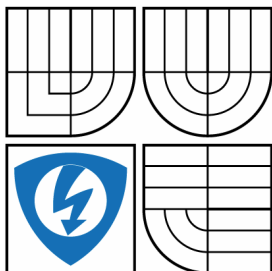


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

INTERNETOVÉ OVLÁDÁNÍ MODELU POMOCÍ PLC A RASPBERRY-PI

REMOTE ACCESS TO CONTROLL SYSTEMS

DIPLOMOVÁ PRÁCE

MASTER THESIS

AUTOR PRÁCE

AUTHOR

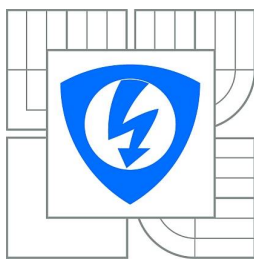
Bc. MATÚŠ KOZOVSKÝ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. ONDŘEJ HYNČICA

BRNO 2015



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Diplomová práce

magisterský navazující studijní obor
Kybernetika, automatizace a měření

Student: Bc. Matuš Kozovský

ID: 136550

Ročník: 2

Akademický rok: 2014/2015

NÁZEV TÉMATU:

Internetové ovládání modelu pomocí PLC a Raspberry-Pi

POKYNY PRO VYPRACOVÁNÍ:

Úkolem práce je realizace vzdáleného přístupu k řídicím systémům za účelem vzdáleného monitorování, správy, konfigurace a programování PLC. Výsledkem by měl být systém, který umožní přes Internet naprogramovat, spustit a monitorovat PLC, které bude řídit laboratorní model.

- 1) Proveďte analýzu možností připojení k vnitřní síti z pohledu funkčnosti a informační bezpečnosti (security), navrhnete vhodné řešení.
- 2) Popište možné způsoby vzdáleného programování a monitorování, navrhnete způsob řešení.
- 3) Navrhnete a realizujete potřebné aplikace pro vzdálené programování (např. platformu Raspberry Pi).
- 4) Prozkoumejte problematiku informační bezpečnosti a navrhnete vyhovující zabezpečení.
- 5) Systém vzdáleného přístupu realizujete a demonstujete na příkladu řízení laboratorního modelu.

DOPORUČENÁ LITERATURA:

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.

Termín zadání: 9.2.2015

Termín odevzdání: 18.5.2015

Vedoucí práce: Ing. Ondřej Hynčica

Konzultanti diplomové práce:

doc. Ing. Václav Jirsík, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Práca sa zaoberá možnosťou vzdialeného programovania priemyselných zariadení PLC alebo HMI, prípadne iných zariadení podobného účelu , ako aj možnosťou vzdialeného monitorovania ich stavov. Práca tiež obsahuje prehľad použiteľných nástrojov pre tento účel, čiže možné nastavenie a hrubý prehľad možností dostupného HW. Taktiež je tu obsiahnutý základný návrh vytváraného systému pre programovanie a monitorovanie, zo všetkými základnými časťami. Práca tiež obsahuje popis realizácie potrebných programov a následne ich testovanie. Ďalej sú výsledky merania zhodnotené a celé riešenie bolo odskúšané na reálnych príkladoch.

Klíčová slova

Vzdialené programovanie, monitorovanie, Raspberry-Pi.

Abstract

This work deals with the possibility of remote programming of industrial devices like PLCs or HMIs and other devices of similar aim, as well as the possibility of remote monitoring of their status. The work also includes an overview of usable tools for this purpose, possible settings and rough overview of possibilities of available HW. There is also a basic proposal of developed system for programming and monitoring, with all essential parts. This work also includes a description of the implementation of the necessary programs subsequently testing them. Furthermore, the measurement results evaluated the whole solution was tested on real examples.

Keywords

remote programming, monitoring, Raspberry-Pi

Bibliografická citace:

KOZOVSKÝ, M. *Internetové ovládanie modelu pomocou PLC a Raspberry-PI*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2014. 64s. Vedoucí diplomové práce byl Ing. Onřej Hynčica

Prohlášení

Prohlašuji, že svou diplomovou práci na téma Internetové ovládání modelu pomocí PLC a Raspberry-Pi jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené semestrální práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: 13. mája 2015

.....
podpis autora

Pod'akovanie

Rád by som sa poďakoval vedúcemu Diplomovej práce pánovi Ing. Ondřeji Hynčici, za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci.

V Brne dňa: 13. mája 2015

.....
podpis autora

OBSAH

1Úvod.....	11
2Využitie vzdialeného programovania a monitorovania.....	12
3Možnosti vzdialeného programovania.....	14
3.1Prepojenie s použitím dostupného nastavenia.....	14
3.2Komunikácia s využitím komerčného HW.....	16
4Informačná bezpečnosť v riešení.....	18
4.1Diffie-Hellmanov protokol.....	19
4.2Využitie kryptosystému AES.....	20
5Návrh riešenia.....	21
5.1Možné spôsoby realizácie.....	21
5.1.1Využitie VPN.....	21
5.1.2Využitie modemov.....	22
5.2HW pre vzdialené programovanie.....	22
5.2.1 Raspberry Pi a jeho možnosti.....	23
5.3 Využitie programu USB-IP.....	24
5.3.1Kompilácia modulov jadra pre USB-IP.....	25
5.3.2Výsledky testovania na lokálnej sieti.....	25
5.4Vytvorenie peer to peer spojenia.....	25
5.4.1NAT traversal.....	26
5.4.2Hole punching (dierovanie).....	27
5.5Vytvorenie serverovej aplikácie pre spojenie v Internete.....	30
5.5.1 Realizovateľnosť pridania serverovej aplikácie.....	30
5.6Štruktúra navrhovanej serverovej časti programu.....	30
5.6.1Synchronizačný modul.....	33
5.6.2Správa užívateľov.....	33
5.6.3Správa zariadení.....	33
5.6.4Modul záložného prepojenia.....	34
5.7Aplikácia pre Raspberry-Pi.....	34
5.8Návrh aplikácie pre užívateľa.....	35
6Realizácia navrhnutého riešenia.....	36
6.1Vytvorenie dynamickej knižnice pre Windows.....	36
6.1.1Implementované režimy pripájania.....	36
6.1.2Testovanie knižnice.....	37
6.2Úprava aplikácie v Raspberry Pi.....	38

6.2.1	Modifikácia časti programu usbipd.....	38
6.2.2	Program správy zariadení.....	38
6.3	Serverová aplikácia.....	39
6.3.1	Modul správy užívateľov.....	39
6.3.2	Modul správy zariadení.....	40
6.3.3	Modul záložného prepojenia.....	40
6.4	Užívateľská aplikácia.....	41
7	Vyhodnotenie navrhnutého riešenia.....	46
7.1	Testovanie USB kľúča.....	46
7.2	Testovanie USB prevodníka na RS232.....	48
7.3	Test prevodníka USB na Ethernet.....	49
7.4	Zhodnotenie.....	51
8	Ukážková komunikácia s Priemyselnými zariadeniami.....	52
8.1	Pripojenie k PLC CP1L cez RS232.....	52
8.2	Pripojenie k PLC WAGO 750-881 cez Ethernet.....	54
8.3	Pripojenie k HMI stanici Weintek MT6050iP cez USB.....	55
8.4	Pripojenie k laboratórnemu modelu vŕtačky.....	56
9	Záver.....	60
10	Zoznam skratiek.....	63
11	Zoznam príloh na CD.....	64

Zoznam obrázkov

Programovanie pri správnom nastavení.....	15
programovanie s využitím prídavného HW.....	16
Jedno-doskový počítač Raspberry Pi[9].....	24
Dierovanie, pred nadviazaním spojenia [8].....	28
Dierovanie, Po nadviazaní spojenia[8].....	29
Dierovanie, proces naväzovania spojenia[8].....	29
štruktúra serverovej aplikácie.....	31
Ukážka registračného formuláru.....	42
formulár pridávania zariadení.....	42
Ukážka zoznamu zariadení.....	43
Ukážka formuláru pre nastavenie zariadenia.....	44
ukážka formuláru pripájania v sieti LAN.....	45
Výsledky testovania rýchlosti USB kľúča.....	47
Rýchlosť prevodníka na RS232.....	49
Rýchlosť prevodníka USB-Ethernet.....	50
Grafické znázornenie zapojenia (Omron PLC).....	52
Ukážka pripojeného prostredia CX-programer ku PLC.....	53
Reálne zapojenie PLC CP1L.....	53
Grafické znázornenie zapojenia (WAGO PLC).....	54
Ukážka aplikácie komunikujúcej s PLC cez Modbus a zobrazujúcej I/O.....	54
Grafické znázornenie pripojenia (Weintek HMI).....	55
komunikácia so vzdialene pripojenou HMI stanicou.....	56
Model laboratórnej vrtačky.....	57
Ukážka aplikácia pre riadenie laboratórneho modelu.....	58

Zoznam tabuliek

Postup výpočtu v Diffie-Hellmanovom protokole[13].....	19
Techniky NAT traversal[3].....	27
Štruktúra tabuliek.....	32
Merané hodnoty rýchlosti USB kľúča.....	47
Meranie prevodníka RS232.....	48
Meranie prevodníka USB-Ethernet.....	50
Adresy protokolu Modbus TCP pre PLC WAGO [19].....	55

1 ÚVOD

Práca sa zaoberá riešením vzdialeného monitorovania alebo programovania systémov PLC, alebo im podobných priemyselných zariadení prostredníctvom siete Internet. Cieľom je vytvorenie cenovo dostupnej a spoľahlivej varianty pre nízko rozpočtové projekty. Pri riešení je snaha o univerzálnosť celého riešenia, aby ho bolo možné využívať v rôznych aplikáciách bez nutnosti ďalších úprav. A zároveň aby bolo možné celkové riešenie rozšíriť o nové zariadenia napríklad pre dataloging alebo inú funkcionality. Snahou je zjednodušiť konfiguráciu na minimum, aby s daným zariadením za pomoci jednoduchého manuálu dokázal pracovať prakticky ktokoľvek.

Všetky navrhnuté prvky sú priemyselne vyrábané s predpokladom dostupnosti najbližších niekoľko rokov.

Pri realizácii je potrebné vytvoriť spojenie medzi vzdialeným PC a riadiacim systémom tak, aby nebol problém s ochranou vnútro podnikovej siete užívateľa riadiaceho systému.

Navrhli sme riešenie klient server s využitím serverového počítača stále pripojeného na sieť a sledujúceho dianie na strane užívateľov aj na strane vzdialených riadiacich systémov.

Zároveň je potrebné riešiť kódovanie dát, aby prenos bol bezpečný z pohľadu zaistenie správnych údajov pri prenose medzi strojom ako aj ochranou voči vonkajším útokom na riadiaci systém.

2 VYUŽITIE VZDIALENÉHO PROGRAMOVANIA A MONITOROVANIA

Väčšina riadiacich systémov je pre užívateľa neprístupná. Pôvodne boli navrhnuté a vytvorené špecializovanou firmou, ktorá sa zaoberá výrobou strojov, či riadiacich systémov. Často krát je firma vzdialená stovky kilometrov od svojho výrobku. Keď však dôjde k poruche je potrebný výjazd servisného technika, ktorého práca je drahá. Na diagnostiku je potrebný špeciálne programové vybavenie, ktorého cena tiež nie je malá. Čas strávený cestovaním tiež možno považovať za stratený, či už z pohľadu človeka, ktorý nevytvára hodnoty, ako aj z pohľadu techniky, ktorá stojí nevyužitá. Preto je potrebné hľadať riešenia, ktoré umožnia okamžite reagovať na vzniknutý problém.

V ideálnom prípade je porucha diagnostikovaná cez vzdialenú správu v spolupráci s miestnym technikom. Keď je známa príčina poruchy, miestny technik už oveľa ľahšie problém odstráni.

Vzdialené programovanie je v dnešnej dobe jedna z vlastností strojov a zariadení, ktorá je zákazníkmi požadovaná. Títo ľudia si uvedomujú aké následky môže mať odstávka stroja v prípade poruchy a zároveň si uvedomujú, že skorá diagnostika a odstránenie poruchy môže minimalizovať dopad na finančné straty, ako aj dopad na chod celého výrobného procesu. Monitorovanie môže predísť stratám alebo dokonca poruchám na strojoch. Predstavte si napríklad situáciu, kedy je na diaľku vykonaná kontrola stavu zariadenia. Pri takejto kontrole sa môže skontrolovať počet zopnutí releových kontaktov alebo počet zopnutí stykačov, ktoré riadia motory, prípadne počet zopnutí ventilov. Pokiaľ sa zistí, že niektoré zariadenia sa blížia k hranici životnosti udávanej výrobcom, alebo už túto hranicu prekročilo, môže byť majiteľ stroja upozornený a v predstihu môže byť naplánovaný servis stroja kedy sa vymenia všetky opotrebované elektronické prvky, ktoré by mohli spôsobiť poruchu na stroji. Takýmto opatrením sa dá predísť poruchám stroja alebo výrobnéj linky s minimálnymi nákladmi. Taktiež je dôležité si uvedomiť, že v prípade poruchy môže dôjsť k poškodeniu výrobkov alebo mechanickému poškodeniu samotnej linky, alebo dokonca k ujme na zdraví. Z tohto dôvodu je v dnešnej dobe dôležitá možnosť vzdialeného monitorovania a programovania strojov.

Samozrejme existujú zariadenia, ktoré sú k takémuto účelu určené, avšak cenovo sú tieto zariadenia často nedostupné hlavne v zariadeniach, pri ktorých sa tlačí na čo najnižšiu cenu. Takýmito zariadeniami môžu byť jednúčelové automaty s pomerne jednoduchou elektroinštaláciou. Z tohto dôvodu by pre projektantov strojov mohlo byť zaujímavé, keby sa k podobnej funkčnosti ako poskytujú komerčné zariadenia dokázali dostať za výrazne nižšiu cenu.

3 MOŽNOSTI VZDIALENÉHO PROGRAMOVANIA

Existuje viacero možností ku vzdialenému programovaniu[1], alebo ku vzdialenej správe zariadení PLC. Tieto zariadenia sa štandardne programujú prostredníctvom siete sériového rozhrania alebo USB rozhrania s využitím dodávaných vývojových prostredí k danému zariadeniu. Vzdialené programovanie môže byť podporované priamo výrobcom a teda SW je na túto funkciu prispôsobený (najčastejšie v prípade ethernetového rozhrania), alebo môžeme vytvoriť transparentné spojenie, ktoré ako už z názvu vyplýva je ako pre SW, tak pre zariadenie neviditeľné. V tejto práci sa zameriame predovšetkým na realizáciu zmieneného transparentného spojenia. V súčasnosti existuje na trhu samozrejme široký sortiment výrobkov, ktoré sú určené práve pre takéto úlohy. Cena týchto zariadení je však v praxi pomerne vysoká a pre jednoduché zariadenia, v ktorých sa tlačí na cenu prakticky nepoužiteľné.

Pri realizácii riadiaceho systému stroja sa veľmi často stretávajú programovateľné zariadenia rôznych výrobcov (PLC, displej, frekvenčné meniče, servopohony) s rôznymi komunikačnými možnosťami. Snahou nášho riešenia je umožniť pripojiť všetky tieto zariadenia k internetu s možnosťou vzdialeného programovania, monitorovania či nastavovania parametrov.

3.1 Prepojenie s použitím dostupného nastavenia

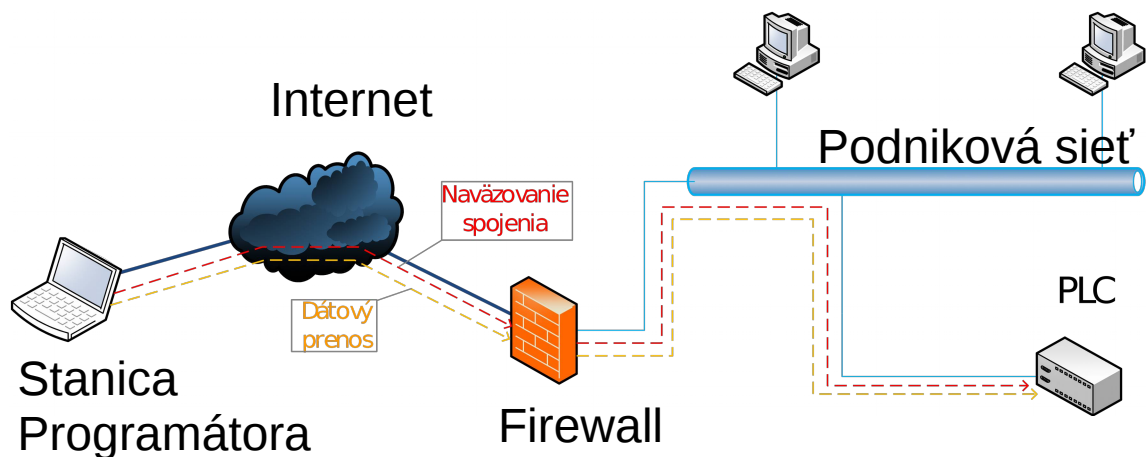
Mnohé PLC zariadenia podporujú možnosť programovania prostredníctvom protokolu TCP/IP. V takom prípade by pre vzdialené programovanie a monitorovanie stačilo správne nakonfigurovať podnikovú sieť, presnejšie smerovanie portov a bránu v zariadení PLC, čo je veľmi jednoduché a samozrejme nevyžaduje prakticky žiadne vynaložené finančné prostriedky.

Tento spôsob má však aj mnohé nevýhody, keďže komunikácia z princípu nie je zabezpečená, a v prípade cieleného útoku by bolo pomerne jednoduché vymazať program v PLC. To by mohlo spôsobiť veľké škody, lebo PLC väčšinou nie je chránené proti nahratiu nového programu, ale iba voči modifikáciám už existujúceho programu,

prípadne jeho skopírovaniu zo zariadenia. Toto opatrenie nájde svoje uplatnenie hlavne v prípade kopírovania strojov a technologických celkov.

Využitie tohto riešenia je predovšetkým vhodné v prípade poruchy na zariadení, kedy je potrebný servisný zásah. V prípade takejto situácie je PLC pripojené do siete a je možné sa k nemu pripojiť a na diaľku vykonať diagnostiku, ktorá často znižuje dobu odstránenia poruchy a taktiež redukuje finančné prostriedky vynaložené na opravu, keďže nie je potrebné platiť výjazd programátora. Po odstránení poruchy je však vhodné PLC znovu odpojiť, aby nedošlo k pripojeniu neoprávnenej osoby. Taktiež pripojenie k sieti môže vyžadovať zásah správcu siete.

Grafické znázornenie komunikácie:



Obrázok 1: Programovanie pri správnom nastavení

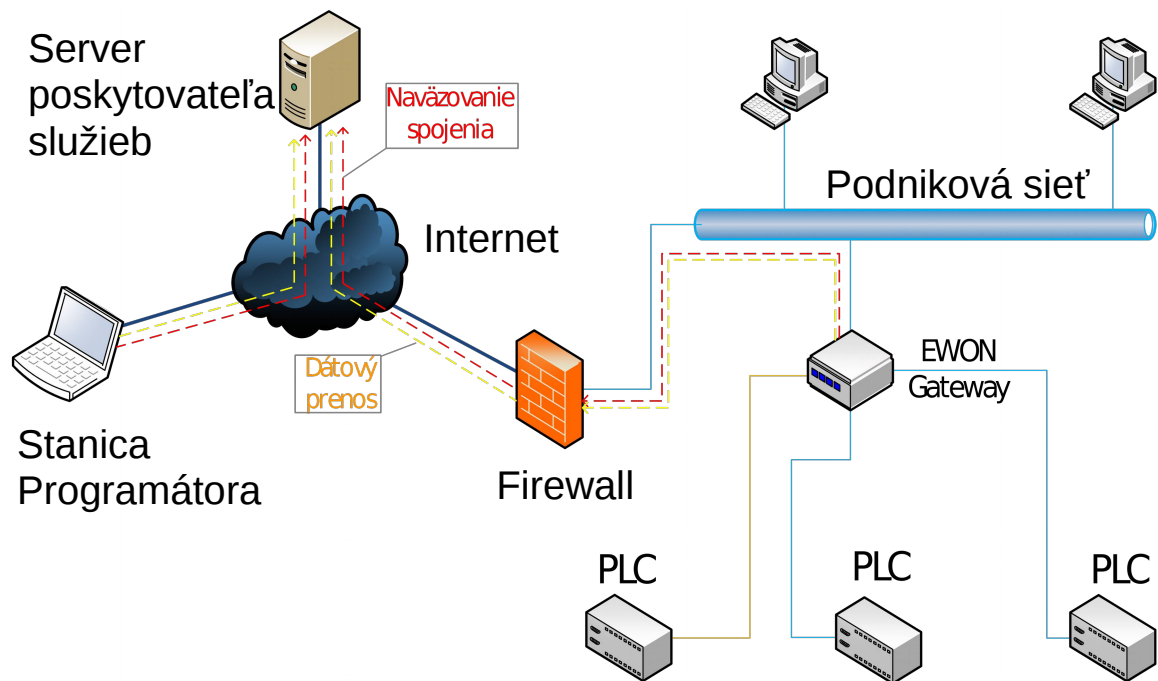
Ako je znázornené, nadväzovanie spojenia ako aj samotný dátový prenos sú inicializované zo strany internetu. Aby toto spojenie bolo možné, musí byť nastavené zmienené smerovanie portov a taktiež je nutné pridať bezpečnostné výnimky v bráne firewall (zníženie bezpečnosti a taktiež nutnosť zásahu správcu siete).

3.2 Komunikácia s využitím komerčného HW

Prostredníctvom bežne dostupného HW môžeme odstrániť nevýhody predchádzajúceho riešenia, akou bolo napríklad zabezpečenie komunikácie. Modul [1] je možné pripojiť do podnikovej siete a následne k jednému alebo viacerým PLC zariadeniam podporujúcich ethernetové rozhranie prípadne RS-232, RS-485, Profibus. Taktiež sú k dispozícii bezdrôtové verzie, ktoré nevyžadujú pevné internetové pripojenie ale bezdrôtové siete (EDGE, 3G)

K nevýhodám patrí ich vysoká obstarávacia cena a tiež prípadné limity pri prenose dát.

Grafické znázornenie komunikácie:



Obrázok 2: programovanie s využitím prídavného HW

Ako je zo znázornenia možné vidieť, komunikácia je vedená na server, čo je výrazne lepšie z pohľadu bezpečnosti, keďže do podnikovej siete neboli vytvorené žiadne bezpečnostné výnimky. Taktiež je výhoda v tom, že pre inštaláciu zariadenia nie je potrebný zásah správcu siete a žiadne prídavné nastavenie. Jediné, čo zariadenie vyžaduje je IP adresa z rozsahu, aby bol možný prístup k internetu. Programátor, ktorý sa snaží vzdialene konfigurovať PLC sa pripája na zabezpečený server patriaci spoločnosti, ktorý zabezpečuje prenos dát medzi programátorom a zariadením spojeným s PLC. Server tiež rieši bezpečnosť a bez patričných hesiel a identifikátorov zariadení nie je možné komunikáciu naviazať. Jednou z firiem, ktorá sa zaoberá práve danou problematikou je spoločnosť Ewon, ktorá ponúka široký sortiment špecializovaných zariadení. Napríklad spomeniem *eWON 2005CD/4005CD priemyselný router*. Ide o zariadenia vybavené jedným WAN portom, 4 LAN portami a sériovým rozhraním. Zariadenie po správnej konfigurácii umožní programátorovi vstup do siete, v ktorej má prístup ku zariadeniam, ktoré sú pripojené do LAN portov daného zariadenia a taktiež má možnosť využívať sériový port (podľa konfigurácie umožňuje využiť rôzne protokoly). Toto zariadenie však neumožňuje pripojenie zariadení s rozhraním USB [1].

4 INFORMAČNÁ BEZPEČNOSŤ V RIEŠENÍ

V dnešnej dobe je veľmi aktuálnou témou informačná bezpečnosť. Keď hovoríme o informačnej bezpečnosti treba brať na vedomie, že žijeme v tzv. informačnom veku, kde dôležitú úlohu zohrávajú práve informácie. Poznáme veľké množstvo možných útokov od počítačových vírusov až po ciele útoky na jednotlivé podniky, inštitúcie alebo štáty. Pokiaľ má byť ľubovoľný systém bezpečný, je dôležité aby bol takto navrhnutý. Pretože prípadné pridávanie bezpečnostných prvkov do realizovaných systémov je buď veľmi nákladné, alebo sa môže stať že takáto zmena nebude možná. Organizácie sa chránia pred útokmi využívaním dostupných HW a SW prostriedkov, ktoré v dnešnej dobe predstavujú firewally a taktiež antivírové programy. Pokiaľ navrhujeme program, ktorý bude komunikovať s internetom, mal by byť navrhnutý tak, aby užívateľom nevzniklo jeho využívaním bezpečnostné riziko. Počas komunikácie s okolitým svetom musíme brať do úvahy, že správy môžu byť dochytené treťou osobou a prípadne zneužitú. Preto by sa v správach nemali za žiadnych okolností objaviť také informácie, ktoré je možné zneužiť. A v prípade, že takéto informácie musia byť prenesené, ich prenos musí byť šifrovaný. Dnešné šifrovanie mechanizmy sa považujú za dostatočne silné pokiaľ sa použije kľúč s minimálnou dĺžkou 112 bitov[15]. V praxi sa používajú 128, 256, prípadne 512bitové kľúče. Šifry sa tiež delia na symetrické a asymetrické. Symetrická šifra používa na dešifrovanie rovnaký kľúč, ktorý bol použitý pre šifrovanie. Sú rýchle a vyvíjané s ohľadom na to, aby ich výpočet nebol priveľmi náročný a zároveň aby boli dostatočne bezpečné. Asymetrické šifry sú zložitejšie na výpočet a majú pomalšiu prenosovú rýchlosť dát. Najčastejšie sa používajú pre prenos kľúčov, ktoré sú ďalej určené pre symetrický kryptosystém. Používajú sa aj pre elektronické podpisy, kde pre použitie je nutná tzv. certifikačná autorita[17], ktorá sa podpíše pod jednotlivé certifikáty. Ideálne je pri každom naviazaní spojenia použiť nové kľúče. Jedným zo spôsobov pre výmenu kľúčov je tzv. Diffie-Hellmanov protokol. Pre zabezpečenie systému vzdialeného programovania je dôležité, aby dáta nemohol získať útočník. Predovšetkým nesmie získať prístupové meno a heslo k účtu užívateľa, pomocou ktorého by mohol uskutočniť útok na zariadenie, ktoré je vzdialene programované. V práci som sa rozhodol využiť symetrický kryptosystém (AES) a prenos kľúčov uskutočniť pomocou Diffie-Hellmanovho protokolu

4.1 Diffie-Hellmanov protokol

Diffie-Hellmanov protokol je kryptografický protokol umožňujúci vytvorenie šifrovaného spojenia medzi komunikačnými stranami prostredníctvom nezabezpečeného kanálu, bez nutnosti predom dohodnutého kľúča. Výsledkom je vytvorenie symetrického šifrovacieho kľúča, ktorý môže byť následne použitý pre šifrovanie ostatnej komunikácie. Výhodou je, že odpočúvajúcí útočník kľúč nezachytí, pretože je skonštruovaný priamo účastníkmi a nie je prítomný v prenosovom kanále. Problémom je útok typu „Man in the middle[16]“, avšak na takýto útok by musela byť narušená bezpečnosť podnikovej siete, kam bude zariadenie umiestnené. Principiálne sa problém opiera o zložitosť výpočtu diskretného logaritmu[13]. Princíp ustanovenia kľúča medzi dvoma účastníkmi je nasledovný:

- Voľba veľkého prvočísla p .
- Nájdenie generátora $q \in \langle 2, p-2 \rangle$.
- Parametre p a q sú verejne známe všetkým užívateľom.
- Strana A náhodne zvolí číslo $a \in \langle 1, p-2 \rangle$ a strane B odošle kryptogram $X1 = q^a \text{ mod } p$.
- Užívateľ B náhodne zvolí číslo $b \in \langle 1, p-2 \rangle$ a Opačnej strane odošle kryptogram $X2 = g^b \text{ mod } p$.
- Užívateľ A vypočíta kľúč $K=X2^b \text{ mod } p$ a užívateľ B vypočíta kľúč $K = X1^y \text{ mod } p$.
- Oba kľúče sú totožné pretože platí :
- $K = X2^a \text{ mod } p = (g^b)^a \text{ mod } p = (g^a)^b \text{ mod } p = X1^b \text{ mod } p = K$

Celý postup je tiež zjednodušene zobrazený v nasledujúcej tabuľke:

Tabuľka 1: Postup výpočtu v Diffie-Hellmanovom protokole[13]

Alica			Bob		Verejnosť
Výpočet	Tajné		Výpočet	Tajné	verejnú
	a			b	p,q
$X1=q^a \text{ mod } p$		->			-> X1
			$X2=q^b \text{ mod } p$		-> X2
$K=X2^a \text{ mod } p$	->K	=	$K=X1^b \text{ mod } p$	->K	

4.2 Využitie kryptosystému AES

Advanced Encryption Standard (AES) je štandardizovaný algoritmus používaný k šifrovaniu dát. Ide o symetrickú blokovú šifru. Táto norma nahradila skôr používanú šifru DES. V praxi sa používa napríklad pre zabezpečenie bezdrôtových WiFi sietí (zabezpečenie WPA2). AES je prvá šifra, ktorá je dostupná širokej verejnosti a ktorá bola zároveň uznaná národnou bezpečnostnou agentúrou NSA ku šifrovaniu najtajnejších dokumentov. Používa sa ako federálny štandard USA. Šifra AES má pevne danú veľkosť bloku 128 bitov a veľkosť kľúča 128, 192 alebo 256 bitov. U pôvodnej Rijndael šifry, z ktorej AES vznikol, bola veľkosť kľúča určená ako ľubovoľný násobok 32 bitov s minimálnou dĺžkou 128 bitov a veľkosť bloku šifrovaných dát bola 256 bitov, zatiaľ čo dĺžka kľúča je prakticky neobmedzená. AES pracuje s maticou bytov 4x4 označovanou ako stav. Priblíženie fungovania šifry je nasledovné:

- Expanzia kľúča – podkľúče sú odvodené z kľúča s využitím Rijndael programu
- Inicializačná časť
 - -pridanie podkľúča – každý byte stavu je skombinovaný pomocou operácie XOR nad všetkými bitmi.
- Iterácia
 - Zámena bajtov- nelineárne nahradenie podľa tabuľky
 - Prehodenie riadkov- každý riadok je posunutý o určitý počet krokov.
 - Kombinovanie stĺpcov - skombinuje 4 byty v každom riadku.
 - Prídanie podkľúča
- Záverečná časť
 - Zámena bytov
 - prehodenie riadkov
 - pridanie podkľúča

Bežne sa v šifre AES využíva 10 iterácií, avšak je možné nastaviť ich ľubovoľný počet[14].

5 NÁVRH RIEŠENIA

Navrhnuté riešenie by malo spájať výhody z oboch zmienených prístupov. Je dôležité, aby vývojové prostredie pre programovanie PLC (prípadne pre nastavovanie iných priemyselných zariadení) nevedelo, že ide o vzdialenú komunikáciu (tzv. transparentné prepojenie). Tento systém využívajú práve komerčné zariadenia, kedy je emulovaná virtuálna sieť, v ktorej sa nachádza stanica programátora ako aj všetky ním programované zariadenia. A taktiež je dôležitým faktorom jednoduchá inštalácia bez výrazného zásahu do podnikovej siete. Pre tento účel je nutné použiť Internetový server, ktorý zabezpečí sprostredkovanie informácií nutných pre pripojenie k zariadeniu. Prenos môže byť realizovaný prostredníctvom serveru, alebo prostredníctvom peer-to-peer prepojenia. Oba spomenuté spôsoby majú svoje výhody a svoje nevýhody, ktoré budú vysvetlené neskôr.

5.1 Možné spôsoby realizácie

Samozrejme, ako na všetky problémy, aj na vyriešenie vzdialeného programovania existuje viacero pohľadov a všeobecne je ťažko povedať, ktoré je najvhodnejšie, pretože každé riešenie je vhodné na riešenie inej aplikácie. Každé riešenie ponúka isté výhody, ktoré však z iného pohľadu môžu byť nevýhodami, a zároveň s niektorými vhodnými vlastnosťami vznikajú aj nežiadúce vlastnosti.

5.1.1 Využitie VPN

Pre naviazanie vzdialeného spojenia je možné využiť virtuálne privátne siete VPN, ako napríklad aplikáciu OpenVPN. Tento program musí bežať na počítači, je možné využiť buď bežný počítač, alebo ho spustiť na jednodoskových počítačoch (napríklad Raspberry-Pi). Následne by stačilo nastaviť na tomto počítači most medzi virtuálnym rozhraním a fyzickým ethernetovým rozhraním. Týmto spôsobom by bolo možné programovať ethernetové PLC zariadenia tak, ako keby sa nachádzali na lokálnej sieti spolu s počítačom programátora.

Dané riešenie má však aj mnohé problémy v prípade, že programátor potrebuje programovať viaceré zariadenia pre rôzne podniky. Tieto by mali byť od seba oddelené, pretože adresácia v rôznych projektoch a rôznych firmách sa môže zhodovať.

Problémom je tiež, že nie je možné programovať PLC ľubovoľného typu, čiže programovanie cez sériový port by bolo značne problematické. Zároveň by bolo potrebné na serveroch udržiavať aktívne všetky virtuálne siete pre všetky zariadenia, čo by bolo veľmi náročné na výpočtový výkon serverov. Taktiež by bolo problematickým riešenie prístupov [6][7].

5.1.2 Využitie modemov

V minulosti bolo možné programovať PLC prostredníctvom klasickej telefónnej káblovej prenosovej linky (rýchlosť 56kb/s). Tieto modemy je však možné využiť iba pre programovanie prostredníctvom sériového rozhrania. Problémom je dostupnosť telefónnej linky so samostatnou klapkou pri každom stroji. Riešením je využitie GSM modemov, ktoré dokážu medzi sebou vytvoriť prenosový kanál, ktorý sa pri vhodnej konfigurácii javí ako transparentný. Riešenie je vhodné, pretože nie sú potrebné žiadne zásahy do siete a použitie je možné všade v dosahu telekomunikačných vysielačov. Žiaľ modemy využívajú pre vzájomné spojenie takzvaný vytáčaný internet. A odozva sa môže vyšplhať až na 5s. Toto riešenie som testoval a práve z dôvodu vysokej odozvy sa ukázalo veľmi nekomfortné a neodporučil by som ho. Avšak riešenie je veľmi lacné a je možné ho použiť napríklad pre vzdialenú diagnostiku v prípade servisu (je vhodné aby diagnostická rutina bola pripravená a preniesol sa iba výstup z diagnostiky). Nevýhodou je potreba aktívnej SIM karty, ktorú je potrebné udržiavať len na občasné spojenie. V prípade, že nie je dlhodobo potrebný servisný zásah, mobilní operátori zrušia príslušnú SIM kartu.

5.2 HW pre vzdialené programovanie.

Pre uskutočnenie vzdialeného programovania je nutné využiť nejaký dostupný HW, ktorý nám umožní spustenie programov a pripojenie potrebných periférií. Existuje rad jedno-doskových počítačov[18], ktoré sa líšia prevedením, výkonom a taktiež perifériami. Je nutné vybrať počítač, ktorý je možné pripojiť k Internetu (disponuje sieťovým rozhraním) a musí mať porty pre pripojenie samotného PLC (USB port, RS232 port prípadne druhý sieťový port). Keďže zariadenie, ktoré by malo všetky tieto požadované porty nie je bežne vyrábané, rozhodol som sa využiť prevodníky (USB-

ethernet, USB-RS232,). Pokiaľ ide o potrebný výkon, nie je potrebný vysoký výkon, lebo programovací modul bude iba preposielať správy a starať sa o ich šifrovanie. Pri porovnávaní rôznych zariadení som sa rozhodol pre jedno-doskový počítač Raspberry-Pi ktorý má požadované porty a je možné ho vložiť do rôznych puzdiar (niektoré púzdra na DIN lištu).

5.2.1 Raspberry Pi a jeho možnosti

Raspberry Pi je jedno-doskový počítač veľkosti približne 9x6 cm. Vývoj zastrešuje Britská nadácia Raspberry Pi Foundation s cieľom podporiť výučbu informatiky na školách. Jeho základom je jednočipový počítač BCM2835 firmy Broadcom, ktorý obsahuje centrálny procesor ARM1176JZF-S s taktom približne 700MHz (s možnosťou pretaktovania až na 1GHz) a grafický procesor VideoCore IV. Operačná pamäť tohto počítača je 256MiB – 1GiB podľa modelu a typu. Namiesto pevného disku sa na uchovávanie dát používa SD karta. Samotný výrobca poskytuje k počítaču ARMové verzie operačných systémov na linuxových distribúciách Debian a Arch. Počítač má štandardné grafické rozhranie HDMI pre pripojenie monitora. Verzia B má k dispozícii v základnej výbave taktiež ethernetový port, pomocou ktorého je možné zariadenie pripojiť k internetu. Taktiež obsahuje USB porty, prostredníctvom ktorých je v dnešnej dobe možné pripojiť väčšinu priemyselných zariadení. A to buď priamo alebo za pomoci prevodníkov. Preto sa toto zariadenie ukazuje ako vhodné pre požadované použitie. Použiteľné sú verzie B/B+ s operačnou pamäťou 512MiB, prípadne je možné využiť verziu Raspberry Pi2, ktorá má výkonnejší procesor a operačnú pamäť 1GiB. Raspberry Pi2 je možno vidieť na nasledovnom obrázku



Obrázok 3: Jedno-doskový počítač Raspberry Pi[9]

Pri vzdialenom programovaní sa bude Raspberry Pi používať ako samostatný modul a nebude ho možné nastavovať prostredníctvom priamo pripojených rozhraní ako je klávesnica, myš a monitor, ale vzdialene, prostredníctvom siete s využitím aplikácie, ktorá bude na tento účel prispôbena. Pri kontrole výkonu sa ukázalo, že Raspberry má dostatočný výkon, aby zvládol potrebnú komunikáciu. Pre funkčnosť nie je napríklad nutné spúšťať grafickú nadstavbu Linuxu (Xserver). Raspberry bude mať za úlohu predovšetkým komunikovať a sprostredkovať komunikáciu. Výpočty budú prebiehať len pri zabezpečovaní komunikácie a na tento účel bude teda dostupný výkon viac než postačujúci. [9][12]

5.3 Využitie programu USB-IP

Po zvážení požiadaviek som sa rozhodol pre využitie programu USB-IP[4] a jeho následnú modifikáciu. Je to program vytvorený pre operačný systém Linux a umožňuje vzdialené zdieľanie USB zariadení. Keďže Raspberry-Pi podporuje práve Linux, je možné tento program využiť. Výhodou je, že pri zdieľaní USB portov bude možné využívať prevodníky ako napríklad USB-Ethernet, a taktiež bude možné priame pripojenie zariadení, ktoré sa programujú prostredníctvom USB rozhrania, ako sú napríklad niektoré typy frekvenčných meničov, servo driver či HMI stanica. Táto vlastnosť umožní nasadenie zariadenia k riadiacim systémom nízkej cenovej kategórie. [4]

5.3.1 Kompilácia modulov jadra pre USB-IP

Bežne dostupné distribúcie pre stolové počítače obsahujú v hlavičkových súboroch kernelu priamo skompilované moduly, ktoré je možné využiť. Avšak na rozdiel od klasických distribúcií, hlavičky kernelu pre Raspberry-Pi neobsahujú tieto moduly. Preto je nutné kompiláciu spraviť manuálne. Pri prvých testoch som použil moduly, ktoré boli pre starší kernel a boli dostupné na internete, no pre nový kernel sa mi tieto moduly nepodarilo nájsť. Pre kompiláciu je potrebné stiahnuť kernel a následne ho skompilovať. Počas kompilácie sa vytvoria aj potrebné moduly, ktoré je následne možné spúšťať a využívať.

5.3.2 Výsledky testovania na lokálnej sieti

Po spoznaní serveru s USB-IP sa stačilo na tento server pripojiť pomocou programu určeného pre operačný systém Windows. Po prvých testoch sa ukázalo, že je možné využívať rôzne typy prevodníkov a prostredníctvom nich sa pripájať priamo k cieľovému PLC (test prebiehal na CP1L-E od firmy Omron). Keďže išlo o PLC programovateľné prostredníctvom siete, bol využitý prevodník USB-Ethernet. Počas testu sa mi podarilo pripojiť sa ku PLC, sledovať stavy a taktiež nahráť nový program do PLC.

5.4 Vytvorenie peer to peer spojenia

Pre správnu funkčnosť v internete je z hľadiska čo najnižšej odozvy potrebné vytvoriť dátové spojenie, ktoré bude naviazané z miesta programovania priamo k programovanému zariadeniu. Týmto spôsobom nevzniknú oneskorenia pri komunikácii na server a ani oneskorenia spôsobené spracovávaním a preposielaním na strane serveru. Avšak dané spojenie nie je možné vytvoriť bez pomoci serveru, pomocou ktorého sa prepošlú potrebné informácie, ktorými sú napríklad IP adresa port a podobne. [5]

5.4.1 NAT traversal

NAT technológia sa bežne používa z dôvodu nedostatku IPv4 adres. Táto technika umožňuje využívanie súkromných IP adres na firemných a taktiež domácich sieťach za smerovačmi, takže všetky počítače na súkromnej sieti vystupujú na internete pod jedinou verejnou IP adresou. Pokiaľ komunikuje zariadenie z vnútornej siete na internet, NAT zabezpečí zmenu zdrojovej adresy (súkromnej) na danú jedinečnú verejnú IP adresu. Zariadenie taktiež môže zmeniť zdrojový port a tým si identifikovať prislúchajúcu súkromnú adresu. Týmto sa však vnútorná sieť stáva nevhodná pre umiestnenie serverov, pretože NAT nemá automatickú metódu ako stanoviť cieľové zariadenie na vnútornej sieti v prípade prichádzajúceho spojenia zo strany internetu. Toto však nepredstavuje problém pre vnútorných užívateľov. Pokiaľ ide o prístup ku webovým serverom, e-mailom a podobne, je bezproblémový, avšak obmedzenia sa vyskytujú v prípade naväzovania peer-to-peer spojenia alebo tiež pri využívaní VoIP služieb. Toto môže predstavovať problém pre užívateľov na vnútornej sieti, pretože

externé požiadavky nemôžu byť jednoducho preposlané na požadované zariadenie. Z tohto dôvodu sú dôležité techniky NAT traversal, pomocou ktorých je možné nadviazať spojenia cez NAT [2][3].

Pre naväzovanie spojenia cez NAT existuje viacero techník. Žiaľ vo všeobecnosti sa nedá povedať, ktorá metóda je najlepšia, pretože každá funguje v inej situácii, keďže správanie sa NAT nie je štandardizované. Mnohé z týchto technológií využívajú server s verejnou IP adresou. Niektoré ho využívajú iba pre nadviazanie spojenia a samotná komunikácia prebieha už bez potreby serveru, iné potrebujú server neustále, čo zvyšuje náklady na šírku pásma a zvyšuje oneskorenie, ktoré je pri real-time komunikácii kľúčové.

Tabuľka 2: Techniky NAT traversal[3]

Socket Secure (SOCKS)	Vznik 90-te roky, využíva proxy server pre spojenie sietí
UPnP IGD	Podporovaná mnohými NAT zariadeniami, predovšetkým domácnosti a malé firmy
Interactive Connectivity Establishment (ICE)	Využívané pre VoIP a P2P komunikácie využíva Session Traversal Utilities for NAT (STUN)
Application-level gateway (ALG)	Súčasť firewallu alebo priamo NATu ktorá dovoľuje nastaviť filtre
Hole punching (dierovanie)	Využíva znalosti o fungovaní NAT a jeho zaobchádzaní s paketmi typu UDP,TCP,ICMP ...

5.4.2 Hole punching (dierovanie)

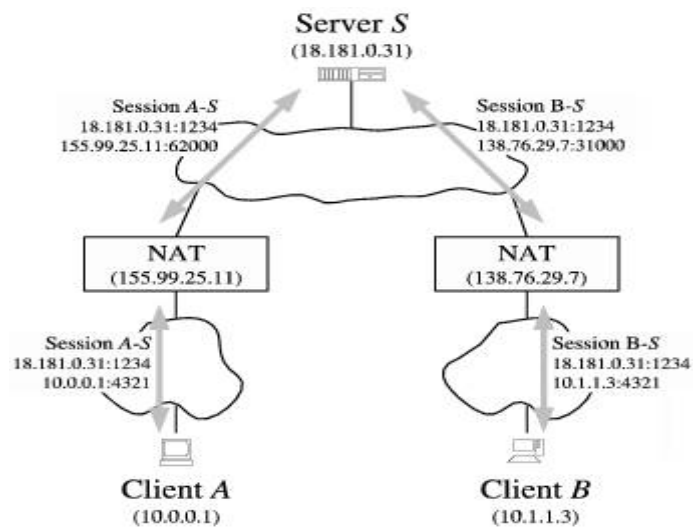
Dierovanie umožňuje dvom klientom vytvoriť priame P2P spojenie protokolom UDP alebo TCP za pomoci známeho „zoznamovacieho“ servera. Technológia funguje aj v prípade, že sú obaja klienti umiestení na NAT. Táto technika bola uvedená v bode 5.1 RFC 3027. Princíp sa tiež používa v niektorých experimentálnych internetových protokoloch. Rôzne proprietárne protokoly, ktoré sú využívané pre hranie online hier tiež používajú technológiu dierovania.

Dierovanie predpokladá, že dvaja klienti, **A** a **B**, už majú aktívne spojenia so „zoznamovacím“ serverom **S**. Keď sa klienti pripájajú k serveru **S**, server si zaznamená koncové body oboch klientov (zdrojová IP adresa, zdrojový komunikačný port), Tieto

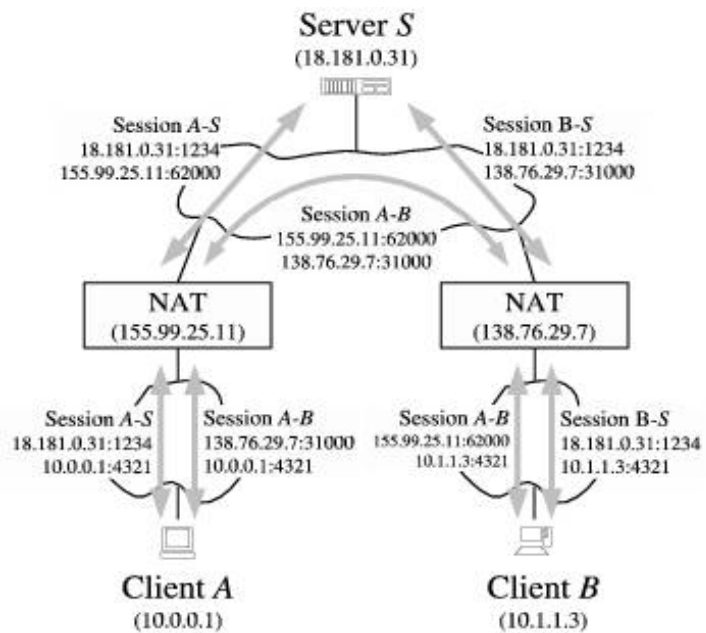
údaje server využíva na komunikáciu s klientmi. Klient taktiež v tele správy pošle informácie o svojom vlastnom koncovom bode. V prípade, že správa prechádza cez NAT, zdrojová IP adresa a port sú zmenené, o čom samotný klient nevie, ale server tieto informácie získava s hlavičky správy. V prípade, že sa klient nenachádza za NAT, informácie získané serverom z hlavičky správy a informácie poslané klientom v tele správy sa zhodujú a server dokáže jednoznačne identifikovať, že klient sa nenachádza za NAT. Dajme tomu, že klient **A** chce nadviazať spojenie priamo s klientom **B**. Dierovanie potom prebieha nasledovne:

- Klient **A** použije svoje aktívne spojenie so serverom **S** a požiada server o pomoc pri pripájaní sa na klienta **B**
- Server **S** odpovie klientovi **A** a poskytne mu informácie o verejnom a privátnom koncovom bode klienta **B**. V rovnakom čase pošle klientovi **B** zhodné informácie o klientovi **A**.
- klienti **A** a **B** začnú synchronne vytvárať spojenia na koncové body, ktoré v predchádzajúcom kroku obdržali od serveru **S**. Dôležité je, aby obaja klienti použili rovnaký lokálny port, ktorý použili na komunikáciu so serverom. Popri nadväzovaní spojenia klienti taktiež aktívne načúvajú na svojich lokálnych portoch, ktoré odpovedajú ich zdrojovým portom pre komunikáciu.
- Obaja klienti **A** aj **B** čakajú, či sa im podarí pripojiť, alebo či sa neobjaví prichádzajúce spojenie. Ak jeden z odchádzajúcich pokusov o pripojenie zlyhá z dôvodu chyby siete, ako je „reset pripojenie“ alebo „Host nedosiahnuteľný“, klient sa snaží vytvoriť nové spojenie a to až do okamihu, kedy nie je prekročená maximálna doba stanovená pre pripojenie.
- Po nadviazaní spojenia sa obaja klienti navzájom overia, či prebehlo pripojenie k požadovanému klientovi. Ak overenie zlyhá, klienti ukončia toto pripojenie a naďalej čakajú na úspech.

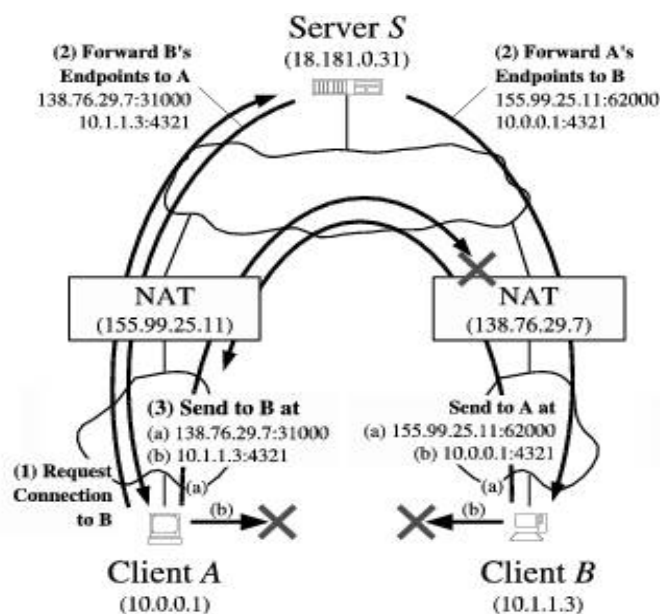
Slovne popísaný postup je možné vidieť graficky na nasledovných obrázkoch:



Obrázok 4: Dierovanie, pred nadviazaním spojenia [8]



Obrázok 5: Dierovanie, Po nadviazaní spojenia[8]



Obrázok 6: Dierovanie, proces naväzovania spojenia[8]

Táto technológia využíva vlastnosť NAT zariadenia, kedy sa určitá kombinácia privátneho koncového bodu preloží vždy rovnako na určitý verejný koncoví bod. Taktiež pri nadväzovaní spojenia NAT vytvorí dočasný záznam o smerovaní portov pre dané cieľové zariadenie. [5][8]

5.5 Vytvorenie serverovej aplikácie pre spojenie v Internete

Keďže systém vzdialeného programovania a monitorovania musí dokázať obslúžiť zariadenia, ktoré sú na rôznych miestach a ich internetové adresy sa môžu stále meniť, je nutné mať server s nemennými adresami, ktorý sa bude starať o správu zariadení, na tento účel bude vyvinutá serverová aplikácia.

Pre komunikáciu prostredníctvom servera je nutné upraviť aplikáciu USB-IP pre Linux, taktiež aj pre Windows tak, aby sa obe pripájali na server. Serverová aplikácia by mala dokázať obsluhovať jednotlivých užívateľov a zariadenia tak, aby každý užívateľ mohol pristupovať len k zariadeniu, ku ktorému má oprávnenie.

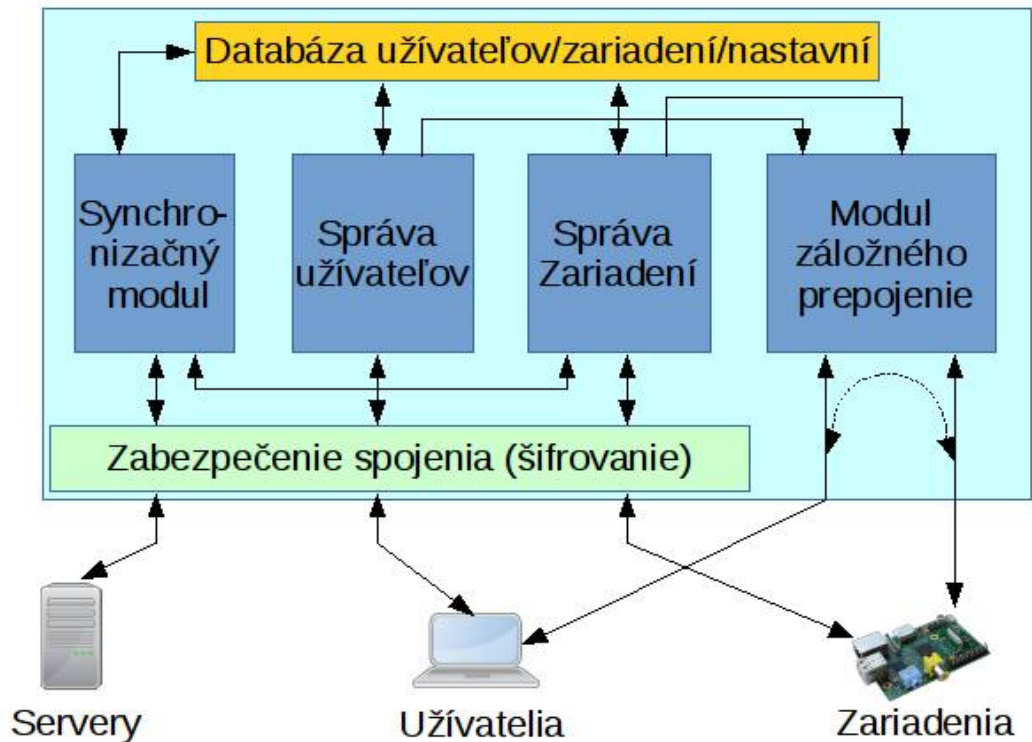
5.5.1 Realizovateľnosť prídania serverovej aplikácie

Z dôvodu vyskúšania navrhnutého riešenia som realizoval jednoduchú serverovú aplikáciu, ktorá dokáže preposielať správy medzi dvomi klientmi. Pre Windows som v prvom kroku vytvoril dynamickú knižnicu, pomocou ktorej je možné pripojiť sa k cieľovému zariadeniu a zakomponoval som ju do programu, ktorý sa pripája na server. Na server odošle žiadosť o pripojenie, ktorá obsahuje unikátne ID zariadenia, ku ktorému sa chce pripojiť. Server odpovie podľa toho, či je k nemu zariadenie s daným ID pripojené alebo nie. Ak je odpoveď kladná, program volá dynamickú knižnicu, ktorá sa stará o pripojenie na server, ako to bolo v prípade testovania na lokálnej sieti. Server ďalej všetku komunikáciu preposiela na cieľové zariadenie. V prípade, že je takto vytvorené spojenie, komunikácia prebieha bez toho, aby zariadenia vedeli, že komunikujú cez server.

Spomenuté riešenie vytvorilo funkčný prenos. V praxi bude nutné knižnicu ako aj program rozšíriť o prihlásenie a zabezpečenie, ktoré pri pokuse samozrejme implementované nebolo. Spojenie prebiehalo kompletne cez server, takže nešlo o rýchle P2P spojenie ale o pomalšie spojenie s využitím systému preposielania správ serverom. Test ukázal že je možné pokračovať týmto smerom a vytvoriť spojenie cez server.

5.6 Štruktúra navrhovanej serverovej časti programu

Serverová aplikácia je samozrejme najdôležitejšia časť celého systému. Servery sú nevyhnutné pre nadviazanie spojenia a taktiež pre správu užívateľov a priradovanie práv pre prihlasovanie k jednotlivým zariadeniam. Návrh štruktúry danej aplikácie môžeme vidieť na nasledovnom obrázku.



Obrázok 7: štruktúra serverovej aplikácie

Na strane serveru je využitá SQL databáza s niekoľkými tabuľkami,

- **Tabuľka užívateľov**, obsahuje informácie o užívateľoch, ich prihlasovacie mená a heslá (presnejšie heš hesiel) a základné informácie vyplnené pri registrácii.
- **Tabuľka užívateľských zariadení**, tabuľka obsahuje jednotlivé zariadenia a ich vlastníka a nastavené prístupové heslo k danému zariadeniu. Jedno zariadenie môže mať viacerých vlastníkov, pokiaľ užívateľ vlastní zariadenie má právo sa k nemu pripojiť
- **Tabuľka zariadení**, táto tabuľka obsahuje informácie o zariadeniach, ktoré majú oprávnenie využívať server na základe ich výrobného čísla. Zariadenia budú do databázy pridané pri prvom spustení, prostredníctvom inicializačnej rutiny. Po pridaní zariadenia je možné zmeniť jeho aktuálne heslo vyžadované pre prácu s daným zariadením. Týmto opatrením sa zabráni prípadnému kopírovaniu SW

- **Tabuľka nastavení serverov**, slúži pre ukladanie informácií o ostatných serveroch, slúži pre správu verzií a taktiež pre synchronizáciu medzi jednotlivými servermi, ktoré dokážu spolu kooperovať

Pre realizáciu tabuliek sa ukazuje ako vhodné riešenie použiť už predpripravené riešenie v podobe SQL serveru. V praxi je možné použiť niektorý z platených SQL serverov, ako napríklad Microsoft SQL server. Pre testovanie je možné použiť open source riešenia ktorým je napríklad MariaDB [10][11].

Tabuľka 3: Štruktúra tabuliek

Názov tabuľky	Názov premennej (význam)	Typ remennej
Užívatelia	KEY (ID záznamu)	BIGINT
	Užívateľ (prihlasovacie meno)	VARCHAR [50]
	Heslo (používateľa pre prihlásenie)	VARCHAR [32]
	Mail (používateľa)	VARCHAR [50]
	Synchronizácia	SMALLINT
Zariadenia	KEY (ID záznamu)	BIGINT
	Vlastník (ID s tabuľky užívateľov)	BIGINT
	Zariadenie (ID procesora pridaného Raspberry)	BIGINT
	Typ (pre možné rozšírenie serveru)	INT
	Heslo (ku zariadeniu)	VARCHAR [32]
	Synchronizácia	SMALLINT
povolenia	KEY (ID záznamu)	BIGINT
	Zariadenie (ID procesora povoleného Raspberry)	BIGINT
Server	KEY (ID záznamu)	BIGINT
	IP	VARCHAR [50]

Okrem databázy obsahuje server samotnú aplikáciu, ktorú je možné rozdeliť do 4 základných častí.

5.6.1 Synchronizačný modul

Táto časť serveru sa stará výhradne o synchronizáciu databáz medzi jednotlivými servermi. Po pridaní užívateľa rozpošle informácie všetkým ostatným serverom, aby si daného užívateľa pridali do databázy. Synchronizuje tabuľku zariadení, ktoré majú oprávnenie využívať server. Zo štruktúry si tiež môžeme všimnúť, že daný modul komunikuje s modulom správy zariadení. Keďže jednotlivé zariadenia sa budú

pripájať na rôzne servery rovnako ako užívatelia, môže sa stať, že užívateľ sa pripojí na iný server ako je pripojené zariadenie, s ktorým hodlá naviazať komunikáciu. Z tohto dôvodu by mali mať servery prehľad o zariadeniach, ktoré sú v danom čase on-line a na ktorom servery sú pripojené. Vďaka tomuto môžu všetky servery vedieť o všetkých zariadeniach.

5.6.2 Správa užívateľov

Daný modul sa stará o autentifikáciu užívateľov, ktorí sa prihlasujú k serverom a využívajú ich. Každý užívateľ má v databáze vlastný záznam s heslom, ktoré pri prihlásení musí zadať. Modul sa tiež stará o registráciu nových užívateľov, ako aj o proces registrácie zariadení. Ku každému užívateľovi sú na základe jeho unikátneho ID priradené zariadenia, ktoré sú chránené samostatnými heslami. Pokiaľ si užívateľ napríklad pridá zariadenie, ale nezadá k nemu správne heslo, nebude mu umožnené sa na dané zariadenie pripojiť. V prípade žiadosti o pripojenie a súčasnej zhody hesiel, pošle užívateľovi informácie potrebné pre naviazanie spojenia.

5.6.3 Správa zariadení

Táto časť serveru sa stará o programovacie zariadenia, ktoré sú on-line. Každé zariadenie sa prihlási na server a čaká, svoj stav spojenia ohlasuje v pravidelných intervaloch a tieto informácie sú distribuované medzi všetky servery ako bolo zmienené. V prípade požiadavky o pripojenie na dané zariadenie, modul vyšle do zariadenia informácie potrebné pre naviazanie peer-to-peer spojenia priamo s užívateľom, ktorý o spojenie požiadal. Modul taktiež rozlišuje viaceré zariadenia podľa typu. V súčasnosti je vyvíjané iba jedno zariadenie, ale modul je pripravený pre budúcnosť a je postavený tak, aby implantácia nového zariadenia nevyžadovala zásah do programu ani do databázy.

5.6.4 Modul záložného prepojenia

Tento modul je obsiahnutý z dôvodu spoľahlivosti. Ako bolo už skôr v teórii spomenuté, neexistuje jednoznačný postup ako vytvoriť P2P spojenie medzi dvomi koncovými zariadeniami na internete. Je možné, že pomocou žiadneho z implementovaných postupov sa nepodarí nastaviť dané spojenie. Toto sa môže stať pokiaľ bude sieť užívateľa rovnako ako sieť zariadenia extrémne zabezpečená voči

komunikáciám zo strany internetu. V takomto prípade dostane modul informácie od modulu zariadení a od modulu užívateľov, aké spojenie má sprostredkovať a moduly taktiež zašlú informácie užívateľskej aplikácií ako aj zariadeniu a následne je vytvorené spojenie s využitím prenosovej kapacity serveru.

5.7 Aplikácia pre Raspberry-Pi

Na jedno-doskovom počítači Raspberry-Pi beží operačný systém Rasbian ktorý je postavený na Linuxovom jadre. Ako už bolo skôr spomenuté na Raspberry-Pi nebude bežať grafické rozšírenie Xserver takže ide o konzolovú aplikáciu.

Aplikácia vyvíjaná pre Raspberry-Pi bude mať dva hlavné moduly. Jeden bude slúžiť na komunikáciu so serverom. Keďže každé zariadenie bude v stave on-line pokiaľ samozrejme bude mať prístup k internetu a bude zapnuté, po pripojení na server by mala prebehnúť kontrola nastavení. Nastaveniami sa myslí, či program obsahuje databázu všetkých dostupných serverov a ich adries. Toto opatrenie je dôležité v prípade, že sa bude systém rozširovať, aby bolo možné vzdialene modifikovať nastavenia vo všetkých zariadeniach. Nastavenie bude tiež obsahovať časové údaje ako často sa má hlásiť zariadenie na server, aby bolo možné sledovať jeho stav, prípadne iné údaje, ktoré sa počas testovania ukážu ako nevyhnutné. V prípade že sa tieto nastavenia ukážu ako zastaralé, priamo v procese pripájania by mala prebehnúť ich aktualizácia.

Druhou dôležitou časťou bude modul, ktorý zabezpečí naviazanie P2P spojenia v prípade, že to nastavenie a HW prvky siete umožnia (ide o nastavenie smerovania portov cez UPnP protokol, prípadne iné z techník NAT Traversal). Pokiaľ sa toto spojenie nepodarí vytvoriť, využije sa Serverový modul pre záložné pripojenie s preposielaním. Druhou časťou programu je samotný modul umožňujúci zdieľanie USB portov. Po úspešnom prepojení užívateľskej aplikácie a príslušného Raspberry-Pi, by mal mať užívateľ plný prístup ku USB zariadeniu, ktoré si vyberie a ku ktorému sa pripojí. Výber bude samozrejme možný zo zariadení, ktoré sú pripojené do USB portov daného Raspberry-Pi.

5.8 Návrh aplikácie pre užívateľa

Užívateľská aplikácia má v prvom rade za úlohu naviazať spojenie s cieľovým zariadením a tým vytvoriť cestu pre programátora k cieľovému zariadeniu (PLC, HMI,

Frekvenčný menič, Servo drive, ...). Na tento účel slúži komunikačný modul so serverom, ako bolo popísané už skôr v tejto práci. Tento program musí mať podobný modul pre naväzovanie P2P spojenia pokiaľ je to možné, inak treba využiť záložnú možnosť spojenia. V druhom rade musí program užívateľovi poskytnúť komfort a jednoduché intuitívne ovládanie. Užívateľ by mal mať prehľad o svojich zariadeniach. Mal by vedieť, ktoré zariadenia sú online a ktoré sú naopak nedostupné a tiež by mal mať prehľad o USB zariadeniach, ktoré sú k modulom Raspberry Pi pripojené. Dôležité je, aby aplikácia zabezpečovala bezpečnosť komunikácie so serverom a aj bezpečnosť komunikácie so samotnými koncovými zariadeniami.

6 REALIZÁCIA NAVRHNUTÉHO RIEŠENIA

Ako už z predošlej kapitoly vyplýva, pre úspešnú realizáciu je potrebné vytvoriť a postupne overiť funkčnosť viacerých samostatných programov, ktoré budú vo výsledku prepojené do jedného celku. Ide o modifikáciu existujúcich programov USB-IP a zakomponovanie bezpečnosti do týchto modulov a aj o vytvorenie užívateľského prostredia a ďalších programov využívajúcich tieto modifikované programy.

6.1 Vytvorenie dynamickej knižnice pre Windows

V prvom kroku boli upravené dostupné zdrojové kódy z pôvodnej klientskej aplikácie pre Windows. Táto aplikácia pôvodne mala nasledovné funkcie:

- vypísanie zoznamu exportovaných USB zariadení z určitého vzdialeného Raspberry Pi (identifikovaného pomocou adresy IP), ku ktorým je možné sa pripojiť
- Pripojenie jedného z USB zariadení exportovaných vzdialeným Raspberry Pi (pre pripojenie nutné zadať IP a identifikátor USB portu).
- Odpojenie externého USB zariadenia v prípade že bolo toto zariadenie už pripojené

Tieto zdrojové kódy boli upravené do podoby dynamickej knižnice (DLL), ktorá z vyššie spomenutých funkcionalít podporuje iba pripájanie a odpájanie vzdialeného USB zariadenia. Vyhľadávanie dostupných exportovaných zariadení bolo presunuté priamo do aplikácie užívateľského prostredia. Tento krok bol uskutočnený z dôvodu jednoduchšej implementácie serverovej aplikácie. Zároveň bol upravený spôsob vytvárania spojenia. Bol rozdelený do troch režimov.

6.1.1 Implementované režimy pripájania

Prvý režim je pôvodný, kedy sa pripája USB zariadenie exportované z Raspberry Pi pripojeného na lokálnej sieti. Avšak naväzovanie spojenia je upravené tak, aby pred samotným pripojením prebehla kontrola hesla. Heslo je kontrolované nasledovne: Aplikácia v Raspberry Pi generuje náhodné dostatočne dlhé číslo (použitie 128bit) toto číslo je následne pomocou hesla zašifrované šifrou AES a odoslané. V aplikácií, ktorá sa snaží pripojiť, prebehne dešifrovanie a následné zašifrovanie hornou

časťou hesla, táto správa je následne poslaná ako odpoveď. V prípade, že sa náhodné číslo po opätovnom dešifrovaní v aplikácii bežiacej na Raspberry Pi nezhodujú, je spojenie ukončené. V opačnom prípade je pripojenie zariadenia umožnené. Tento režim je tiež možné použiť pre vytváranie P2P spojenia prostredníctvom dierovania. Obe koncové zariadenia (Raspberry Pi a počítač užívateľa) vysielajú žiadosti o pripojenie s tým, že navzájom poznajú svoje verejné IP adresy a tiež zdrojové porty žiadostí. Tieto sú obom zariadeniam sprostredkované prostredníctvom serveru. Tento postup je podrobnejšie popísaný v teórii.

Druhý režim pracuje podobne ako prvý, avšak je využívaný v pripájaní prostredníctvom serveru. V inicializačnej správe je požiadavka na server, aby bola komunikácia preposlaná k cieľovému zariadeniu na základe jeho unikátneho identifikátora. V prípade, že je k serveru pripojené takéto zariadenie, začne s preposielaním vzájomných informácií medzi nimi. V opačnom prípade sa čaká 10 sekúnd, či sa takéto zariadenie nepripojí a po tej dobe je spojenie ukončené serverom. Toto opatrenie je z dôvodu odľahčenia prenosových kanálov serveru, a taktiež z dôvodu bezpečnosti, keď nie je možné typovať a skúšať sa pripojiť. O túto synchronizáciu sa stará aplikácia užívateľského rozhrania, ktorej funkčnosť je opísaná neskôr.

Tretí režim pracuje opačne ako prvý. Aplikácia sa nepripája ku Raspberry Pi ale pracuje ako server a čaká, kým sa k nej pripojí vzdialené Raspberry. Vďaka tomuto režimu sa môže pomocou vhodného nastavenia siete (napríklad nastavenie smerovania portov) vytvárať priame P2P spojenie medzi oboma koncovými zariadeniami bez nutnosti využívania výpočtového výkonu a prenosovej linky serveru. Po vytvorení spojenia už zabezpečenie a následné spracovanie funguje presne ako v prvom režime.

6.1.2 Testovanie knižnice

Pre testovanie knižnice bola vytvorená jednoduchá aplikácia v jazyku C++ (keďže aj knižnica je písaná v tomto jazyku), v ktorej je volaná funkcia pripájania z knižnice. Aplikácia má všetky konštanty priamo vpísané v programe a slúži výhradne k debugovaniu knižnice. Užívateľské rozhranie má iba dve tlačidlá jedno pre pripojenie a druhé pre odpojenie zariadenia. Tento program je spolu s knižnicou v jednom projekte pre uľahčenie práce s touto knižnicou. Počas testovania bola odskúšaná požadovaná funkčnosť knižnice, to znamená funkčnosť všetkých troch režimov pripájania, a boli

odstránené chyby, ktoré sa vyskytli počas úpravy pôvodnej knižnice. Takto odskúšaná knižnica môže byť ďalej použitá v užívateľskej aplikácii.

6.2 Úprava aplikácie v Raspberry Pi

Raspberry Pi bude slúžiť len ako server, takže bude spracovávať požiadavky na pripojenie ku UDB zariadeniam zapojeným v jeho portoch. O túto časť sa stará program „usbipd“. Ide o časť pôvodnej aplikácie USB-IP. Pre požadovanú funkčnosť bol tiež vytvorený ďalší program správy zariadení. Slúži predovšetkým ku komunikácii so serverom. Tento program tiež umožňuje užívateľovi nastavovať zariadenie.

6.2.1 Modifikácia časti programu usbipd

Časť programu usbipd bola upravená, aby sa okrem klasického pripájania na lokálnej sieti umožnilo aj pripájanie prostredníctvom serveru.. K pôvodnému režimu serveru čakajúceho na prichádzajúce spojenie bol pridaný aj režim klienta, kedy sa Raspberry Pi snaží pripojiť. Tento režim je možné využiť v prípade pripájania ku stanici užívateľa prostredníctvom serveru, ale je možné ho využiť aj pri vytváraní P2P spojenia dierovaním. Potrebné parametre ako cieľová IP adresa a port sú dodávané prostredníctvom zdieľanej pamäte z aplikácie správy zariadení. Keďže tieto spôsoby pripájania na lokálnej sieti nemajú žiadne opodstatnenie a sú potrebné predovšetkým v sieti Internet a pre ich správne fungovanie, je potrebný server.

6.2.2 Program správy zariadení.

Tento program umožňuje užívateľovi na lokálnej sieti nastavovať zariadenie. Môže nastaviť základné parametre, ktorými sú napríklad heslo zariadenia, IP adresa a brána zariadenia. Nastavovanie prebieha prostredníctvom broadcastu, takže nie je nutné poznať presnú IP adresu nastavovaného zariadenia. Zariadenie však musí ležať v rovnakej sieti ako užívateľ, ktorý sa zariadenie snaží nastaviť. Po výzve zariadenia odpovedia svojimi ID a užívateľ si môže vybrať, ktoré zariadenie bude nastavovať. Pre vloženie nového hesla je potrebné poznať aktuálne heslo. V prípade, že užívateľ aktuálne heslo nepozná, musí priamo zo zariadenia spustiť inicializačnú rutinu, ktorá nastaví heslo na počiatočné. Program komunikuje so serverom, v pravidelných časových intervaloch a posiela serveru informáciu o tom že je zariadenie dostupné.

Počas pripájania k serveru prebieha kontrola nastavenia a jeho prípadná aktualizácia ak je to nutné. Nastavenia obsahujú predovšetkým IP adresu serveru a adresy záložných serverov v prípade výpadku hlavného serveru. Vďaka tomuto postupu bude v prípade potreby možná zmena serverov. Modul ďalej spracováva požiadavky od serveru.

Jednou z požiadaviek je žiadosť o zoznam USB zariadení, ktoré sú k danému Raspberry Pi pripojené. Na požiadavku odpovedá identifikátormi všetkých USB zariadení, ktoré sú pripojené k portom. Ide o tzv. VID a PID, podľa ktorého je možné jednoznačne identifikovať aké zariadenie je k USB portu pripojené.

Ďalšou požiadavkou, ktorú môže server zariadeniu vyslať je žiadosť o pripojenie. Žiadosť obsahuje IP adresu a port, ku ktorému sa má zariadenie pripojiť. Keďže túto požiadavku nespracováva samotný modul usbipd je dôležité, aby tieto informácie dostal spolu so žiadosťou o spracovanie. Táto výmena informácií prebieha prostredníctvom zdieľanej pamäte. A žiadosť je odoslaná pomocou signálu. Ide o štandardné spôsoby výmeny informácií medzi vlákнами pod Unixovou platformou.

6.3 Serverová aplikácia

Serverová aplikácia je v podstate najdôležitejšou časťou v prípade, že komunikácia prebieha prostredníctvom internetu. Ako už z teoretického rozboru tejto aplikácie vyplýva je vhodné, aby bola členená do modulov, ktoré medzi sebou komunikujú a tak vytvárajú jediný funkčný celok. Vzhľadom k tomu, že je nutná spoľahlivosť celého systému, nie je možné takýto systém prevádzkovať s jediným serverom. Celá aplikácia by preto mala byť pre server postavená tak, aby neustále prebiehala synchronizácia so záložnými servermi.

6.3.1 Modul správy užívateľov

Počas implementácie bola v prvom kroku vytvorená databáza používateľov a systém prihlásenia ako aj systém registrácie. Tento modul bol označený ako „Správa užívateľov“. Užívateľ pri registrácii dostáva svoje unikátne identifikačné číslo pomocou ktorého vystupuje v ostatných tabuľkách z dôvodu čo najmenšieho miesta potrebného pre tabuľku (64 bitové číslo zaberá zväčša menej miesta ako meno používateľa a taktiež je práca s ním omnoho jednoduchšia). Na prihlásenie je nutné zadať prihlasovacie meno

a heslo, komunikácia so serverom je zabezpečená voči odpočúvaniu tým, že je použitý šifrovaný prenos.

6.3.2 Modul správy zariadení

V druhom kroku bol vytvorený modul „správy zariadení“. Táto časť programu sa stará predovšetkým o pripojené zariadenia. Z dôvodu univerzálnosti modul dokáže spracovávať požiadavky nielen od Raspberry Pi, ale aj od ľubovlného zariadenia, ktoré by bolo v budúcnosti vyvíjané a ktoré by malo využívať rovnaké servery bez nutnosti ich programovej úpravy. Každé zariadenie je identifikované svojím ID (v prípade zariadenia typu Raspberry Pi ide o unikátne 64bitové výrobné číslo procesora, ktoré je možné získať) a taktiež svojím typom. V danom prípade, keďže sa práca zaoberá vzdialeným programovaním a monitorovaním, ide iba o zariadenia určené pre zdieľanie svojich USB portov.

6.3.3 Modul záložného prepojenia

Nakoniec bol vytvorený modul „záložného prepojenia“. Ten rovnako ako v predchádzajúcom prípade bol realizovaný tak, aby umožnil komunikáciu nielen medzi koncovými aplikáciami, ktoré boli vytvorené, ale aj medzi aplikáciami, ktoré by mohli vzniknúť v budúcnosti. Predstavme si napríklad program pod operačným systémom Windows, ktorý by umožnil zdieľanie svojho sériového pripojenia, prípadne ethernetového pripojenia. Takýmto počítačom by mohla byť napríklad stanica SCADA systému nasadeného priamo vo výrobe a prostredníctvom daných liniek tohto počítača by bolo možné pripojiť sa k PLC zariadeniam, ku ktorým je počítač pripojený už z dôvodu vizualizácie stavu jednotlivých technologických procesov.

Prepojenie medzi stanicami, ktoré budú chcieť tento modul využiť bude prebiehať na základe požiadaviek. Prepojenie prebieha v nasledovných krokoch:

- Užívateľ so svojou aplikáciou **A** vyšle požiadavku na server **S** o pripojenie k určitej cieľovej aplikácie **B**
- Server **S** skontroluje dostupnosť zariadenia a odpovie aplikácií **A** buď pozitívne, že cieľová aplikácia **B** je dostupná, alebo negatívne, ak je zariadenie odpojené alebo nemá prístup k serveru. Server v prípadne dostupnosti odošle cieľovej aplikácií **B** požiadavku, aby sa pripojila k záložnému modulu záložného prepojenia.

- V prípade že je cieľová aplikácia **B** dostupná, klient **A** odošle požiadavku na daný modul. V požiadavke je obsiahnutý unikátny identifikátor aplikácie **B**, ku ktorej sa snaží pripojiť. Následne sa čaká počas krátkej doby (približne 10s), kým sa pripojí aj klient **B**.
- Klient **B** v rovnakom čase ako užívateľ **A** spracováva požiadavku a pripojí sa k modulu. Rovnako ako v správe od klienta **A**, obsahuje správa od cieľovej aplikácie **B** unikátny identifikátor a aplikácia taktiež čaká približne 10s na spracovanie.
- Pokiaľ sa v module záložného prepojenia objaví správa so žiadosťou na nejaké ID a zároveň sa v časovom intervale prihlási aj zariadenie, ktoré sa týmto ID identifikuje. Modul si vytvorí komunikačný pár a začne s preposielaním paketov od aplikácie **A** k aplikácií **B** a naopak.
- V prípade, že do časového okamihu daný stav nenastane, modul uzavrie spojenia.

Týmto spôsobom je možné naviazať spojenie medzi ľubovoľnými aplikáciami, ktoré splnia požadovanú postupnosť a majú svoje unikátne ID.

6.4 Užívateľská aplikácia

Ako posledná bola vytvorená užívateľská aplikácia. Táto aplikácia vo svojom jadre využíva vytvorenú dynamickú knižnicu, ktorá už bola skôr spomenutá a taktiež bola opísaná jej základná funkčnosť. Užívateľská aplikácia umožňuje užívateľovi pripájať sa ku zariadeniam na lokálnej sieti. Samozrejme na pripojenie je nutné poznať správne heslo. Zároveň umožnenej užívateľovi pripájať sa ku vzdialeným zariadeniam prostredníctvom serveru. Pre túto funkcionálnosť je nutné, aby si používateľ vytvoril profil, jednoducho vyplnením registračného formuláru v aplikácii. Tento formulár môžete vidieť na nasledovnom obrázku:

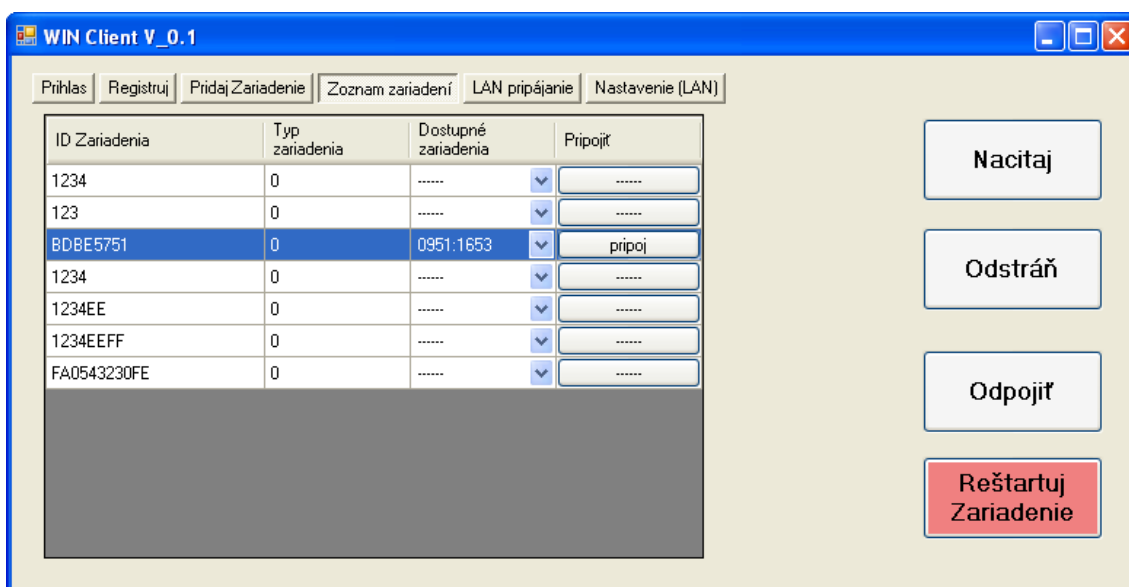
Obrázok 8: Ukážka registračného formuláru

Užívateľ pre registráciu musí zadať svoje prihlasovacie meno. Meno musí byť unikátne a v prípade, že je už použité iným užívateľom, registrácia nebude úspešná. Ďalej je dôležité, aby zadal heslo ku svojmu účtu. Z dôvodu aby nespravil v zadávaní chybu, musí heslo zadať dvakrát. Posledným parametrom je e-mail. Ten je dôležitý iba pri prípadnej korešpondencii s užívateľom.

Po úspešnej registrácii sa používateľ môže prihlásiť. Do prihlasovacieho formuláru užívateľ napíše svoje prihlasovacie meno a heslo. Po prihlásení je užívateľ oprávnený k pridávaniu nových zariadení. Formulár na pridanie zariadenia možno vidieť na nasledovnom obrázku:

Obrázok 9: formulár pridávania zariadení

Užívateľ si môže skontrolovať dostupnosť už pridaných zariadení. Zariadenia sa pridávajú do databázy, ktorá sa nachádza na strane serveru, takže používateľ si môže kontrolovať svoje zariadenia z ľubovoľného miesta, pokiaľ má prístup k internetu. V prípade kontroly dostupnosti sa odošle žiadosť na server. Server si z databázy načíta zoznam zariadení, ktoré patria užívateľovi, čo odoslal žiadosť a postupne skontroluje ich stav, či sú dostupné alebo nie a taktiež aké USB zariadenia sa poskytujú k zdieľaniu. Ukážkový zoznam zariadení možno vidieť na nasledovnom obrázku:

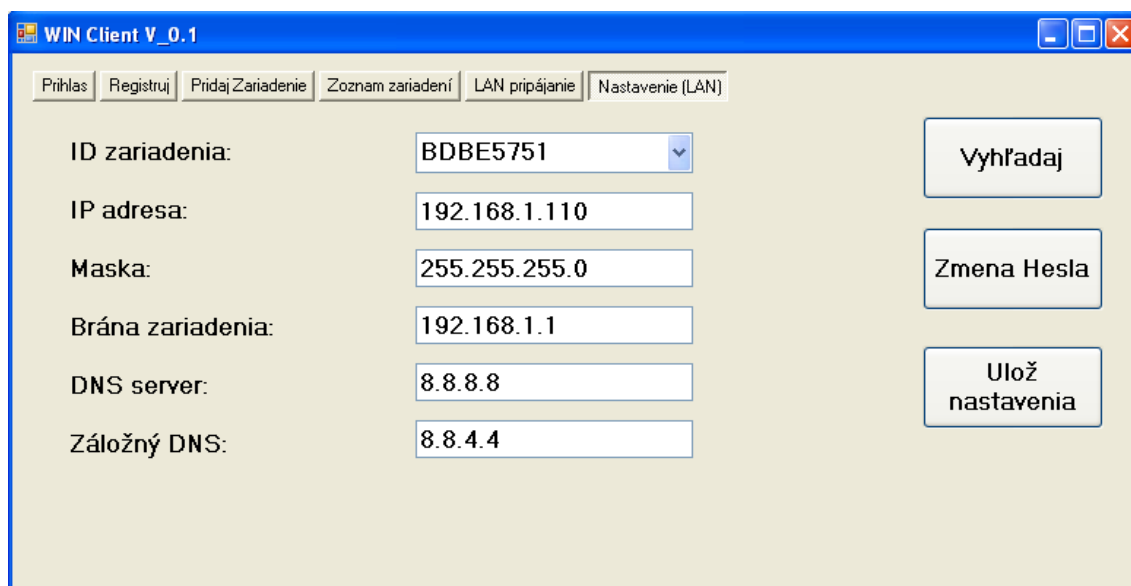


Obrázok 10: Ukážka zoznamu zariadení

Ako je vidieť na obrázku, ukážkový užívateľ má pridaných 7 zariadení. V prvom stĺpci vidí ID jednotlivých zariadení. Ide o číslo v šestnástkovej sústave. V druhom stĺpci je typ zariadenia, keďže zatiaľ existuje iba jediné zariadenie tak typ všetkých zariadení je 0 toto pole bude využité v budúcnosti, keď sa vytvoria viaceré aplikácie, prípadne môže byť použité pre presnejšiu špecifikáciu zariadenia. Stĺpec dostupné zariadenia ukazuje identifikátory zariadení, ktoré sú pripojené. Formát je nasledovný VID:PID, ide o čísla zobrazené v šestnástkovej sústave. Tieto dve čísla jednoznačne identifikujú každé USB zariadenie. V prípade, že v kolónke nie je zobrazené žiadne číslo, znamená to, že zariadenie nie je dostupné. Buď je zle nakonfigurované alebo nemá prístup k internetu. V prípade, že je zariadenie dostupné, je zobrazený spomenutý identifikátor. Ak sa chce užívateľ pripojiť, môže si vybrať, na ktoré zariadenie sa chce pripojiť a po nastavení klikne na tlačidlo pripoj, ktoré sa nachádza v odpovedajúcom riadku v tabuľke. V prípade vzdialeného pripájania nie je

nutné zadávať heslo, pretože heslo bolo zadané počas pridávania zariadenia do zoznamu a o heslo sa ďalej stará server. Na ukázkovom obrázku môžeme vidieť, že je dostupné jediné zariadenie a je k nemu pripojené zariadenie s identifikátorom 0951:1653. Užívateľ má taktiež možnosť vymazať zariadenie zo zoznamu v prípade, že sa k nemu už v budúcnosti neplánuje pripájať.

Ďalšími dvoma záložkami v aplikácii sú nastavenie a využívanie programu v lokálnej sieti. Na nasledovnom obrázku možno vidieť formulár pre nastavenie zariadenia.

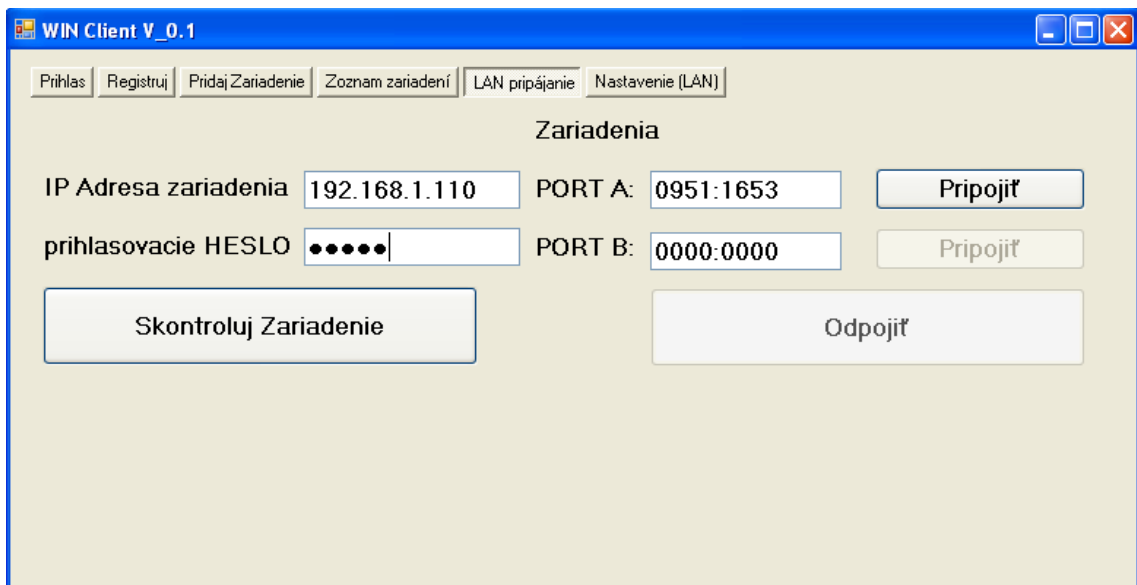


WIN Client V_0.1		
Prihlas Registruj Pridaj Zariadenie Zoznam zariadení LAN pripájanie Nastavenie (LAN)		
ID zariadenia:	BDBE5751	Vyhľadaj
IP adresa:	192.168.1.110	Zmena Hesla
Maska:	255.255.255.0	Ulož nastavenia
Brána zariadenia:	192.168.1.1	
DNS server:	8.8.8.8	
Záložný DNS:	8.8.4.4	

Obrázok 11: Ukážka formuláru pre nastavenie zariadenia

Používateľ si v tomto formulári najskôr vyhľadá dostupné zariadenia na lokálnej sieti kliknutím na tlačidlo „Vyhľadaj“. Po tejto operácii sa používateľovi zobrazí jedno zo zariadení, ktoré bolo nájdené na lokálnej sieti. Medzi zariadeniami môže užívateľ prepínať tým, že vyberie ID zariadenia, ktoré chce nastavovať. Užívateľ môže zmeniť nastavenie zariadenia a ak chce nové nastavenie uložiť, klikne na tlačidlo „Uložiť nastavenia“. Následne bude požadované heslo ku zariadeniu potrebné pre zapísanie nového nastavenia. Ak chce používateľ aktuálne heslo zmeniť, klikne na príslušné tlačidlo. Opäť je od neho vyžiadané pôvodné heslo ku zariadeniu, ale na rozdiel od predchádzajúceho prípadu, má tiež možnosť zadať nové heslo.

Poslednou záložkou, ktorá sa v aplikácii nachádza je Pripájanie na lokálnej sieti LAN. prostredníctvom tejto záložky je možné využívať zariadenia na lokálnej sieti bez nutnosti pripájania ku serveru. Záložka vyzerá nasledovne:



Obrázok 12: ukážka formuláru pripájania v sieti LAN

Užívateľ najskôr zadá IP adresu zariadenia a následne klikne na tlačidlo „Skontroluj zariadenie“. Aplikácia vyhľadá zariadenia podľa zadanej IP adresy. V prípade, že sa dané zariadenia na lokálnej sieti skutočne nachádzajú, zobrazia sa identifikátory zariadení, ktoré má pripojené k USB portom. Ako môžeme na obrázku vidieť, je obsadený iba jeden port a je v ňom pripojené rovnaké zariadenie ako bolo použité v demonštrácii vzdialeného pripájania. Po vyhľadaní zariadení si užívateľ môže vybrať, ku ktorému zariadeniu sa chce pripojiť. Pred pripojením je samozrejme nutné zadať správne heslo ku zariadeniu, bez ktorého pripojenie nebude možné. Po zadaní hesla môže kliknúť na tlačidlo pripojiť a následne prebehne pripojenie.

7 VYHODNOTENIE NAVRHNUTÉHO RIEŠENIA

V tejto kapitole sa pozrieme na výsledky, ktoré boli dosiahnuté po úspešnej programovej realizácii všetkých požadovaných modulov a taktiež si povieme o limitujúcich parametroch, ktoré boli počas testovania zistené. Na test som sa rozhodol overiť rýchlosť viacerých USB zariadení v prípade, že sú pripojené vzdialene (s využitím serveru) prostredníctvom tohto riešenia. Test prebiehal na USB kľúči (Kingston DataTraveler 100G2), ďalej bol použitý USB prevodník na RS232 (STLab model U-700), a taktiež prebehlo meranie na prevodníku USB-Ethernet (RIKOMAGIC USB 2.0 - 10/100MBit LAN adaptér) ktorý používa ako prevodník čip od spoločnosti asix elec. corp. S označením Ax88:72a Počas testovania som sa snažil zistiť, akými rýchlosťami je možné so zariadeniami komunikovať a či je takto pripojené zariadenie možné použiť.

7.1 Testovanie USB kľúča

Ako už bolo spomenuté, pre testovanie som použil USB kľúč od výrobcu Kingston DataTraveler 100G2. Po pripojení bolo nutné nainštalovať ovládač manuálne, pretože automatická inštalácia nedokázala identifikovať presný typ zariadenia. Toto bolo spôsobené tým, že v počítači počas testovania nebol inštalovaný originálny ovládač spoločnosti Kingston k danému zariadeniu, ale bol použitý univerzálny ovládač USB pre „Mass storage device“. Po manuálnom priradení tohto ovládača sa zariadenie pripojilo a začalo pracovať.

Na meranie rýchlosti USB kľúča som použil program „Check Flash 1.15.2“, pomocou ktorého som načítaval dáta z kľúča a program postupne priemerovoval rýchlosť. Odpočet som vykonal približne po desiatich minútach, kedy bol priemer rýchlosti stabilizovaný.

Oneskorenie, ktoré som potreboval na sieti vytvoriť, aby som dokázal simulovať skutočné parametre pri prenose dát cez internet som použil príkaz `tc` z balíčka *iproute2*, ktorý je už obsiahnutý v základnej verzii Rasbianu (operačný systém Raspberry Pi). Pomocou tohoto príkazu je možné nastaviť oneskorenie na sieti a taktiež je možné

nastaviť maximálnu prenosovú rýchlosť s ktorou bude zariadenie prijímať a odosielať pakety. Zariadenie bolo pripojené pomocou štandardnej 100Mb siete.

Pre nastavenie oneskorenia som teda použil nasledovný príkaz:

- `sudo tc qdisc add eth0 root netem delay 10ms`

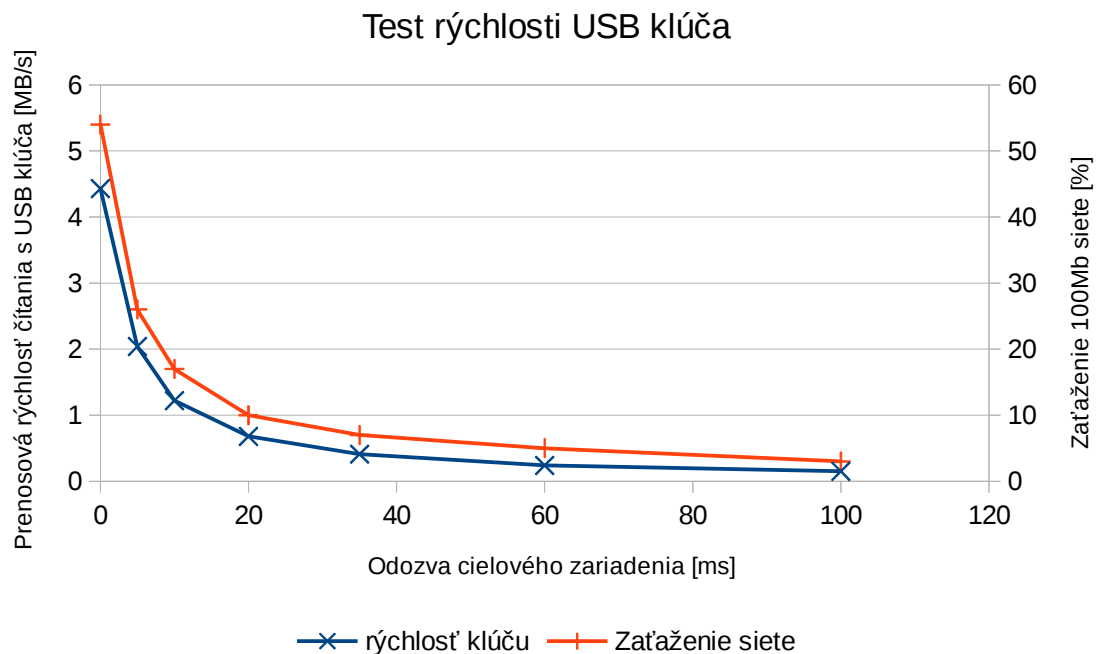
následne je možné nastavené dopravné oneskorenie meniť pomocou príkazu:

- `sudo tc qdisc change dev eth0 netem delay 20ms`

Dosiahnuté výsledky je možné vidieť v tabuľke a následne sú zobrazené v grafe

Tabuľka 4: Merané hodnoty rýchlosti USB kľúča

Odozva cieľového zariadenia [ms]	Prenosová rýchlosť [MB/s]	Zaťaženie siete [%]	Rýchlosť voči priamo pripojenému kľúču [%]
0	4,43	54	66
5	2,04	26	30
10	1,22	17	18
20	0,68	10	10
35	0,41	7	6
60	0,24	5	4
100	0,15	3	2



Obrázok 13: Výsledky testovania rýchlosti USB kľúča

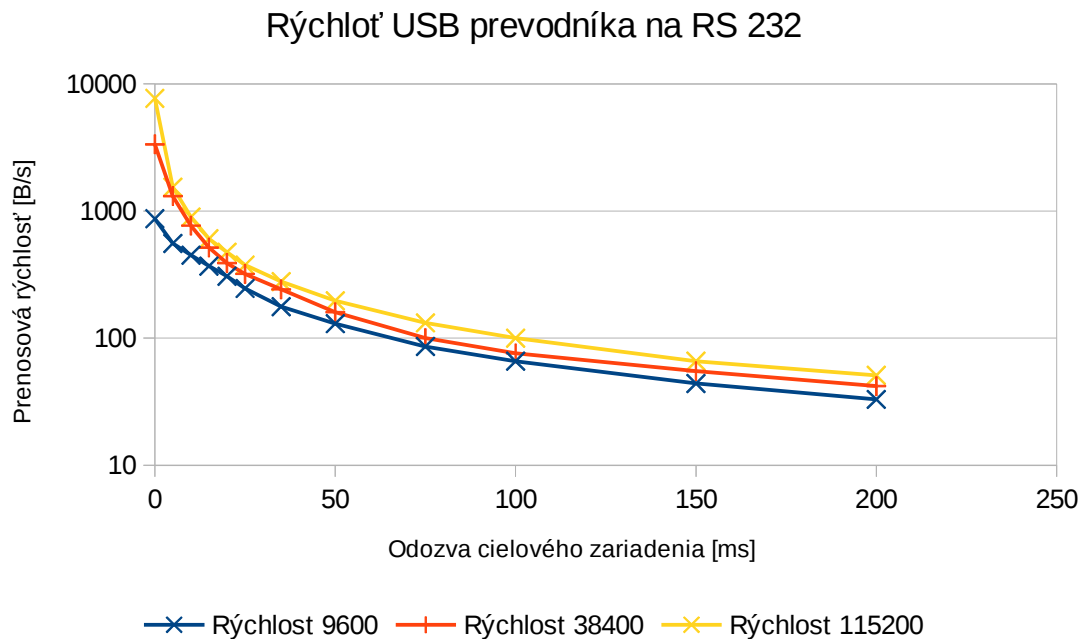
7.2 Testovanie USB prevodníka na RS232

Ďalším testovaným zariadením bol prevodník S USB na RS232 od výrobcu STLab model U-700. Na pripojenie prevodníka priamo k počítaču bolo nutné nainštalovať ovládač priamo k tomuto zariadeniu. Po inštalácii ovládača a následnom zapojení prevodníka do voľného portu Raspberry Pi, bolo možné si toto zariadenie vzdialene pripojiť. Po vzdialenom pripojení prebehla automatická inštalácia ovládača a po dokončení bolo možné začať používať prevodník.

Na testovanie rýchlosti som si vytvoril jednoduchý program. Užívateľ vyberie sériový port, ktorý chce testovať. Nastaví prenosové parametre a spustí meranie. Počas merania mu ukazuje priemernú rýchlosť komunikácie ako aj uplynutý čas od spustenia. Princíp aplikácie je nasledovný - po spustení aplikácie sa vytvorí vlákno, ktoré nepretržite posieľa na sériový port správy o nastavenej dĺžke a v pravidelných časových intervaloch načítava dostupné správy, ktoré mu prichádzajú a na základe času, ktorý uplynul a množstva správ, ktoré sa vrátili, sa prepočítava prenosová rýchlosť. Aby som mohol týmto spôsobom merať na sériovom porte bolo nutné, aby som do portu zapojil spätnú väzbu, ktorá mi všetky vyslané správy okamžite preposielala nazad. Ide o prepojenie TR a RX pinu. Výsledky merania možno vidieť v nasledovnej tabuľke:

Tabuľka 5: Meranie prevodníka RS232

Nastavenie:	rýchlosť-9600 baud/s	počet stop bitov-1	parita-párna
Odozva cieľového zariadenia [ms]	Prenosová rýchlosť [B/s]	Zaťaženie siete [%]	Rýchlosť voči priamo pripojenému kľúču [%]
0	870	3	99
5	556	0,64	64
10	450	0,36	51
15	369	0.25	42
20	307	0.2	35
25	246	0,16	28
35	177	0,11	20
50	130	0,08	15
75	86	0,05	10
100	66	0,04	8
150	44	0,03	5
200	33	0,02	4



Obrázok 14: Rýchlosť prevodníka na RS232

Ako sa ukázalo, nastavená dĺžka správy nemá na prenosovú rýchlosť vplyv z čoho vyplýva, že po USB zbernici sa pravdepodobne posielajú jednotlivé bajty, ktoré je nutné poslať a nie celé správy. Tieto vlastnosti sú dané použitým HW ako aj jeho ovládačom. Nastavenie a spustenie aplikácie

7.3 Test prevodníka USB na Ethernet

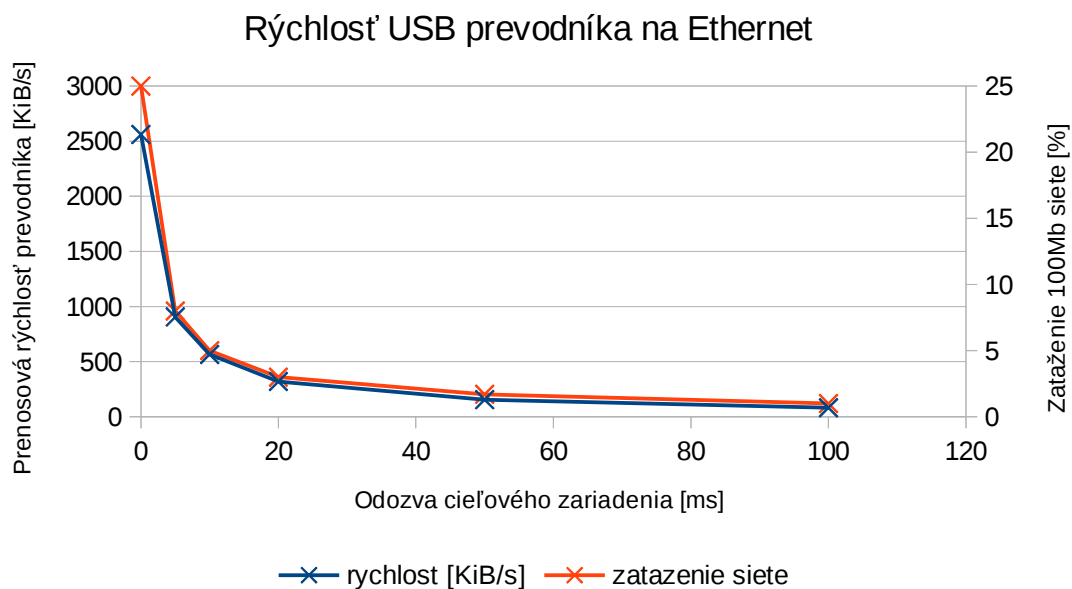
Posledným testovaným zariadením bol prevodník USB-Ethernet (RIKOMAGIC USB 2.0 - 10/100MBit LAN adaptér). Podobne ako v predchádzajúcom teste, aj na tento prevodník bolo potrebné nainštalovať ovládač pred jeho použitím. Po nainštalovaní ovládača bol postup pripájania totožný s postupom v predošlom prípade.

Pomocou takto pripojeného prevodníka som sa pripájal na vzdialený počítač, testoval som jeho odozvou (Ping) a tiež prenosovú rýchlosť takto vzniknutej linky kopírovaním súborov. Získané výsledky merania možno vidieť v nasledovnej tabuľke:

Tabuľka 6: Meranie prevodníka USB-Ethernet

Odozva cieľového zariadenia [ms]	Test odozvy [ms]	Prenosová rýchlosť [KiB/s]	Zaťaženie siete [%]
0	2,23	2560	25
5	7,17	905	8
10	12,14	566	5
20	22,09	320	3
50	52,19	155	1,7
100	101,93	82	1

Ako môžeme vidieť prenosová rýchlosť je nižšia ako keby sme sa pripájali priamo, a aj zaťaženie siete v prípade monitorovania alebo programovania je taktiež pomerne nízke. V prípade vzdialeného programovania sa toto spomalenie viditeľne prejaví iba pri nahrávaní programu, ktoré bude pochopiteľne trvať dlhšie. Namerané hodnoty je možné vidieť na nasledovnom grafe:



Obrázok 15: Rýchlosť prevodníka USB-Ethernet

7.4 Zhodnotenie

výhodou daného riešenia je možnosť pripájania napríklad servo meničov, lacnejších typov PLC, alebo niektorých HMI staníc, ktoré môžu byť programované a nastavované jedine prostredníctvom USB rozhrania. Možnosť vzdialeného pripájania USB zariadení sa bežne nevyskytuje a komerčné zariadenia touto možnosťou nedisponujú. V prípade, že na oboch koncoch žiadaného spojenia je dostatočne rýchly internet s nízkym oneskorením, je možné vytvoriť spojenie, ktorého odozva nepresiahne 50ms a tým pádom by zariadenie malo pracovať dostatočne rýchlo a spoľahlivo, aby s ním bolo možné pracovať.

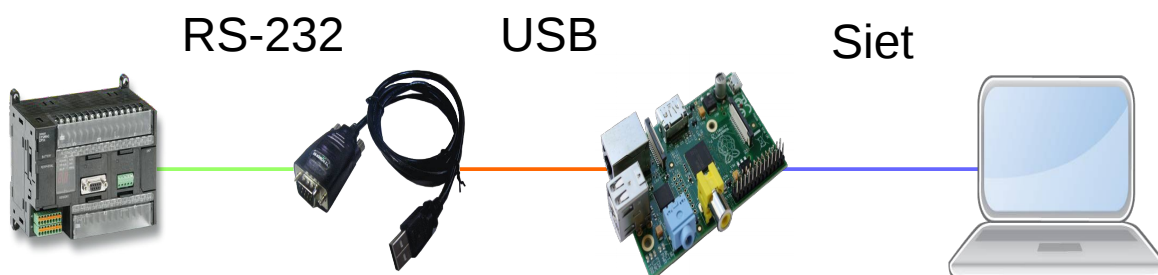
Počas pripájania a testovania sa ukázalo, že najdlhší čas zohráva spracovanie nového zariadenia operačným systémom. Ako určite všetci zo skúseností poznajú, že pripojený USB kľúč je niekedy možné používať až po cca 5-10 sekundách. Tento čas spôsobuje najdlhšie čakanie pri pripájaní vzdialeného zariadenia. Horšie parametre spojenia tento čas pripájania úmerne predlžia, keďže OS musí dokončiť inicializačnú komunikáciu, ktorá je zhoršenými parametrami siete spomalená. Časové intervaly potrebné pre samotné naviazanie spojenia a jeho zašifrovanie môžeme úplne zanedbať, keďže tento čas sa pohybuje do jednej sekundy (v prípade, že odozvy jednotlivých klientov na server neprekračujú hodnotu 50ms).

8 UKÁŽKOVÁ KOMUNIKÁCIA S PRIEMYSELNÝMI ZARIADENIAMÍ

Testovanie prebehlo na troch priemyselných zariadeniach. Prvým testovaným zariadením bolo PLC Omron CP1L s osadeným sériovým rozhraním. Pripojenie prebehlo prostredníctvom prevodníku s USB na RS232 (STLab model U-700), ktorý bol taktiež použitý na meranie rýchlosti pri rôznych vlastnostiach siete v predošlej kapitole. Druhým zariadením je PLC WAGO 750-881, na ktoré je vytvorená ukážková aplikácia pre načítavanie vstupov a výstupov PLC. Tretím zariadením je HMI stanica Weintek MT6050iP programovateľná priamo prostredníctvom USB rozhrania. Nakoniec bolo realizované overenie na laboratórnom modeli vŕtačky pripojenom k rovnakému PLC ako v druhom teste.

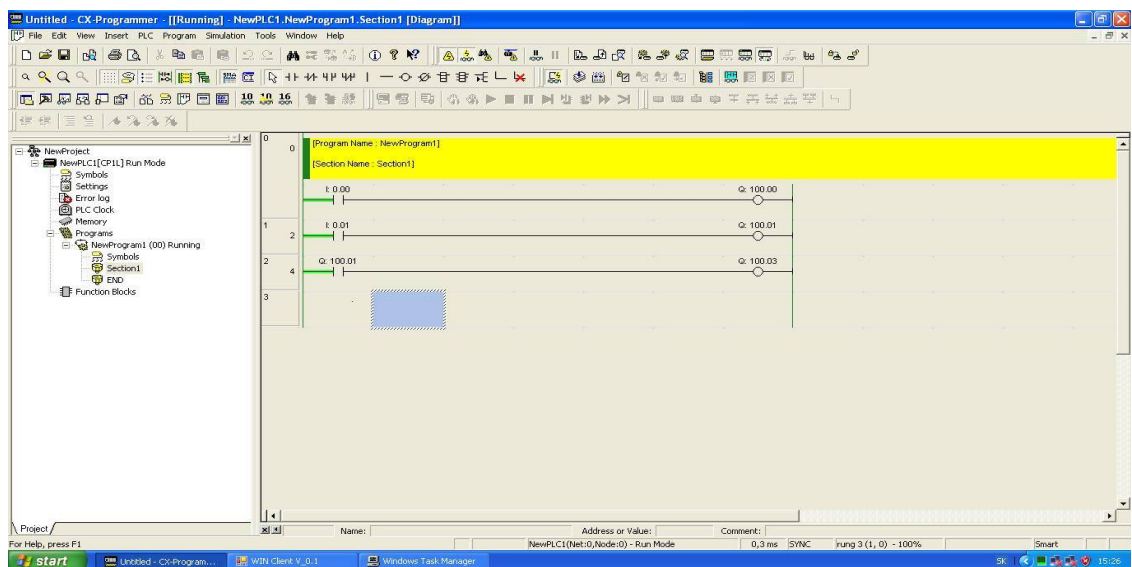
8.1 Pripojenie k PLC CP1L cez RS232

V prvom kroku bolo potrebné všetky zariadenia správne zapojiť. Do sériového rozhrania PLC bol zapojený prevodník Omron štandardu na klasický štandard RS-232. pre komunikáciu po sériovej linke sú využívané iba tri žily (RX, TX a GND) a tieto sú v konektore Omron štandardu zapojené na iných pinoch ako v klasickom štandarde. Tento prevodník bol nasledovaný prevodníkom USB-RS232 (STLab model U-700), ktorý bol zapojený v Raspberry-Pi. Raspberry-Pi bolo následne prepojené s počítačom prostredníctvom siete. Celkové grafické znázornenie zapojenia možno vidieť tiež na nasledovnom obrázku:



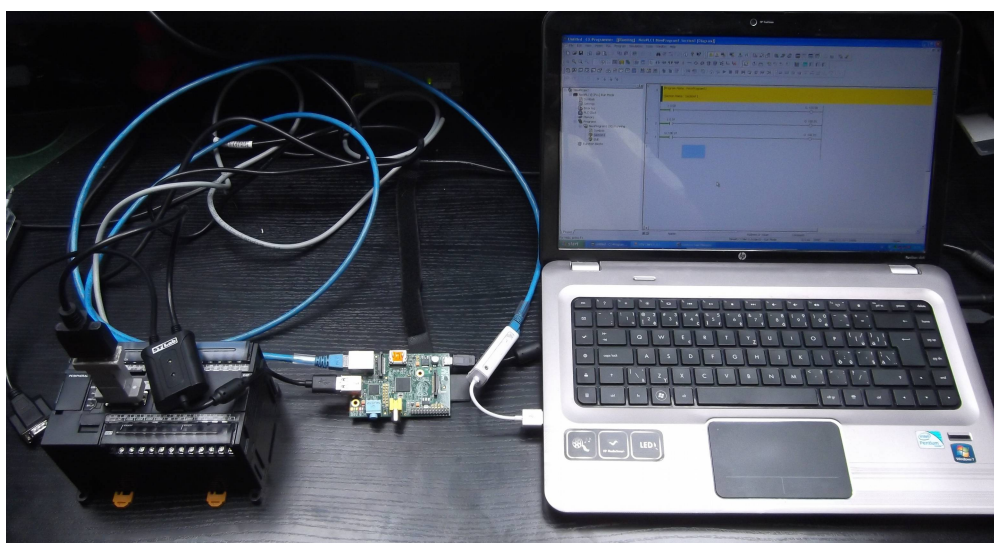
Obrázok 16: Grafické znázornenie zapojenia (Omron PLC)

Z počítača sa prostredníctvom užívateľskej aplikácie naviazalo spojenie s Programovacím zariadením. Následne sa vybral prevodník, ku ktorému sme sa chceli pripojiť. Po pripojení prevodníka bolo možné zapnúť program CX-Programmer, pomocou ktorého prebieha programovanie PLC zariadení firmy Omron. Po bežnom nastavení, ktoré sa musí vykonávať aj v prípade priamo pripojeného PLC bolo možné sa pripojiť a sledovať vnútorný stav zariadenia. Ukážku pripojeného stavu možno vidieť na nasledovnom obrázku:



Obrázok 17: Ukážka pripojeného prostredia CX-programer ku PLC

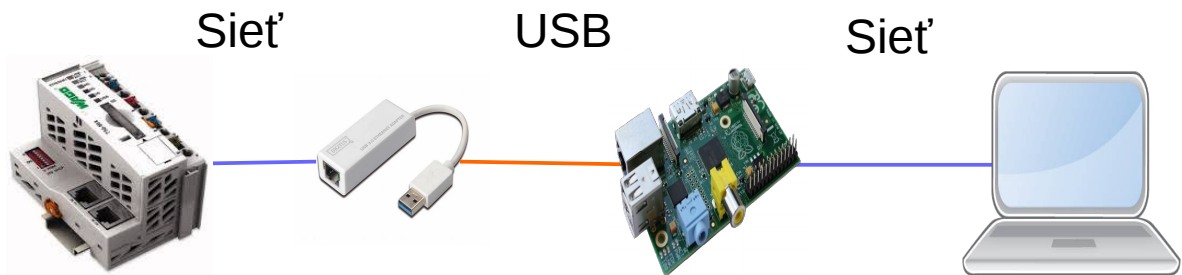
Celková zostava v reálnom prevedení s kompletným zapojením v pripojenom stave je zobrazená na nasledovnom obrázku:



Obrázok 18: Reálne zapojenie PLC CP1L

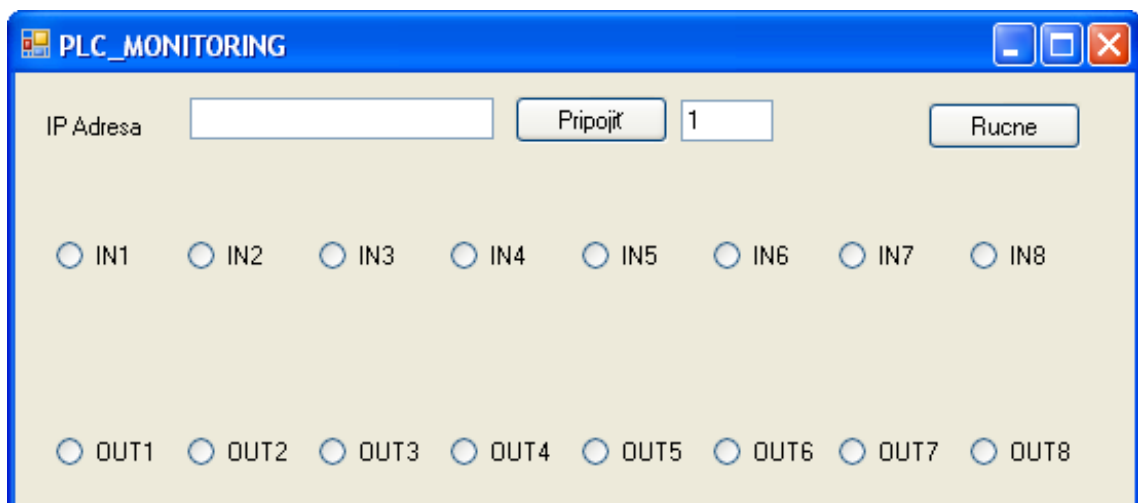
8.2 Pripojenie k PLC WAGO 750-881 cez Ethernet

Na druhý test bolo použité PLC spoločnosti WAGO model 750-881. pripájanie k danému PLC prebieha prostredníctvom siete. Zapojenie je principiálne rovnaké so zapojením v predošlom prípade s tým, že bol namiesto prevodníka USB-RS232 použitý prevodník USB-Ethernet. Zapojenie je teda nasledovné:



Obrázok 19: Grafické znázornenie zapojenia (WAGO PLC)

Po pripojení prevodníka sa zobrazí tento HW ako priamo pripojený k počítaču. Keďže ide o prevodník na ethernet, tak sa po správnom nainštalovaní zobrazí v dostupných sieťových pripojeniach nové sieťové pripojenie, ktoré zodpovedá vzdialenému prevodníku. Na tomto rozhraní je nutné nastaviť IP adresu podľa toho, ako je nastavené vzdialené PLC, aby bolo možné spojenie. Postup je v podstate zhodný ako v prípade že by bolo toto PLC priamo pripojené k ethernetovému rozhraniu počítača, kde je nutné nastaviť správnu IP adresu. Po nastavení siete je možné sa pripojiť priamo k PLC. Pre testovanie bola vytvorená aplikácia, ktorá komunikuje s PLC prostredníctvom Modbus TCP a je schopná vizualizovať vstupy a výstupy s PLC. Ukážku komunikácie možno vidieť na nasledovnom obrázku:



Obrázok 20: Ukážka aplikácie komunikujúcej s PLC cez Modbus a zobrazujúcej I/O

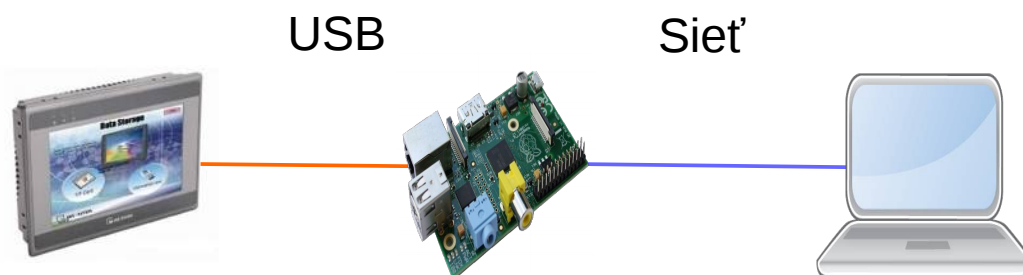
Aplikácia komunikuje s PLC prostredníctvom protokolu Modbus TCP. Pre túto komunikáciu aplikácia využíva komunikačnú knižnicu. Presnejšie využívaním funkcie pre čítanie stavových registrov. Aplikácia umožňuje prepnutie do ručného režimu a následné ovládanie výstupov PLC. Pre tento účel však musí byť správne prispôbená aj aplikácie v PLC. Pre lepšie porozumenie komunikácie s PLC odporúčam naštudovanie dokumentácie k danému typu PLC (viz. [19]). Základné Modbus adresy nutné ku komunikácii s Wago PLC možno vidieť v nasledujúcej tabuľke:

Tabuľka 7: Adresy protokolu Modbus TCP pre PLC WAGO [19]

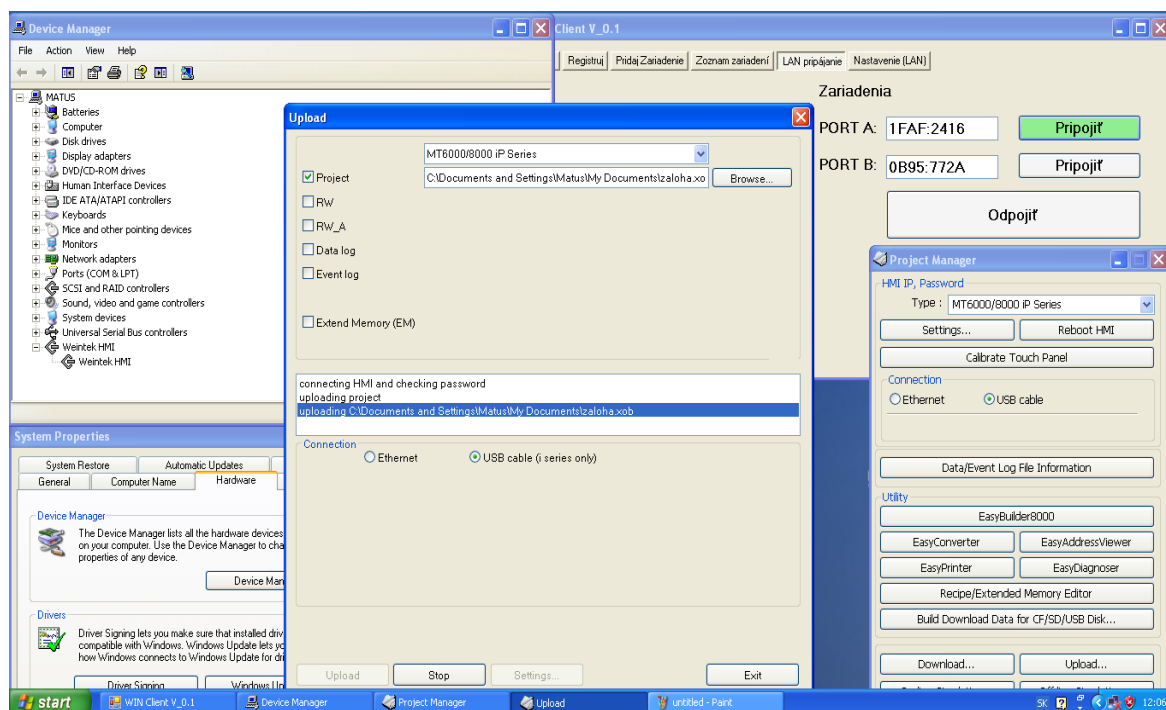
Popis	Adresný rozsah
Fyzické vstupy do PLC	0-255
Fyzické výstupy s PLC	512-767
Vnútorne premenné PLC s označením M (NOVRAM)	12288-24575

8.3 Pripojenie k HMI stanici Weintek MT6050iP cez USB

Ďalším zariadením, na ktorom prebehli testy, bola HMI stanica spoločnosti Weintek Labs., Inc. Model použitého HMI bol MT6050iP. Keďže toto zariadenie sa programuje priamo prostredníctvom USB rozhrania, bolo jeho pripojenie k programovaciemu modulu priamo bez prevodníkov, ktoré bolo nutné využiť v predchádzajúcich prípadoch. Postup pripájania bol zhodný s postupom v predošlých prípadoch. Po pripojení a nainštalovaní ovládača bolo možné začať komunikovať s HMI stanicou. Na nasledujúcich obrázkoch možno vidieť grafické znázornenie pripojenia, ktoré je bez prevodníkov, ktoré boli použité v predošlých testoch a tiež sťahovanie aktuálneho programu s HMI stanice.



Obrázok 21: Grafické znázornenie pripojenia (Weintek HMI)

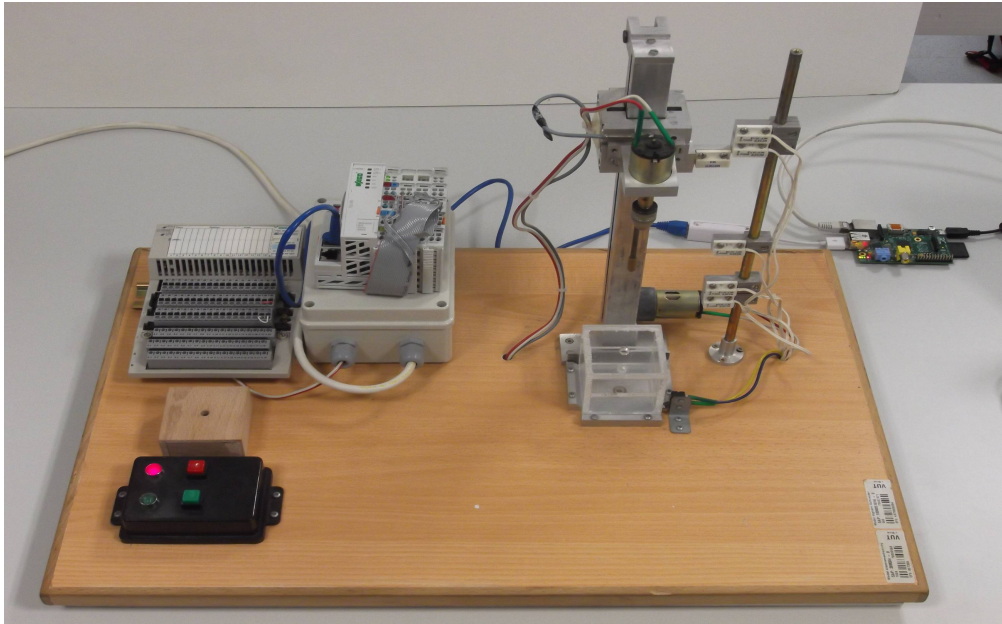


Obrázok 22: komunikácia so vzdialene pripojenou HMI stanicou

Na obrázku je možné vidieť v pravom hornom rohu aplikáciu vzdialeného programovania. Môžeme si všimnúť, že HMI má unikátny identifikátor 1FAF:2416, ku ktorému prebehlo pripojenie. V ľavom hornom rohu je možné vidieť okno správcu zariadení. V ňom je v poslednom riadku vidieť samotné HMI zariadenie označené ako „Weintek HMI“. V pravom spodnom rohu je už okno programu EasyBuilder 8000, ktorý slúži pre programovanie a nastavovanie tejto série HMI od daného výrobcu. Pre overenie funkčnosti pripojenia som použil možnosť „Upload“, ktorá vytvorí zálohu aktuálneho programu v HMI. Táto záloha prebehla úspešne. Tiež bolo možné nahráť nový program, rovnako aj na diaľku celú stanicu spravovať. Tento test overil, že prostredníctvom navrhutej koncepcie je možné sa priamo pripájať aj k zariadeniam, ktoré disponujú programovacím USB portom. Avšak v prípade iných USB zariadení je rozumné overiť, že systém spoľahlivo pracuje aj s tými.

8.4 Pripojenie k laboratórnemu modelu vrtačky

Posledným bodom zadania bolo pripojenie a otestovanie na laboratórnom modeli. Pre tento účel bol vybraný jednoduchý model vrtačky. Tento model bol následne pripojený k PLC WAGO 750-881, ktoré bolo použité v druhom teste. Fyzické prevedenie možno vidieť na nasledovnom obrázku:

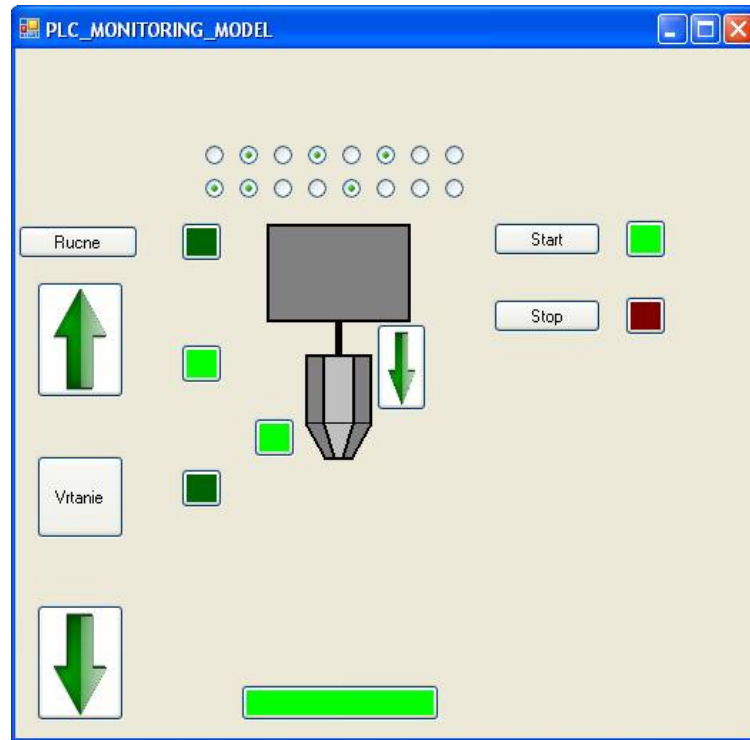


Obrázok 23: Model laboratórnej vrtačky

Model vrtačky obsahuje 2 akčné členy. Motor pre pohyb vrtačky hore a dole ovládaný dvomi digitálnymi výstupmi. A tiež motor samotnej vrtačky. Tento je možné točiť iba jedným smerom, takže je potrebný jeden výstup. Model obsahuje LED signalizáciu stavu (zelená a červená LED), ktoré sú ovládané digitálnymi výstupmi. Celkovo 5 digitálnych výstupov. Model je možné ovládať pomocou dvoch tlačidiel (štart a stop). Poloha vrtačky má tri polohy, podľa výšky. Ďalej sú tu vstupy od snímaču obrábaného materiálu a snímač priblíženia vrtačky k materiálu. Celkovo ide o 7 digitálnych vstupov.

Pre tento laboratórny model bola vytvorená jednoduchá riadiaca aplikácia v PLC, ktorá umožňuje okrem lokálneho riadenia a vzdialeného monitorovania stavov aj vzdialené ovládanie. Je možné sa vzdialene prepnúť do ručného režimu a ovládať

jednotlivé akčné členy bez ohľadu na požadovaný stav automatického režimu. Program pre PLC je priložený v prílohe. Ďalej bola vytvorená aplikácia, ktorá s týmto programom komunikuje a dokáže riadiť model vzdialene. Ukážku rozhrania aplikácie je možné vidieť na nasledovnom obrázku:



Obrázok 24: Ukážka aplikácia pre riadenie laboratórneho modelu

Celé pripojenie prebehlo prostredníctvom modulu vzdialeného pripájania rovnako, ako v prípade pokusu s WAGO PLC. Vďaka tomuto konceptu bolo možné model ovládať vzdialene a tiež sledovať jednotlivé vstupy a výstupy. V praxi by bolo možné sa k takto pripravenému zariadeniu pripojiť z ľubovoľného miesta prostredníctvom Internetu a vzdialene sledovať jeho obsluhu, prípadne identifikovať poruchu. (Napríklad stav kedy je výstup z PLC zopnutý ale motor sa nehýbe. V takomto prípade by mohol byť poškodený motor, jeho riadiaca jednotka alebo fyzický výstup s PLC. Vďaka tomuto prístupu je možné sa zamerať na možné chyby s rýchlym vylúčením nepravdepodobných, akými môže byť napríklad čakanie na dosiahnutie polohy alebo čakanie na prítomnosť obrobku, ktoré by v prípade poškodenia snímačov nikdy nemohli nastať). Týmto testom som ukázal, že nasadenie zariadenie je možné. Nielen pre vytvorenie ukážkového spojenia, ale aj pre ovládanie a monitorovanie stavu

zariadenia. Pri správnom navrhnutí aplikácie je preto možné sledovať ľubovoľne zložitý stroj, výrobný automat, alebo dokonca celé automatizované celky.

9 ZÁVER

Po preštudovaní problematiky bola navrhnutá koncepcia obsahujúca popis jednotlivých aplikácií, ktoré boli následne realizované. Popis obsahuje nielen návrh prepojenia medzi jednotlivými aplikáciami, ale aj štruktúru jednotlivých aplikácií a popis samotnej činnosti. Počas realizácie samotných aplikácií bolo nutné pridať niektoré nové vlastnosti, ktoré pri návrhu neboli uvažované, avšak implementácia si ich vyžiadala. Ich pridanie predovšetkým znížilo užívateľské nároky a zvýšilo komfort.

Po úspešnej realizácii všetkých programových častí bolo možné prejsť k testovaniu celkového riešenia. Prebehlo niekoľko testov na rôznych zariadeniach (USB kľúč, prevodník USB-Ethernet, Prevodník USB-RS232 ...) ktoré poukazujú na nutnosť dostatočne rýchleho prepojenia s nízkou latenciou, ktorá je pre využívanie tejto koncepcie kľúčová. Pre účely testovania bola vytvorená jednoduchá aplikácia na meranie rýchlosti na sériovej linke. Z grafov a meraní bolo zistené výrazné spomalenie a zníženie prenosových rýchlostí pri zvýšení odozvy.

Testy prebehli na fyzických priemyselných zariadeniach (presnejšie PLC od dvoch rôznych výrobcov a HMI stanica). Počas testov sa podarilo vzdialene pripojiť k týmto zariadeniam a bolo možné ich programovať alebo monitorovať. Pre účely vzdialeného sledovania a riadenia stavov PLC bola vytvorená aplikácia komunikujúca protokolom Modbus TCP s PLC zariadením.

V poslednom kroku bolo riešenie experimentálne odskúšané na laboratórnom modeli vráčky. K tomuto modelu bola vytvorená jednoduchá vizualizácia a následne bolo odskúšané vzdialené pripájanie k zariadeniu.

Testy dopadli úspešne. Pri teste merania rýchlosti bolo naväzovanie spojenia prostredníctvom serveru, vďaka čomu je odskúšané, že pri správnom nastavení celého systému je možné sa pripojiť z ľubovoľného miesta v Internete k zariadeniam, na ktorých prebehli testy, prípadne k podobnému alebo k zložitejšiemu zariadeniu a následne ho vzdialene programovať, monitorovať či diagnostikovať chybu na zariadení.

Celkové riešenie bolo navrhnuté a realizované, tak aby bolo možné vykonávať zmeny a celý systém rozšíriť o nové možnosti, ktorými môžu byť napríklad moduly pre dataloging, či iné v praxi využiteľné moduly.

Literatúra

- [1] EWON: machines can talk [online]. [2001] [cit. 2014-11-10]. Dostupné z: <http://www.ewon.biz/en/home.html>
- [2] *1 Solving the Firewall/NAT Traversal Issue of SIP: Who Should Control Your Security Infrastructure?* [online]. [2006][cit. 2015-03-30]. Dostupné z: https://www.ingate.com/files/Solving_Firewall-NAT_Traversal.pdf
- [3] NAT traversal. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2014-10-15]. Dostupné z: http://en.wikipedia.org/wiki/NAT_traversal
- [4] *USB/IP Project* [online]. 2009- [cit. 2015-03-09]. Dostupné z: <http://usbip.sourceforge.net/>
- [5] WEI, Yuan, Suguru YOSHIDA, Daisuke YAMADA a Shigeki GOTO. A New Method for Symmetric NAT Traversal in UDP and TCP. In: GOTO Laboratory [online]. 2008 [cit. 2014-11-05]. Dostupné z: <http://www.goto.info.waseda.ac.jp/~wei/file/wei-apan-v10.pdf>
- [6] Introduction to Open VPN. In: OpenVPN - The Open Source VPN [online]. 2004 [cit. 2014-12-08]. Dostupné z: <http://openvpn.net/papers/openvpn-101.pdf>
- [7] Virtual private network. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-03-16]. Dostupné z: http://en.wikipedia.org/wiki/Virtual_private_network
- [8] FORD, Bryan, Pyda SRISURESH a Dan KEGEL. *Peer-to-Peer Communication Across Network Address Translators* [online]. 17.2.2005[cit. 2015-03-03]. Dostupné z: <http://www.brynosaurus.com/pub/net/p2pnat/>
- [9] Raspberry Pi. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-03-09]. Dostupné z: http://en.wikipedia.org/wiki/Raspberry_Pi
- [10] MARIADB FOUNDATION. *Welcome to MariaDB! - MariaDB* [online]. © 2015 [cit. 2015-03-09]. Dostupné z: <https://mariadb.org/>
- [11] Co je systém SQL Server 2012–2014 | Microsoft. MICROSOFT. *Oficiální domovská stránka Microsoft* [online]. © 2015 [cit. 2015-03-09]. Dostupné z: <http://www.microsoft.com/cs-cz/server-cloud/products/sql-server/>
- [12] RASPBERRY PI FOUNDATION. *Raspberry Pi* [online]. [2009-] [cit. 2015-03-16]. Dostupné z: <http://www.raspberrypi.org>

- [13] Diffie–Hellman key exchange. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-03-17]. Dostupné z: http://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange
- [14] Advanced Encryption Standard. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-03-18]. Dostupné z: http://cs.wikipedia.org/wiki/Advanced_Encryption_Standard
- [15] BURDA, K. *Bezpečnost informačních systémů*. Brno: VUT v Brně, 2013. s. 1-152. ISBN: 978-80-214-4890-2.
- [16] Man-in-the-middle attack. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-03-30]. Dostupné z: http://en.wikipedia.org/wiki/Man-in-the-middle_attack
- [17] Certifikační autorita. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-03-30]. Dostupné z: http://cs.wikipedia.org/wiki/Certifika%C4%8Dn%C3%AD_autorita
- [18] Androidmarket. *Srovnání androidích minipočítačů* [online]. © 2015 [cit. 2015-03-30]. Dostupné z: <http://androidmarket.cz/android/minipc/>
- [19] WAGO-I/O-SYSTEM 750: *Programmable Fieldbus Controller ETHERNET 750-881* [online]. 2012, 402 s. [cit. 2015-04-14]. Dostupné z: http://global.wago.com/media/2_products/m07500881_00000000_0en.pdf

10 ZOZNAM SKRATIEK

PLC	Programovateľný logický automat
UPnP	Universal Plug and Play
SQL	Structured Query Language
VoIP	Voice over Internet Protocol
P2P	Peer-to-peer
AES	Advanced Encryption Standard
HMI	Human Machine Intefrafe

11 ZOZNAM PRÍLOH NA CD

Zdrojové kódy pre dynamickú knižnicu

Program užívateľskej aplikácie

Zdrojové kódy pre program v Raspberry-Pi

Ukážkový program pre PLC WAGO

Monitorovací program stavov (Modbus TCP)

Program pre vizualizáciu a riadenie laboratórneho modelu vrtačky

Program pre PLC obsluhujúci laboratórny model vrtačky.