ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta elektrotechnická
Katedra telekomunikační techniky

Inteligentní zásuvka – SW část

Intelligent socket – SW part

bakalářská práce

Studijní program: Komunikace, multimédia a elektronika
Studijní obor: Síťové a informační technologie

Vedoucí práce: Ing. Pavel Bezpalec, Ph. D.

Jakub Slatinský

Praha 2014

Čestné prohlášení

Prohlašuji, že jsem zadanou bakalářskou práci zpracoval sám s přispěním vedoucího práce a konzultanta a používal jsem pouze literaturu v práci uvedenou. Dále prohlašuji, že nemám námitek proti půjčování nebo zveřejňování mé bakalářské práce nebo její části se souhlasem katedry.

Datum: 23. 5. 2014

.....

podpis bakalanta

Anotace:

Tato bakalářská práce se zabývá návrhem a realizací ovládacího systému inteligentních zásuvek, disponující funkcionalitou pro zapnutí nebo vypnutí přívodu elektrické energie a měření spotřeby elektrické energie připojeného elektrického přístroje. Tento systém je založený na cetrálním řídícím uzlu komunikace, který představuje miniaturní počítač Raspberry Pi. V práci je popsán komunikační protokol použitý pro ovládání inteligentních zásuvek, ovládací program a je provedena analýza komunikace pomocí programu Wireshark

Klíčová slova: inteligentní zásuvka, Raspberry Pi, Wireshark, komunikační protokol

Summary:

This thesis describes the design and implementation of intelligent sockets control systems, possessing functionality on or off the power supply and measure the power consumption of the connected electrical devices. This system is based on a central control node communication, which is a miniature computer Raspberry Pi. The thesis describes the communication protocol used for controlling intelligent sockets, control program and the analysis of communication using Wireshark

Index Terms: Inteligent socket, Raspberry Pi, Wireshark, communication protocol

Obsah

1. Analýza zadání

Zadáním práce je navrhnout a realizovat inteligentní zásuvku a její ovládací podpůrné softwarové a hardwarové prostředky, tak aby sloužila jako součást inteligentní budovy. Zásuvka bude umožňovat minimálně dálkové spínání a měření spotřeby. Dále ještě je potřeba zvolit vhodnou technologii použitou pro komunikaci mezi hlavním serverem a zásuvkou.

V zadání je kromě samotné inteligentní zásuvky zmíněn i hlavní server. Kromě návrhu softwarových a hardwarových prostředků samotné inteligentní zásuvky, je nutné ještě vybrat vhodné hardwarové a softwarové prostředky pro hlavní server. Hlavní myšlenka celé práce, je vytvořit centralizovaný systém ovládání zásuvek. Celkový cíl bakalářské práce, lze rozdělit do několika dílčích úkolů, které je nutné vyřešit.

1. Vytvoření softwaru, který bude simulovat chování reálné inteligentní zásuvky – schopnost dálkového spínání a měření spotřeby elektrické energie

2. Vytvoření ovládacího programu, který bude zastřešovat ovládání inteligentních zásuvek a který bude běžet na hlavním serveru

3. Vytvoření uživatelského rozhraní, které umožní interakci uživatele a ovládacího programu inteligentních zásuvek a také bude sloužit uživateli pro zobrazení informací o naměřené spotřebě tabulkou nebo pomocí grafu

4. Výběr hardware pro hlavní server a instalace všech potřebných softwarových prostředků, které umožní běh ovládacího programu a uživatelského rozhraní na hardware hlavního serveru

5. Vytvoření komunikačního protokolu, pomocí kterého bude hlavní server komunikovat se všemi inteligentními zásuvkami.

6. Způsob jakým budou povely pro zapnutí, vypnutí a měření spotřeby distribuovány mezi všechny inteligentní zásuvky v domácnosti

7. Výběr vhodné technologie, pomocí které bude komunikovat hlavní server s inteligentní zásuvkou s ohledem na praktičnost a cenu realizace použité technologie.

2. Úvod do problematiky

Pro správné porozumění problematice a následnému řešení je nutné definovat a vysvětlit, jak mají být pojmy inteligentní zásuvka a hlavní server chápany.

Inteligentní zásuvka

Pojem inteligetní zásuvka se skládá ze dvou slov – zásuvka a inteligence. Pojem zásuvka představuje běžnou síťovou zásuvku umožňující připojení elektrických spotřebičů k přívodu elektrické energie. Pojmu inteligence je nutné rozumět jako schopnosti, které jsou běžné síťové zásuvce přidány. Mezi tyto schopnosti patří možnost zásuvku odpojit od přívodu elektrické energie respektive takto odpojenou zásuvku opětovně připojit na přívod elektrické energie. Tato schopnost je užitečná z hlediska bezpečnostního tak i z hlediska ekonomického. Z hlediska bezpečnostního lze zamezit úrazu elektrickým proudem v případech, kdy se neopatrná osoba může dostat k vývodům volné nepoužívané zásuvky. Pokud je daná zásuvka aktuálně nepoužíváná, může být odpojena od přívodu elektrické energie a tak se dá úrazu elektrickým proudem zabránit. Z hlediska ekonomického lze zásuvku odpojit od přívodu elektrické energie v případě, kdy připojené zařízení je aktivně nepoužívané a nachází se pouze v pohotovostním režimu, při kterém stále odebírá menší množství elektrické energie a tím tyto ztráty eliminovat a snížit tak celkovou spotřebu domácnosti.

Další schopností inteligentní zásuvky je možnost měření odebírané elektrické energie připojeným elektrickým spotřebičem. Tato schopnost je především informativního charakteru, aby bylo možné nahlédnout na aktuální spotřebu zařízení a případně sledovat historiii odebírané energie ze sítě prostřednictvím konkrétní zásuvky. Jak údaje o naměřené spotřebě dále využít při provozu inteligentní budovy není náplní této bakalářské práce.

Pokročilou schopností inteligentní zásuvky pak může být například funkce proudového chrániče. Inteligentní zásuvka by tak obsahovala stejné funkce, kterými disponuje proudový chránič a pokud nastane situace, kdy se zařízení porouchá zásuvka se sama odpojí od přívodu elektrické energie a tím připojené zařízení ochrání.

Další možnou pokročilou vlastností zásuvky by mohla být schopnost plánovaného zapnutí respektive vypnutí na základě nějaké vnější události, kterou by vyhodnotila řídící jednotka inteligentní domácnosti.

Hlavní server

Hlavní server představuje zařízení, prostřednictvím kterého jsou inteligentní zásuvky ovládány uživatelem. Inteligentní zásuvka samotná by mohla obsahovat vlastní ovládací rozhraní (například webové rozhraní), nicméně při větším počtu inteligentních zásuvek v domácnosti by ovládání jednotlivých zásuvek nebylo příliš efektivní. Zavedením centrálního uzlu se pak ovládání všech zásuvek zjednodušší. Ovládací rozhraní se nachází pouze na hlavním serveru. Pomocí ovládacího rozhraní uživatel předá hlavnímu serveru svůj požadavek. Hlavním serverem je požadavek zpracován a následně je zaslán zásuvce příslušný povel.

Informace o všech dostupných zásuvkách se nachází pouze na jediném místě. Dále naměřená data o spotřebě elektrické energie všech zásuvek jsou ze zásuvkami odesílány hlavnímu serveru, který příslušná data ukládá a příslušně prezentuje uživateli.

3. Příklady jiných inteligentních zásuvek

Na trhu lze již najít funkční řešení realizovaných inteligentních zásuvek. Níže jsou uvedeny některé z nich.

3.1. GSM spínač IQSW-GSM

Zařízení je určeno pro ovládání libovolného elektrického spotřebiče z mobilního telefonu. GSM spínač se zapojí do elektrické zásuvky a spotřebič, který se má ovládat se připojí do zásuvky na GSM spínači.

Aplikační možnosti GSM zásuvky:

- Zapínání a vypínání spotřebičů pomocí SMS a prozvoněním: 230V, 16A
- Zapínaní a vypínání spotřebičů pomocí SMS a prozvoněním: 50V, 0.5A
- Reset serverů
- Monitor stavu vstupu a teploty
- Monitorování prostoru přídavnými senzory: detekce pohybu, úniku plynu, otevření dveří
- Odposlech
- Funkce termostatu
- Funkce teplotního alarmu
- Funkce časového pánovače (plánované spínaní)
- Nastavení alarmu pro funkci zabezpečení
- Možnost ovládání SMS z Internetu
- Napájení pro zařízení připojená do vstupu JACK 5V

Obr. 3.1: GSM spínač IQSW-GSM. Převzato z [11]

Způsob ovládání:

V manuálu zařízení je popsán způsob ovládání zásuvky. Konkrétní příkaz je poslán na číslo SIM karty vložené do GSM zásuvky. SMS příkazy jsou dvojího druhu – ovládací a konfigurační. Po přijmutí příkazu GSM zásuvka odešle zpět potvrzovací příkaz.

Cena: 2902,79 Kč (k 28.4.2014)

3.2. GSM spínač GS900L + Easy Socket

GSM spínač GS900L slouží k ovládání napájení zařízení pomocí SMS a prozvoněním pomocí mobilního telefonu. Navíc obsahuje bezdrátový interface, který umožňuje připojit až 10 externích Easy Socket zásuvek

Aplikační možnosti GSM zásuvky:

- Zapínání a vypínání spotřebičů pomocí SMS a prozvoněním: 240V, 10A
- Restart serverů
- Funkce časového plánovače

Aplikační možnosti Easy Socket zásuvky:

- Zapínaní a vypínání spotřebičů 230V, 10A

Způsob ovládání:

Způsob ovládání popsaný v manuálu zařízení je obdobný jako způsob popsaný výše u GSM spínače. Kromě ovládání samotného spínače je možné ovládat až 10 Easy Socket zásuvek prostřednictvím GSM spínače. Pro ovládání těchto Easy Socket zásuvek je nejprve nutné je vložit do paměti GSM spínače pomocí jejich sériového čísla opět prostřednictvím SMS a přiřadit jim alias. Poté je možné pomocí SMS zprávy, ve které je uveden příkaz a alias, odeslané na GSM spínač ovládat konkrétní Easy Socket zásuvku. Bezdrátový přenos je realizován na frekvenci 868 MHz. Dále je veškerý bezdrátový přenos je obousměrný a zabezpečený plovoucím kódem.

Obr. 3.3: GSM spínač GS900L. Převzato z [11]

Obr. 3.2: Easy Socket. Převzato z [11]

Cena GSM spínače GS900L: 2286,90 Kč (k 28.4.2014)

Cena Easy Socket zásuvky: 1439,90 Kč (k 28.4.2014)

3.3. IP Watchdog

IP Watchdog je zařízení pro automatické hlídání funkce a restartování zařízení připojených do výstupu 230VAC, umožňuje také manuální zapínání a vypínání výstupní zásuvky.

Aplikační možnosti zařízení IP Watchdog:

- Zapínaní a vypínání spořebičů pomocí HTTP: 230V, 16A
- Hlídácí funkce a funkce testování na základě pravidel

Způsob ovládání:

Zařízení disponuje jedním ethernetovým rozhraním a svou správu umožňuje pomocí protokolu HTTP, tedy pomocí běžného webového prohlížeče. Po zapojení je zařízení dostupné na své výchozí IP adrese. Zadáním této výchozí IP adresy do prohlížeče se zobrazí informační webové rozhraní. Vedle mnoha rozdílných položek menu se zde nachází položka Control SOCKET, pomocí které je možné výstupní zásuvku ovládat.

Obr. 3.4: IP Watchdog. Převzato z [11]

Cena: 2178 Kč (k 30.4.2014)

4. Komunikační protokol

Před vlastní tvorbou programu je potřeba vytvořit návrh komunikačního protokolu. Podle pravidel protokolu pak bude probíhat komunikace mezi hlavním serverem a zásuvkou. Komunikační protokol dále definuje formát dat, které přenáší a také jejich význam pro ovládání zásuvek a sběr dat o naměřené spotřebě. Při návrhu celého systému komunikace mezi zásuvkou a hlavním serverem je vhodné využít vlastností již existujících funkčních protokolů, například z rodiny protokolů TCP/IP, analyzovat tyto protokoly a uvážit jaké zásadní vlastnosti komunikace musí splňovat.

4.1. Způsob komunikace

Hlavní server má roli centrálního řídícího uzlu komunikace. Komunikace mezi inteligentní zásuvkou a hlavním serverem je vždy inicializována ze strany serveru. Aby server obhospodařil všechny ovládané zásuvky při sběru dat o naměřené spotřebě, používá se techniky round-robin, kdy na hlavním serveru je udržován seznam všech aktivně ovládaných zásuvek a hlavní server vždy v pravidelném časovém intervalu každou zásuvku zvlášť osloví. V případě, že se nachází inteligentní zásuvka ve stavu zapnuto, odpovídá hlavnímu serveru zprávou obsahující data o naměřené spotřebě. V případě, že se nachází ve stavu vypnuto odpovídá serveru zprávou, ve které potvrzuje svoji přítomnost a správou funkčnost. Vyjma periodické obsluhy všech zásuvek, může být asynchroně odeslána z hlavního serveru zpráva, která je odeslána na základě události způsobené uživatelem.

Inspirací pro tento způsob komunikace byl způsob, jakým probíhá komunikace na univerzální sběrnici USB, kdy datové přenosy na sběrnici řídí jeden řídící uzel – hostitel. Další inspirací byl takzvaný centralizovaný systém měření dat. Tento systém obsahuje jeden centrální uzel, do které jsou připojena všechna měřící zařízení. Centrální uzel je také centrem chytrosti celého systému, kdežto měřící zařízení, kromě samotného měření určité veličiny neobsahují žádnou další funkcionalitu.

4.2. Definice zpráv a jejich význam

Mezi hlavním serverem a inteligentními zásuvkami dochází k výměně datagramů s textovými zprávami. Po přijmutí zprávy, od hlavního serveru, inteligentní zásuvka tuto zprávu příslušně vyhodnotí a odešle hlavnímu serveru potvrzující zprávu, že přijatou zprávu správně zpracovala. Podle zadání musí existovat zprávy pro zapnutí respektive vypnutí zásuvky, zpráva pro požadavek na odeslání dat o změřené spotřebě elektrické energie a inicializační zpráva, která umožní detekci všech přítomných inteligentních zásuvek. Seznam zpráv s jejich textovým tvarem a významem je popsán v tabulce níže.

Textový tvar	Směr	Význam	Odpověď
„ON“	Server - Zásuvka	Zapnutí vypnuté zásuvky	„OK ON“
		Periodický odběr naměřených dat ze zapnuté zásuvky	„OK ON <DATA>
„OFF“	Server - Zásuvka	Vypnutí zapnuté zásuvky	„OK OFF“
		Periodická kontrola přítomnosti zásuvky	„OK OFF“
„SCAN“	Server - Zásuvka	Detekce nových zásuvek nebo neovládaných zásuvek	„OK <STAV>“

Tab. 4.1: Přehled textových zpráv pro ovládání inteligentních zásuvek. Vlastní zdroj

Pro lepší názornost a vysvětlení je na obrázku Obr. 4.1. zobrazen statový diagram komunikačního protokolu pro ovládání zásuvek a sběru naměřených dat o spotřebě. V diagramu jsou červenou barvou znázorněny zprávy, které jsou odeslány ze serveru směrem k zásuvce a modrou barvou jsou znázorněny odpovědi odeslané zpětně zásuvkou. Z diagramu lze vyčíst, že zásuvka se v jednom okamžiku může nacházet pouze ve stavu zapnuto nebo vypnuto. Informace o stavu všech zásuvek je serveru známa. Server odesílá zprávy podle znalosti o stavu příslušné zásuvky. Podle stavu zásuvky tak dochází k rozličným scénářům komunikace. Zpráva „SCAN“ je zvláštní tím, že není adresována žádné konkrétní zásuvce, ale je odeslána na všechny zásuvky pomocí všesměrového vysílání, čímž je zajištěno, že zpráva bude doručena opravdu všem zásuvkám.

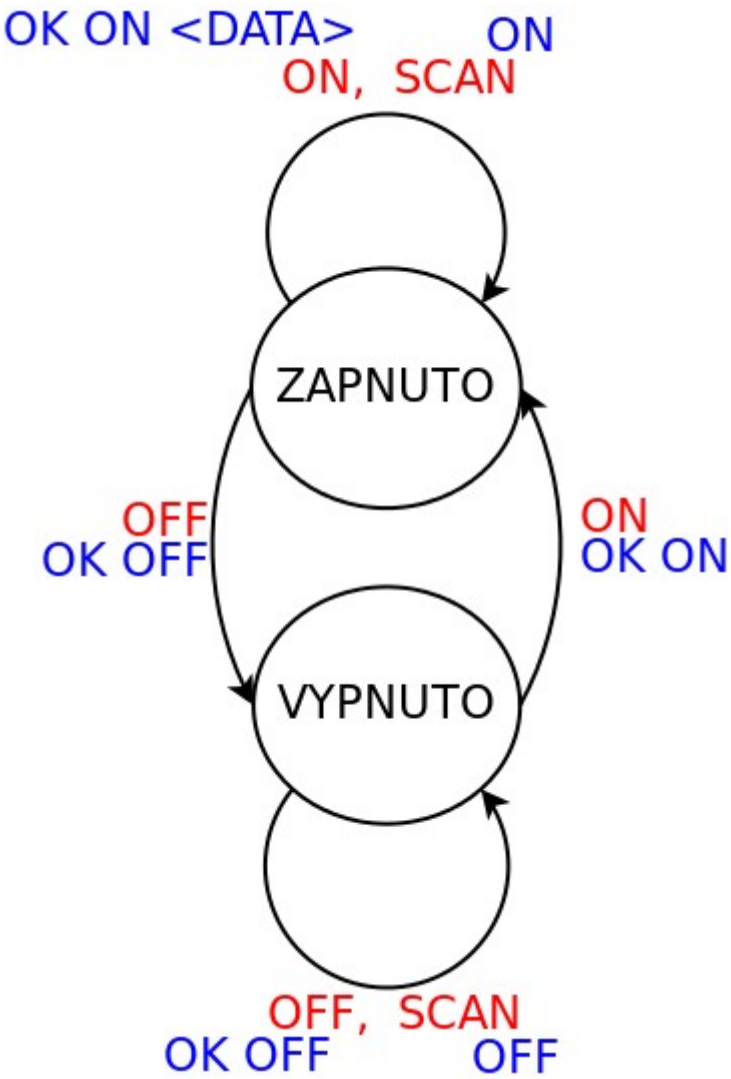
Stav ZAPNUTO:

- Vypuntí zásuvky
  1. Server odešle zásuvce „OFF“ zprávu
  2. Zásuvka změní svůj stav a odešle podtvrzení „OK OFF“
- Detekce nové nebo neaktivní zásuvky a jejího stavu
  1. Server odešle zásuvce „SCAN“ zprávu
  2. Zásuvka odpoví zprávou „ON“
- Periodické vysílání - sběr dat o naměřené spotřebě
  1. Server odešle zásuvce „ON“ zprávu
  2. Zásuvka odpoví zprávou „OK ON <DATA>, kde DATA reprezentují spotřebu v podobě desetimístného čísla

Stav VYPNUTO:

- Zapnutí zásuvky
  1. Server odešle zásuvce „ON“ zprávu
  2. Zásuvka změní svůj stav a odešle podtvrzení „OK ON“

- Detekce nové nebo neaktivní zásuvky a jejího stavu
  1. Server odešle „SCAN“ zprávu
  2. Zásuvka odpoví zprávou „OFF“
- Periodické vysílání – kontrola přítomnosti a funkčnosti zásuvky
  1. Server odešle „OFF“ zprávu
  2. Zásuvka odpoví zprávou „OK OFF“



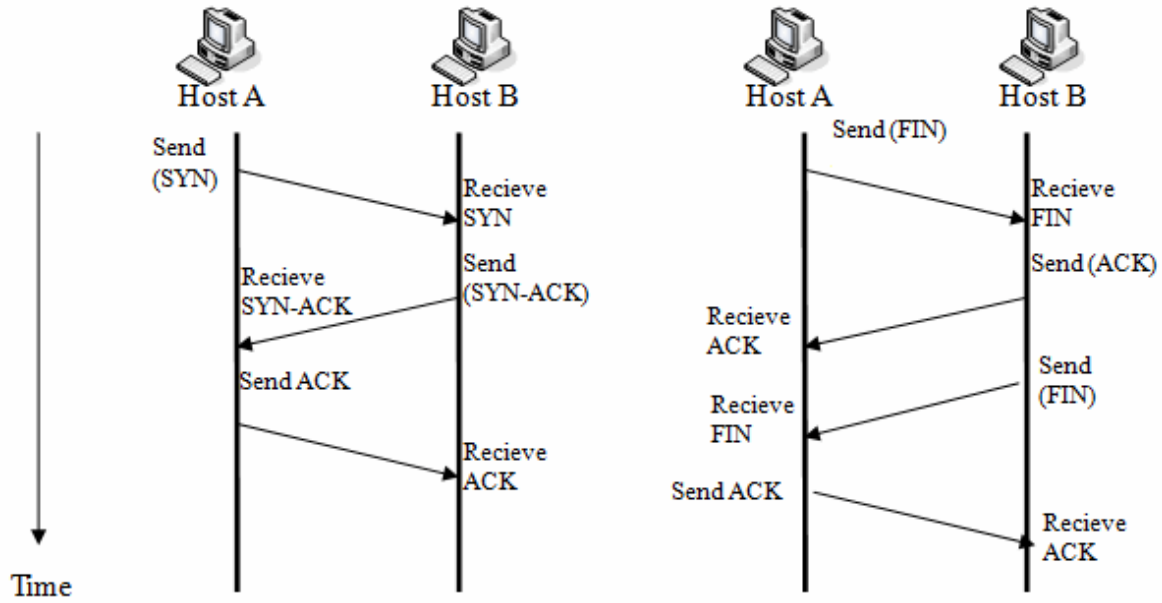
Obr. 4.1: Stavový diagram komunikačního protokolu. Vlastní zdroj

4.3. Volba transportního protokolu

Důležitým rozhodnutím je jaký zvolit pro komunikační protokol charakter komunikce. Zda zvolit spojově orientovaný nebo nespojově orientovaný přenos a zda zvolit spolehlivý nebo nespolehlivý způsob doručení dat. Úkolem transportního protokolu je přizpůsobit požadavky na komunikaci aplikačního protokolu na ovládání zásuvek a měření spotřeby energie protokolům nižších vrstev. V rodině protokolů TCP/IP se nachází dva důležité transportní protokoly TCP a UDP. Jejich základní vlastnosti jsou rozebrány níže.

TCP – Trasmission Control Protocol

"Protokol TCP je spojovanou službou (connection oriented) tj. službou, která mezi dvěma aplikacemi naváže spojení – vytvoří na dobu spojení virtuální okruh. Tento okruh je plně duplexní (data se přenášejí současně na sobě nezávisle oběma směry). Přenášené bajty jsou číslovány. Ztracená nebo poškozená data jsou znovu vyžádána. Integrita přenášených dat je zabezpečena kontrolním součtem"[1]



Obr. 4.2: Způsob navázání (vlevo) a ukončení (vpravo) spojení protokolu TCP. Převzato z [10]

UDP – User Datagram Protocol

"Protokol UDP je jednoduchou alternativou protokolu TCP. Protokol UDP je nespojovaná služba, tj. nenavazuje spojení. Odesílatel odešle UDP datagram příjemci a už se nestará o to, zdali se datagram náhodou neztratil (o to se musí postarat aplikační protokol). Zvláštností protokolu UDP je skutečnost, že adresátem UDP datagramu nemusí být pouze jednoznačná IP adresa."[1]

Použitím transportního protokolu TCP získá komunikace následující vlastnosti:

- Komunikace bude spolehlivá a spojově orientovaná
- Pro každou transakcí mezi hlavním serverem a inteligentní zásuvkou musí být napřed vytvořen virtuální okruh, ve kterém se budou přenášet data. Tento okruh může být trvalý, kdy se okruh vytvoří po inicializaci zásuvky, nebo dočasný, kdy se okruh vytvoří pouze na dobu nutnou pro přenos jedné zprávy a poté se okruh přeruší.

Použitím transportního protokolu UDP získá komunikace následující vlastnosti:

- Komunikace bude nespolehlivá a nespojově orientovaná
- Bude možné odeslat všesměrový oběžník všem zásuvkám
- Každá zpráva bude odeslána ihned bez nutnosti sestavení virtuálního okruhu

Komunikace mezi hlavním serverem a intelignentní musí být:

- Spolehlivá, aby byl zajištěn spolehlivý přenos zpráv pro ovládání zásuvek a nevznikaly tak stavy, kdy by se zásuvka nacházela v jiném stavu, než v jakém je její stav uchován na serveru.
- Jednoduchá, bez zbytečně velké režie.
- Umožnit odeslání všesměrového oběžníku

Použitím transportního protokolu TCP by byla splněna první podmínka na spolehlivost komunikace. Bohužel transportní protokol TCP je v rozporu se zbylými podmínkami. Vytvoření virtuálního okruhu a jeho následné zrušení vyžaduje jistou formu režie. Z obrázku Obr. 4.2. lze vyčíst, že pro sestavení okruhu je nutná výměna 3 TCP segmentů a po výměně samotných dat je ještě nutná výměna 3 TCP segmentů pro ukončení virtuálního okruhu. Transportním protokolem pro komunikaci byl proto vybrán protokol UDP. Spolehlivost musí být poté implementována na samotné aplikační vrstvě.

Implementace mechanismu potvrzování a spolehlivosti

V podkapitole 4.2. byly definovány a popsány druhy zpráv, dále bylo v této kapitole uvedeno, že každá zpráva odeslaná z hlavního serveru na inteligentní zásuvku je zpetně potvrzena potvrzovací zprávou respektive její menší modifikací. Mechanismus potvrzování je již tedy implementován do komunikačního protokolu a zbývá vyřešit otázku spolehlivosti komunikace. Spolehlivost komunikace může být chápána jako snížení pravděpodobnosti rizika ztráty dat při přenosu. V případě použití komunikační technologie, která se vyznačuje nízkou ztrátovostí datových jednotek a nízkou chybovostí, by otázka spolehlivosti nemusela být řešena vůbec, jelikož se předpokládá malá rozlehlost sítě propojující inteligentní zásuvky. Nicméně pro zajištění spolehlivosti bez ohledu na použitou komunikační technologii, je implementován mechanismus, který stanovuje počet pokusů o znovuodeslání zprávy, pakliže není ve stanoveném časovém limitu obrženo potvrzení odeslané zprávy.

4.4. Adresace inteligentních zásuvek

Aby mohl hlavní server odeslat zprávu konkrétní inteligentní zásuvce, musí být každá zásuvka v celém systému inteligentních zásuvek jednoznačně identifikována svou jedinečnou adresou. V rodině protokolů TCP/IP se používají dva různé způsoby adresace zařízení. Na spojové (linkové) vrstvě se jedná o fyzickou adresu zařízení respektive adresu karty síťového rozhraní a je závislá na použité komunikační technologii. Fyzická adresa je také označována jako adresa MAC (Media Access Control) Na vrstvě síťové se jedná o adresu logickou.

"Protokol Ethernet používají šestibajtovou linkovou adresu. Těchto 6 bajtů se dělí na dvě poloviny:

- První polovina je určena pro identifikaci výrobce karty
- Druhá polovina je pak identifikace konkrétní karty v rámci výrobce" [1]

"Protokol IP verze 4 používá IP adresu o délce čtyři bajty. IP adresa adresuje jednoznačně síťové rozhraní systému."[1]

Pro možnost komunikace hlavního serveru s inteligentními zásuvkami bez ohledu na použitou komunikační technologii je zvolen způsob adresace podle protokolu IPv4 a tedy každá inteligentní zásuvka bude mít svou vlastní jedinečnou IP adresu.

Způsob přidělení adresy inteligentním zásuvkám

Přidělit IP adresu lze realizovat dvěma způsoby:

- Staticky – stanici se IP adresa přidělí manuálně
- Dynamicky – stanici se IP adresa přidělí automaticky

Dynamický způsob přidělení IP adresy je zprostředkován protokolem DHCP (Dynamic Host Configuration Protocol). Pomocí protokolu DHCP je přidělena koncové stanici IP adresa z určité množiny IP adres, kterou spravuje. Kromě IP adresy se protokolem DHCP koncové stanici sděluje IP adresa výchozí brány, IP adresy DNS serverů. Použitím správného nastavení DHCP serveru pak je možno inteligentní zásuvce přidělit nejen IP adresu, ale i poskytnout informaci o IP adrese hlavního serveru, který tak může být uveden jako výchozí brána.

4.5. Shrnutí

Inteligentní zásuvky jsou ovládány pomocí textových zpráv, které jsou zabaleny do UDP datagramu, který je zabalen do IP paketu a následně rámce podle použité komunikační technologie. Díky znalostem minimálních velikostí přídavných hlaviček protokolů UDP (8 bytů) a IP (20 bytů) a maximální velikosti zprávy v bytech lze vypočítat maximální velikost přenášeného IP paketu sečtením velikostí dílčích záhlaví a dat. Maximální velikost jednoho IP paketu je pak 44 bytů.

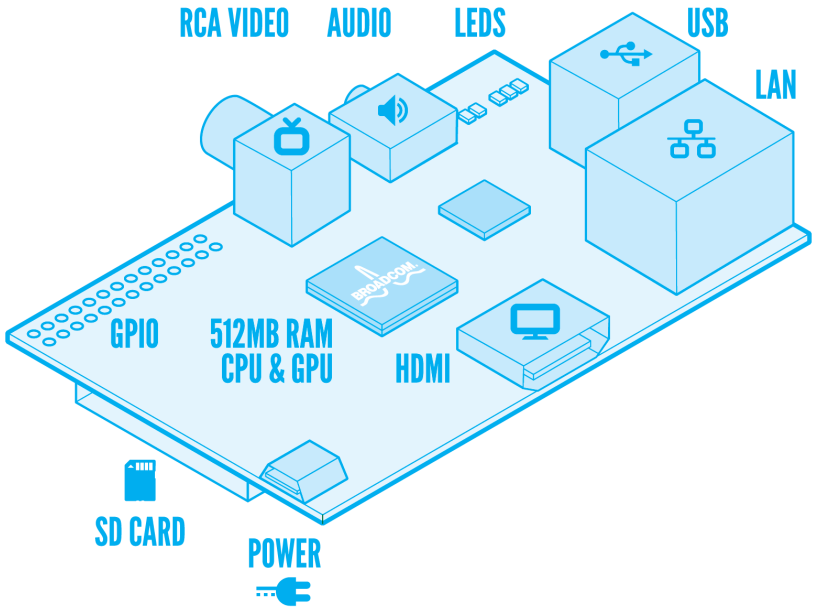
5. Výběr hardware pro hlavní server

Výběr hardwarových prostředků pro hlavní server je dalším dílčím úkolem bakalářské práce. Výhodné je použít již funkčního vestavěného (embeeded) systému. Jedním z příkladů funkčních vestavěných zařízení je miniaturní počítač Raspberry Pi, který byl vybrán jako hardwarový prostředek pro hlavní server. Raspberry Pi je levný a miniaturní počítač, který lze také použít do systémů automatizace a řízení.

Technická specifikace počítače Raspberry Pi model B:

Typ SoC (System On Chip)	Broadcom BCM2835
Procesor	ARM1176JZFS s pracovní frekvencí 700MHz
Grafika	Videocore 4
Video výstup	HDMI, RCA VIDEO
Audio výstup	HDMI, 3,5 mm jack
Velikost operační paměti	512 MB
Rozhraní	GPIO rozhraní (UART, I^2C, SPI) 2x USB rozhraní Síťové rozhraní Ethernet
Rozměry (d x š x v)	85,6 mm x 56 mm x 21 mm
Napájení	5V přes micro USB
Spotřeba	Maximálně 3,5 W

Tab. 5.1: Technická specifikace Raspberry Pi model B. Převzato z [6]



Obr. 5.1: Deska Raspberry Pi model B. Převzato z [6]

6. Výběr softwarových prostředků

Softwarovými prostředky pro realizaci systému inteligentních zásuvek se rozumí:

- Operační systém
- Sada programů pro funkci systému ovládání inteligentních zásuvek

6.1. Operační systém

Na miniaturní počítač Raspberry Pi existuje několik druhů operačních systémů založených především na operačních systémech GNU/Linux.

Raspbian

Raspian je svobodný operační systém založeny na distribuci Debian, která byla optimalizována pro hardware počítače Raspberry Pi. Operační systém je sada základních programů a utilit, které umožní běh Raspberry Pi. Raspbian poskytuje více než jen čistý operační systém. Poskytuje přes 35 000 balíčků, předkompilovaného software svázaného ve formátu pro jednoduchou instalaci na počítač Raspberry Pi.

Pidora

Pidora je Fedora Remix (kombinace software operačního systému Fedora) optimalizována pro počítač Raspberry Pi.

Další operační systémy:

- RISC OS
- ARCH LINUX

Postup instalace operačního systému na SD kartu počítače Raspberry Pi není obsahem této bakalářské práce. Konkrétní postup instalace je možné nalézt na stránkách www.raspberrypi.org. Nutno zmínit, že pro instalaci operačního systému je doporučeno použít NOOBS (New Out of the Box Software) instalátor, pomocí kterého lze jednoduše nainstalovat libovolný operační systém. NOOBS instalátor je možné stáhnout na stránce www.raspberry.org nebo si zakoupit SD kartu s již předinstalovaným NOOBS instalátorem.

Jako operační systém pro počítač Raspberry Pi byl vybrán operační systém Raspbian z následujících důvodů:

- Dobrá znalost operačního systému Debian, na kterém je operační systém založen
- Jednoduchá instalace programů použitím správce balíčků **aptitude**

6.2. Potřebná sada programů

Pro funkci celého systému ovládání inteligentních zásuvek a sběru naměřených dat o spotřebě elektrické energie jsou zapotřebí následující programy:

- Webový server Lighttpd
- Databázový systém MySQL
- DHCP server
- Interpretr programovacího jazyka Python

Webový server Lighttpd

Uživatelské rozhraní je tvořeno webovými stránkami, které umožňují uživateli spustit aplikaci pro ovládání inteligentních zásuvek. Aby bylo možné přistoupit na webové stránky je nutné nainstalovat webový server. Lighttpd je jednoduchý webový server s nízkými požadavky na výkon.

Databázový server MySQL

Informace o aktivních inteligentních zásuvkách, stejně tak jako informace o naměřené spotřebě musí být smysluplně uložena. Pro uložení těchto dat lze použít databázového systému MySQL.

Interpretr programovacího jazyka Python

Vlastní aplikace zastřešující ovládání inteligentních zásuvek a sběr dat o naměřené spotřebě je napsána v programovacím jazyce Python.

„Python je interpretovaný, interaktivní a objektově orientovaný programovací jazyk. Často je srovnáván s Tcl, Perl, Scheme nebo Javou. Python se jednoduše učí a je to mocný programovací jazyk. Má výkonné vysokoúrovňové datové struktury a jednoduchý, přesto mocný, přístup k objektovému programování. Pythonovská elegantní syntaxe a dynamické typování, společně s jeho interpretovanou povahou, ho činí ideálním jazykem pro skriptování a rychlý vývoj aplikací v mnoha oblastech na většině platforem. Pythonovský interpret a rozsáhlá standardní knihovna jsou zcela volně šiřitelné ve zdrojové nebo binární formě na všech hlavních platformách z pythonovské webové stránky Python.org.“[12]

DHCP Server

DHCP server je program, který umožňuje dynamické přidělování IP adres všem stanicím na síti. Program, který je na hlavní server nainstalován, se nazývá **isc-dhcp-server**

7. Realizace programového vybavení

7.1. Simulační program inteligentní zásuvky

V době psaní bakalářské práce ještě nebylo k dispozici příslušné zařízení, které by plnilo funkci inteligentní zásuvky, byl vytvořen program v programovacím jazyce Python, který chování inteligentní zásuvky simuluje. Simulační program simuluje následující funkce inteligentní zásuvky:

- Schopnost komunikovat na určitém UDP portu s hlavním serverem
- Schopnost vyhodnocovat přijaté zprávy od hlavního serveru a příslušně na tyto zprávy reagovat

Příslušnou reakcí na zprávy se rozumí simulace zapnutí respektive vypnutí zásuvky, odeslání dat o naměřené spotřebě a odeslání potvrzení. Stav zásuvky (zapnuto/vypnuto) je v programu uložen v logické proměnné. Stav proměnné se změní v případě přijmutí zprávy na zapnutí nebo vypnutí zásuvky. Data o naměřené spotřebě elektrické energie jsou vygenerována jako náhodné číslo z rozsahu.

Popis programu

Simulační program inteligentní zásuvky využívá pouze jednoho modulu programovacího jazyka Python. Tímto modulem je modul socket, který umožňuje realizovat síťovou komunikaci. Popis modulu včetně jeho implementovaných metod lze nalézt v dokumentaci programovacího jazyka Python.

V simulačním programu je inteligentní zásuvka vytvořena jako objekt třídy Zásuvka, který má své atributy a metody. Důležitým atributem je atribut, který je logického datového typu a obsahuje informaci o stavu zásuvky. Dalším atributem je reference na objekt soketu pro síťovou komunikaci. Třída zásuvky dále definuje čtyři metody (včetně inicializační) pro příjem dat, zpracování přijatých dat a odeslání odpovědi hlavnímu serveru. Níže jsou uvedeny popisy těchto metod.

Inicializační metoda `__init__(self)`:

Konstruktor třídy `Zasuvka`, pro inicializaci důležitých proměnných instance. Definuje a vytváří dohromady dva atributy. Atribut `status` a atribut `sock`. Výchozí hodnota atributu `status` je definována jako `False`, tedy zásuvka je při spuštění ve vypnutém stavu.

Metoda `receivePacket(self)`:

Metoda implementuje metodu objektu `sock` pro přijetí paketu. Návratovou hodnotou metody je typu tuple (n-tice) a obsahuje všechny informace potřebné ke zpracování – Text zprávy, IP adresu hlavního serveru a port, na kterém hlavní server s inteligentní zásuvkou komunikuje.

Metoda `processPacket(self,packet)`:

Metoda představuje rozhodovací logiku pro zpracování zprávy. Zpracování zprávy je realizováno podle stavového diagramu komunikačního protokolu popsaného v kapitole 4.2. a na obrázku Obr. 4.1. Parametr `packet` představuje paket s daty, který byl přijat metodou `receivePacket`. Podle přijaté zprávy jsou provedeny patřičné operace (zapnutí/vypnutí změnou stavu proměnné instance `status`, vygenerování náhodného čísla reprezentující naměřenou spotřebu elektrické energie) a nakonec se vytvoří paket obsahující odpověď, který je předán jako parametr metodě pro odeslání paketu.

Metoda `sendPacket(self,packet)`:

Metoda implementuje metodu objektu `sock` pro odeslání paketu. Parametr `packet` představuje paket připravený k odeslání v metodě processPacket.

Běh programu

Běh programu se skládá:

1. Vytvoření instance třídy `Zasuvka`
2. Opakování metod v pořadí `receivePacket()`, `processPacket()` v nekonečné smyčce pro neustálý běh programu

7.2. Program hlavního serveru

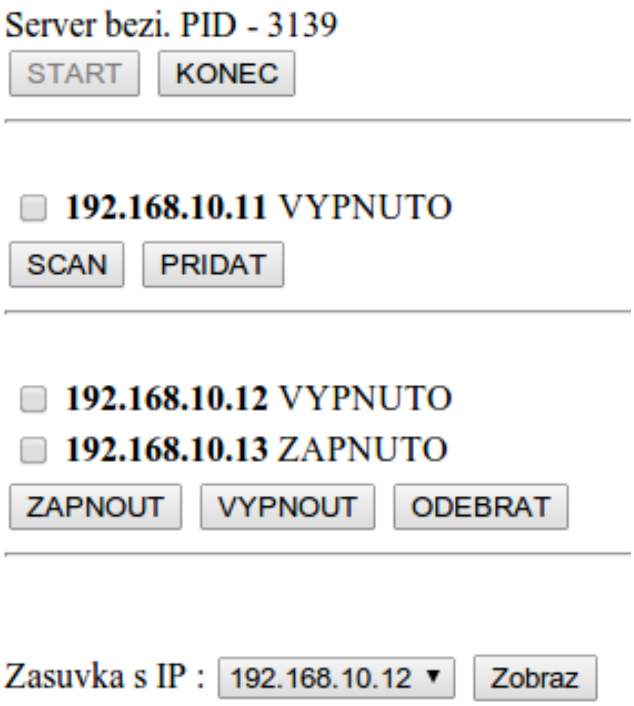
Program hlavního serveru pro ovládání inteligentních zásuvek a sběru dat o naměřené spotřebě elektrické energie se skládá ze dvou částí. Z vlastního ovládacího programu napsaného v programovacím jazyce Python a webového rozhraní, napsaného v programovacím jazyce PHP, pomocí kterého předává uživatel požadavky ovládacímu programu.

7.2.1. Webové rozhraní

Úkolem webového rozhraní je umožnit interakci mezi uživatelem a ovládacím programem inteligentních zásuvek. Webové rozhraní se stavá ze dvou jednoduchých skriptů napsaných v programovacím jazyce PHP a implementuje tyto funkce:

- Spuštění a ukončení ovládacího programu inteligentních zásuvek
- Kontrola spuštěné aplikace a povolení ovládacích tlačítek
- Předání akce uživatele ovládacímu programu a zobrazení reakce
- Přístup k datům o naměřené spotřebě elektrické energie a jejich zobrazení

Vzhled webového rozhraní



Obr. 7.1: Vzhled webového rozhraní. Vlastní zdroj

Na obrázku Obr. 7.1. je zobrazen vzhled webového rozhraní. Rozhraní je rozděleno do 4 částí, s různou funkcionalitou, oddělených horizontální čárou. První část tvoří dvě tlačítka sloužící pro spuštění a vypnutí ovládacího programu. Druhou část tvoří dvě tlačítka a zaškrtávací políčka. Druhá část slouží k detekci inteligentních zásuvek, které nejsou doposud ovládány a nejsou tak uloženy v paměti ovládacího programu. Po stisknutí tlačítka „SCAN“ jsou zobrazena zaškrtávací políčka s popisem obsahujícím IP adresu a stav jednotlivých inteligentních zásuvek. Tlačítko „PRIDAT“ slouží k přidání zásuvek vybraných pomocí zaškrtávacích políček do paměti ovládacího programu. Takto vybrané zásuvky se poté zobrazí ve třetí části rozhraní. Třetí část rozhraní tvoří tři tlačítka. Tlačítka slouží k ovládání stavu zásuvek vybraných v zaškrtávacích políčkách nebo k odebrání zásuvek z paměti ovládacího programu. Poslední část rozhraní je tvořena výběrovým menu a tlačítkem pro zobrazení informací o zásuvce s vybranou IP adresou.

Realizace webového rozhraní

Webové rozhraní je realizováno pomocí dvou PHP skriptů. První skript slouží k vytvoření a zobrazení rozhraní. Druhý pak slouží k odchycení a zpracování stisků tlačítek. Zpracování stisku tlačítka probíhá následovně:

1. Identifikace stisknutého tlačítka
2. Vytvoření zprávy pro ovládací program
3. Odeslání zprávy skrze unixový soket ovládacímu programu

7.2.2. Ovládací program

Celý ovládací program je rozdělen do dvou hlavních spolupracujících modulů.

Modul `Interface`

Úkolem modulu je přijímat zprávy od webového rozhraní popsaného výše a provést příslušné operace. Mezi tyto operace patří zapnutí nebo vypnutí vybrané zásuvky podle její IP adresy. Přidání inteligentní zásuvky do paměti nebo naopak její odebrání z paměti programu podle IP adresy zásuvky.

Modul pro svou vlastní funkci vyžaduje přítomnost několika dalších nativních modulů programovacího jazyka Python, jejichž metody implementuje. Těmito moduly jsou modul `os` poskytující funkce operačního systému, modul `socket` poskytující funkce pro síťovou komunikaci a modul `threading` pro funkce vláknového programování. V modulu samotném je definována třída `Interface`, která definuje své atributy a metody. Těmito atributy jsou reference na objekt unixového soketu, který je použit pro meziprocesovou komunikaci s webovým rozhraním, reference na objekt síťového soketu, který je použit pro komunikaci s inteligentními zásuvkami a reference na objekt realizující přístup k MySQL databázi.

Inicializační metoda `__init__(self,modul):`

Inicializační metoda je zodpovědná za vytvoření všech potřebných zdrojů pro komunikaci, ať už unixový soket pro meziprocesovou komunikaci, soket pro síťovou komunikaci s inteligentními zásuvkami a zavolání inicializační metody vlákna. Dále jsou pak v inicializační metodě načteny informace o zásuvkách, které zůstaly uloženy v databázi po posledním ukončení programu.

Metoda `run(self):`

Metoda, která je zavolána po spuštění vlákna. V této hlavní metodě vlákno čeká na start meziprocesové komunikace od webového rozhraní na unixovém soketu. Přijatá zpráva je pomocí řetezce podmínek rozpoznána a vyhodnocena. Po vyhodnocení přejde vlákno opět do stavu čekání na start meziprocesové komunikace.

Metoda `scan(self):`

Metoda, která je zavolána pro skenování přítomnosti inteligentních zásuvek. Odesláním všesměrového oběžníku je zaručeno oslovení všech přítomných inteligentních zásuvek. V metodě jsou poté postupně zpracovány odpovědi od všech zásuvek a pokud je některá ze zásuvek už uložena v paměti programu je ignorována. Všechny inteligentní zásuvky, které nejsou v paměti jsou zapsány do textového souboru, ze kterého jsou pak načteny webovým rozhraním.

Metoda `turnon(self, adresa, client):`

Metoda, která je zavolána pro zapnutí zásuvky s IP adresou specifikovanou

v parametru `adresa`. Parametr `client` obsahuje referenci na soket pro meziprocesovou komunikaci. Metoda nejprve sestaví paket, který zásuvce odešle, poté vyčká na potvrzení o přijetí paketu pro zapnutí a nakonec signalizuje zpětně webovému rozhraní, zda zapnutí zásuvky bylo úspěšné. Pokud není přijato potvrzení o přijetí paketu pro zapnutí, signalizuje se webovému rozhraní, že zapnutí bylo neúspěšné.

Metoda `turnoff(self, adresa, client)`:

Funkce metody je naprosto stejná jako funkce metody `turnon`. Rozdíl je ve tvaru odeslané zprávy.

Modul `Vysilac`

Úkolem modulu je provádět pravidelný sběr dat o naměřené spotřebě elektrické energie. V pravidelném intervalu je postupně oslovena každá zásuvka nacházející se v paměti programu a podle známého stavu konkrétní zásuvky, je odeslána konkrétní zpráva s žádostí o zaslání odpovědi obsahující data o naměřené spotřebě.

Modul využívá funkcí nativních modulů `socket`, `threading` a `time`. Modul definuje třídu `Vysilac` jejíž atributy jsou pouze reference na objekt soketu pro síťovou komunikaci a reference na objekt pro přístup k databázi.

Metoda `run(self)`:

Metoda, která je zavolána po spuštění vlákna. V této hlavní metodě je postupně z paměti čtena informace o každé zásuvce, podle stavu každé zásuvky se vytvoří paket k odeslání. Po odeslání paketu se čeká na odpověď obsahující informaci o naměřené spotřebě od zapnuté zásuvky, nebo pouze potvrzující informaci o existenci a funknčnosti vypnuté zásuvky. Po oslovení a získání odpovědí od všech zásuvek, je vlákno uspáno po zbytek času, než proběhne další periodický odečet naměřených dat.

Metoda `sendPacket(self, packet)`:

Metoda implementuje metodu objektu soketu pro odeslání zprávy.

Metoda `recvPacket(self)`:

Metoda implementuje metodu objektu soketu pro příjmutí zprávy

Metoda `processPacket(self, packet)`:

Metoda pro zpracování odpovědi od inteligentní zásuvky. V případě odpovědi od zapnuté zásuvky, je do databáze vložena hodnota spotřeby obsažená v odpovědi. V případě vypnuté zásuvky je do databáze vložena hodnota spotřeby nulová.

8. Analýza komunikace

Pro analýzu komunikace mezi hlavním serverem a inteligentními zásuvkami bylo použito protokolového analyzéru a paketového snifferu Wireshark. Simulační program inteligentní zásuvky byl spuštěn na třech virtuálních strojích s operačním systémem Debian. Pomocí metalického ethernetového kabelu kategorie 5E byl přímo propojen hostující počítač s hlavním serverem. Níže jsou uvedeny výsledky analýzy.

8.1. Analýza funkčnosti DHCP serveru

Time	Source	Destination	Protocol	Length	Info
57.810204	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xa05d6315
58.813397	192.168.10.200	192.168.10.15	DHCP	342	DHCP Offer - Transaction ID 0xa05d6315
58.817128	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0xa05d6315
58.832745	192.168.10.200	192.168.10.15	DHCP	342	DHCP ACK - Transaction ID 0xa05d6315
132.192584	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xc6bc0b7f
133.195692	192.168.10.200	192.168.10.11	DHCP	342	DHCP Offer - Transaction ID 0xc6bc0b7f
133.199011	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0xc6bc0b7f
133.215950	192.168.10.200	192.168.10.11	DHCP	342	DHCP ACK - Transaction ID 0xc6bc0b7f
157.778013	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x324e9c1d
158.781187	192.168.10.200	192.168.10.13	DHCP	342	DHCP Offer - Transaction ID 0x324e9c1d
158.786174	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x324e9c1d
158.800822	192.168.10.200	192.168.10.13	DHCP	342	DHCP ACK - Transaction ID 0x324e9c1d

Obr. 8.1: Zachycení komunikace protokolem DHCP. Vlastní zdroj

Na obrázku Obr. 8.1. je zachycena DHCP komunikace mezi virtuálními stroji a hlavním serverem. Komunikace probíhá podle normy DHCP protokolu RFC 2131. Sled zpráv mezi DHCP serverem a DHCP klientem – DHCP Discover, DHCP Offer, DHCP Request a DHCP ACK. Z obrázku lze dále vyčíst jaké jsou použity IP adresy:

- Hlavní server: 192.168.10.200
- Inteligentní zásuvky: 192.168.10.11, 192.168.10.13, 192.168.10.15

8.2. Detekce přítomných zásuvek

Time	Source	Destination	Protocol	Length	Info
10.328153	192.168.10.200	255.255.255.255	UDP	60	Source port: 50001 Destination port: 50000
10.328961	192.168.10.13	192.168.10.200	UDP	44	Source port: 50000 Destination port: 50001
10.329053	192.168.10.15	192.168.10.200	UDP	45	Source port: 50000 Destination port: 50001
10.330973	192.168.10.11	192.168.10.200	UDP	45	Source port: 50000 Destination port: 50001

Obr. 8.2: Průběh komunikace při detekci zásuvek. Vlastní zdroj

Stisknutním tlačítka „SCAN“ ve webovém rozhraní, dojde k oskenování sítě a detekci přítomných zásuvek. Z obrázku Obr. 8.2. je vidět, že z hlavního serveru je odeslán všesměrový oběžník, kterým se zajistí doručení paketu nesoucího zprávu na kterou musí všechny přítomné zásuvky odpovědět informací o svém stavu. Z podrobnějšího výpisu níže lze vyčíst, že zásuvka s IP adresou 192.168.10.13 je ve stavu zapnuto, jelikož text odpovědi je ON, který indikuje zapnutý stav zásuvky. Zbylé dvě zásuvky jsou ve stavu vypnuto a proto je text jejich odpovědi OFF.

```
Internet Protocol Version 4, Src: 192.168.10.200, Dst: 255.255.255.255
User Datagram Protocol, Src Port: 50001, Dst Port: 50000
Data (4 bytes)          SCAN

Internet Protocol Version 4, Src: 192.168.10.13, Dst: 192.168.10.200
User Datagram Protocol, Src Port: 50000, Dst Port: 50001
Data (2 bytes)          ON

Internet Protocol Version 4, Src: 192.168.10.15, Dst: 192.168.10.200
User Datagram Protocol, Src Port: 50000, Dst Port: 50001
Data (3 bytes)          OFF

Internet Protocol Version 4, Src: 192.168.10.11, Dst: 192.168.10.200
User Datagram Protocol, Src Port: 50000, Dst Port: 50001
Data (3 bytes)          OFF
```

8.3. Periodický sběr dat

K periodickému sběru dat dochází v pravidelných intervalech. Pro analýzu komunikace byl tento časový interval zvolen o velikosti 30 sekund. Z obrázku Obr. 8.3. lze vyčíst ve sloupci Time, že časový rozestup mezi dvěma osloveními jedné zásuvky je 30 sekund. Dále si lze všimnout, že je při komunikaci použit jiný UDP port na straně serveru, než jaký je použit při detekování přítomných zásuvek. Důvodem je oddělení řídící komunikace pro detekci a ovládání zásuvek od komunikace pro sběr naměřených dat. Níže jsou podrobněji vypsány obsahy zpráv komunikace. Všechny tyto zprávy odpovídají předpokladům ze stavového diagramu komunikačního protokolu na obrázku Obr. 4.1.

Time	Source	Destination	Protocol	Length	Info
25.020556	192.168.10.200	192.168.10.15	UDP	60	Source port: 50002 Destination port: 50000
25.021548	192.168.10.15	192.168.10.200	UDP	48	Source port: 50000 Destination port: 50002
25.043308	192.168.10.200	192.168.10.13	UDP	60	Source port: 50002 Destination port: 50000
25.044439	192.168.10.13	192.168.10.200	UDP	58	Source port: 50000 Destination port: 50002
25.071909	192.168.10.200	192.168.10.11	UDP	60	Source port: 50002 Destination port: 50000
25.073131	192.168.10.11	192.168.10.200	UDP	48	Source port: 50000 Destination port: 50002
55.048861	192.168.10.200	192.168.10.15	UDP	60	Source port: 50002 Destination port: 50000
55.049836	192.168.10.15	192.168.10.200	UDP	48	Source port: 50000 Destination port: 50002
55.080479	192.168.10.200	192.168.10.13	UDP	60	Source port: 50002 Destination port: 50000
55.081665	192.168.10.13	192.168.10.200	UDP	58	Source port: 50000 Destination port: 50002
55.108864	192.168.10.200	192.168.10.11	UDP	60	Source port: 50002 Destination port: 50000
55.109844	192.168.10.11	192.168.10.200	UDP	48	Source port: 50000 Destination port: 50002

Obr. 8.3: Průběh komunikace při periodickém sběru dat. Vlastní zdroj

```
Internet Protocol Version 4, Src: 192.168.10.200, Dst: 192.168.10.15
User Datagram Protocol, Src Port: 50002, Dst Port: 50000
Data (3 bytes)          OFF

Internet Protocol Version 4, Src: 192.168.10.15, Dst: 192.168.10.200
User Datagram Protocol, Src Port: 50000, Dst Port: 50002
Data (6 bytes)          OK OFF

Internet Protocol Version 4, Src: 192.168.10.200, Dst: 192.168.10.13
User Datagram Protocol, Src Port: 50002, Dst Port: 50000
Data (2 bytes)          ON
```

```
Internet Protocol Version 4, Src: 192.168.10.13, Dst: 192.168.10.200
User Datagram Protocol, Src Port: 50000, Dst Port: 50002
Data (16 bytes)                      OK ON 4944052132

Internet Protocol Version 4, Src: 192.168.10.200, Dst: 192.168.10.11
User Datagram Protocol, Src Port: 50002, Dst Port: 50000
Data (3 bytes)                 OFF

Internet Protocol Version 4, Src: 192.168.10.11, Dst: 192.168.10.200
User Datagram Protocol, Src Port: 50000, Dst Port: 50002
Data (6 bytes)                 OK OFF
```

8.4. Simulace výpadku jedné ze zásuvek

Time	Source	Destination	Protocol	Length	Info
0.000000	192.168.10.200	192.168.10.15	UDP	60	Source port: 50002 Destination port: 50000
0.001330	192.168.10.15	192.168.10.200	UDP	48	Source port: 50000 Destination port: 50002
0.029873	192.168.10.200	192.168.10.13	UDP	60	Source port: 50002 Destination port: 50000
0.031116	192.168.10.13	192.168.10.200	UDP	58	Source port: 50000 Destination port: 50002
0.061997	192.168.10.200	192.168.10.11	UDP	60	Source port: 50002 Destination port: 50000
0.062588	192.168.10.11	192.168.10.200	ICMP	73	Destination unreachable (Port unreachable)
0.563107	192.168.10.200	192.168.10.11	UDP	60	Source port: 50002 Destination port: 50000
0.563649	192.168.10.11	192.168.10.200	ICMP	73	Destination unreachable (Port unreachable)
1.064136	192.168.10.200	192.168.10.11	UDP	60	Source port: 50002 Destination port: 50000
1.064668	192.168.10.11	192.168.10.200	ICMP	73	Destination unreachable (Port unreachable)
30.003905	192.168.10.200	192.168.10.15	UDP	60	Source port: 50002 Destination port: 50000
30.005008	192.168.10.15	192.168.10.200	UDP	48	Source port: 50000 Destination port: 50002
30.605402	192.168.10.200	192.168.10.13	UDP	60	Source port: 50002 Destination port: 50000
30.606492	192.168.10.13	192.168.10.200	UDP	58	Source port: 50000 Destination port: 50002

Obr. 8.4: Průběh komunikace v případě výpadku jedné ze zásuvek. Vlastní zdroj

Na obrázku Obr. 8.4. je zaznamenám průběh periodické komunikace sběru dat o naměřené spotřebě elektrické energie. Před samotnou komunikací, byl ukončen jeden ze simulačních programů inteligentní zásuvky. Z obrázku lze vyčíst, že IP adresa síťového rozhraní virtuálního stroje byla 192.168.10.11. Při odeslání paketu došlo chybě, což je hlavnímu serveru zpětně signalizováno protokolem ICMP, že na cílové stanici s IP adresou 192.168.10.11 není dosažitelný cílový port použitý pro komunikaci s inteligentní zásuvkou. Následují další 2 pokusy o odeslání paketu inteligentní zásuvce. Nejsou-li tyto pokusy o odeslání úspěšné, je informace o inteligentní zásuvce odebrána z paměti ovládacího programu. Odebraná zásuvka není v dalším časovém okamžiku periodického odběru naměřených dat již oslovována.

9. Výběr komunikační technologie pro přenos dat

Pro reálnou aplikaci systému inteligentních zásuvek je zapotřebí použít některou z komunikačních technologií, pomocí které budou přenášena data mezi hlavním serverem a inteligentními zásuvkami. Komunikační technologie lze rozdělit do dvou základních druhů na drátové a bezdrátové. Mezi drátové komunikační technologie patří například technologie Ethernet nebo technologie přenosu dat po drátech silového vedení známá pod zkratkou PLC (Power-Line Communication). Mezi bezdrátové komunikační technologie patří například technologie Wi-Fi nebo technologie Zigbee. Níže jsou stručně popsány vlastnosti těchto komunikačních technologií a jejich výhody a nevýhody při nasazení.

9.1. Technologie Ethernet

„Ethernet je rozsáhlá rodina síťových technologií. Ethernet dříve využíval pro přenos dat sdílené komunikační médium, kterým může být metalický či optický kabel. Ethernet je specifikován normou IEEE 802.3. Ethernet je majoritní technologií při budování lokálních sítí (LAN), ale má velké zastoupení i u větších sítí.

Původně se jako přenosové médium využíval tlustý koaxiální kabel (10Base-5 či slabý koaxiální kabel 10Base-2) v zapojení sdílené sběrnice. Přenosová rychlost byla 10Mb/s. Postupně byl nahrazen kroucenou dvojlinkou (twisted pair) zapojenou do hvězdy (využíval se rozbočovač). Při rychlosti 10Mb/s se označoval pouze ethernet (10Base-T). Poslední médium pro Ethernet je optické vlákno (10Base-F). Postupem času rostly rychlosti na 100Mb/s označované jako Fast Ethernet (třeba 100Base-TX), 1Gb/s označované jako Gigabit Ethernet (třeba 1000Base-T) a dnes máme standard i pro desetigigabitový Ethernet (10GBase-T). Stejně tak byly rozbočovače nahrazeny přepínači, tedy spoji bod-bod, a technologie se označuje přepínaný Ethernet (switched Ethernet), kde se již nevyužívá sdílené médium.

Pro určité typy Ethernetu je vždy definován patřičný kabel. Pro optiku můžeme mít jednovidová či mnohovidová vlákna daných parametrů. Pro metaliku se používá stíněná (STP - Shielded Twisted Pair) či nestíněná (UTP -Unshielded Twisted Pair) kroucená dvojlinka. Navíc je řazena do několika kategorií podle vlastností a možnosti použití pro určitý ethernet, máme například UTP kabel kategorie 3, 4, 5, 5e, 6, 6a či 7.“ [7]

Hlavní výhodou pro použití komunikační technologie Ethernet je přítomnost FastEthernetového rozhraní na minaturním počítači Raspberry Pi. Není tedy nutné investovat další finanční prostředky do hlavního serveru. Nevýhodou je nutnost ke každé zásuvce vést vyhrazený UTP nebo STP kabel, což není přísliš komfortní a levné řešení v případě většího počtu inteligentních zásuvek v domácnosti.

9.2. Technologie PLC

„Již velmi dlouho se rozvíjí technologie pro přenos dat po silových vedeních. Jelikož je infrastruktura silových vedení velmi rozsáhlá, jedná se o velice elegantní řešení. Tato technologie se nazývá Powerline Communication (PLC). Podle používaných kmitočtů

je možné ji rozdělit na širokopásmovou a úzkopásmovou komunikaci. Širokopásmová je podle větší šířky pásma (do 30 MHz) přizpůsobena pro přenos většího objemu dat. V této oblasti však jiné technologie dosahují lepších výsledků. Oproti tomu úzkopásmová komunikace s kmitočtovým pásmem do 150 kHz je využívána především energetickými distribučními společnostmi. Tento druh komunikace není určen pro všeobecné využití, ale je vhodný pro automatizovaný sběr dat jako například dálkové odečítání hodnot z elektroměrů, automatické odečítání dat z čidel nebo pokročilé ovládání čidel. Úzkopásmový systém komunikace by se také uplatnil v oblasti komerčních aplikací jako například ovládání a monitorování solárních panelů, ovládání pouličního osvětlení nebo ovládání výdejních automatů. “ [9]

Technologie přenosu dat po silovém vedení je z výše uvedeného popisu nejvhodnější kandidát z hlediska komfortnosti řešení. K inteligentní zásuvce není nutné zavádět žádné další komunikační médium, jelikož na silové vedení, by byla připojena už při instalaci na silové rozvody elektrické energie. Hlavní nevýhoda nasazení technologie PLC je nákladnost celkového řešení. Jelikož technologie PLC je ještě stále „mladou“ komunikační technologií.

9.3. Technologie Wi-Fi

„Bezdrátové lokální sítě (Wireless Local Area Network) WLAN jsou v současné době na velkém vzestupu. Označují se též Wi-Fi, což je obchodní značka, kterou uvedlo konsorcium výrobců „Wi-Fi Aliance“, aby označovala skutečnost, že jím testované výrobky splňují standardy skupiny IEEE 802.11.“

„WLAN používají jako přenosové médium rádiové vysílání o kmitočtu 2,4 GHz nebo 5 GHz. Avšak na 5 GHz není u nás povoleno. Na provoz WLAN není třeba licence Českého telekomunikačního úřadu – ČTU. Nikdo tak ani nekoordinuje přidělování licencí, a tak se může stát, že budete rušeni i od jiných sítí WLAN. Při nevhodném uspořádání své sitě se dokonce můžeme úspěšně rušit i sami. Dalším zdrojem rušení mohou být zařízení využívající stejného pásma, jako jsou například mikrovlnné trouby, bezdrátové telefony apod. WLAN specifikuje norma IEEE 802.11“ [1]

Protokol	Frekvence	Propustnost	Max. přenosová rychlost
Původní 802.11	2,4 GHz	0,9 Mb/s	2 Mb/s
802.11a	5 GHz	23 Mb/s	54 Mb/s
802.11b	2,4 GHz	4,3 Mb/s	11 Mb/s
802.11g	2,4 GHz	19 Mb/s	54 Mb/s

Tab. 9.1: Přehled standardů IEEE 802.11. Převzato z [1]

Použitím bezdrátové technologie pro přenos dat se by se snížily náklady na vybudování infrastruktury mezi inteligentními zásuvkami a hlavním serverem. Miniaturní počítač Raspberry Pi nedisponuje žádným zabudovaným Wi-Fi modulem, nicméně se může zakoupit Wi-Fi USB dongle, který počítači Raspberry Pi přidá možnost komunikovat bezdrátově pomocí technologie Wi-Fi. Cena tohoto modulu se pohybuje v řádech stovek korun.

9.4. Technologie Zigbee

„Bezdrátový komunikační standard Zigbee, spravovaný organizací ZigBee Alliance a označovaný též jako IEEE 802.15.4, poskytuje cenově nenákladnou, nízkopříkonovou, bezdrátovou komunikaci pro monitorování a řízení systémů. V současné době patří mezi nové, specifikace byla vydána v roce 2004, perspektivní komunikační technologie, která se snaží vyplnit mezeru mezi rozšířenými technologiemi WIFI a Bluetooth. Zde je totiž mezera v podobě velké skupiny aplikací, pro které nejsou Bluetooth ani WIFI, příp. Irda, ideálním řešením, i když se dají použít.

Standard Zigbee lze použít pro jednoduchou bezdrátovou komunikaci s nízkými požadavky na samotný hardware a napájení. Proto jeho hlavní doménou jsou aplikace s bateriovým napájením, kde při výrazně nižší spotřebě energie poskytuje výrazně delší dosah komunikace v porovnání s technologií Bluetooth. To je vykoupeno nižší přenosovou rychlostí, která však v mnoha aplikacích plně postačuje. Například dálkové bezdrátové zapínání/vypínání přístrojů v domácnosti (osvětlení, stahování rolet, odmykání a otvírání dveří) nebo programování a ovládání spotřebičů (televize, DVD rekordér, HIFI systém,klimatizace). Nižší přenosová rychlost poskytuje vyšší odolnost proti rušení, což ZigBee předurčuje pro využití v průmyslu. Zde může zastávat funkci bezdrátové náhrady sériového přenosu RS-232 nebo RS-485. Zde je nevýhodné používat zbytečně složité a drahé WIFI, když přenosová rychlost je jen desítky kb/s. Proti dalším bezdrátovým řešením (např. RF) naopak vyniká topologií sítě, kterou může vytvořit díky propracovanému způsobu adresování. Navíc při bezdrátové komunikaci senzoru s řídícím procesem je opět výhodná nízká spotřeba na straně senzoru, takže může být napájen bateriově a tedy plně oddělen od rušení ve zbytku systému.“ [8]

Z výše popsaných vlastností se přenosová technologie Zigbee jeví jako další vhodný kandidát na technologii pro přenos dat mezi hlavním serverem a inteligentní zásuvkou, jelikož bylo primárně vyvinuto pro ovládací systémy. Miniaturní počítač Raspberry Pi nedisponuje integrovaným Zigbee modulem, který by umožňoval přímé použití této technologie. Počítač Raspberry Pi však může být rozšířen o podporu technologie Zigbee buď formou Zigbee USB dongle nebo použitím Xbee modulu připojeným k rozšiřující kartě Raspberry Pi to Arduino Shield Connection Bridge.

10. Závěr

Cílem bakalářské práce bylo vytvořit návrh a realizovat softwarové prostředky pro inteligentní zásuvku. Tato inteligentní zásuvka disponuje funkcí pro zapnutí nebo vypnutí přívodu elektrické energie a dále disponuje funkcí pro měření spotřeby elektrické energie, kterou ze sítě odebírá elektrický spotřebič, který je připojen k elektrické rozvodné síti prostřednictvím této inteligentní zásuvky.

Hlavní myšlenka návrhu spočívala ve vytvoření cetralizovaného systému pro ovládání inteligentních zásuvek a sběru naměřených dat o spotřebě elektrické energie. V tomto systému vystupuje centrální řídící prvek - hlavní server a zařízení inteligentních zásuvek. Výhodou použití centralizovaného systému je jednoduchost jeho návrhu, kde se veškerá potřebná logika ovládání inteligentních zásuvek implementuje do jednoho uzlu komunikace, který ovládá nejen stav jednotlivých inteligentních zásuvek, ale i řídí veškerou komunikaci. Použití jednoho centrálního řídícího uzlu přináší nevýhodou, že v případě havárie tohoto centrálního řídícího uzlu veškerý systém ovládání inteligentních zásuvek havaruje také a není ani částečně použitelný.

Jako hardware pro zařízení hlavního serveru byl vybrán miniaturní počítač Raspberry Pi model B. Tato volba se ukázala velmi vhodná z důvodu dostatečně výkoného hardware s minimální spotřebou a velmi nízkou cenou. Na Raspberry Pi byl nainstalován operační systém Raspbian spolu s dalšími programy potřebnými pro správný běh celého systému pro ovládání inteligentních zásuvek.

Pro možnost ovládat inteligentní zásuvky byl vytvořen jednoduchý komunikační protokol, který pomocí krátkých textových zpráv sděluje inteligentním zásuvkám požadavky na zapnutí, vypnutí a sběr dat o naměřené spotřebě elektrické energie. Tento komunikační protokol operuje na aplikační vrstvě TCP/IP modelu komunikace a využívá služeb síťového protokolu IP a transportního protokolu UDP. Z důvodu nespolehlivosti přenosu informace pomocí transportního protokolu UDP je v komunikačním protokolu implementován jednoduchý mechanismus potvrzování zpráv a znovuodeslání dat v případě neobdržení informace o potvrzení přijetí zprávy.

Z důvodu absence hardwarového zařízení inteligentní zásuvky byl vytvořen simulační program inteligentní zásuvky, který funkcionalitu zařízení inteligentní zásuvky napodoboval.

Pro hlavní server byl vytvořen ovládací program, který podle navrženého komunikačního protokolu komunikuje s inteligentními zásuvkami a zajišťuje tak jejich zapnutí nebo vypnutí. Tento program také obsahuje přehledné uživatelské rozhraní, které umožňuje uživateli jedoduché ovládání inteligentních zásuvek, spolu se zobrazením informací o spotřebě elektrické energie vybrané zásuvky formou přehledné tabulky.

Jako komunikační technologie pro přenos dat byla zvolena technologie Ethernet. Technologie Ethernet není nejlepší volbou z hlediska praktičnosti a komfortu, ale cena realizace této technologie byla nejmenší ve srovnání s ostatními technologiemi pro přenos dat. Průběh komunikace a test funkčnosti návrhu a realizace systému pro ovládání inteligentních zásuvek byl proveden pomocí programu Wireshark, kterým se sledovala

komunikace mezi hlavním serverem a třemi simulačními aplikacemi inteligentních zásuvek.

V rámci svého dalšího studia, bych rád vytvořil hardwarové zařízení inteligentní zásuvky a vylepšil celkový systém pro ovládání inteligentních zásuvek přidáním dalších funkcionalit jako například časové vypnutí a zapnutí inteligentní zásuvky.

11. Seznam použité literatury

[1] KABELOVÁ, Alena a Libor DOSTÁLEK. *Velký průvodce protokoly TCP/IP a systémem DNS*. 5., aktualiz. vyd. Brno: Computer Press, 2008, 488 s. ISBN 978-80-251-2236-5

[2] KOCOUREK, Petr. *Přenos informace*. 1. vyd. Praha: ČVUT Praha, 2004, 164 s. ISBN 80-010-2892-5.

[3] PILGRIM, Mark. *Ponořme se do Python(u) 3: Dive into Python 3*. 1. vyd. Praha: Cz.Nic, c2010, 430 s. CZ.NIC. ISBN 978-80-904248-2-1

[4] BOHÁČ, Leoš a Pavel BEZPALEC. *Datové sítě: přednášky*. 1. vyd. V Praze: České vysoké učení technické, 2011, 204 s. CZ.NIC. ISBN 978-80-01-04694-4.

[5] VALADE, Janet. *PHP*. 4th ed. Hoboken, NJ: Wiley Pub., Inc., c2010, xviii, 438 p. ISBN 04-705-2758-7

[6] Raspberry Pi: Dokumentace [online]. 2014 [cit. 2014-05-19]. Dostupné z: http://www.raspberrypi.org/documentation/

[7] BOUŠKA, Petr. Ethernet: CSMA/CD, kolizní doména, duplex. BOUŠKA, Petr. *SAMURAJ-CZ* [online]. 2007 [cit. 2014-05-19]. Dostupné z:http://www.samuraj-cz.com/clanek/ethernet-csmacd-kolizni-domena-duplex/

[8] VOJÁČEK, Antonín. ZigBee - novinka na poli bezdrátové komunikace. *HW.cz: Vše o elektronice a programování* [online]. 2005 [cit. 2014-05-19]. Dostupné z: http://www.hw.cz/navrh-obvodu/rozhrani/zigbee-novinka-na-poli-bezdratove-komunikace.html

[9] KOLÁŘ, Jan *Technologie PLC v systémech sběru dat:* bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2013. 54 s. Vedoucí práce byl doc. Ing. Jiří Mišurec, CSc.

[10] Manual:Connection oriented communication (TCP/IP). *MikroTik Wiki* [online]. 1.9.2010 [cit. 2014-05-19].Dostupné z : http://wiki.mikrotik.com/wiki/Manual:Connection_oriented_communication_(TCP/IP)

[11] PŮHONÝ, Jan. *PŮHY.CZ SPOLEHLIVÝ OBCHOD S ELEKTRONIKOU A TECHNIKOU* [online]. 2014 [cit. 2014-05-19]. Dostupné z: http://www.puhy.cz/

[12] KOSINA, Pavel. Python - popis jazyka. *Programujte.com* [online]. 2005 [cit. 2014-05-19]. Dostupné z:http://programujte.com/clanek/1970010106-python-popis-jazyka/

12. Seznam obrázků

13. Seznam tabulek

A Seznam použitých zkratek

GSM – Groupe Spécial Mobile

SMS – Short Message Service

TCP – Transmission Control Protocol

UDP – User Datagram Protocol

IP – Internet Protocol

DHCP – Dynamic Host Configuration Protocol

DNS – Domain Name System

IEEE – Institute of Electrical and Electronics Engineers

USB – Universal Serial Bus

Wi-Fi – Wireless Fidelity

IrDa – Infrared Data Association

LAN – Local Area Network

WLAN – Wireless Local Area Network

ČTU – Český telekomunikační úřad

STP – Shielded Twisted Pair

UTP – Unshielded Twisted Pair

PLC – PowerLine Communication

B Obsah přiloženého CD nosiče

slatinsky_jakub.pdf.....Text bakalářské práce ve formátu PDF

MainServer.....Adresář s programem hlavního serveru

Moduly.....Adresář s vlastními moduly programu

config.py.....Konfigurační modul programu

Interface.py.....Modul rozhraní ovládacího programu

Vysilac.py.....Modul vysilace periodického sběru dat

Model.py.....Modul pro generování SQL dotazů do databáze

tmp.....Adresář obsahující unixový soket

ipc.....Soubor unixového soketu

index.php.....Hlavní stránka webového rozhraní

view.php.....Skript pro zpracování stisků tlačítek

Zasuvka.....Adresář s programem inteligentní zásuvky

Zasuvka.py.....Program inteligentní zásuvky

Interface.svg.....Vývojový diagram modulu rozhraní

Vysilac.svg.....Vývojový diagram modulu vysílače