TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Inteligentní domovní systém

Diplomová práce

Studijní program: N2610 – Elektrotechnika a informatika
Studijní obor: 1802T007 – Informační technologie

Autor práce: **Bc. Tomáš Moravec**
Vedoucí práce: doc. Ing. Josef Chaloupka, Ph.D.

Liberec 2018

TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies

Smart home system

Master thesis

Study programme: N2610 – Electrical Engineering and Informatics
Study branch: 1802T007 – Information Technology

Author: **Bc. Tomáš Moravec**
Supervisor: doc. Ing. Josef Chaloupka, Ph.D.

Liberec 2018

Technická univerzita v Liberci
Fakulta mechatroniky, informatiky a mezioborových studií
Akademický rok: **2017/2018**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Tomáš Moravec**

Osobní číslo: **M16000179**

Studijní program: **N2612 Elektrotechnika a informatika**

Studijní obor: **Informační technologie**

Název tématu: **Inteligentní domovní systém**

Zadávající katedra: **Ústav informačních technologií a elektroniky**

Zásady pro vypracování:

1. Seznamte se s problematikou návrhu mobilních aplikací, programování mikropočítačů Arduino a vytváření složitějších digitálních systémů.
2. Vytvořte komplexní domovní zabezpečovací a meteorologický systém, který bude ovládaný pomocí mobilního telefonu, případně z webových stránek.
3. Jádrem systému bude modul s mikropočítačem řady Arduino, který bude sbírat a zpracovávat data z domovních/venkovních bezpečnostních (pohybové čidlo, čidla otevření dveří a oken?) a meteorologických senzorů (teplotní, srážková čidla?) a zároveň bude vhodný pro elektronické ovládání některá domovní zařízení (elektrické topení, klimatizace?). K mikropočítači bude vhodně připojen GSM/GPRS modul pro příjem a odesílání dat.
4. Vytvořené zařízení (inteligentní domovní systém) bude konfigurovatelné a ovládané programem vytvořeným v C# WPF, případně ovládacím programem z mobilního telefonu nebo z vlastních responzivně navržených webových stránek.

Rozsah grafických prací: **Dle potřeby dokumentace**

Rozsah pracovní zprávy: **cca 40-50 stran**

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

[1] **Nussey, J.: Arduino For Dummies. In Wiley - Wiley - the trusted publisher of academic, scientific, and professional books since 1807, ISBN**

[2] **9781118446379, 2013 Voda, Z.: Průvodce světem Arduina, e-Book, 2014 Burd, A., B.: Android Application Development All-in-One For Dummies, ISBN 978-1118973806, 2015**

Vedoucí diplomové práce: **doc. Ing. Josef Chaloupka, Ph.D.**
Ústav informačních technologií a elektroniky

Konzultant diplomové práce: **Ing. Karel Paleček, Ph.D.**
Ústav informačních technologií a elektroniky

Datum zadání diplomové práce: **19. října 2017**

Termín odevzdání diplomové práce: **14. května 2018**

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

L.S.

prof. Ing. Ondřej Novák, CSc.
vedoucí ústavu

V Liberci dne 19. října 2017

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 mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 14. 5. 2018

Podpis:

Poděkování

Děkuji vedoucímu práce panu doc. Ing. Josefu Chaloupkovy, Ph.D. za odborné vedení a poskytnuté informace při zpracování závěrečné diplomové práce.

Abstrakt

Diplomová práce se zabývá vytvořením inteligentního domovního systému, určenému ke zpracování dat z bezpečnostních a meteorologických čidel, a řízení vybraných domovních zařízení. Komplexní řešení zahrnuje jak samotný domovní systém, tak mobilní, webovou a desktopovou aplikaci. V úvodu jsou definovány základní pojmy a požadované vlastnosti. Následuje rešerše a výběr vhodných hardwarových a softwarových řešení. Na základě rešerše je navržen koncept komplexního domovního systému, který je následně realizován v jednotlivých etapách. Výstupem práce je funkční cílové zařízení, které aplikuje zvolené hardwarové i softwarové řešení.

Abstract

The master thesis deals with development of an intelligent home system, designed to process data from security and meteorological sensors, and to control of selected home devices. The solution includes not only home system, but also mobile, internet and desktop applications. The introduction defines basic concepts and required functions, followed by a research and selection of the best hardware and software solutions. Based on the research, a concept of the complex system is designed and the solution is realized in separate stages. The result of this work is target device, which follows the chosen hardware and software solution.

Obsah

Seznam zkratek 10

1 Úvod **11**

2 Inteligentní domovní systém **12**
2.1 Zabezpečovací systém 12
2.1.1 Ústředna 12
2.1.2 Ovladač 12
2.1.3 Detektor 13
2.1.4 Komunikátor 13
2.2 Inteligentní dům 13
2.2.1 Centrální jednotka 14
2.3 Požadované vlastnosti 14
2.3.1 Domovní systém 14
2.3.2 Desktopová aplikace 15
2.3.3 Mobilní aplikace 15
2.3.4 Webová aplikace 16
2.3.5 Komunikační server 16

3 Koncept **17**
3.1 Domovní systém 17
3.1.1 Řídící jednotka 17
3.1.2 Programovací jazyk C++ 18
3.1.3 Vývojové prostředí Arduino 18
3.1.4 Detektory 19
3.1.5 Spínací prkvy 19
3.1.6 Komunikátor 19
3.2 Desktopová aplikace 20
3.2.1 Programovací jazyk C# 20
3.2.2 Vývojové prostředí Visual Studio 20
3.3 Mobilní aplikace 20
3.3.1 Programovací jazyk Java 21
3.3.2 Vývojové prostředí Android Studio 21
3.4 Webová aplikace 21
3.5 Komunikační server 21

4 Prototyp domovního systému **22**
4.1 Využití mobilního telefonu . . . 22
4.1.1 Využití mobilního telefonu pro datovou komunikaci . . . 22
4.1.2 Výběr vhodného telefonu . . . 23
4.1.3 Zakoupený tlačítkový mobilní telefon . . . 23
4.1.4 Informativní kontaktování výrobce . . . 24
4.1.5 Rozebrání mobilního telefonu . . . 24
4.1.6 Závěr rozboru . . . 25
4.2 Prototyp domovního systému . . . 25
4.2.1 Blokové schéma . . . 25
4.2.2 Výsledný prototyp . . . 26
4.3 Firmware . . . 27
4.3.1 Nastavení . . . 28
4.3.2 API . . . 28
4.3.3 Notifikace . . . 29
4.3.4 Chybová hlášení . . . 30

5 Desktopová aplikace **31**
5.1 Blokové schéma . . . 31
5.1.1 Connection . . . 32
5.1.2 Logic . . . 32
5.1.3 GUI . . . 32
5.2 Grafické uživatelské rozhraní . . . 32
5.2.1 Připojení/odpojení řídící jednotky (Connection) . . . 33
5.2.2 Přehled připojených komponent (Overview) . . . 34
5.2.3 Nastavení senzoru (Sensor Edit) . . . 35
5.2.4 Nastavení spínače (Switch Edit) . . . 35
5.2.5 Možnosti programu (Features) . . . 36
5.2.6 Notifikace . . . 36
5.3 Kompatibilita . . . 37
5.3.1 Testované verze Windows 10 . . . 37
5.3.2 Podporované řídící jednotky . . . 37

6 Mobilní aplikace **39**
6.1 Blokové schéma . . . 39
6.1.1 Communication . . . 39
6.1.2 Logic . . . 40
6.1.3 Activities . . . 40
6.2 Grafické uživatelské rozhraní . . . 40
6.2.1 Přehled připojených komponent . . . 41
6.2.2 Podrobnosti vybrané komponenty . . . 42
6.3 Kompatibilita . . . 43

7 Webová aplikace **44**

8 Komunikační server **45**
8.1 Slepé cesty vývoje . . . 45
8.1.1 Přímá komunikace mezi zařízeními . . . 45
8.1.2 Implementace serveru v řídící jednotce . . . 45
8.2 Blokové schéma . . . 45
8.2.1 Communication . . . 46
8.2.2 Logic . . . 46
8.2.3 Logger . . . 46
8.3 Zprovoznění . . . 47
8.4 Konzolový výstup . . . 47
8.5 Testovací klient . . . 47

9 Závěr **49**

Literatura **50**

A Obsah přiloženého CD **53**

Seznam zkratek

API	Application Programming Interface, rozhraní pro přístup k aplikaci
COM	Communication port, komunikační rozhraní
GBS	Glass break detector, detektor rozbití skla
GPRS	General Packet Radio Service, služba pro přenos dat v mobilní síti
GSM	Groupe Spécial Mobile, globální systém pro mobilní komunikaci
GUI	Graphical User Interface, grafické uživatelské rozhraní
IDE	Integrated development environment, integrované vývojové prostředí
IP	Internet Protocol, internetový protokol
LAN	Local Area Network, místní počítačová síť
LED	Light-Emitting Diode, dioda emitující světlo
PIR	Passive infrared sensor, pasivní infračervené čidlo
SMS	Short message service, služba krátkých textových zpráv
TUL	Technical University of Liberec, Technická univerzita v Liberci
USB	Universal Serial Bus, univerzální sériová sběrnice

1 Úvod

Podnětem pro vytvoření této práce mi byl můj vlastní zájem o zabezpečovací systémy, inteligentní domovní zařízení a propojování hardware a software.

V dnešní době je běžné, že je rodinný dům vybaven zabezpečovacím zařízením, typicky proti zlodějům. V těchto případech se běžně používá tlačítková klávesnice jako vstupní prvek do zabezpečovacího systému. V posledních letech se také oblibě začínají těšit domácí asistenti, chytré domy a všeobecně přidávání chytrých prvků do našich domovů, řízených většino pomocí mobilního telefonu. Navazujíc na svou bakalářskou práci, zabezpečovací zařízení pro osobní automobily, jsem měl v plánu tematicky pokračovat touto cestou. Proto jsem se rozhodl pro spojení klasického zabezpečovacího systému s chytrými prvky, jako je například ovládání pomocí mobilní aplikace, nebo skrze webové rozhraní. Vlastním řešení jsem se rozhodl dokázat, že tvorba takovéhoto systému je možná, a také, že výsledný produkt je praktický zároveň. Při této příležitosti jsem se rozhodl vytvořit kompletní paletu produktů, které reprezentují všechny velké softwarové platformy, tedy desktopovou, mobilní a webovou.

Svým výzkumem jsem zjistil, že vše lze realizovat. Proto jsem se rozhodl, že v rámci své diplomové práci provedu první etapu, a to vytvoření funkčního prototypu a implementace kompletního softwarového řešení, které již bude možné využívat s případným reálným produktem.

2 Inteligentní domovní systém

Inteligentní domovní systém je kombinací zabezpečovacího systému a prvků chytré domácnosti. Tímto spojením se zvyšuje komfort obyvatel domu, s cílem zvýšit jejich pohodlí a zaručit nejvyšší možnou bezpečnost. Následující část se věnuje vysvětlení elementárních pojmů této problematiky, které se v dané tématice běžně vyskytují. Tato část rozdělena na zabezpečovací systém a inteligentní dům, z kterýchž se inteligentní domovní systém sestává.

2.1 Zabezpečovací systém

Elektronická zabezpečovací signalizace, neboli zabezpečovací systém, je zařízení, které vizuálně nebo akusticky vyhlašuje poplach a dává na vědomí, že nastaly nějaké potíže nebo došlo ke splnění sledované podmínky [21]. Jde tedy o zařízení, které slouží k ochraně osob a majetku. Systém je řízen ústřednou a může se spustit analogovou (např. dveřní, či okenní čidlo) i digitální (detektor pohybu) detekcí. Komunikace mezi detektory a ústřednou může být vedena kabelem, bezdrátově anebo kombinací předešlých způsobů, tj. jeden detektor může být připojen kabelem a druhý bezdrátově [13].

2.1.1 Ústředna

Mozkem celého systému je ústředna. Propojena je s ostatními prvky systému kabely nebo bezdrátově a obstarává komunikaci mezi jednotlivými komponenty systému. V integrované paměti má uložené nejdůležitější informace a nastavení [13]. V závislosti na připojených komponentech pak může různě reagovat na splnění sledovaných podmínek. Často bývá vybavena jenom tím nejnutnějším pro vyvolání poplachu, to ať už akustického (siréna), nebo tichého (informování majitele) [21].

2.1.2 Ovladač

K ovládání, případně k programování ústředny slouží ovladač. Dnešní alarmy je možné ovládat několika způsoby. Jako ovladač se nejčastěji používá klávesnice vybavená tlačítky, případně čtečkou (čipovými kartami a přívěšky) nebo též dálkové ovládání. U některých systémů se dá přes klávesnici provést nastavení celého systému. Klávesnice slouží k zastřežení i odstřežení systému [12]. Dalšími způsoby je ovládání

přes internet, kdy se většinou používá integrované webové rozhraní, ke kterému se uživatel může připojit po zadání hesla, nebo ovládání přes mobil (SMS příkazy) [18].

2.1.3 Detektor

Detektor je prvek systému, který je rozmístěn v hlídaném objektu a má za úkol reagovat aktivací při narušení (otevření, pohyb, rozbití atd.) a to tak, že tuto informaci předá ústředně, která ji následně zpracuje [12]. Nejčastěji používané detektorové prvky jsou:

- Magnetický kontakt (dveřní čidlo)
- Detektor pohybu (PIR detektor)
- Detektor tříštění skla (GBS detektor)
- Detektor plynu
- Infra závora

2.1.4 Komunikátor

V případě nutnosti odeslání informací o narušení objektu, případně o odchylce od normálního provozního stavu zabezpečovacího systému, je možné využít komunikátor. Ten může být řešen více způsoby, ale běžně se využívá radiový vysílač. Přítomnost komunikátoru v systému, na rozdíl od ostatních částí, není podmínkou. [12]. Nejčastější typy komunikátorů jsou:

- GSM komunikátor
- LAN komunikátor
- Telefonní komunikátor
- Komunikátor využívající radiové sítě s vyhrazenou frekvencí

2.2 Inteligentní dům

Inteligentní dům, neboli chytrá domácnost, znamená možnost ovládání osvětlení, elektroinstalace, termostatu, domácího alarmu, kamer nebo zámku dveří na dálku skrze chytrý telefon, tablet, ale i počítač nebo televizi. Běžně je vše propojeno centrální jednotkou, která většinou poskytuje dálkové ovládání skrze internet.

2.2.1 Centrální jednotka

Srdcem chytré domácnosti je centrální jednotka (neboli řídící jednotka), která komunikuje prostřednictvím konkrétního protokolu. K takové jednotce je možné kdykoliv dokupovat další zařízení, a tvořit tak komplexní chytrou domácnost. Ovládání a řízení celého systému je velmi jednoduché a intuitivní. Inteligentní dům se také stará o zabezpečení domácnosti a poskytuje aktuální přehled informací o svém stavu.

Nejčastější typy připojovaných řízení:

- Osvětlení
- Elektroinstalace
- Termostaty
- Zabezpečení
- Kamery
- Meteostanice

2.3 Požadované vlastnosti

Na základě požadovaných vlastností, byl sestaven seznam požadavků všech částí inteligentního domovního systému. Každá z pěti částí obsahuje popis požadovaných vlastností a výsledný seznam požadavků.

2.3.1 Domovní systém

Z úvodní části je zřejmé, že domovní systém obsahuje jak prvky zabezpečovacího systému, tedy musí obsahovat detektory, komunikátor a ústřednu. Zároveň však obsahuje prvky inteligentního domu, tedy musí být schopen monitorovat a řídit různá zařízení v domácnosti. V rámci domovního systému jsou výše zmíněné prvky řízeny ovladačem. V této práci se jedná o desktopovou, mobilní aplikaci a webovou aplikaci. Komunikace bude probíhat přes internet.

Požadavky na domovní systém:

- Poskytnutí API
- Komunikace přes internet
- Vzdálené ovládání všech připojených zařízení
- Vzdálená správa a monitoring všech připojených zařízení
- Spuštění poplachu při splnění podmínek (kontaktování uživatele)
- Komunikaci s desktopovou, mobilní a webovou aplikací

2.3.2 Desktopová aplikace

Desktopová aplikace musí být schopna využít plného potenciálu ústředny, tedy umožnit kompletní správu a nastavování jednotlivých komponent (spínačů a senzorů). Komunikace s ústřednou bude umožněna pomocí sériové linky (COM port), v případě prototypu bude použito připojení skrze USB.

Požadavky na desktopovou aplikaci:

- Komunikace po seriové lince
- Zobrazení seznamu všech připojench komponent
- Zobrazení všech informací o zvolené komponentě
- Změna libovolného nastavení všech komponent
- Sledování aktuálního stavu všech komponent
- Přidávání nových komponent
- Odstraňování stávajících komponent
- Notifikace v případě změny stavů senzorů
- Změna stavu spínačů
- Připojování a odpojování od zvolené ústředny

2.3.3 Mobilní aplikace

Mobilní aplikace je učena pouze jako monitorovací zařízení, ze kterého bude možné sledovat stavy jednotlivých senzorů, případně spínat veškeré spínací prvky. Její návrh je tedy značně jednodušší oproti komplexnější desktopové aplikaci. Komunikace bude probíhat přes internet.

Požadavky na mobilní aplikaci:

- Komunikace přes internet
- Zobrazení seznamu všech připojench komponent
- Zobrazení všech informací o zvolené komponentě
- Sledování aktuálního stavu všech komponent
- Notifikace v případě změny stavů senzorů
- Změna stavu spínačů
- Připojování a odpojování od zvolené ústředny

2.3.4 Webová aplikace

Od webové aplikace se očekává možnost sledování aktuálního denní online. Od aplikace se neočekává možnost editace, ani a jakékoliv řízení. Komunikace bude probíhat přes internet.

Požadavky na webovou aplikaci:

- Komunikace přes internet
- Sledování aktuálního stavu všech komponent

2.3.5 Komunikační server

Komunikační server je nutný pro navázání spojení mezi řídící jednotkou a mobilní a webovou aplikací. Od serveru se očekává udržování a správa všech spojení, stejně tak jako propagace zpráv do jejich cílových destinací. Komunikaci je nutné logovat a bude probíhat přes internet.

Požadavky na komunikační server:

- Komunikace přes internet
- Udržování stálého spojení s klienty
- Zajištění komunikace mezi klienty
- Logování probíhající komunikace

3 Koncept

Na základě požadovaných vlastností a přechozích zkušeností autora, byl vybrán

3.1 Domovní systém

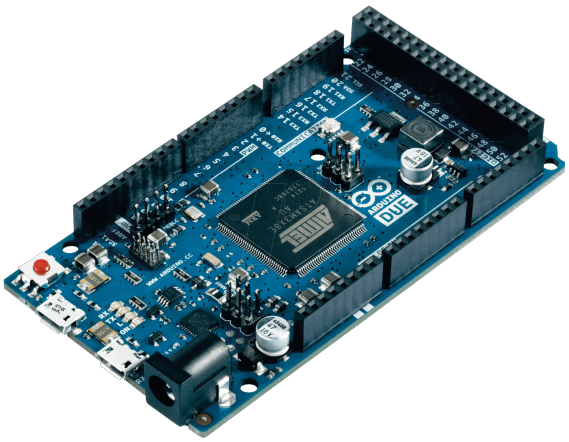
Domovní systém je kombinací zabezpečovacího zařízení a spínacích prvků. Těmto částem se budu věnovat níže, kde vyberu nejvhodnější varianty.

3.1.1 Řídící jednotka

Řídící jednotka musí být schopna spravovat všechny připojené detektory, senzory a spínací prvky, a komunikovat s desktopovou a mobilní aplikací, a zároveň musí řídit a zpracovávat jednotlivé komponenty. Již ze zadání diplomové je zřejmé, že řídící jednotkou bude vývojová platforma Arduino [25], která je se svými periferiemi [5] a čipem Atmega [8], případně ARM Cortex-M3, více než vhodnou pro tyto účely. Firmware řídící jednotky bude psán v jazyku C++ s nadstavbou Wiring (knihovna pro řízení hardwaru) [27] a vývoj bude probíhat jak v oficiálním vývojovém prostředí Arduino[6], tak v prostředí Visual Studio 2017 s rozšířením Visual Micro [24], které poskytuje zvýrazňování syntaxe knihovny Wiring a umožňuje komplikaci a nahrávání kódu přímo na desku Arduino. Komunikace bude probíhat skrze mobilní datovou síť.

Zvolená řídící jednotka:

- Arduino DUE (914 Kč s DPH), (`arduino-shop.cz`)



Obrázek 3.1: Vývojová platforma Arduino DUE

3.1.2 Programovací jazyk C++

Vývoj softwaru bude probíhat v programovacím jazyce C++ s nadstavbou vývojové platformy Wiring (knihovna Wire)[6], která jazyk rozšiřuje o nové příkazy, pro přímé řízení hardwarových součástek, vše zastřešeno sadou knihoven [2] (od tvůrců desky Arduino), které přidávají nové funkce, aby potencionální vývojář nepotřeboval hlubší znalosti programování a hardwaru. Tento kompletní balík příkazů [4] je někdy také nazýván programovacím jazykem Arduino [1].

3.1.3 Vývojové prostředí Arduino

Pro část vývoje bude použito oficiální vývojové prostředí, od tvůrců desky Arduino s identickým názvem Arduino [6]. Jedná se o počítačový software s otevřeným zdrojovým kódem (open-source) [7], určený k jednoduchému psaní a nahrávání zdrojových kódů na desku. Prostředí lze nainstalovat na operační systém Windows, MAC a Linux. Z vlastní zkušenosti vyjmenuji výhody, mezi které patří zvýraznění a barevné rozlišení jednotlivých příkazů, plná podpora vývojové platformy Wiring, podpora všech oficiálních i neoficiálních desek Arduino, zabudovaný klient pro komunikaci na sériové lince a dalších funkce. Nevýhodou je absence předvídání a dokončování kódu (predikce), nápověda při volání částí programu (funkcí, knihoven atd.), nemožnost krokování programu a velice obecné chybové hlášky, kvůli kterým je náročné odhalit případné chyby.

Jako alternativní vývojové prostředí k IDE Arduino bylo zvoleno Visual Studio 2017, díky dobrému zvýraznění syntaxe, možnost skrytí obsahů funkcí a krokování kódu, díky rozšíření Visual Micro [24], které navíc umožňuje komplikaci a nahrávání kódu přímo na desku Arduino. Více o aplikaci Visual Studio níže.

3.1.4 Detektory

Detektory jsou všechny velmi podobné a proto stačí připravit implementaci jejich snímání. V případě detektorů musí být možné připojit libovolný detektor, který lze nastavit jako spínací (normálně rozepnutý), nebo rozpínací (normálně sepnutý). Pro testovací účely byly zvoleny dva detektory, každý jednoho typu.

Zvolené detektory:

- Dveřní čidlo (rozpínací), (poskytl vedoucí)
- PIR detektor (spínací), (25 Kč s DPH), (`aliexpress.com`)

Obrázek 3.2: Detektor pohybu (PIR detektor)

3.1.5 Spínací prkvy

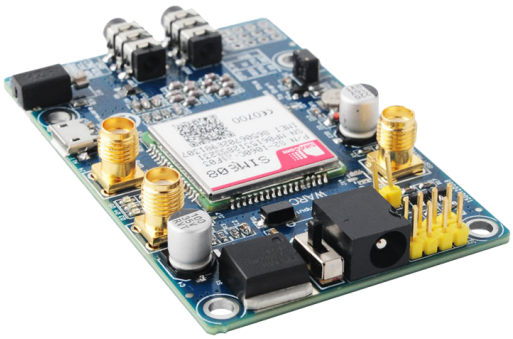
U spínacích prvků není nutné volit jednotlivé komponenty, ale stačí připravit implementaci jejich spínání. Poté je možné připojit libovolnou spínatelnou součástku.

3.1.6 Komunikátor

Požadavek na komunikátor je přenos dat na server přes mobilní data. Původní zadání jako komunikátor určuje mobilní telefon, pomocí kterého máme umožnit řídící jednotce datové přenosy. Zvolen byl nový a zároveň nejlevnější mobilní telefon na trhu. Tento způsob přístupu do mobilní datové sítě se nezdařil a zadání bylo upraveno. Byla zvolena alternativa v podobě GSM/GPRS modulu, který byl zakoupen z Číny, dodací doba této součástky, stejně jako na všechn ostatních, byla přes jeden kalendářní měsíc.

Zvolené komunikátory:

- Mobilní telefon STK R45i Black (449 Kč s DPH), (`alza.cz`)
- GSM/GPRS modul (290 Kč s DPH), (`aliexpress.com`)



Obrázek 3.3: Komunikační modul GSM/GPRS - SIM808

3.2 Desktopová aplikace

Na základě požadovaných vlastností byl zvolen cílový operační systém Windows 10, jakožto v Evropě nejrozšířenější [11]. Programovací jazyk C# a vývojové prostředí Visual Studio 2017. Jednotlivé požadavky kladené na desktopovou aplikaci lze nalézt níže. Komunikace s aplikací bude probíhat skrze připojení USB (COM).

3.2.1 Programovací jazyk C#

Vývoj desktopové aplikace bude probíhat v programovacím jazyce C#. Jedná se o vysokoúrovňový objektově orientovaný programovací jazyk vyvinutý firmou Microsoft zároveň s platformou .NET Framework. C# lze využít k tvorbě databázových programů, webových aplikací a stránek, webových služeb, formulářových aplikací ve Windows, softwaru pro mobilní zařízení (PDA a mobilní telefony) a dalších.

3.2.2 Vývojové prostředí Visual Studio

Pro vývoj softwaru bude použito vývojové prostředí (IDE) od Microsoftu, Visual Studio 2017. Vývojové prostředí může být použito pro vývoj konzolových aplikací a aplikací s grafickým rozhraním. Visual Studio obsahuje editor kódu podporující IntelliSense a refaktoring. IDE má integrovaný debugger. Vestavěné nástroje zahrnují designer formulářů pro tvorbu aplikací s GUI, designer webu, tříd a databázových schémat. Je možné přidávat dodatečná rozšíření.

3.3 Mobilní aplikace

Cílový operační systém byl zvolen Android, jakožto v Evropě nerozšířenější [17]. Programovací jazyk Java a vývojové prostředí a Android Studio. Jednotlivé požadavky kladené na mobilní aplikaci lze nalézt níže. Komunikace bude probíhat skrze mobilní datovou síť.

3.3.1 Programovací jazyk Java

Java je objektově orientovaný programovací jazyk, který se používá pro vývoj na platformě Android. Jedná se o jeden z nejpoužívanějších programovacích jazyků na světě. Díky své přenositelnosti je používán pro programy, které mají pracovat na různých systémech, jako například mobilní telefony, různá zabudovaná zařízení a aplikace pro desktopové počítače.

3.3.2 Vývojové prostředí Android Studio

Android Studio je nové vývojové prostředí založené na IntelliJ IDEA, vytvořené firmou Google (dnes již Alphabet Inc.). Je dostupné pro systémy Windows, Mac OS X a Linux. Součástí instalace je samotné IDE, Android SDK Tools, kompilátor Android a základní emulátory s plnohodnotným systémem Android. Díky emulátoru lze testovat aplikace, pro libovolnou verzi systému Android, nebo pro libovolné rozlišení, bez fyzického zařízení.

3.4 Webová aplikace

Webová aplikace bude pro jednoduchost vývoje navázána na komunikační server, jako webový náhled. Shodovat se tedy bude použitý programovací jazyk (C#) i vývojové prostředí (Visual Studio 2017).

3.5 Komunikační server

Operační systém byl zvolen Windows 10, kvůli rychlosti a jednoduchosti vývoje. Programovací jazyk C# a vývojové prostředí Visual Studio 2017. Komunikace s aplikací bude probíhat skrze pevné připojení k internetu.

4 Prototyp domovního systému

4.1 Využití mobilního telefonu

Následující část pojednává o snaze využít levný tlačítkový mobilní telefon pro přístup do datové sítě, přesněji využít jeho GSM/GPRS modulu, integrovaném na desce telefonu. Práce na této části zabraly přibližně dva měsíce, po kterých se tato cesta ukázala jak slepá, proto byla zavrhnuta.

Za vším stála myšlenka, že existuje velké množství poškozených a nepoužívaných tlačítkových mobilních telefonů, které jsou dnes již jednoduše nepoužitelné. Takový nepoužitelný telefon je prakticky bezcenný, ale většinou stále obsahuje funkční součástky, které je možné využít.

Jedním z cílů práce bylo zjistit, zda již existují jiná řešení daného problému, případně pokrýt chybějící část těchto řešení, například pro nezdokumentované modely telefonů.

4.1.1 Využití mobilního telefonu pro datovou komunikaci

Aktuálně se mobilní telefony rozdělují do dvou hlavních kategorií. Telefony chytré (smartphone), a telefony klasické, tlačítkové.

U chytrých telefonů je získání přístupu do sítě GPRS triviální. Stačí napsat software, který tyto zdroje poskytne přes port USB a tím je problém vyřešen. Klasické telefony se dělí do dvou podkategorií.

První podkategorie jsou telefony s dokumentací a s API. Některé starší telefony měli otevřené dokumentace a některá dokonce i API přímo pro tyto účely (Motorola, Nokia). Pro tyto mobilní telefony existuje velké množství návodů a postupů, jak připojení do sítě GPRS docílit.

Druhou podkategorií jsou mobilní telefony bez otevřených dokumentací a bez API, která by zpřístupňovala zdroje mobilního telefonu. Pro tyto mobilní telefony nebylo nalezeno žádné dostupné řešení, přitom jsou to telefony nejrozšířenější, tím se staly pro tento projekt zajímavými.

Získání přístupu do sítě GPRS, skrze:

- Chytrý telefon (řešení existují)
- Klasický telefon s API (řešení existují)
- Klasický telefon bez API a dokumentace (řešení nenalazeno)

4.1.2 Výběr vhodného telefonu

Vzhledem k výsledkům průzkumu bylo rozhodnuto, pokusit se o vyřešení využití klasického telefonu bez API a dokumentace. Prvním úkolem bylo vybrání vhodného tlačítkového mobilního telefonu pro tyto účely. Po provedení průzkumu trhu bylo zjištěno, že je možné vybírat jak z bazarových kusů, tak z nových. Zajímavým zjištěním bylo, že ceny nových telefonů podporujících funkci GPRS se pohybují kolem 450 Kč s DPH za kus, zatím co bazarové kousky začínají na 600 Kč s DPH za kus. Vzhledem k nižší ceně, záruce a garanci funkčnosti bylo rozhodnuto zakoupit nový model.

4.1.3 Zakoupený tlačítkový mobilní telefon

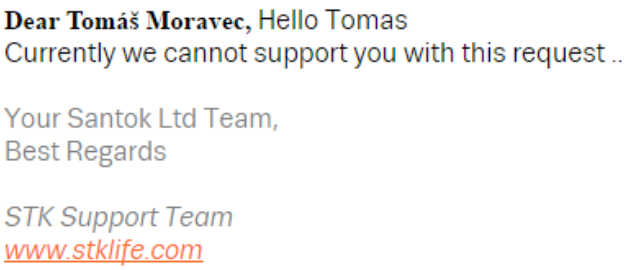
Při výběru nového tlačítkového telefonu byla hlavním sledovaným prvkem cena a přítomnost GPRS. Bezkonkurenčně nejlevnějším telefonem byl tlačítkový mobil od britského výrobce STK R45i. Zakoupen byl prostřednictvím internetového obchodu Alza.cz za cenu 419 Kč s DPH bez dopravy. Cena s dopravou činila celkových 449 Kč s DPH.

Před obdržením zakoupeného telefonu byl proveden průzkum, zda existují dokumentace, nebo zda se již někdo, s touto značkou, pokusil o něco podobného. Kromě oficiálního letáku s velice obecnými parametry přístroje nebylo možné nalézt žádné veřejné dokumentace. Stejně tak nebyly nalezeny žádné informace o tom, že by někdo prováděl, třeba i rozbor, tohoto, či jiného zařízení od této značky. Ihned po obdržení bylo zahájeno vlastní snažení o zpřístupnění GPRS.

Obrázek 4.1: Zakoupený mobilní telefon STK R45i

4.1.4 Informativní kontaktování výrobce

Bez existujících dokumentací bylo rozhodnuto kontaktovat samotného výrobce. Byl zaslán email s vysvětlením tohoto projektu a žádostí o poskytnutí dokumentačních podkladů. Napsáno bylo celkem na 3 oddělení této britské společnosti, přesněji na obchodní oddělení, oddělení podpory a servisní centrum. Do jednoho týdne přišla shodná odpověď od všech tří kontaktovaných oddělení. Zprávou bylo, že dokumentace nejsou schopni poskytnout a nemohou nijak pomoci.


Dear Tomáš Moravec, Hello Tomas
Currently we cannot support you with this request ..

Your Santok Ltd Team,
Best Regards

STK Support Team
www.stklife.com

Obrázek 4.2: Jedna z odpovědí od společnosti STK

4.1.5 Rozebrání mobilního telefonu

Vzhledem k negativní odpovědi výrobce byl zahájen vlastní výzkum. Prvním pokusem byla komunikace s mobilním telefonem skrze USB, stejně jako to bylo možné u telefonů s API k těmto účelům. Pokusy o přímou komunikaci po sériové lince byly systematicky prováděny po dobu tří týdnů. Bez úspěchu.

Po dohodě s vedoucím práce, bylo rozhodnuto porušit záruku a rozebrat telefon. Primární zaměření bylo na případné servisní piny pro připojení a diagnostiku. Po hlubším prozkoumání se podařilo nalézt servisní piny, kde byla snaha je co nejlépe analyzovat a skrze ně komunikovat s mobilním telefonem. Po několika týdnech opět bez úspěchu.

Obrázek 4.3: Servisní piny

4.1.6 Závěr rozboru

Po dvou měsících snažení se z mobilního telefonu nepodařilo získat žádnou informaci. První týden se podařilo zachytávat neznámé signály, nicméně po hlubším přezkoumání spektrálním analyzátorem bylo zjištěno, že se nejedná o číslicový signál.

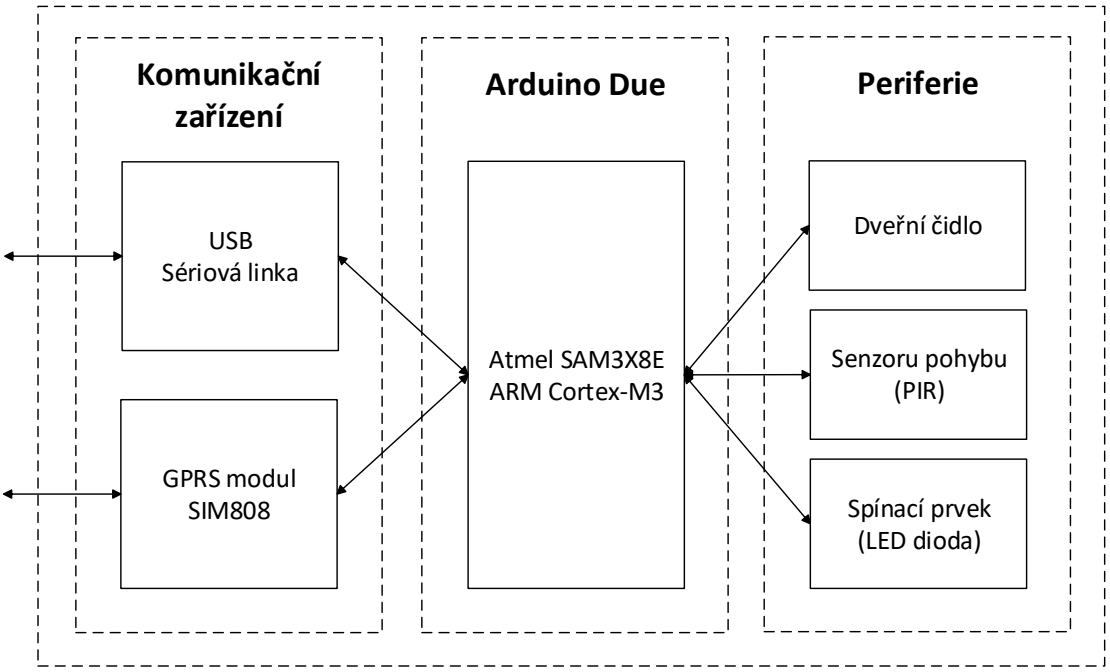
Po dvou měsících práce, byla možnost využití tlačítkového telefonu, pro přístup do sítě GPRS, uzavřena. Od využití mobilního telefonu bylo tedy upuštěno a jako vstupní brána do GPRS byl pořízen samostatný GSM/GPRS modul.

4.2 Prototyp domovního systému

Účelem prototypu je vývoj a testování firmware a vzájemné komunikace s jednotlivými aplikacemi. Zapojení musí být modulární, snadno přepojitelné a jednoduše přístupné k měření.

4.2.1 Blokové schéma

Prototyp je rozdělen do třech hlavních bloků, které spolu vzájemně komunikují. Pod blokovým schématem následuje detailní popis těchto jednotlivých částí.



Obrázek 4.4: Blokové schéma zapojení prototypu domovního systému

Komunikační zařízení: Tento blok obsahuje všechna zařízení zprostředkovávající komunikaci. Vždy potřebujeme komunikovat pomocí USB a internetového připojení. V případě prototypu složí k USB komunikaci vestavěn řadič USU, který se nachází na desce Arduino. Internetové připojení zprostředkovává GPRS modul, v případě prototypu model SIM808, případně podobný.

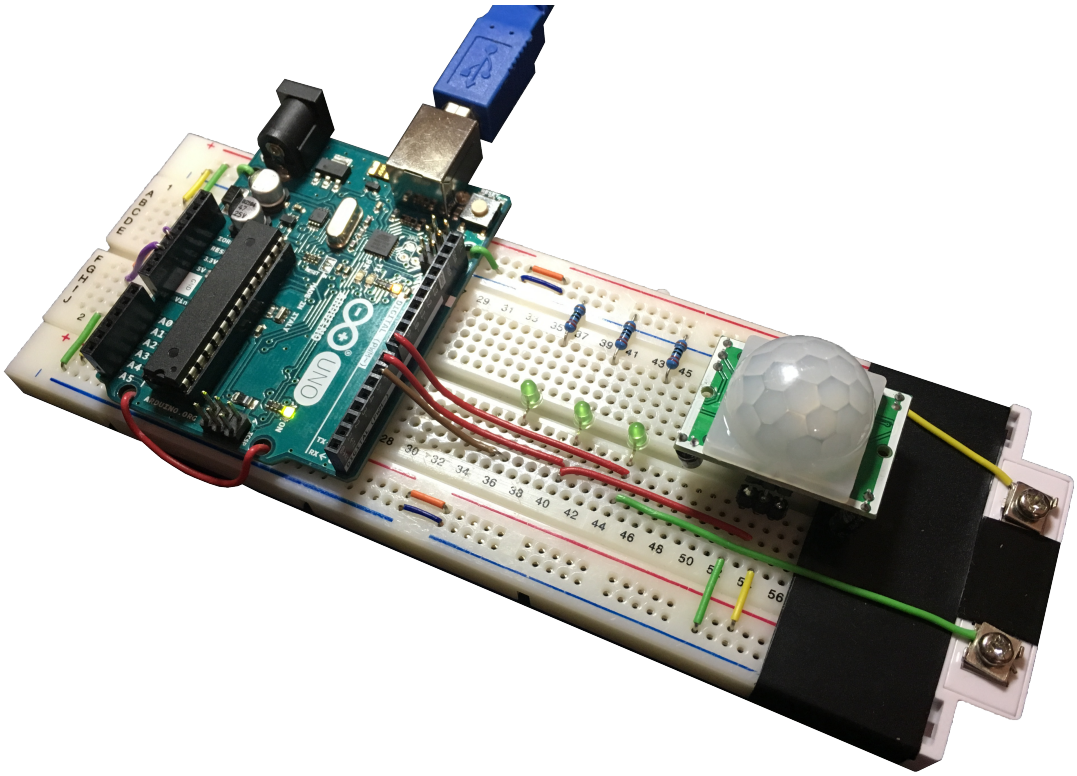
Arduino Due: Hlavní blok Arduino Due obsahuje vývojovou desku Arduino Due, která slouží jako centrální jednotka celého domovního systému. Do této jednotky jsou zapojeny veškeré periferie, ať již komunikátory, nebo senzory a spínače.

Periferie Blok periferie obsahuje všechna čidla, senzory, spínače a podobná zařízení, které má za úkol řídící jednotka kontrolovat. Pro demonstraci základních funkcionalit bylo zvoleno dveřní čidlo, senzor pohybu PIR a spínací prvek typu LED dioda. Všechny tyto periferie lze jednoduše nahradit jinou libovolnou součástkou, případě nové přidat.

4.2.2 Výsledný prototyp

Prototyp je schopen simulovat reálné nasazení domovního systému. Obsahuje řídící desku Arduino, GSM modul pro připojení do sítě internet, jeden spínací prvek (LED)

a dva senzory (PIR a dveřní čidlo). Každý senzor má svou vlastní kontrolní diodu. Ta signalizuje, zda je senzor sepnutý či nikoliv.



Obrázek 4.5: Jeden z prototypů na nepájivém poli

4.3 Firmware

Hlavním účelem firmware (software pro řídící jednotku) je možnost kompletního řízení domovního systému a umožnění komunikace, včetně poskytování API pro přístup aplikací. Software byl vyvinut pro platformu Arduino (čip ARM Cortex-M3) v programovacím jazyce C++ s nadstavbou Wiring (knihovna pro řízení hardwaru) [27], za pomocí vývojového prostředí Visual Studio 2017 s rozšířením Visual Micro [24], které poskytuje zvýrazňování syntaxe knihovny Wiring a umožňuje komplikaci a nahrávání kódu přímo na desku Arduino.

Dále budou zmíněny pouze ty části, které jsou důležité pro pochopení nastavení zařízení a přístupu k aplikaci (API), samotný kód lze nalézt v příloze a zmiňován zde nebude.

4.3.1 Nastavení

Inicializační část slouží k prvotnímu nastavení mikrokontroleru před spuštěním. Toto nastavení lze měnit pouze před první kompilací kódu, za běhu programu jej změnit nelze. Jedinou cestou k jeho změně je tedy opětovná kompilace kódu s upraveným nastavením a následné nahrání do mikrokontroleru.

Nastavit lze:

- Komunikační rychlost (baudRate)
- Maximální počet senzorů
- Maximální počet spínačů
- Kódy chybových hlášek
- Klíčová slova API

4.3.2 API

Funkce API jsou nazvány podle své činnosti. Při volání jednotlivých funkcí se posílá nejdříve název funkce, které následují kulaté závorky, tedy „(“ a „)“, v nichž jsou uvedeny jednotlivé parametry. Každý jeden parametr je oddělen čárkou. Ve výsledku může výstup vypadat například takto „SetSensor(10,5)“. V následující tabulce jsou vypsány všechny dostupné funkce, typu getter.

Funkce	Parametry	Popis	Vrací
GetAllSensors()	/	Vrací list všech senzorů	id, pin, name, state
GetAllSwitches()	/	Vrací list všech spínačů	id, pin, name, state

Tabulka 4.1: Tabulka API - Getters

Ukázka vrácených senzorů:

Sensors((Id = 0,Pin = 8,Name = Door Sensor,State = 0,Type = 0)(Id = 1,Pin = 7,Name = PIR Sensor,State = 0,Type = 1))

Ukázka vrácených spínačů:

Switches((Id = 0,Pin = 6,Name = Led Switch,State = 0))

Další tabulka obsahuje výpis všech dostupných funkcí, typu setter.

Dotaz	Parametry	Popis	Vrací
SetSensorName()	id, name	Změna jména senzoru	OK, ERROR
SetSwitchName()	id, name	Změna jména spínače	OK, ERROR
SetSensorPin()	id, pin	Změna pinu senzoru	OK, ERROR
SetSwitchPin()	id, pin	Změna pinu spínače	OK, ERROR
SetSensorType()	id, type	Změna typu senzoru	OK, ERROR
SetSwitchState()	id, state	Změna stavu spínače	OK, ERROR
AddSensor()	pin, name, type	Přidání nového senzoru	OK, ERROR
AddSwitch()	pin, name, state	Přidání nového spínače	OK, ERROR
DeleteSensor()	id	Smazání senzoru	OK, ERROR
DeleteSwitch()	id	Smazání spínače	OK, ERROR

Tabulka 4.2: Tabulka API - Setters

U nastavovacích dotazů musí být „id“ a „pin“ číslo od 0 do 100, „type“ a „state“ musí být číselné hodnoty 0 (false) nebo 1 (true) a „name“ musí být maximálně 30 znaků dlouhé (30 bytů). Pro „state“ znamená 1 zapnuto, 0 vypnuto. Pro „type“ znamená 1 „normálně vypnuto / push to make“ a 0 „normálně zapnuto / push to break“. Pokud jedna z podmínek nebude splněna, dotaz bude odmítnut. Parametr „ID“ lze získat pomocí příkazu „GetAllSensors()“ a „GetAllSwitches“ z již existujících zařízení.

Ukázka přidání nového senzoru:

AddSensor(8, Muj novy senzor, 0)

Ukázka změny pinu senzoru:

SetSensorName(1, Moje nove jmeno senzoru)

Ukázka smazání spínače:

DeleteSwitch(1)

4.3.3 Notifikace

Řídící jednotka má uložené všechny poslední stavy senzorů a při každém průchodu hlavní smyčkou proběhne čtení, pomocí funkce „checkSensorStateChangedAndSendIfTrue()“. Pokud nový stav nesouhlasí se starým, je zaslána notifikace s ID zařízení a aktuálním stavem. Poté záleží na cílové aplikaci, jak informaci zpracuje.

Ukázka notifikace:

$$\text{Sensor}(\text{Id} = 1,\text{State} = 0)$$

4.3.4 Chybová hlášení

Chybová hlášení (při neúspěšném provedení operace nebo jiné chybě) byla původně tvořena pomocí stavových kódů HTML [14]. Z důvodu nedostatku paměti (při vývoji na platformě Arduino Uno) byla nahrazena jednoduchým „OK“ a „ERROR“. Kontrola chyb se provádí vždy při zpracovávání kterékoliv z operací podporující API. Vždy se kontroluje, zda je možné zapsat, zda se provedl zápis, zda je nová hodnota po přečtení stejná jako zapisovaná, a tak dále. Pokud nastane chyba, je vše vráceno do původního stavu. Pokud příkaz nebyl rozpoznán, nebo nesplňuje kritéria tohoto API, je vrácena chybová hláška o nerozpoznání příkazu.

Ukázka odpovědi při nerozpoznání příkazu:

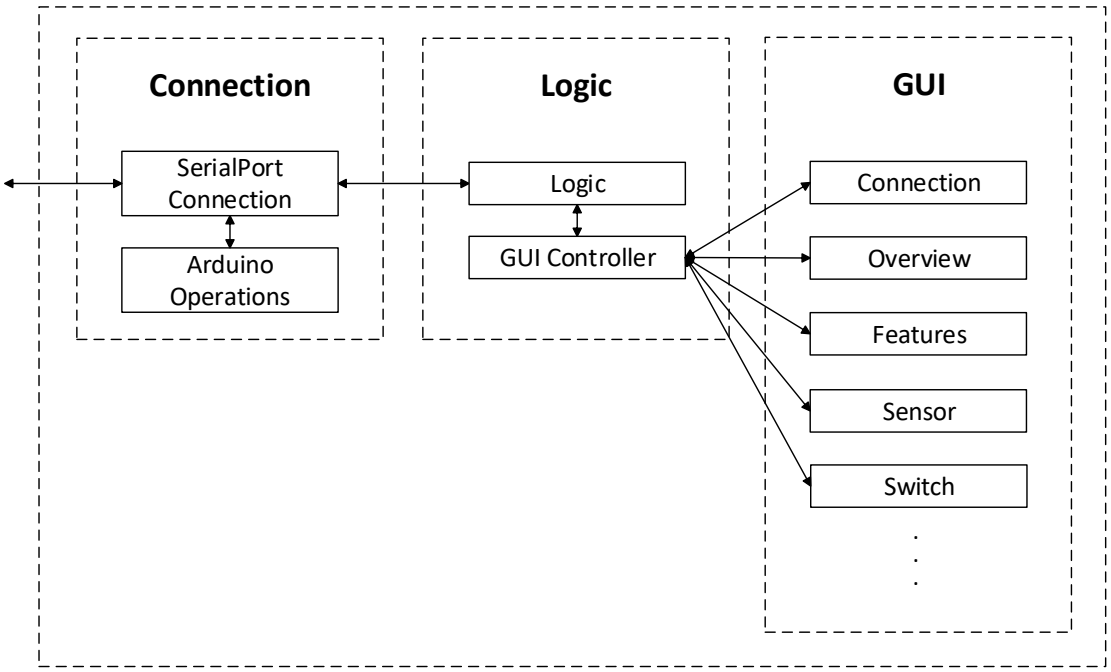
Command ’SetSensor(15,5)’ not recognized.

5 Desktopová aplikace

Hlavním účelem desktopové aplikace je možnost kompletního nastavení a monitorování zabezpečovacího zařízení. Aplikace se zaměřuje zejména na správu (vytváření, upravování, mazání) všech senzorů a spínačů. Software byl vyvinut pro operační systém Windows v programovacím jazyce C#, za pomocí IDE Visual Studio 2017. Design byl inspirovaný manuálem jednotného vizuálního stylu TUL [15].

5.1 Blokové schéma

Kód aplikace je rozdělen do třech hlavních bloků, které spolu vzájemně komunikují. Pod blokovým schématem následuje detailní popis těchto jednotlivých částí.



Obrázek 5.1: Blokové schéma objektového návrhu aplikace

5.1.1 Connection

Tento blok slouží k připojení řídící jednotky skrze sériovou linku (SerialPort Connection), po které komunikuje za pomocí API, poskytované touto jednotkou. Po navázání spojení se určí typ připojené řídící jednotky, kvůli poskytnutí seznamu použitelných pinů. Výstupem tohoto bloku jsou funkce pro jednoduché zapisování a čtení z jednotky (Arduino Operations), včetně kompletního seznamu použitelných pinů této jednotky. Seznam podporovaných řídících jednotek Arduino lze nalézt v části Kompatibilita.

5.1.2 Logic

Logic je hlavní vlákno, které načítá veškeré viditelné části aplikace (GUI Controller) z bloku GUI a zároveň zprostředkovává komunikaci mezi blokem Connection a jednotlivými funkcemi v oknech aplikace (GUI). Nachází se zde tedy napojení na Arduino Operations z bloku Connection. Zároveň zpracovává všechny informace, přicházejících směrem od řídící jednotky (například notifikace, chybová hlášení, stavové kódy atd.).

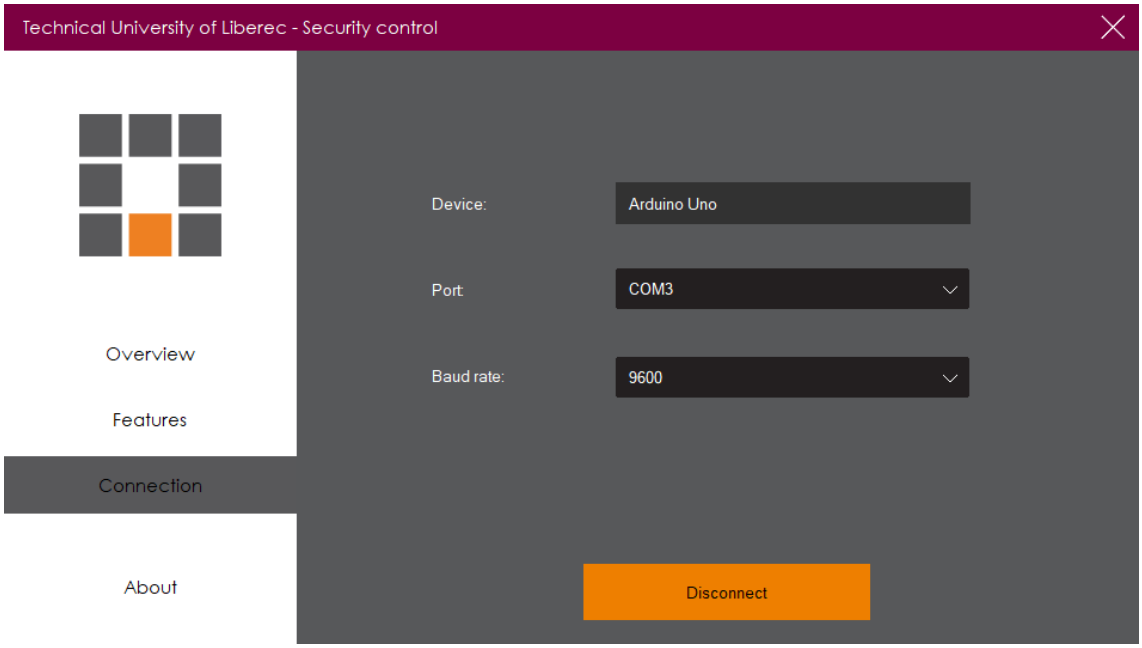
5.1.3 GUI

Jednotlivé funkce aplikace, včetně jejich grafického uživatelského rozhraní poskytuje blok GUI. Například Overview poskytuje GUI pro přehled všech připojených senzorů a spínačů, Senzor poskytuje GUI pro vytváření a upravování senzorů. Tyto funkce jsou využívány v bloku Logic, za pomocí GUI Controlleru, který je načítá do hlavního okna aplikace.

5.2 Grafické uživatelské rozhraní

V této části textu budou zmíněny pouze ty části GUI, které jsou důležité pro pochopení ovládání aplikace.

5.2.1 Připojení/odpojení řídící jednotky (Connection)



Obrázek 5.2: Okno s možností připojení/odpojení řídící jednotky

Záložka Connection slouží k připojení či odpojení řídící jednotky. Tato záložka hlídá, zda je připojení stále aktivní. Při ztrátě spojení ihned informuje uživatele a upraví chování celé aplikace. Záložka rovněž hlídá, zda se neobjevilo nové zařízení k připojení, pokud ano, informuje uživatele a zobrazí nově dostupná zařízení.

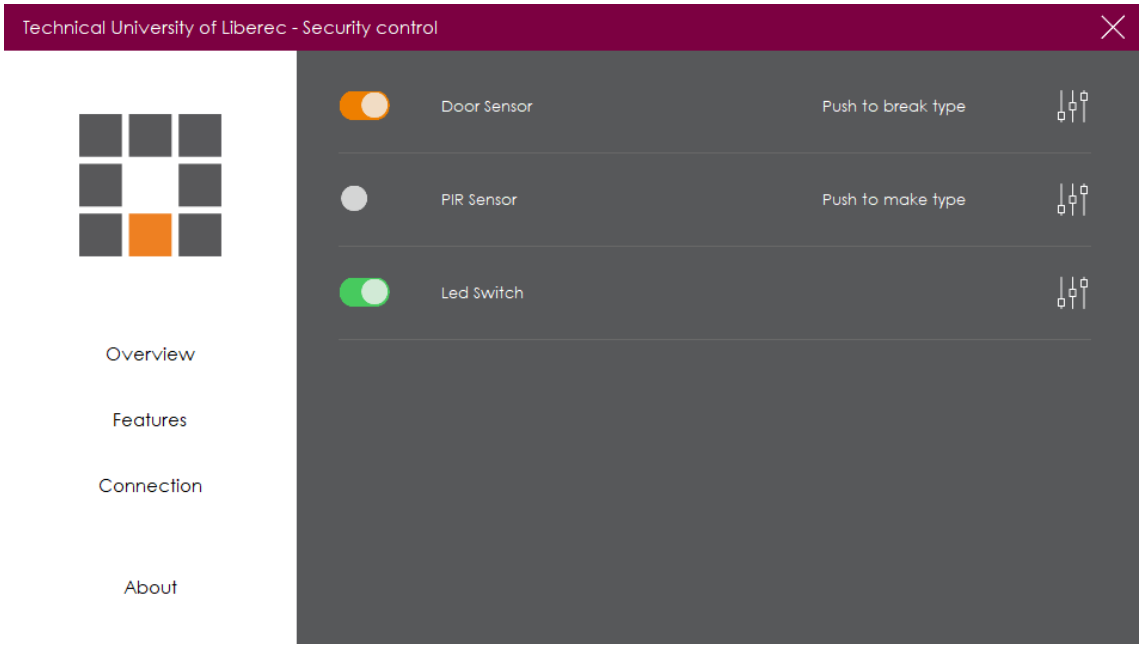
Zařízení pro připojení se vybírá pomocí Port, které obsahuje seznam všech dostupných COM portů. K vybranému COM portu je následně zjištěn název zařízení, který se objeví v Device. Rychlost komunikace (BaudRate), lze vybrat ze standardních komunikačních rychlostí [9]. Vybraná komunikační rychlost musí být stejná, jako v řídící jednotce. Automaticky nastavená hodnota 9600 by měla odpovídat přednastavené rychlosti řídící jednotky.

Po zvolení řídící jednotky a komunikační rychlosti stačí zmáčknou oranžové tlačítko Connect pro připojení, případně Disconnect pro odpojení.

Dostupné komunikační rychlosti (BaudRate):

- 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, 115200

5.2.2 Přehled připojených komponent (Overview)

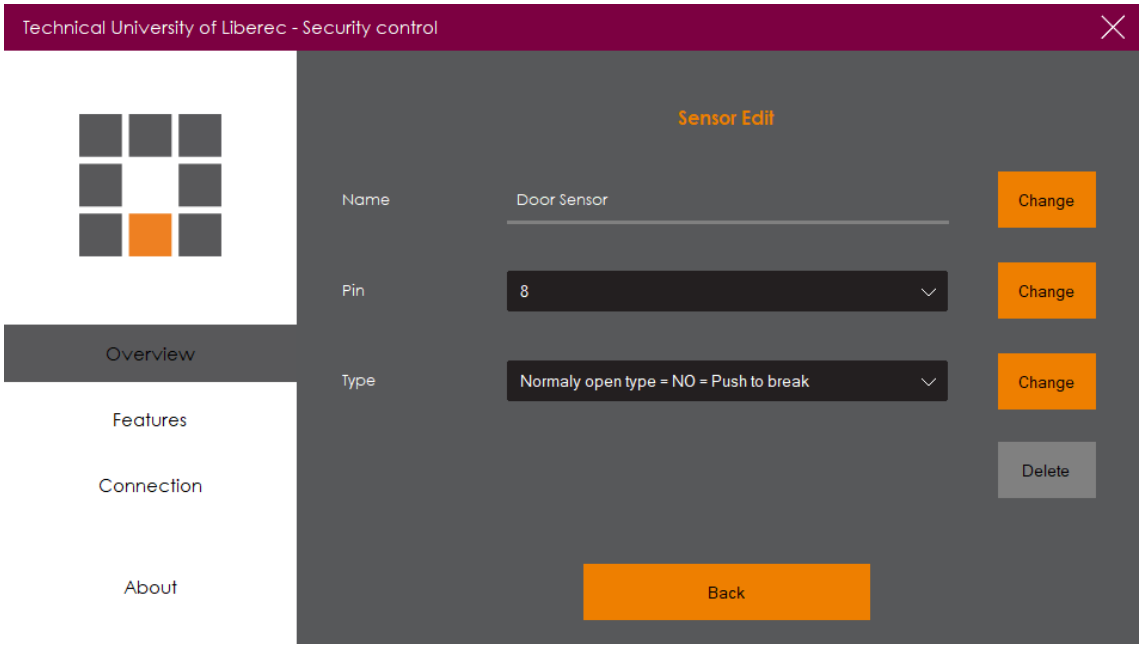


Obrázek 5.3: Okno s přehledem jednotlivých komponent

Přehled připojených komponent (Overview) slouží pro zobrazení všech dostupných spínačů a senzorů, které jsou v řídící jednotce dostupné. Obsah se generuje dynamicky pod sebe, lze tedy zobrazit neomezený počet zařízení (po přetečení viditelné části se objeví scrollbar). Každý dynamický řádek je zobrazen dle svého typu, tedy zda se jedná o senzor nebo spínač. U spínačů lze měnit jejich aktuální stav, u senzorů lze aktuální stav pouze sledovat. V obou případech lze měnit interní nastavení každého zařízení, pomocí tlačítka na pravé straně. V případě, že žádná zařízení neexistují, je uživatel upozorněn.

Žádná data nejsou editována v objektech uvnitř aplikace, ale žádosti o změnu jsou zasílány do řídící jednotky. Po obdržení odpovědi se buď projeví požadovaná změna, nebo je zobrazeno chybové hlášení. V obou případech je uživatel upozorněn zda změna proběhla, či zda neproběhla.

5.2.3 Nastavení senzoru (Sensor Edit)



Obrázek 5.4: Okno s nastavením senzoru

Nastavení senzoru slouží k změně interního nastavení uvnitř řídící jednotky. Lze měnit název zařízení (omezeno na 30 znaků), pin na kterém se senzor nachází, typ senzoru (rozpínací, spínací) a poslední možností je smazání senzoru. Výběr pinů závisí vždy na připojené desce. Pro tento účel byl vytvořen kompletní seznam všech testovaných a kompatibilních desek, které lze k softwaru připojit. Kompletní seznam desek, které lze použít, je k nalezení v části textu nazvané Podporované řídící jednotky.

Žádná data nejsou editována v objektech uvnitř aplikace, ale žádosti o změnu jsou zasílány do řídící jednotky. Po obdržení odpovědi se buď projeví požadovaná změna, nebo je zobrazeno chybové hlášení. V obou případech je uživatel upozorněn zda změna proběhla, či zda neproběhla.

5.2.4 Nastavení spínače (Switch Edit)

Nastavení je téměř identické s předchozím, proto není nutné náhledový obrázek. Lze měnit název (omezeno na 30 znaků), pin na kterém se spínač nachází. Jedinou změnou možnost změny stavu tohoto spínače (zapnuto/vypnuto).

Žádná data nejsou editována v objektech uvnitř aplikace, ale žádosti o změnu jsou zasílány do řídící jednotky. Po obdržení odpovědi se buď projeví požadovaná změna, nebo je zobrazeno chybové hlášení. V obou případech je uživatel upozorněn zda změna proběhla, či zda neproběhla.

5.2.5 Možnosti programu (Features)

Tato záložka zpřístupňuje různá nastavení a funkce, které nejsou běžně potřebné.

Přehled funkcí:

- Přidat nový senzor
- Přidat nový spínač
- Znovu načíst všechna data z řídící jednotky
- Zobrazovat notifikace, pokud se změní stav senzoru (přepínač)
- Zobrazovat notifikace, pokud se senzor navrátí do výchozího stavu (přepínač)

5.2.6 Notifikace

Aplikace podporuje širokou škálu notifikací, upozornění a varovných hlášení, která slouží pro upozornění uživatele na nastalou situaci. Například se jedná o změnu stavu pinů, chybové stavy řídící jednotky, nebo odpojení zařízení. Notifikace se zachytávají v hlavním okně, v části kódu nazvané „Autocalled functions (Events from Arduino)“, pomocí eventů.

Přehled implementovaných notifikací:

- Připojení zabezpečovacího zařízení
- Odpojení zabezpečovacího zařízení
- Ztráta spojení zabezpečovacího zařízení
- Automatický pokus o připojení se nezdařil
- Stav spínače přenastaven
- Stav senzoru se změnil
- Interní chyba řídící jednotky
- Jméno se podařilo/nepodařilo změnit
- PIN se podařilo/nepodařilo změnit
- Typ senzoru se podařilo/nepodařilo změnit
- Nový senzor se podařilo/nepodařilo přidat
- Nový spínač se podařilo/nepodařilo přidat
- Komponenty arduina se podařilo/nepodařilo znovu načíst
- Nastavení notifikací změněno

5.3 Kompatibilita

5.3.1 Testované verze Windows 10

Desktopová aplikace byla testována pod operačním systémem Windows 10 ve všech (v době psaní práce) vydaných sestavách a pro tento systém byla také optimalizována. Nicméně cílová platforma Microsoft .NET Framework 4.5.2 [26] by měla zajišťovat zpětnou kompatibilitu až do systému Windows Vista. Jmenovitě pro Windows Vista SP2, Windows 7 SP1, Windows 8, Windows 8.1, Windows 10 a novější [16].

Seznam všech testovaných sestav systému Windows 10:

- Version 1507 - Build 10.0.10240 (čistá instalace systému)
- Version 1511 - Build 10.0.10586 (November Update)
- Version 1607 - Build 10.0.14393 (Anniversary Update)
- Version 1703 - Build 10.0.15063 (Creators Update)
- Version 1709 - Build 10.0.16299 (Fall Creators Update)
- Version 1803 - Build 10.0.17134 (April 2018 Update)

5.3.2 Podporované řídící jednotky

Řídící jednotka, v případě tohoto prototypu deska Arduino, existuje v několika variantách [3]. Jako řídící jednotku je vhodné použít jednu z řady produktů ENHANCED FEATURES (Mega, Due a další) [3], vzhledem k tomu, že obsahují dostatečné množství paměti [10] pro udržování většího množství senzorů a spínačů. Vybrané jednotky, které jsou pro tento projekt použitelné, byly otestovány a zařazeny mezi jednotky podporované. Tyto podporované jednotky se vyznačují tím, že desktopová aplikace obsahuje seznam použitelných pinů pro každou z těchto desek. Tento seznam se získává ihned po připojení jednotky, a to na základě názvu desky poskytnutého řadičem. Pokud připojená deska není v seznamu, použije se seznam pinů pro desku Arduino Uno. V takovémto případě není zaručena správná funkčnost softwaru. Pokud použitý pin není v seznamu pinů desky, zařízení na tomto pinu nebude možné ovládat, upravovat, ani s ním jakkoliv pracovat.

Seznam všech podporovaných řídících jednotek:

- Arduino/Genuino Uno (nedoporučeno)
- Arduino/Genuino Mega
- Arduino/Genuino Mega 2560
- Arduino Due (doporučeno)

Seznam desek kompatibilních s Arduino/Genuino Uno:

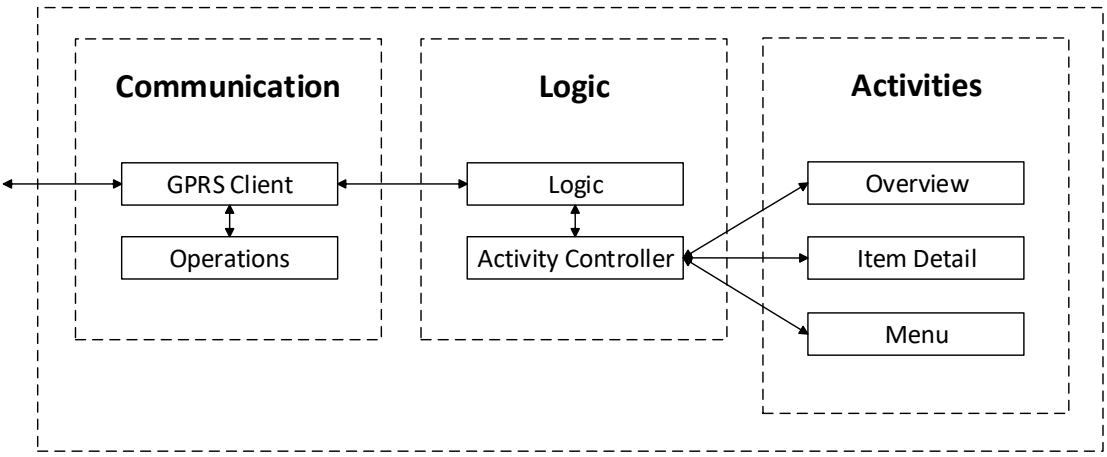
- Arduino/Genuino Duemilanove
- Arduino/Genuino Diecimila
- Arduino/Genuino Nano
- Arduino/Genuino NG
- Arduino/Genuino Mini
- Arduino/Genuino Extreme
- Arduino/Genuino USB
- Arduino/Genuino Serial
- Arduino/Genuino 101

6 Mobilní aplikace

Hlavním účelem mobilní aplikace je čisté monitorování řídící jednotky. Není tedy možno měnit jakékoliv nastavení (k čemuž slouží desktopová aplikace). Tato aplikace byla vyvinuta pro mobilní operační systém Android 4.0.3 (Ice Cream Sandwich - API 15) a vyšší, v programovacím jazyce Java, za pomocí IDE Android Studio 2.3.2. Design byl inspirovaný manuálem jednotného vizuálního stylu TUL [15].

6.1 Blokové schéma

Kód aplikace je rozdělen do třech hlavních bloků, které spolu vzájemně komunikují. Pod blokovým schématem následuje detailní popis těchto jednotlivých částí.



Obrázek 6.1: Blokové schéma objektového návrhu aplikace

6.1.1 Communication

Tento blok slouží k připojení řídící jednotky skrze internetové připojení (GPRS), za pomocí mobilní datové sítě. Toto připojení komunikuje pomocí API, poskytované řídící jednotkou. Po navázání spojení se zažádá o všechna dostupná data, která může

řídící jednotka poskytnout. Samotnou komunikaci zpracovává vlákno GPRS Client, které je napojené na Operations, kde se nachází implementace API.

Aby internetová komunikace fungovala, musel být napsán komunikační server, který je pevným bodem v síti. Server na dané IP adrese a portu, zajišťuje komunikaci všech existujících řídících jednotek a mobilních aplikací. Více o této problematice v části textu nazvané Komunikační server.

6.1.2 Logic

Blok Logic je hlavní vlákno, které díky Activity Controlleru načítá veškeré viditelné části aplikace z bloku Activities a zároveň zprostředkovává komunikaci mezi blokem Communication a jednotlivými funkcemi v oknech aplikace (Activities). Nachází se zde tedy napojení na Arduino Operations z bloku Communication. Zároveň zpracovává všechny informace, přicházejících směrem od řídící jednotky (například notifikace, chybová hlášení, stavové kódy atd.).

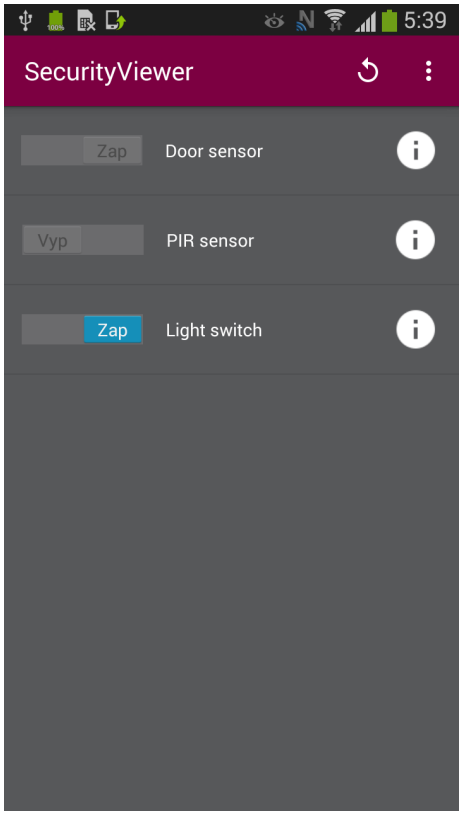
6.1.3 Activities

Jednotlivé funkce aplikace, včetně jejich GUI, poskytuje blok Activites, v Androidu nazvaném jako Activity. Například Overview poskytuje Activity pro přehled všech připojených senzorů a spínačů, Item Detail poskytuje Activity přehled o všech dostupných informací, které o vybrané komponentě aplikace má, a Menu poskytuje Acitivity s nastavením. Tyto funkce jsou využívány v bloku Logic, za pomocí Activity Controlleru, který je načítá do hlavního okna aplikace.

6.2 Grafické uživatelské rozhraní

V této části textu budou zmíněny pouze ty části GUI (Activities), které jsou důležité pro pochopení ovládání aplikace.

6.2.1 Přehled připojených komponent



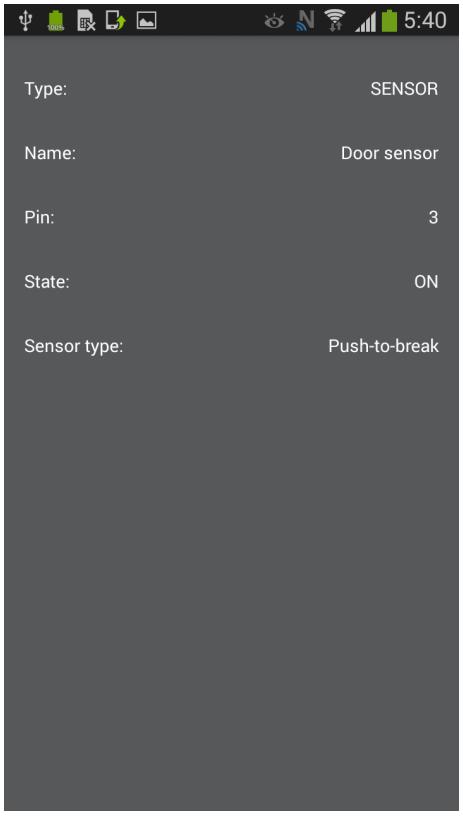
Obrázek 6.2: Okno s přehledem jednotlivých komponent

Stejně jako v případě desktopové aplikace, slouží přehled připojených zařízení pro zobrazení všech dostupných spínačů a senzorů, které jsou v řídící jednotce dostupné. Obsah se generuje dynamicky pod sebe, lze tedy zobrazit neomezený počet zařízení (po přetečení viditelné části se objeví scrollbar). Každý dynamický řádek je zobrazen dle svého typu, tedy zda se jedná o senzor nebo spínač. U spínačů lze měnit jejich aktuální stav, u senzorů lze aktuální stav pouze sledovat. V obou případech lze zobrazit interní nastavení každého zařízení, pomocí tlačítka na pravé straně. V případě, že žádná zařízení neexistují, je uživatel upozorněn.

Tlačítko na pravé straně „i“, tedy značí zobrazení podrobností vybrané komponenty. Dále v horní liště vidíme zatočenou šipku, která značí ruční aktualizaci všech zařízení. Posledním prvkem horního menu jsou tři tečky, takzvané „Hamburger Menu“, které složí pro otevření nastavení aplikace.

Žádná data nejsou editována v objektech uvnitř aplikace, ale žádosti o změnu jsou zasílány do řídící jednotky. Po obdržení odpovědi se buď projeví požadovaná změna, nebo je zobrazeno chybové hlášení. V obou případech je uživatel upozorněn zda změna proběhla, či zda neproběhla.

6.2.2 Podrobnosti vybrané komponenty



Obrázek 6.3: Detailní přehled nastavení komponenty

Zobrazení podrobností o vybrané komponentě ukazuje všechny dostupné údaje, které má aplikace k dispozici. Běžně se tak jedná o typ zařízení (senzor, spínač), název zařízení (omezeno na 30 znaků), číslo pinu, na kterém je zařízení připojení, a stav zařízení (zapnuto, vypnuto). Pokud se jedná o senzor, je zde vyobrazen i typ vybraného senzoru (spínací, rozpínací).

Vrátit se lze tlačítkem zpět, které má každý mobilní telefon se systémem Android implementované (ať již softwarově nebo hardwarově).

Seznam všech podrobností komponenty:

- Typ (senzor, spínač)
- Název (maximálně 30 znaků)
- Pin (číslo pinu, na kterém je zařízení připojeno)
- Stav (zapnuto, vypnuto)
- Typ senzoru (spínací, rozpínací)

6.3 Kompatibilita

Aplikace byla testována na mobilním telefonu Samsung Galaxy S3 s operačním systém Android 4.0 Ice Cream Sandwich a Android 4.3 Jelly Bean a také na telefonu Sony Xperia Z1 s operačním systémem Android 5.1.1 Lollipop. Pro tyto verze systémů byla také optimalizována. Nicméně cílová verze systému Android 4.0 Ice Cream Sandwich by měla zajišťovat zpětnou kompatibilitu až do nejnovějších verzí systému (v době vydání Android 8.1 Oreo).

Seznam všech testovaných verzí systému Android:

- API 14: Android 4.0 (Ice Cream Sandwich)
- API 16: Android 4.1 (Jelly Bean)
- API 21: Android 5.0 (Lollipop)

Seznam všech podporovaných verzí systému Android:

- API 14: Android 4.0 (Ice Cream Sandwich)
- API 15: Android 4.0.3 (Ice Cream Sandwich)
- API 16: Android 4.1 (Jelly Bean)
- API 17: Android 4.2 (Jelly Bean)
- API 18: Android 4.3 (Jelly Bean)
- API 19: Android 4.4 (KitKat)
- API 21: Android 5.0 (Lollipop)
- API 22: Android 5.1 (Lollipop)
- API 23: Android 6.0 (Marshmallow)
- API 24: Android 7.0 (Nougat)
- API 25: Android 7.1.1 (Nougat)
- API 26: Android 8.0 (Oreo)
- API 27: Android 8.1 (Oreo)

7 Webová aplikace

Poslední fáze diplomové práce bylo vytvoření webového prostředí pro monitoring domácího systému. Webová aplikace, čerpající data z komunikačního serveru byla otestována a zprovozněna. Webovou aplikaci čekají finální grafické úpravy, které budou k nalezení na přiloženém CD.

Aplikace je napojena a byla vyvíjena spolu s komunikačním serverem. Slouží jako jedna z možností zobrazení historie komunikace. Otevření aplikace je možné na adrese IP serveru.

8 Komunikační server

K zajištění komunikace mezi řídící jednotkou a mobilní aplikací fungovala, musel být napsán komunikační server, který má pevnou adresu v síti. Server na dané IP adrese a portu, zajišťuje komunikaci všech existujících řídících jednotek a mobilních aplikací. Díky tomuto spojení si ukládá přenášená data, která jsou následně používána k zobrazení statistik ve webové aplikaci. Server teoreticky umožňuje připojení libovolného počtu mobilních aplikací k jedné řídící jednotce naráz.

8.1 Slepé cesty vývoje

8.1.1 Přímá komunikace mezi zařízeními

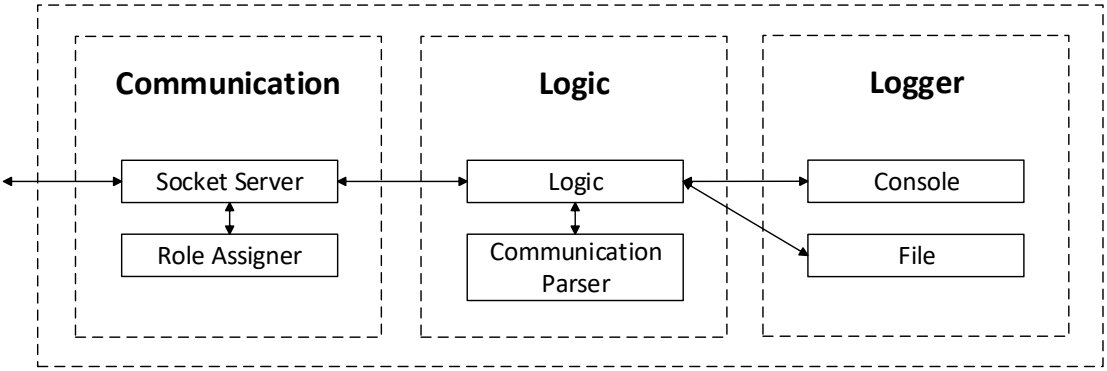
Prvním vyvíjeným řešením bylo propojení řídící jednotky a aplikací bez komunikačního serveru. Po půl roce výzkumu byla tato varianta zavrhnuta, protože veřejné IP adresy v mobilní datové sítí jsou dynamické (včetně IPv6). Bez pevného bodu v internetu, nebylo možné určit adresu ostatních zařízení.

8.1.2 Implementace serveru v řídící jednotce

Druhým vyvíjeným řešením bylo vytvoření komunikačního serveru přímo v řídící jednotce. Po čase bylo odhaleno, že přidání další zátěže jednočipovému počítači, mělo za následek nestabilitu celého systému. Řídící jednotka nestíhala obsluhovat všechny požadavky najednou (jak hardwarové, tak softwarové) a s rostoucí komunikací začala mít nedostatek paměti. Stejně tak jako předchozí řešení, i tato varianta byla po půl roce výzkumu zavržena.

8.2 Blokové schéma

Kód aplikace je rozdělen do třech hlavních bloků, které spolu vzájemně komunikují. Pod blokovým schématem následuje detailní popis těchto jednotlivých částí.



Obrázek 8.1: Blokové schéma objektového návrhu komunikačního serveru

8.2.1 Communication

Tento blok obsahuje vlákno Socket Server, které poslouchá na portu 6666 a navazuje komunikaci přes technologii Socket. Server udržuje všechna spojení aktivní a stará se o produkci všech zpráv ke všem cílům. Pokud je některé spojení přerušeno, zajistí informování všech ostatních uživatelů a uvolní zdroje.

Role Assigner se stará o oddělení neidentifikovaných klientů od těch identifikovaných. Takovýý klient se vyznačuje tím, že provedl autorizaci jako řídící jednotka, nebo mobilní aplikace. Dále se vyznačuje tím, že je mu umožněno komunikovat s ostatními protějšky, tedy řídící jednotka může kontaktovat všechny mobilní aplikace, a všechny mobilní aplikace mohou kontaktovat všechny řídící jednotky. Dokud není klient identifikovaný, není mu umožněno komunikovat s nikým jiným, než s komunikačním serverem.

8.2.2 Logic

Blok Logic je hlavní vlákno, které řídí tok komunikace identifikovaných klientů. Protékající data zpracovává a uchovává k zobrazení ve webové aplikaci. Dále se stará o tisk komunikace na zvolených výstupech z bloku Logger.

8.2.3 Logger

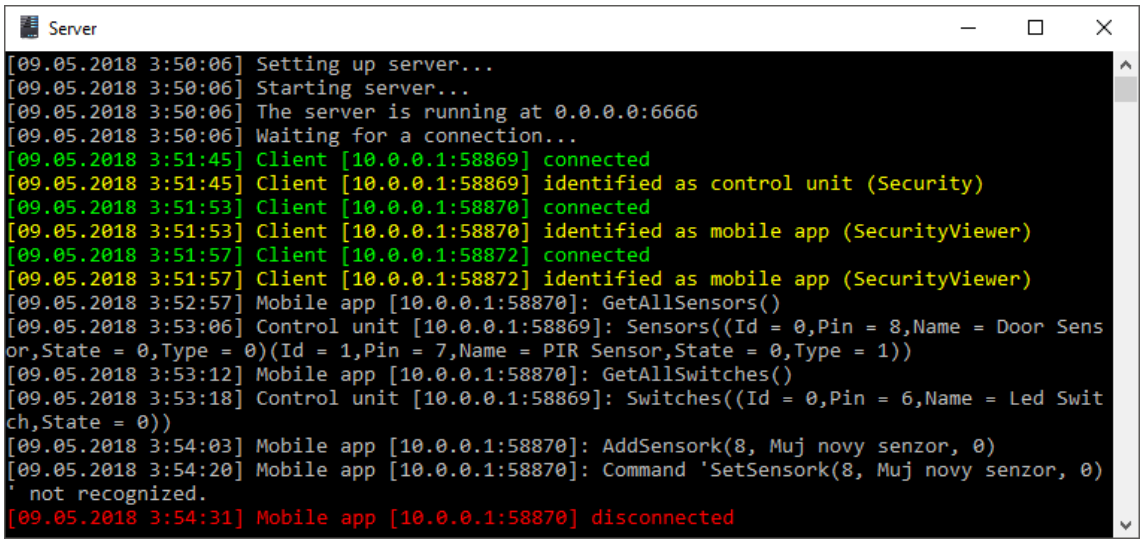
Logger slouží k výpisu či logování probíhající komunikace a dalších možných informací. Aktuálně je možné výpis provádět do konzole, která je ideální pro ladění uživatelem, případně je možný výpis do souboru, kde slouží pro zpětné dohledávání chyb.

8.3 Zprovoznění

Pro správnou funkcionalitu komunikačního serveru, je nutné server provozovat na veřejné IP adrese, s přesměrováním portu 6666 na lokální adresu serveru v interní síti.

8.4 Konzolový výstup

Pro lepší vývoj a testování byl vytvořen konzolový náhled na komunikaci serveru, kde jsou časově a barevně odlišeny různé operace.



Obrázek 8.2: Konzolové okno komunikačního server

Významy barev v serverovém logu:

- Zelená - Zařízení připojeno
- Žlutá - Zařízení identifikováno (řídící jednotka, nebo mobilní aplikace)
- Červená - Zařízení odpojeno
- Šedivá - Běžná komunikace

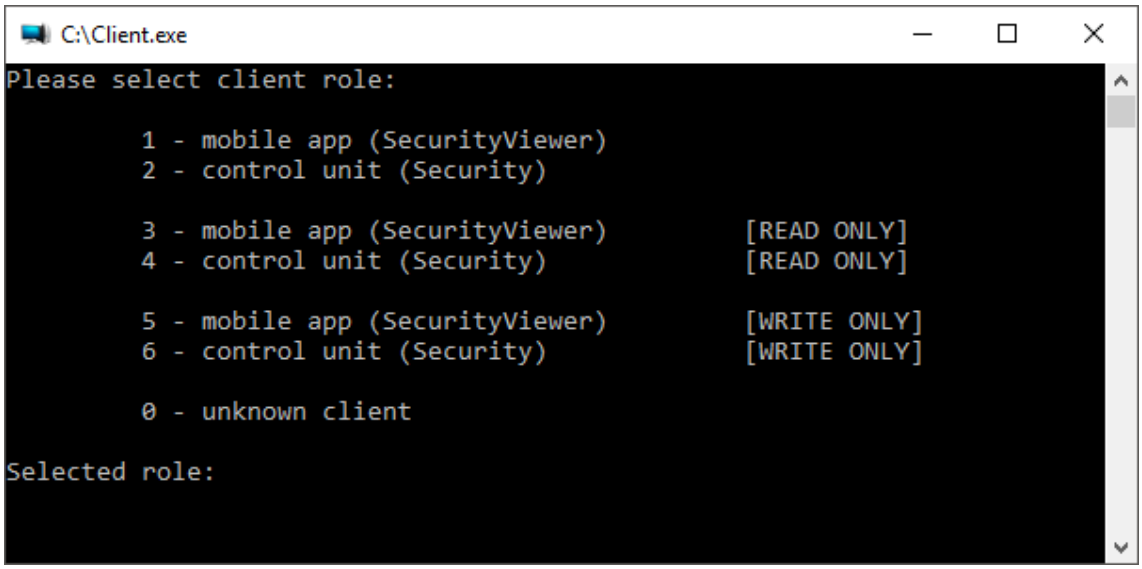
8.5 Testovací klient

Testovací klient vznik za účelem testování serverové části, bez nutnosti komunikace reálných zařízení nad reálnými daty. Klient umožňuje výběr z několika režimů, kde každý umožňuje simulovat něco jiného. Po výběru jednoho z režimů, se testovací

klient připojí na komunikační server pod zvoleným nastavením. Poté lze například zasílat testovací řetězce se seznamem komponent, odpovídat na API dotazy a podobné.

Seznam všech režimů testovacího klienta:

- Simulace mobilní aplikace
- Simulace mobilní aplikace (READ ONLY)
- Simulace mobilní aplikace (WRITE ONLY)
- Simulace řídící jednotky
- Simulace řídící jednotky (READ ONLY)
- Simulace řídící jednotky (WRITE ONLY)



Obrázek 8.3: Konzolové okno testovacího klienta

9 Závěr

V rešeršní části byly definovány základní pojmy z témat zabezpečovacích systémů a inteligentních domů, ze kterých se sestává inteligentní domovní systém. Dále byly definovány požadované vlastnosti jednotlivých součástí inteligentního domovního systému, kterými jsou domovní systém, desktopová, mobilní a webová aplikace, a komunikační server. Na základě požadovaných vlastností, byl sestaven seznam požadavků všech částí inteligentního domovního systému.

Na základě častých konzultací s vedoucím práce, byl navržen koncept cílového produktu, byly doporučeny realizační postupy, vybrán hardware, nebo bylo uvedeno více možností jeho výběru, kterými může být dosaženo výsledného zařízení. Dále byl pro každou z jednotlivých částí zvolen vhodný programovací jazyk a jedno, nebo více vývojových prostředí (IDE). Nakonec byl zvolen způsob datové komunikace, pomocí mobilních sítí GPRS.

Tvorba inteligentního domovního systému byla rozdělena na několik fází. První fáze byla tvorba prototypu, zbylé fáze vývoj jednotlivých softwarových produktů.

V první fázi byla na základě konceptu zvolena nejvhodnější kombinace komponent vývoj prototypu, která byla následně zakoupena. Při práci na prototypu bylo odhaleno, že jeden z původních cílů, využít tlačítkový mobilní telefon pro přístup do sítě GPRS, je nerealizovatelný, zvolen by tedy alternativní postup. Dále byl vyvinut firmware řídící jednotky, který byl úspěšně otestován na prototypu. Po dopsání vlastního kódu byly podrobně popsány všechny funkce programu.

Dále byla vyvinuta desktopová aplikace pro systém Windows, která byla plně otestována pro více verzí systému Windows 10. Následně byly všechny funkce programu a zajištěna kompatibilita pro starší verze systému Windows.

Při tvorbě mobilní aplikace bylo zjištěno, že přímá komunikace aplikace a řídící jednotky je nerealizovatelná. Nejdříve bylo nutné vytvořil komunikační server pro systém Windows, který zprostředkovával požadované spojení v síti internet. Následně byla mobilní aplikace pro systém Android plně dokončena. Aplikace byla otestována, všechny funkce popsány a nakonec byla zajištěna kompatibilita pro starší i novější verze systému Android.

Poslední fáze byla tvorba webového prostředí pro monitoring domácího systému. Webová aplikace, čerpající data z komunikačního serveru byla otestována a zprovozněna. Webovou aplikaci čekají během následujícího týdne finální grafické úpravy, které budou k nalezení na přiloženém CD.

Funkční prototyp, včetně všech příslušných aplikací byl otestován a byla zajištěna stabilitu celého systému.

Do budoucna by bylo vhodné předělat komunikační server ze systému Windows

na běžnější formu serveru, kterou bude možné spustit na některé z běžných hostingových služeb. V plánu je také vytvoření mobilní aplikace pro systém iOS.

Literatura

[1] Arduino introduction. Arduino [online]. [cit. 2017-05-23]. Dostupné z: https://www.arduino.cc/en/Guide/Introduction

[2] Arduino libraries. Arduino [online]. [cit. 2016-05-08]. Dostupné z: https://www.arduino.cc/en/Reference/Libraries

[3] Arduino Products [online]. [cit. 2018-05-08]. Dostupné z: https://www.arduino.cc/en/Main/Products

[4] Arduino programming language. Arduino [online]. [cit. 2016-05-08]. Dostupné z: https://www.arduino.cc/en/Reference/HomePage

[5] Arduino Schematic [online]. 1 s. [cit. 2016-05-07]. Dostupné z: https://www.arduino.cc/en/uploads/Main/Arduino_Uno_Rev3-schematic.pdf

[6] Arduino software. Arduino [online]. [cit. 2016-05-08]. Dostupné z: https://www.arduino.cc/en/Main/Software

[7] Arduino source code. GitHub [online]. [cit. 2016-05-08]. Dostupné z: https://github.com/arduino/Arduino/tree/1.6.8

[8] ATmega328/P Datasheet [online]. [cit. 2018-05-14]. Dostupné z: http://www.microchip.com/wwwproducts/en/ATmega328p

[9] BaudRates for communicating with the computer [online]. [cit. 2018-05-08]. Dostupné z: https://www.arduino.cc/en/serial/begin

[10] Compare board specs [online]. [cit. 2018-05-08]. Dostupné z: https://www.arduino.cc/en/Products/Compare

[11] Desktop Operating System Market Share Europe [online]. [cit. 2018-05-07]. Dostupné z: http://gs.statcounter.com/os-market-share/desktop/europe/#monthly-201704-201804-bar

[12] Elektronická zabezpečovací signalizace. Wikipedia [online]. [cit. 2017-05-23]. Dostupné z: https://cs.wikipedia.org/wiki/Elektronick%C3%A1_zabezpe%C4%8Dovac%C3%AD_signalizace

[13] Elektronické zabezpečovací systémy [online]. , 1 [cit. 2017-05-23]. Dostupné z: http://www.ezasys.cz/elektronicke-zabezpecovaci-systemy/

[14] FIELDING, R., UC IRVINE, J. GETTYS, et al. Hypertext Transfer Protocol – HTTP/1.1 [online]. 1999, , 176 [cit. 2017-05-27]. Dostupné z: http://www.ietf.org/rfc/rfc2616.txt

[15] Manuál jednotného vizuálního stylu Technické univerzity v Liberci [online]. , 27 [cit. 2017-05-27]. Dostupné z: http://www.ft.tul.cz/document/126

[16] Microsoft .NET Framework 4.5.2. Microsoft.com [online]. [cit. 2018-02-21]. Dostupné z: https://www.microsoft.com/en-us/download/details.aspx?id=42642

[17] Mobile Operating System Market Share Europe [online]. [cit. 2018-05-07]. Dostupné z: http://gs.statcounter.com/os-market-share/mobile/europe/#monthly-201704-201804-bar

[18] MORAVEC, Tomáš. Koncept nízkonákladového sledovacího zařízení pro osobní automobily: The concept of a low cost tracking device for personal cars. Liberec: Technická univerzita v Liberci, 2016.

[19] SATRAPA, Pavel. LaTeX pro pragmatiky [online]. 2011, 87 s. [cit. 2016-05-07]. Dostupné z: http://www.nti.tul.cz/~satrapa/docs/latex/latex-pro-pragmatiky.pdf

[20] SATRAPA, Pavel. Stručný přehled příkazů LaTeXu [online]. 2011, 2 s. [cit. 2016-05-07]. Dostupné z: http://www.nti.tul.cz/~satrapa/docs/latex/latex-prehled.pdf

[21] Security alarm. Wikipedia [online]. [cit. 2017-05-23]. Dostupné z: https://en.wikipedia.org/wiki/Security_alarm

[22] SIM908 AT Command Manual [online]. Jinzhong, 2011, 249 s. [cit. 2016-05-07]. Dostupné z: http://www.dfrobot.com/image/data/TEL0051/3.0/SIM908_AT%20Command%20Manua_V1.01.pdf

[23] SIM908 Hardware Design [online]. 2. Jinzhong, 2012, 53 s. [cit. 2016-05-07]. Dostupné z: http://www.niplesoft.net/blog/wp-content/uploads/2016/02/SIM908-Hardware-Design-V2.00-1.pdf

[24] Visual Micro [online]. [cit. 2017-05-23]. Dostupné z: http://www.visualmicro.com/

[25] VODA, Zbyšek. Průvodce světem Arduina [online]. Vydání první. Bučovice: Martin Stříž, 2015 [cit. 2016-05-07]. ISBN 978-80-87106-90-7.

[26] What's new in the .NET Framework 4.5.2. Microsoft.com [online]. [cit. 2018-02-21]. Dostupné z: https://docs.microsoft.com/en-us/dotnet/framework/whats-new/index#v452

[27] Wiring [online]. [cit. 2016-05-08]. Dostupné z: http://wiring.org.co/

A Obsah přiloženého CD

Struktura a obsah adresářů je následující:

/Dokumentace

Text bakalářské práce ve formátu pdf.

/Hardware

Podklady k hardwaru.

/Hardware/Dokumentace (Datasheets)

Hardwarová dokumentace od výrobců použitých součástek.

/Hardware/Fotografie

Fotografie součástek, zapojení, informačních šťítků.

/Software

Zdrojové kódy jednotlivých aplikací.

/Software/SecurityControl (Desktopová aplikace)

Desktopová aplikace pro operační systém Windows.

/Software/SecurityFirmware (Řídící jednotka)

Firmware pro řídící jednotku Arduino.

/Software/SecurityServer (Komunikační server)

Komunikační server a testovací klientská aplikace pro operační systém Windows.

/Software/SecurityViewer (Mobilní aplikace)

Mobilní aplikace pro operační systém Android.