

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

## SYSTÉM PRO KONTROLU PŘÍSTUPOVÝCH BODŮ

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

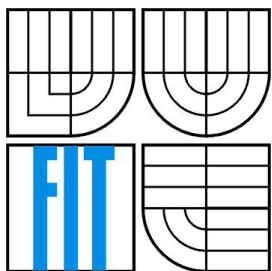
AUTOR PRÁCE  
AUTHOR

DANIEL STODŮLKA

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ  
FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# SYSTÉM PRO KONTROLU PŘÍSTUPOVÝCH BODŮ

SYSTEM FOR SUPERVISION OF ACCESS POINTS

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

DANIEL STODŮLKA

VEDOUCÍ PRÁCE  
SUPERVISOR

ING. LUKÁŠ MARŠÍK

BRNO 2015

## **Abstrakt**

Úkolem této práce je navrhnout a implementovat systém pro kontrolu přístupových bodů a zajistit jeho vzdálenou správu. Celý systém se skládá ze tří částí – řídicího programu, webového klienta a server systému sdružujícího všechna zařízení. Systém má předcházet možným výpadkům připojení k Internetu u koncových zákazníků, pomocí kontroly napětí na záložních bateriích a ochlazení interiéru v případě vysokých teplot. Systém na tyto události reaguje a uvědomí poskytovatele prostřednictvím emailu. K dispozici je také přehled měřených hodnot a ovládání zařízení prostřednictvím webového klienta. Jako monitorovací zařízení byl vybrán výrobek SDS MICRO.

## **Abstract**

The goal of this thesis is to design and implement system for supervision of accesspoints and also implement remote control system. The system consists of three sections – control program, web client and server system which gives an overview of all connected monitoring devices. The purpose of this system is to prevent connection outage by constant backup battery voltage checking and temperature control. The system will also warn provider by sending an email if these values go over certain limits. Overview of readings is also available in the form of web client which displays all necessary data. The monitoring device can also be controlled via the client. As a monitoring device the SDS MICRO product was chosen.

## **Klíčová slova**

SDS MICRO, PLC, SDS-C, Foxtrot PLC, JavaScript, PHP, JpGraph, přístupový bod, monitorovací zařízení

## **Keywords**

SDS MICRO, PLC, SDS-C , Foxtrot PLC, JavaScript, PHP, JpGraph, access point, monitoring device

## **Citace**

Daniel Stodůlka: Systém pro kontrolu přístupových bodů v českém jazyce, bakalářská práce, Brno, FIT VUT v Brně, 2015

# System pro kontrolu přístupových bodů

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Lukáše Maršíka. Další informace mi poskytl Ing. Tomáš Kratochvíl. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Daniel Stodůlka  
18. května 2015

## Poděkování

Rád bych poděkoval svému vedoucímu Ing. Lukáši Maršíkovi za odborné vedení a náměty k prezentaci. Dále také Ing. Tomáši Kratochvílovi za poskytnutí potřebného vybavení a místa pro testování systému.

© Daniel Stodůlka, 2015

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

Obsah.....	1
1 Úvod.....	2
2 Použité technologie.....	3
2.1 Jazyk SDS-C.....	3
2.2 Jazyk HTML.....	5
2.3 Jazyk CSS.....	6
2.4 Jazyk JavaScript.....	6
2.5 Jazyk PHP.....	7
2.6 Jazyk MySQL.....	8
3 Návrh.....	8
3.1 Návrh monitorovacího zařízení.....	9
3.1.1 Alternativní řešení.....	9
3.1.2 SDS MICRO.....	12
3.2 Návrh řídicího programu.....	14
3.3 Návrh klienta.....	15
3.4 Návrh server systému.....	15
4 Implementace systému.....	16
4.1 Implementace řídicího programu.....	16
4.2 Inicializace programu.....	16
4.3 Řídicí smyčka.....	16
4.2 Implementace klienta.....	18
4.2.1 Sběr a prezentace dat.....	18
4.2.2 Manipulace s aktivními prvky.....	19
4.3 Implementace server systému.....	20
5 Konečná instalace.....	23
6 Testování výkonnosti a spolehlivosti.....	24
6.1 Testování.....	24
6.2 Požadavky na instalaci.....	24
7 Závěr.....	25
Literatura.....	27
Seznam příloh.....	28
Přílohy.....	28
Příloha A – Obrazovky systému.....	28
Příloha B – CD se zdrojovými soubory.....	29

# 1 Úvod

Softwarových řešení monitorujících tok sítě a hlídajících připojené klienty je již v dnešní době hodně a jsou velice rozšířené, ovšem možností sledování samotného přístupového bodu není tolik. Navíc chceme-li sledovat několik bodů najednou, docílit určité schopnosti komunikace s okolím a mít možnost přizpůsobit si systém vlastním požadavkům nezávisle na jeho typu a značce, dostáváme se do podstatně těžší situace. Jako alternativní řešení jsem vybral jedno jiné PLC zařízení se srovnatelnou funkčností.

Samotný systém reaguje na různé podněty a poté uvědomí poskytovatele pomocí e-mailu. Praktické nasazení se předpokládá ve špatně přístupných a méně frekventovaných lokalitách, kde výpadek proudu nebo extrémní teploty mohou způsobit výpadek komunikace pro stovky až tisíce lidí. Systém tomuto předchází např. při výpadku napětí, kdy varuje, že záložní zdroje již budou brzy bez energie a je tak možné je přijet v předstihu vyměnit nebo provést jiné opatření k zajištění bezproblémového chodu sítě a koncový uživatel tak nic nepozná a nebude omezován.

Druhá kapitola zahrnuje výčet a popis jednotlivých použitých technologií, některé jejich specifikace a něco z jejich historie. Ukazuje také, jak jsou v práci použity. Představuje také programovací jazyk monitorovacího zařízení.

Třetí kapitola pojednává o návrhu celkového systému a také představí monitorovací zařízení samotné tj. SDS MICRO. Je zde ukázáno také jeho alternativní řešení. Oba výrobky jsou porovnány formou výhod a nevýhod. Na konci této podkapitoly se nachází jejich zhodnocení. Dále jsou zde probrány návrhy všech částí – řídicího programu monitorovacího zařízení, webového řešení klienta a server systému.

Čtvrtá kapitola popisuje implementaci řídicího programu monitorovacího zařízení, klienta a server systému. Jsou zde vysvětleny funkce jednotlivých částí systému a také rozebrány některé důležité funkce. Tato kapitola ukazuje například způsob získávání dat a manipulaci s aktivními prvky zařízení.

Pátá kapitola ukazuje konečnou instalaci monitorovacího zařízení v praxi a vysvětluje jednotlivé komponenty a jejich funkci.

Šestá kapitola se zabývá testováním celého systému, jeho výkonnosti a spolehlivosti. Vyhodnocuje také použití systému v praxi. Dále obsahuje instrukce pro instalaci a uvedení systému do provozu. Jsou zde také zmíněny nutné požadavky pro jeho funkčnost.

V závěru je diskutováno zhodnocení systému a také návrh možného pokračování či rozšíření monitorovacího systému.

## 2 Použité technologie

Výsledná aplikace se skládá z více částí, pro každou její část byla použita jiná technologie resp. jazyk, nebo jejich kombinace. Následuje stručné seznámení s jednotlivými technologiemi a popis, kde a za jakým účelem se v systému využijí.

### 2.1 Jazyk SDS-C

Jazyk SDS-C je upravený/zjednodušený jazyk C specifický pro výrobky SDS ( např. MICRO LIGHT, MICRO DIN, MACRO ), novější verze již podporují klasický jazyk C. Jazyk C vznikl v roce 1972. Jeho autorem je Dennis Ritchie ve spolupráci s Bell Labs. Jazyk se řadí mezi imperativní a strukturální. Jazyk C stále patří mezi velmi populární a rozšířené jazyky.

Jelikož první výrobky byly omezeny pamětí, kam se přeložený program musel uložit, bylo třeba přistoupit na tuto alternativu a jazyk C zjednodušit. Pro práci se zařízením SDS MICRO bylo nutné přidat některé prvky a funkce, a to zejména systémové pole `sys`, které má přímou návaznost na hardware. Každý jeho prvek obsahuje informaci o zařízení samotném a také funkce pro práci s tímto polem. Tyto proměnné jsou použity primárně k ovládání a komunikaci se zařízením. Skrz ně je možné data číst, některá i nastavovat a tím přímo ovlivnit chod a činnost hardwaru. Na prvky je možné přistupovat přímo z SDS-C programu nebo také z webového rozhraní pomocí HTTP požadavků.

V rámci programu se pak mimo uživatelem vytvořených proměnných pracuje výhradně se třemi poli a to:

- **sys** – Pole systémových proměnných obsahující informace o zařízení a také sloužící jako nástroj pro komunikaci s ním. Pole o velikosti 1024 prvků.
- **text** - Díky tomuto poli je možné v omezení míře pracovat s textem, hlavní využití je při komunikaci přes sériovou linku a odesílání informací pomocí emailu.
- **ram** – Samostatné nezávislé pole sloužící jako alternativa pro uživatelsky vytvořená pole. Obsah pole je stále uložen pouze v dočasné paměti. Jeho velikost závisí na verzi použitého zařízení (nejmenší velikost je 128 prvků u verze č. 03).

Ve výchozím nastavení zařízení je každou milisekundu proveden jeden celý příkaz programu (nikoliv instrukce ale celý řádkový příkaz). Je také možné nastavit rychlost provádění příkazů programu. Toto je možné nastavením systémové proměnné `sys[63]`. K dispozici jsou dva stavy a to již zmiňovaný jeden příkaz za milisekundu nebo je toto omezení odstraněno a příkazy jsou prováděny s co největší rychlostí. V tomto případě je pak nežádoucí používat čekací funkce `wait`, kdy by čas v jejím parametru byl nepodstatný. K čekání je pak nutné využít systémové proměnné `sys[64]`, která slouží k časování a obsahuje vteřinový čítač.

Postupem času se výrobky zdokonalovaly a některé omezení se začaly odstraňovat až nakonec nynější verze (průběh roku 2015) budou podporovat jazyk C (FULL-C) bez omezení jako takových. Budou ovšem chybět některé standardní funkce, které budou postupem času doplňovány. K

dispozici bude také pár funkcí specifických pro zařízení SDS. V jazyce FULL-C se již nebude pracovat s přímým přístupem k systémovému poli, ale budou využity funkce get a set.

Pro nahrání/naprogramování zařízení je potřebný editor/kompilátor, ve kterém po přihlášení máme možnost program zkompilovat a poté nahrát do monitorovacího zařízení. Po nahrání se program okamžitě spustí. V tomto programu je také možné provést stáhnutí řídicího programu ze zařízení a slouží i k aktualizaci firmwaru.

#### **Některé specifikace jazyka SDS-C**

- Použití pouze globálních proměnných
- Pouze void funkce tj. funkce bez návratové hodnoty (využití globálních proměnných)
- Funkce bez parametrů (opět lze řešit pomocí globálních proměnných)
- Práce pouze se staticky alokovanou pamětí
- Nejsou k dispozici cykly for a while (řešeno pomocí skoků goto a návěstí)
- Pole nejsou povoleny až na pole systémových proměnných (sys,ram,text)
- Jako uživatelské pole je možné využít systémové pole ram
- Textový řetězec pouze jako konstanta nebo znaky v poli text
- Všechna čísla jsou 32-bit signed integer, nedá se přímo pracovat s desetinnými čísly, kvůli tomuto je například teplota uvedena jako čtyřmístné číslo, aby se zachovala desetinná přesnost [1][2][3][4]

V jazyce SDS-C bude napsán řídicí program monitorovacího zařízení, který bude obstarávat kontrolu námi požadovaných údajů a také příslušně reagovat na vzniklé události.



## 2.2 Jazyk HTML

Jazyk HTML ( HyperText Markup Language ) je značkovací jazyk používaný pro tvorbu webových stránek. Je to jeden z hlavních a nejpoužívanějších jazyků pro tvorbu stránek v systému WWW.

Pro potřeby jednoduché tvorby dokumentů (dříve Tex, PostScript, SGML) byl roku 1990 Tim Berners-Leem navržen jazyk HTML a také protokol HTTP, zároveň také napsal první webový prohlížeč WorldWideWeb. V prvním standardu z roku 1991 je uvedeno 18 elementů utvářející jednoduchý návrh jazyka HTML. Tyto elementy byly výrazně ovlivněny jazykem SGML.

S rychlým rozvojem webu se také rychle rozvíjel jazyk sám a bylo proto třeba definovat standardy.

- Verze 0.9 – 1.2 – Původní verze z roku 1991 do 1993. Roku 1993 také vznikl první prohlížeč s grafickým uživatelským rozhraním – Mosaic.
- Verze 2.0 – Vydána roku 1995 organizací IETF, tato specifikace měla být použita jako standard, které by se budoucí implementace měly držet.
- Verze 3.2 – Vydána roku 1997 komunitou W3C, přidává tabulky a některé stylové prvky.
- Verze 4.0 – Vydána ke konci roku 1997 a nabízí tři varianty strict, transitional a frameset.
- dasfasdf
- HTML 5 – Po pozastavení vývoje v roce 2000 kvůli práci na XHTML, byl roku 2014 vydán nový standard, přidává nové sémantické prvky např. elementy pro video, audio.

### Některé specifikace jazyka HTML

- HTML elementy jsou psány ve formě značek např. <html>, některé jsou párové nebo nepárové
- Prohlížeče nezobrazují přímo kód, ale využívají jej pro interpretaci obsahu stránky [5]

Jazyk HTML je použit pro tvorbu jak klientské stránky tak server systému.

## 2.3 Jazyk CSS

CSS je jazyk pro popis vzhledu a formátování elementů na stránkách napsaných jazyky HTML, XHTML, XML. Hlavní myšlenkou je oddělit pomocí CSS vzhled dokumentu od jeho struktury a obsahu.

První verze CSS1 byla vydána roku 1996, zavedla dosud používanou syntaxi. V počátku byly problémy s korektním zobrazením těchto stylů, kdy trvalo více než tři roky než první webový prohlížeč byl schopen implementovat celou jejich specifikaci (Internet Explorer roku 2000).

CSS 2 vydána roku 1998 a přináší nové prvky např. absolutní, relativní a fixní pozicování, z-index.

CSS 2.1 Po mnoha změnách a opravách chyb byl tento standard vydán v roce 2011.

CSS 3 Oproti CSS 2 je CSS 3 rozděleno do mnoha modulů, které mají různé data vydání.

### Některé specifikace jazyka CSS

- Definice stylů se skládá z pravidel
- Každé pravidlo obsahuje selektor a blok deklarácí [6]

Veškerý vzhled stránek v systému je nastýlován pomocí CSS.

## 2.4 Jazyk JavaScript

JavaScript je multiplatformní, objektový jazyk, který v roce 1995 vyvinul Brendan Eich. Jeho primární použití je určeno ke změně obsahu nebo vzhledu WWW stránek (po jejich stažení) bez nutnosti komunikace se serverem, funguje tedy na straně klienta.

Syntaxe je velmi podobná jazyku C/C++ nebo Java. Jak by se na první pohled mohlo zdát, tak JavaScript nemá s Javou kromě shodnosti v názvu nic společného. Slovíčko Java bylo použito čistě z komerčních účelů.

### Některé specifikace jazyka JavaScript

- Ačkoli je možné jej použít na straně serveru, využívá se v mnohem větší míře na straně klienta tj. přímo v prohlížeči
- Typicky nemůže pracovat se soubory
- Rozšíření také mimo web např. v PDF dokumentech [7]

JavaScript bude v práci sloužit ke sběru dat z monitorovacího zařízení a jejich zobrazení na kontrolní stránce klienta. Také bude využíván k ovládání aktivních prvků zařízení.

## 2.5 Jazyk PHP

PHP (Hypertextový preprocesor) je skriptovací jazyk využíván z větší části k programování dynamických internetových stránek a webových aplikací. Veřejně jej vydal v roce 1995 Rasmus Lerdorf. V počátku se jednalo pouze o skripty určené k údržbě osobní domovské stránky ( Personal Home Page), jako například zjistit počet návštěvníků. Po spojení těchto skriptů s FI ( form interpreter) vznikl silnější nástroj, který navíc už mohl komunikovat s databázemi a byl později vydán jako první veřejná verze - PHP 2.

Roku 1998 přišlo uvolnění verze PHP 3 s řadou vylepšení a změnou názvu na rekurzivní zkratku PHP = PHP:Hypertext Preprocesor. V roce 2000 byla vydána nová verze PHP 4, která je již postavená na Zend Enginu. Zatím poslední verze PHP 5 byla vydána v roce 2004 a je postavena na Zend Enginu II, ta přináší různá vylepšení pro objektově orientované programování a práci s databázemi. Nová verze ponese překvapivě název PHP 7 jež byl rozhodnut v hlasování.

Pro práci s nasbíranými daty je použita knihovna jgraph pro tvorbu grafů, je to objektově orientovaná knihovna, napsaná čistě v PHP ve verzi 5.1 – 5.5 .

### Některé specifikace jazyka PHP

- Program prováděn na straně serveru (je k dispozici i kompilovaná forma jazyka)
- Syntaxe jazyka je inspirována několika jazyky (Perl, C, Pascal, Java)
- Je nezávislý na platformě
- Podpora mnoha knihoven pro různé účely (práce s grafikou, databázemi, práce se soubory)
- Slabá podpora UNICODE [8]

Server systém je postaven na PHP a prezentace dat pomocí JpGraph využívá pouze PHP.

## 2.6 Databáze MySQL

MySQL je v dnešní době jeden z nejrošířenějších databázových systémů. Jeho první verze byla představena roku 1995, tato byla původně vytvořena pro osobní použití a vycházela ze staršího databázového systému mSQL. MySQL bylo vlastněno švédskou firmou MySQL AB, která je nyní již součástí Oracle Corporation.

MySQL je multiplatformní a komunikace mezi klienty a serverem probíhá pomocí jazyka SQL. Systém je soustředěný spíše na rychlost a výkon, proto v minulosti některé funkce chyběly např. trigger, pohled. MySQL využívají i některé velké projekty jako např. Google, Facebook, Youtube.

### Některé specifikace MySQL

- Podpora SSL
- Podpora UNICODE
- Nedodrží celý standard SQL
- Full-textové indexování a vyhledávání
- Možnost uložení procedur [9]

MySQL je použito jako systém pro ukládání/uchovávání nasbíraných dat. Je vytvořena jedna databáze pro ukládání teplotních údajů.

## 3 Návrh

Požadované funkce na systém jsou reakce na okolní události, sběr dat, jejich prezentace a také centralizovaná správa všech dostupných a pracujících monitorovacích zařízení. Každé monitorovací zařízení bude schopno díky více vstupům a výstupům sledovat více bodů. Kvůli technologickým omezením je nutno systém rozdělit do několika částí.

- Monitorovací zařízení – Obsahuje řídicí program, který reaguje na okolní vlivy a je srdcem celého systému.
- Webový klient – Zahrnuje zobrazení naměřených dat, umožňuje nastavení limitů pro jednotlivá čidla a také základní ovládání monitorovacího zařízení.
- Server systém – Poskytuje přehled nad všemi připojenými zařízeními, ukazuje jejich stav a odkazuje na jejich klientské stránky. Umožňuje také zobrazení naměřených hodnot pomocí grafů.

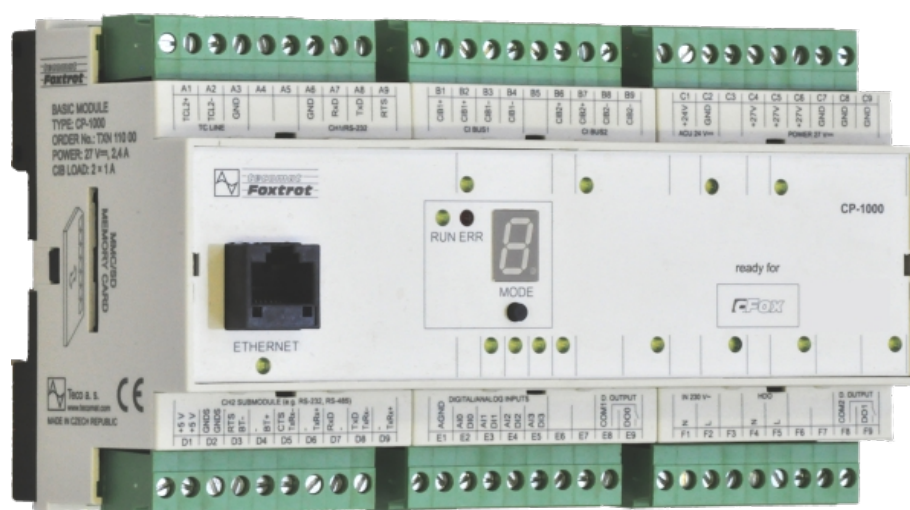
## 3.1 Návrh monitorovacího zařízení

### 3.1.1 Alternativní řešení

Jako alternativní řešení jsem vybral jedno PLC, výběr je poněkud složitější jde-li o celkovou funkčnost systému jako takového. Jelikož je monitorovací systém do značné míry upraven podle našich specifických požadavků, tyto konkurenční výrobky tedy spíše reprezentují náhražku samotného monitorovacího zařízení SDS MICRO. Tato zařízení by tedy stejně bylo nutné naprogramovat na námi požadovanou funkčnost. Některá požadovaná čidla, vstupy, výstupy také nejsou přítomny na všech zařízeních a komunikace přes Ethernet a možnost tvoření uživatelských stránek také ještě není standardem u těchto automatizovaných PLC. Jako alternativu a také pro některé přívětivější verzi jsem tedy vybral PLC Foxtrot.

### PLC Foxtrot

Malý kompaktní automat s možností modulárního rozšíření. Deska a její komponenty jsou na rozdíl od modulu SDS MICRO dodávány již v ochranném pouzdře. Komunikace probíhá pomocí Ethernetu a je také k dispozici sériové rozhraní.



Obrázek 1: Alternativní řešení - PLC Foxtrot

## Přehled vlastností

- Ethernet 100MBit/sec
- Čtyři analogové vstupy, na nich je také možné realizovat teplotní snímače
- Dva reléové výstupy
- Slot pro paměťové karty SDHC / SD / MMC
- Dva binární vstupy
- Možnost osazení přídatných submodulů
- Možnost měnit PLC program za jeho chodu
- Uživatelský projekt je možné archivovat přímo do zařízení. Archivace a obnovení probíhá v prostředí Mosaic
- Obsahuje vlastní webserver, pro jeho funkčnost je ale nutné mít připojenou paměťovou kartu, na niž se stránky ukládají
- Výrobek českého původu
- Vzdálená správa pomocí TecoRoute

Programování PLC probíhá ve vývojovém prostředí Mosaic. Prostředí slouží jako textový editor, překladač, debugger, modul pro komunikaci s PLC a v neposlední řadě také jako simulátor PLC. Systém lze programovat v jazycích ST, IL, LD, FBD.

Tvorba stránek pro interní webserver probíhá pomocí nástroje WebMaker, který je dostupný přímo v prostředí Mosaic. Tvorba webového rozhraní je založena na XML, kdy se na stránky přidávají nejrůznější objekty a pomocí formulářů se mění jejich vlastnosti. Např. blikání LED podle stavu bool proměnné v programu PLC apod. Odtud je možné také měnit přímo rozměry stránek a také práv pro jejich zobrazení (různé zobrazení pro různé uživatele). Pro všechny tyto objekty je vygenerován XML kód, který je zařízení schopno interpretovat jako webové stránky. Prostředí Mosaic také nabízí možnost simulace námi vytvořených stránek. V této simulaci se také pracuje přímo s PLC programem, výsledek je tedy stejný jako při práci z webového prohlížeče.

Jako něco navíc je u PLC Foxtrot dostupná služba TecoRoute, tato služba, ačkoli placená, slouží ke vzdálené správě PLC zařízení. PLC se po zapojení a nastavení příslušných údajů samo přihlásí na server, odkud je možné jej vzdáleně spravovat. V tomto případě také PLC nemusí mít veřejnou IP adresu a stačí, pokud je umístěno pouze v lokální síti.

### **Výhody oproti SDS MICRO**

- Jednodušší tvorba klientských stránek
- Více než jedna klientská stránka
- Přímochařejší postup vytváření klientské stránky
- Rozsáhlejší a celkově lepší dokumentace
- Možnost změny PLC programu za jeho běhu (toto je částečně možné i pomocí SDS MICRO díky změně obsahu uživatelských proměnných)
- Rychlejší zpracování instrukcí v řídicím programu
- PLC Foxtrot poskytuje lepší možnost rozšíření přídatnými moduly
- Ačkoli placená, ale možná vzdálená správa zařízení (opět u SDS MICRO částečně možné pomocí uživatelských proměnných)
- Simulace webových stránek přímo ve vývojovém prostředí Mosaic

### **Nevýhody oproti SDS MICRO**

- Foxtrot PLC se pohybuje v mnohonásobně vyšší cenové kategorii (10 000 Kč a výše)
- Složitější proces programování řídicího programu
- Vlastní konfigurace webových stránek pouze do určité míry
- Vzhled stránek je omezený pouze tvorbou v prostředí WebMaker [10][11][12]

### **Hodnocení**

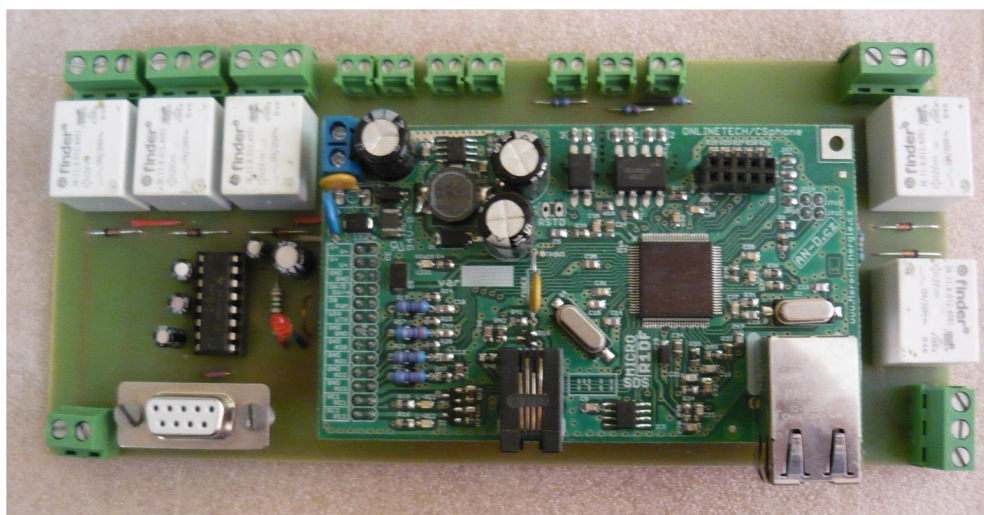
Foxtrot PLC je zařízení poměrně jiné cenové kategorie, tomu také odpovídá jeho výbava a zázemí produktu. Nicméně pro námi navrhovaný systém toto zařízení nepřináší nic nového co by SDS MICRO nezvládlo. Jako malé plus se samozřejmě jeví tvorba klientských stránek, kdy není potřeba pomocí JavaScriptu číst data ze zařízení a celé prostředí WebMakeru působí přívětivěji. Na druhou stranu ztrácí při tvorbě řídicího programu kdy jazyk SDS-C je velice jednoduchý a přímočarý. Také postrádá některé nestandardní funkce pro ostatní PLC ( odeslání emailu, ping apod. ).

Foxtrot PLC se jeví jako lepší variantu pokud uvažujeme o větší investici a řízení složitějších projektů (např. inteligentní dům spolu s řízením vytápěním, hlídáním spotřeby apod.).

### 3.1.2 SDS MICRO

Jako monitorovací zařízení jsem vybral výrobek řady SDS MICRO. K dispozici jsou také ostatní řady MACRO, TTCPRO. SDS MICRO se jeví jako cenově přijatelnější varianta oproti ostatním řadám SDS, kdy poskytuje všechny potřebné funkce a vybavení pro implementaci systému. Mezi hlavní přednosti tohoto řešení řadím :

- Dostupnost
- Cena
- Rychle se rozvíjející komunita
- Pravidelné aktualizace firmwaru a přidávání nových funkcí
- Interní webový server
- Možnost konfigurace vlastní klientské stránky
- Funguje bez nutnosti připojeného PC
- Nízká spotřeba
- Výrobek českého původu
- Poměrně jednoduché programování PLC programu pomocí SDS-C
- Univerzální program pro editaci a nahrávání programu a také pro update firmwaru



Obrázek 2: SDS MICRO R1DF



SDS MICRO je univerzální typ monitorovacího/programovatelného a řídicího zařízení, je dodáváno jako samostatná deska (k dispozici také DIN verze dodávána s krytem) , čímž umožňuje jednoduché a rychlé zapojení. Je možné jej použít pro měření teploty, optické vstupy pro měření odběru elektřiny, plynu, vody, jako IP/DNS/WWW watchdog, dále je možné spínat, jak integrované, tak přídatné relé a to buď pomocí SDS-C programu nebo přímo z webového rozhraní.

#### **Některé hlavní vlastnosti:**

- Vícenásobný webový teploměr (je možné připojit více čidel)
- Schopnost komunikovat s ostatními SDS zařízeními
- Komunikace pomocí Ethernet 100MBit/sec
- Umí pracovat s protokoly HTTP, SNMP
- Ethernet to serial konvertor
- Zařízení disponuje optickými vstupy
- Výstup pro externí relé
- Integrované relé
- Čidlo vlhkosti
- Analogové vstupy
- Programování řídicího programu PLC pomocí jazyka SDS-C

Zařízení poskytuje v základní verzi již všechny potřebné vstupy a výstupy a není jej třeba na další náklady rozšiřovat. Využity budou analogové vstupy pro měření napětí, webový teploměr a relé pro spínání větráku. Relé se zde chová jako nezávislý prvek a monitorovací zařízení tedy slouží pouze jako způsob jeho spínání či vypínání. Pro jeho plnou funkčnost tj. v tomto případě zapnutí větráčku je nutný externí zdroj napětí. Což se jeví jako potencionální nedostatek, kdy deska samotná není schopna napájet své výstupy. Ovšem relé je schopno dodávat mnohem větší napětí než zdroj desky samotné, což je vlastně záměr aby, bylo možné relé použít pro řízení mnohem náročnějších zařízení. Nicméně externí relé je možné propojit se zdrojem zařízení, a tím dodávat napětí alespoň 12V.

Využití opto vstupů bude diskutováno dále v možném rozšíření či vylepšení této práce.

Po zapnutí zařízení je nutné změnit MAC adresu a případně nastavit IP adresu. To se provede ve výchozím webovém rozhraní výrobku, které je závislé na firmwaru. Toto rozhraní ovšem není možné měnit uživatelem.

Programování PLC probíhá pomocí editoru/kompilátoru, ve kterém je nutné program nejdříve přeložit a poté do zařízení nahrát. Stejným způsobem se také nahrává uživatelská stránka. Zařízení jsou také schopna komunikovat mezi sebou a to pomocí odeslání HTTP požadavku přímo z řídicího programu, který změní hodnotu některé sdílené proměnné jiného SDS zařízení.

Nutno podotknout, že při nahrávání programu nebo klientské stránky se Ethernetové spojení někdy ukázalo jako nestabilní a proto by bylo lepší pro tyto operace použít spojení seriové. Dalším menším problémem je neucelená dokumentace a pro některé funkce zařízení zcela chybějící.

Možnou alternativou by byl také výrobek stejné řady a to SDS MACRO, který nabízí stejné funkce jako MICRO verze, má však navíc větší množství vstupů a výstupů a zabudovaný termostat pro řízení vnějších zařízení podle teploty. Všechno pochopitelně za větší cenu. [1][3][12]

## 3.2 Návrh řídicího programu

Řídicí program je srdcem celého systému, zajišťuje reakce systému na okolní události. Jež je měření teploty, při měření teploty spíná větráček na dobu nezbytně nutnou pro ochlazení interiéru, měření napětí na záložních bateriích a také reaguje na změnu limitů pro oba sledované údaje z klientské stránky. Jako upozornění také v případě překročení limitů odešle email s informací o události.

Při návrhu se proto bylo potřeba zaměřit na některé specifické požadavky.

### Nutnost nepřetržitého chodu řídicího programu

- Program by měl být funkční 24 hodin denně, program tedy bude nutně pracovat ve smyčce po nastavení všech řídicích proměnných, neměl by také být paměťově náročný a provádět pouze nezbytně nutný počet instrukcí. Výhodou se zde jeví práce pouze ze staticky alokovanou paměti.
- Nenadálý výpadek proudu u kontrolního zařízení a následný restart řídicího programu by neměl mít vliv na jeho opětovný chod. Všechny proměnné a funkce by měly být funkční okamžitě po startu zařízení nikoliv po následné interakci s uživatelem.

### Sledování a reakce na události

- Řídicí program bude potřebná data odečítat ze vstupů, převádět je a ukládat do proměnných, které budou následně zobrazeny na klientské stránce
- Pro potřebu reakce na události je také třeba stanovit, kdy se na ni bude reagovat, to je řešeno ve formě limitů pro jednotlivé čidla, kdy při jejich překročení dojde k upozornění uživatele
- Upozornění uživatele bude probíhat formou odeslání emailu o nastanulé události díky zabudovaného webserveru. Tohle zvládne monitorovací zařízení samotné (ovládá SMTP protokol) a není třeba data dále zpracovávat a přeposílat.
- Základní limity pro čidla budou nastaveny přímo v řídicím programu na defaultní hodnotu. Ty pak bude možné dále měnit z webového rozhraní. Nastává problém uchování takových nově uživatelsky změněných limitů při restartu zařízení, kdy by data byla obvykle ztracena, nicméně zařízení SDS MICRO poskytuje paměťový prostor v podobě několika prvků systémového pole, které jsou uloženy do trvalé paměti a zařízení je schopno uchovávat nově nastavené limity i po jeho restartu.
- Nutná je taky určitá zpětná vazba v podobě zobrazení neočekávaných chyb či neúspěchu některých systémových funkcí. K dispozici bude tedy výpis z konzole programu.

### 3.3 Návrh klienta

Díky faktu, že monitorovací zařízení má přímo zabudovaný WWW server, můžeme kontrolní stránku provozovat přímo na něm a není nutné další externí zařízení nebo přeposílání všech dat na vzdálený server. Mohlo by se zdát, že tedy zařízení plní obě role najednou (monitoring i klientskou sekci) pro lepší rozlišení a vymezení funkčnosti jsem je ale v návrhu i implementaci pojmul odděleně. Klient pak bude odkazovat na souhrnný přehled všech monitorovacích zařízení tj. server systém. Kvůli možnosti manipulace s vnitřními proměnnými samotného řídicího programu (nastavení limit a názvů čidel) a také ručního spínání relé, bude tato stránka dostupná vždy až po přihlášení. Toto také platí pro případné samostatné odesílání požadavků.

Bohužel výrobek je technologicky omezen a je schopen uchovat pouze jednu uživatelsky upravenou stránku, bylo tedy nutné stránku navrhnout tak, aby splňovala tyto vlastnosti.

- Jednoduchost
- Přehlednost
- Stabilitu

A v neposlední řadě, nehledě na toto jednostránkové omezení, by měla být schopna prezentovat veškerá data v uživatelsky přívětivé podobě.

Sledovaná data tedy budou:

- Napětí na záložním zdroji
- Teplota na teplotním čidle
- Stav výstupů ( sepnuto/nesepnuto )
- Způsob ovládání výstupů ( ručně/automaticky )
- Výpis z programové konzole

Bude také třeba určitým způsobem nastavit limity pro sledované hodnoty. Jelikož není možné po startu zasahovat do běhu řídicího programu, budeme skrze klientskou stránku přistupovat na uživatelské proměnné. Tyto proměnné bude řídicí program kontrolovat a pokud dojde k jejich změně, změní také příslušné limity.

### 3.4 Návrh server systému

Jelikož monitorovacích zařízení bude nasazeno více a každé bude mít vlastní klientskou stránku, bylo třeba vytvořit jednoduchý systém, který by mezi nimi usnadňoval orientaci. Navíc se otevřela možnost uchovávat některá potenciálně zajímavá data přímo na serveru.

Server systém tedy plní úlohu centralizované správy všech bodů, poskytuje přehled nad všemi aktivní i neaktivními monitorovacími zařízeními a plní funkci pomyslného mostu mezi klientskými kontrolními stránkami.

Na hlavní stránce je uveden přehled všech monitorovacích zařízení a také ukázán jejich stav spolu s IP adresou, která odkazuje na příslušnou klientskou stránku. K dispozici je také odkaz na

statistiky ve formě grafů, které ukazují teplotu za dané období – je možné zvolit denní nebo měsíční přehled.

Sběr teplotních údajů provádí samostatný skript běžící na pozadí na serveru, ten periodicky každou hodinu (nebo jiný časový úsek, pokud je třeba) ukládá data do databáze.

## 4 Implementace systému

V této kapitole bude popsána implementace jednotlivých částí systému. Bude zde popsán běh a úkoly řídicího programu, způsob prezentace dat a manipulace s aktivními prvky na klientské stránce a také implementace server systému, sběru dat a prezentace dat v grafech.

### 4.1 Implementace řídicího programu

Řídicí program je implementován ve výše zmiňovaném jazyce SDS-C, využívá systémového pole `sys`, ostatní pole (`ram`, `text`) nejsou v tomto programu využity. Skládá se z inicializace a definice proměnných, funkcí, které reagují na události vzniklé v řídicí smyčce a řídicí smyčky samotné.

Celý řídicí program je pak možné rozdělit na dvě pomyslné části.

#### 4.1.1 Inicializace programu

Před spuštěním samotné řídicí smyčky, která nepřetržitě kontroluje vstupy a výstupy monitorovacího zařízení, je nutné nadefinovat a inicializovat některé důležité proměnné – stanovení limitů pro čidla ( získané z trvalé paměti `sys[100-103]`), získání systémového času (`sys[5-10]`), kontrolní proměnné pro odeslání emailu, časovače (`sys[64]`) a hystereze pro teplotní čidla. Po těchto prvotních úkonech následuje malá pauza ve vykonávání programu kvůli inicializaci a naběhnutí celého systému zařízení. Poté program pokračuje již do řídicí smyčky, kde kontroluje vstupy, získává a převádí data. Po doběhnutí na její konec se přesune pomocí příkazu `goto` na její začátek.

#### 4.1.2 Řídicí smyčka

Zajišťuje odečet hodnot ze všech vstupů a jejich následný přepočet. Na základě těchto hodnot pak zavolá příslušné funkce pro odeslání emailu, sepnutí relé apod. Kontroluje také limity čidel, které je možné nastavit z webového rozhraní. Program je pak automaticky zapíše do trvalé paměti zařízení. Činnost řídicí smyčky je možné rozdělit do těchto úloh.

### Teplotní údaje

Teplotní údaje jsou odečítány při každém průběhu řídicí smyčky. Jejich hodnota je pak ještě upravena, aby odpovídala zápisu ve stupních Celsia. Odečet probíhá přečtením prvku z pole systémových proměnných, pro první teplotní čidlo je hodnota uložena v `sys[310]`.

Pokud teplota překročí stanovenou mez, dojde k zavolání funkce, která na příslušnou emailovou adresu odešle zprávu o tom, na kterém čidle byla teplota překročena a také v jaký čas.

Po odeslání emailu je nutné stanovit, kdy se teplota opět zklidní a bude možné v případě dalšího jejího překročení odeslat email/sepnout relé znova tj. vrátí se na ustálenou hodnotu a také tím zamezí neustálému spínání resp. vypínání relé při kolísání teploty na hranici stanoveného limitu. V programu je proto použita hystereze a to v hodnotě 5 stupňů, kdy teplota musí klesnout alespoň o 5 stupňů oproti svému předchozímu limitu, aby bylo možné říci, že je teplota ustálená.

Poté nastává akce sepnutí relé napojeného na větráček, který má za úkol ochladit interiér a zabránit tak přehřátí/přehřívání přístupových bodů. Pro potřeby sepnutí relé je nutné nastavit způsob ovládání relé, a to v systémové proměnné `sys[196-xxx-` podle počtu relé]. Samotné sepnutí pak již probíhá podobně a to opět zápisem do proměnné `sys[231]`. Po jeho sepnutí je nutné způsob ovládání nastavit opět na ruční řízení, aby bylo možné relé sepnout/vypnout i z webového rozhraní. Jinak by měl řídicí program prioritu.

Po snížení teploty dojde k opětovnému sepnutí relé, v tomto případě k jeho vypnutí, tím je vypnut i větráček, jelikož je teplota považována za ustálenou. Jako v předchozím případě je nutné správně nastavit řízení relé. Po této události je také možné opět odeslat email.

## Napětí

Odečet napětí ze vstupního čidla probíhá podobným způsobem jako u teploty. Údaje jsou uloženy v systémové proměnné `sys[431-xxx]` podle počtu převodníků. Hodnotu je pak dále nutno upravit na hodnotu pro nás jednoduše čitelnou, a to volt.

Pokud napětí klesne pod stanovenou mez (vybití záložního zdroje energie) je podobně jako u překročení teploty odeslán email s časem, názvem čidla potažmo záložního zdroje a informační text.

V případě napětí není třeba žádné hystereze, jelikož vybití baterie je jev jednosměrný a nemělo by dojít k opětovnému odeslání emailu, pokud by baterie nebyla znova plně nabitá. Po nabití záložního zdroje (nad stanovený limit) řídicí program umožní opět odeslat email. Výpadky proudu nejsou tak časté, takže s největší pravděpodobností by nemělo dojít k jevu, že nastane další výpadek, aniž by záložní baterie byly plně nabity.

## Optické vstupy

Optické vstupy nejsou v monitorovacím zařízení prozatím využity. Je však naprogramována jejich obsluha, kdy při detekci napětí na čidlech, podobně jako při ostatních událostech odešlou email.

## Odesílání emailu

Samotné odeslání emailu zprostředkovává interní funkce zařízení `smtp_send`, obsahem emailu je předmět zprávy a samotný text. Po odeslání emailu resp. po pokusu o odeslání emailu je nutné zkontrolovat jestli, akce byla úspěšná, tato informace se nachází v prvku `sys[65]`, který značí, zda řádně proběhla některá funkce zajišťující komunikaci (HTTP požadavky, SMTP apod.). Ihned po odeslání emailu je také nutné vyčkat než proběhne komunikace mezi serverem a zařízením nebo nedojde k timeoutu, jelikož řídicí program provádí operace rychleji než probíhá komunikace, nezjistili bychom jestli byl email úspěšně odeslán.

V případě odeslání emailu, ať už úspěšného či neúspěšného, je na konzoli programu vypsaná informace o této události (výpis z konzole je pak přístupný přes klientskou stránku). Odesílání času by se mohlo zdát redundantní, ovšem v případě momentálním výpadku nebo zpoždění emailové služby by čas přijetí zprávy nemusel odpovídat času překročení teploty/snížení napětí.

## 4.2 Implementace klienta

Klientská stránka musí být kvůli technickému omezení monitorovacího zařízení jenom jedna, navíc jelikož na něj není možné ukládat další soubory, celý styl stránky je tedy definován přímo v ní a není použito externího CSS souboru. Co se vzhledu stránky týče, v tomto případě čím méně, znamená více. Stránka by měla mít co možno nejmenší velikost aby zbytečně nezatěžovala zařízení a měla by sloužit primárně k odečtu dat a ovládní monitorovacího zařízení. Podle návrhu můžeme implementaci klientské stránky rozdělit do dvou sekcí.

### 4.2.1 Sběr a prezentace dat

#### Získávání dat

Jak je z popisu implementačního jazyku SDS-C patrné, všechny informace a naměřené hodnoty jsou uloženy v poli systémových proměnných `sys`. Tyto proměnné je možné vyčíst dvěma způsoby, a to využitím JavaScriptu, přesněji pomocí HTTP požadavků na daný prvek systémového pole. Druhé řešení nabízí zařízení samotné a to ve formě odesílání dat třetí straně – např. na email. Toto řešení není příliš efektivní ani vhodné, data by se musela odesílat po celou dobu běhu programu v malých časových intervalech, což by zbytečně zatěžovalo provoz v síti.

Je tedy využít první způsob, kdy se každá informace zjistí pomocí HTTP požadavku ve tvaru `192.168.1.250/get_sys[x]` – IP adresa zařízení a prvek v poli systémových proměnných, kde například `sys[310]` značí teplotu na prvním teplotním čidle, `sys[232]` stav relé č.2 sepnuto/nesepnuto apod. Takovýchto požadavků by bylo třeba odeslat několik desítek každou vteřinu či dvě v závislosti na potřebě zobrazení aktuálních informací. Platí tedy to samé, co v předchozím případě a je nutné toto řešení nějakým způsobem vylepšit.

Konečné řešení využívá HTTP požadavku, ovšem místo dotazování se na jednotlivé hodnoty odešle se požadavek jen jeden – `192.168.1.250/cgi_dys?rand=x`, jež vrátí většinu požadovaných hodnot, které je už jen potřeba zpracovat a zobrazit. X v tomto požadavku značí náhodné číslo, bez jeho použití by zařízení vracelo stále stejné hodnoty. Nicméně druhý způsob samostatných GET požadavků bude použit pro hodnoty, které se tímto hromadným způsobem získat nedají ( např. konzolový výstup, sdílené proměnné ).

## Prezentace dat

Data jsou zobrazeny na kontrolní stránce v několika tabulkách jejichž obsah se dynamicky mění při každém získání nových dat. Aby se stránka nemusela neustále obnovovat, provádí celý sběr dat a jejich následné zobrazení jedna funkce, která je volána v intervalu jedné vteřiny místo aktualizace celé stránky.

Dále je také k dispozici výpis z konzole řídicího programu, který má za úkol informovat o jeho běhu – zobrazuje se zde init programu tj. jeho spuštění, úspěšně odeslaného emailu, po kliknutí na tlačítko také názvy čidel monitorovacího zařízení. Konzole slouží také k výpisu možných problémů při běhu programu, primárně při neúspěšném pokusu o odeslání emailu. Tento výpis se získá podobně jako ostatní data a to HTTP požadavkem s parametrem echo.

Je zde také možnost vymazání výpisu konzole. Tento krok nevymaže pouze pole na obrazovce, ale také ovlivňuje přímo konzoli monitorovacího zařízení. Při dalším zobrazení konzole nejsou uchovávány staré data, ale výpis je prázdný. Výpis z konzole není stěžejní, limit jeho obnovy je tedy 15 vteřin.

### 4.2.2 Manipulace s aktivními prvky

#### Aktivní prvky

Mimo prosté zobrazení naměřených údajů je také možné ovládat aktivní prvky přímo z webového rozhraní klientské stránky. Je možné ovládat relé výstupy.

##### Relé má tři možné způsoby ovládání

- Z webového rozhraní – stav trvale zapnuto/vypnuto
- Z řídicího programu – spíná se zápisem do systémové proměnné, má větší prioritu než nastavení z webového rozhraní
- Z IP watchodu – podobné způsobu ovládání řídicího programu

Sepnutí relé probíhá podobným způsobem jako sběr dat a to využitím HTTP požadavku ve tvaru `192.168.1.250/cgi_rtcl?rmode=cson?&rn=1`, kde mode značí sepnutí (cson), vypnutí (cson) a rn je číselné označení relé. Tímto se automaticky nastaví způsob ovládání relé a to na ovládání z webového rozhraní ( ruční ). V případě spínání z řídicího programu je nutné řízení nastavit zpět na automatické. Důležité je, že tímto se nedá implicitně změnit priorita řízení ( pokud SDS-C program nereaguje na nastavení některé uživatelské proměnné ) a relé, které má zůstat podle řídicího programu vypnuté tak při zapnutí z klientské stránky pouze sepne a okamžitě se vypne.

#### Uživatelské proměnné

Uživatelské proměnné jsou v této práci využity jako způsob nastavení limitů pro čidla. Jejich potenciální využití záleží čistě na programátorovi, kdy program může na tyto proměnné reagovat mnoha různými způsoby. Tyto proměnné je nutno měnit přímo z webového rozhraní a tím je možné určitým způsobem měnit chod řídicího programu bez nutnosti jeho pozastavení.

K dispozici je 8 až 10 sdílených proměnných - `sys[140]`-`sys[150]` (podle verze firmware). Tyto proměnné jsou bohužel uloženy pouze v dočasné paměti a všechna uživatelská data v nich jsou při restartu zařízení ztracena. Jsou tedy využity spíše jako prostředek pro komunikaci s řídicím programem.

Obečný princip komunikace se zařízením probíhá podobným způsobem jako při získávání dat a je opět založen na HTTP požadavku ze strany webového klienta a to ve tvaru – `192.168.1.250/sdscep?p=0&sys140=15`, kde `p` udává akci po nastavení proměnné v tomto případě zobrazení nově nastavené hodnoty, `sys[140]` určuje prvek v poli systémových proměnných, který má být nastaven na hodnotu 15.

Z webového rozhraní je možné změnit limity všech aktivních čidel, po odeslání formuláře se změni hodnota sdílené systémové proměnné (zatím pouze v dočasné paměti), řídicí program na tuto událost zareaguje a nový limit uloží již do trvalé paměti. Při restartu zařízení jsou již tedy k dispozici nové limity. Navíc tento způsob zápisu dat vyžaduje přihlášení jelikož je tímto možné pozměnit chod zařízení.

Poslední funkcí klientské stránky je změna názvů čidel, která probíhá podobně jako změna limitů čidel. Je odeslán HTTP požadavek s novým jménem. Rozlišení jmen čidel je obzvláště důležité při reakcích na události, kdy je třeba rozlišit které čidlo patří ke kterému sledovanému bodu.

## 4.3 Implementace server systému

Server systém sdružuje všechny monitorovací zařízení a poskytuje nad nimi přehled, také zajišťuje sběr dat a to konkrétně teplotních údajů a prezentuje je v jednoduchých grafech. Na hlavní stránce poskytuje přístup na klientské stránky jednotlivých monitorovacích zařízení.

### Sběr dat

Pro potřeby zobrazení teplotních údajů je třeba nějakým způsobem získat data z monitorovacího zařízení. Vybral jsem řešení pomocí samostatného skriptu běžícím nepřetržitě na serveru, který dotazuje potřebná data z čidel všech monitorovacích zařízení a v intervalu 1 hodiny je pak ukládá do databáze. Další možnou variantou by bylo, aby zařízení sama posílala data na server, který by je dále zpracovával. Na serveru by v tomto případě také musel existovat skript, který by data přijímal a po následném napojení na databáze je do ní ukládal.

Skript data získává pomocí funkce `curl`, která vyšle HTTP požadavek pro získání hodnoty příslušného prvku z pole systémových proměnných - `http://192.168.1.250/get_sys[310]` pro první teplotní čidlo. Podobně získá údaje i z ostatních čidel.

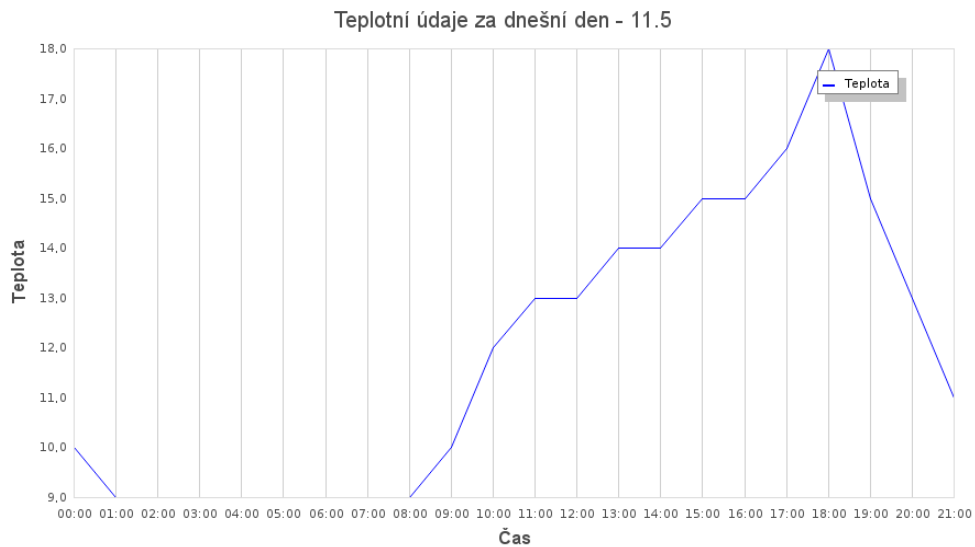
Data jsou uložena do jednoduché databáze o jedné tabulce reprezentující teplotu. Různá monitorovací zařízení, potažmo čidla, jsou rozlišeny pomocí jejich ID při zápisu do databáze.



## Zobrazení bodů a prezentace dat

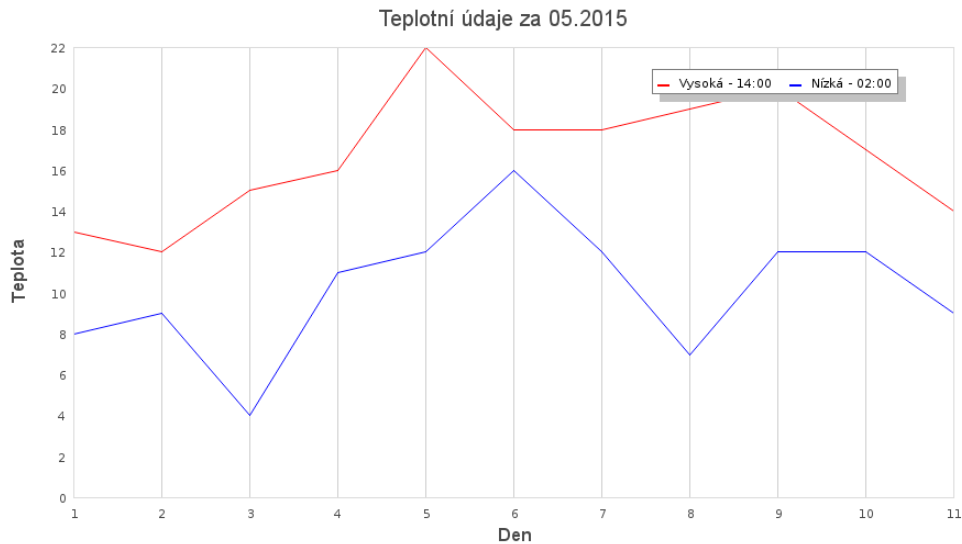
Prezentační část server systému se skládá ze dvou částí resp. stránek. První část zobrazuje všechny body napojení do tohoto systému, jejich stav a také adresu na které jsou přístupné. Stav je zjišťováním pomocí knihovni funkce PHP – `ping` s poměrně nízkým timeoutem a to 500ms, zvolen je kvůli rychlosti načítání stránky, kdy by s větším množstvím monitorovacích zařízení čas načítání neúměrně narůstal. Stav zařízení se obnovuje každých 30 vteřin.

Dále je k dispozici přehled nasbíraných teplotních údajů ve formě grafu. Prezentovány jsou dva grafy a to přehled teplot za poslední měsíc – nikoliv však posledních 30 dní, ale vždy přehled k prvnímu dni v měsíci a přehled teplot po celý den (opět nikoliv celých 24 hodin, ale započatý den od půlnoci).



Obrázek 3: Denní teplota

Teplotní údaje jsou zaznamenávány každou hodinu po celý den, výpis všech naměřených hodnot do grafu pro celý měsíc by tedy byl značně nepřehledný. Byly vybrány pouze dvě denní doby, které budou ve grafu znázorněny. A to časy s potenciálně nejvyšší resp. nejnižší denní teplotou. Pro nejvyšší teplotu byl použit čas 14:00 a pro teplotu nejnižší čas 02:00. V grafu jsou potom tyto teploty rozlišeny barvou.

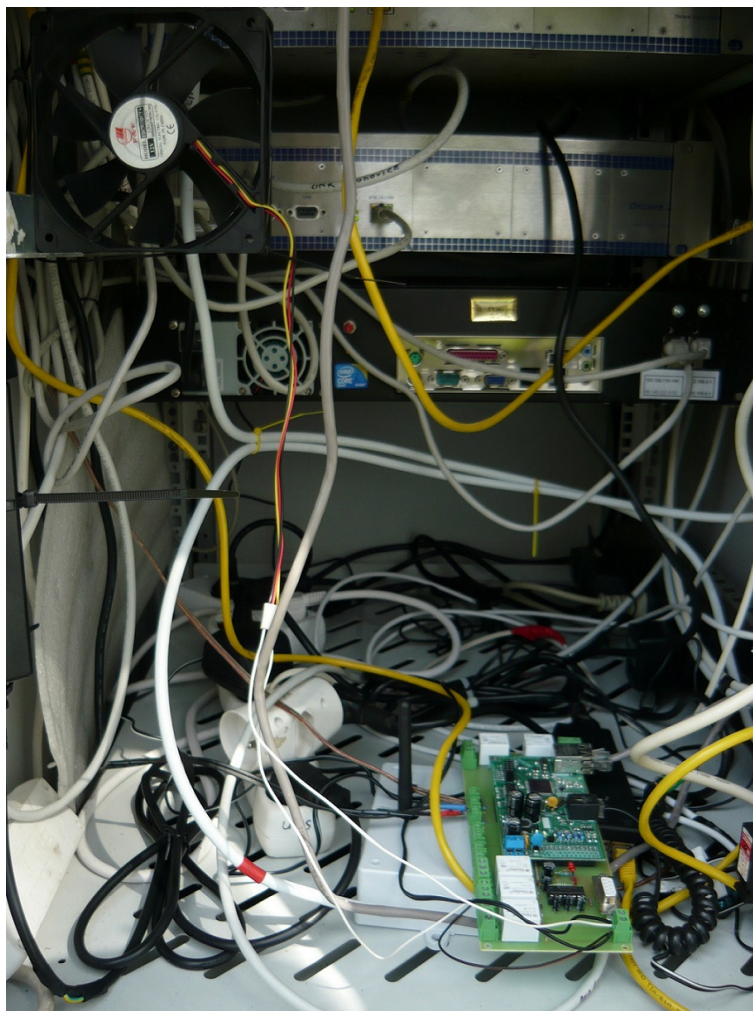


*Obrázek 4: Měsíční teplota*

Pro tvorbu všech grafů byla použita knihovna JpGraph, tato knihovna je postavena výhradně na jazyce PHP. Pro oba typy grafů platí, že je nutné mít pro dané období naměřené alespoň dvě hodnoty, které graf zobrazí, jinak se nejedná o validní graf. Pro každý typ grafu je vytvořen jeden soubor, ten graf vykreslí a uloží jako obrázek. Ten je pak na požadavek uživatele zobrazen na stránce.

## 5 Konečná instalace

Kapitola popisuje konfiguraci zařízení, tak jak bylo testováno a ukazuje jeho zapojení. Kvůli ukázce bylo zařízení odebráno ze svého krytu a vneseno do popředí.



*Obrázek 5: Ukázkové zapojení*

Zařízení je připojeno na standardní 12V zdroj, větrák je napojen na relé č.2, zdrojové napětí je na něj nakonec vedeno ze svorek napětí desky samotné (v případě připojení větší zátěže by bylo vhodné doplnit o externí zdroj). Na analogový převodník je pak vedena UPS baterie. Teplotní čidlo je vyvedeno do prostoru mimo větrák, aby jej neovlivňoval.

## 6 Testování výkonnosti a spolehlivosti

Výsledný systém bylo třeba otestovat. Kvůli podstatě jeho účelu – kontroly jiných zařízení bylo třeba zajistit 100% funkčnost a stabilitu.

### 6.1 Testování

Testování probíhalo ve dvou fázích a to již během vývoje, kdy bylo třeba testovat funkčnost jednotlivých funkcí monitorovacího zařízení. V této fázi testování zařízení řídicí program nebo klientská stránka nevykazovaly žádné poruchové stavy a byly vždy stabilní. Jediným nedostatkem je v některých případech nespolehlivá komunikace pomocí Ethernetu, kdy je lepší pro nahrání programu či stránky použít serial. Řídicí program a klientská stránka také potřebují určitý čas po jejich nahrání do zařízení (řádově sekundy) aby se nastavily všechny proměnné a celé zařízení se inicializovalo. Funkčnost systému tedy není okamžitá hned po nahrání programu/stránky do zařízení.

Při téměř měsíčním testování v terénu mělo zařízení 100% uptime a zůstalo funkční po celou jeho dobu. Řídicí program běžel nepřerušovaně a nedocházelo k jeho zásekům či restartům. Pro potřebu otestování požadovaných funkcí byl teplotní limit snížen na pouhých 20 stupňů a hystereze na 1 stupeň, aby opakovaně docházelo ke spínání chlazení a odesílání emailů. Totéž bylo provedeno pro napěťové čidla. Zařízení bylo také opakovaně vypínáno/zapínáno aby se zjistilo zda jsou všechny údaje potřebné pro jeho chod správně nastaveny a je schopno pracovat po restartu i bez uživatelského vstupu.

Všechny požadované funkce nevykazovaly žádnou poruchu a pracovaly správně po celou dobu testování. Jediným problémem bylo zablokování odesílání emailu od SMTP serveru, kvůli přílišnému počtu odeslání emailů (řádově stovky za hodinu) v ostrém provozu toto omezení nehrozí. Po celou dobu testování zařízení server systém sbíral data, která uchovával, a podle očekávání je schopen vykreslovat příslušné grafy.

Testování proběhlo v pořádku a monitorovací systém funguje podle očekávání.

### 6.2 Požadavky na instalaci

Postup zprovoznění systému je uveden níže. Některá nastavení ale bude třeba pozměnit v závislosti na konkrétních požadavcích uživatele. Např. IP adresy monitorovacích zařízení na webserveru (pro potřeby testování jsou nyní vedeny na lokální síti). Změna adresy SMTP serveru, údajů o databázi apod.

#### Řídicí program

Řídicí program je naprogramován pod firmware verze hw SoC-ARM REV4 100Mbit DF5 . Pro ostatní verze je zde možné, že program nebude plně funkční ( např. systémové proměnné nemusí zcela odpovídat starší či novější verzi ). Řídicí program po jeho zkontrolování stačí poté pouze zkompilovat a nahrát do zařízení.

## Klientská stránka

V tomto případě je situace stejná jako s řídicím programem, stránku stačí pouze nahrát do zařízení. Je ale také nutné pohlídat, jestli se všechny použité systémové proměnné shodují s použitou verzí firmware a zařízení. Pro komunikaci s SDS zařízením je nutný speciální program, který je nahrán na přiloženém CD.

## Webserver

Webserver byl testován a běží pod verzí PHP 5.3.27 a MySQL verzí 5.5.34. Pro potřebu sběru dat je nutné pustit na serveru v pozadí skript `data.php`.

Pro zobrazení grafů je třeba knihovna JpGraph, tato knihovna je součástí CD se zdrojovými soubory.

# 7 Závěr

Tato bakalářská práce měla za úkol navrhnout, implementovat a také otestovat systém pro kontrolu přístupových bodů. Po stanovení požadavků na systém tj. kontroly napětí záložních zdrojů a kontroly teploty okolí monitorovacího zařízení. Bylo nutné také vymyslet způsob reakce na tyto události. Místo prostého zobrazení údajů byla impelentována funkce odesílání emailu. Pro potřeby zobrazení aktuálních údajů byla také navržena a implementována klientská stránka, sloužící také pro ovládání aktivních prvků zařízení. V důsledku nasazení více zařízení byl implementován server systém odkazující na všechny nasazené monitorovací zařízení. Server systém také sbírá a uchovává teplotní údaje, které prezentuje ve formě grafů.

Systém byl testován v provozu u přístupových bodů v obci Police u Valašského Meziříčí. Během téměř měsíčního testování řídicí program pracoval podle očekávání a monitorovací zařízení tak po celou dobu plnilo svoji funkci. Klientská stránka byla vždy přístupná a bylo možné kontrolovat údaje a také ovládat aktivní prvky.

Díky nižším venkovním teplotám bylo nutné pro potřeby testování upravit teplotní limit a v plné míře tak otestovat schopnost zařízení odesílat emaily a spínat/vypínat relé. Systém splňuje předepsané požadavky, ukázal se jako stabilní. Systém je již nasazen v terénu ve výše zmiňované obci. V budoucnu se plánuje rozšíření i na další stanice přístupových bodů.

Původním nápadem pro pokračování práce bylo vytvoření mobilní aplikace pro klientskou stránku a případně také pro server. Tato aplikace by zobrazovala měřená data a umožnila by ovládat aktivní prvky tak jako stránka uživatelská. V průběhu vývoje tohoto systému, přesněji na konci ledna 2015 byla již výrobcí zařízení SDS taková aplikace uvolněna a nyní je dostupná na GOOGLE PLAY, od svého uvolnění prošla některými změnami a vylepšeními a nyní je na takové úrovni, která odpovídá mé představě o možném pokračování této práce. [13]

V tomto případě je tedy možné navrhnout některá vylepšení a rozšíření monitorovacího systému. Například napojení řídicího programu na SMS bránu, kdy by pak upozornění nebylo posíláno na email, ale přímo na mobil prostřednictvím SMS zpráv.

Další možností je rozšíření funkčnosti systému samotného, a to například v podobě využití optických vstupů, jejichž obsluha je již v řídicím programu naprogramována. Tyto vstupy detekují přítomnost napětí a v případě kontaktu uvědomí uživatele pomocí emailu, podobně jako u měření napětí či teploty. Tohoto by bylo vhodné využít jako detekci výpadku elektřiny přímo na rozvodech přístupových bodů. Systém by pak detekoval, jak samotný výpadek proudu, tak i případné vybíjení záložních zdrojů.

Vhodným rozšířením by také bylo přiřazení dalších funkcí na ostatní relé výstupy. Jako vítaným vylepšením se jeví pořízení externího zdroje pro externí relé výstupy, kde by se naskytla možnost pracovat s větším napětím a pohánět tak například celou chladicí soustavu, apod. V tomto případě by se také dal využít výrobek řady SDS MACRO se zabudovaným termostatem.

Bylo by také vhodné rozšířit server systém a sledovat a uchovávat další potenciálně zajímavá data. Jelikož je zařízení opravdu velice univerzální, přidání nových funkcí a případné rozšíření celého modulu je úzce svázáno s požadavky na cílový systém.

# Literatura

- [1] SDS-C. *Sitovy Dohledovy System: SDS* [online]. Opava (Česká Republika): AN-D.cz, 1.3.2012- [cit. 2015-05-17]. Dostupné z: <http://wiki.merenienergie.cz/index.php/SDS-C>
- [2] FULL-C. *Sitovy Dohledovy System: SDS* [online]. Opava (Česká Republika): AN-D.cz, 1.3.2012- [cit. 2015-05-17]. Dostupné z: <http://wiki.merenienergie.cz/index.php/FULL-C>
- [3] Sdsc syp. *Sitovy Dohledovy System: SDS* [online]. Opava (Česká Republika): AN-D.cz, 1.3.2012- [cit. 2015-05-17]. Dostupné z: [http://wiki.merenienergie.cz/index.php/Sdsc\\_syp](http://wiki.merenienergie.cz/index.php/Sdsc_syp)
- [4] C (programovací jazyk). *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 20.6.2004- [cit. 2015-05-17]. Dostupné z: [http://cs.wikipedia.org/wiki/C\\_%28programovac%C3%AD\\_jazyk%29](http://cs.wikipedia.org/wiki/C_%28programovac%C3%AD_jazyk%29)
- [5] HyperText Markup Language. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 16.7.2004- [cit. 2015-05-17]. Dostupné z: [http://cs.wikipedia.org/wiki/HyperText\\_Markup\\_Language](http://cs.wikipedia.org/wiki/HyperText_Markup_Language)
- [6] Cascading Style Sheets. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 14.11.2001- [cit. 2015-05-17]. Dostupné z: [http://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://en.wikipedia.org/wiki/Cascading_Style_Sheets)
- [7] JavaScript. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 12.8.2004- [cit. 2015-05-17]. Dostupné z: <http://cs.wikipedia.org/wiki/JavaScript>
- [8] PHP. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 1.10.2001- [cit. 2015-05-17]. Dostupné z: <http://en.wikipedia.org/wiki/PHP>
- [9] MySQL. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 22.11.2004 [cit. 2015-05-17]. Dostupné z: <http://cs.wikipedia.org/wiki/MySQL>
- [10] BLAZEK, . PLC Foxtrot s vestavěným WEB serverem. *Automatizace.hw.cz: rady a poslední novinky z oboru* [online]. 2010 [cit. 2015-05-17]. Dostupné z: <http://automatizace.hw.cz/plc-foxtrot-s-vestavenym-web-serverem>
- [11] TECOMAT. *PROGRAMOVATELNÉ AUTOMATY: PROGRAMOVATELNÉ AUTOMATY TECOMAT FOXTROT CP-1000, CP-1001, CP-1020*. 6. Kolín, 2014. Dostupné také z: [http://www.tecomat.com/wpimages/other/DOCS/cze/TXV00430\\_01\\_Foxtrot\\_CP\\_1000.pdf](http://www.tecomat.com/wpimages/other/DOCS/cze/TXV00430_01_Foxtrot_CP_1000.pdf)
- [12] ONLINETECHNOLOGY. *SDS - Síťový Dohledový Systém: Inteligentní Domovní Systémy a Měření Energií* [online]. 2010 [cit. 2015-05-17]. Dostupné z: <http://www.onlinetechnology.cz/>
- [13] SDS Control Panel. 2015. *Google Play* [online]. Mountain View, CA [cit. 2015-05-18]. Dostupné z: <https://play.google.com/store/apps/details?id=com.sds.sds4smarhome.sdscontrolpanel>

# Seznam příloh

- Příloha A – obrazovky systému
- Příloha B – cd se zdrojovými soubory

## Příloha A – obrazovky systému

**Přístupový bod Branky**

[Přehled](#)

Čas	Uptime
23:24:12 17.05.2015	0 d 0:1:51

Vstup	Aktuální stav	Název	
AD1	0.058 V	<input type="text" value="AD3"/>	<input type="button" value="Změňt"/>
AD2	0.088 V	<input type="text" value="AD2"/>	<input type="button" value="Změňt"/>
AD3	0.088 V	<input type="text" value="AD3"/>	<input type="button" value="Změňt"/>
AD4	0.088 V	<input type="text" value="AD4"/>	<input type="button" value="Změňt"/>

Název - Hodnota	Limit
AD1 - min - <input type="text" value="0"/>	<input type="button" value="Nastav!"/> 8
AD2 - min - <input type="text" value="0"/>	<input type="button" value="Nastav!"/>
AD3 - min - <input type="text" value="0"/>	<input type="button" value="Nastav!"/>
AD4 - min - <input type="text" value="0"/>	<input type="button" value="Nastav!"/>

Vstup	Aktuální stav
Teplota[0]	21.43 °C
Teplota[1]	167772.16 °C
Teplota[2]	167772.16 °C

Název - Hodnota	Limit
Temp[0] - max - <input type="text" value="0"/>	<input type="button" value="Nastav!"/> 0
Temp[1] - max - <input type="text" value="0"/>	<input type="button" value="Nastav!"/> 0
Temp[2] - max - <input type="text" value="0"/>	<input type="button" value="Nastav!"/> 0

Výstupy	Aktuální stav	Název	
D0	výstup log-0	D0	
D1	OFF	D1	
D2	OFF	D2	
RE1	ovládáno:ručně - neaktivní (off)	<input type="text" value="RE1"/>	<input type="button" value="Změňt"/>
RE2	ovládáno:ručně - neaktivní (off)	<input type="text" value="RE2"/>	<input type="button" value="Změňt"/>

Výstup	Stav
D0	<input type="button" value="Vypnout"/> <input type="button" value="Zapnout"/>
D1	<input type="button" value="Vypnout"/> <input type="button" value="Zapnout"/>
D2	<input type="button" value="Vypnout"/> <input type="button" value="Zapnout"/>
RE1	<input type="button" value="Vypnout"/> <input type="button" value="Zapnout"/>
RE2	<input type="button" value="Vypnout"/> <input type="button" value="Zapnout"/>

Vstup	Aktuální stav
Opto1	vstup:OFF, imp:0
Opto2	vstup:OFF, imp:0
Opto3	vstup:OFF, imp:0

Výpis konzole
<input type="button" value="Vymazat!"/> <input type="button" value="Zobraz názvy!"/>
Program init!

©Valnet 2015

Obrázek 6: Klient



[Správa zařízení](#)

Bod	Adresa	Stav
Branky	<a href="#">192.168.1.250</a>	✓
	<a href="#">Statistiky</a>	
Štěpánov	<a href="#">192.168.1.251</a>	✗
	<a href="#">Statistiky</a>	
Police	<a href="#">192.168.1.252</a>	✗
	<a href="#">Statistiky</a>	

Obrázek 7: Server systém

## Příloha B – CD se zdrojovými soubory

Příložené CD obsahuje následující soubory

src	Složka se všemi potřebnými soubory pro instalaci systému
readme.txt	Info o systému
pokyny.txt	Pokyny pro instalaci systému
BAK.odt	Bakalářská práce ve zdrojovém formátu
BAK.pdf	Pdf verze bakalářské práce