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

INTELLIGENT ACCESS CONTROL SYSTEM

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **VIKTOR STEINGART**
AUTHOR

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

BRNO 2017

Zadání bakalářské práce/19831/2016/xstein08

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačových systémů Akademický rok 2016/2017

Zadání bakalářské práce

Řešitel: **Steingart Viktor**
Obor: Informační technologie
Téma: **Inteligentní přístupový systém**
Intelligent Access Control System
Kategorie: Vestavěné systémy

Pokyny:

1. Seznamte se problematikou identifikace osob pomocí RFID, zaměřte se na dostupné technologie umožnující číst RFID čipy. Seznamte se s platformou ESP8266, jejími možnostmi a dostupným programovým vybavením.
2. Navrhněte autonomní vestavěný systém na bázi ESP8266, který bude umožňovat řízení přístupů do objektu, jejich správu a evidenci. Předpokládejte přítomnost elektronicky ovládaného zámku a zdroje střídavého napětí 12/24V.
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.

Literatura:

- Dle pokynů vedoucího.

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

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačových systémů a sítí
612 66 Brno, Božetěchova 2

L.S.

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

Abstrakt

Práce se zabývá tvorbou autonomního vestavěného systému sloužícího jako přístupový systém postaveném na platformě ESP8266. Systém umožňuje přístup správce do administrátorského webového rozhraní pomocí mobilního telefonu.

Abstract

This thesis deals with the creation of an autonomous embedded system serving as an access control system built on the platform ESP8266. The system allows administrator access to the admin web interface via a mobile phone.

Klíčová slova

ESP8266, přístupový systém, RFID, WiFi

Keywords

ESP8266, access control system, RFID, WiFi

Citace

STEINGART, Viktor. *Inteligentní přístupový systém.* Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Zdeněk Vašíček, Ph.D.

Inteligentní přístupový systém

Prohlášení

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

.....
Viktor Steingart
16. května 2017

Poděkování

Chtěl bych poděkovat své rodině, která mne při tvorbě této bakalářské práce podporovala. Dále děkuji panu Ing. Zdeňku Vašíčkovi, Ph. D. za odborné vedení a cenné rady, které mi pomohly tuto práci zpracovat.

Obsah

Kapitola 1

Úvod

V současnosti je na českém trhu celá řada přístupových systémů od desítek výrobců. Jedním z nejznámějších je například přístupový systém od české firmy Jablotron. Většina systémů je postavena na RFID technologii, ale lze nalézt přístupové systémy, které jsou vybaveny snímačem některého z biometrických údajů a identifikují osoby pomocí něj, díky čemuž poskytují vyšší bezpečnost. Velikou nevýhodou biometrických přístupových systémů je jejich cena, která se pohybuje v desítkách tisíců korun. Přístupové systémy založené na identifikaci pomocí RFID jsou mnohem levnější, ale stále stojí řádově tisíce korun.

Většina komerčních řešení je složena z RFID čtečky, terminálu pomocí kterého se provádí nastavení a relé, pomocí kterého se spíná elektronicky ovládaný zámek. Některá řešení nabízejí možnost konfigurace pomocí počítače, nejčastěji pomocí USB nebo přes síťové rozhraní.

Cílem této bakalářské práce je vytvoření přístupového systému do objektu, který nabídne podobnou funkčnost jako komerčně nabízené řešení za nižší cenu. Toho je dosaženo využitím platformy ESP8266, která umožní obsluhu dalších periferních zařízení a současně připojení k WiFi síti a vytvoření serveru, na kterém je webové konfigurační rozhraní.

Práce je členěna následovně. První kapitola popisuje obecně technologii RFID a dále se zaměřuje na karty MIFARE. Ve druhé kapitole je popsána platforma ESP8266. Třetí kapitola se věnuje síťovým protokolům, které jsou relevantní této práci. V další kapitole jsou rozebrány technologie související s vytvořením uživatelského rozhraní. Dále práce obsahuje návrh řešení s využitím platformy ESP8266 pro řešení přístupového systému. V sedmé kapitole je popsána implementace celého řešení. V poslední kapitole je popsán výsledný systém a jeho vlastnosti.

Kapitola 2

RFID

Radiofrekvenční systém identifikace (RFID - z anglického Radio Frequency Identification) je technologie, umožňující jednoduchou bezkontaktní identifikaci předmětů, osob nebo míst. Technologie využívá transpondéru, skládajícího se z čipu a antény, který komunikuje s čtečkou pomocí rádiových vln.

2.1 Historie

Počátky této technologie lze nalézt již v dobách druhé světové války. V této době již většina zemí využívala radiových vln k detekci objektu. Radar vysílal radiové vlny, které se při nárazu do pevné překážky odrážely zpět. Pomocí odražených vln se dalo určit polohu a rychlost objektu, od kterého se vlny odrazily. Takto se využívalo radaru především pro detekci letounů v okolí letiště. Problémem však bylo, že nešlo rozpoznat, zda se blíží spojenecký letoun, nebo nepřítel.

Němečtí vojáci přišli na skutečnost, že pokud při návratu na letiště nakloní letoun, lze tuto skutečnost pozorovat na radaru.

V této době přichází Harry Stockman ve svém článku [17] s prvním popisem komunikace pomocí odražených radiových vln. V Británii byl vyvinut systém IFF (Identify Friend or Foe), který umožňoval detekovat, zda se jedná o spojenecký letoun. Využívalo se vysílače umístěného na letounu, který když zachytil signál z pozemní radarové stanice, začal vysílat signál zpět, čímž se přihlásil jako přátelský letoun.

Do komerční sféry se technologie RFID začala dostávat v 60. letech 20. století. V této době se začínaly objevovat systémy pro sledování zboží, které využívaly pouze jednoduchý čip spojený s anténou, pomocí něhož šlo detekovat, zda bylo zboží zaplaceno. V následujících dekádách se vývoj postupně posouval, až v roce 1975 ve svém článku [10] Alfred Koelle, Steven Depp a Robert Freyman představili koncept zcela pasivního transpondéru využívajícího modulovaný zpětný rozptyl.

V 90. letech je tato technologie již ve velkém zaváděna. Své využití nachází například při výběru mýtného na dálnicích, řízení přístupu, či různé aplikace v obchodech.

2.2 Možná aplikace technologie

V současné době lze technologii RFID nalézt na nejrůznějších místech, což je možné především díky veliké variabilitě tvarů a velikostí. Transpondér lze umístit na jakýkoliv objekt, který chceme sledovat. Obrovskou výhodou například proti čárovému kódu spočívá v mož-

Obrázek 2.1: Fotografie RFID přístupového systému

nosti přečíst hodnotu z čipu bez nutnosti přímé viditelnosti z určité vzdálenosti v závislosti na technologii. Mezi oblasti, ve kterých je technologie RFID využívána, mimo jiné patří:

- přístupové systémy
- sledování zboží
- sledování osob či zvířat
- bezkontaktní platby
- doprava
- časomíra při různých sportech

V obchodním prostředí se využívá RFID pro sledování pohybu zboží bez potřeby zásahu člověka. Transpondérem ve formátu samolepících štítků může být označeno veškeré zboží, nebo dílčí části, ze kterých je poté složen finální výrobek. Typicky je tohoto však využíváno v oblastech, kde cena zboží mnohokrát přesahuje cenu RFID štítku a vyplatí se jej použít namísto čárového kódu.

Při přepravě mohou být pomocí technologie RFID označeny například lodní kontejnery, vagóny, či zavazadla na letišti. Díky možnosti jednoduchého strojového přečtení je poté zpracování mnohem efektivnější. V některých státech jsou postupně vybavovány osobní i nákladní automobily RFID transpondéry a je pomocí něj zajištěn výběr poplatků za zpoplatněné oblasti, kontrola zda je auto řádně zaregistrované, nebo pojištěné. [1]

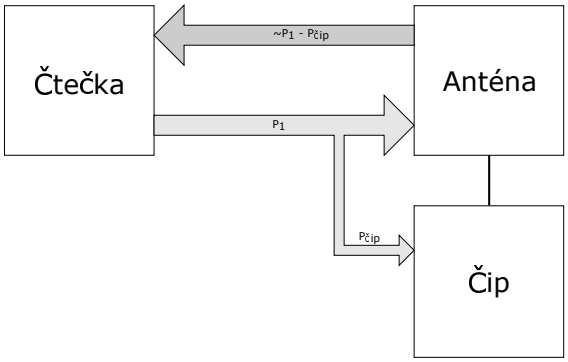
RFID technologie se vyskytuje na mnoha dalších místech. Různé instituce a úřady používají technologii pro identifikaci osob, zavádějí univerzální městské karty, které lze využít v knihovnách, veřejných parkovištích a na mnoha dalších místech. Na obrázku 2.1 je fotografie zachycující použití RFID pro přístup do objektu.

2.3 Transpondér

Transpondér může mít mnoho podob – od karet přes přívěšky na klíče či samolepící štítky až po komplikovanější zařízení. Kromě podoby lze transpondéry rozdělit podle způsobu a množství informací, které do nich lze uložit. Existují transpondéry které dokáží uchovat pouze 1 bit, existují takové které mají 96bitové identifikační číslo (EPC) a poté jsou zde transpondéry, které mají paměť v řádech kilobytů. Důležitým rozdílem v jednotlivých transpondérem je způsob, jakým získávají energii. Podle způsobu je rozdělujeme na aktivní, polo-aktivní a pasivní.

2.3.1 Pasivní transpondéry

Pasivní transpondéry nemají žádný zdroj energie. Energii získávají pomocí cívky sloužící jako anténa z elektromagnetického pole čtečky. Pro zaslání odpovědi se využívá modulace elektromagnetického pole ze čtečky, nebo se do kondenzátoru uloží náboj, který slouží k zaslání odpovědi. Z toho vyplývá že jak pro přenos dat k čipu, tak pro přenos zpět k čtečce se využívá pouze energie získaná ze čtečky. V případě, že je čip příliš daleko od elektromagnetického pole čtečky, nemá potřebnou energii pro vysílání vlastní identifikace. Na obrázku 2.2 vidíme že čtečka vysílá radiový signál o výkonu P_1. Z konfigurace plyne, že odpověď nemůže být vyslána rádiovým signálem s výkonem vyšším než-li $P_1 - P_{ip}$.

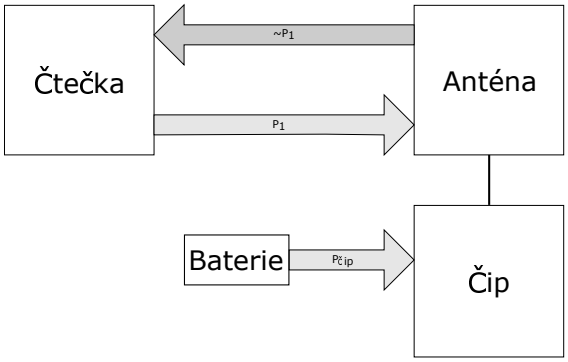


Obrázek 2.2: Pasivní čip využívá pro napájení čipu indukčního pole ze čtečky

2.3.2 Polo-aktivní transpondéry

Polo-aktivní transpondéry obsahují vlastní zdroj energie ve formě baterie nebo solárního článku, který dodává potřebnou energii pro čip. Díky tomu nemusí být elektromagnetické pole tak silné a lze tak dosáhnout větší čtecí vzdálenosti, než v případě pasivních transpondérů.

Přesto polo-aktivní transpondéry nedokáží generovat vysokofrekvenční signál samy, ale používají modulaci radiového signálu ze čtečky podobně jako pasivní čipy. Na obrázku 2.3 vidíme, že čtečka vysílá radiový signál P_1. Protože není třeba napájet čip, který má vlastní zdroj elektrické energie, je možné, aby odpověď byla vyslána radiovým signálem o výkonu P_1.

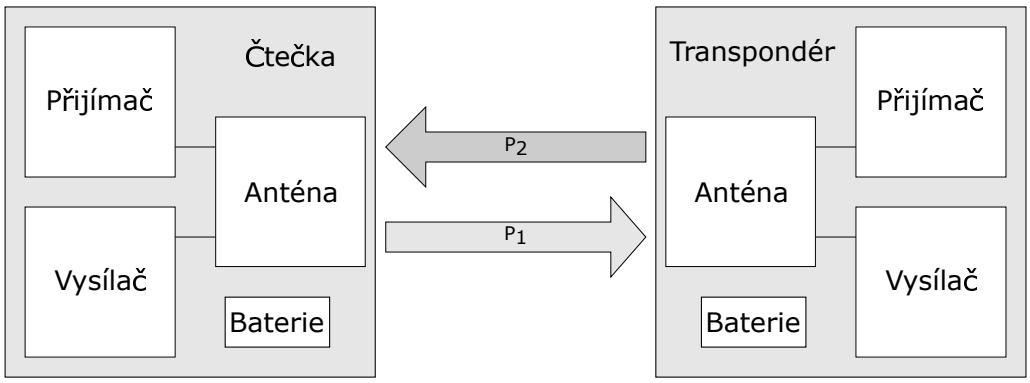


Obrázek 2.3: Polo-aktivní transpondér má baterii, která dodává energii čipu

2.3.3 Aktivní transpondéry

Aktivní transpondéry mají stejně jako polo-aktivní vlastní zdroj energie. Narozdíl od polo-aktivních nepoužívají modulaci radiového signálu ze čtečky, ale vytvářejí vlastní radiový signál, pomocí nějž přenášejí data. Jedná se tedy o radiové zařízení, které má vlastní vysílač a přijímač, a může být zcela nezávislý na čtečce.

Na obrázku 2.4 můžeme vidět, že zatímco čtečka vysílá radiový signál o výkonu P_1, transpondér zcela nezávisle na tomto signálu zasílá odpověď radiovým signálem o výkonu P_2, který je závislý na zdroji a vysílači v transpondéru.



Obrázek 2.4: Aktivní transpondér obsahuje přijímač a vysílač, jímž zasílá odpověď

2.4 Čtečky

Podle toho zda čtečka obsahuje i vysíláč, lze je rozdělit na aktivní a pasivní. Pasivní čtečka nevysílá sama žádný signál, pouze přijímá radiový signál z aktivních transpondérů. Dosah záleží především na výkonu vysílače v aktivních transpondérech, může dosahovat až 600m.

Aktivní čtečka obsahuje přijímač i vysílač radiového signálu. Pro komunikaci s transpondéry vytváří elektromagnetické pole, jenž čip využije pro zaslání odpovědi.. Paralelně na anténu je připojen kondenzátor, jehož kapacita je zvolena tak, že spolu s indukčností antény tvoří rezonanční obvod s rezonanční frekvencí odpovídající přenosové frekvenci čtečky. Cívku ve čtečce spolu s cívkou v transpondéru vytvářejí obvod pracující na stejném principu jako transformátor. Účinnost přenosu energie je úměrná pracovní frekvenci, počtu vinutí,

úhlu cívek vůči sobě a jejich vzdálenosti. Při zvyšující se frekvenci se snižuje požadavek na indukčnost cívky a lze tedy použít méně vinutí.

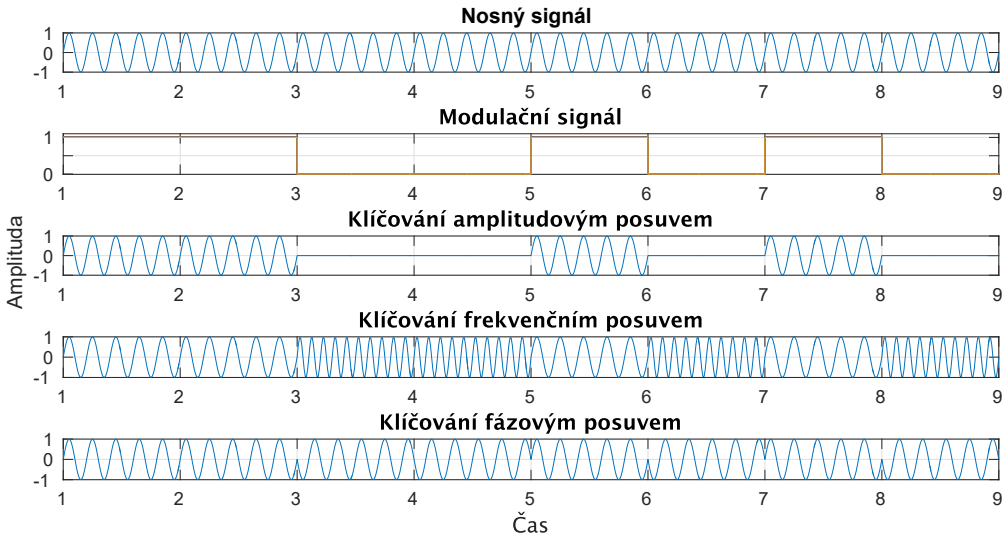
2.5 Princip komunikace

Přenos dat probíhá pomocí radiových vln na určité frekvenci, která se pohybuje řádově od stovek kilohertzů do gigahertzů. Systémy využívající indukční napájení zpravidla pracují v rozmezí 100 kHz až 30 Mhz, nejrozšířenější frekvence jsou 125 kHz a 13,56 MHz.

Při přenosu data procházejí přes několik systémů. Prvním z nich je kódovací systém, jenž má na vstupu datový signál, která se mají přenést a na výstupu má signál, který se bude lépe přenášet pomocí zvolené modulace. Mezi nejpoužívanější způsoby kódování patří NRZ (Non-return-to-zero), Manchester, Unipolární RZ kódování, Millerovo kódovaní a diferenciální kódování.

Další ze sytémů je modulátor, v němž se mění charakter nosného signálu pomocí modulačního signálu. Jedním ze způsobů modulace je klíčování amplitudovým posuvem (anglicky Amplitude-shift keying). U této modulace modulační signál (data k odeslání) ovlivňuje amplitudu nosného signálu tak, že přepíná mezi amplitudami u_0 a u_1(klíčovaná). Poměr těchto dvou amplitud je znám jako činitel m. Klíčování amplitudovým posuvem lze matematicky popsat jako násobení modulačního signálu nosným signálem.

Dále se využívá klíčování frekvenčním posuvem



Obrázek 2.5: Ukázka klíčování datového signálu na nosnou frekvenci různými modulacemi

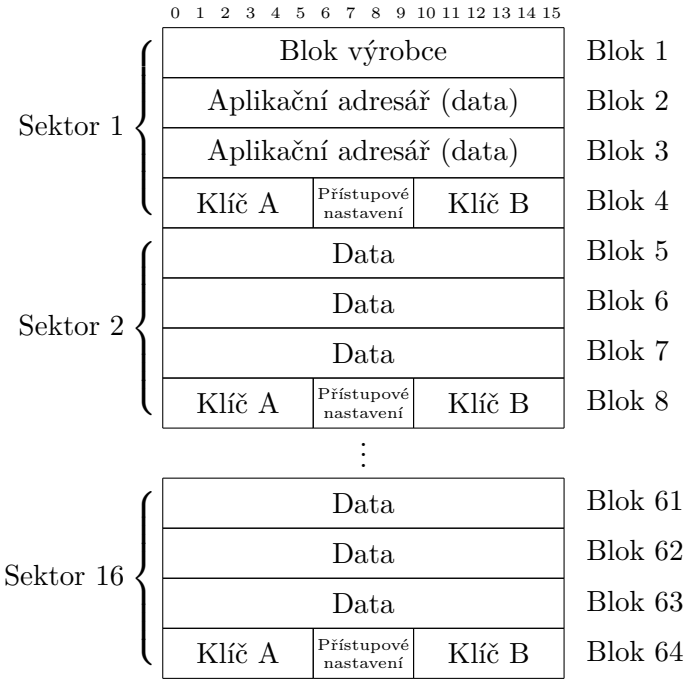
2.6 Karta MIFARE

Karty MIFARE jsou implementací identifikačních karet dle standardu ISO/IEC 14443 Type A, které využívají technologii RFID na frekvenci 13.56 MHz. Karty MIFARE nabízejí především větší paměť a také podporu zabezpečení této paměti. Mezi nejvíce rozšířený typ patří MIFARE Classic, který nabízí 1 nebo 4kB vlastní paměti spolu s proprietárním bezpečnostním protokolem NXP pro autentizaci a šifrování. Protože však již není toto zabezpečení pro

kritické aplikace dostatečné[5], přechází mnohé instituce na novější a bezpečnější typy MIFARE karet (například na kartu MIFARE DESFire EV2).

2.6.1 Struktura paměti

Struktura paměti bude popsána na typu MIFARE Classic 1k. Jak můžeme vidět na obrázku 2.6, paměť je rozdělena do 16 sektorů, každý je chráněn proti neoprávněnému přístupu dvěma různými klíči. Pro každý sektor mohou být přiřazena různá práva pro každý ze dvou klíčů. Díky tomu je možné, aby jednu univerzální kartu používalo až 16 aplikací, kde každá má svůj sektor chráněný tajným klíčem.



Obrázek 2.6: Struktura paměti v kartě MIFARE Classic 1k

	7	6	5	4	3	2	1	0
Byte 6	$\overline{C2_3}$	$\overline{C2_2}$	$\overline{C2_1}$	$\overline{C2_0}$	$\overline{C1_3}$	$\overline{C1_2}$	$\overline{C1_1}$	$\overline{C1_0}$
Byte 7	$C1_3$	$C1_2$	$C1_1$	$C1_0$	$\overline{C3_3}$	$\overline{C3_2}$	$\overline{C3_1}$	$\overline{C3_0}$
Byte 8	$C3_3$	$C3_2$	$C3_1$	$C3_0$	$C2_3$	$C2_2$	$C2_1$	$C2_0$
Byte 9	Data uživatele							

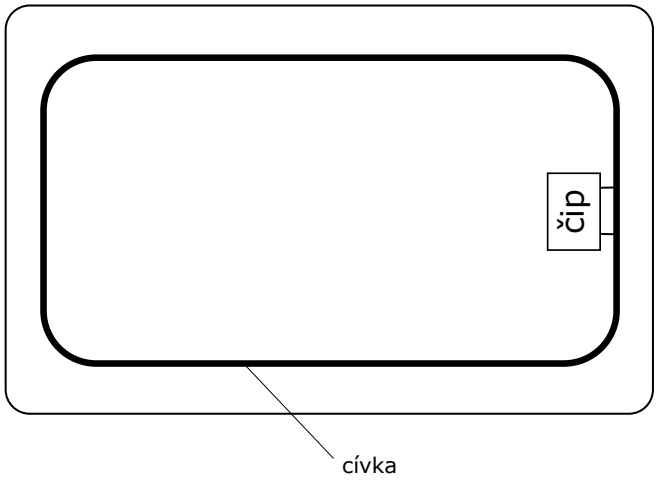
Obrázek 2.7: Struktura přístupového nastavení

Každý sektor obsahuje 4 bloky po 16 bytech. Poslední blok v sektoru je nazýván zavaděč (anglicky trailer) a obsahuje dva 6-bytové přístupové klíče a 4-bytovou konfiguraci přístupových pravidel. Přístupová pravidla se zapisují pomocí 3 bitů pro každý ze 4 bloků. Data jsou zapsána nejdříve invertovaná a poté neinvertovaná, čehož se využívá pro kontrolu zda jsou přístupová pravidla konzistentní. Struktura je na obrázku 2.7 . Pro datové bloky i pro zavaděč lze nalézt v produktovém listu [14] kombinaci 3 bitů odpovídajících přístupovým podmínkám. Například bity `[0 1 1]` pro zavaděč značí, že po autentizaci klíčem A i klíčem

B můžeme z tohoto bloku přečíst pouze bity s přístupovým nastavením. Klíč B může navíc ještě zapsat klíč A i B a také zapsat do bitů s konfigurací. Díky tomuto lze nastavit jednomu klíči například oprávnění číst i psát do daného bloku, ale zamezit manipulaci s klíčem. Na kartě může tedy být v každém sektoru jiná zcela nezávislá aplikace, kde každá bude mít vlastní přístupová práva. První blok v nultém sektoru je vyhrazen pro data výrobce. Zde je uložen také unikátní identifikátor čipu. Do tohoto sektoru nelze zapisovat a díky tomu nelze unikátní identifikátor změnit. V dalších dvou blocích je MAD (MIFARE aplication directory), v němž jsou informace o jednotlivých aplikacích na kartě a ukazatel na jejich pozici. Pro ostatní sektory platí, že vždy 3 bloky jsou volné pro ukládání informací.

2.6.2 Fyzické vlastnosti

Karty MIFARE se skládají z vlastního čipu a z vodiče tvořícího cívku (obrázek 2.8). Cívka plní funkci antény pro přenos dat a také se pomocí ní získává energie. Vyžívá se silného vysokofrekvenčního elektromagnetického pole, které je kolem cívky. Používaná vlnová délka je v případě používané frekvence 13,56 MHz přibližně 22,1 metru, což je mnohokrát více, než je vzdálenost karty od čtečky. Díky tomu lze pole považovat za jednoduché magnetické střídavé pole. Část tohoto pole prochází cívkou a generuje v ní díky indukčnosti elektrické napětí. To je využíváno mikročipem v kartě pro napájení.



Obrázek 2.8: umístění čipu a antény na kartě MIFARE Classic

Kapitola 3

Platforma ESP8266

Platformou ESP8266 jsou nazývaný všechny systémy postavené na čipu ESP8266EX, který vyrábí čínský výrobce Esspressif Systems. Čip je postaven na 32bitovém procesoru 32-bitové architektury běžícím na 80 MHz (může být přetaktován na 160MHz), má 64 KB paměti pro instrukce, 96 KB paměti RAM, zpravidla připojenou externí flash paměť, 16 GPIO (General Purpose Input/Output) a 10b analogově digitální převodník. Další periferní zařízení lze připojit pomocí sběrnice SPI, I^2C, I^2S a UART.

V roce 2014 byl představen poprvé modul a okamžitě zaujal mnoho výrobců. Mezi jeho přednosti patřilo, že za velmi dobrou cenu (kolem 5 dolarů) umožnil mnoha mikroprocesorům, ke kterým se připojoval pomocí sběrnic SPI nebo UART, realizovat TCP/IP spojení přes WiFi síť. [2] Z počátku bylo velkým problémem, že neexistovala dokumentace čipu a příkazů, které modul akceptoval. Později byl k dispozici katalogový list produktu v čínštině, seznam AT příkazů a také sada nástrojů pro vývoj, které však byly spíše pro samotný čip, než pro jednoduché připojení modulu k mikrokontroléru. Modul se tou dobou již dal zakoupit a během krátké doby vnikla komunita, jež přeložila všechny produktové listy do angličtiny a také prozkoumala, jakým způsobem s ním lze pracovat. Za největší úspěch lze považovat fakt, že se podařilo vytvořit překladač umožňující používat samostatný čip jako řídící mikrokontrolér celého projektu. [3] V krátkém čase společnost Al-Thinker začíná

Obrázek 3.1: Fotografie modulu NodeMCU obsahující ESP-12E

vyrábět řadu modulů „ESP-XX“, kde se jednotlivé moduly liší především počtem vývodů a jejich rozmístěním, možností připojení na určitý typ vývojové platformy, přítomností svítivé diody, typem antény (používá se anténa vytištěná na desce plošného spoje, keramická

anténa a nebo konektor pro připojení externí antény), různou velikostí vnitřní paměti a různými rozměry.

Těchto modulů využívají mnohé další moduly, které vedle samotného čipu ESP8266 přidávají převodník z USB na UART (nejčastěji CP2102 nebo CH340G) , stabilizátor napětí na 3,3 V společně s USB konektorem, který slouží pro komunikaci s počítačem a také může sloužit jako zdroj elektrické energie. Dále přidávají úložiště SPIFFS (SPI Flash file system) ve velikosti od 0,5 MB do 4 MB připojené přes SPI rozhraní. Díky vybavenosti se tyto rozšířené moduly, nejčastěji postavené na ESP-12E, setkávají s větší oblíbeností než modul obsahující pouze čip ESP8266 a několik vývodů. Na obázku 3.1 je modul NodeMCU, který je založen na ESP-12E

V následujících podkapitolách budou popsány technologie, kterými je modul ESP8266 vybaven. Bude popsána technologie WiFi sloužící k bezdrátovému připojení k síti, sběrnice SPI, I^2C a I^2S pomocí kterých lze připojit další periferní zařízení. Poslední podkapitola se věnuje možnostem, jakými lze vyvíjet programy běžící na této platformě.

3.1 WiFi

Wifi je technologie založená na standardu IEEE 802.11 sloužící pro bezdrátové připojení k síti. Standard specifikuje implementaci bezdrátové lokální sítě (wireless local area network, WLAN) v pásmu 2,4 GHz. Pro přenos lze využít 13 kanálů.

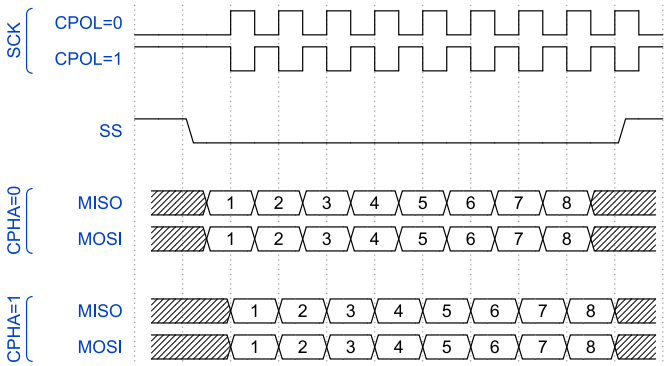
Existuje několik doplňků označovaných **802.11x**, kde x je nahrazeno písmenem daného dodatku. ESP8266 podporuje dodatky **b**, **g** a **n**. Standard **IEEE 802.11b** je rozšířením původního standardu, který zvyšuje přenosovou rychlost na pásmu 2,4 GHz ze dvou na jedenáct megabitů za sekundu. Tento standard je při zachování zpětné kampatibility dále rozšířen ve standardu **IEEE 802.11g** na rychlost 54 Mb/s. Dalšího zrychlení až na 600 Mb/s se dosahuje ve standardu **IEEE 802.11n** využitím technologie **MIME** (Multiple Input Multiple Output), která využívá více vysílacích a přijímacích antén a také umožňuje navíc používání na frekvence 5 GHz.

Každá síť má vlastní identifikátor (Service Set Identifier, SSID), který je přístupovým bodem periodicky vysílá kvůli možnosti detekce sítě klientem. [4, 23]

Ve vývojovém prostředí Arduino IDE slouží pro práci s WiFi na platformě ESP8266 knihovna ESP8266WiFi. Ta umožňuje provozovat WiFi ve dvou režimech: v režimu stanice, kdy se modul připojí k již existující síti, nebo v režimu přístupového bodu, ve kterém vytváří vlastní síť. WiFi sítě mohou být šifrované pomocí WEP nebo WPA. Dále je možné, aby modul fungoval jako server nebo klient. Je možné zasílat i přijímat UDP pakety. IP adresa může být nastavena staticky i dynamicky pomocí služby DHCP a také lze využívat službu DNS.

3.2 SPI

Sériové periferní rozhraní (anglicky Serial Peripheral Interface, SPI) je synchronní sériová sběrnice, která slouží pro komunikaci několika zařízení. Zařízení komunikují v režimu plného duplexu po sdílené sběrnici. Na sběrnici je jedno řídící (master) a jedno, nebo více podřízených (slave). Řídící zařízení obsahuje generátor hodinového signálu, jenž je rozveden do všech ostatních zařízení vodičem označeným SCK. Pro výběr zařízení se kterým bude řídící uzel komunikovat, slouží vodič SS (Select Slave). Poslední dvojicí vodičů jsou MISO (Master In, Slave Out) a MOSI (Master Out, Slave In). Díky oddělení těchto dvou vodičů



Obrázek 3.2: Komunikace pomocí SPI

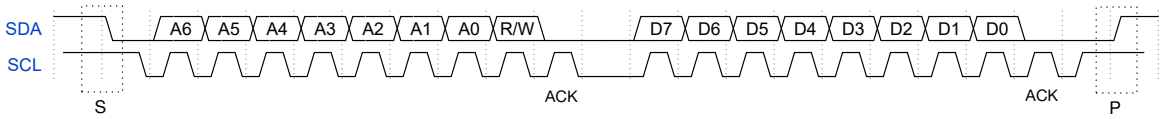
jsou potřeba pouze jednosměrné vstupy a výstupy a díky využití TTL logiky je i elektrické zapojení jednoduché.

Pro komunikaci obsahují všechny komunikující uzly dva posuvné registry . Prvním registrem je přijímací datový registr, který není programově přístupný a slouží pouze ke čtení. Vysílací registr je naopak programově přístupný a slouží pro zápis. Oba registry jsou přístupné na stejné adrese nazývané datový registr (SPDR). Kromě těchto dvou registrů jsou v zařízení další dva registry – řídící SPCR a řídící/stavový SPSCR. SPCR slouží pro uchování konfigurace SPI, za zmínku stojí konfigurační bity 2 (Clock Phase, CPHA) a 3 (Clock Polarity, CPOL) sloužící na nastavení fáze a polarity hodinového signálu. SPSCR se kromě konfigurace používá i na uchování stavu. Při zahájení přenosu začne řídící uzel generovat hodinový signál, pomocí adresovacího vodiče SS vybere zařízení pro komunikaci nastavením logické 0. Pokud je v registru SPSCR nastaven 3. bit nazývaný SPTE (SPI Transmitter Empty) na hodnotu logické 1, zapíší se data která se mají přenést do datového registru. Pokud SPTE má hodnotu logické 0, musí se počkat na uvolnění registru. Po zapsání do datového registru se data okamžitě přesouvají do vysílacího registru a začíná se vysílat. Současně se začínají přijímat data od protějšku. Jakmile je odeslán a současně i přijat celý byte, nastaví se v SPSCR 7. bit nazvaný SPRF (SPI Receiver Full) na logickou 1. V tento okamžik je možné přečíst přijatá data. Nejprve je vynulován bit SPRF a následně přečten byte ze SPDR. Po skončení komunikace se zruší nastavení vodiče SS a pokud nebude pokračovat komunikace s jiným zařízením, ukončí se generování hodinového signálu. [18, 21]

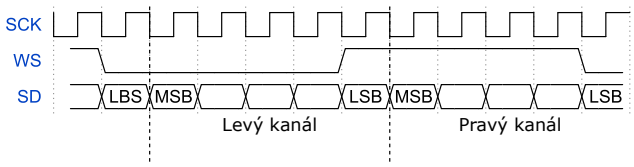
Modul ESP8266 obsahuje dvě sběrnice SPI, které bývají označeny jako SPI a HSPI. Pokud modul obsahuje flash paměť, zpravidla bývá připojena na první sběrnici. Druhá sběrnice bývá nevyužita a lze ji tedy použít pro připojení komponent. K dispozici jsou vysokoúrovňové funkce pro nastavení SPI komunikace, přijmutí a odeslání dat. Dále je možné využít nízkoúrovňové funkce pro čtení MISO a nastavení MOSI.

3.3 I^2C

I^2C (anglicky Inter-Integrated Circuit) je synchronní sériová sběrnice navržená firmou Philips, využívající dva vodiče: signál hodinového signálu SCL (serial clock) používaný pro synchronizaci a jeden datový vodič SDA (serial data) se 7-bitovou adresací. Vodiče SCL i SDA potřebují ještě společnou zem a také napájecí napětí, které slouží k přepnutí datového vodiče pomocí pull-up rezistorů na hodnotu logické 1.



Obrázek 3.3: Komunikace pomocí I^2C



Obrázek 3.4: Komunikace pomocí I^2S

Zařízení může pracovat ve 4 modech:

- řídící vysílač (master transmiter)
- řídící přijímač (master reciever)
- podřízený vysílač (slave transmiter)
- podřízený přijímač (slave reciever)

Každé zařízení připojené na sběrnici má vlastní softwarovou adresu. Na obrázku 3.3 vidíme komunikaci na sběrnici. Při zahájení komunikace řídící uzel zasílá start bit (S). Následně zasílá adresou uzlu se kterým se bude komunikovat (A0 - A6) a bit (R/W) určující zda bude z uzlu číst nebo zapisovat. Následuje přenos dat (D0-D7). Každý přenesený byte je potvrzován zasláním potvrzovacího bitu. Vysílající uzel tento byt vysílá jako logickou 1, přijímací uzel jej přepne na 0. Po ukončení přenosu je zaslán stop bit (P). V klidovém stavu jsou oba dva datové vodiče ve stavu logické 1. [15, 18, 19, 20]

Platforma ESP8266 nemá hardwarovou podporu I^2C, je nutné využít programové řešení v podobě knihovny **Wire**, nabízející vytvoření sběrnice na jakékoliv dvojici vývodů.

3.4 I^2S

I^2S (Inter-IC Sound) je synchronní sériová sběrnice sloužící k propojení digitálních zařízení zpracovávající zvuk. Pro přenos se využívá pulzně kódová modulace zvuku. Data se přenáší pomocí jednoho datového vodiče (SD) a dvou hodinových (SCK, WS).

Na datovém vodiči se přenáší nejdříve nejvíce významný bit. Díky tomu vysílač ani přijímač nepotřebuje znát délku přenášeného slova. V případě že délka systémového slova je delší než délka slova vysílače, jsou méně významné bity odseknuty v okamžiku, kdy dojde ke změně hodinového signálu WS (word select) . Pokud je naopak délka systémového slova na přijímači větší, než je velikost přenášeného slova, jsou na nejméně významné pozice doplněny nuly. Pomocí signálu WS se přepíná mezi pravým (WS = 1) a levým kanálem (WS = 0). [9, 22]

Platforma ESP8266 je vybavena jedním rozhraním I^2S, které je na vývodech GPIO2 (WS), GPIO3 (DATA) a GPIO15 (SCK). Pro obsluhu slouží knihovna **i2s**.

3.5 Vývojové prostředky

Možnosti vývoje na platformě ESP8266 započaly v říjnu 2014, kdy Espressif uvolnil SDK umožňující programování čipu. Později byly uvolněny další sady vývojových nástrojů. Mezi nejčastěji používané patří NodeMCU založené na jazyce Lua, MicroPython a také integrace do vývojového prostředí Arduino IDE umožňující programování ESP8266 stejně jako jiné vývojové desky platformy Arduino.

v prostředí Arduino IDE se využívá pro programování C++ s knihovnou Wiring. Zvláštností je přítomnost dvou základních funkcí `setup` a `loop`. Kód ve funkci `setup` se provede pouze jednou na začátku běhu programu. Používá se nejčastěji pro nastavení defaultních hodnot a inicializaci všech využívaných součástí. Funkce `loop` je poté cyklicky volána během běhu programu a obsahuje výkonný kód. Tyto dvě funkce musejí být v programu vždy, i když jsou prázdné. Je možné vytvářet další funkce, které budou volány z těchto dvou funkcí.

Kapitola 4

Síťová komunikace

V této kapitole budou popsány síťové protokoly pro přenos dat, které jsou relevantní pro tuto práci, která předpokládá vytvoření inteligentního přístupového systému, který bude vyžadovat komunikaci pomocí protokolu HTTP, sloužícího pro přenos hypertextových dokumentů, a protokolu NTP sloužícího pro synchronizaci času.

4.1 Hypertext Transfer Protokol

Jedná se o bezstavový síťový aplikační protokol sloužící pro přenos hypertextových dokumentů. Hypertextové dokumenty jsou strukturované texty obsahující odkazy na další uzly obsahující text. HTTP verze 1.0 je definován normou RFC 1945, novější verze 1.1 je difinována v RFC 2616 a následně v RFC 7230.

V HTTP se využívá pro určení umístění zdroje na internetu jednotný identifikátor prostředků (Uniform Resource Identifier, URI) v podobě jednotného lokátoru prostředků (Uniform Resource Locator, URL) a jednotného názvu zdroje (Uniform Resource Name, URN). Zprávy se přenášejí ve formátu, který je podobný formátu využívaném internetovou poštou, MIME (Multipurpose Internet Mail Extension) umožňující přenášet i jiné typy souborů než jsou hypertextové soubory.

HTTP funguje na principu dotaz/odpověď. Níže vidíme http požadavek, který zašle klient na server. Zpráva obsahuje na prvním řádku použitou metodu, požadovaný dokument a verzi protokolu. Následuje adresa, na kterou je požadavek směrován. Dále následuje část, ve které prohlížeč zašle svoji totožnost a mnohé další informace jako očekávané kódování nebo typ dat. Také zasílá cookies, pomocí nichž lze uchovat data pro dané zařízení.

```
GET /index.html HTTP/1.1
Host: 10.0.0.44
Connection: keep-alive
Cache-Control: max-age=0
Authorization: Basic am1lbm86aGVzbG8=
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/58.0.3029.96 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate, sdch
Accept-Language: cs,en-US;q=0.8,en;q=0.6
Cookie: mojeCookie=hodnota
```

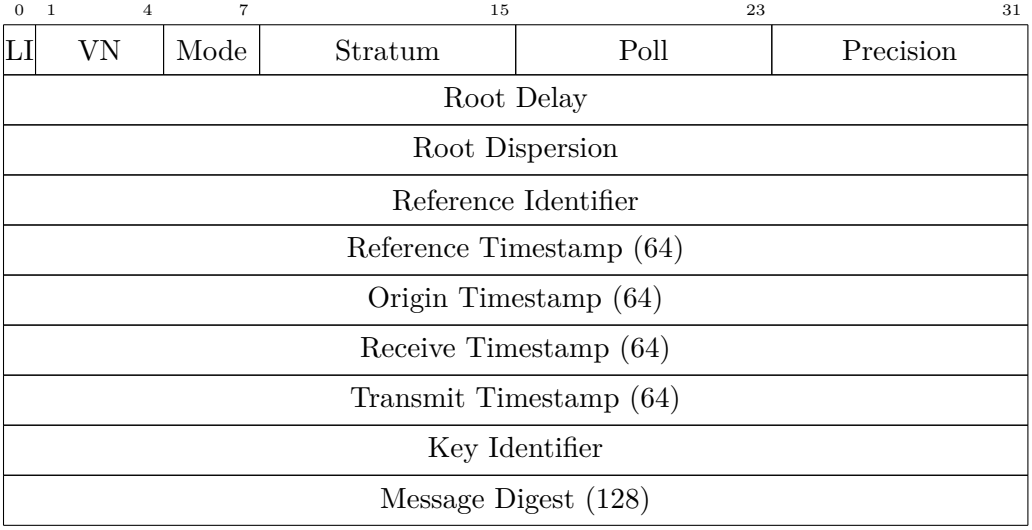
Níže je vidět odpověď serveru. Na prvním řádku obsahuje opět verzi protokolu a stavový kód. Následují hlavičky informující o přenosu a zasílaných datech. Následuje prázdný řádek a poté přenášená data. [23]

```
HTTP/1.1 200 OK
Content-Type: text/html
Connection: close
Access-Control-Allow-Origin: *
Transfer-Encoding: chunked
```

4.2 NTP protokol

Protokol síťového času (Network Time Protocol, NTP) slouží k synchronizaci systémového času pomocí internetu. Poskytuje mechanizmy umožňující synchronizaci napříč různými systémy s nedeterministickou dobou odezvy. Je toho docíleno sestavou serverů v hierarchickém systému využívající pojem „strata hodin“. Servery se strata 1 mají čas odvozen od přesných externích hodin nejčastěji řízenými radiovým signálem nebo GPS. Strata 2 odvozuje svůj čas od jednoho nebo více serverů úrovně strata 1 . Hierarchický sytém může mít hloubku až 15 úrovní. Pokud má zařízení strata 16, je považován za nesynchronizovaný.

Časová razítka jsou 64-bitová, kde 32 bitů představuje sekundy. NTP sekundy počítá od 1. ledna 1900. Rozsah 32 bitů umožňuje unikátně rozlišovat přes 136 let než dojde k přetečení hodnot a k jejich opakování. Druhá část časového razítka představuje desetinou část sekundy, umožňuje tedy rozlišení 233 pikosekund.



Obrázek 4.1: Struktura NTP paketu

NTP paket (obrázek 4.1) obsahuje informaci o přestupné sekundě(LI), číslo verze (VN), mód, úroveň strata, maximální interval mezi úspěšnými zprávami a přesnost. Dále paket obsahuje zpáteční zpoždění, relativní chybu vůči primárnímu zdroji, identifikátor, časové razítko z okamžiku kdy paket odcházel od klienta, časové razítko okamžiku, kdy dotaz přišel na server, a časové razítko okamžiku, kdy paket ze serveru odcházel. Volitelně může být součástí ještě identifikátor klíče a podpis zprávy. [23]

Kapitola 5

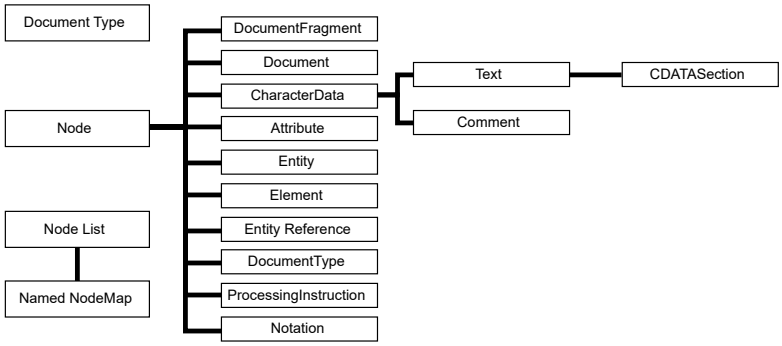
Technologie pro vývoj webových stránek

V této kapitola budou popsány technologie využitelné při tvorbě webového rozhraní.

5.1 Document object model

Objektový model dokumentu (anglicky Document Object Model, DOM) umožňuje dynamicky a současně platformě nezávisle pracovat s dokumenty. Jedná se specifikaci rozhraní implementovaného v objektovém jazyce. Model spravuje konsorcicium W3C a vydává jednotlivé verze. Poslední verze **DOM Level 4** se pokouší sloučit předchozí standardy do jednoho, zjednodušit je a současně je přiblížit dalším existujícím standardům, především JavaScriptu a HTML5.

DOM lze chápat jako uspořádaný strom popisující dokument, jehož každý uzel může obsahovat další podstromy. Kořenový uzel je typu **Document**. Všechny třídy objektů, které může dokument obsahovat jsou odvozeny ze třídy **Node**. Na obrázku 5.1 lze vidět znázorněnu dědičnost jednotlivých tříd. Pro třídu **Document** platí, že může obsahovat pouze jeden synovský uzel typu **DocumentType** a jeden typu **Element**, který reprezentuje jakýkoliv HTML nebo XML element. Každý **Element** má jméno a může mít atributy. Také může mít synovské uzly typu **Element**, **Text** a **Comment**. Text a Comment jsou vždy listové uzly, protože nemohou obsahovat další elementy.



Obrázek 5.1: Document Object Model

5.2 HTML

Hypertextový značkovací jazyk (anglicky Hypertext Markup Language, HTML) je typem SGML(Standard Generalized Markup Language) dokumentu u kterého mají značky přiřazenu sémantiku hypertextového dokumentu. První verze byly vytvořeny v letech 1991 – 1993. Od roku 1999 se dlouhou dobu používala verze 4.01. V této verzi mají jednotlivé prvky odpovídat sémantice jednotlivých částí dokumentu a vzhled je řešen připojenými styly. Předpokládalo se že je to poslední verze HTML, kterou měl vystřídat jazyk XHTML založený na univerzálním jazyku XML. V roce 2014 byla vydána specifikace HTML 5.0. Ta opravuje mnoho chyb předchozí verze, odstraňuje nepoužívané prvky, zjednodušuje zápis některých značek a přidává podporu novým technologiím.

HTML 5.0 se snaží zvýšit přehlednost kódu a zavadí nové sémantické značky: section, article, main, aside, hgroup, header, footer, nav, figure, figcaption. Důležitá je také podpora multimédii: audio, video, source a track. Ve formulářích přidává nové typy formulářových polí a také nové atributy. Přidává také podporu offline stránek, které se v případě nedostupnosti načítají z lokální cache. Užitečnou funkčností je asociativní pole localStorage umožňující perzistentní ukládání dat. Podobně funguje sessionStorage, ale pouze po dobu trvání sezení.

Ukázka jednoduchého HTML dokumentu:

```
<!DOCTYPE html>
<html>
   <head>
      <title>HTML dokument</title>
   </head>
   <body>
      <h1>Nadpis </h1>
      <p>Odstavec</p>
   </body>
</html>
```

5.3 JavaScript

Jedná se interpretovaný programovací jazyk s objektově orientovanou koncepcí. Jazyk má podobnou syntaxi jako C, C++ nebo Java. Existuje standardizovaná verze **ECMAScript** zastřešená neziskovou organizací **ECMA**.

Nejčastěji se s JavaScriptem setkáváme ve webových prohlížečích, ve kterých je integrován a lze pomocí něj manipulovat s DOM, což umožňuje dosáhnout dynamického chování webu.

JavaScript je dynamicky typovaný jazyk, což znamená že datový typ je asociován s hodnotou a ne s proměnnou. Jak již bylo řečeno, JavaScript je objektově orientovaným jazykem. V JavaScriptu je zavedena jednoduchá dědičnost, kdy všechny objekty jsou potomky **Object**, dokonce i objekt **Function**, představující funkci. Veškeré objekty jsou asociativní pole, kde názvy atributů objektu jsou klíči v asociativním poli.

5.4 CSS

Kaskádové styly (Cascading Style Sheets, CSS) jsou jazyk pro popis stylů jednotlivých HTML elementů. Na počátku vzniku HTML se předpokládalo, že se pomocí HTML popíše dokument a prohlížeč mu dodá vzhled a vykreslí jej. Postupem času však vznikla potřeba měnit vzhled a chování jednotlivých prvků. Konsorcium W3C v roce 1996 navrhlo standard kaskádových šablon stylů. V roce 1998 vznikla druhá verze a spolu s nástupem HTML5 přichází CSS3.

```
h2 {
        color: red;
         font-size: 12pt;
}
h2.nadpis {
        color: blue;
}
```

Na příkladu můžeme vidět základní syntaxi a princip kaskádových stylů. Je zde dvakrát definovana barva nadpisu druhé úrovně. První selektor platí pro všechny nadpisy druhé úrovně, kterým nastaví barvu písma na červenou. Pokud by však měl html element nastavenu třídu na „nadpis“ (`class="nadpis"`), bude odpovídat i druhému selektoru, který je definován později. Zdědí tedy vlastnosti od nadřazených selektorů, ale atribut, který je nastaven pro přesnější selektor bude mít vždy větší prioritu.

Nejnovější specifikace CSS 3, jejíž vývoj byl zahájen v roce 2005, sice ještě stále není dokončena, avšak ve většině prohlížečů jsou nové vlastnosti již podporovány. CSS 3 přidává podporu animací, které se již nemusí vytvářet pomocí JavaScriptu, přidává průhlednost prvků, možnost zaoblených rohů, stíny, různé transformace, přidává podporu dalších barevných modelů a spoustu dalších novinek. Velmi užitečnou vlastností pro vývoj responsivních webů jsou **Media Queries**, umožňující aplikovat různá pravidla v různém kontextu. Lze například měnit vzhled na základě rozlišení nebo poměru stran obrazovky, na základě orientace zařízení, nebo například hustotě pixelů na obrazovce.

5.5 Framework7

Framework7 je framework pro vytváření aplikací pro mobilní telefony. Obsahuje knihovny a řešení pro efektivní tvorbu webové stránky, která poté může být zabalena do balíčku aplikace.Framework7 je celý postaven na HTML, CSS a JavaScriptu. Mezi hlavní přednosti frameworku patří že se jedná o plně JavaSriptové řešení běžící u klienta, včetně možnosti využívat šablony ze kterých se vygeneruje na základě získaných dat obsah dokumentu.

Webová aplikace vytvořená pomocí Framework7 se skládá z pohledů (Views). Jsou to oddělené vizuální části aplikace s možností vlastního nastavení, navigace a historie. V každém pohledu je poté další obsah, který vytváří výsledný dokument. Jedná se o další stránky (Pages), které lze chápat jako jednotlivé stránky u běžné webové aplikace. Základní uspořádání stránky je v příloze B. Každá stránka má události, díky kterým lze provést specifický kód v přesně daný okamžik. Události stránky jsou: beforeinit, init, reinit, beforeanimation, afteranimation, beforeremove, back, afterback.

Framework7 nabízí ještě další způsob jakým spouštět kód v přesně daný okamžik - jsou to callback funkce. Existují alternativy pro všechny události stránek. Protože se nejedná

o události, mají mnohem menší spotřebu paměti a také jsou z hlediska struktury kódu mnohem přívětivější.

Framework7 umožňuje několik různých způsobů, jakým způsobem získat stránku. Základním způsobem je načtení pomocí Ajax dotazu z jiného souboru. Je ale také možné vytvořit a načíst dynamickou stránku pomocí JavaScriptu, je možné vložit všechny stránky do jediného souboru, takže všechny stránky již jsou v DOM a není tedy potřeba načíst další. Dalším způsobem je načtení stránky vytvořené pomocí šablonovacího nástroje Template7. Různé způsoby je také možné v jedné aplikaci libovolně kombinovat. Další velmi užitečnou komponentou je navigace, která je vždy spojena s pohledem. Lze tedy mít v jedné aplikaci několik pohledů, kde každý má vlastní navigaci.

Pro manipulaci s DOM Framework7 užívá vlastní knihovnu Dom7, poskytující vysoce výkonné metody. Syntaxe je velmi podobná populární JavaScriptové knihovně jQuery. Dom7 nabízí metody pro práci s třídami elementu, atributy a vlastnostmi, daty, umožňuje pracovat s událostmi, styly, zpracovávat aktuální DOM. Dále nabízí podporu AJAX (Asynchronous JavaScript and XML) pro snadné zasílání dotazů na server.

Právě díky JavaScriptu, HTML 5 a CSS 3 je možné vytvořit webovou aplikaci která se drží doporučení pro vzhled aplikace v operačním systému iOS od společnosti Apple, případně vizuálního jazyka Material Design od Google. Díky tomu, že zobrazení stránek má plně ve své režii Framework7, lze veškeré přechody animovat a docílit tak plynulého procházení aplikace a také efektivních přechodů.

Framework7 nabízí mnoho připravených elementů. Jedním ze základních je seznam, který lze použít k zobrazení seznamu různých objektů, ale také jej lze velice jednoduše použít například jako formulář s před připravenými typy vstupních polí, lze data dynamicky načítat, je připraveno vyhledávání. Mezi další elementy které stojí za zmínku je kalendář, upravená dialogová okna, která se zobrazují přímo v aplikaci, notifikace, boční panely a mnoho dalšího.

Nemalou výhodou je velikost celého frameworku, která se včetně předdefinovaných stylů pro různé elementy použitelné v aplikaci pohybuje v závislosti na použitých modulech řádově ve stovkách kB.

Kapitola 6

Návrh řešení

Přístupový terminál po přiložení RFID karty identifikuje osobu a rozhodne, zda má tato osoba povolený nebo zakázaný přístup do budovy. Identifikace proběhne pomocí unikátního identifikátoru, který má RFID karta v sobě uložený. Terminál na přiložení karty reaguje zvukovým signálem, který je různý pro povolení či zamítnutí přístupu. V případě povoleného přístupu se sepne na určitou dobu relé, přes které je připojeno ovládání elektronického zámku dveří a dále se do vnitřní paměti uloží záznam o přístupu.

Celý systém je postaven na bázi čipu ESP8266, který umožňuje připojení k Wi-Fi síti. Čip umožňuje kromě komunikace přes sériové rozhraní také nahrání firmware, díky němuž je možné do něj nahrát vlastní programy a použít jej jako řídící mikrokontroler. Navíc obsahuje sběrnice SPI, I^2C a I^2S, pomocí kterých mohou být připojeny další periferní zařízení. Tato vlastnost spolu spolu s nízkou cenou umožňuje vytvořit komerčně velmi zajímavý produkt.

Pro správu systému bude sloužit webové rozhraní, které je zabezpečeno jménem a heslem. V něm je možné prohlížet historii přístupů, zaregistrovat novou kartu, upravovat existující karty a také provádět nastavení.

Celé řešení lze rozdělit na dvě části:

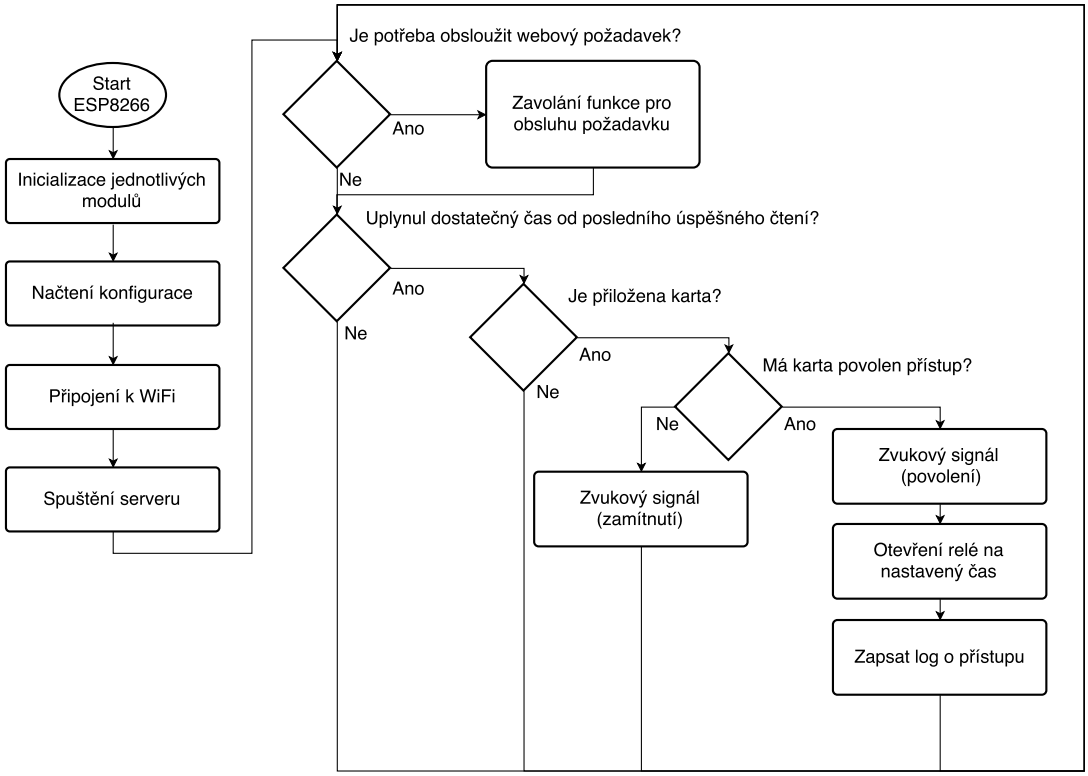
- webové rozhraní pro správu
- vestavěný systém zařizující obsluhu dveří a současně fungující jako webový server

Na ESP8266 poběží webový server, ke kterému se správce připojí pomocí WiFi sítě a umožní mu spravovat povolené karty, nastavit různé vlastnosti systému a zobrazit historii přístupů. Současně bude ESP8266 na základě identifikátoru získaného z karty, která může být přiložena ke čtečce, spínat relé, na kterém jsou připojeny dveře, zapisovat do úložiště log o přístupu.

6.1 Vestavěný systém

Pro řešení této práce se jeví jako nejvhodnější modul NodeMCU V3 obsahující 4MB paměti, který je programovaný pomocí Andruido IDE jako řídící mikrokontrolér. K ESP8266 budou dále připojeny další periferie zajišťující funkčnost, mezi něž patří čtečka čipových karet pro identifikaci uživatele pomocí karty MIFARE Classic a hodiny reálného času.

Na obrázku 6.1 je zobrazen vývojový diagram funkčnosti vestavěného systému. Po zapnutí ESP8266 se inicializuje komunikace s čtečkou karet, s modulem hodin reálného času. Následně je načten z úložiště konfigurační soubor, v němž jsou uloženy informace o SSID sítě, ke které se je třeba připojit a také heslo. Následně se systém připojí na tuto síť a



Obrázek 6.1: Vývojový diagram funkce systému

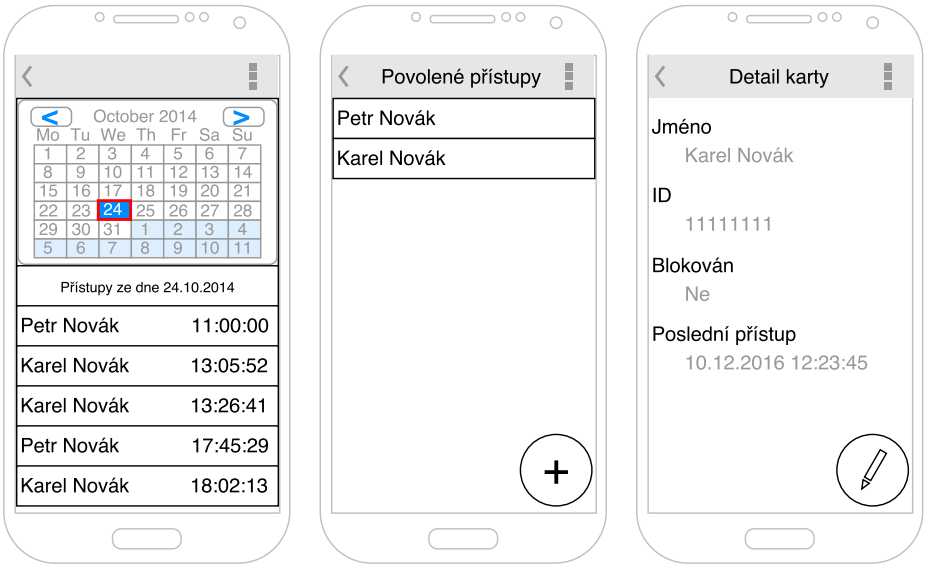
spouští se server. Od tohoto okamžiku se následující část kódu neustále opakuje. Nejdříve je obsloužen dotaz z webu, pokud nějaký je. Následně se vyhodnotí, jestli uplynul nastavený čas od posledního přístupu (řádově sekundy, kvůli několikanásobnému zápisu jednoho přístupu). Pokud ano, a je přiložena karta, ověří se, zda má osoba s touto kartou povolen přístup. Jestliže osoba přístup má, zazní zvukový signál, sepne se relé na určený čas a zapíše se záznam o přístupu. Pokud přístup nemá povolený, zazní zvukový signál značící nepovolený přístup.

6.2 Uživatelské rozhraní

Uživatelské rozhraní bude zobrazeno převážně na mobilním telefonu (smarthphone), případně na tabletu, nebo na počítači. Z tohoto vyplývá, že by rozhraní mělo být responzivního designu, s důrazem na přizpůsobení ovládání dotykové obrazovce.

Po přihlášení do aplikace se zobrazí úvodní stránka. Předpokládám, že častěji bude správce kontrolovat přístupy jednotlivých osob, než upravovat konfiguraci. Proto bude na úvodní stránce kalendář, pomocí kterého se vybere den, který správce zajímá. Výchozím dnem, který se zobrazí, bude aktuální den. Pod kalendářem budou zobrazeny přístupy z vybraného dne. Návrh této stránky je na obrázku 6.2a.

Pomocí menu bude mít správce možnost zobrazit další stránky. Pro úpravu přístupů bude sloužit položka „Správa přístupů“. Na této stránce bude možné prohlížet všechny karty, které jsou v systému. Zobrazuje se název karty namísto identifikátoru karty, z důvodu lepší rozlišitelnosti. Pro přidání nové karty slouží plovoucí tlačítko vpravo dole. Po jeho stisknutí se otevře dialogové okno, které vybídne k přiložení karty ke čtečce. Kliknutím na některou



(a) Návrh domovské obrazovky

(b) Návrh zobrazení uživatelů v systému

(c) Návrh zobrazení detailu karty

Obrázek 6.2: Návrh uživatelského rozhraní

z karet se uživatel dostane na detail karty. Návrh stránky, sloužící pro zobrazení přístupů, je zobrazen na obrázku 6.2b.

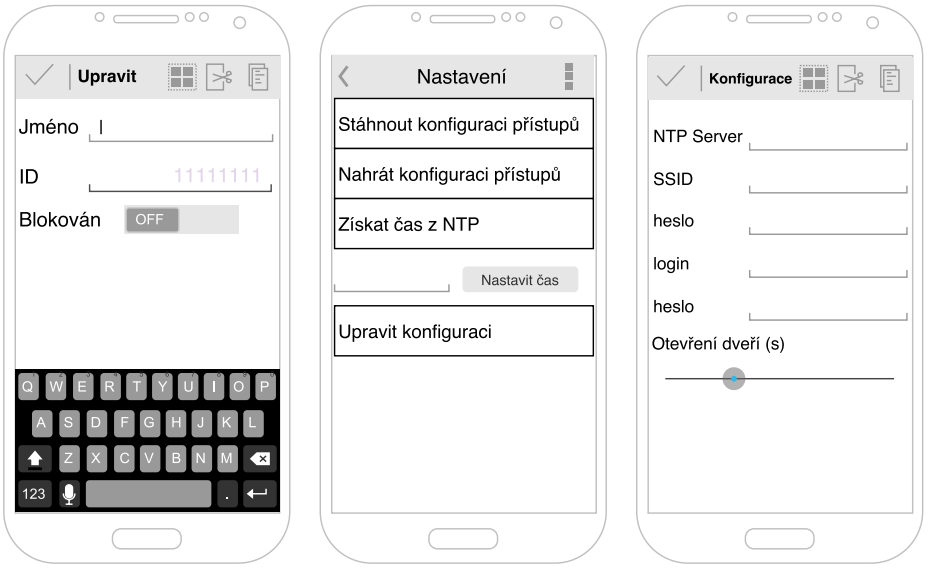
Na obrázku 6.3a je zobrazen návrh detailu karty. Detail karty má za úkol zobrazit tyto informace: jméno karty, unikátní identifikátor karty, informaci zda je zablokována a informaci o posledním přístupu. Dále nabízí okno možnost editace této karty ve formě plovoucího tlačítka.

Po kliknutí na tlačítko editovat se zobrazí stránka editace karty, která je podobná stránce detailu karty. Rozdíl spočívá v umístění formulářových prvků místo statického textu. Editace spočívá v možnosti přejmenovat kartu a také nastavit zablokování karty. Identifikátor ani poslední přístup nelze editovat. Návrh stránky s editací karty je zobrazen na obrázku 6.3a.

Z hlavní nabídky je možné pomocí menu přejít do sekce nastavení. Návrh nastavení je na obrázku 6.3b. V této části aplikace je možné stáhnout soubor obsahující informace o zaregistrovaných kartách, k čemuž slouží tlačítko na stránce. Další tlačítko slouží k nahrání této stažené konfigurace. Po stisknutí tohoto tlačítka bude otevřen v závislosti na platformě standardní dialog pro nahrání souboru. Dále je možné v nastavení upravit čas. Správce má dvě možnosti:

- získání času z NTP serveru
- nastavení správného času pomocí formulářového pole. Vstupní pole je typu „date“, což umožní v závislosti na platformě různým způsobem zadat datum a čas.

Poslední tlačítko na stránce nastavení vede k úpravě konfigurace. Tato stránka slouží pro úpravu konfigurace uložené v paměti systému. Na této stránce bude možné upravit adresu NTP serveru, která se použije, pokud správce v nastavení zvolí možnost „Získat čas z NTP“. Dále je v konfiguraci možné upravit SSID a heslo WiFi sítě, ke které se systém připojí, přihlašovací jméno a heslo pro přihlášení do administrátorského rozhraní a posuvný jezdec,



(a) Návrh zobrazení editace karty

(b) Návrh zobrazení uživatelů v systému

(c) Návrh zobrazení detailu karty

Obrázek 6.3: Návrh uživatelského rozhraní

pomocí nějž lze nastavit dobu, po kterou budou otevřené dveře, v případě že přiložená karta opravňuje k přístupu. Vstupní pole obsahující hesla jsou HTML typu „password“, díky čemuž se zobrazí místo hesla hvězdičky, což brání přečtení hesla náhodným kolemjdoucím. Návrh stránky s úpravou konfigurace je zobrazen na obrázku 6.3c.

Kapitola 7

Implementace

V této kapitole je popsáno, jakým způsobem byl navržený systém implementován.

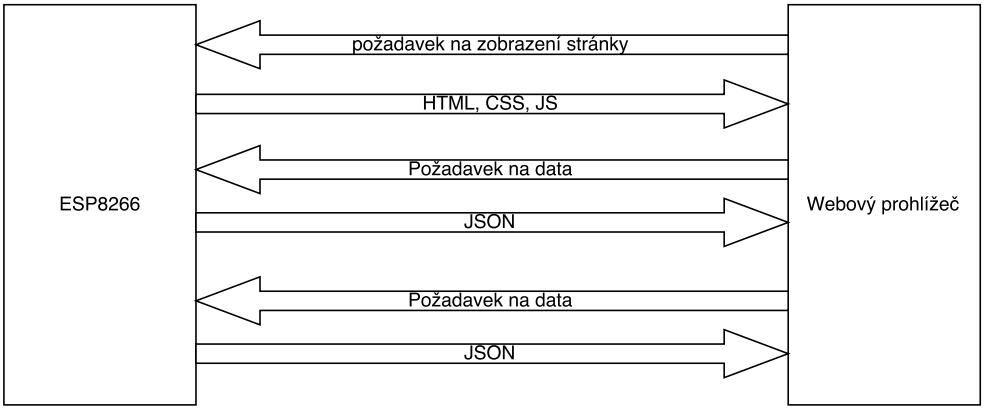
7.1 Webové rozhraní

Jak již bylo zmíněno, jako webový server slouží ESP8266, na kterém vzhledem k výkonu nelze provozovat běžný webový server, jak jej známe z běžných linuxových distribucí. Z tohoto důvodu se nabízely dvě možnosti, jak dynamicky vytvářet obsah webu:

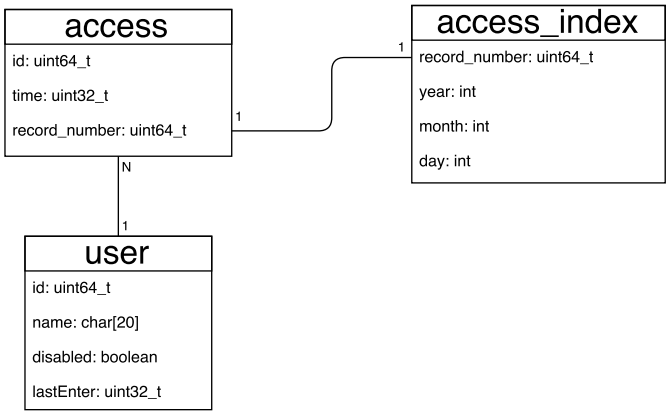
- generovat HTML na serverové straně
- ze serveru odeslat statické soubory a následně data, ze kterých se přímo v prohlížeči vytvoří obsah

Z důvodu co nejmenšího zatěžování ESP8266 jsem zvolil druhý přístup. Na obrázku 7.1 je znázorněna posloupnost zobrazení stránky. Při zaslání požadavku na server je odeslán HTML soubor, který v sobě má další závislosti na CSS, JavaScript a odkazy na další stránku. Po načtení těchto souborů obsahující šablony, se ze serveru na pozadí stránky dalším GET požadavkem stáhne JSON obsahující data k zobrazení. Pomocí těchto dat a šablony je vygenerován HTML obsah který je následně vložen na správné místo.

Pro webové rozhraní jsem zvolil Framework7, protože i když má minimální velikost CSS a JavaScriptových souborů, nabízí všechny potřebné vlastnosti – především se jedná o



Obrázek 7.1: Princip komunikace mezi koncovým zařízením a ESP8266



Obrázek 7.2: Schéma databáze

mobilní rozhraní, které je přizpůsobeno pro zobrazení na malém displeji mobilního telefonu a také nabízí spoustu již předpřipravených prvků.

Kvůli co největší optimalizaci webového rozhraní jsem využil metody minimalizace JavaScriptu, CSS a HTML souborů. Dále bylo využito vložení JavaScriptu a CSS souboru do HTML souboru. Následně bylo využito zabalení souboru pomocí programu gzip. K tomuto účelu byl upraven skript [16] pro nástroj Gulp běžící v prostředí Node.js. Tento skript nejdříve provede minimalizaci CSS souborů odstraněním komentářů a veškerých mezer a sloučí je do jednoho css souboru. Následně provádí minimalizaci JavaScriptových souborů. Podobně jako v případě CSS odstraní komentáře, odstraní mezery a navíc nahradí lokální proměnné ve funkcích, za co nejkratší možné názvy a následně je opět sloučí do jednoho souboru. Poté je provedena i minimalizace HTML souboru. Následně jsou všechny soubory pomocí gzip metody zkomprimovány, což zapříčiní, že je potřeba přenést menší objem dat při zachování funkčnosti.

7.2 Obsluha ESP8266

Pro programování mikrokontroléru jsem zvolil Arduino IDE především z důvodu známého prostředí a také předchozích zkušeností z programováním v tomto prostředí.

Pro ukládání přístupů jsem zvolil knihovnu EDB (Extended Database Library) která umožňuje teoreticky až 2^{32} záznamů o velikosti až 2^{16} bytů, reálně je ale omezena volným místem v úložišti. Uživatelé budou v další databázi, kde bude řádek **id**, což je unikátní identifikátor karty, cizím klíčem v tabulce přístupů. Vzhledem k tomu, že databáze je ve skutečnosti jen serializace dat s možností přístupu na n-tý záznam, je vyhledávání velice neefektivní z důvodu nutnosti sekvenčního procházení všech záznamů, proto jsem přidal druhou databázi, ve které má databáze přístupů index každého dne v měsíci ve zvláštní databázi, což umožní rychleji přečíst záznamy z daného dne bez nutnosti procházet všechny záznamy. Struktura databází je na obrázku 7.2.

Databáze jsou spolu s konfigurací a soubory obsahujícími zdrojový kód webu uloženy v SPIFFS. V souborovém systému je potřeba oddělit soubory pro web a ostatní soubory, z toho důvodu je v souborovém souboru speciální složka pro web a jiná pro databázi.

Pro obsluhu modulu reálného času jsem zvolil z několika různých knihoven RTClib, především z důvodu dostupnosti metody vracející časové razítko ve formátu počtu sekund od 1. 1. 1970, což jsem vyhodnotil jako vhodný formát pro uchovávání času v databázi.

Obsluhu čtečky zajišťuje knihovna MFRC522, umožňující kromě přečtení identifikátoru karty také práci s vnitřní pamětí karty.

7.3 Bezpečnost

Protože aplikace má sloužit jako přístupový systém do budovy, je žádoucí aby byla odpovídajícím způsobem zajištěna bezpečnost. Bezpečnost bych tedy rozdělil do tří úrovní. Jde o zabezpečení karet a s tím i přístupu do objektu, zabezpečení administrátorského rozhraní a zabezpečení WiFi připojení.

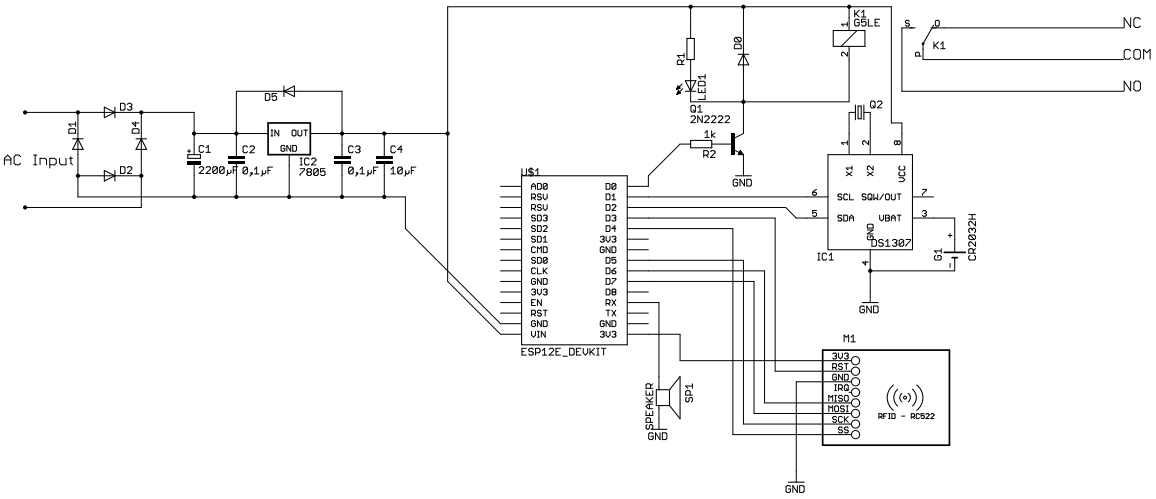
Jako identifikační karta je používána karta MIFARE Classic. V mnoha přístupových systémech se touto kartou uživatelé identifikují tím způsobem, že je přečten unikátní identifikátor karty a na jeho základě je uživatel autentizován. Problémem je, že unikátní identifikátor není problém klonovat. Existují totiž karty, u nichž je povolena modifikace tohoto identifikátoru. Jedná se většinou o karty, které pocházejí z neoficiálních distribucí, a nebo přímo výrobce u nich umožní tuto změnu. Z tohoto důvodu jsem se rozhodl využít paměti v kartě a v druhém sektoru nastavit při registraci karty do systému kryptografický klíč, kterým je chráněn sektor. Následně tedy vždy při autentizaci osoby se kromě unikátního identifikátoru vyzkouší i autentizace do druhého sektoru.

Druhou úrovní je zabezpečení přístupu do administrace. Systém se pokouší připojit na známou WiFi síť, kterou je možné nakonfigurovat v nastavení v položce „Upravit konfigurační soubor“. Tato síť by neměla být dostupná nikomu, kromě správce. Pokud se připojení nepodaří, přepíná se systém do režimu přístupového bodu, ve kterém vytváří svoji vlastní síť. V obou případech se počítá s WiFi sítí zabezpečenou pomocí WPA2, která je považována při kombinaci s netriviálním heslem odhalitelným hrubou silou za bezpečnou. Další úrovní je zabezpečení samotného uživatelského rozhraní. Zde je využita autentizace pomocí HTTP protokolu. Vhodnější by byla autentizace pomocí protokolu HTTPS, ale vyžadované kryptografické operace by ESP8266 mělo problém zvládat, protože jsou paměťově a výkonově náročné. Vzhledem k tomu, že se nepočítá s tím, že by ESP8266 bylo připojeno na veřejné síti, kde by komunikace mohla být odposlechnuta, mělo by toto zabezpečení dostačovat. Případný útočník by musel nejdříve získat přístup do WiFi sítě a následně odposlechnout komunikaci správce, odkud by šlo získat přístupová práva.

7.4 Realizace prototypu

Prototyp se skládá z několika modulů, které bylo potřeba propojit. Jádrem celého řešení je modul NodeMCU obsahující ESP8266. Karty MIFARE Classic jsou čteny pomocí čtečky RC522 pracující s frekvencí 13,56 MHz, která komunikuje po sběrnici SPI. Pro získávání přesného času byl připojen modul hodin reálného času, konkrétně DS1307 komunikující pomocí I^2C.

Protože lze v místě, kde bude celý systém umístěn, očekávat zdroj střídavého napětí 12 V, případně 24 V, bylo potřeba vytvořit napájecí obvod. Na vstupu je Graetzův můstek, který usměrní střídavý proud na stejnosměrný. Pro vyhlazení kladných půlvln je připojen kondenzátor C1. Kondenzátor C2 je v obvodu umístěn pro případ rychlé změny zatěžovacího proudu. Pro stabilizaci vstupního napětí na hodnotu 5V je použit lineární stabilizátor kladného napětí s označením 7805. Dioda D5 chrání stabilizátor proti zpětnému proudu. Kondenzátor C3 slouží jako blokovací a kondenzátor C4 je pro jemnou filtraci výsledného proudu. [6]



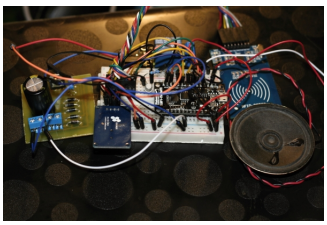
Obrázek 7.3: Schéma zapojení

Pro ovládání elektronického zámku dveří je připojeno elektromagnetické relé a pro zvukovou signalizaci reproduktor.Elektromagnetické relé je připojeno přes NPN tranzistor, pomocí kterého je připojována zem, čímž se uzavírá obvod. Schéma celého zapojení je zobrazeno na obrázku 7.3.

Kapitola 8

Výsledek práce

Byl vytvořen funkční prototyp „inteligentního přístupového systému“. Tvoří jej uživatelské rozhraní a program běžící na mikrokontroléru a umožňující řídit přístupy. Prototyp je na obrázku 8.1.



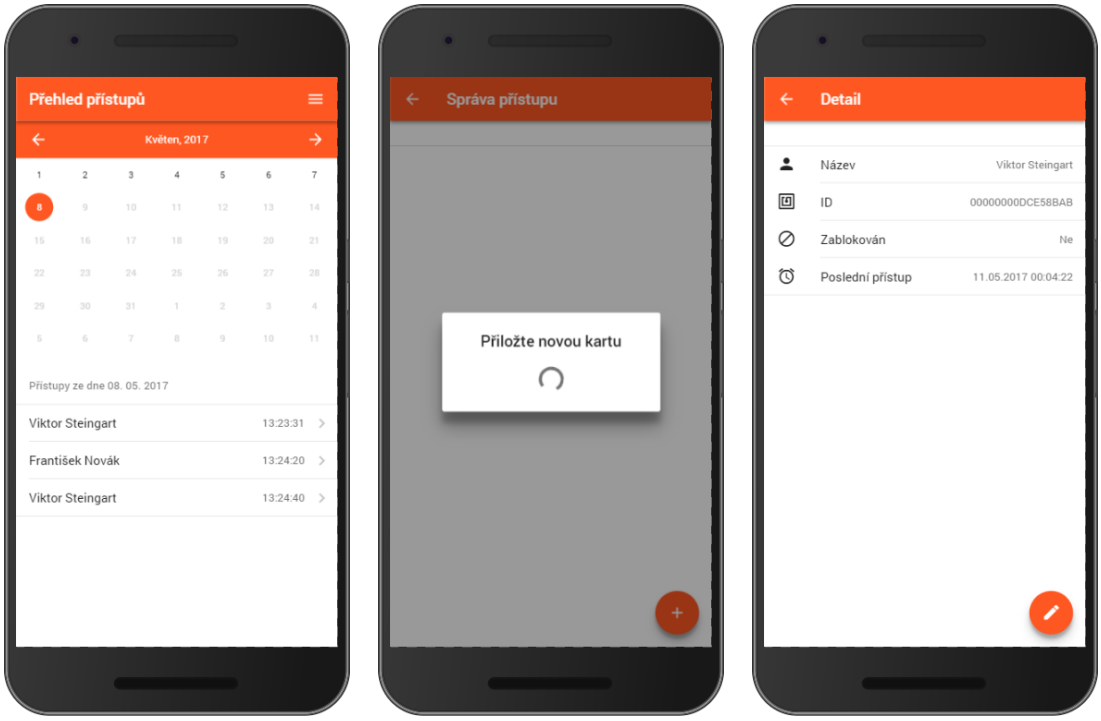
Obrázek 8.1: Fotorafie prototypu

Podařilo se dosáhnout velikosti zdrojových souborů webu 68 kB, což se povedlo především díky veškerým optimalizacím. Webová stránka se načítá a vykresluje na osobním počítači při prvním dotazu průměrně 4,5 sekund. Při opakovaném načítání jsou statické soubory v cache paměti prohlížeče, čímž se doba načítání a vykreslování snížila na průměrně půl sekundy . Webové rozhraní lze zobrazit na širokém spektru zařízení od mobilních telefonů až po osobní počítač díky responsivnímu designu.

Počet přístupů za den	**doba trvání dotazu (s)**
0	0,12
1	0,12
10	0,13
100	0,36
200	0,57
500	1.39
1000	2.91

Tabulka 8.1: Závislost počtu přístupů v dotazovaném dni na době trvání dotazu

Nejdéle trvá načtení historie přístupů. Tabulka 8.1 ukazuje vliv počtu přístupů v daném dni na dobu, jakou trvá celý dotaz. Lze pozorovat, že s rostoucím počtem záznamů, které je potřeba přečíst, roste také doba, která zabere serveru odpověď, což bylo očekávané vzhledem k povaze databáze.

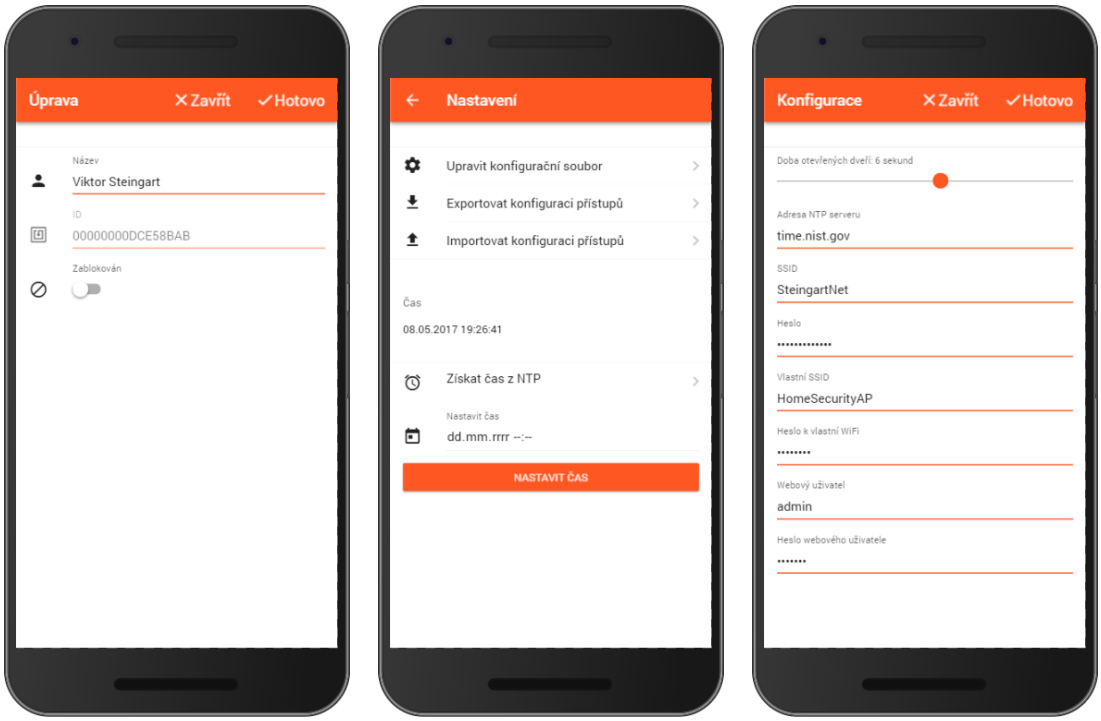


(a) Domovská obrazovka (b) Přidávání nové karty (c) Detail karty

Obrázek 8.2: Snímky obrazovky uživatelského rozhraní

Na obrázcích 8.2 a 8.3 jsou zachyceny snímky obrazovky s webovým rozhraním. Obrázek 8.2a zachycuje úvodní stránku, která se správci zobrazí po přihlášení. Pomocí kalendáře lze přepínat mezi libovolnými dny v minulosti. Kliknutím na některý ze záznamů se správce lze dostat na detail karty, jenž je zobrazen na obrázku 8.2c. Na druhém obrázku 8.2b je obrazovka, která se zobrazí při přidávání nové karty. Další obrázek 8.3a zachycuje formulář pro editaci registrované karty. Na obrázku 8.3b je snímek stránky s nastavením aplikace. V nastavení je možnost exportovat a importovat konfiguraci přístupů pro možnost jednoduchého nastavení dalších zařízení. Další část se týká nastavení času. Pokud je zařízení připojeno k internetu, lze využít synchronizace pomocí NTP. Dále je možnost manuálně nastavit čas. Stiskem tlačítka „Upravit konfigurační soubor“ se lze dostat do editace konfiguračního souboru, ve kterém lze upravit nastavení WiFi sítě jak pro připojení k existující, tak pro vlastní síť, přihlašovací údaje do webového rozhraní, adresu NTP serveru a také dobu, po jakou budou otevřeny dveře. Tato konfigurace je zobrazena na obrázku 8.3c .

Do systému je možné uložit přes 131 tisíc záznamů o přístupech, které mohou být indexovány v 1364 dnech. Do systému může být zaregistrováno až 255 karet. Na systému byla ověřena funkčnost i po dlouhé době běhu – více než 24 hodin byl systém zapnutý a při každém průchodu kolem něj byly přiloženy náhodně dvě karty, z čehož jedna byla zaregistrována a tedy povolovala průchod a druhá povolená nebyla a dle očekávání zazněl zvukový signál znázorňující odmítnutí přístupu, relé nebylo sepnuto a přístup nebyl zapsán.



(a) Editace karty (b) Nastavení (c) Konfigurace

Obrázek 8.3: Snímky obrazovky uživatelského rozhraní

Kapitola 9

Závěr

V mé bakalářské práci se mi podařilo vytvořit prototyp přístupového systému, který lze konfigurovat pomocí mobilního telefonu, případně počítače skrze WiFi síť.

Byl vytvořen webový portál umožňující zobrazení přístupů a konfiguraci celého řešení. Dále bylo navrženo zapojení RFID čtečky, hodin reálného času a elektromagnetického relé na desku NodeMCU osazenou čipem ESP8266. Navržené zapojení bylo realizováno a následně byl vytvořen řídící program pro mikrokontrolér.

Musel jsem vyřešit otázky týkající se optimalizace vytvořeného webového portálu pro běh na mikrokontroléru, což zahrnovalo především techniky minimalizace souborů, zmenšení počtu souborů na minimum a využití *gzip* komprese u všech souborů. Dále jsem zvýšil bezpečnost použitím jednoho ze sektorů v paměti karet Mifare, u kterého jsem nastavil klíč, který se ověřuje společně s unikátním identifikátorem karty.

Do budoucna by bylo vhodnější využít karty s vyšší bezpečnostní, například Mifare DESfire. Dále by bylo vhodné rozšířit možnosti synchronizace při využívání systému například u více vchodů do budovy automatickou synchronizací nastavení a centralizací záznamů o přístupu. Další možností rozšíření je podpora dalších typů vstupů do budovy. Kromě stávajícího elektronicky ovládaného zámku by bylo možné přidat řízení elektromotorů pro ovládání různých typů vrat, nebo například závory na parkovišti.

Literatura

[1] Bermuda TCD Uses TransCores RFID Technology for Electronic Vehicle Registration System. *Wireless News*, Feb 13 2008.

[2] Benchoff, B.: New Chip Alert: The ESP8266 WiFi Module (It's $5). August 26, 2014, [Online; navštíveno 03.05.2017].
URL http://hackaday.com/2014/08/26/new-chip-alert-the-esp8266-wifi-module-its-5/

[3] Benchoff, B.: The Current State of ESP8266 Development. September 6, 2014, [Online; navštíveno 03.05.2017].
URL http://hackaday.com/2014/09/06/the-current-state-of-esp8266-development/

[4] Brisbin, S.: *Wi-fi*. Praha: Neocortex, 2003, ISBN 8086330133.

[5] Courtois, N. T.; Nohl, K.; O'Neil, S.: Algebraic Attacks on the Crypto-1 Stream Cipher in MiFare Classic and Oyster Cards. Cryptology ePrint Archive, Report 2008/166, 2008.

[6] Stabilizace pomocí obvodu 7805. Elektroportal, [Online; navštíveno 10.05.2017].
URL http://www.elektroportal.xf.cz/index.php?p=stabilizace-pomoci-obvodu-7805

[7] ESP8266EX Datasheet. 2017, [Online; navštíveno 04.05.2017].
URL http://espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf

[8] Finkenzeller, K.: *RFID Handbook*. Hoboken, NJ: Wiley, třetí vydání, c2010, ISBN 9780470695067.

[9] Kilts, S.: *Advanced FPGA design – architecture, implementation, and optimization*. New York: IEEE Press, Wiley-Interscience, 2007, ISBN 9780470054376.

[10] Koelle, A.; Depp, S.; Freyman, R.: Short-range radio-telemetry for electronic identification, using modulated RF backscatter. *Proceedings of the IEEE*, ročník 63, č. 8, 1975: s. 1260–1261, ISSN 00189219, doi:10.1109/PROC.1975.9928.

[11] Landt, J.: The history of RFID. *Potentials, IEEE*, ročník 24, č. 4, 200510: s. 8–11, ISSN 02786648, doi:10.1109/MP.2005.1549751.

[12] DS1307 – 64 x 8, Serial, I2C Real-Time Clock. 2015, [Online; navštíveno 03.05.2017].
URL https://datasheets.maximintegrated.com/en/ds/DS1307.pdf

[13] MFRC522 – Standard performance MIFARE and NTAG frontend. 27 April 2016, [Online; navštíveno 12.05.2017].
URL https://www.nxp.com/documents/data_sheet/MFRC522.pdf

[14] MIFARE Classic EV1 1K - Mainstream contactless smart card IC for fast and easy solution development. 3 March 2014, [Online; navštíveno 02.05.2017].
URL http://www.nxp.com/documents/data_sheet/MF1S50YYX_V1.pdf

[15] I^{2}C - bus specification and user manual. 4 April 2014, [Online; navštíveno 07.05.2017].
URL http://www.nxp.com/documents/user_manual/UM10204.pdf

[16] Pérez, X.: Optimizing files for SPIFFS with Gulp. [Online; navštíveno 05.05.2017].
URL http://tinkerman.cat/optimizing-files-for-spiffs-with-gulp/

[17] Stockman, H.: Communication by Means of Reflected Power. *Proceedings of the IRE*, ročník 36, č. 10, 1948: s. 1196–1204, ISSN 00968390, doi:10.1109/JRPROC.1948.226245.

[18] Tišnovský, P.: Externí sériové sběrnice SPI a I2C. 30. 12. 2008, [Online; navštíveno 07.05.2017].
URL https://www.root.cz/clanky/externi-seriove-sbernice-spi-a-i2c/

[19] Tišnovský, P.: Komunikace po sériové sběrnici I^{2}C. 8. 1. 2009, [Online; navštíveno 07.05.2017].
URL
https://www.root.cz/clanky/komunikace-po-seriove-sbernici-isup2supc/

[20] I^{2}C. Wikimedia Foundation, 2001-2017, [Online; navštíveno 07.05.2017].
URL https://en.wikipedia.org/w/index.php?title=I%C2%B2C&oldid=776628648

[21] Serial Peripheral Interface Bus. Wikimedia Foundation, 2001-2017, [Online; navštíveno 07.05.2017].
URL https://en.wikipedia.org/w/index.php?title=
Serial_Peripheral_Interface_Bus&oldid=776789750

[22] I^{2}S. Wikimedia Foundation, 2001-217, [Online; navštíveno 07.05.2017].
URL https://en.wikipedia.org/w/index.php?title=I%C2%B2S&oldid=775659338

[23] Yan, K.: *Network protocols handbook*. Saratoga: Javvin, vyd. 2. vydání, 2005, ISBN 0974094528.

Přílohy

Příloha A

Obsah CD

- `/doc/` – tento dokument a jeho zdrojové soubory
- `/data/` – zdrojové soubory webu, konfigurace
- `/BP/BP.ino` – zdrojový soubor
- `/BP/data/` – složka která je nahrána do ESP8266

Příloha B

Základní uspořádání aplikace napsané ve Framework7

```
<!DOCTYPE html>
<html>
  <head>
    <!-- Required meta tags-->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
        maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <!-- Color theme for statusbar -->
    <meta name="theme-color" content="#2196f3">
    <!-- Your app title -->
    <title>My App</title>
    <!-- Path to Framework7 Library CSS, Material Theme -->
    <link rel="stylesheet" href="path/to/framework7.material.min.css">
    <!-- Path to Framework7 color related styles, Material Theme -->
    <link rel="stylesheet" href="path/to/framework7.material.colors.min.css">
    <!-- Path to your custom app styles-->
    <link rel="stylesheet" href="path/to/my-app.css">
  </head>
  <body>
    <!-- Views -->
    <div class="views">
      <!-- Your main view, should have "view-main" class -->
      <div class="view view-main">
        <!-- Pages container, because we use fixed navbar and toolbar, it has
            additional appropriate classes-->
        <div class="pages navbar-fixed toolbar-fixed">
          <!-- Page, "data-page" contains page name -->
          <div data-page="index" class="page">

            <!-- Top Navbar. In Material theme it should be inside of the page-->
            <div class="navbar">
              <div class="navbar-inner">
                <div class="center">Awesome App</div>
              </div>
            </div>
```

```
            <!-- Toolbar. In Material theme it should be inside of the page-->
            <div class="toolbar">
              <div class="toolbar-inner">
                <!-- Toolbar links -->
                <a href="#" class="link">Link 1</a>
                <a href="#" class="link">Link 2</a>
              </div>
            </div>

            <!-- Scrollable page content -->
            <div class="page-content">
              <p>Page content goes here</p>
              <!-- Link to another page -->
              <a href="about.html">About app</a>
            </div>
          </div>
        </div>
      </div>
    </div>
    <!-- Path to Framework7 Library JS-->
    <script type="text/javascript" src="path/to/framework7.min.js"></script>
    <!-- Path to your app js-->
    <script type="text/javascript" src="path/to/my-app.js"></script>
  </body>
</html>
```