém. Tato varianta je založená na časovačích a voláních funkcí (callback). Díky odbourání režie na provoz operačního systému je možné s tímto SDK dosáhnout vyššího výkonu. Další výhodou je vyšší úroveň podpory ze strany společnosti Espressif Systems a komunity uživatelů.

3.2.2 Nástroje

Pro vývoj aplikací na této platformě jsou komunitou i společností Espressif Systems vyvíjeny nástroje, které umožňují veškeré operace nezbytné pro práci s touto platformou.

- **esptool** – jedná se o nástroj umožňující práci s flash a RAM pamětí modulů ESP. Tento nástroj umožňuje nahrávání, čtení a ověření paměťových bloků, získání informací o typech použitých čipů, MAC adresy a mnoho jiných informací. Možnost stažení aktuálního obsahu operační paměti a paměti flash je velmi přínosná při ladění kódu a opravě chyb.
- **mkspiffs** – nástroj určený pro vytváření a rozbalování obrazů souborového systému SPIFFS. Vytvořený obraz souborového systému je potřeba do zařízení nahrát pomocí utility esptool.
- **Xtensa lx106 architecture toolchain** – jedná se o sadu kompilačních nástrojů určenou pro sestavení programů v jazyce C / C++ pro platformu procesoru Xtensa. Sada je založena na základě GNU Compiler Collection, zkráceně GCC.

  Sada obsahuje i variantu nástroje GDB, který je určen k lazení programů, analýze chyb a zpracování výjimek této platformy. Za tímto účelem je možé GDB připojit k UART rozhraní tohoto čipu.

[2] http://www.freertos.org/

3.3 Platforma Arduino

Arduino[2] je open-source platforma založená na jazyce C++, která je zaměřena na jednoduché vytváření prototypů elektronických zařízení založených na mikrokontrolérech a mikroprocesorech. Obsahuje velké množství knihoven, které umožňují realizaci pokročilých operací, řízení specializovaných zařízení, poskytují metody pro práci s různorodými formáty a mnohé další. Použití těchto knihoven umožňuje velice rychlou a efektivní realizaci prototypů vestavěných zařízení.

Uvolněná verze Non-OS SDK byla komunitou uživatelů ESP8266 přepracována tak, aby ji bylo možné využít v prostředí Arduino. Tento krok umožnil použití již existujících knihoven a nástrojů, což značně zjednodušilo práci s touto platformou a vedlo ke vzniku rostoucí komunity uživatelů, která nadále přispívá k jejímu rozvoji.

Vývoj aplikací platformy Arduino se liší oproti běžným programům v jazyce C++ zejména v tom, že základní konstrukce programu nevychází pouze z hlavní funkce *main()*, ale je zde rovněž speciální funkce *setup()*, která je volána před spuštěním funkce *main()* a slouží k nastavení zařízení, inicializaci periferií a proměnných používaných v aplikaci.

3.3.1 Souborový systém SPIFFS

Moduly s ESP8266 mají externí flash paměť s kapacitou pohybující se od 0.5 MB do 4 MB. Část této paměti je určena pro řídící program, ale zbývající paměť může být využita pro uživatelská data.

Do flash paměti lze přistupovat přímo nebo využít některou z knihoven, které simulují funkce souborového systému. Takovou knihovnou je například SPIFFS, která je součástí základního balíku knihoven Arduino pro tuto platformu. Knihovna je optimalizována tak, aby umožnila co nejefektivnější práci se soubory a má rovněž implementovanou podporu vyrovnávání opotřebení paměťových buněk.

3.3.2 Arduino IDE

Arduino IDE je vývojové prostředí, které umožňuje jednoduché vytváření programů pro mikrokontroléry a mikroprocesory podporované touto platformou. Toto prostředí umožňuje jak vytváření programů, tak jejich kompilaci a nahrávání do paměti zařízení. Dále obsahuje nástroj pro komunikaci se zařízením pomocí sériové linky a pomocí specializovaných balíčků mohou být doplněny i další pokročilé nástroje.

V případě platformy ESP8266 je nezbytná instalace specializovaného vývojového balíčku[3], založeného na Non-OS SDK, který obsahuje všechny nezbytné součásti, nástroje a knihovny potřebné pro sestavení a nahrání programu do zařízení.

Dalším vhodným doplňkem je ESP8266FS[4], který přidává možnost použít Arduino IDE k vytváření a nahrávání obrazů souborového systému SPIFFS. Tento nástroj umožňuje použití složky s názvem data umístěné uvnitř adresáře projektu jako zdroje pro automatické vytvoření a nahrání souborů.

[3] https://github.com/esp8266/Arduino

[4] https://github.com/esp8266/arduino-esp8266fs-plugin

3.4 Podpora mesh sítí

Běžné Wi-Fi sítě jsou provozovány v režimu infrastruktury, což znamená, že v síti existuje minimálně jeden přístupový bod sloužící jako brána a jeden, nebo více klientů. Abychom mohli používat Wi-Fi sítě v decentralizovaném režimu, kde jednotlivé uzly komunikují mezi sebou, je nezbytné použít buď režim ad-hoc (viz kapitola 2.4.1), nebo toto chování emulovat pomocí současného spuštění Wi-Fi klienta a přístupového bodu.

Tato platforma bohužel režim ad-hoc nepodporuje, takže je jediným možným způsobem vytvoření mesh sítě emulace. Přesto existuje několik různých implementací, z nichž každá poskytuje jiné možnosti pro realizaci těchto sítí.

3.4.1 ESP-Mesh

ESP-Mesh je oficiální řešení s uzavřeným kódem, které je vyvíjeno společností Espressif Systems. Je určeno primárně k řízení zařízení spadajících do kategorie internetu věcí, jako jsou například inteligentní světla a regulátory vytápění. Součástí je i aplikace pro mobilní zařízení s operačním systémem Android, která umožňuje ovládání zařízení v síti.

Řešení vyniká svou kvalitou zpracování jak po programové stránce, tak po stránce dokumentace. Používá vlastní komunikační protokol, jehož zprávy jsou přenášeny pomocí protokolů TCP/IP. U zpráv lze definovat typ jejich obsahu, který může být HTTP, JSON, MQTT, nebo binární data.

Předávání zpráv probíhá pomocí techniky směrování, ke kterému jsou 6B adresy jednotlivých zařízení. Adresy mohou buď obsahovat MAC adresu zařízení, broadcastovou adresu, nebo IP adresu a port cíle, což umožňuje přímou komunikaci se zařízením mimo mesh síť.

Řešení však obsahuje značné nevýhody ve formě omezeného maximálního počtu zařízení a délky větve sítě. Celková velikost mesh sítě je omezena na maximálně 87 uzlů, při maximální délce větve sítě o délce 5 zařízení.

3.4.2 WiFiMesh-ESP8266

WiFiMesh je knihovna platformy Arduino, určená k vytváření mesh sítě pomocí současného běhu Wi-Fi klienta a přístupového bodu. Knihovna je součástí základního balíku Arduino pro vývoj na této platformě, obsahuje pouze elementární části, kterými jsou inicializace Wi-Fi klienta a přístupového bodu, vyhledání ostatních uzlů, připojení k uzlu s nejsilnějším signálem a callback funkci reagující na přijaté zprávy.

Ostatní pokročilé funkce jako směrování zpráv, vyhledávání nepokrytých uzlů a detekce výpadků uzlů nejsou implementovány. To může být nevýhodou z pohledu jednoduchosti vytvoření sítě, ale umožňuje implementovat síť přesně dle požadavků dané aplikace.

3.4.3 Knihovna painlessMesh

Další z knihoven určených pro použití s platformou Arduino je knihovna painlessMesh, která je vyvíjená skupinou dobrovolníků jako projekt s otevřeným kódem. Knihovna je plně připravena k jednoduchému vytváření mesh sítí ze zařízení ESP8266 a vyniká zejména svou propracovaností a kvalitou zpracování. Knihovna využívá vlastní komunikační protokol založený na datovém formátu JSON. Řídící i datové zprávy jsou přenášeny v tomto formátu pomocí protokolu TCP.

Tento formát neumožňuje přenos dat v binární podobě, je tudíž nezbytné tento nedostatek řešit použitím vhodného kódování, kterým je například Base64. To má však za následek vyšší režii při vytváření a přijímání datových zpráv a zvyšuje objem přenášených dat.

Knihovna umožňuje odesílání zpráv typu unicast pomocí techniky směrování a odesílání zpráv typu broadcast použitím záplavové techniky. To, že je implementována technika směrování, značně zvyšuje efektivitu přenášení zpráv tím, že redukuje zbytečnou komunikaci v rámci sítě.

Největší z výhod tohoto řešení však představuje jeho schopnost vytvářet mesh sítě o teoreticky neomezené velikosti. Limitem velikosti sítě je pouze množství dostupné operační paměti, ve které jsou uloženy seznamy uzlů a jejich přímých spojení. Další z nesporných výhod je pravidelné aktualizování mesh sítě a automatická rekonfigurace v případě výpadku některého z uzlů. Tyto vlastnosti činí knihovnu painlessMesh i navzdory nepříliš výhodnému formátu zpráv jasným favoritem pro použití k řešení této práce.

Kapitola 4

Realizace prototypu přístupového terminálu

Aby bylo možné teoretické poznatky o přístupových systémech a SoC ESP8266 prakticky použít k vytvoření vlastního přístupového systému, bylo nezbytné vytvořit vhodné zařízení, které by poskytlo základ pro realizaci všech softwarových komponent. Proces realizace zařízení byl rozdělen do několika částí, kde každá vytváří základ následující část.

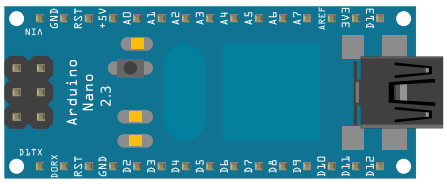
4.1 Výběr hardware

Před započetím výběru hardware jsem se zabýval vyhledáním a analýzou existujících přístupových systémů a podobných projektů na bázi Arduino. Zaměřil jsem se zejména na použité technologie, moduly, periferní zařízení a knihovny.

Prvním krokem byl výběr vhodného mikrokontroléru, který poskytuje základ pro realizaci celého zařízení. Poté následoval výběr periferních zařízení, které umožňují specializované činnosti.

4.1.1 Vývojová sada NodeMCU

V původním návrhu jsem plánoval použít modul ESP-01 pouze pro Wi-Fi komunikaci a veškeré operace s ním a modulem pro čtení RFID karet měly být realizovány pomocí jednodeskového mikrokontroléru Arduino Nano. S postupem analýzy a konzultací s vedoucím mé práce jsem původní návrh, vzhledem k jeho značným nedostatkům, opustil.



Obrázek 4.1: Mikrokontrolér Arduino Nano. [1]

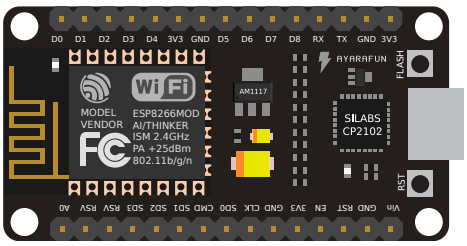
Mezi největší nedostatky předchozího řešení patřilo zejména to, že ke svému řízení potřebovalo oddělený mikrokontrolér. Během analýzy jsem zjistil, že lze tento čip použít jako

[1] Převzato z: https://github.com/fritzing/fritzing-parts/blob/master/svg/obsolete/breadboard/controller_arduino_nano.svg

mikrokontrolér s dostatečným výpočetním výkonem pro splnění všech požadavků zadání. V tomto bodě jsem rovněž opustil myšlenku použití modulu ESP-01 pro nedostatečný počet GPIO vývodů a nízkou kapacitu flash paměti.

Modulem, který jsem se rozhodl využít, je novější typ ESP-12E, který oproti ESP-01 disponuje mnohem vyšším počtem GPIO vývodů a množstvím flash paměti. Modul ESP-12E je integrován ve velkém množství vývojových desek, které usnadňují práci s tímto modulem.

Největším přínosem těchto vývojových desek je zejména to, že obsahují převodník z USB rozhraní na UART, který umožňuje snadné programování[9] tohoto modulu z USB portu. Vzhledem k této výhodě, velké uživatelské základně, kvalitě dokumentace a referencím na diskusních fórech zabývajících se programováním těchto modulů, jsem se rozhodl pro použití vývojové desky NodeMCU Devkit 1.0 (viz obrázek 4.2), která integruje modul ESP-12E s flash pamětí o kapacitě 4 MB.



Obrázek 4.2: Vývojová deska NodeMCU 1.0 s integrovaným modulem ESP-12E. [2]

4.1.2 Čtečka RFID karet RFID-RC522

Při průzkumu trhu a projektů s podobným zaměřením jsem narazil na modul RFID-RC522, který je založený na čipu MFRC522. Modul podporuje komunikaci pomocí rozhraní SPI, I^2C a UART. Bezdrátová komunikace probíhá dle standardu ISO 14443A na frekvenčním pásmu 13.56 MHz. Čtečka umožňuje práci s RFID čipy v režimu čtení a zápisu.

Obrázek 4.3: Fotografie sady RFID-RC522 a RFID čipů. [3]

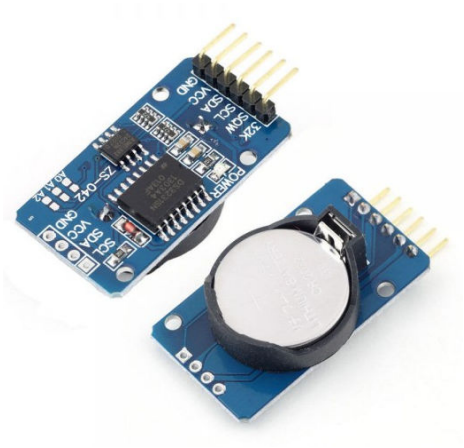
[2]Převzato z: http://fritzing.org/projects/simoria-1p2
[3]Převzato z: https://goo.gl/FqAv62

Po analýze dokumentace, uživatelských referencí a ověření kompatibility jsem se rozhodl pro jeho použití. Tento modul je v současné době prodáván na zahraničních serverech jako vývojová sada s několika RFID čipy za cenu přibližně $3 USD.

4.1.3 Modul reálného času DS3231

S postupující analýzou platformy ESP8266 a přístupových systémů jsem došel k závěru, že bude vhodné použít modul reálného času, který umožní synchronizaci spouštění mesh sítě a stahování aktualizací jednotlivých zařízení. Prvním modulem, jehož použití jsem zvažoval, byl DS1302[4]. Avšak při průzkumu referencí a existujících řešení jsem opakovaně narážel na zmínky o problémech způsobených časovacím krystalem, jehož nespolehlivost způsobuje výpadky. Naproti tomu byl často vyzdvihován modul DS3231 (viz obrázek 4.4) pro jeho jednoduché použití, stabilitu a přesnost. Z těchto důvodů jsem se rozhodl pro jeho použití.

Modul je založen na integrovaném teplotně kompenzovaném oscilátoru a krystalu, což umožňuje časování s vysokou přesností. Obsahuje pouzdro pro baterii typu CR2030, která zajišťuje napájení časového obvodu i v případě odpojení napájení. Komunikace s tímto modulem je realizována pomocí I^2C sběrnice.



Obrázek 4.4: Fotografie RTC modulu DS3231. [5]

4.1.4 Spínací modul HL-51

Pro řízení elektricky ovládaného zámku jsem se rozhodl použít spínací modul HL-51 (viz obrázek 4.5), protože umožňuje jednoduché použití tohoto zařízení jak s elektronickými zámky ovládanými signály, tak se zámky s přímým řízením.

Tento jednokanálový relé modul je určen pro spínání proudů o maximální velikosti 10A při 250V AC, nebo 30V DC. Modul je řízen pomocí změny logické úrovně na vstupu *In1*.

4.1.5 Napájecí měnič LM2596S

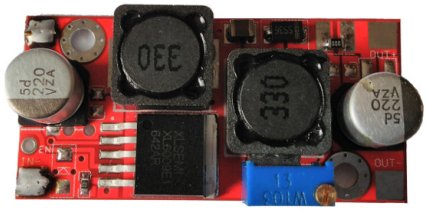
Zadání požaduje, aby bylo možné zařízení napájet pomocí střídavého zdroje 12 V, nebo 24 V. Pro dodržení těchto parametrů jsem se rozhodl použít napájecí měnič, který dokáže transformovat tato napětí na 3.3 V, což umožňuje přímé napájení celého zařízení.

[4] http://playground.arduino.cc/Main/DS1302

[5] Převzato z: http://arduino-shop.cz/arduino/1261-rtc-hodiny-realneho-casu-ds3231-at24c32-iic-pametovy-modul-pro-arduino-1459971336.html

Obrázek 4.5: Fotografie relé modulu HL-51. [6]

Jedná se o spínaný měnič napětí (viz obrázek 4.6), který pracuje v rozsahu vstupních napětí od 3.8 V do 32 V a umožňuje nastavení výstupního napětí v rozsahu od 1.25 – 35 V, při maximálním proudu 3A. Další velkou výhodou tohoto modulu je, že udržuje výstupní napětí na zvolené hodnotě i při změně napájecího napětí. Tato vlastnost umožňuje použití se zdrojem 12 V i 24 V, bez nutnosti přenastavení. Výrobce uvádí, že tento modul dosahuje účinnosti až 94%, což výrazně snižuje množství odpadního tepla, které vzniká při konverzi.



Obrázek 4.6: Měnič napětí LM2596S.

4.2 Knihovny Arduino pro řízení periferních zařízení

Jelikož platforma Arduino obsahuje velké množství různorodých knihoven pro řízení periferních zařízení, bylo nezbytné vybrat ty, které nejlépe umožní realizaci této práce. Při jejich výběru jsem čerpal zejména z analýzy podobných projektů, kterou jsem provedl při výběru periferních zařízení. Postupným srovnáním jednotlivých knihoven jsem nakonec určil ty, které považuji za nejvhodnější pro realizaci této práce. Ostatní knihovny umožňující například komunikaci po sběrnicích zde nejsou uvedeny, jelikož jsou součástí základního balíku knihoven Arduino a mají v této práci pouze podpůrnou roli.

- **RFID-RC522** – Knihovna RFID-RC522[7] je určena k řízení čteček RFID karet založených na čipu MFRC522 pomocí periferního rozhraní SPI a umožňuje práci v režimu čtení a zápisu pro různé typy RFID čipů. Nevýhodou však je, že knihovna nepodporuje pokročilé možnosti zabezpečení a šifrování komunikace s RFID tagy. Výjimkou je šifrovaný komunikační protokol Crypto-1, který je však v současné době překonaný[17] a neposkytuje téměř žádné zabezpečení komunikace.

  Pro její použití jsem se rozhodl, jelikož se jedná o nejpropracovanější a současně jedinou oficiálně podporovanou knihovnu, určenou pro platformu ESP8266.

[7] https://github.com/miguelbalboa/rfid

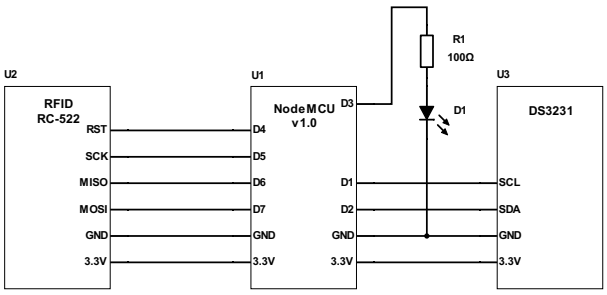
- **DS3231** – Knihovna DS3231[8] poskytuje plnohodnotné rozhraní pro řízení tohoto modulu pomocí sběrnice I^2C.

  Tuto knihovnu jsem zvolil zejména proto, že disponuje všemi potřebnými metodami pro práci s tímto zařízením a zároveň poskytuje velmi propracované metody ke zpracování datových formátů, časů a časových značek.

4.3 Propojení modulů

Po dokončení výběru modulů a knihoven bylo nezbytné navrhnout vhodný způsob jejich propojení. Při návrhu jsem vycházel z dokumentace vývojového kitu NodeMCU a jednotlivých modulů.

První verze zapojení (viz obrázek 4.7) obsahovala pouze moduly nezbytné pro realizaci základních prvků tohoto systému. Jako základní prvky jsem zvolil NodeMCU, RFID a RTC modul.



Obrázek 4.7: Schéma zapojení nepájivého kontaktního pole.

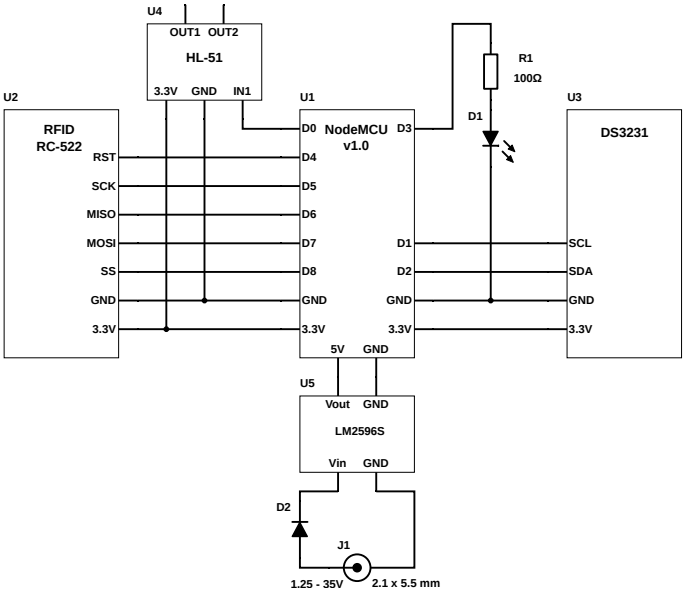
Navržené schéma jsem realizoval pomocí nepájivého kontaktního pole, ve kterém jsem propojil moduly s NodeMCU. Poté jsem provedl vizuální kontrolu správnosti zapojení a připojil hotový obvod k napájení pomocí USB portu NodeMCU. Jako zdroj napájení byl použit nabíjecí adaptér pro mobilní telefony s USB.

První část ověření funkčnosti spočívala v kontrole LED diod indikujících napájení jednotlivých modulů. Jelikož byly diody obou modulů rozsvícené, bylo zřejmé, že napájení modulů je funkční a je možné provést softwarovou kontrolu správnosti zapojení pomocí knihoven pro práci s moduly.

K softwarovému testování byly použity ukázkové kódy, které jsou součástí knihoven Arduino. Pro testování RTC modulu jsem zvolil ukázkovou aplikaci, která provádí inicializaci RTC modulu podle času sestavení programu a následně vypisuje každou sekundu aktuální čas na sériovou linku. Druhou testovací aplikací byl ukázkový kód, který umožňuje přečtení unikátního identifikátoru z RFID čipu.

V další části práce na prototypu bylo zapojení doplněno o zbývající periferní moduly, které jsou nezbytné k realizaci této práce. Jejich zapojení je naznačeno na obrázku 4.8.

[8] https://github.com/Makuna/Rtc

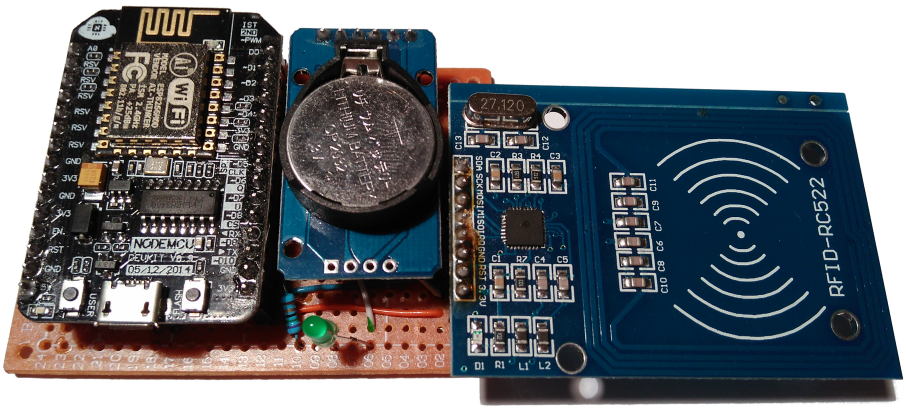


Obrázek 4.8: Schéma doplněného zapojení nepájivého kontaktního pole.

4.4 Sestavení prototypu

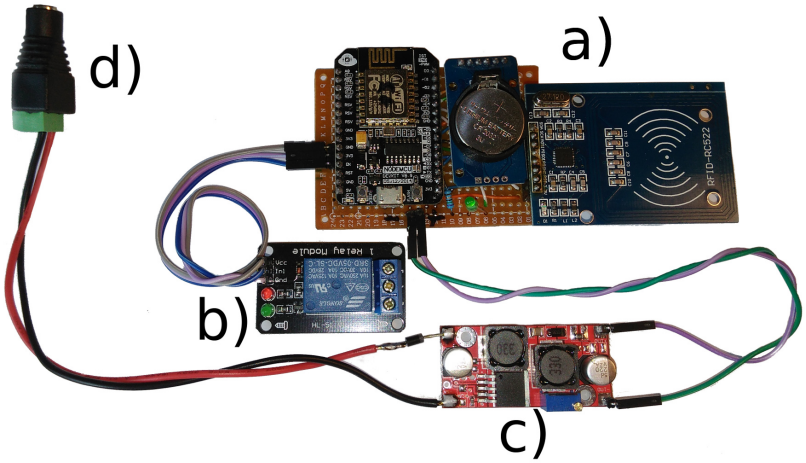
Po úspěšném otestování komunikace s moduly zapojenými na kontaktním poli jsem se rozhodl ověřené schéma realizovat pomocí pájecího kontaktního pole o velikosti 5×7 cm, které umožňuje rychlé vytvoření desky prototypu zařízení. Na tuto desku jsem umístil konektory, které umožňují snadnou výměnu modulů i vývojového kitu.

Po dokončení výroby jsem celou desku osadil a opakoval testy komunikace s periferiemi. Při těchto testech jsem detekoval problémy s komunikací přes sběrnici SPI. Pomocí digitálního multimetru jsem postupně ověřoval jednotlivé propojení, až jsem nalezl studený spoj, který znemožňoval přenos hodinového signálu a tím i komunikaci se zařízením. Následovala oprava tohoto spoje a opětovné testování, které již neodhalilo žádné problémy.



Obrázek 4.9: Fotografie sestaveného prototypu.

Další fází bylo rozšíření této desky o konektory a nová spojení, které jsem doplnil v druhém schématu zapojení (viz obrázek 4.8). Po tomto kroku a další sérii testů komunikace bylo již zařízení po elektrotechnické stránce kompletní a připravené vykonávat všechny požadované operace.



Obrázek 4.10: Fotografie kompletního prototypu s propojením jednotlivých modulů. a) Hlavní deska prototypu s klíčovými moduly. b) Spínací modul HL-51. c) Napájecí měnič LM2596S. d) Napájecí konektor IEC 60130-10 5.5 × 2.1 mm.

Poslední fází výroby prototypu bylo jeho umístění do plastové průmyslové krabičky (viz obrázek 4.11) o rozměrech 13 × 9 cm, která slouží k ochraně celého zařízení. Tato krabička je vhodná pro instalaci do zdiva.

Obrázek 4.11: Fotografie výsledného zařízení.

Kapitola 5

Implementace software

Po dokončení realizace prototypu zařízení jsem začal pracovat na návrhu a implementaci software umožňujícího řízení a správu tohoto zařízení.

Při návrhu implementace přístupového systému jsem zvažoval dva možné přístupy. Prvním z nich bylo nevyužívat interní databázi karet a záznamů o přístupech ve flash paměti ESP-12E, ale přistupovat skrz Wi-Fi připojení přímo k databázi na hlavním serveru. Tento přístup by přinášel výhody ve formě značně jednodušší implementace na straně terminálu, ale přinášel by nevýhody ve formě neustálého Wi-Fi vysílání, vyššího zatížení sítě a nepoužitelnosti v případě výpadku připojení nebo hlavního serveru. Z těchto důvodů jsem se rozhodl pro použití druhé varianty.

Tou je implementace s databází karet uloženou v interní paměti. Tento přístup odstraňuje nedostatky zmíněné v předchozí variantě, je však značně náročnější na hardware terminálu. Největší zatížení způsobuje pravidelné aktualizování interní databáze, vyhledávání v ní a ukládání záznamů o přístupech. Z tohoto důvodu byly v práci implementovány mechanizmy, jejichž cílem je redukovat opotřebení buněk flash paměti a časovou náročnost těchto operací.

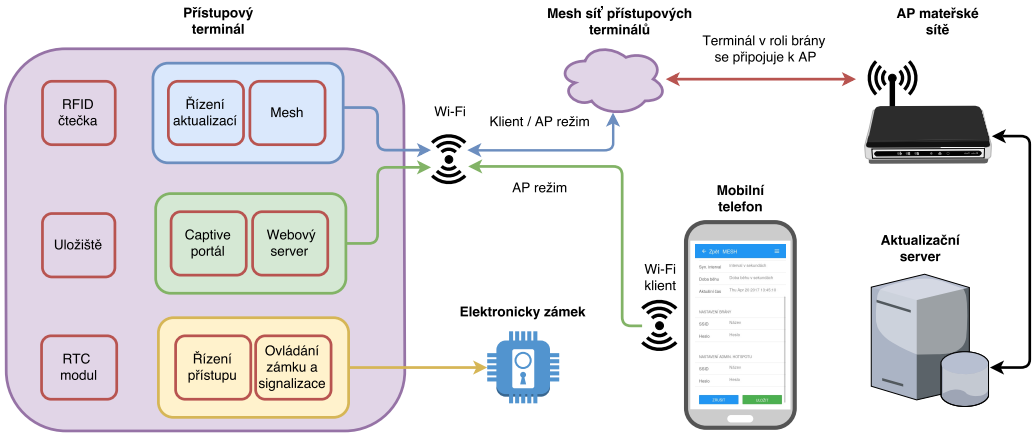
Pro vývoj jsem se rozhodl použít inkrementální přístup, kdy jsem vytvářel jednotlivé samostatně fungující bloky, ty následně testoval a propojoval do větších spolupracujících celků. Před započetím vývoje bylo nezbytné navrhnout celkový koncept a určit vhodné rozložení paměti, které by umožnilo realizaci všech následujících částí. Poté jsem se zaměřil na implementaci řízení přístupu a s ním spojených operací, jako je ukládání záznamů. Další v pořadí bylo vytvoření webového serveru, aplikace a její následná optimalizace. Po dokončení předchozího bloku jsem začal pracovat na vytvoření mesh sítě, komunikačního protokolu a celkové optimalizaci této části. V poslední fázi jsem se zabýval vyhodnocením výsledných vlastností tohoto systému a návrhem vhodných vylepšení.

5.1 Koncept přístupového systému

Přístupový systém je poměrně rozsáhlý a komplikovaný, proto považuji za důležité uvést koncept, který ozřejmí jeho celkovou funkci a spojení mezi jednotlivými částmi a moduly. Za tímto účelem je v této části uvedeno a popsáno několik schémat, jejichž cílem je usnadnit pochopení celého systému.

5.1.1 Celkový koncept

Celkový koncept (viz obrázek 5.1) je založen na kooperaci tří hlavních částí systému a tří podpůrných modulů. Těmito částmi jsou mesh síť, rozhraní správce a přístupový systém. K podpůrným modulům, které jsou používané všemi částmi, řadím RFID čtečku, datové úložiště a RTC modul.

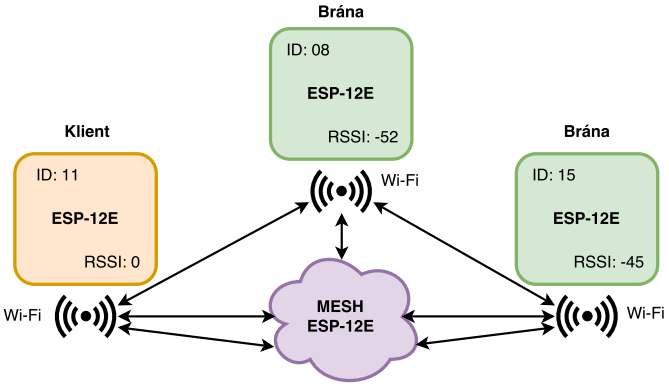


Obrázek 5.1: Schéma systému.

Na obrázku 5.1 jsou pomocí barev jednotlivých částí a spojů naznačeny spolupracující celky, které umožňují realizaci dané části systému. Důležité však je, že zeleně označené části a modře označené části nejsou nikdy v provozu současně. Modrá část (mesh síť) je aktivována pouze v době aktualizací, a to za podmínky, že není v provozu zelená část (webové rozhraní správce). Webové rozhraní správce je aktivováno pouze po přiložení administrátorského RFID čipu ke čtecímu zařízení. Žlutá část (přístupový systém) je v provozu neustále.

5.1.2 Koncept mesh sítě

Mesh síť (viz obrázek 5.2) vychází z principů zmíněných v kapitole 2.4.2 a je realizována z jednotlivých přístupových terminálů pomocí modulů ESP-12E. Uzly této sítě jsou současně přístupové body i klienti a do sítě se organizují samy podle síly přijímaného signálu sousedních uzlů.

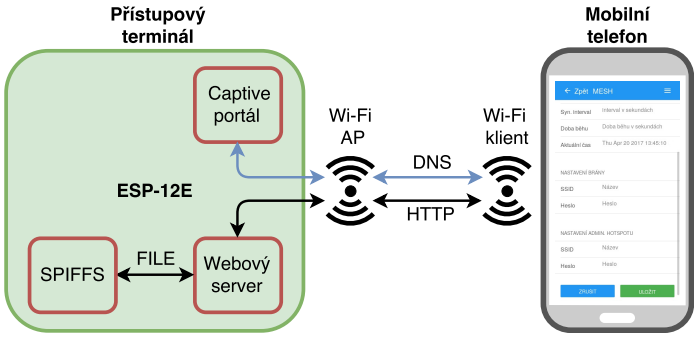


Obrázek 5.2: Schéma systému.

Na obrázku 5.2 je naznačena část hypotetické sítě se vzájemným propojením mezi jednotlivými uzly a propojení směrující do zbytku mesh sítě, která obsahuje ostatní uzly. Dále je u jednotlivých uzlů vyznačena kvalita signálu mateřské sítě (RSSI) a jejich role. Role uzlu je určena podle kvality RSSI, kde uzly s RSSI vyšším než -85 dBm mají roli brány. Kvalita signálu s hodnotou 0 označuje, že mateřská síť není z tohoto uzlu dosažitelná, což znamená, že může zastávat pouze roli klienta. Brána, kterou klient použije k získání aktualizací, je určena podle kvality RSSI.

5.1.3 Koncept činnosti rozhraní správce

Rozhraní správce umožňuje správci konfigurovat zařízení skrz webový prohlížeč. Za tímto účelem může být použit mobilní telefon, nebo jiné zařízení s Wi-Fi připojením.



Obrázek 5.3: Schéma činnosti rozhraní správce.

Na obrázku 5.3 je naznačen princip činnosti tohoto rozhraní a jednotlivých částí, které se podílejí na jeho realizaci. Dále jsou zde uvedeny spojení a protokoly, které jsou využívané k jejich realizaci.

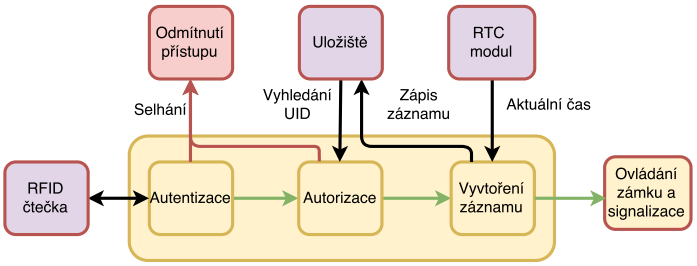
Nejdůležitější z částí je webový server, který umožňuje komunikaci a poskytování dat webové aplikace prohlížeči pomocí protokolu HTTP. Na základě komunikace pak realizuje veškeré operace nezbytné pro její chod. Webový server poté pracuje se souborovým systémem SPIFFS, který obsahuje data a zdrojové soubory webové stránky.

Druhou z částí je captive portál, který umožňuje po připojení zařízení k přístupovému bodu správce automatické zobrazení webové aplikace v prohlížeči. Přístupový bod správce, webový server a captive portál jsou během normálního provozu deaktivovány.

5.1.4 Koncept činnosti přístupového systému

Koncept přístupového systému (viz obrázek 5.4) je založen na reakci na čtení nového RFID čipu. Po jeho detekci je spuštěna série operací, které realizují řízení přístupů.

Řízení přístupu je realizováno pomocí autentizace čipové karty, která slouží k odhalení klonovaných karet, nebo podvržených dat. V případě úspěšného dokončení této operace následuje autorizace karty vyhledáním jejího UID (unikátního identifikátoru) v interní databázi karet. V případě nalezení shody je vytvořen a zapsán záznam o přístupu, tuto operaci následuje signalizace úspěšné autorizace a odemčení zámku. Jakákoliv z chyb v předchozích krocích znamená odmítnutí přístupu.

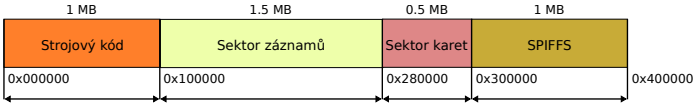


Obrázek 5.4: Schéma činnosti přístupového systému.

5.2 Realizace datového úložiště

Pro realizaci celého systému je nezbytné vytvořit vhodné rozdělení paměti, které by umožnilo efektivní ukládání všech potřebných dat a poskytlo dostatečný prostor pro všechny části systému. Celková kapacita flash paměti modulu ESP-12E je 4 MB, z této kapacity je 1 MB vyhrazen pro strojový kód aplikace. Zbývající prostor je možné použít k uložení uživatelských dat.

Pro implementaci tohoto terminálu jsem se rozhodl rozdělit zbývající 3 MB paměť na tři části (viz obrázek 5.5). První část paměti o velikosti 1.5 MB je využita pro uložení záznamů o přístupech, druhá část o velikosti 0.5 MB je určena pro databázi karet. Poslední z částí je přidělena souborovému systému SPIFFS a má velikost 1 MB. Souborový systém slouží k uložení webové aplikace, konfigurace a pomocných souborů.



Obrázek 5.5: Schéma rozložení flash paměti.

5.2.1 Databáze karet

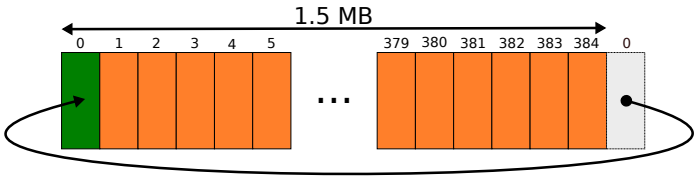
Při implementaci interní databáze karet jsem se potýkal s problémem životnosti paměťových buněk. Buňky flash paměti modulu ESP-12E mají životnost 100 000 přepsání a při krátkém intervalu aktualizací (například 10 minut) by mohla být dosažena mez jejich životnosti již za několik let.

Z tohoto důvodu jsem se rozhodl rozdělit tento paměťový blok na 8 částí o velikosti 64 KB, které jsou při aktualizacích pravidelně střídány, což přináší osminásobné zvýšení životnosti paměti. Velikost databáze (64 KB) považuji za dostatečnou, jelikož poskytuje prostor pro více než 16 tisíc karet u každých dveří. Povolené UID jsou v paměti uloženy jako pole záznamů typu uint32.

5.2.2 Historie přístupů

Aby byla zajištěna co nejdelší historie přístupů a maximalizována životnost buněk, bylo nezbytné zvolit vhodný formát záznamů a použít funkce pro přímou práci s pamětí. Uložení dat v tomto bloku paměti je realizováno jako pole struktur o dvou prvcích typu uint32, kde první obsahuje UID karty a druhý časovou značku přístupu. Maximální délka historie přístupů je 196 608 záznamů.

Díky použití přímé práce s pamětí je možné zapsat každý záznam, aniž by bylo nutné smazat celou paměťovou stránku a obnovit její obsah, což značně prodlužuje životnost paměťových buněk. Po naplnění paměti jsou sektory obsahující nejstarší historii postupně přepisovány (viz obrázek 5.6).



Obrázek 5.6: Diagram znázorňující využívání sektorů paměti ve smyčce.

5.2.3 Souborový systém SPIFFS

Poslední blok paměti je obsazen souborovým systémem SPIFFS o velikosti 1 MB. SPIFFS obsahuje soubory webové aplikace, konfiguraci a další pomocná data. Souborový systém je vytvořen pomocí aplikace makespiffs.

Výchozí konfigurace, data a soubory webové aplikace musí být do souborového systému nahrány před prvním spuštěním systému. K tomu je použit nástroj esptool, který umožňuje nahrání vytvořeného obrazu na zadanou adresu paměti.

5.3 Přístupový systém

Řízení přístupu je základní součástí přístupového systému, která je přímo spojená se všemi ostatními částmi systému. Realizuje základní operace, jako je autentizace čipu, autorizace a vytváření historie záznamů.

5.3.1 Autorizace

Základním principem autorizace je vyhledání unikátního identifikátoru čipu v aktuálně používaném bloku interní databáze karet. Tento přístup však sám o sobě není dostatečně bezpečný, jelikož může být unikátní identifikátor karty podvržen (viz kapitola 2.3.4). Pro omezení možnosti klonování RFID čipů a podvržení UID jsem se rozhodl využít možnost zabezpečení paměťových sektorů čipu pomocí přístupového kódu.

5.3.2 Autentizace čipu

Za účelem autentizace čipu jsem použil funkci knihovny MFRC522 *PCD_Authenticate()*, která umožňuje odemčení paměťového sektoru karty skrz šifrovaný komunikační protokol Crypto-1. V případě úspěšného přihlášení k paměťovému sektoru je umožněna autorizace karty. V opačném případě je komunikace s kartou ukončena a odmítnut přístup.

Tento způsob zabezpečení zamezuje přímému klonování karet bez znalosti přístupového kódu, ale vzhledem k tomu, že protokol Crypto-1 je v současné době překonán, může být komunikace mezi čtecím zařízením a čipem odposlechnuta, dešifrována a tím získán přístupový kód paměťového sektoru.

Pro ochranu před tímto typem útoku by bylo nezbytné využít některý z dalších šifrovacích protokolů, které tento typ čipu a čtecího zařízení umožňuje, ale vzhledem ke složitosti

implementace těchto protokolů a tomu, že je tato knihovna nepodporuje, nebyly tyto úpravy zapracovány.

5.3.3 Omezení opakovaného čtení čipu

Čtecí zařízení umožňuje identifikaci až několika karet za sekundu, což bez omezení způsobuje redundantní čtení karet, vytváření záznamu o přístupech a celkové zatěžování systému zbytečnými operacemi.

Za tímto účelem je implementována funkce *Check_repeat()*, která ověřuje, zda není karta čtena opakovaně v daném časovém intervalu. Výchozí interval je nastaven na 2 sekundy.

5.4 Rozhraní správce

Webová aplikace je založena na Framework7[15], který je určený k vytváření responsivních aplikací pro mobilní telefony a tablety. Při realizaci byly využity technologie zmíněné v kapitole 2.4.3.

Pro aktivaci rozhraní správce je nezbytné přiložit autentický RFID čip jehož unikátní identifikátor se shoduje s identifikátorem nastaveným při kompilaci. Po úspěšné autorizaci následuje vytvoření přístupového bodu Wi-Fi a spuštění webového serveru s aplikací umožňující správu.

Konfigurace zařízení se provádí skrz webovou aplikaci, která umožňuje nastavení všech nezbytných informací, jako je například SSID mateřské sítě, heslo pro přístup k této síti, název mesh sítě a IP adresu aktualizačního serveru. Při konfiguraci zařízení je rovněž provedena synchronizace času RTC modulu se zařízením, ze kterého bylo nastavení provedeno.

Konfigurace je po odeslání uložena do ve formátu JSON souboru config.json, který je umístěn v souborovém systému SPIFFS. Před provedením konfigurace soubor obsahuje výchozí hodnoty, které jsou nastaveny v okamžiku vytvoření souborového systému.

Aktuální konfigurace je rovněž uložena v globální struktuře *config*, která je využívána při řízení chodu systému.

5.4.1 Webový server

Server webové aplikace je realizován pomocí knihoven ESP8266WebServer a SPIFFS platformy Arduino. Implementace serveru je založena na callback funkcích, které reagují na požadavky klienta odesláním požadovaného souboru, nebo provedením operace.

Tyto callback metody jsou nastaveny ve funkci *Create_handlers()*, která určuje chování serveru při dotazech klienta. Dotazy typu GET jsou zpracovávány ve funkci *Handle_request()*, dotazy typu POST zpracovává funkce *Handle_post()*, která obdržená data vyhodnocuje a předává k dalšímu zpracování.

5.4.2 Captive portál

Pro usnadnění připojení k webovému serveru, běžícímu na tomto zařízení, byl vytvořen takzvaný captive portál, který umožňuje přesměrování všech požadavků na libovolná doménová jména na webový server tohoto zařízení. Tento přístup umožňuje připojení k webovému rozhraní správce i bez znalosti IP adresy serveru a doménového jména a u některých webových prohlížečů umožňuje i automatické zobrazení rozhraní správce po připojení k přístupovému bodu.

Za tímto účelem byla použita knihovna DNSServer, která umožňuje vytvoření jednoduchého DNS serveru. Ten poté odpovídá na všechny DNS dotazy svou vlastní IP adresou.

5.4.3 Princip komunikace se serverem

Aby bylo možné přistupovat k datům uloženým v blocích paměti pro historii přístupů a databázi karet pomocí webové aplikace, bylo nezbytné vytvořit vhodné funkce, které by realizovaly tyto operace.

Za tímto účelem byly vytvořeny dvě specializované callback funkce, které emulují existenci souborů cards.json a log.json v kořeni webové aplikace. Tato emulace je vytvořena na základě proudového převodu požadovaného bloku paměti do formátu JSON, který realizují funkce *Read_cards()* a *Read_log()*.

Formát JSON (viz kapitola 2.4.3) je určen pro přenos dat a byl zvolen pro svou přímou podporu programovacím jazykem JavaScript, který je použit pro zpracování a zobrazení dat uživateli. Převod do tohoto formátu jsem zvolil, aby byla odstraněna nutnost parsovat binární data ve webovém prohlížeči.

5.4.4 Optimalizace rozhraní správce

Vzhledem k velmi omezenému množství paměti, které je dostupné pro soubory webové aplikace a nízkým přenosovým rychlostem, bylo nezbytné použít vhodné optimalizační techniky pro minimalizaci velikosti webové aplikace. S nasazením těchto technik bylo nezbytné provést úpravy webového serveru, jelikož knihovna ESP8066WebServer použitá k realizaci tohoto serveru neobsahuje metody, které by umožnily použití těchto optimalizací.

- **Optimalizace kaskádových stylů Framework7** – z důvodu velké paměťové náročnosti Framework7 (přibližně 4.7 MB) jsem se rozhodl pro použití pokročilých optimalizačních technik, které umožňují výrazné snížení velikosti zdrojových souborů webové aplikace a tím nejen šetření paměti, ale i značné zrychlení načítání.

  Pro první fázi optimalizace jsem použil nástroj PurifyCSS[1], který dokáže analyzovat zdrojové kódy webové aplikace a odstranit pravidla kaskádových stylů, které nejsou v aplikaci využity. Dalším přínosem je sloučení optimalizovaných souborů, což snižuje dobu načítání aplikace díky redukci počtu požadovaných souborů.

  Tato optimalizace umožnila celkové zmenšení kaskádových stylů z původní velikosti 1.6 MB na 307 KB, což představuje necelých 20% původní velikosti.

- **Komprese webové aplikace** – pro další optimalizaci využitého paměťového prostoru a zvýšení rychlosti načítání webové aplikace jsem se rozhodl komprimovat všechny soubory webové aplikace pomocí programu gzip, který je přímo podporován webovými prohlížeči. Po úpravě serveru tato vlastnost umožňuje přímé odesílání komprimovaných souborů, bez nutnosti dekomprimace na straně terminálu.

  Původní velikost webové aplikace s optimalizovanými kaskádovými styly byla 1.4 MB, po kompresi všech souborů je výsledná velikost pouhých 128.5 KB, což představuje pouhých 2.7% původní velikosti. Díky těmto optimalizacím bylo možné využít pro uložení celé webové aplikace SPIFFS o velikosti 1 MB.

[1] https://github.com/purifycss/purifycss

- **Úprava serveru** – knihovna, která byla použita k vytvoření tohoto webového serveru, nepodporuje použití komprimovaných souborů. Z tohoto důvodu bylo potřeba implementovat tyto operace v rámci callback funkce *Handle_request()*, která realizuje reakci na GET požadavky.

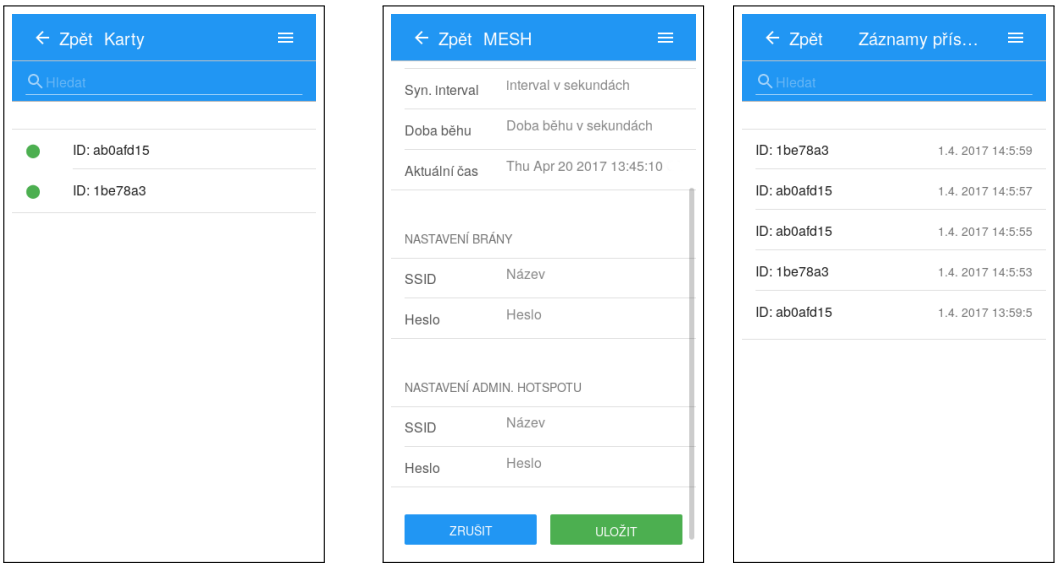
  Úprava spočívá zejména v nastavování hlaviček HTTP protokolu pro indikaci použité komprese webovému prohlížeči a přepisování názvů souborů. Přenášení souborů webové stránky je díky této úpravě realizováno v komprimované podobě, což významně snižuje množství přenášených dat a zvyšuje rychlost načítání.

- **Mezipaměť webového prohlížeče** – Pro odstranění redundantních dotazů na již odeslané soubory, jsem se rozhodl implementovat na straně webového serveru podporu HTTP Etagů, které fungují jako identifikátor pro vyhledání ve webové mezipaměti prohlížeče a označují verze souborů.

  Tato úprava spočívá v porovnání Etagu obdrženého v hlavičce GET požadavku webového prohlížeče, který je uložen v položce If-None-Match. V případě shody je odeslán HTTP kód 304, který signalizuje, že soubor v mezipaměti je platný a má být použit. V opačném případě je odeslán HTTP kód 200, který je následován požadovaným souborem. Jako hodnotu identifikátoru Etag jsem použil velikost požadovaného souboru v bytech.

5.5 Uživatelské rozhraní

Při vytváření grafického uživatelského rozhraní jsem se zaměřil zejména na jednoduchost a účelnost. Celá aplikace je rozdělena do pěti stránek s minimalistickým rozhraním, kde tři slouží ke specializovaným činnostem, jedna jako hlavní rozcestník a poslední obsahuje informace o zařízení.



(a) Stránka s kartama. (b) Stránka s konfigurací. (c) Stránka s historií přístupů.

Obrázek 5.7: Snímky s ukázkou jednotlivých hlavních částí GUI.

Hlavní rozhraní (viz obrázky 5.7) tvoří záhlaví s názvem stránky a tlačítka pro otevření postranní nabídky a návrat na předchozí stránku. Pod záhlavím následuje hlavní část obsahující obsah specifický pro danou stránku.

- **Rozhraní pro prohlížení historie přístupů** – k tomuto účelu slouží stránka logs.html, která zobrazuje chronologicky seřazený seznam přístupů. Pro usnadnění je na stránce umístěn vyhledávač přístupů podle identifikátoru karty.

  Alternativním způsobem je přímý přístup k souboru log.json, který může být uložen a následně využit pro pokročilejší analýzu záznamů o přístupech.

- **Rozhraní pro prohlížení databáze karet** – pro prohlížení interní databáze karet je možné využít stránku cards.html, která prezentuje aktuálně platnou databázi karet s povoleným přístupem.

  Pro přímý přístup k databázi karet je možné využít soubor cards.json, který obsahuje interní databázi karet převedenou do formátu JSON.

5.6 Realizace mesh sítě

Realizace mesh sítě je jedním ze základních prvků tohoto systému, jelikož umožňuje získávání aktualizací databáze karet, který je nezbytná pro chod přístupového systému. Pro její realizaci jsem se rozhodl, vzhledem k velkému počtu výhod zmíněných v kapitole 3.4.3), použít knihovnu painlessMesh.

Knihovna painlessMesh však obsahuje pouze základ pro realizaci mesh sítě a textové komunikace mezi uzly, což znamená, že je nezbytné vytvořit vlastní komunikační protokol a metody pro obsluhu přijatých zpráv, jejich vytváření a kódování pro přenos binárních dat. Dále je nezbytné implementovat specializované operace jako je odpojování od mesh sítě a připojení k mateřské síti, komunikace se serverem, stahování aktualizací a mnohé další operace, které umožňující operace naznačené v konceptu.

5.6.1 Knihovna painlessMesh

Knihovna painlessMesh[2] funguje na bázi nešifrovaného TCP spojení a zpráv ve formátu JSON. Tento formát není k tomuto účelu ideální, zejména pro svou objemnost a problematický přenos binárních dat, která musí být překódována na tisknutelné znaky (například pomocí Base64). Tato nevýhoda je však převážena jednoduchostí vytvoření mesh sítě o rozměrech limitovaných pouze množstvím volné operační paměti. Knihovna dále vyniká stabilitou, vysokou úrovní propracovanosti a podporou ze strany vývojářů.

Tato knihovna funguje na základě současného provozu Wi-Fi klienta a přístupového bodu. Jednotlivé uzly při spuštění mesh sítě vyhledají přístupové body ostatních uzlů a připojí se k těm, ke kterým ještě neexistuje žádná trasa a mají nejvyšší sílou přijímaného signálu (RSSI). Pokud již existuje trasa ke všem nalezeným uzlům, vybere se uzel s nejsilnějším RSSI. Tento proces je pravidelně opakován i během chodu sítě, aby byla zajištěna rekonfigurace v případě výpadků a připojení nových uzlů.

Zasílání zpráv typu unicast je řešeno pomocí techniky směrování, kdy je před odesláním vyhledána trasa vedoucí k cíli a data jsou odeslána na první uzel trasy. Tato operace je zopakována každým uzlem na trase k cíli, dokud není zpráva doručena.

[2] https://gitlab.com/BlackEdder/painlessMesh

Zasílání zpráv typu broadcast je realizováno záplavovou technikou, při které je zpráva z uzlu rozeslána na všechny uzly s přímým připojením, s výjimkou uzlu, ze kterého byla zpráva obdržena.

5.6.2 Aktivace a deaktivace mesh sítě

Při implementaci mesh sítě pomocí knihovny painlessMesh bylo největším problémem to, že knihovna neumožňuje deaktivaci mesh sítě a následné obnovení. Z tohoto důvodu bylo nezbytné analyzovat zdrojový kód knihovny a vytvořit funkce, které umožní provádění těchto operací. Za tímto účelem byly vytvořeny následující funkce.

- *Connect_to_AP()* – tato funkce provede uložení všech důležitých nastavení mesh sítě a změní konfiguraci Wi-Fi klienta pro připojení k mateřské síti.
- *Disconnect_from_AP()* – funkce obnoví uložená nastavení Wi-Fi klienta a umožní opětovné připojení k mesh síti.

Při jejich vývoji jsem opakovaně narážel na problémy s opětovným připojením uzlů k mesh síti, které byly způsobené nekompatibilitou knihoven painlessMesh a ESP8266WiFi, která byla použita pro připojení k Wi-Fi síti. Z tohoto důvodu jsem tuto knihovnu přestal používat a funkce realizoval pomocí funkcí SDK pro přímé řízení Wi-Fi modulu.

5.6.3 Komunikační protokol

Pro zajištění komunikace mezi uzly jsem navrhl komunikační protokol fungující způsobem dotaz-odpověď. Pro určení typu dotazu je využit tříciferný číselný kód, který může být následován dvojtečkou a daty, které daná zpráva přenáší.

Informace o používaných kódech a jejich významu jsou uvedeny v tabulce 5.1.

Kód	Název	Popis funkce
100	INFO_RSS	Uzel informuje o kvalitě přímého pokrytí.
101	REQUEST_RSSI	Uzel požaduje informace o kvalitě pokrytí ostatních uzlů.
102	INFO_ALIVE	Uzel informuje o svém stavu.
200	DATA_REQUEST	Uzel požaduje po příjemci aktualizační data. Součástí zprávy je sekvenční číslo.
201	DATA_RESPONSE	Uzel posílá aktualizační data příjemci. Součástí zprávy je sekvenční číslo.
202	DATA_CACHE	Uzel informuje příjemce, že má v mezipaměti jemu určená aktualizační data.
203	DATA_CACHE_FLUSH	Uzel odmítá data v mezipaměti a iniciuje jejich odstranění.
204	DATA_REFUSE	Uzel má problémy s připojením a žádá o odstranění ze seznamu bran.

Tabulka 5.1: Tabulka kódů komunikačního protokolu.

5.6.4 Role uzlů v mesh síti

Při provozu mesh sítě mohou uzly pracovat ve dvou rolích.

- **Klient** – V této roli je uzel, který není pokryt mateřskou sítí. Uzel vysílá broadcast žádost REQUEST_RSSI a seřazuje odpovědi podle kvality signálu. Pro aktualizaci následně využije bránu s nejvyšší kvalitou připojení.
- **Brána** – V roli brány je uzel, který je přímo pokryt mateřskou Wi-Fi sítí. Uzel reaguje na dotazy REQUEST_RSSI broadcast zprávou INFO_RSSI, která obsahuje informaci o síle signálu mateřské sítě.

  Brána nevyužívá k získání aktualizačních dat mesh síť, ale připojuje se přímo k mateřské síti a serveru. Tím je značně redukována zátěž mesh sítě a urychlena aktualizace uzlů.

5.6.5 Aktualizační server

Aktualizační server je implementován v jazyce Python, což umožňuje jeho nasazení na různých platformách. Server naslouchá na TCP portu 54320 a reaguje na platné žádosti klientů odesláním fragmentu souboru s aktualizacemi. Požadavek klienta je složen ze dvou částí oddělených dvojtečkou. První z částí je identifikátor klienta požadujícího aktualizaci, druhá je sekvenční číslo fungující jako offset, který určuje část souboru, která má být odeslána. Délka odesílaného fragmentu je 1 KB. Jména souborů s aktualizacemi jsou shodná s identifikátory uzlů. Identifikátor uzlu je odvozen z unikátního výrobního čísla modulu ESP.

Server byl implementován jako samostatně běžící program mimo mesh síť, protože jednotlivé moduly ESP-12E nedisponují dostatečným výkonem ani paměťovou kapacitou pro tento úkol. Dále tento přístup přináší výhody v tom, že může být v rámci budov, či budovy obsahující více samostatných mesh sítí, které využívají společný aktualizační server a tím umožňují mnohem jednodušší centralizovanou správu.

5.6.6 Plánování a získávání aktualizací

Aktualizace jsou plánovány podle hodnot proměnných *config.runtime*, *config.interval* a dat z modulu reálného času. První z proměnných určuje dobu běhu mesh sítě v sekundách, po jejímž vypršení je síť vypnuta. Druhá určuje časový interval, po kterém bude síť znovu spuštěna a požadována aktualizace interní databáze.

Programová realizace je založena na zprávách komunikačního protokolu definovaném v kapitole 5.6.3. Realizace vychází ze dvou hlavních funkcí, kterými jsou *Create_request()* a *Receive()*. Funkce *Create_request()* slouží k vyhledání vhodného uzlu v roli síťové brány a vytvoření požadavku aktualizace. Funkce *Receive()* slouží ke zpracování všech přijatých zpráv od uzlů v mesh síti a v reakci na jejich přijetí spouští specializované obslužné funkce, které obsahují realizaci požadované operace. V případě aktualizace se jedná o funkci *Download_data()*, která umožňuje stažení aktualizačních dat do mezipaměti a jejich následné odeslání klientovi. Celkově je systém realizován pomocí 25 funkcí, které realizují jednotlivé operace a reakce na zprávy.

5.6.7 Přidání UID a vyhledání držitele

Pro přidání nového povoleného UID pro daný přístupový terminál je nezbytné doplnit tento UID do souboru se jménem shodným s identifikátorem terminálu, který se nachází ve složce

aktualizačního serveru a obsahuje kompletní databázi karet. Současně je nezbytné aktualizovat časovou značku na začátku tohoto souboru. Zároveň by měl být doplněn záznam do souboru identity.json umístěného ve složce serveru, který obsahuje databázi unikátních identifikátorů a jmen. Druhou důležitou operací je vyhledání držitele podle UID.

Obě tyto operace jsou realizovány pomocí skriptu manage.py, který je rovněž umístěn v kořenové složce serveru. Tento způsob přidávání karet byl zvolen pro lepší centralizovanost. Vyhledávání v databázi jmen je možné pouze pomocí tohoto skriptu, jelikož terminály nedisponují dostatkem volné paměti pro udržování kompletní databáze UID a jmen.

5.6.8 Optimalizace mesh sítě

Jelikož je opakované přepojování mezi mateřskou sítí a mesh sítí časově velmi náročné, rozhodl jsem se do implementace mesh sítě zapracovat mechanizmy, které omezují zbytečné operace, redukují množství přenášených dat a celkovou časovou náročnost aktualizací.

- **Detekce prázdného prostoru** – Pro snížení zátěže mesh sítě a zrychlení aktualizací je v aktualizačním serveru implementován mechanizmus, který kontroluje, zda nejsou požadovány fragmenty aktualizačního souboru obsahující prázdný prostor. V případě, že je takovýto požadavek detekován, je odeslán fragment obsahující binární nuly, čímž je klientovi signalizován konec aktualizačních dat. Klient po obdržení tohoto fragmentu ukončí stahování aktualizačních dat a zbývající fragmenty nahradí prázdným prostorem.

- **Mezipaměť aktualizací** – z důvodu velké časové náročnosti připojení k aktualizačnímu serveru jsem se rozhodl implementovat mezipamět, která umožňuje stažení několika následujících fragmentů aktualizace v rámci jednoho připojení. Počet těchto předběžně stažených fragmentů závisí na množství neobsazené operační paměti brány.

  Maximální množství paměti použité za tímto účelem činí 40% volné kapacity. Z optimalizačních důvodů je paměť automaticky vyprázdněna po dvou minutách.

- **Časová značka aktualizace** – aby nedocházelo k opakovanému stahování a přepisování interní databáze karet v případech, kdy její data nebyla aktualizována, je na začátku prvního fragmentu dat uvedena časová značka o délce 32 bitů, která je před přepsáním dat porovnána se současnou časovou značkou interní databáze.

  V případě shody je interní databáze aktuální a stahování aktualizačních dat může být ukončeno. Tento krok je doprovázen odesláním zprávy DATA_CACHE_FLUSH, která informuje bránu, že může odstranit nepotřebná aktualizační data z mezipaměti.

5.7 Vyhodnocení parametrů navrženého zařízení

Tato část práce se věnuje analýze výsledného řešení a poskytuje ucelené shrnutí jeho klíčových parametrů. Dále jsou pak zhodnoceny výhody, nevýhody a navrženy kroky pro zlepšení.

5.7.1 Proces analýzy

Pro analýzu vlastností byla použita skupina tří zařízení, která byla tvořena prototypem a dvěma zařízeními NodeMCU. Všechna zařízení byla současně naprogramována identickým kódem, aby byly zajištěny stejné výchozí podmínky.

Při vyhodnocování parametrů jsem se zaměřil zejména na klíčové vlastnosti jako je rychlost sestavení mesh sítě, její propustnost, dosah, kapacita pro záznamy o přístupech a úroveň zabezpečení.

- **Rychlost sestavení mesh sítě** – za tímto účelem jsem použil jednoduchý program pro měření času v milisekundách od spuštění mesh sítě, po úspěšné připojení. Po shromáždění dostatečného množství údajů jsem došel k závěru, že sestavení sítě trvá přibližně 15 sekund.

- **Propustnost mesh sítě** – pro otestování této vlastnosti jsem mezi uzly v rámci mesh sítě opakovaně odesílal blok náhodných dat o velikosti 1024 bytů. Měřením časové prodlevy mezi odesláním a doručením jsem určil, že přenosová rychlost mesh sítě kolísá v rozsahu 96-228 kb/s. Výsledná přenosová rychlost není vysoká, ale pro přenos řídících zpráv a aktualizačních dat je dostatečná.

- **Dosah mesh sítě** – pro měření dosahu přístupového bodu vytvořeného pomocí prototypu zařízení jsem použil mobilní telefon s operačním systémem Android a aplikací Wifi Analyzer[3]. Tato aplikace umožňuje měření síly signálu přístupových bodů v reálném čase.

  Měřením jsem určil, že dosah přístupového bodu v chodbě bez překážek se pohybuje mezi 20 až 30 metry. Měření dosahu skrz stěny je neprůkazné, jelikož velmi záleží a tloušťce stěn a použitém stavebním materiálu.

- **Maximální délka záznamů o přístupech** – maximální počet záznamů lze vypočítat jako podíl velikosti paměti pro záznamy (1.5 MB) a velikosti jednoho záznamu o přístupu (8 B). Výsledkem je hodnota 196 608 záznamů.

  Tento limit nemusí být v některých případech dostatečný, a proto je navrženo možné řešení tohoto problému v části, která se věnuje navrhovaným vylepšením.

5.7.2 Vlastnosti navrženého řešení

Mezi největší výhody tohoto řešení patří jeho schopnost vytvářet mesh sítě o velikosti a maximální délce větve limitované pouze množstvím volné operační paměti. To umožňuje nasazení například v budovách s dlouhými chodbami a velkým počtem přístupových terminálů. Mezi další výhody patří například nízká výrobní cena zařízení (celkově přibližně 250 Kč), jednoduchost a malé rozměry. Toto řešení však má i svá omezení, která jsou v následující části zhodnocena.

- **Zabezpečení komunikace v mesh síti** – značným omezením je, že řešení nepodporuje šifrování přenosů pomocí SSL, čímž je mesh síť v případě překonání zabezpečení Wi-Fi sítě zranitelná. Největším rizikem je v tomto případě možnost získání databáze karet útočníkem, či narušení chodu mesh sítě podvrženými zprávami.

  Pro odstranění bezpečnostních nedostatků mesh sítě by bylo nezbytné implementovat šifrování pomocí SSL v rámci knihovny painlessMesh, nebo vytvořit vlastní knihovnu. Vzhledem k velké náročnosti tohoto úkolu, problematické podpoře SSL ze strany ESP8266 a značnému snížení propustnosti sítě, způsobeném režií šifrování, jsem se rozhodl nezapracovat toto vylepšení.

[3] https://play.google.com/store/apps/details?id=com.farproc.wifi.analyzer&hl=cs

- **Rychlost přenosu dat z mateřské sítě** – jelikož platforma ESP8266 nepodporuje současné připojení k více Wi-Fi sítím v režimu klienta, je nezbytné pro získání dat z mateřské sítě provést odpojení od mesh sítě a připojení k mateřské síti, které je po získání dat následováno opačným procesem. Tyto několikanásobné změny připojení jsou časově velmi náročné a silně snižují propustnost.

  Řešením by mohlo být upravení knihovny painlessMesh tak, aby umožňovala připojení k mateřské síti, bez nutnosti deaktivace přístupového bodu mesh sítě na tomto zařízení. Tato úprava by si však vyžádala rozsáhlé úpravy základních částí této knihovny.

- **Necentralizovanost záznamů o přístupech** – každé ze zařízení si udržuje vlastní historii přístupů, která není přístupná bez aktivace rozhraní správce. Pro zjednodušení přístupu k historii průchodů navrhuji jako řešení implementaci mechanizmu, který během aktualizací databáze karet odešle nově vytvořené záznamy na server, kde se budou shromažďovat. Tento přístup by odboural limit historie přístupů a zároveň umožnil centralizovanou správu záznamů o přístupech.

  Implementace tohoto rozšíření by však znamenala rozsáhlé úpravy komunikačního protokolu, algoritmu řízení aktualizací a dalších částí této práce. Z tohoto důvodu jsem se rozhodl nezapracovat toto vylepšení do výsledného řešení.

Kapitola 6

Závěr

Cílem této práce bylo navrhnout a ve formě prototypu realizovat přístupový systém s podporou mesh sítí a možností správy pomocí Wi-Fi rozhraní s použitím mobilního telefonu. Systém měl být založen na čipu ESP8266 a bezkontaktní identifikaci.

V teoretické části této práce jsem se zaměřil zejména na problematiku přístupových systémů, bezkontaktní identifikace, platformu ESP8266 a technologie spojené s přístupovými systémy a realizací této práce. Získané znalosti byly následně použity při realizaci prototypu zařízení a zejména při implementaci software.

Proces vytváření prototypu zařízení spočíval ve výběru vhodného hardware, návrhu zapojení a jeho realizaci. V této části práce se podařilo úspěšně vytvořit plně funkční prototyp zařízení, který byl použit v dalších částech práce jako výchozí hardwarová platforma.

Po realizaci prototypu bylo nezbytné navrhnout a implementovat vhodný software realizující komponenty tohoto systému. Mezi základní implementační požadavky zadání patřila podpora funkcí přístupového systému, možnost správy zařízení pomocí webového rozhraní přístupného přes Wi-Fi přístupový bod vytvořený na tomto zařízení a podpora sítí se smíšenou topologií.

Dokončený přístupový systém jsem poté analyzoval jak po hardwarové, tak po softwarové stránce, aby bylo možné určit jeho vlastnosti. Při této analýze jsem odhalil omezení plynoucí z nedostatečného množství prostředků této platformy, které se i přes veškerou mou snahu projevily na výsledném řešení následujícím způsobem. Komunikace v mesh síti je možná pouze bez šifrování SSL, jelikož by jeho použití značně zvýšilo režii spojenou se zprávami a rapidně snížilo propustnost celé sítě.

Celkově považuji práci za úspěšnou, jelikož se podařilo úspěšně realizovat veškeré dílčí části a vytvořit plně funkční přístupový systém, který splňuje všechny požadavky zadání a přidává další vylepšení ve formě optimalizací.

Literatura

[1] Wireless access control. Kaba Holding AG in Rümlang, Switzerland, [Online; navštíveno 08.04.2017].
URL http://www.kaba.com/access-control/en/solutions/technologies/1408614/wireless-access-control.html

[2] Introduction. Arduino AG in Cham, Switzerland, 2017, [Online; navštíveno 11.04.2017].
URL https://www.arduino.cc/en/Guide/Introduction#

[3] ESP8266 System Description. Espressif Systems (Shanghai) Pte. Ltd., China, 2017.
URL https://espressif.com/sites/default/files/documentation/0b-esp8266_system_description_en.pdf

[4] ESP8266 Technical Reference. Espressif Systems (Shanghai) Pte. Ltd., China, 2017.
URL https://espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf

[5] A.Ashok; V.Thangavelu; N.Harikrishnan: Electronic Toll Collection using Active RFID System. *GESJ: Computer Sciences and Telecommunications*, ročník 2010, č. 1, 2010: s. 5–6, ISSN 1512-1232.
URL http://gesj.internet-academy.org.ge/download.php?id=1487.pdf&t=1

[6] Al-Zewairi, M.; Alqatawna, J.; Al-Kadi, O.: Privacy and security for RFID Access Control Systems. In *Applied Electrical Engineering and Computing Technologies (AEECT), 2011 IEEE Jordan Conference on*, Amman: IEEE Publishing, 2012, ISBN 978-1-4577-1083-4, s. 1–6, doi:10.1109/AEECT.2011.6132520.

[7] Basagni, S.; Conti, M.; Giordano, S.; aj.: *Mobile ad hoc networking*. Hoboken: Wiley-IEEE Press, druhé vydání, 2013, ISBN 9781118087282.

[8] Bowers, D. M.: *Access control and personal identification systems*. Boston: Butterworths, druhé vydání, c1988, ISBN 04-099-0083-4.
URL http://www.worldcat.org/title/access-control-and-personal-identification-systems/oclc/894791114

[9] Cintra, J.: Programming the ESP8266 (NodeMCU) with the Arduino IDE . CodeProject in Toronto Ontario, Kanada, Leden 2016, [Online; navštíveno 10.04.2017].
URL https://www.codeproject.com/Articles/1073160/Programming-the-ESP-NodeMCU-with-the-Arduino-IDE

[10] Finkenzeller, K.: *Fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication*. Hoboken, NJ: Wiley, třetí vydání, c2010, ISBN 978-0-470-69506-7.
URL http://aries.ektf.hu/~dream/e107/e107_files/downloads/rfidhand.pdf

[11] Fried, L.: MiFare Cards & Tags. Adafruit Industries in New York City, Prosinec 2012, [Online; navštíveno 9.04.2017].
URL https://learn.adafruit.com/adafruit-pn532-rfid-nfc/mifare

[12] Hynčica, O.: Bezdrátové sítě typu mesh. *Automa: časopis pro automatizační techniku*, ročník 2005, č. 12, 2005.
URL http://automa.cz/cz/casopis-clanky/bezdratove-site-typu-mesh-2005_12_30826_1141/

[13] IEEE-SA: LOCAL AND METROPOLITAN AREA NETWORK STANDARD. IEEE 802. IEEE-SA, 2012, [Online; navštíveno 9.04.2017].
URL "http://standards.ieee.org/getieee802/download/802.11-2012.pdf

[14] Karel, B.: Systémy elektronického zabezpečení pro integrovanou výuku VUT a VŠB-TUO. In *Společné aktivity VUT a VŠB-TUO při vytváření obsahu a náplně odborných akreditovaných kurzů ICT*, Brno: Vysoké učení technické v Brně, první vydání, 2014, ISBN 978-80-214-5060-8, s. 47–48.

[15] Kharlampidi, V.: Framework7. Rostov na Donu, Russia, [Online; navštíveno 15.04.2017].
URL http://framework7.io/

[16] Knopse, H.; Pohl, H.: RFID Security. 2015, [Online; navštíveno 10.04.2017].
URL http://www.softscheck.com/publications/Pohl_Knospe_RFID_Security_050126.pdf

[17] Mitrokotsa, A.; Rieback, M.; Tanenbaum, A.: Classifying RFID attacks and defenses. *Information Systems Frontiers*, ročník 12, č. 5, 2010: s. 491–505, ISSN 13873326, doi:10.1007/s10796-009-9210-z.
URL https://search.proquest.com.ezproxy.lib.vutbr.cz/docview/807576115?accountid=17115

[18] Mushtaq, N. U.: Zigbee in Home Automation. CCTV Institute, Listopad 2016, [Online; navštíveno 9.04.2017].
URL http://cctvinstitute.co.uk/zigbee/

[19] Novotný, V.: *Pevné a bezdrátové síťové technologie pro integrovanou výuku VUT a VŠB-TUO*. Brno: Vysoké učení technické v Brně, první vydání, 2014, ISBN 978-80-214-5120-9.
URL https://vut-vsb.cz/home/get-file?file=464&amp%3Bportal=Portal2

[20] Violino, B.: The History of RFID Technology. Emerald Expositions, LLC. in San Juan Capistrano, California, United States, 2005.
URL http://www.rfidjournal.com/article/articleview/1338/1/129/

Přílohy

Příloha A

Obsah CD

Soubory na tomto disku jsou rozděleny do adresářů podle svého typu, účelu a dané části práce.

- **Adresář MCU** – adresář MCU obsahuje všechny nezbytné zdrojové kódy a knihovny pro sestavení řídícího programu terminálu. Tento adresář rovněž obsahuje podadresář data, ve kterém se nachází komprimované soubory webové aplikace a konfigurační soubory, ze kterých má být vytvořen SPIFFS.
- **Adresář server** – adresář server obsahuje aktualizační server a nástroj pro úpravu aktualizačních souborů a databáze držitelů karet. V této složce jsou obsaženy i aktualizační soubory pro jednotlivé terminály.
- **Adresář tex** – adresář tex obsahuje zdrojové kódy technické zprávy a všechny grafické prvky, které v ní byly použity.
- **Adresář www** – adresář www obsahuje nekomprimované zdrojové kódy webové aplikace a soubory nezbytné pro její provoz.
- **xtruhl05.pdf** – soubor pdf obsahující technickou zprávu v podobě odevzdané do informačního systému FIT.
- **xtruhl05_print.pdf** – soubor pdf obsahující technickou zprávu v tiskové podobě.
- **README.txt** – soubor popisující strukturu adresářů na disku a jejich obsah.