
Technická univerzita v Liberci



Systém řízení chytrého domu s inteligentní elektroinstalací na základě OS Android

diplomová práce

Fakulta :Strojní

Studijní obor :Automatizované systémy řízení ve strojírenství

Vypracoval : Yudakhin Mikhail

Zadání

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Datum

Podpis

Poděkování

Touto cestou bych rád poděkoval vedoucímu diplomové práce Ing. Radku Votrubcovi, Ph.D za rady a pomoc při řešení této práce.

Abstrakt

Cílem diplomové práce je navrhnutí, realizace a test systému „inteligentní dům“. Úkol byl rozdělen na tři části: vytvoření plně funkční aplikace pro operační systém Android, naprogramování PLC a vytvoření modelu pro demonstrace. Systém umožní kontrolu a řízení a zajistí souvislost mezi jednotlivými členy systému.

V teoretické části diplomové práce jsou popsány systémy ovládání inteligentního domu různých výrobců, konkrétně GILD smart house system, Teco a Jablotron. Je zde zároveň provedena analýza uvedených systémů a popsány jejich výhody a nevýhody.

V praktické části diplomové práce je popsán proces vytvoření programu pro tablet a programovatelný automat a znázorněn postup vytvoření makety domu.

Klíčová slova: Android OS, mobilní zařízení, chytrý dům, PCL, FOXTROT, krokový motor, Modbus, Java, vývojové prostředí Eclipse, vývojové prostředí Mosaic.

Abstract

The purpose of this thesis is design, implementation and test of the system „Intelligent Home“. The task was divided into three parts: Android OS application implementation, PLC programming and demonstration model creation. This system allows the control and management as well as the assurance of the connection between individual members of the system.

In the theoretical part of this work will be about intelligent building control systems from different manufacturers, to be more specific GILD smart house system, Teco and Jablotron. There will be an analysis of the described systems and their advantages and disadvantages.

The practical part of the thesis describes the process of creating a program for a pad and programmable logic controller, which shows the creation of a house model.

Keywords: Android OS, mobile devices, intelligent home, PCL, FOXTROT, stepper motor , Modbus, Java, integrated development environment Eclipse, integrated development environment Mosaic.

Obsah

1. Úvod.....	1
1.1. Stanovení cíle	1
2. Rozbor problematiky	2
2.1. Existující zařízení a software.....	2
2.1.1. GILD smart house system	2
2.1.2. Systém „Ovládej svůj dům“	4
2.1.3. Jablotron 100	7
2.2. Navržený systém.....	9
3. Popis použitých zařízení	11
3.1. Programovatelný automat.....	11
3.2. Zdroj	13
3.3. Tablet.....	14
3.4. WiFi směrovač.....	16
3.5. Krokový motor	18
3.6. Řízení krokového motoru	20
3.7. Teplotní čidlo.....	22
4. Realizace programu v tabletu.....	24
4.1. Vývojové prostředí	24
4.2. Operační systém Android	26
4.2.2. Architektura Android-aplikace.....	26
4.3. Komunikační protokol ModBus	31
4.4. Aplikace „IntelliHome“	33
4.4.1. Běh aplikace	34
4.4.2. Třídy aplikace.....	36
4.5. Vytváření interfacu	40
4.5.1. Soubor manifest.xml	42
5. Realizace programu v PLC	44
5.1. Vývojové prostředí	44
5.2. Aplikace „IntelliHome_PLC“	50
5.2.1. Popis uživatelských registrů.....	51
5.2.2. Popis programu	52
6. Realizace makety	56
6.1. Model domu.....	56
7. Závěr	63
Použitá literatura	64
Příloha	66

Seznam obrázků a tabulek

Obrázek 1. Základní modul CP-1016.....	11
Obrázek 2. Zdroj Siemens LOGO!Power 6EP1332-1SH51	13
Obrázek 3. Tablet ASUS EEE Pad Transformer.....	14
Obrázek 4. WiFi směrovač TP-LINK TL-WR340G.....	16
Obrázek 5. Krokový motor Mitsumi M35SP-8.....	19
Obrázek 6. EM-121 Ovladač krokovým motorem.....	20
Obrázek 7. Bloková schéma ovladače.....	20
Obrázek 8. Připojení ovladače	21
Obrázek 9. Teplotní čidlo CRZ-2005-1000-B-0.5	22
Obrázek 10. Vývojové prostředí Eclipse.....	25
Obrázek 11. Architektura Androidu.....	26
Obrázek 12. Životní cyklus aktivity.....	28
Obrázek 13. Aplikace se třemi vlákny	30
Obrázek 14. Hlavní okno aplikace	33
Obrázek 15. Hlavní aktivita aplikace	34
Obrázek 16. Vlákno PollModbus	35
Obrázek 17. Ukázka kódu - zadání nastavení	36
Obrázek 18. Ukázka kódu - inicializace elementů interfacu.....	37
Obrázek 19. Ukázka kódu – zpracování připojení a odpojení k/od serveru.....	37
Obrázek 20. Ukázka kódu – zpracování stisknutí tlačítek	38
Obrázek 21. Ukázka kódu – provedení připojení se serverem.....	38
Obrázek 22. Ukázka kódu – provedení odpojení od serveru	39
Obrázek 23. Ukázka kódu – čtení a psaní registrů dle adresy.....	39
Obrázek 24. Hierarchie XML souboru.....	40
Obrázek 25. Ukázka kódu – začátek souboru main.xml	41
Obrázek 26. Ukázka kódu – souboru AndroidManifest.xml	43
Obrázek 27. Manažer projektu	45
Obrázek 28. Nastavení vstupů/výstupů.....	46
Obrázek 29. Vývojové prostředí Mosaic.....	49
Obrázek 30. Algoritmus programu “IntelliHome_PLC”	50
Obrázek 31. Ukázka kódu – definice registrů	52
Obrázek 32. Ukázka kódu – definice proměnných	53
Obrázek 33. Ukázka kódu – porovnání času.....	54
Obrázek 34. Ukázka kódu – řízení osvětlení.....	54
Obrázek 35. Ukázka kódu – řízení topení	55
Obrázek 36. Ukázka kódu – řízení rolety.....	55
Obrázek 37. Model domu v prostředí AutoDesk Inventor 2012.....	56
Obrázek 38. Maketa domu	57
Obrázek 39. Maketa domu bez stěn	57
Obrázek 40. Použité materiály	58
Obrázek 41. Připojení zařízení	59
Obrázek 42. Připojení FOXTROT CP-1016	60
Obrázek 43. Připojení ovladače EM-121	60
Obrázek 44. Připojení zdroje LOGO!Power	61
Obrázek 45. Připojení žárovek	61
Obrázek 46. Připojení motoru	62
Obrázek 47. Připojení čidla.....	62

Tabulka 1. Srovnání systémů	10
Tabulka 2. Základní parametry modulu CP-1016.....	12
Tabulka 3. Technické parametry zdroje Siemens LOGO!Power 6EP1332-1SH51.....	13
Tabulka 4. Technické parametry tabletu ASUS EEE Pad	15
Tabulka 5. Technické parametry směrovače.....	17
Tabulka 6. Technické parametry motoru	19
Tabulka 7. Technické parametry ovladače.....	21
Tabulka 8. Technické parametry čidla	23
Tabulka 9. Odpor čidla dle teploty.....	23
Tabulka 10. Typ přenášených dat	32
Tabulka 11. Kódy základních funkcí	32
Tabulka 12. Popis uživatelských registrů.....	51

1. Úvod

Mým úkolem v této diplomové práci bylo navrhnutí, realizace a test systému „inteligentní dům“. „Chytrý dům“ nebo „inteligentní dům“ je automatické zařízení, které spojí různé systémy v bytě nebo domě. Systém umožní kontrolu a řízení a zajistí souvislost mezi jednotlivými členy systému. V inteligentním domě tak odpadají zbytečné činnosti, např. zvedat se pokaždé, když je třeba zapnout topení nebo vypnout světlo apod. S tímto systémem stačí jen stisknout tlačítko a činnost bude provedena.

V teoretické části diplomové práce budou popsány systémy ovládání inteligentního domu různých výrobců, konkrétně GILD smart house system, Teco a Jablotron. Bude zde provedena analýza uvedených systémů a popsány jejich výhody a nevýhody.

V praktické části diplomové práce bude popsán proces vytvoření programu pro tablet a programovatelný automat a znázorněn postup vytvoření makety domu.

1.1. Stanovení cíle

Prvním úkolem bylo studium a analýza dosavadních poznatků o procesu řízení elektroinstalačních elementů, kterými jsou osvětlení, topení a žaluzie. Dalším cílem bylo na základě předchozí analýzy a formulace problému zadaného úkolu navrhnout vhodnou variantu řídicího systému. K tomuto účelu bylo zvoleno takové zařízení, které vyhovovalo podmínkám řízení elementů a ovládání přes bezdrátovou síť. Řídící centrálou byl zvolen programovatelný automat FOXTROT CP-1016. Jako ovládací element byl vybrán tablet pod řízením operačního systému Android OS Asus eeePad TF101.

Pro vytvoření řídicího programu pro tablet bylo použito vývojové prostředí Eclipse. Programování v tomto prostředí umožňuje vytvářet klientské aplikace pro řízení a nastavení řídicího programu v PLC. GUI aplikace pro tablet se provádí pomocí jazyku XML, zdrojový kód je psán v jazyku Java. Vývojovým prostředím pro vytváření programu pro PLC je Mosaic od firmy Teco a.s., které umožňuje programování dle IEC-61131-3.

Posledním cílem byla realizace modelu chytrého domu a aplikace na vytvořeného řídicího systému. Model byl vytvořen ze dřeva a sololitu. Byl navržen tak, aby byl vhodný pro demonstraci práce krokového motoru resp. PLC FOXTROT.

2. Rozbor problematiky

2.1.Existující zařízení a software

2.1.1. GILD smart house system

Tento systém je základní sestavou pro ovládání světelných nebo zásuvkových okruhů v rámci jednoho rozvaděče. Pro ovládání lze volit vypínače, tlačítka,



dálkové ovladače nebo pohybové snímače pro venkovní prostory. Ovládat lze také světelné stmívače nebo žaluzie. Zvláště výhodné je jeho nasazení ve formě uzavřeného systému pro ovládání žaluzií.

Lze vytvářet jednoduché logické návaznosti vstupů a výstupních okruhů. Veškeré konfigurace jsou prováděny přímo na řídicí jednotce bez použití počítače. V případě potřeby lze počítač připojit pomocí rozhraní USB. Ovládání je velmi intuitivní a nevyžaduje hlubší znalosti v oblasti elektroinstalací.

Hlavní rysy:

- ovládání tlačítka nebo vypínači a IR ovladači
- řízení zásuvkových a světelných okruhů
- ovládání žaluzií a rolet
- pouze ovládání (bez časových funkcí)
- programování přímo na řídicí jednotce
- možnost připojení počítače přes USB

Typické použití:

- Malé byty a jednoduché elektroinstalace s jedním rozvaděčem
- Uzavřené systémy ovládání žaluzií [6]

Ovládání systému je možné pouze pomocí počítače nebo dotykového panelu. Nevýhodou této koncepce je vysoká pořizovací cena panelu a nízká mobilita počítače. Také existuje varianta řízení pomocí web-serveru, ale tento systém zajistí řízení jen několika funkcí a nastavení omezeného počtu elementů.

2.1.2. Systém „Ovládej svůj dům“

Systém je od Teco a.s., předního českého výrobce průmyslových řídicích systémů a

The logo consists of the text "ovladejstvujdum.cz" in a white, lowercase, sans-serif font, centered within a solid orange rectangular background.

systémů pro inteligentní řízení budov. Systémy, které vyvíjí a vyrábí, úspěšně fungují v náročném prostředí průmyslu v těch nejtěžších podmínkách strojírenského, těžbařského, potravinářského či ropného průmyslu. Řídí velká vodní díla v íránské poušti stejně jako např. nadnárodní výrobu piva.

Systémy řídí budovu českého dědictví – hradu Karlštejn, ale také tisíce dalších aplikací od běžných rodinných domů po rozsáhlé komerční či průmyslové objekty. Díky mimořádné spolehlivosti, výjimečným vlastnostem a výhodné ceně jsou systémy od Teco a.s. populární.

Hlavní rysy:

Světla a světelné scény

- Rozsvítí světla, když si to přejete, a zhasne po Vás, když zapomenete.
- Světelné scény pro různé příležitosti v závislosti na vašem vlastním scénáři nebo čase, venkovním osvětlení či přítomnosti, dokonce i se zvukovou kulisou.

Topení, klimatizace a ventilace

- Koordinovaná regulace topení, klimatizace, a zejména řízená ventilace s rekuperací individuálně v každé místnosti.
- Závislost na čase, venkovní teplotě, větru, vydýchaném vzduchu v místnosti nebo otevřeném okně.
- Přináší často až 35% úspory v nákladech na energie.

Žaluzie, rolety, markýzy

- Automatizované ovládání podle více kritérií (čas, venkovní osvětlení, světelná scéna, vítr, déšť).
- Dotvoření světelného dojmu (vpuštění ranního slunce, zatažení v době nepřítomnosti) a zvýšené zabezpečení domu.
- Automatické zatažení nebo vytažení při silném větru a dešti nebo slunci či mrazu.

Garážová vrata, brány

- Ovládá i hlídá, zda jsou zavřená a umí Vás varovat.
- Zavlažování zahrady.

- Zavlažuje podle nastaveného programu.
- Šetří vodu, pokud je po dešti nebo právě prší.

Bazén, vířivka

- Řízení optimálního ohřevu z různých zdrojů – solární panely, tepelné čerpadlo apod.
- Zapojení ovládání krytu bazénu nebo jeho osvětlení včetně řízení barevných nálad.
- Automatizované ovládání (zapínání, zakrývání, otevírání) v závislosti na dešti, větru, teplotě, časovém programu.

Bezpečnostní systém

- Do systému lze integrovat snímače pohybu, okenní a dveřní kontakty, detektory rozbití skla, snímače kouře.
- Systém hlídá a varuje vás nebo zásahovou jednotku, pokud dojde k narušení domu.
- Automatické zhasnutí světel v místnostech, kde nikdo není.
- Vypne topení, když je otevřené okno.

Kamerový systém

- Zprostředkuje snímky z kamer pohotově přes WEB prohlížeče ve vašem mobilním telefonu nebo počítači.
- Při detekci neočekávaného pohybu nebo jiné alarmující události zaznamená do své paměti jednotlivé obrázky jako důkazní materiál.
- Obrázky lze zároveň odesílat několika cestami na vzdálený server např. zásahové služby.

Elektrické spotřebiče, zásuvky

- Ovládání elektrických spotřebičů automatizovaně například přes ovládanou zásuvku.
- Ovládání „chytrějších spotřebičů“ například simulací jejich dálkového ovladače – Foxtrot pak vydává akční povely automatizovaně na základě podmínek, které si určíte.
- Výhodné pro akce, které můžeme odložit až na dobu s nízkým tarifem.

Vzdálený přístup

- Ovládáte dům odkudkoli na světě přes internet nebo svým mobilním telefonem pomocí SMS zprávy.
- V případě zásahu do systému instalační firmou není nutný výjezd technika, vše diagnostikuje na dálku.

Automatizace funkcí

- Vhodným nastavením funkcí systému získáte volný čas, protože váš dům udělá spoustu práce za vás.
- Ušetříte peníze za energie, protože váš dům bude optimálně ovládat topení, klimatizaci, světla i elektrické spotřebiče. [7]

2.1.3. Jablotron 100

Zcela nový alarm JABLOTRON 100 s převratným systémem ovládání je komplexním řešením nejen pro ochranu majetku a osob proti napadení a požáru, ale integruje v jednom jádru i funkce inteligentního řešení domu.

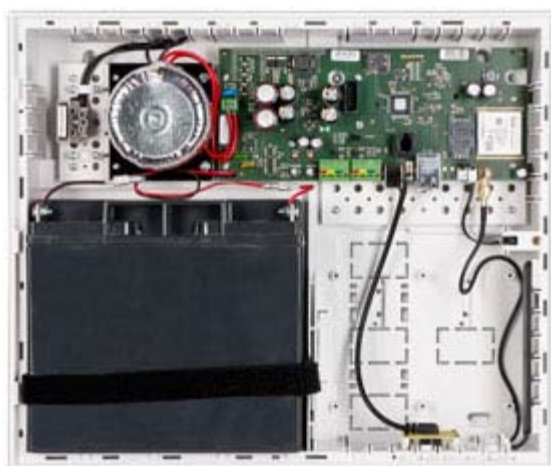
Alarm umožňuje variabilně využít sběrnicových a bezdrátových systémových prvků. Pružně se přizpůsobí jak nárokům na zabezpečení rozsáhlé firmy, tak potřebám domácností a zároveň zůstává velmi jednoduše ovladatelný.



Tyto zdánlivě neslučitelné požadavky jsou uspokojeny revolučním patentovaným systémem ovládání. Nové na tomto alarmu není jen ovládání, ale celá jeho architektura. Rozsah komponent, variabilita počtu periférií, sběrnicové uspořádání, nový bezdrátový protokol, možnosti komunikace a vzdáleného servisního i uživatelského přístupu z něj dělají nadčasové řešení elektronického zabezpečení.

Patentovaný systém ovládání

Ústřednu lze ovládat až 300 uživatelskými kódy. Pro ovládání systému jsou k dispozici sběrnicové i plně bezdrátové verze přístupových modulů. Všechny jsou vybaveny RFID čtečkou a dle modelu také klávesnicí nebo klávesnicí i displejem. K modulům se připojují ovládací segmenty, které s jednoduchou logikou semaforu umožňují snadné ovládání sekcí, výstupů, tísňových volání a zobrazování stavu systému. Díky ovládacím segmentům zobrazujícím zelenou a červenou signálkou přehledně stav jednotlivých sekcí a výstupů je JABLOTRON 100 unikátně přehledný pro každého uživatele. Systém se ovládá stiskem příslušného segmentu a verifikuje zadáním kódu nebo přiložením RFID karty nebo čipu.



Vlastnosti systému

Systém JABLOTRON 100 je navržen a certifikován dle EN 50131 do stupně zabezpečení 2. Ústředna kombinuje sběrníkovou architekturu drátového připojení s bezdrátovým připojením jednotlivých prvků. Bezdrátová komunikace je postavena na novém obousměrném dynamickém přenosovém protokolu Jablotron 100 využívajícím frekvenci 868 MHz se zajištěním násobné ochrany bezpečnosti a integrity systému proti spontánním i záměrným kolizím komunikace systému. Dosahy komunikace z jednotlivých periférií jsou na přímou viditelnost v řádech stovek metrů a dají se prodlužovat připojením dalších rádiových modulů na sběrnici. Pro dokonalé pokrytí zabezpečovaného prostoru lze do systému instalovat až 3 moduly rádia.

JABLOTRON 100 má až 120 bezdrátových nebo sběrníkových zón pro přiřazení detektorů, klávesnic a ovladačů, sirén případně dalších prvků. Systém lze rozdělit na 15 sekcí. K využití se nabízí 32 programovatelných výstupů.

Komunikace

Ústředna je vybavena GSM/GPRS/LAN komunikátorem, který umožňuje hlasovou, SMS nebo GPRS komunikaci s uživateli a středisky PCO. komunikátor je možné doplnit do pevné telefonní linky. Komunikátory kromě funkce přenosu informací ze systému plní funkci modemu pro plnohodnotný vzdálený přístup, monitoring, programování a v neposlední řadě i ovládání výstupů (a tím např. spotřebičů v domě). Vzdálený přístup lze realizovat z internetové aplikace, SMS příkazy i přímým připojením z nastavovacího softwaru F-Link.

Pro individuální řešení návrhu zabezpečení objektu má k dispozici širokou škálu prvků ve sběrníkovém i bezdrátovém provedení. V jedné designové řadě nabízí klasický PIR detektor i PIR s integrovaným nezávislým akustickým detektorem tříštění skla. V bezdrátové variantě je k dispozici i kombinovaný PIR a mikrovlnný detektor pohybu. Takové řešení je v bezdrátovém provedení zcela ojedinělé. Pro zajištění oken a dveří je k dispozici miniaturní magnetický detektor a speciální "neviditelný" magnet pro skrytou montáž do plastových rámců oken nebo dveří.

Pro speciální aplikace je určený miniaturní stropní PIR detektor a detektor tříštění skla. Ty jsou vhodné i pro použití k zabezpečení vozu parkujícího v dosahu ústředny systému. Plášťovou ochranu objektu doplňují venkovní detektory pohybu a infrazávora. Požární bezpečnost řeší optický kouřový detektor a detektor úniku výbušných plynů.[8]

2.2.Navržený systém

Po posouzení silných a slabých stránek popsaných systémů jsem se snažil vytvořit systém, který by měl mít méně nedostatků, ale zároveň více výhod a možnost rozšíření. To znamená, že v navrženém systému existuje možnost integrovat více různých funkcí, které chybějí v popsaných již existujících systémech, to vše za rozumnou finanční částku.

Pro vzdálené řízení jsem zvolil zařízení fungující na operačním systému Android. Výhodou tohoto zařízení je velký výběr produktů pracujících na operačním systému Android, a také poměrně nízká cena v porovnání s produkty, které fungují na operačních systémech iOS a WindowsPhone. Další výhodou je stejný SDK pro programování mobilu a tabletu, což umožňuje se vyhnout zbytečným nákladům při přenosu programu z tabletu do mobilu nebo do PC.

Pro ovládání byl vybrán prvek PLC společnosti Tecomat. Navržený PLC má vysoký výkon a nízkou cenu. Také podporuje několik protokolů komunikace, což je výhodou při integraci do jakéhokoli systému. Tento PLC podporuje otevřený protokol Modbus, který byl zvolen jako protokol komunikace mezi tabletem a PLC.

Hlavním důvodem volby tohoto protokolu je jeho dostupnost na internetu na rozdíl od Siemens PROFIBUS. Právě to, že tento protokol je veřejný, dovoluje vytvořit aplikace v jakémkoliv jazyku programování a pod jakoukoliv platformou a bude-li potřeba, umožňuje také modernizaci.

Výhody a nevýhody popsaných systémů a seznam funkcí navrženého systému jsou znázorněny v tabulce číslo 1.

Funkce	GILD	FOXTROT	Jablotron	IntelliHome
Osvětlení	●	●	●	●
Roleta/žaluzie	●	●	◐	●
Topení	○	●	◐	●
Garážová vrata	○	●	◐	◐
Bezpečnostní systém	○	●	●	◐
Kamerový systém	○	●	○	○
Zásuvky	○	●	●	◐
Automatizace	○	●	●	●
Vzdálený přístup	○	● (pouze SMS)	●	●
Ovládání přes počítač	●	●	● (pouze web)	◐
Ovládání přes mobilní telefon	○	● (pouze SMS)	● (pouze web)	●
Ovládání přes tablet	○	●	●	●
Ovládání přes dotykový panel	●	●	○	◐
Ovládání přes webové rozhraní	○	○	●	◐

● - umožňuje

◐ - existuje možnost přidat

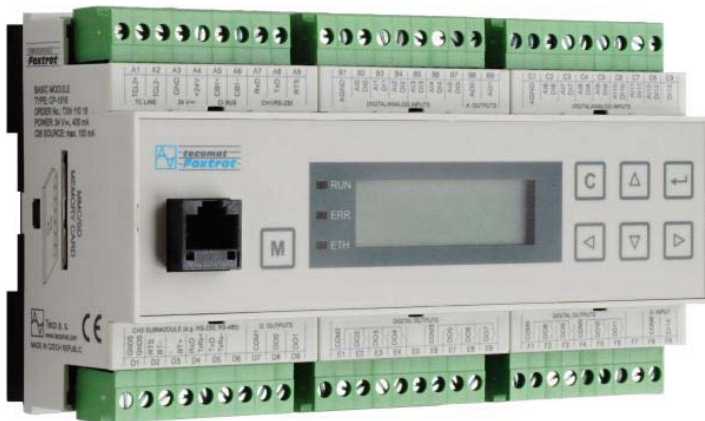
○ - neumožňuje

Tabulka 1. Srovnání systémů

3. Popis použitých zařízení

3.1. Programovatelný automat

Programovatelný automat byl dodán katedrou KKY. Jedná se o moderní automat Teco Tecomat Foxtrot CP-1016. Základní modul CP-1016 (viz. obr. 1) je ze sortimentu základních modulů modulárních programovatelných automatů řady Foxtrot. Jednotlivé ZM se liší počtem nebo typem vstupů a výstupů a indikačními a ovládacími prvky.



Obrázek 1. Základní modul CP-1016

Základní modul CP-1016 je vybaven třinácti víceúčelovými vstupy, z nichž každý je využitelný buď jako analogový nebo jako binární, jedním rychlým binárním vstupem využitelným jako vstup čítače, jedním binárním vstupem pro napětí 230 V AC, dvěma polovodičovými výstupy využitelnými jako výstupy PWM, dvěma analogovými výstupy 0 až 10 V a deseti reléovými výstupy. Základní ovládací a zobrazovací prvky jsou rozšířeny o LCD displej 4 × 20 znaků a 6 uživatelských tlačítek.

Základní modul CP-1016 je osazen centrální jednotkou (CPU) řady K, která je určena pro aplikace s vysokými požadavky na výkon. Obsahuje zálohovanou paměť CMOS RAM pro uživatelské programy, data, tabulky, uživatelské registry a DataBox, paměť Flash pro zálohování uživatelského programu, slot pro MMC/SD paměťovou kartu, obvod reálného času, rozhraní Ethernet, dva sériové kanály (jeden s pevným rozhraním RS-232, druhý s pozicí pro volitelné submoduly), jeden komunikační kanál s rozhraním CIB pro připojení externích periférií a systémové rozhraní TCL2 určené pro připojení rozšiřovacích modulů, které zvyšují počet I/O systému.[6] Základní technické parametry modulu jsou znázorněny v tabulce číslo 2.

Typ zařízení	vestavné
Napájecí napětí (SELV)	24 V DC, +25 %, –15 %
Příkon modulu	max. 10 W
Rozměry (v × š × h)	90 × 158 × 58 mm
Obvod reálného času (RTC)	ano
Paměť uživatelského programu a tabulek	192 + 64 KB
Záložní paměť programu EEPROM	ano
DataBox (interní přídatná paměť dat)	512 kB
Paměť pro archivaci projektu	2 MB
Slot pro MMC/SD kartu	ano
Doba cyklu na 1k log. instrukcí	0,2 ms
Počet uživatelských registrů	64 KB
z toho remanentních registrů	32 KB
Rozhraní Ethernet	10/100 Mb/s
Integrovaný Web server	ano
Parametry binárních vstupů DI0 – DI14	
Počet vstupů	15
Vstupní napětí	
pro log.0 (UL)	max. 12 V DC min. 2,3 V DC
pro log.1 (UH)	min. 0 V DC max. 1 V DC
Parametry triakových výstupů DO0, DO1	
Počet výstupů	2
Spínané napětí	max. 260 V min. 20 V
Spínaný proud	max. 1 A min. 5 mA
Parametry reléových výstupů DO2 - DO11	
Počet výstupů	10
Spínané napětí	max. 250 V min. 5 V
Spínaný proud	max. 3 A min. 10 mA
Parametry analogových vstupů AI0 – AI12	
Počet vstupů	13
Měřicí rozsah/rozlišení	Pt1000, W100 = 1,385 - –90 až +270°C Pt1000, W100 = 1,391 - –90 až +270°C

Tabulka 2. Základní parametry modulu CP-1016

3.2.Zdroj

Napájecí zdroj je zařízení sloužící k přeměně střídavého napětí dodávaného ze sítě na nízké napětí potřebné napájení. Napájecí zdroje jsou hodnoceny na základě maximálního výkonu a výstupního napětí. Při vytvoření makety domu jsem použil zdroj Siemens LOGO!Power 6EP1332-1SH51 (viz. obr. 2). Základní technické parametry zdroje jsou znázorněny v tabulce číslo 3.

Hlavní rysy:

- Hlavní spínané PSU se širokým rozsahem vstupů AC
- Provoz také na napětí DC, 110 až 300 VDC
- Velmi kompaktní konstrukce
- Nastavitelné výstupní napětí
- Možné zapojení s vysokými nárazovými proudy
- Rezerva výkonu 150 % po dobu 200 ms
- Možnost paralelního spojení
- Úroveň potlačení rušení EN 55022 třída B
- Elektrická izolace, SELV (EN 60950 a EN 50178)[9]



Obrázek 2. Zdroj Siemens LOGO!Power 6EP1332-1SH51

Střídavé vstupní napětí	85...264 VAC
Jmenovité napájecí napětí při AC 50 Hz	100...240 V
Výstupní napětí	22.2...26.4 VDC
Výstupní proud	4.0 A
Odevzdaný výkon	96 W
Krytí	IP20
Provozní teplota	-20...+70 °C
Zbytkové zvlnění	<200 mV
Rozměry š x v x h	90 x 90 x 55 mm
Vyrovňávání síťového napětí	> 40 ms @ 187 V
Způsob elektrického připojení	šroubovací připojení

Tabulka 3. Technické parametry zdroje Siemens LOGO!Power 6EP1332-1SH51

3.3. Tablet

Jako ovládací jednotka byl zvolen tablet od Asus – eeePad Transformer TF 101 (viz. obr. 3).

Tablet zvládá díky dvoujádrovému procesoru NVIDIA Tegra 2 surfování na internetu bleskovou rychlostí a nabízí rychlou odezvu a výkon při více spuštěných aplikacích najednou. 10" IPS displej vyrobený z tvrdého skla odolného proti poškrábání má pozorovací úhly až 178° a zobrazuje jasně všechny barvy. Integrovaná zvuková



Obrázek 3. Tablet ASUS EEE Pad Transformer

technologie SRS Sound se stará o dynamické podání 3D stereo zvuku, zvýraznění basů a široké rozpětí zvuku v podání samostatných reproduktorů umístěných v 13 mm úzkém těle přístroje, který váží pouhých 680 g. Zadní 5Mpx kamera i přední 1,2Mpx kamera umožňují pořizovat fotografie a nahrávat video, které lze přehrát na HD televizi prostřednictvím výstupního portu mini HDMI, což z tohoto tabletu dělá skutečné přenosné multimediální zařízení.[10] Základní technické parametry modulu jsou znázorněny v tabulce číslo 4.

Bez ohledu na programování pro výše uvedený tablet, vytvořený program lze nainstalovat a používat na jakémkoliv zařízení, které podporuje AndroidOS 3 nebo 4 verze bez zvláštních nároků. Seznam kompatibilního zařízení:

- Tablety: Acer Iconia Tab , Asus eeePad, HUAWEI MediaPad, Lenovo IdeaPad, Lenovo ThinkPad, MOTOROLA Xoom, Prestigio Multipad, Samsung Galaxy PAD, Toshiba AT serie a další zařízení s podporou AndroidOS 3.1, 3.2, 4.0
- Mobilní telefony: HTC One serie, HUAWEI Honor, Samsung Galaxy Nexus, Samsung Galaxy Note a další zařízení s podporou AndroidOS 3.1, 3.2, 4.0

Displej	10.1" s LED podsvícením WXGA (1280x800) Screen Podpora vícedotykového ovládání až 10 prsty Sklo odolné proti poškrábání
Procesor	NVIDIA® Tegra™ 2
Operační paměť	1GB
Úložný prostor	16GB
Bezdrátová síť	WLAN 802.11 b/g/n při 2,4 GHz Bluetooth V2.1+EDR
Kamera	Přední kamera s rozlišením 1.2 Mpx Zadní kamera s rozlišením 5 Mpx
Rozhraní	1× kombinovaný audio jack (sluchátka/mikrofon) 1× mini HDMI 1.3a 1× čtečka karet (Micro SD)
Senzor	Senzor pohybu, světelný sensor, gyroskop, elektronický kompas, GPS
Aplikace	Podpora multitaskingu: Ano Podpora Flash: Ano
Baterie	9,5 hodiny; 24,4Wh li-polymerová baterie
Rozměry	271 x 171 x 12.98 mm
Hmotnost	680 g

Tabulka 4. Technické parametry tabletu ASUS EEE Pad

3.4. WiFi směrovač

TP-LINK TL-WR340G (viz. obr. 4). je perfektním řešením pro domácnost, či malou firmu v podobě bezdrátového routeru nabízející možnost sdílení připojení k internetu pomocí čtyř vestavěných ethernetových rozhraní a podpoře standardu IEEE 802,11g, který dosahuje přenosové rychlosti až 54 Mb/s. Bezdrátový signál je šířen pomocí napevno zabudované



Obrázek 4. WiFi směrovač TP-LINK TL-WR340G

3 dBm anténě s maximálním vysílacím výkonem 17 dBm, router pracuje ve frekvenčním pásmu 2,4 GHz.

Samozřejmostí je také podpora mnoha modulací, standardních operačních módů, šifrování WPA, WEP a nechybí ani podpora skrývání SSID a podpora filtrace MAC adres. Základní technické parametry směrovače jsou znázorněny v tabulce číslo 5. [11]

Podporované standardy:	IEEE 802,11g, přenosová rychlost až 54 Mb/s (dynamicky) IEEE 802,11b, přenosová rychlost až 11 Mb/s (dynamicky)
Frekvenční rozsah:	2,4 - 2,4835 GHz
Maximální vysílací výkon:	17 dBm
Modulace:	IEEE 802.11b: DQPSK, DBPSK, DSSS, a CCK IEEE 802.11g: BPSK, QPSK, 16QAM, 64QAM, OFDM
Citlivost:	54M: -68 dBm @ 10 % PER 11M: -85 dBm @ 8 % PER 6M: -88 dBm @ 10 % PER 1M: -90 dBm @ 8 % PER 256K: -105 dBm @ 8 % PER
Bezdrátové módy:	AP Router Mode Bridge mode (point-to-point / point to Multi-point)
Bezpečnost:	Podpora SSID Filtrace MAC adres 64/ 128/ 152 bit WEP šifrování WPA/ WPA2/ WPA-PSK/ WPA2-PSK (AES/TKIP) šifrování
Podporuje:	Flow Statistics Virtual Server, Special Application a DMZ host UPnP, Dynamic DNS, Static Routing, VPN Pass-through ICMP-Flood, UDP, TCP-SYN Flood-filtr Připojení/ odpojení od internetu na určitou denní dobu. Řízení přístupu Upgrade firmware Web management
Funkce:	Vestavěný NAT a DHCP server podporující distribuci statické IP adresy Podpora: PPPoE, Dynamic IP, Static IP, L2TP, PPTP, Guru

	kabelové připojení k internetu Vestavěný firewall s podporou filtrování IP adres, Domain Name filtrování a filtrování MAC adres Podpora WDS bezdrátového mostu
Rozhraní:	4x LAN porty RJ45 10/100 MB/s 1x WAN port RJ45 10/100 MB/s
Anténa:	3 dBi
Napájení:	9 V AC, 0,8 A
Operační teplota:	0 °C ~ 40 °C

Tabulka 5. Technické parametry směrovače

3.5.Krokový motor

Krokový motor – je synchronní bezkartáčový elektropohon, který obsahuje několik vinutí. Postupné aktivace vinutí způsobuje úhlový posun rotoru.

Druhy krokových motorů

Podle konstrukčního provedení se krokové motory rozdělují do čtyř skupin

- Pasivní krokové motory – někdy označované jako reluktanční, krokové motory s proměnnou reluktancí v anglicky psané literatuře jsou uvedeny pod zkratkou VR nebo VRM – Variable reluctance Motors. Rotor tohoto typu krokového motoru je tvořen vzniklými póly z magneticky měkkého materiálu.
- Aktivní krokové motory – neboli krokové motory s aktivním rotorem, někdy označované jako krokové motory s radiálně polarizovaným permanentním magnetem, v anglické literatuře jsou označované zkratkou PM – Permanent Magnet. Rotor tohoto typu je tvořen permanentním magnetem, po obvodu rotoru se střídajícím se severním a jižním pólem permanentního magnetu.
- Hybridní krokové motory – jinak nazývané aktivními krokovými motory s axiálně orientovaným permanentním magnetem v anglické literatuře označené jako HB-Hybrid. Jde o speciální druh aktivních krokových motorů, jejichž rotor je tvořen axiálně uloženým permanentním magnetem, na jehož koncích (severním a jižním pólu) jsou umístěny feromagnetické pólové nástavce. Jedná se tedy o jakousi kombinaci obou předcházejících typů krokového motoru.
- Lineární krokové motory - jsou druhem strojů, které vykonávají nespojitý posuvný pohyb. Stator tohoto motoru je rozvinut do přímky.[12]

Výhody

Nejdůležitější výhodou krokových motorů je přesnost. Další výhodou je cena – přibližně poloviční než u servopohonů.

Krokový motor Mitsumi M35SP-8 (viz. obr. 4)

Vlastnosti:

- Kompaktní rozměry a velký kroutící moment
- Tichost při práci a stabilita[13]

Základní technické parametry motoru jsou znázorněny v tabulce číslo 6.



Obrázek 5. Krokový motor Mitsumi M35SP-8

Jmenovité napětí	DC 24V
Pracovní napětí	DC 21.6~26.4V
Jmenovitý proud/fáze	173mA
Množství fází	4 Phase
Odpor cívky	150Ω/fáze±7%
Délka kroku	7.5°/krok
Metoda řízení	2-2 Phase excitation (Unipolar driving)
Třída izolace	Class E
Statický moment	42.1mN·m
Nominální moment	27.9mN·m/200pps
Max. výstupní otáčky	520pps

Tabulka 6. Technické parametry motoru

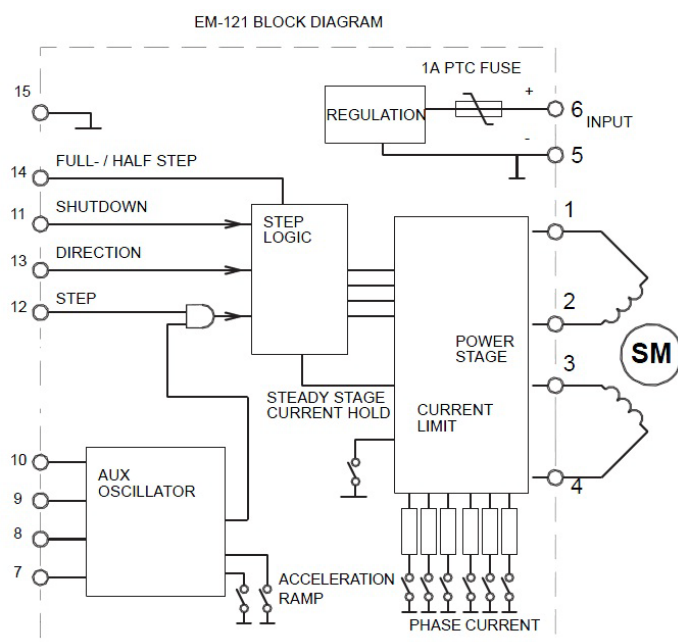
3.6. Řízení krokového motoru

EM-121 (viz. obr. 4) je ovladač bipolárního krokového motoru. Bipolární operace se hodí pro většinu krokových motorů a zajišťuje nejlepší kroutící moment. Fázový proud se nastavuje na požadovanou úroveň pomocí určování proudového limitu.



Obrázek 6. EM-121 Ovladač krokovým motorem

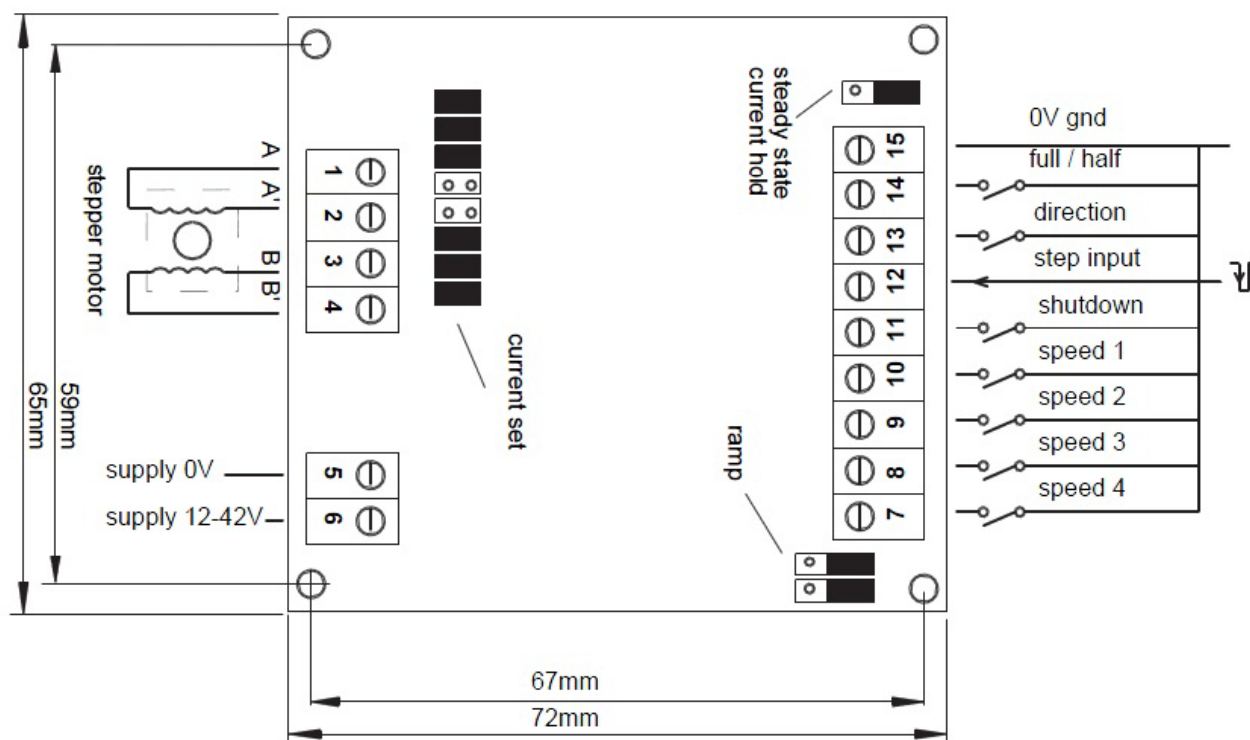
Fázový proud (proudový limit) může být nastaven na sedm různých hodnot pomocí jumperů. Široký rozsah nastavení proudu fáze umožňuje použít přístroj s několika různými motory.



Obrázek 7. Bloková schéma ovladače

Ovladač je vybaven pomocným oscilátorem (viz. obr. 7), který má patnáct přeprogramovatelných frekvencí. Frekvence se nastavuje pomocí čtyř řídicích vstupů. Pomocný oscilátor má zároveň rys zrychlovací rampy, který lze použít ke změně frekvence flexibilně tak, aby se motor startoval spolehlivě i při vysokých frekvencích. Jsou tři intervaly zrychlovací rampy.

Vstupy se řídí pomocí tzv. negativní logiky (viz. obr. 8), to znamená, že vstup se aktivuje, když se propojen k zemi (0 V). Vstupy také pracují i s TTL-logikou. Silová část EM-121 má teplotní ochranu proti přetížení a ochranu proti zpáteční polaritě. Základní technické parametry ovladače jsou znázorněny v tabulce číslo 7. [14]



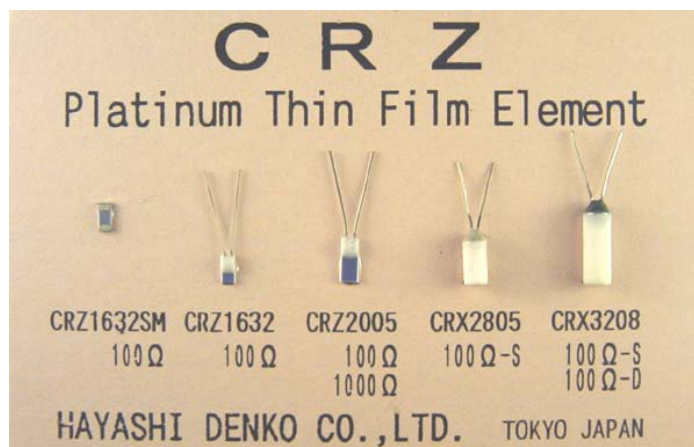
Obrázek 8. Připojení ovladače

Napájení	12-40Vdc
Klidový proud cca	40mA / (20mA shutdown "0")
Nastavení proudu	0.25; 0.37; 0.5; 0.63; 0.75; 0.9; a 1.0A
Ustálený stav I	cca 35 nebo 67% nastaveného proudu
Ztráta napeti	3V pokud $I_m=1A$
Pojistka	1A samoobnova.
Frekvence oscilátoru	20, 50, 100, 200, 300 Hz 400, 600, 800, 1000 Hz 1.5; 2; 2.5; 3; 3.5; 4 kHz
Frekvenční přesnost	1%
Digitální řízení	"off" pokud $U_{in} > 30V$ nebo otevřeno "on" pokud $U_{in} < 1V$
Frekvence kroku	max. 9kHz
Operační teplota	0-50°C
Rozměry	67x75x25mm
Hmotnost	přibližně. 100g

Tabulka 7. Technické parametry ovladače

3.7. Teplotní čidlo

Odporové snímače teploty patří mezi dotykové snímače (snímač je v přímém dotyku s měřeným prostředím - médiem) používané k dálkovému měření teploty. Jejich čidlo (měřicí odpor) převádí teplotní



Obrázek 9. Teplotní čidlo CRZ-2005-1000-B-0.5

změnu prostředí na změnu elektrického odporu. Využívá se přitom toho, že některé (zvláště kovové) materiály mění svůj elektrický odpor v závislosti na změně své teploty.

Vlastnosti odporových čidel teploty lze popsat např. následujícími parametry:

- R_0 , $R_{0,01}$ – základní odpor – hodnota elektrického odporu čidla při teplotě 0 °C, resp. při +0,01 °C (teplota trojného bodu vody),
- R_{100} – hodnota elektrického odporu čidla při teplotě 100 °C,
- R_t - hodnota elektrického odporu čidla při teplotě t °C,
- W_{100} , W_t – poměr odporu při 100 °C, resp. t °C a 0 °C.

Popis čidla CRZ-2005-1000-B-0.5 (viz. obr. 9):

Na keramický substrát je napařena tenká vrstva platiny. Tato vrstva je fotolitograficky strukturována a pomocí laseru je odpor snímače přesně nastaven na jmenovitou hodnotu (Pt100, Pt500, Pt1000). Aktivní vrstva je pasivována izolační vrstvou, která chrání senzor proti vnějšímu chemickému a mechanickému poškození. Kapka fixující dva vývody dlouhé 10 mm je také z keramiky. Vývody jsou z pozlaceného niklu a jsou obdélníkového průřezu. Rozměr keramické destičky záleží na tom, zda jde o Pt100 nebo o Pt500 a Pt1000. U čidel Pt100 je rozměr 2,0x5,0x1,0 mm a 1,6x3,2x1,0 mm u Pt500 a Pt1000 je rozměr 2,0x5,0x1,0 mm. Standardně jsou k dispozici čidla ve třídách přesnosti A a B. Stabilitu čidel popisuje výrobce následovně:

Používáme-li tato čidla průběžně při 400°C po dobu 300 hodin, je případná odchylka od normálu při 0°C do 0,008 W, tzn. maximálně 0,02°C. Výrobce doporučuje používat čidla v rozmezí od - 50°C do + 500°C. V praxi se jeví ideální (z hlediska linearit) pracovní rozsah

od 0°C do 400°C pro třídu A a od 0°C do 500°C pro třídu B. Základní technické parametry čidla jsou znázorněny v tabulkách číslo 8 a 9. [15]

Hodnota při 0°C:	1000 Ohm
Tolerance:	±0.3 °C (±0.12 Ohm)
Doporučovaný rozsah prac. teplot:	-50...+500°C
Měřicí proud:	0.5mA
Vývody:	zlacený nikl
Rozměry:	2x5x1mm

Tabulka 8. Technické parametry čidla

Teplota, °C	Odpor, Ω	Teplota, °C	Odpor, Ω	Teplota, °C	Odpor, Ω
-50	803.1	0	1000.0	50	1194.0
-45	822.9	5	1019.5	55	1213.2
-40	842.7	10	1039.0	60	1232.4
-35	862.5	15	1058.5	65	1251.6
-30	882.2	20	1077.9	70	1270.7
-25	901.9	25	1097.3	75	1289.8
-20	921.6	30	1116.7	80	1308.9
-15	941.2	35	1136.1	85	1328.0
-10	960.9	40	1155.4	90	1347.0
-5	980.4	45	1174.7	95	1366.0

Tabulka 9. Odpor čidla dle teploty

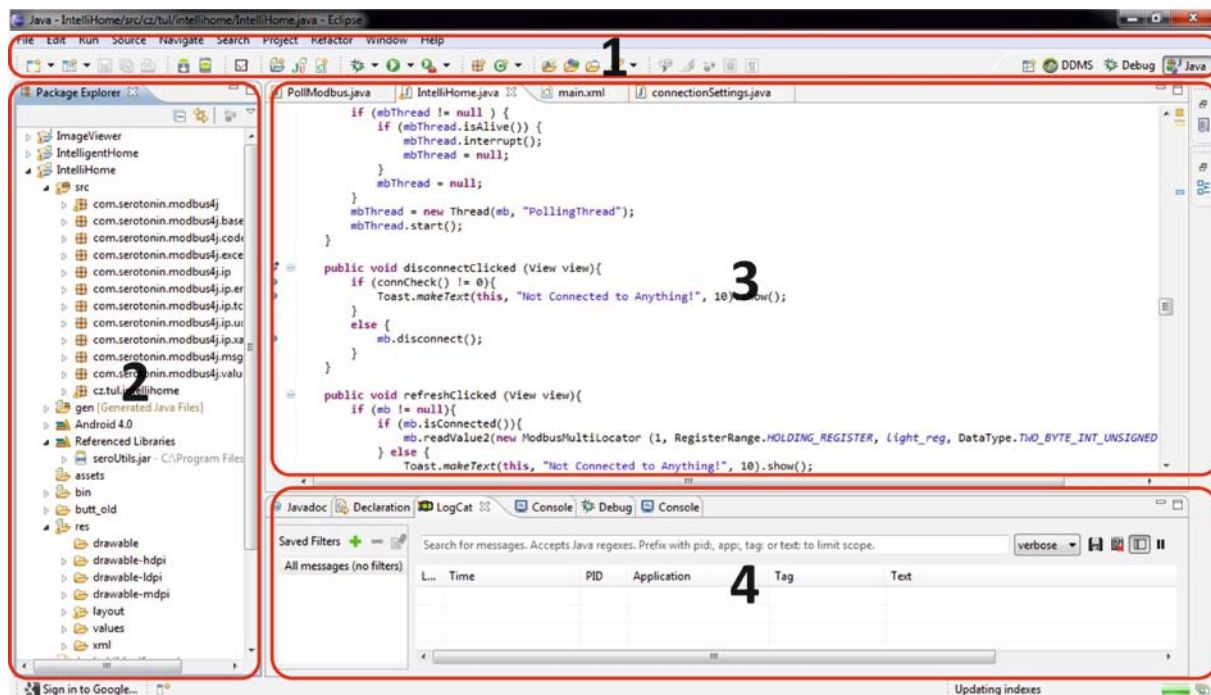
4. Realizace programu v tabletu

4.1. *Vývojové prostředí*

Eclipse (viz. obr. 10) je open source vývojová platforma, která je pro většinu lidí známa jako vývojové prostředí (IDE) určené pro programování v jazyce Java. Flexibilní návrh této platformy dovoluje rozšířit seznam podporovaných programovacích jazyků za pomoci pluginů například o C++ nebo PHP. Právě pluginy umožňují toto vývojové prostředí rozšířit např. o návrh UML, či zápis HTML nebo XML.

Oproti ostatním vývojovým prostředím v Javě, jakými jsou kupříkladu Netbeans, je filozofie Eclipse úzce svázána právě s rozšiřitelností pomocí pluginů. V základní verzi obsahuje Eclipse pouze integrované prostředky pro vývoj standardní Javy jako kompilátor, debugger atd., ale neobsahuje například nástroj pro vizuální návrh grafických uživatelských rozhraní desktopových aplikací nebo aplikační server – všechna taková rozšíření je potřeba dodat formou pluginů. Z tohoto důvodu přímo pod křídly Eclipse vznikly takzvané subprojekty, které zastřešují rozšíření pro jednotlivé oblasti softwarového vývoje v Javě. Tyto subprojekty usnadňují integraci potřebných rozšíření do samotného vývojového prostředí. Eclipse je v současnosti nejpopulárnější IDE pro Javu.

Projekt Eclipse (Eclipse 1.0) vznikl uvolněním kódu IBM pod EPL licencí. Hodnota tohoto příspěvku open source se odhaduje na 40 miliónu dolarů. Pro účely tohoto projektu byl vyvinut grafický framework SWT. Výhodou SWT je nativní vzhled aplikací na každé platformě, kde je SWT portován (SWT využívá nativního kódu operačního systému). Oproti tomu konkurenční framework Swing využívá pouze služby JVM, což umožňuje lepší portovatelnost (omezenou pouze dostupností Javy pro danou platformu).[16]



Obrázek 10. Vývojové prostředí Eclipse

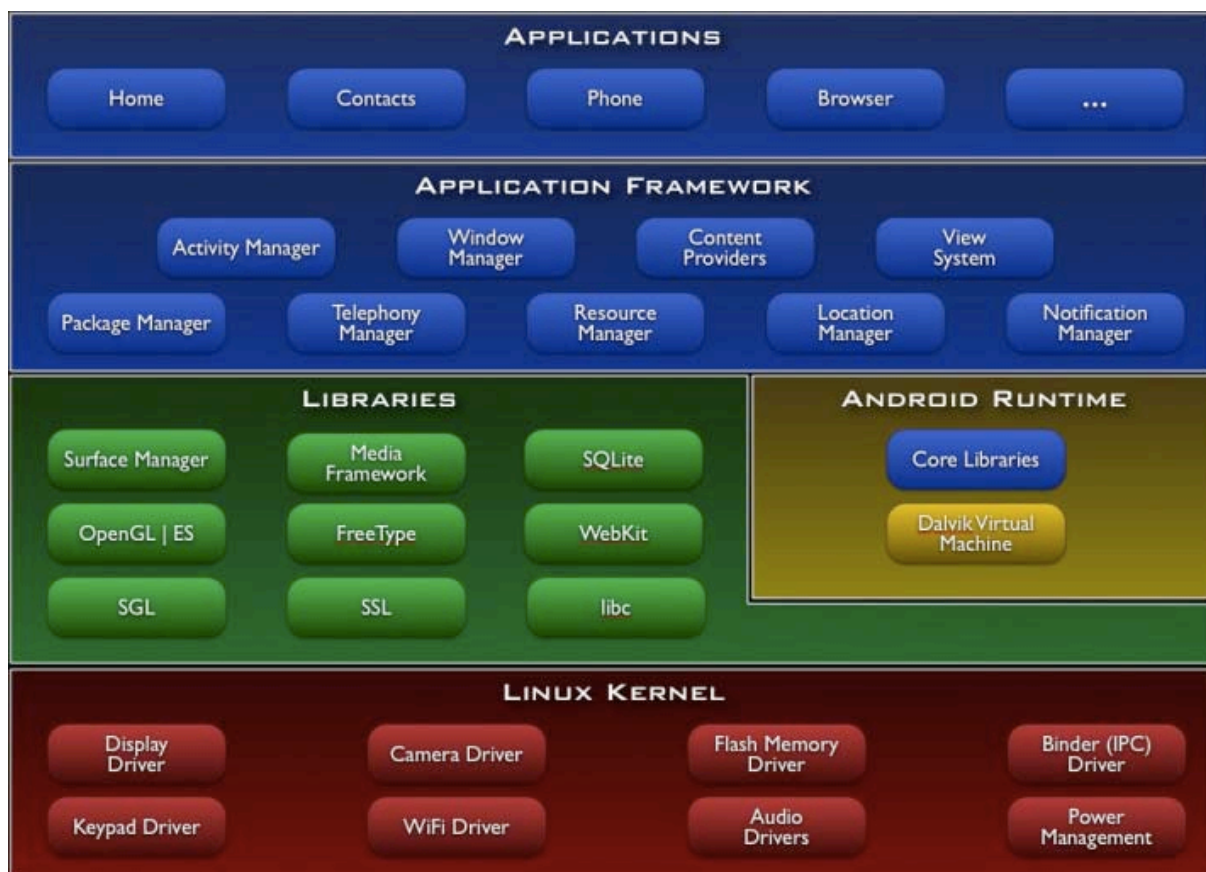
1. Lišta nabídek a nástrojů
2. Manažer projektů
3. Editor kódu
4. Konzole

4.2. Operační systém Android

Google Android je mobilní operační systém pro komunikátory, tablety a multimediální zařízení. Základem je zjednodušené jádro Linuxu. Android OS vyvíjí konsorcium Open Handset Alliance, které bylo založeno Googlem. Android umožňuje vytvářet Java-aplikace, které řídí zařízení přes Google knihovny. Rovněž existuje „Native Development Kit“, který poskytuje možnost vytváření aplikací napsaných v jazyce C a dalších jazycích.[17]

4.2.1. Architektura Android OS

Android OS se rozděluje do čtyř vrstev (viz. obr. 11): Linux Kernel, Libraries/Android Runtime, Application Framework a Applications.



Obrázek 11. Architektura Androidu

4.2.2. Architektura Android-aplikace

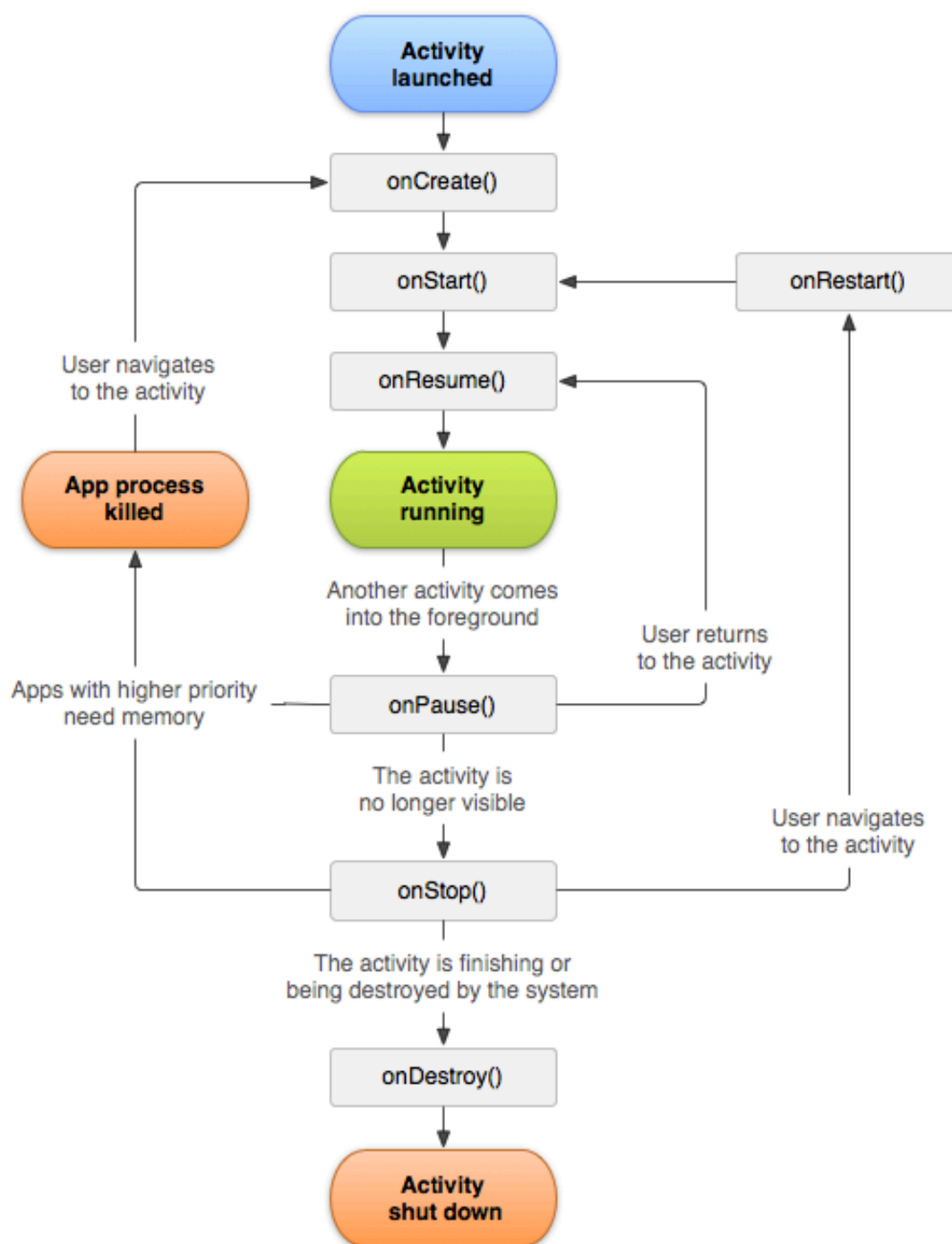
OS Android je založen na možnostech distribuce dat mezi více aplikací na základě opakovaného používání komponentů, které mohou spolupracovat pomocí tzv. intentů. Těmi jsou konkrétně komponenty aktivity (obrazovka), service (akce na pozadí), kontent provider (přístup k datům) nebo broadcast reciever (reakce na příchozí oznámení), který jediný

neumožňuje komunikaci pomocí intentů, přičemž je nutné všechny definovat v kořenovém adresáři, v souboru `AndroidManifest.xml`.

Activity

Jak již bylo zmíněno, activity je komponent odpovídající obrazovce, jenž v sobě skrývá uživatelské grafické rozhraní pro komunikaci s uživatelem. Ten může přepínat i mezi více aktivitami.

Aby byla aktivita zahájena, je nutné, aby se vytvořil nový proces, alokovala se paměť pro obsah uživatelského rozhraní, které je následně rozloženo na obrazovku, jíž zobrazí. Aby byl životní cyklus aktivity efektivní, existuje Activity Manager, který střeží zásobník aktivit, na jehož vrcholu se objevuje aktuálně zobrazovaná aktivita. To zároveň zabraňuje nadbytečnému plýtvání výpočetních prostředků (viz. obr. 12).



Obrázek 12. Životní cyklus aktivity

Service

Například pro připojení k serveru se používá komponent service. Stejně tak je vhodný k vykonávání dalších dlouhotrvajících úkolů a přístupu ke vzdáleným zdrojům. Tento komponent přímo nekomunikuje s uživatelem, nýbrž běží na pozadí. Spustit jej je možné dvěma způsoby. Metoda `startService` umožňuje i samostatné ukončení service, metoda `bindService` je vyvolávána dalším komponentem, a v tom případě je nutné, aby service ukončil klient, který jej spustil. Zároveň je možné na service navázat i více komponent, přičemž se ukončí po jejich celkovém odpojení. Existují tři stavy, ve kterých se service může nacházet:

- Component calls – service se inicializuje zavoláním nebo navázáním komponenty na service.
- Service is running – service vykonává funkci na pozadí.
- Service is shut down – nezpůsobu spuštění byl service ukončen.

Content provider

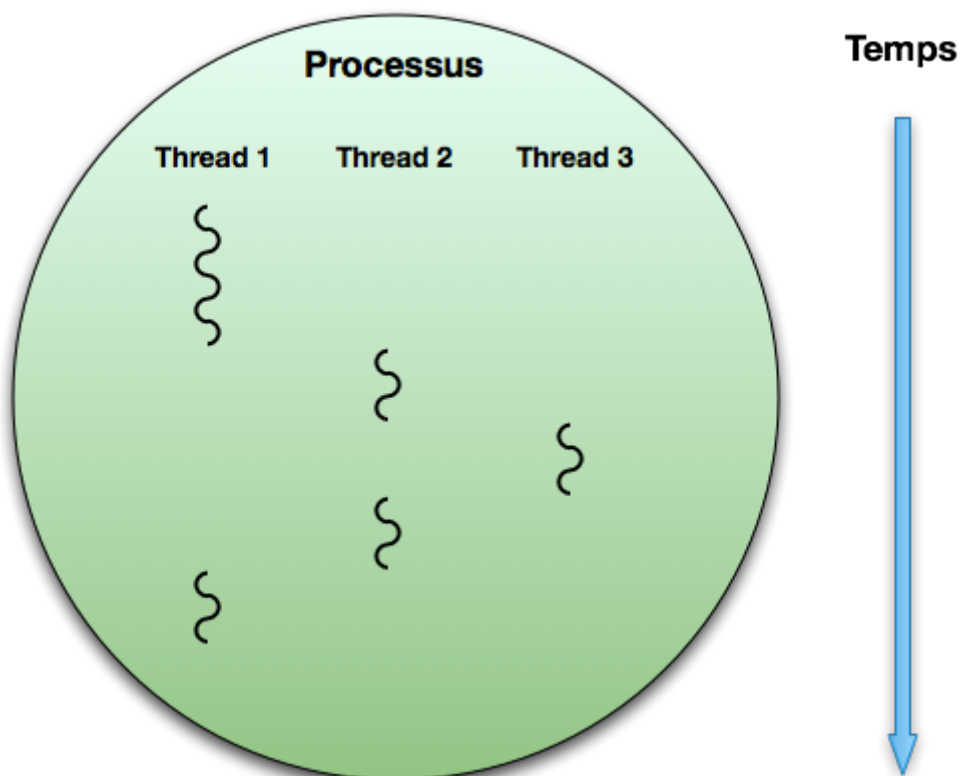
Ke sdílení dat napříč aplikacemi slouží Content provider. Využitelný je ale i ke sdílení dat v rámci jedné aplikace, ale mezi jednotlivými aktivitami. Data jsou uchovávána v souborech, na webu nebo v SQLite databázi, přičemž k těmto datům mají dále přístup i ostatní aplikace (není-li to zakázáno). Tento komponent je založen na stejných metodách jako standardní databázové servery (`insert`, `update`, `delete`, `query`) s elementárním rozhraním. Tímto oddělením dat od uživatelského rozhraní je možné nahrazovat původní aplikace novými – např. díky tomu může fungovat aplikace načítající data uživatelských kontaktů namísto původní aplikace určené pro jejich zobrazování.

Broadcast receiver

Broadcast receiver slouží k příjmu oznámení, na nějž následně reaguje, ať už výpisem ve stavovém řádku nebo vyvoláním dalšího komponentu. Broadcasty jsou buď vytvářeny nově, nebo je využíváno systémových. Stejně jako u service zde neexistuje uživatelské rozhraní. Využíván je např. pro oznamování nízkého stavu baterie, přijetí fotografie nebo doručení textové zprávy.[18]

Thread

Thread je vlákno – tím je myšlen paralelní proces, ve kterém se vytváří osobní seznam instrukcí. V systému, který má jeden procesor, jsou všechna vlákna realizována postupně. Na obrázku 13 je aplikace, která má 3 vlákna:



Obrázek 13. Aplikace se třemi vlákny

4.3. Komunikační protokol ModBus

Modbus je otevřený protokol pro vzájemnou komunikaci různých zařízení (PLC, dotykové displeje, I/O rozhraní, apod.), který umožňuje přenášet data po různých sítích a sběrnicích. Komunikace funguje na principu předávání datových zpráv mezi klientem a serverem (master a slave).

Popis protokolu

Na sběrnici je jedno "master" zařízení (tedy jeden "klient", v případě verze Modbus TCP jich může být více) posílající dotazy, ostatní zařízení jsou "slave" (tedy "server"). "Slave" zařízení odpovídá na dotazy, které jsou mu adresovány. V pozici mastera je tedy řídicí prvek (např. PLC nebo průmyslové PC), v roli slave zařízení jsou ovládané nebo sledované prvky (např. čidla, měřicí přístroje, PLC, prvky výrobních linek atp.).

Struktura rámce

Protokol Modbus definuje strukturu zprávy na úrovni protokolu (*PDU* – Protocol Data Unit) nezávisle na typu komunikační vrstvy. V závislosti na typu sítě, na které je protokol použit, je *PDU* rozšířena o další části, a tvoří tak zprávu na aplikační úrovni (*ADU* – Application Data Unit).

Kód funkce udává, jaký druh operace server (též *slave* – otrok) má provést (viz. tab. 11). Rozsah kódů je 1 až 255, přičemž kódy 128 až 255 jsou vyhrazeny pro oznámení záporné odpovědi (chyby). Některé kódy funkcí obsahují i kód podfunkce upřesňující blíže požadovanou operaci. Obsah datové části zprávy poslané klientem (též *master*) slouží serveru k uskutečnění operace určené kódem funkce. Obsahem může být například adresa a počet vstupů, které má server přečíst nebo hodnota registrů, které má server zapsat. U některých funkcí nejsou pro provedení operace zapotřebí další data, a v tom případě může datová část ve zprávě úplně chybět. Zabezpečení je CRC pro RTU Mode a LRC (kontrolní součet) pro ASCII Mode.

Přenosová media, verze protokolu

- Ethernet přes TCP/IP
- asynchronní sériový přenos (RS-232C, RS-422, RS-485, optické vlákno, radiový přenos)
- MODBUS PLUS vysokorychlostní síť

Adresace

- adresa 0 - broadcast - zpráva je určena všem zařízením, ale žádné na ní neodpovídá
- adresy 1 až 247 - unicast - po přijetí a zpracování požadavku vyše slave odpověď
- adresy 248 až 255 jsou v rezervě

Označení	Význam
<i>Discrete Input</i>	Jeden bit určený pouze ke čtení, např. binární vstup.
<i>Coil</i>	Jeden bit, který lze číst i zapisovat, např. cívka relé, lze ji ovládat i zjišťovat její stav.
<i>Input Register</i>	16bitový registr určený pouze ke čtení, např. analogový vstup.
<i>Holding Register</i>	16bitový registr, který lze číst i zapisovat, např. čítač, lze jej nastavit i číst jeho hodnotu.

Tabulka 10. Typ přenášených dat

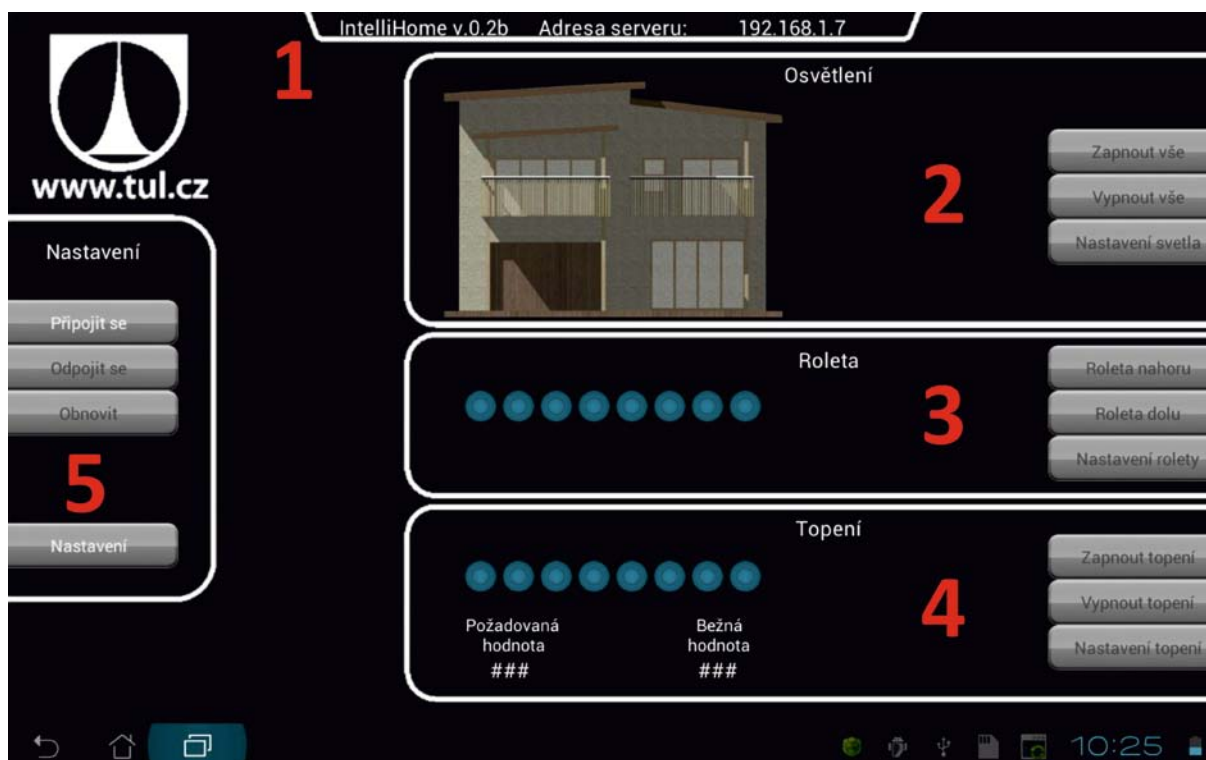
Kód	Název funkce	Popis
01	<i>Read Coils</i>	Čtení jednoho nebo více bitů
02	<i>Read Discrete Inputs</i>	Čtení jednoho nebo více bitů
03	<i>Read Holding Registers</i>	Čtení jednoho nebo více 16bitových registrů
04	<i>Read Input Registers</i>	Čtení jednoho nebo více 16bitových registrů
05	<i>Write Single Coil</i>	Zápis jednoho bitu
06	<i>Write Single Register</i>	Zápis jednoho 16bitového registru
15	<i>Write Multiple Coils</i>	Zápis více bitů
16	<i>Write Multiple Registers</i>	Zápis více 16bitových registrů

Tabulka 11. Kódy základních funkcí

V praxi někdy splývají pojmy *Coils* a *Discrete Inputs* (respektive *Holding Registers* a *Input Registers*), záleží pouze na výrobci zařízení, jak interpretuje tyto pojmy. V takových případech lze v případě čtení zaměnit funkce 01 a 02 (respektive 03 a 04).[19]

4.4.Aplikace „IntelliHome“

Aplikace je schopna řídit se současným stavem parametrů, jakými jsou např. osvětlení, topení nebo poloha rolety. Aplikace obsahuje hlavní okno (viz. obr. 14) a 4 okna pro nastavení parametrů.



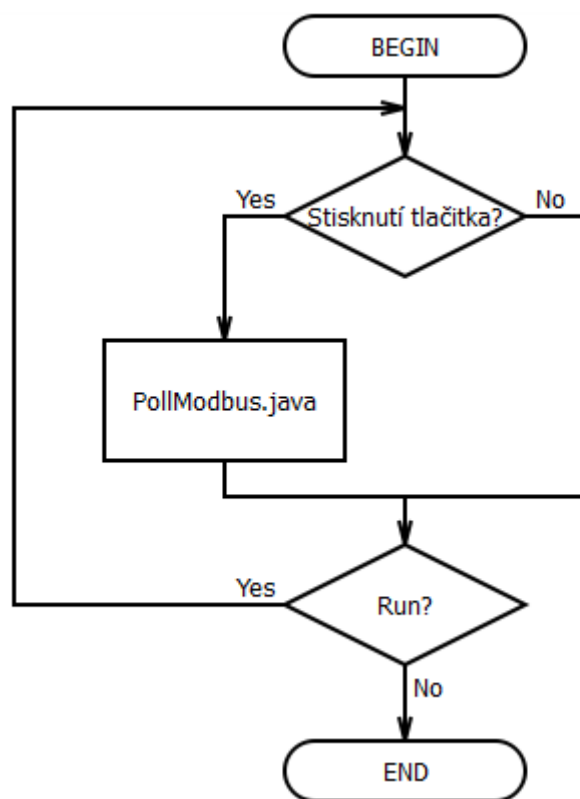
Obrázek 14. Hlavní okno aplikace

Hlavní okno obsahuje 5 částí:

1. Informační oblast – ukazuje současnou verzi aplikace a nastavení serveru
2. Oblast osvětlení – ukazuje stav lamp a obsahuje možnosti pro zapnutí a vypnutí všech lamp a nastavení automatického řízení osvětlení.
3. Oblast rolety – ukazuje stav rolety a obsahuje možnosti vytažení a stažení rolety a nastavení automatického řízení rolety.
4. Oblast topení – ukazuje stav topení a obsahuje možnosti pro zapnutí a vypnutí topení a nastavení automatického řízení topení.
5. Oblast nastavení – obsahuje tlačítka pro připojení/odpojení k/od serveru, tlačítko pro obnovení zobrazení stavu domu a tlačítko pro nastavení připojení k serveru.

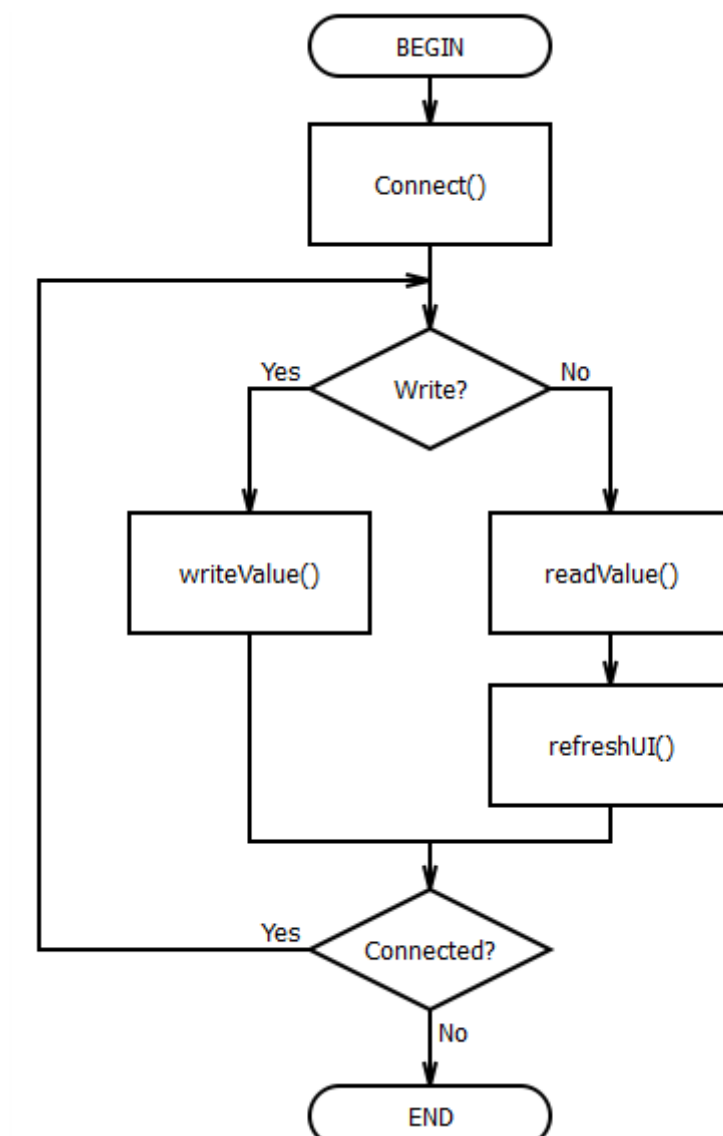
4.4.1. Běh aplikace

Na následujícím diagramu (viz. obr. 15) je zjednodušeně znázorněna základní rutina aplikace.



Obrázek 15. Hlavní aktivita aplikace

Po stisknutí tlačítka začíná pracovat druhé vlákno aplikace, ve kterém probíhá komunikace se serverem (viz. obr. 16).



Obrázek 16. Vlákno PollModbus

4.4.2. Třídy aplikace

Aplikace je rozdělena do 12 balíčků. Balíčky s názvem, který začíná „com.serotonin.modbus4j“ jsou knihovnou modbusu pro Java. Je to otevřená knihovna, která se šíří dle licence GPL. Modbus4j je vysokovýkonný a jednoduchý k používání implementací protokolu Modbus. Napsaná je v jazyce Java firmou Serotonin Software. Podporuje ASCII, RTU, TCP a UDP dopravu jako slave (server) a master (klient). Rovněž podporuje automatické rozdělení dotazů a rozbor tvaru odpovědi.

Balíček s názvem cz.tul.intellihome obsahuje 6 tříd specifických pro tuto aplikaci. Tady je jejich krátký popis:

1. connectionSettings.java – tato třída zobrazuje okno nastavení připojení. Umožňuje změnit ip adresu a port pro připojení (viz. obr. 17).

```
@Override
protected void onResume() {
    super.onResume();
    getPreferenceScreen().getSharedPreferences().registerOnSharedPreferenceChangeListener(this);
}

@Override
protected void onPause() {
    super.onPause();
    getPreferenceScreen().getSharedPreferences().unregisterOnSharedPreferenceChangeListener(this);
}

public void onSharedPreferenceChanged(SharedPreferences sharedPreferences, String key) {
    if (key.equals(IP_ADDRESS_PREFERENCE))
    {
        IPAddressPreference.setSummary(sharedPreferences.getString(key, "Nastavte novou IP adresu"));
    }
    else if (key.equals(PORT_PREFERENCE))
    {
        PortPreference.setSummary(sharedPreferences.getString(key, "Nastavte nový port (implicitní = 502)"));
    }
}
```

Obrázek 17. Ukázka kódu - zadání nastavení

2. IntelliHome.java - základní třída, která se spustí jako první po instalaci. Obsahuje všechna tlačítka pro řízení chytrého domu a zobrazuje stav systému (viz. obr. 18, 19 a 20).

```
lamp1 = (ToggleButton) findViewById(R.id.light1);
lamp2 = (ToggleButton) findViewById(R.id.light2);
lamp3 = (ToggleButton) findViewById(R.id.light3);
lamp4 = (ToggleButton) findViewById(R.id.light4);
lamps_all_on = (Button) findViewById(R.id.all_lights_on);
lamps_all_off = (Button) findViewById(R.id.all_lights_off);
Connect = (Button) findViewById(R.id.Connect);
Disconnect = (Button) findViewById(R.id.Disconnect);
settings = (Button) findViewById(R.id.settings);
refresh = (Button) findViewById(R.id.refresh);
lihgts_settings = (Button) findViewById(R.id.lihgts_settings);
blind_up = (Button) findViewById(R.id.blind_up);
blind_down = (Button) findViewById(R.id.blind_down);
blind_settings = (Button) findViewById(R.id.blind_settings);
heating_on = (Button) findViewById(R.id.heating_on);
heating_settings = (Button) findViewById(R.id.heating_settings);
heating_off = (Button) findViewById(R.id.heating_off);
roleta = (RatingBar) findViewById(R.id.ratingBar1);
teplota = (RatingBar) findViewById(R.id.ratingBar2);
req_temp = (TextView) findViewById(R.id.textView2);

addListenerOnRatingBar();
addListenerOnRatingBar2();
```

Obrázek 18. Ukázka kódu - inicializace elementů interfacu

```
public void connectClicked (View view){
    if (mb == null) {
        mb = new PollModbus(mbTCPMaster, 0, mbLocator, pollHandler, cbHandler/*, checkBox0*/);
    }
    else if (mb.isConnected()) {
        mb.disconnect();
    }

    if (mbThread != null ) {
        if (mbThread.isAlive()) {
            mbThread.interrupt();
            mbThread = null;
        }
        mbThread = null;
    }
    mbThread = new Thread(mb, "PollingThread");
    mbThread.start();
}

public void disconnectClicked (View view){
    if (connCheck() != 0){
        Toast.makeText(this, "Not Connected to Anything!", 10).show();
    }
    else {
        mb.disconnect();
    }
}
}
```

Obrázek 19. Ukázka kódu – zpracování připojení a odpojení k/od serveru

```

public void onLight1Clicked(View view){
    mbWriteValue = LightValueCreation();
    mb.writeValue(new ModbusMultiLocator (1, RegisterRange.HOLDING_REGISTER, Light_reg, DataType.TWO_BYTE_INT_UNSIGNED, 1)
    , mbWriteValue);
}

public void onLight2Clicked(View view){
    mbWriteValue = LightValueCreation();
    mb.writeValue(new ModbusMultiLocator (1, RegisterRange.HOLDING_REGISTER, Light_reg, DataType.TWO_BYTE_INT_UNSIGNED, 1)
    , mbWriteValue);
}

```

Obrázek 20. Ukázka kódu – zpracování stisknutí tlačítek

3. ModbusMultiLocator.java – je rozšířením třídy ModbusLocator.java z balíčku com.serotonin.modbus4j pro komfortnější nastavení údajů modbus dotazu.
4. ModbusTCPFactory.java – třída pro vytváření TCP připojení se serverem.
5. ModbusTCPMaster – třída pro vytváření TCP připojení se serverem.
6. PollModbus.java – je třída pro připojení, odpojení, posílání dotazů a zpracování odpovědí (viz. obr. 21, 22 a 23).

```

public synchronized void connect() throws Exception {
    if (this.isConnected()) {
        this.disconnect();
    }
    try {
        mbTCPMaster.init();
    }
    catch (ModbusInitException initException) {
        Log.e(getClass().getSimpleName(), initException.getMessage() );
        m_connected = false;
        throw initException;
    }
    catch (Exception e) {
        Log.e(getClass().getSimpleName(), e.getMessage() );
        m_connected = false;
        throw e;
    }

    m_connected = true;

    Message m = this.mainThreadHandler.obtainMessage();

    m.arg1 = 1;
    mainThreadHandler.sendMessage(m);
}

```

Obrázek 21. Ukázka kódu – provedení připojení se serverem

```

public synchronized void disconnect() {
    Message m = this.mainThreadHandler.obtainMessage();
    if (m_connected || mbTCPMaster.isInitialized() ) {
        if (DEBUG)
            Log.i(getClass().getSimpleName(), "Try to destroy connection");
        mbTCPMaster.destroy();
        if (DEBUG)
            Log.i(getClass().getSimpleName(), "Destroyed connection" );
        m.arg2 = 0;
    }
    else {
        m.arg2 = 1;
        if (DEBUG)
            Log.i(getClass().getSimpleName(), "Tried to destroy connection, but nothing was there!" );
    }

    m_connected = false;

    m.arg1 = 0;
    mainThreadHandler.sendMessage(m);
}

```

Obrázek 22. Ukázka kódu – provedení odpojení od serveru

```

if (doReadValue2) {

    readValue = mbTCPMaster.getValues(mbLocator);

    test0.post(new Runnable(){
        public void run(){
            test0.setText(readValue[0].toString() + readValue[1].toString() + readValue[2].toString());
        }
    });
    doReadValue2 = false;
}

// after we call writeValue from the UI thread,
// then we call notify() and this code gets processed
if (doWriteValue) {
    if (DEBUG)
        Log.i(getClass().getSimpleName(), "Writing Value: " + writeValue );
    mbTCPMaster.setValue(mbWriteLocator, writeValue);
    doWriteValue = false;
}
}

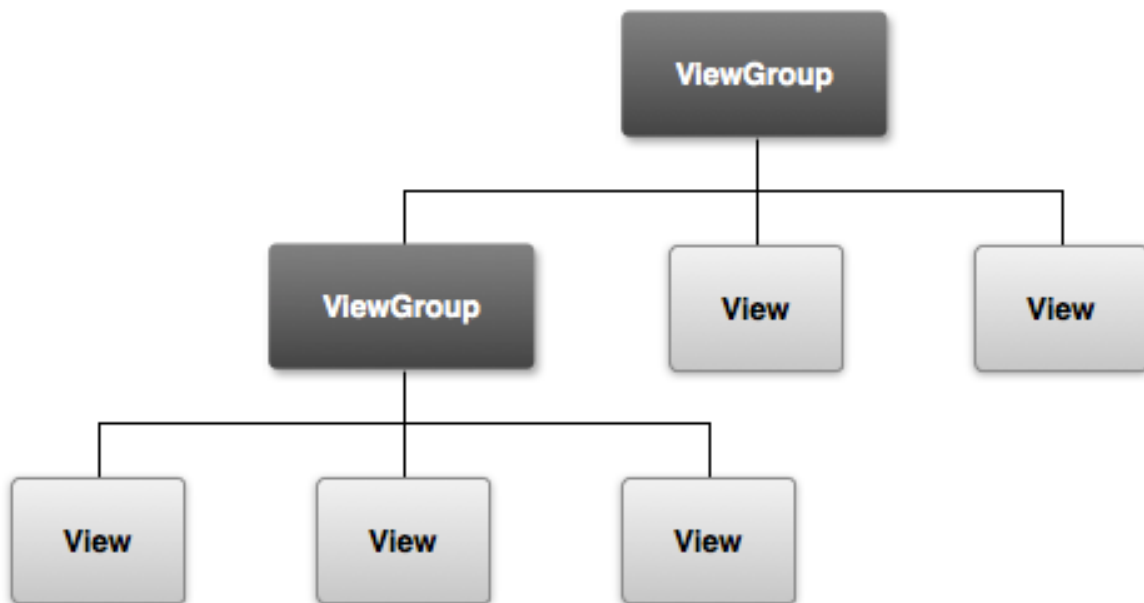
```

Obrázek 23. Ukázka kódu – čtení a psaní registrů dle adresy

4.5. Vytváření interfacu

XML neboli Extensible Markup Language (rozšiřitelný značkovací jazyk) je založen na principu jazyka SGML v jednodušší podobě. Vyvinulo a standardizovalo jej konsorcium W3C. V tomto jazyce se dají vytvářet aplikace (konkrétní značkovací jazyky) určené k různým účelům se zastoupením různých typů dat. Používán bývá pro serializaci dat stejně jako např. JSON či YAML. XML a jeho zpracování podporuje řada nástrojů a programovacích jazyků. Jeho účelem je zejména výměna dat mezi aplikacemi a publikace dokumentů, u nichž popisuje věcný obsah, nikoliv vzhled. Ten může být definován kaskádovými styly. Dále existuje možnost zpracování transformací do jiného typu dokumentu nebo další aplikace XML.

Jako nejčastější způsob, jak definovat rozložení aplikace a její hierarchii je rozložení souboru XML. Poskytuje srozumitelnou strukturu rozvržení srovnatelně s HTML. Každý prvek v XML zobrazuje objekt (Viewgroup), konkrétně stromy nebo odbočky (viz uvedený obrázek 24).



Obrázek 24. Hierarchie XML souboru

Název daného prvku XML je určen příslušnou třídou Java. Stejně jako <TextView> prvek vytváří TextView rozhraní, prvek <LinearLayout> vytváří skupinu názorů. Při načítání rozložení zdrojů OS Android inicializuje tyto objekty run-time.

Existuje několik způsobů, jak rozložit názory. Použitím více různých druhů zobrazení skupin lze strukturovat názory potomků a zobrazit skupiny nekonečně mnoha způsoby. Zde je uvedeno některé předdefinované zobrazení skupiny nabízené Androidem – ten je nazývá rozložení: `LinearLayout`, `RelativeLayout`, `TableLayout`, `GridLayout`, aj. Každý z nich nabízí jedinečnou sadu parametrů rozložení, které definují názory potomků a rozložení struktury.[20] Pro vytváření interfacu aplikace IntelliHome jsem použil `RelativeLayout` (viz. obr. 25), protože to je nejpoužívanější způsob a umožňuje vytvářet interface pro tablety a pro telefony.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="horizontal"
    android:background="@drawable/fon" >

    <ToggleButton
        android:id="@+id/light1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="460dp"
        android:layout_marginTop="70dp"
        android:background="@drawable/togg_but1_states"
        android:onClick="onLight1Clicked"
        android:text=""
        android:textOff=""
        android:textOn="" />

    <ToggleButton
        android:id="@+id/light2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="657dp"
        android:layout_marginTop="70dp"
        android:background="@drawable/togg_but2_states"
        android:onClick="onLight2Clicked"
        android:text=""
        android:textOff=""
        android:textOn="" />
```

Obrázek 25. Ukázka kódu – začátek souboru `main.xml`

4.5.1. Soubor manifest.xml

Každá žádost musí v kořenovém adresáři obsahovat soubor AndroidManifest.xml (s přesně daným názvem). Manifest zobrazuje základní informace o aplikaci Android (viz. obr. 26), informační systém musí běžet před tím, než je možné spustit některý z kódu aplikace. Manifest má mimo jiné následující funkce:

- Pojmenuje balíček Java aplikace. Název balíčku slouží jako jedinečný identifikátor pro aplikaci.
- Popisuje komponenty aplikace – činnosti, služby, přijímače vysílání a obsahu poskytovatelů. Aplikace sestává z názvu třídy, který implementuje všechny komponenty a publikuje své schopnosti. Tato prohlášení systému počítají s komponenty a vědí, za jakých podmínek můžou být spuštěny.
- Určuje, které procesy bude hostit aplikace komponentů.
- Deklaruje oprávnění, která aplikace musí mít, aby měla přístup k chráněné části rozhraní API a k interakci s jinými aplikacemi.
- Rovněž deklaruje oprávnění, které musí mít ostatní součásti, aby mohly interaktivně pracovat se součástmi aplikace.
- Zobrazí se seznam Instrumentation třídy, která poskytuje profilace a další informace, jakmile je aplikace spuštěna. Tato prohlášení jsou přítomna v manifestu jen tehdy, pokud jsou aplikace vyvinuty a testovány. K odstranění dojde před publikováním aplikace.
- Deklaruje minimální úroveň Android API, který aplikace vyžaduje.
- Zobrazí se seznam knihoven, se kterými aplikace musí být propojena. [20]

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="cz.tul.intellihome"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="13" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"></uses-permission>
    <uses-permission android:name="android.permission.INTERNET"></uses-permission>

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".IntelliHome"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity android:name="connectionSettings" android:label="@string/Conn_Settings">
        </activity>

    </application>

    <uses-permission android:name="android.permission.INTERNET"></uses-permission>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"></uses-permission>
    <uses-permission android:name="android.permission.CHANGE_WIFI_STATE"></uses-permission>
    <uses-permission android:name="android.permission.CHANGE_NETWORK_STATE"></uses-permission>
    <uses-permission android:name="android.permission.WRITE_APN_SETTINGS" ></uses-permission>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" ></uses-permission>

</manifest>
```

Obrázek 26. Ukázka kódu – souboru AndroidManifest.xml

5. Realizace programu v PLC

5.1. Vývojové prostředí

Mosaic je vývojové prostředí (viz. obr. 29) vytvořené v roce 2000, určené pro tvorbu nebo ladění programů pro programovatelné logické systémy (PLC, Programmable Logic Controller) TECOMAT a TECOREG z produkce firmy Teco a.s. Kolín. Prostředí je vyvíjeno ve shodě s mezinárodní normou IEC EN-61131-3, ta definuje strukturu programů a programovací jazyky pro PLC.[4]

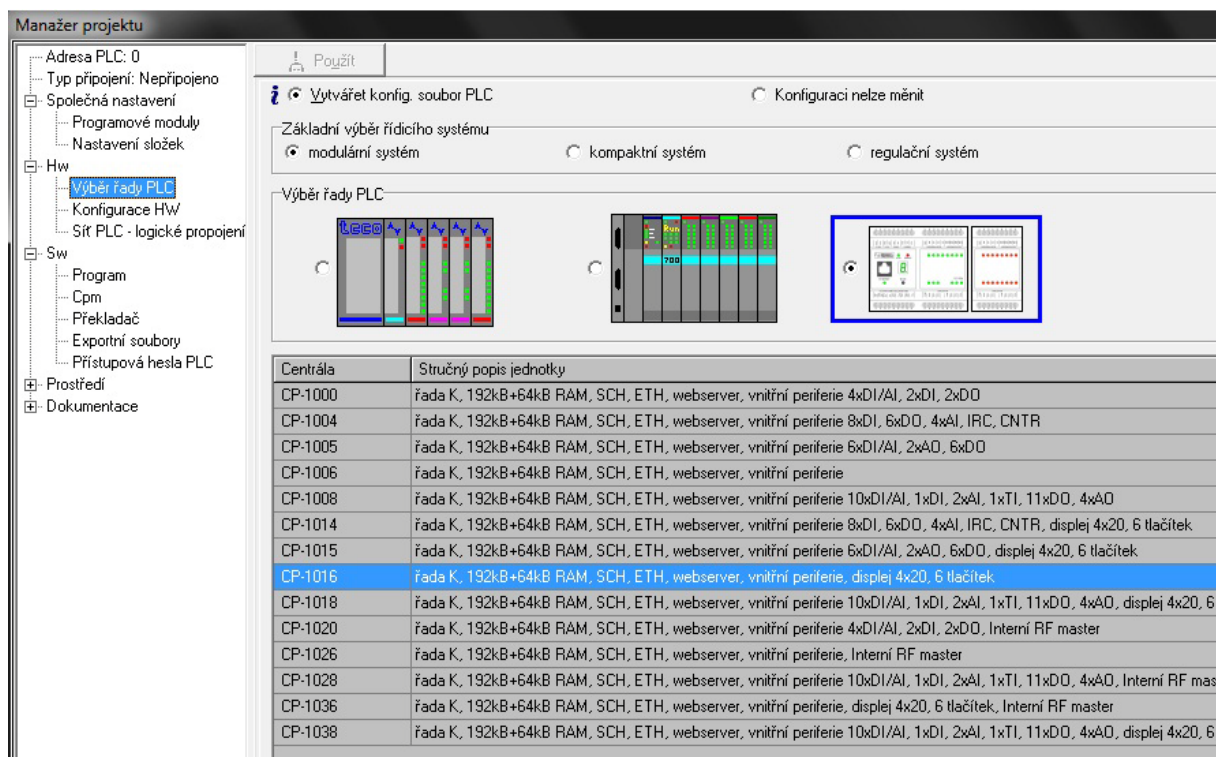
Nástroje prostředí Mosaic:

1. Nástroje pro automatické generování kódu částí programu

Všechny zdrojové kódy uživatelského programu je možné psát přímo jako text. Usnadnění práce a snížení rizika chyb v prostředí Mosaic představují nástroje, které některé činnosti zjednodušují a samy jsou schopny automaticky vygenerovat zdrojový text.

Některé tyto nástroje je možné zapisovat jak textově, tak graficky. Pracují tedy obousměrně. To je i případ IEC manažera. Ostatní nástroje ale pracují pouze „jednosměrně“ a vygenerují automaticky jen zdrojový text. Výsledné soubory jsou v seznamu souborů pro překlad v projektu označeny ikonou a nelze je v textové formě editovat, mají nastaven atribut „read only“ a vždy se znovu obnovují podle nastavení nástroje.

K definování typu PLC, jeho sestavení a nastavení funkcí jednotlivých modulů PLC je určen *Manažer Projektu* (viz. obr. 27). Dále je schopen nastavit obecné funkce SW, driverů pro komunikaci, vzájemné propojení dat mezi jednotlivými projekty PLC navzájem, a také propojení k textovým operátorským panelům, jež jsou zahrnuty do této skupiny projektů. Otevírá se klepnutím na ikonu nebo z nabídky menu Projekt a je pak implicitně otevřen do plovoucího okna vždy na vrchu. Generována je automaticky část kódu programu s informacemi o konfiguraci systému uložené v souborech *.hwc, *.hwn, HWConfig.st a dalších.



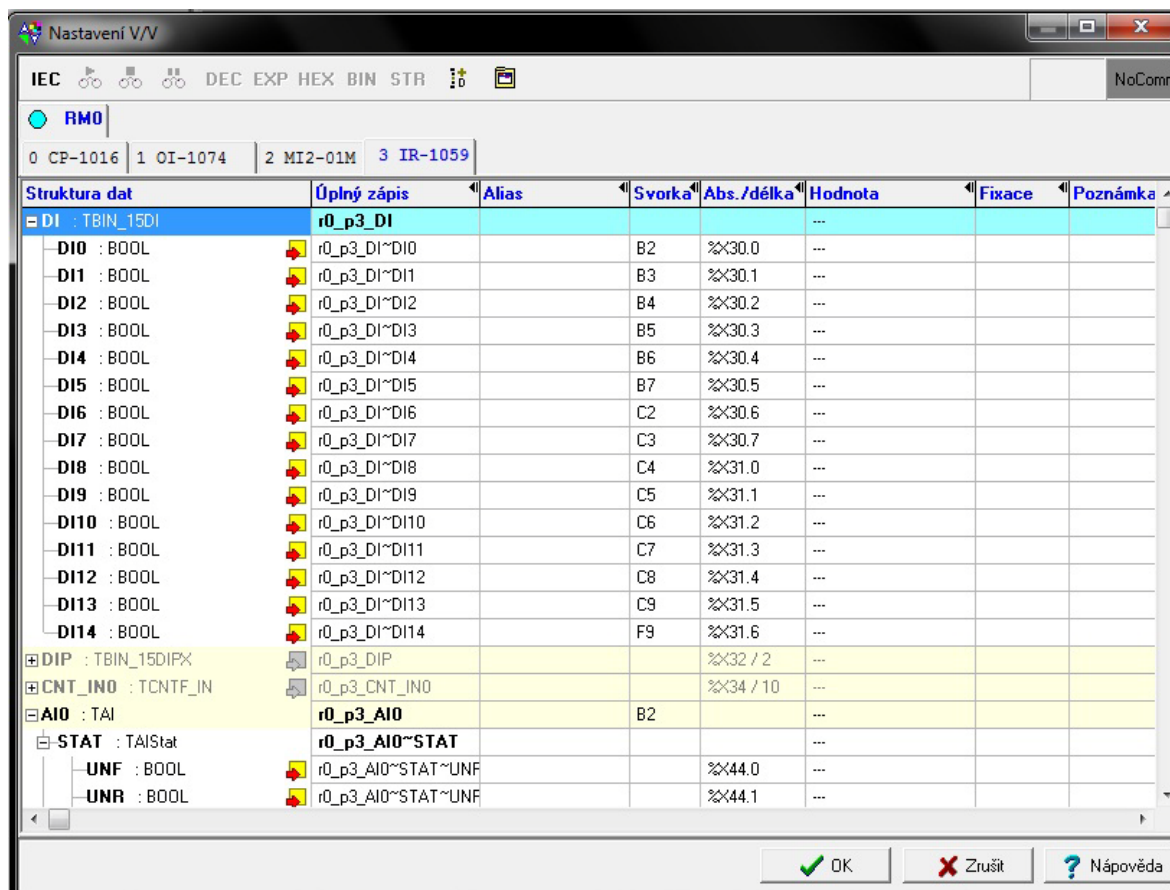
Obrázek 27. Manažer projektu

2. Nastavení vstupů/výstupů (aliasy, data a fixace I/O)

Okno zobrazuje data vstupů a výstupů, umožňuje vstupním a výstupním signálům přiřadit jména (aliasy), umožňuje během ladění programu zafixovat hodnoty vstupů a výstupů do libovolných stavů, zobrazuje po překladu výsledné absolutní adresy vstupů a výstupů a umožňuje jim je přiřadit. Otevírá se klepnutím na ikonu a je implicitně otevřen do plovoucího okna. (viz. obr. 28)

IEC manažer je určen pro organizaci a editaci položek v uživatelském programu podle IEC 61131-3. IEC manažer je otevírán automaticky a je implicitně zadokován do levého panelu. Je rozdělen do několika záložek:

- POU - programovatelné organizační jednotky
- Typy - typy proměnných
- Globální proměnné - globálně dostupné proměnné
- Konfigurace - organizace úloh a instancí v programu
- Knihovny - přehled zařazených knihoven a jejich obsahu



Obrázek 28. Nastavení vstupů/výstupů

3. Textové editory uživatelského programu

Textový editor ST je používán pro jazyk „Strukturovaný text“, dále ST. Editor zajišťuje barevné zvýraznění podle jazykové syntaxe a nástroje pro editaci. Otevírá se implicitně zadokován do hlavního panelu spolu se všemi soubory s příponou *.ST.

Textový editor IL je používán pro jazyk „Instrukční list“ IL. Editor zajišťuje barevné zvýraznění podle jazykové syntaxe. Otevírá se implicitně zadokován do hlavního panelu spolu se všemi soubory s příponou *.IL.

Textový editor Txt je používán pro editaci obecných textových souborů bez zvýraznění. Implicitně se otevírá zadokován do hlavního panelu spolu se všemi soubory s příponou *.txt.

4. Grafické editory uživatelského programu

Editor LD je používán pro grafický jazyk příčkových diagramů s reléovými kontakty. Implicitně se otevírá zadokován do hlavního panelu spolu se všemi soubory s příponou *.LD.

Editor FBD je používán pro grafický jazyk funkčních bloků. Implicitně se otevírá zadokován do hlavního panelu spolu se všemi soubory s příponou *.FBD.

5. Další nástroje pro automatické generování kódu částí programu

PIDMaker je vizuální nadstavba nad PID a PIDMA instrukce PLC. Slouží k snadné implementaci, ladění a správě regulačních algoritmů. Otevírá se klepnutím na ikonu a je implicitně zadokován do levého panelu. Generuje automaticky část kódu programu s PID regulátory.

PanelMaker je určen pro definici obsahu obrazovek pro textové operátorské panely. Otevírá se klepnutím na ikonu a je implicitně zadokován do hlavního panelu. Generuje automaticky část kódu programu pro obsluhu textových panelů HMI.

Grafický PanelMaker je určen pro definici obsahu obrazovek pro grafické operátorské panely. Otevírá se klepnutím na ikonu a je implicitně zadokován do hlavního panelu. Generuje automaticky soubory s popisy obrazovek pro obsluhu grafických panelů HMI.

6. Nástroje pro řízení projektu

Skupiny projektů zobrazují všechna jména projektových skupin v aktuálním adresáři a jména v nich obsažených projektů. Umožňuje snadné přepínání mezi projekty. Otevírá se automaticky a je implicitně zadokován do levého panelu.

Nástroj *Soubory v projektu* je určen pro přehled souborů projektu, které jsou zařazeny do překladu, a dovoluje měnit jejich pořadí pro překlad. Je možné zde ručně přesouvat, přidávat a odebírat soubory v projektu. Obvykle jsou soubory automaticky zařazovány do projektu jinými nástroji, např. IEC manažerem. Nástroj se otevírá automaticky a je implicitně zadokován do levého panelu.

Nástroj *Otevřené soubory* zobrazuje seznam otevřených souborů a cest jejich umístění a ve spodní polovině je zobrazen seznam plovoucích oken. Otevírá se automaticky a je implicitně zadokován do levého panelu.

7. Nástroje pro ladění a simulaci

POU Inspektor slouží pro základní náhled na program, když je PLC v režimu RUN. V podstatě je speciálním režimem okna editorů. Zdrojový program je animován hodnotami aktuálních dat tak, aby programátor mohl sledovat správnost zapsané funkce. Vytváří se přímo v aktivním okně na místě editoru.

WebMaker slouží k tvorbě XML stránek pro webový server v centrálních jednotkách a základních modulech, které tuto funkci podporují. Využitelný je i ke zobrazování a nastavování proměnných přímo v MOSAICu. Dalším využitím je použít jej jako jednoduché vizualizace pro odlaďování algoritmu v simulaci v MOSAICu. Otevírá se klepnutím na ikonu a je implicitně zadokován do hlavního panelu.

GraphMaker slouží ke grafickému zobrazení až 16-ti průběhů proměnných PLC ve formě časového grafu. Má dva režimy:

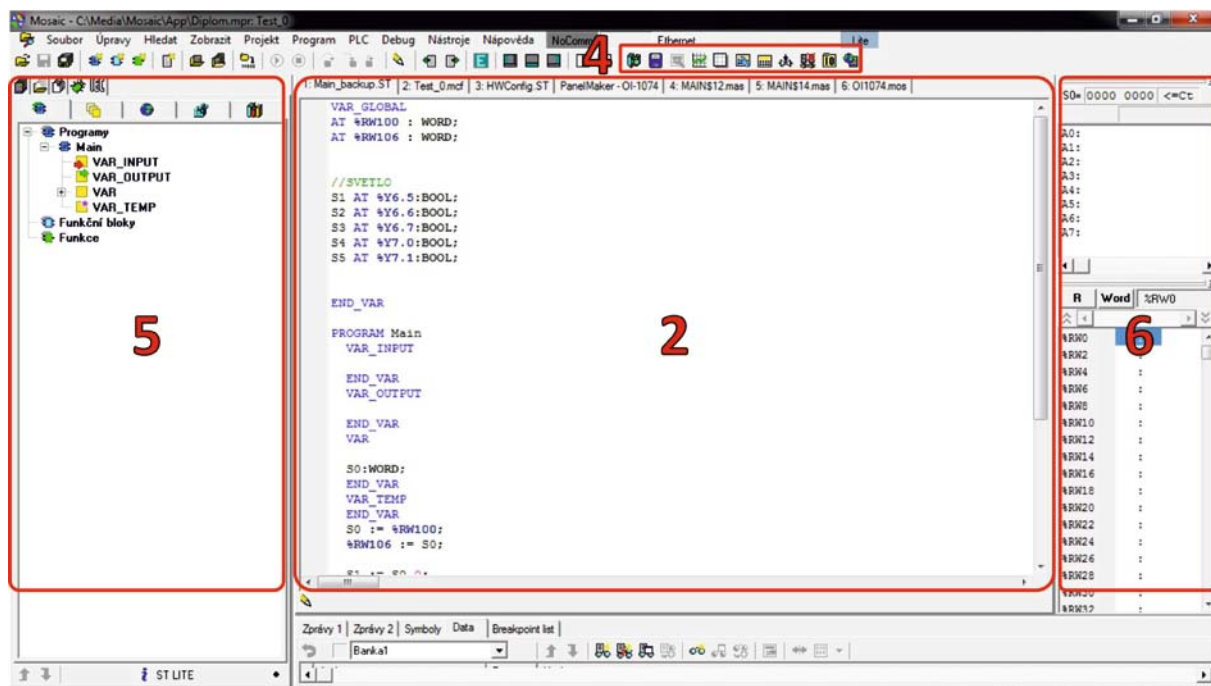
- paměťový osciloskop
- logický a signálový analyzátor

s maximálním rozlišením na jeden cyklus otáčky programu PLC. Otevírá se klepnutím na ikonu a je implicitně zadokován do hlavního panelu.

Simulátor textových panelů (HMI) slouží k testování programové obsluhy operátorského panelu bez připojeného HW. Otevírá se klepnutím na ikonu a je implicitně otevřen do plovoucího okna. Oknu je vhodné nastavit funkci ‚Vždy nahoře‘, a to pravým tlačítkem myši na horní liště okna. Nástroj je možné konfigurovat klepnutím pravým tlačítkem myši v ploše okna.

Nástroj Panel je určen pro semigrafické zobrazování a nastavování proměnných v programu. Pracuje jako jednoduchá vizualizace a je vhodný pro odlaďování algoritmu v simulaci. Otevře se z menu Soubor/Nový/Nový panel. Implicitně se otevírá zadokován do hlavního panelu spolu se všemi soubory s příponou *.PAM. Nástroj je v prostředí Mosaic z důvodu kompatibility se staršími systémy. Pro nové aplikace je k dispozici výše uvedený komfortnější nástroj WebMaker s grafikou.

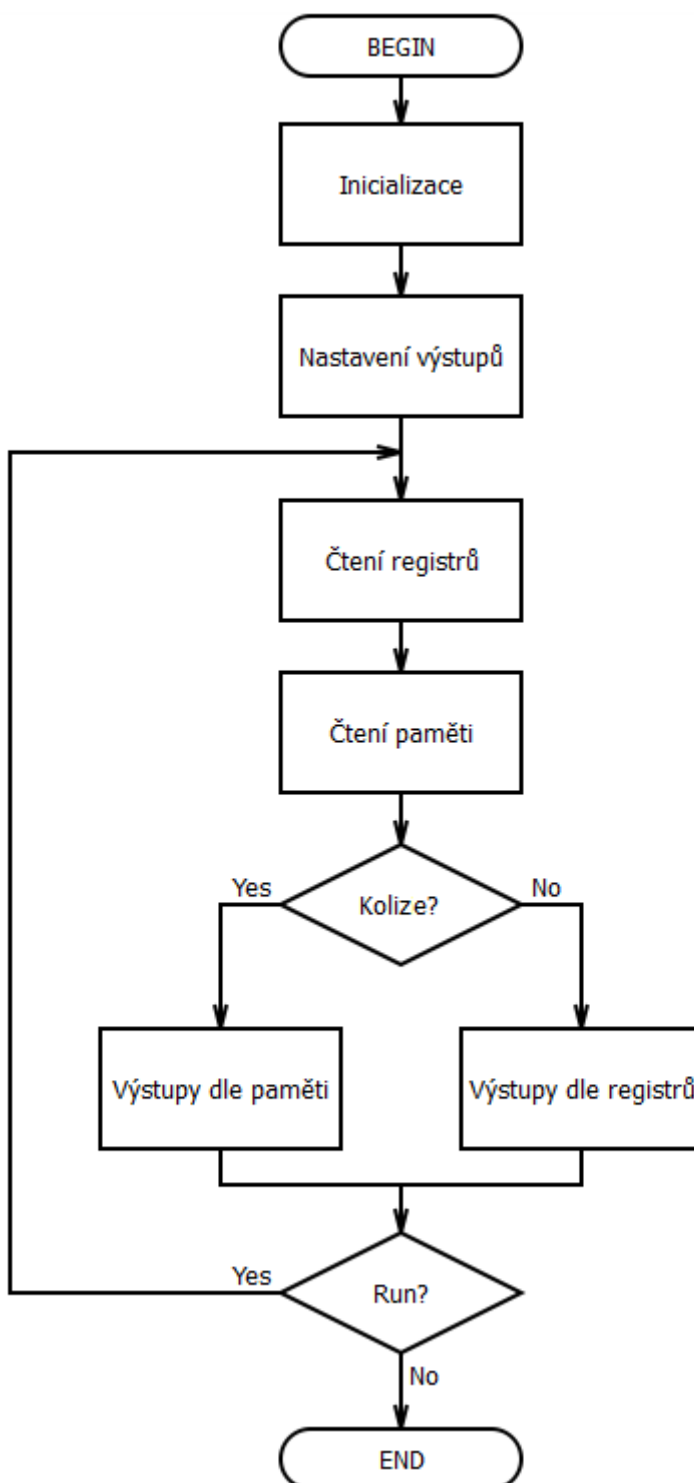
Mapa uživatelských registrů - zobrazuje obsazení paměti s uživatelskými registry %R v PLC a umožňuje kontrolovat možné překrytí definic přiřazených proměnným. Otevírá se klepnutím na ikonu a je implicitně otevřen do plovoucího okna. [4]



Obrázek 29. Vývojové prostředí Mosaic

5.2.Aplikace „IntelliHome_PLC“

Celý program pro řízení chytrého domu je napsán v jazyce „Strukturovaný text“. Program umožňuje při zadání hodnot registrů ovládat fyzickými elementy lampy, krokový motor a topení. Zjednodušený algoritmus programu je na následujícím diagramu (viz. obr. 30):



Obrázek 30. Algoritmus programu “IntelliHome_PLC”

5.2.1. Popis uživatelských registrů

Pro komunikaci pomocí protokolu Modbus Foxtrot PLC se používají tzv. „uživatelské registry“. Jsou to bloky paměti, u kterých můžeme dle tvaru psát nebo číst data. Popis registrů je uveden v následující tabulce č.12:

Číslo registru	Popis
%RW200	Registr pro lampy. 4 první bity jsou stavy lamp. Pomocí tabletu můžeme číst a psát tento registr.
%RW202	Registr pro roletu. Obsahuje hodnotu stavu rolety. Pomocí tabletu můžeme číst a psát tento registr.
%RW204	Registr pro topení. Obsahuje hodnotu nastavení teploty. Pomocí tabletu můžeme číst a psát tento registr.
%RW206	Registr pro topení. Obsahuje běžnou hodnotu teploty. Pomocí tabletu můžeme jenom číst tento registr.
%RW208 %RW210	Registry pro automatické řízení lamp. Pomocí tabletu můžeme číst a psát tyto registry.
%RW212 %RW214	Registry pro automatické řízení rolety. Pomocí tabletu můžeme číst a psát tyto registry.
%RW216 %RW218	Registry pro automatické řízení topení. Pomocí tabletu můžeme číst a psát tyto registry.

Tabulka 12. Popis uživatelských registrů

5.2.2. Popis programu

1. *Definice registrů:* Registry %RW* slouží k předávání dat mezi serverem a klientem (Holding registers). Registry %Y* se používají pro řízení výstupů PLC (Coils). Na obrázku č. 31 je popsána definice registrů, které se používají ke komunikaci. Dále jsou popsány výstupy z automatu pro řízení osvětlení (%Y6.5 - %Y7.1) a analogový vstup pro snímání hodnoty teploty (%XF48).

```
VAR_GLOBAL

//DEFINICE REGISTRU
SVT AT %RW200 : WORD;
ROLE AT %RW202 : WORD;
TOPN AT %RW204 : WORD;
TOPB AT %RW206 : WORD;
NASSZ AT %RW208 : WORD;
NASSV AT %RW210 : WORD;
NASRZ AT %RW212 : WORD;
NASRV AT %RW214 : WORD;
NASTZ AT %RW216 : WORD;
NASTV AT %RW218 : WORD;

//SVETLO VYSTUPY
S1 AT %Y6.5:BOOL;
S2 AT %Y6.6:BOOL;
S3 AT %Y6.7:BOOL;
S4 AT %Y7.0:BOOL;

//ROLETA VYSTUPY
R1 AT %Y6.0:BOOL;
R2 AT %Y6.2:BOOL;
R3 AT %Y6.3:BOOL;
R4 AT %Y6.4:BOOL;

//TOPENI VYSTUPY
T1 AT %Y7.1:BOOL;

//TOPENI VSTUPY
T0 AT %XF48:REAL;
```

Obrázek 31. Ukázka kódu – definice registrů

2. *Deklarace proměnných*: Na obrázku č. 32 je vyobrazena deklarace proměnných. Deklarace proměnných začíná klíčovým slovem VAR. Nejprve je vždy uveden název proměnné a za dvojtečkou její typ. Rovněž je možné přidat počáteční hodnotu proměnné.

```
VAR

SO:WORD;
TNOW:DINT;
ROLEB:INT;
ROZDIL:INT;
TimerRolety:TP;

start : BOOL;
start1 : BOOL;
output : BOOL;

TR:BOOL := TRUE;
FR:BOOL := FALSE;
inst_RS : RS;
TOP_LIM : WORD;

END_VAR
VAR_TEMP
END_VAR
```

Obrázek 32. Ukázka kódu – definice proměnných

3. *Funkce porovnání času:* Tato funkce se používá pro porovnání běžného času a nastaveného času. Vstupní hodnoty jsou TIMESET1, TIMESET2 a TIMENOW. Výsledkem je logická hodnota, tzn. TRUE nebo FALSE.

```

FUNCTION POROVN:BOOL

  VAR_INPUT
    TIMESET1, TIMESET2, TIMENOW:DINT;
  END_VAR

  IF TIMENOW > TIMESET1 AND TIMENOW < TIMESET2
  THEN POROVN := TRUE;
  ELSE POROVN := FALSE;
  END_IF;

END_FUNCTION

```

Obrázek 33. Ukázka kódu – porovnání času

4. *Řízení osvětlení:* Na obrázku č. 34 jsou uvedeny 2 způsoby řízení osvětlení: automatické řízení a ruční řízení. V případě automatického řízení se světlo zapíná a vypíná podle nastaveného času. V případě ručního režimu se řízení osvětlení realizuje pomocí tabletu, tj. při změně hodnoty registru SVT nastává změna stavu výstupů.

```

//-----Osvetleni
IF AUTO = 1
THEN IF POROVN(TIMESET1 := WORD_TO_DINT(NASSZ), TIMESET2 := WORD_TO_DINT(NASSV), TIMENOW := TNOW) = TRUE
  THEN S1 := TRUE;
  S2 := TRUE;
  S3 := TRUE;
  S4 := TRUE;
  ELSE S1 := FALSE;
  S2 := FALSE;
  S3 := FALSE;
  S4 := FALSE;
  END_IF;

ELSIF AUTO = 0
THEN
S1 := SVT.2;
S2 := SVT.1;
S3 := SVT.3;
S4 := SVT.0;
END_IF;

```

Obrázek 34. Ukázka kódu – řízení osvětlení

5. *Řízení topení*: Na obrázku č. 35 jsou uvedeny 2 způsoby řízení topení: automatické řízení a ruční řízení. Řízení topení probíhá podobně jako řízení osvětlení, ale v ručním režimu dochází k vypínání topení po dosažení hodnoty o stupeň větší než je nastavená hodnota, zapínání topení probíhá po dosažení hodnoty o stupeň menší než je hodnota nastavená.

```
//-----Topeni
IF AUTO = 1
THEN IF POROVN(TIMES1 := WORD_TO_DINT(NASSZ), TIMES2 := WORD_TO_DINT(NASSV), TIMENOW := TNOW) = TRUE
      THEN TOPN := NASTZ;
      ELSE TOPN := NASTV;
      END_IF;
END_IF;
TOPB := REAL_TO_WORD(TOP);
TOP_LIM := INT_TO_WORD(WORD_TO_INT(TOPB) + 1);
IF TOPN >= (TOP_LIM)
THEN T1 := TRUE;
ELSIF TOPN <= INT_TO_WORD(WORD_TO_INT(TOPB) - 1)
THEN T1 := FALSE;
END_IF;
```

Obrázek 35. Ukázka kódu – řízení topení

6. *Řízení rolety*: Stejně jako na obrázcích č. 34 a č. 35, na obrázku č. 36 jsou popsány 2 způsoby ovládání rolety. První automatický způsob řízení se realizuje stejným způsobem jako řízení topení a osvětlení, tj. podle nastaveného času. V ručním režimu se roleta pohybuje podle hodnoty registru ROLE, který je určován polohou RatingBaru.

```
//-----Roleta
IF AUTO = 1
THEN IF POROVN(TIMES1 := WORD_TO_DINT(NASRZ), TIMES2 := WORD_TO_DINT(NASRV), TIMENOW := TNOW) = TRUE
      THEN ROLE := 0;
      ELSE ROLE := 8;
      END_IF;
END_IF;

ROZDIL := WORD_TO_INT(ROLE) - ROLEB;
IF ROZDIL = 0
THEN start := FALSE;
ELSE start := TRUE;
END_IF;

IF ROZDIL > 0
THEN R2 := FALSE;
ELSIF ROZDIL < 0
THEN R2 := TRUE;
END_IF;

inst_RS(S := start, R1 := not start, Q1 => start1);
TimerRolety(IN := start1, PT := INT_TO_TIME(ABS(ROZDIL)*1000), Q => R3);
ROZDIL := 0;

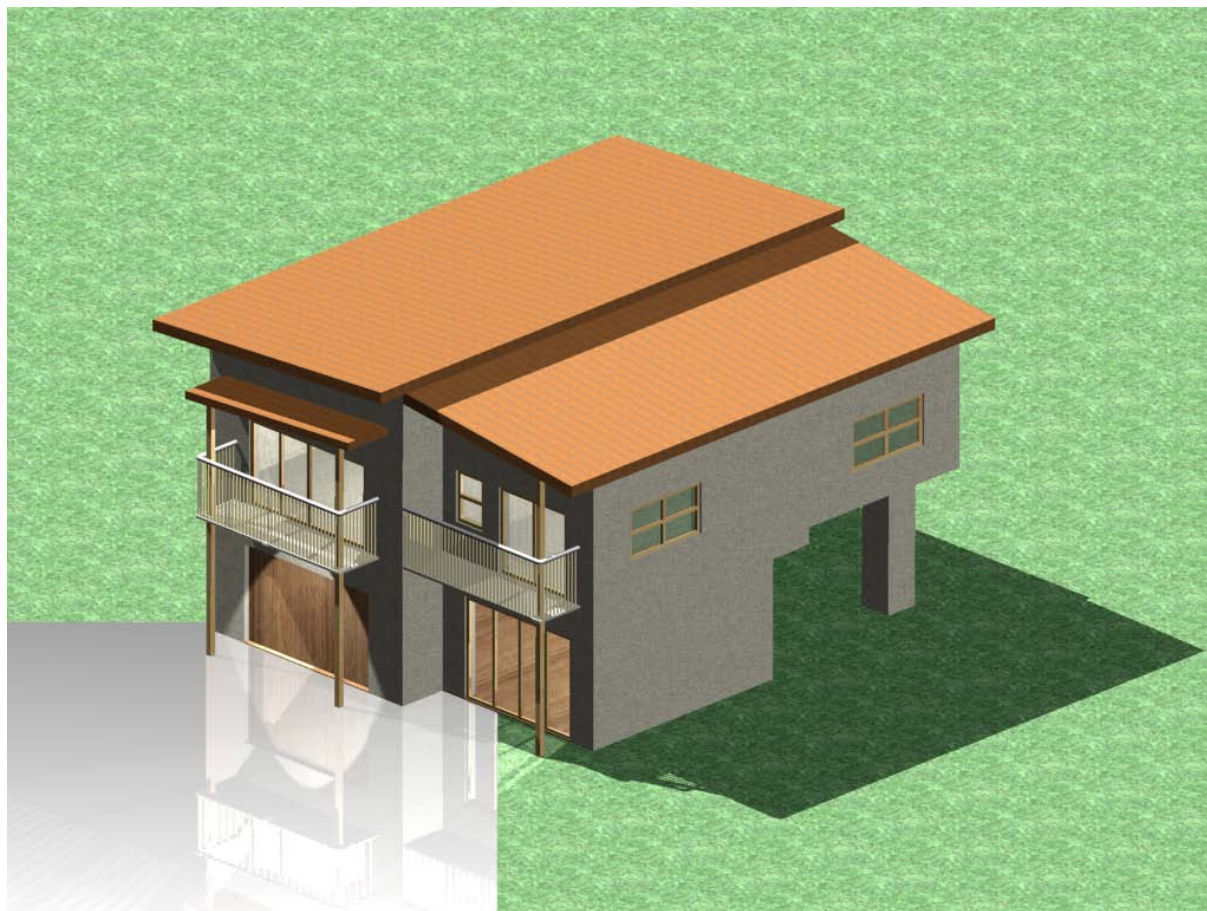
IF not R3
THEN ROLEB := WORD_TO_INT(ROLE);
END_IF;
```

Obrázek 36. Ukázka kódu – řízení rolety

6. Realizace makety

6.1. *Model domu*

Model byl vytvářen v prostředí AutoDesk Inventor 2012. Na následujícím obrázku č. 37 je znázorněn výsledek modelování:

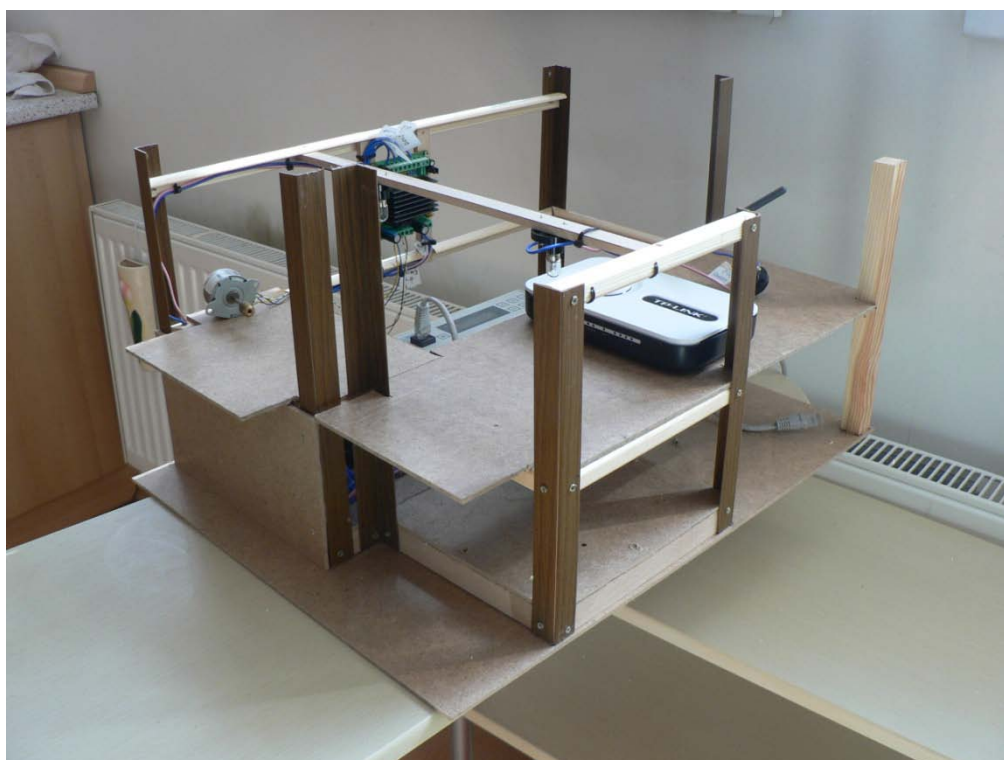


Obrázek 37. Model domu v prostředí AutoDesk Inventor 2012

Maketa byla vytvořena podle modelu (viz. obr. 38 a 39).



Obrázek 38. Maketa domu

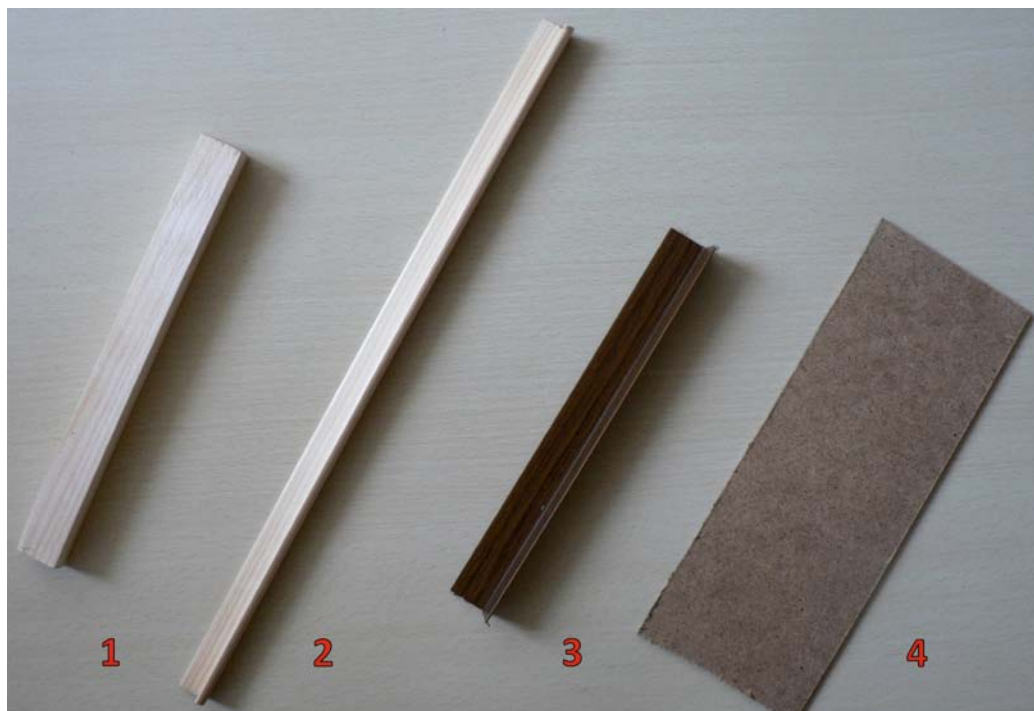


Obrázek 39. Maketa domu bez stěn

Byly použity následující materiály:

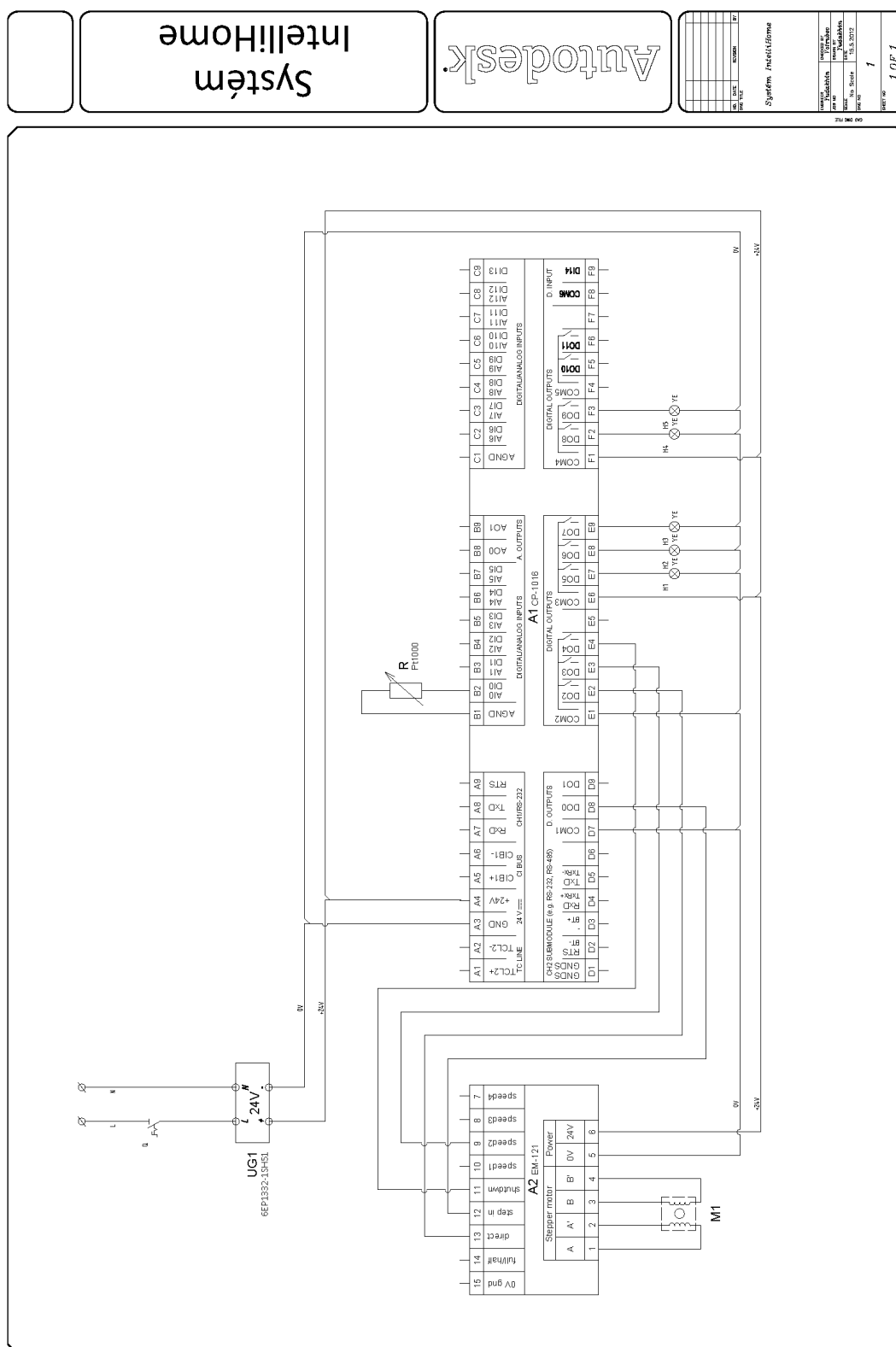
1. trám 20x20
2. dřevěný úhlový profil 15x15
3. plastový úhlový profil 20x20
4. sololit

Na následujícím obrázku č. 40 je příklad použitých materiálů:



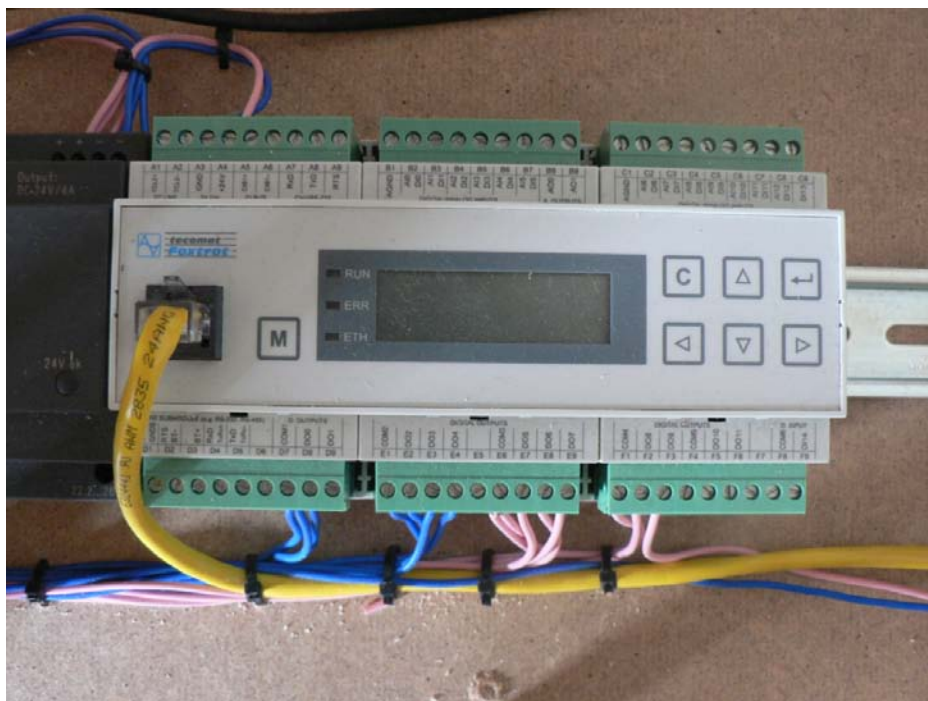
Obrázek 40. Použité materiály

Připojení zařízení bylo provedeno podle vlastního schématu (viz. obr. 41). V horní části schématu je připojení zdroje napájení na síť. Nalevo je modul A2 na ovládání krokového motoru EM-121. K němu je zespodu na svorky 1-4 připojen krokový motor. Řídící vstupy modulu A2 9,11,12 a 13 jsou propojeny s digitálními výstupy modulu A1 - PLC Foxtrot CP-1016, konkrétně na DO0, DO2, DO3 a DO4. Na digitální výstupy DO5, DO6, DO7 a DO8 jsou připojeny žárovky osvětlení jednotlivých místností. Topení je realizováno pomocí žárovky připojené na výstup DO9, která vytváří teplo. Poblíž je umístěno čidlo teploty Pt1000, které měří spojitý signál odpovídající teplotě. Toto čidlo je připojeno na analogový vstup AI0.



Obrázek 41. Připojení zařízení

Na následujícím obrázku č. 42 je uvedeno připojení automatu. Automat je používán pro řízení celého systému:



Obrázek 42. Připojení FOXTROT CP-1016

Na následujícím obrázku č. 43 je znázorněno připojení ovladače. Ovládač EM-121 se používá pro řízení krokového motoru:



Obrázek 43. Připojení ovládače EM-121

Na následujícím obrázku č. 44 je ukázáno připojení zdroje a svorek. Zdroj slouží pro napájení celého systému. První výstup je použit pro napájení PLC, druhý výstup slouží pro napájení zbytku systému. Svorky jsou určeny pro připojení přívodního vodiče:



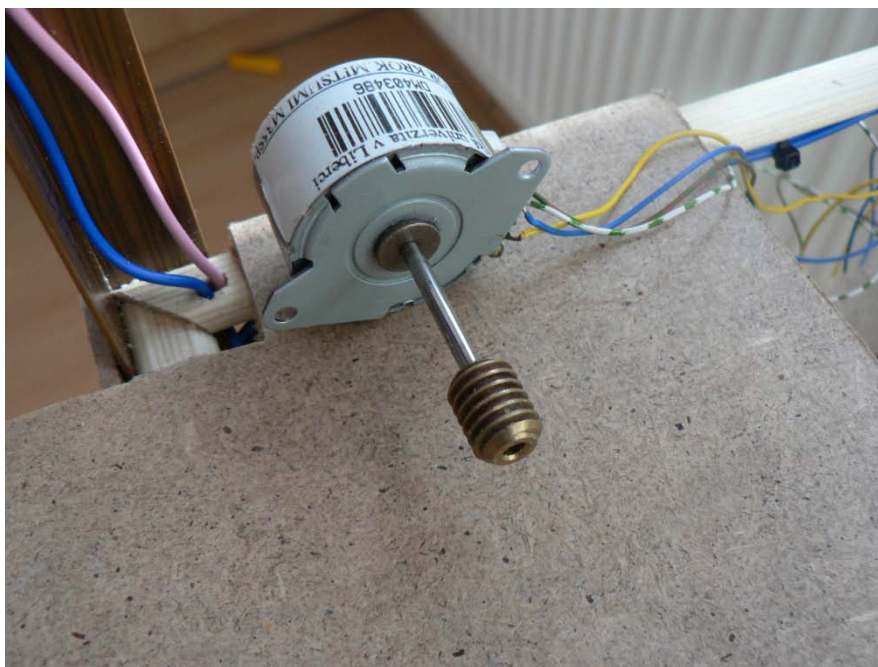
Obrázek 44. Připojení zdroje LOGO!Power

Na modelu inteligentního domu je osvětlení jednotlivých místností realizováno pomocí žárovek. Na následujícím obrázku č. 45 je vidět připojení žárovek:



Obrázek 45. Připojení žárovek

Na modelu inteligentního domu je pohyb rolety realizovan pomocí krokového motoru Mitsumi M35SP-8. Na následujícím obrázku č. 46 je znázorněno připojení motoru.



Obrázek 46. Připojení motoru

Na modelu inteligentního domu je topení realizováno pomocí žárovky, která vytváří teplo a čidla teploty Pt1000, které měří spojitý signál odpovídající teplotě. Na následujícím obrázku č. 47 je vidět připojení čidla.



Obrázek 47. Připojení čidla

7. Závěr

V rámci diplomové práce byl navržen a realizován řídicí systém pro chytrý dům. Byla vytvořena aplikace pro Android OS (viz. kapitola 4), byl naprogramován automat FOXTROT (viz. kapitola 5) a byl vytvořen model pro demonstraci systému (viz. kapitola 6).

Navržený systém umožňuje řízení chytrého domu jak přímo v rámci dosahu Wi-Fi sítě domu, tak i vzdáleně přes internetové připojení. Rovněž podporuje programování času pro automatické řízení. Dále existuje možnost jeho úpravy a doplnění pomocí rozšiřujících modulů pro PLC FOXTROT.

Nehledě na možnost řízení automatem FOXTROT a řízení přes web rozhraní jsem vytvořil samostatné aplikace, což bylo provedeno kvůli tomu, že jsem začal s tvorbou aplikací ještě před výběrem konkrétního PLC a ty byly tím pádem řešeny obecně. Díky tomu je jednoduše transformovat na jiný PLC, což neřeší řízení přes web rozhraní.

Proč ale tvořit vlastní aplikaci, když lze jednoduše používat www? Vlastní aplikace je samozřejmě daleko komfortnější než rozklikávání stránek ve webovém prohlížeči. Např. internetové aplikace bank, samoobsluhy mobilních operátorů, atd. jsou též řešeny i přes www (jednak v klasické verzi a jednak ve verzi pro mobilní zařízení upravené pro menší rozlišení, např. www.seznam.cz versus m.seznam.cz) i skrze samostatné aplikace pro windows mobile, android a iOS aply, které jsou uživatelsky mnohem pohodlnější a příjemnější.

Celý systém je možné použít i pro reálný model bez dalších nákladů, což je dáno tím, že ovladač krokového motoru je určen i pro silnější pohony. Rovněž existuje možnost upravit aplikace pro mobilní telefon (např. HTC One V), což lze provést také velmi jednoduše.

Vytvořený model bude umístěn v laboratoři katedry KKY, kde bude sloužit k výuce logického řízení.

Použitá literatura

Knihy:

- [1] VIRIUS, Miroslav. *Programování v Javě: průvodce programováním mobilních aplikací*. Vyd. 1. V Praze: České vysoké učení technické, 2010, 304 s. ISBN 978-80-01-04518-3.
- [2] MURPHY, Mark L. *Android 2: průvodce programováním mobilních aplikací*. Vyd. 1. Brno: Computer Press, 2011, 375 s. ISBN 978-80-251-3194-7.
- [3] PECINOVSKÝ, Rudolf. *Myslíme objektově v jazyku Java 5.0: průvodce programováním mobilních aplikací*. Vyd. 1. Praha: Grada, 2004, 601 s. ISBN 8024709414.

Návody publikované elektronicky:

- [4] *Programování PLC podle normy IEC 61 131-3 v prostředí Mosaic* [online]. 10. vydání. 2007 [cit. 2012-03-22]. Dostupné z: http://www.tecomat.com/wpimages/other/DOCS/cze/TXV00321_01_Mosaic_ProgIEC_cz.pdf
- [5] *PROGRAMOVATELNÉ AUTOMATY TECOMAT FOXTROT CP-1006, CP-1016, CP-1026, CP-1036* [online]. 3. vydání. 2012 [cit. 2012-03-22]. Dostupné z: http://www.tecomat.com/wpimages/other/DOCS/cze/TXV00436_01_Foxtrot_CP_1006.pdf

Internetové zdroje:

- [6] ESTELAR S.R.O. *Systém inteligentní elektroinstalace GILD* [online]. 2012 [cit. 2012-03-22]. Dostupné z: <http://www.gildsystem.cz/cz/>
- [7] TECO A.S. *Systém inteligentního ovládání domu FOXTROT* [online]. 2011 [cit. 2012-03-22]. Dostupné z: <http://www.ovladejsvujdum.cz/>
- [8] JABLOTRON ALARMS A.S. *Zabezpečení objektů - systém JABLOTRON 100* [online]. 2012 [cit. 2012-03-22]. Dostupné z: <http://www.jablotron.cz/cz/sekce/vyrobky/zabezpeceni+objektu+system+jablotron+100/>
- [9] SIEMENS AG. *Siemens Industry Online Support: 6EP1332-1SH51LOGO!POWER 24 V/4 A* [online]. 2011 [cit. 2012-03-22]. Dostupné z: <http://support.automation.siemens.com/WW/view/en/19060664>
- [10] ASUSTEK COMPUTER INC. *ASUS EEE PAD TRANSFORMER* [online]. 2011 [cit. 2012-03-22]. Dostupné z: <http://www.asuseepad.cz/transformer/>
- [11] I4WIFI A.S. *TP-LINK, TL-WR340GD AP/Router - 2,4 GHz, 4x LAN, 1x WAN* [online]. 2011 [cit. 2012-03-22]. Dostupné z: http://www.tp-link.cz/tl-wr340gd-ap-router-2-4-ghz-4x-lan-1x-wan_d1063.html
- [12] Krokový motor. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2012 [cit. 2012-03-22]. Dostupné z: http://cs.wikipedia.org/wiki/Krokový_motor

- [13] MITSUMI ELECTRIC CO., LTD. *Stepping Motors* [online]. 2012 [cit. 2012-03-22]. Dostupné z: http://www.mitsumi.co.jp/latest/Catalog/compo/motor/m35sp8_e.html
- [14] OEM AUTOMATIC LTD. *Electromen EM-121 Stepper motor control* [online]. 2011 [cit. 2012-03-22]. Dostupné z: http://www.oem.co.uk/products/Motors/Motor_drives/Stepper/Electromen_EM-121_Stepper_motor_control/621763-500661.html
- [15] HYASHI DENKO CO LTD. *PLATINUM RTD ELEMENTRATURE SENSOR* [online]. 2012 [cit. 2012-05-22]. Dostupné z: <http://www.hayashidenko.co.jp/en/rtdelement.html>
- [16] Eclipse (vývojové prostředí). In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2012 [cit. 2012-03-22]. Dostupné z: [http://cs.wikipedia.org/wiki/Eclipse_\(vývojové_prostředí\)](http://cs.wikipedia.org/wiki/Eclipse_(vývojové_prostředí))
- [17] Android (operační systém). In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2012 [cit. 2012-03-22]. Dostupné z: [http://cs.wikipedia.org/wiki/Android_\(operační_systém\)](http://cs.wikipedia.org/wiki/Android_(operační_systém))
- [18] GOOGLE. *Android developers* [online]. 2012 [cit. 2012-03-22]. Dostupné z: <http://developer.android.com/index.html>
- [19] Modbus. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2012 [cit. 2012-03-22]. Dostupné z: <http://cs.wikipedia.org/wiki/Modbus>
- [20] *Android Dev Guide CZ* [online]. 2011 [cit. 2012-03-22]. Dostupné z: <https://sites.google.com/site/androiddevguidecz/>

Příloha

Jako příloha je vloženo CD, které obsahuje:

- zdrojový kód aplikace „IntelliHome_PLC“
- zdrojový kód aplikace a zkompilovaný soubor IntelliHome.apk
- diplomovou práci v elektronické podobě
- schéma připojení ve formátu .dwg (Autodesk AutoCAD)
- model domu ve formátu .iam (Autodesk Inventor)