



TECHNICKÁ UNIVERZITA V LIBERCI  
Fakulta mechatroniky, informatiky  
a mezioborových studií ■

# Zabezpečovací systém pomocí mobilního telefonu

## Magisterský projekt

*Studijní program:* N2610 – Elektrotechnika a informatika

*Studijní obor:* 1802T007 – Informační technologie

*Autor práce:* **Bc. Tomáš Moravec**

*Vedoucí práce:* doc. Ing. Josef Chaloupka, Ph.D.





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

# Security system using a mobile phone

## Project report

*Study programme:* N2610 – Electrical Engineering and Informatics

*Study branch:* 1802T007 – Information Technology

*Author:* **Bc. Tomáš Moravec**

*Supervisor:* doc. Ing. Josef Chaloupka, Ph.D.



Tento list nahrad' te  
originálem zadání.

## Prohlášení

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

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasa- huje do mých autorských práv užitím mého magisterského projektu pro vnitřní potřebu TUL.

Užiji-li magisterský projekt nebo poskytnu-li licenci k jeho 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.

Magisterský projekt jsem vypracoval samostatně s použitím uve- dené literatury a na základě konzultací s vedoucím mého magis- terského projektu a konzultantem.

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

Datum:

Podpis:

## Poděkování

Děkuji vedoucí práce panu doc. Ing. Josefu Chaloupkovy, Ph.D. za odborné vedení a poskytnuté informace při zpracování magisterského projektu.

## Abstrakt

Práce se zabývá problematikou zabezpečovacího systému, postaveného na platformě Arduino, řízeného z desktopové aplikace a s možností monitorování pomocí mobilní aplikace. Komunikace mezi zabezpečovacím systémem a mobilní aplikací má být realizována pomocí mobilního telefonu. V úvodu jsou definovány základní pojmy a požadované vlastnosti, na jejichž základě byly objednány jednotlivé komponenty. První část práce je věnována snaze o využití mobilního telefonu, jako prostředníka pro připojení do mobilní datové sítě. Tato cesta se ukázala jako slepá a byla tedy zvolena druhá alternativa, komunikace pomocí samostatného GPRS modulu. Výstupem práce je firmware, který řídí a zajišťuje komunikaci, desktopová aplikace pro nastavení a monitorování, a mobilní aplikace pro samostatné monitorování zabezpečovacího zařízení.

## Abstract

Not implemented yet.

# Obsah

Seznam zkratek . . . . .	9
<b>1 Úvod</b>	<b>10</b>
<b>2 Zabezpečovací systém</b>	<b>11</b>
2.1 Ústředna . . . . .	11
2.2 Ovladač . . . . .	11
2.3 Detektor . . . . .	11
2.4 Komunikátor . . . . .	12
<b>3 Požadované vlastnosti a zvolené komponenty</b>	<b>13</b>
3.1 Hardware . . . . .	13
3.1.1 Ústředna . . . . .	13
3.1.2 Detektory . . . . .	14
3.1.3 Komunikátor . . . . .	14
3.2 Software (Ovladač) . . . . .	15
3.2.1 Desktopová aplikace . . . . .	15
3.2.2 Mobilní aplikace . . . . .	16
<b>4 Hardware a firmware</b>	<b>17</b>
4.1 Využití mobilního telefonu . . . . .	17
4.1.1 Možné postupy . . . . .	17
4.1.2 Výběr telefonu . . . . .	17
4.1.3 První rozbor . . . . .	18
4.1.4 Kontaktování výrobce . . . . .	18
4.1.5 Hlubší rozbor . . . . .	19
4.1.6 Závěr rozboru . . . . .	19
4.2 Zapojení . . . . .	19
4.2.1 Blokové schéma . . . . .	19
4.2.2 Výsledný prototyp . . . . .	19
4.3 Firmware . . . . .	19
4.3.1 Nastavení . . . . .	20
4.3.2 API . . . . .	20
4.3.3 Notifikace . . . . .	21
4.3.4 Chybová hlášení . . . . .	22

<b>5</b>	<b>Desktopová aplikace</b>	<b>23</b>
5.1	Blokové schéma . . . . .	23
5.2	Spojení . . . . .	24
5.3	Přehled . . . . .	25
5.4	Nastavení senzoru . . . . .	26
5.5	Nastavení spínače . . . . .	26
5.6	Funkce programu a nastavení . . . . .	26
5.7	Notifikace . . . . .	27
<b>6</b>	<b>Mobilní aplikace</b>	<b>28</b>
<b>7</b>	<b>Závěr</b>	<b>29</b>
	<b>Literatura</b>	<b>29</b>
<b>A</b>	<b>Obsah přiloženého CD</b>	<b>32</b>



## Seznam zkratek

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

# 1 Úvod

Téma bylo vypsáno doc. Ing. Josefem Chaloupkou, Ph.D. a zaujalo mne díky své tematické návaznosti na mou bakalářskou práci s názvem „Koncept nízkonákladového sledovacího zařízení pro osobní automobily“ [1]. V bakalářské práci jsem se věnoval vývoji sledovacího zařízení založeného na platformě Arduino [7], které komunikovalo pomocí SMS zpráv s mobilním telefonem v přirozené řeči. Dalším logickým krokem pro mne bylo, jak uvádím v závěru své bakalářské práce, řízení z desktopové aplikace, případně z mobilní aplikace, skrze mobilní datovou síť. Po konzultaci s vedoucím jsem byl obeznámen s tím, že tato práce mi nejen umožní navázat na předešlou bakalářskou práci, ale i prozkoumat možnosti využití těch nejlevnějších mobilních telefonů, podporujících datové přenosy, jako náhradu za mnohdy drahé GPRS moduly, které slouží pro přenos dat do Internetu skrze mobilní síť GSM (zkráceně mobilní datová síť). Nadchla mě myšlenka, že by bylo možné takto využít ty staré mobilní telefony, které má mnoho z nás stále schované doma, z důvodu trendu dnešní doby, kterým je obměňování za nejnovější modely. O téma jsem se tedy přihlásil nejen z důvodu tematické návaznosti na mou bakalářskou práci, ale také proto, že mne obohatí o další nové zkušenosti, které mohu později využít v praxi. Mohu doplnit, že má bakalářská práce již vyústila ve spolupráci se společností Jablotron (BigClown Labs s.r.o.) a zájmu ze strany Škoda Auto, i proto se těším, kam mě nové poznatky dále posunou.

## 2 Zabezpečovací systém

Zabezpečovací systém, někdy také elektronická zabezpečovací signalizace, je zařízení, které vizuálně nebo akusticky vyhlašuje poplach a dává na vědomí, že nastaly nějaké potíže nebo došlo ke splnění sledované podmínky [2]. Jde tedy o zařízení, které slouží k ochraně osob a majetku. Systém je řízen ústřednou a může se spustit analogovou (např. dveřní 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 [3].

### 2.1 Ústředna

Ústředna je „mozek“ celého systému, který je propojen s ostatními prvky systému kabelem nebo bezdrátově. Obstarává komunikaci mezi jednotlivými komponenty systému, má v integrované paměti uložené nejdůležitější nastavení [3]. 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) [2].

### 2.2 Ovladač

Ovladač je prvek, který slouží k ovládání a případně též k programování ústředny. 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 [4]. 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) [1].

### 2.3 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 předají ústředně, která ji následně zpracuje [4]. Nejčastěji používané detektorové prvky jsou:

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

## 2.4 Komunikátor

Komunikátor je zařízení, které předává mimo objekt informaci o narušení objektu, případně o odchylce od normálního provozního stavu zabezpečovacího systému [4]. Nejčastější typy komunikátorů jsou:

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

## 3 Požadované vlastnosti a zvolené komponenty

V následující části se zaměřím na požadované vlastnosti zabezpečovacího zařízení, a jaké komponenty byly na jejich základě zvoleny.

### 3.1 Hardware

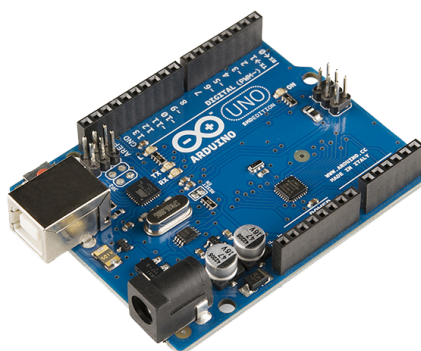
Z úvodní části je zřejmé, že kompletní zabezpečovací systém musí obsahovat detektory, komunikátor a ústřednu. Rozebereme si tedy jednotlivé části.

#### 3.1.1 Ústředna

Požadavky na ústřednu jsou takové, že musí být schopná komunikovat jak s desktopovou aplikací, tak s mobilní aplikací a zároveň musí řídit a zpracovávat jednotlivé komponenty. Zadání informuje o použití vývojové platformy Arduino [15], která je se svými periferiemi [10] a čipem Atmega [13], více než vhodnou pro tyto účely. Firmware pro ústřednu bude psán v jazyku C a C++ s nadstavbou Wiring (knihovny pro řízení hardwaru) [5], vývojové prostředí Visual Studio 2017 s rozšířením Visual Micro [6].

##### Zvolená ústředna:

- Arduino Uno Rev3 (700 Kč s DPH), ([arduino-shop.cz](http://arduino-shop.cz))



Obrázek 3.1: Vývojová platforma Arduino

### 3.1.2 Detektory

Požadavky nejsou kladeny pouze na samotné detektory, ale také na spínače, kterými bude možné spínat různá zařízení. Jako spínací prvky není nutné volit jednotlivé komponenty, stačí připravit implementaci jejich spínání. 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](https://www.aliexpress.com))



Obrázek 3.2: Detektor pohybu (PIR detektor)

### 3.1.3 Komunikátor

Požadavek na komunikátor je přenos dat do počítače a přes mobilní data. Zadáání jako hlavní komunikátor určuje mobilní telefon, pomocí kterého máme umožnit ústředně datové přenosy. Zvolen byl nový a zároveň nejlevnější mobilní telefon na trhu. Pro případ neúspěchu, při získávání zdrojů mobilního telefonu, byla zvolena alternativa v podobě GSM/GPRS modulu, který byl zakoupen z Číny, dodací doba této součástky, stejně jako na všechny ostatní, byla přes jeden kalendářní měsíc.

#### Zvolené komunikátory:

- Mobilní telefon STK R45i Black (449 Kč s DPH), ([alza.cz](https://www.alza.cz))
- GSM/GPRS modul (290 Kč s DPH), ([aliexpress.com](https://www.aliexpress.com))



Obrázek 3.3: Komunikační modul GSM/GPRS - SIM808

## 3.2 Software (Ovladač)

Výše zmíněné prvky musí být řízeny ovladačem, v mém případě aplikací jak desktopovou tak mobilní. Proto bude následující část rozdělena do těchto dvou částí.

### 3.2.1 Desktopová aplikace

Na desktopovou aplikaci jsou kladeny požadavky tak, aby byla schopna využít všech možností, které nám ústředna poskytuje. Musí být schopna kompletní správy a nastavení jednotlivých komponent (spínače a senzory) a tím pádem i komunikaci s ní. Cílový operační systém jsem si zvolil Windows 10, jakožto ve škole nejrozsířenější, programovací jazyk C# a vývojové prostředí Visual Studio 2017. Jednotlivé požadavky kladené na desktopovou aplikaci lze nalézt níže.

#### Požadavky na desktopovou aplikaci:

- Sledování aktuálních stavů komponent
- Přidávání nových komponent
- Odstraňování stávajících komponent
- Přejmenování komponent
- Změna pinů komponent
- Změna nastavení senzorů jako spínacích či rozpínacích
- Notifikace v případě změny stavů senzorů
- Přepínání stavů spínačů
- Připojování a odpojování ke zvolené ústředně

### 3.2.2 Mobilní aplikace

Mobilní aplikace je učena pouze jako monitorovací zařízení, ze kterého bude možné sledovat stavy jednotlivých senzorů, případně spínač spínače. Její návrh je tedy značně jednodušší oproti komplexnější desktopové aplikaci. Cílový operační systém jsem zvolil Android, jakožto nerozšířenější, programovací jazyk Java a vývojové prostředí a Android Studio.

#### Požadavky na mobilní aplikaci:

- Sledování aktuálních stavů komponent
- Přepínání stavů spínačů
- Připojování a odpojování ke zvolené ústředně



## 4 Hardware a firmware

### 4.1 Využití mobilního telefonu

Následující část pojednává o snaze využít GSM/GPRS modulu integrovaném v mobilním telefonu. Práce na této části zabrali přibližně dva měsíce, proto bude rozdělena do více částí.

#### 4.1.1 Možné postupy

První úkolem bylo, jak by bylo možné tyto zdroje z mobilního telefonu získat. Průzkumem existujících prací a internetu jsem mohl tento problém rozdělit na dvě kategorie. Chytré telefony a klasické telefony. U chytrých telefonů není získání těchto zdrojů problém, stačí napsat software, který tyto zdroje poskytne přes port USB a tím je problém vyřešen. Klasické telefony se dělí do dvou podkategorií. Telefony s dokumentací s API a poté telefon bez API a dokumentace. Některé starší telefony měli otevřené dokumentace a některá dokonce i API přímo pro tyto účely (Motorola, Nokia), pro tyto mobilní telefony existuje internet plný návodů, jak se dostat ke zdrojům které chceme, nicméně takových modelů je opravdu malé množství. Poslední kategorií jsou mobilní telefony bez otevřených dokumentací a bez API, která by zpřístupňovala chtěné zdroje mobilního telefonu. Pro tyto mobilní telefony jsem nenalezl žádné dostupné řešení, přitom jsou to telefony nejrozšířenější a tím se staly pro tento projekt zajímavými.

#### Varianty telefonů:

- Chytrý telefon (vyřešeno)
- Klasický telefon s API (vyřešeno)
- Klasický telefon bez API a dokumentace (nevyřešeno)

#### 4.1.2 Výběr telefonu

Prvním úkolem bylo vybrání vhodného kandidáta pro tyto účely. Bylo tedy nutné zvolit telefon, pro který zatím není problém vyřešen. Po provedení průzkumu trhu jsem zjistil, že je možné vybírat jak z bazarových kousků, tak z nových. Zajímavým zjištěním bylo, že ceny nových telefonů podporujících s funkcí GSM i GPRS startují

na 450 Kč s DPH za kus. Bazarové kousky začínají na 600 Kč s DPH za kus. Vzhledem k nižší ceně, záruce a garanci funkčnosti bylo rozhodnuto pro nový kus.

### 4.1.3 První rozbor

Před obdržením obdržení zakoupeného telefonu následoval průzkum, zda existují dokumentace, zda se někdo pokoušel alespoň o něco podobného a tak dále. Výsledkem bylo, že kromě oficiálního letáku s velice obecnými parametry přístroje, neexistují žádné veřejné dokumentace ani rozborů zařízení. Ihned po obdržení jsem se tedy pustil do vlastního průzkumu.



Obrázek 4.1: Zakoupený mobilní telefon

### 4.1.4 Kontaktování výrobce

Jedním z prvních kroků bylo kontaktování výrobce s vysvětlením mého projektu a žádostí o poskytnutí dokumentačních podkladů. Napsal jsem celkem na 3 oddělení britské společnosti, přesněji na obchodní, podporu a servis. Mezitím jsem se pokusil o komunikaci s mobilním telefonem skrze USB, stejně jako to bylo možné u telefonů s API k těmto účelům, to se nepodařilo. Nakonec přišla shodná odpověď ze všech tří kontaktovaných oddělení, tedy že mi dokumentace neposkytnou.

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](http://www.stklife.com)

Obrázek 4.2: Odpověď od společnosti STK

#### 4.1.5 Hlubší rozbor

Bez dokumentací jsem musel začít zkoumat více do hloubky. Po konzultaci s vedoucím byl telefon rozebrán a zkoumán na případné servisní piny pro připojení a diagnostiku. Po hlubším prozkoumání se mi jako jednou variantu podařilo nalézt servisní piny, které jsem se následně snažil analyzovat a skrze ně komunikovat s mobilním telefonem.

#### 4.1.6 Závěr rozboru

Ani po několika týdenním snažení se nepodařilo z pinů dostat žádnou informaci. První týden se mi podařilo zachytávat neznámé signály, nicméně po hlubším přezkoumání spektrálním analyzátozem bylo zjištěno, že se nejedná o číslicový signál. Po dvouměsíčním zkoumání jsem musel celou věc uzavřít a oznámit vedoucímu, že tudy zřejmě cesta nevede. Od využití mobilního telefonu bylo tedy upuštěno.

### 4.2 Zapojení

#### 4.2.1 Blokové schéma

#### 4.2.2 Výsledný prototyp

Výsledný prototyp je schopen simulovat reálné nasazení zabezpečovacího systému. Obsahuje (zleva) řídicí desku Arduino, jednu diodu (LED) pro simulování spínacího prvku, dále diodu která signalizuje, zda je sepnutý senzor PIR, a naposledy diodu signalizující sepnutí dveřního čidla (vpravo).

### 4.3 Firmware

Hlavním účelem softwaru je možnost kompletního řízení zabezpečovacího zařízení a komunikace, včetně poskytování API pro přístup aplikací. Software byl vyvinut pro platformu Arduino (čip ATmega) v programovacím jazyce C++, zapomocí IDE Visual Studio 2017 a kompilátoru Arduino IDE.

Dále budou zmíněny pouze ty části, které jsou důležité pro pochopení nastavení a přístupu k aplikaci (API).

### 4.3.1 Nastavení

Inicializační část slouží k prvotnímu nastavení mikrokontroleru před spuštěním. Toto nastavení nelze měnit pokud zařízení běží.

#### Nastavit lze:

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

### 4.3.2 API

Jednotlivé parametry jsou odděleny čárkou a všechny jsou zapoždřeny kulatými závorkami, tedy „(“ a „)“, ve výsledku potom „SetSensor(10,5)“.

Dotaz	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))
```

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 nový senzor, 0)
```

#### **Ukázka změny pinu senzoru:**

```
SetSensorName(1, Moje nové jméno senzoru)
```

#### **Ukázka smazání spínače:**

```
DeleteSwitch(1)
```

### **4.3.3 Notifikace**

Bezpečnostní zařízení má uložené všechny poslední stavy senzorů. Při každém průchodu hlavní smyčkou proběhne čtení, pomocí funkce „checkSensorStateChangedAndSendIfTrue()“ a pokud nový stav nesouhlasí se starým, je zaslána notifikace s ID zařízení a aktuálním stavem. Poté záleží na jedné z aplikací, jak informaci zpracuje.

#### **Ukázka notifikace:**

Sensor(Id = 1,State = 0)

#### **4.3.4 Chybová hlášení**

Původně byla chybová hlášení, při neúspěšném provedení operace nebo jiné chybě, tvořena pomocí stavových kódů HTML [20]. Z důvodu nedostatku paměti byla nahrazena jednoduchým „OK“ a „ERROR“. Kontrola chyb se provádí vždy při provádění kterékoliv z operací podporující API. Kontroluje se vždy 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. Poslední chybou hláškou je, že příkaz nebyl rozpoznán, ten je navrácen, pokud příkaz nesplňuje kritéria tohoto API.

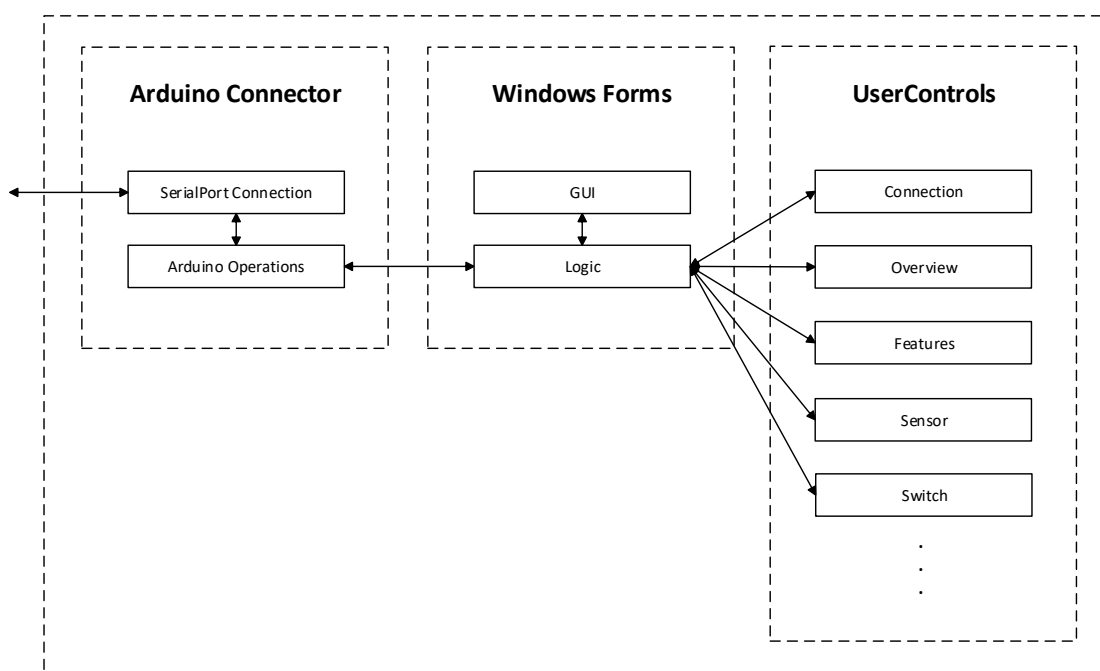
#### **Ukázka nerozpoznání příkazu:**

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

## 5 Desktopová aplikace

Hlavním účelem desktopové aplikace je možnost kompletního nastavení a monitorování zabezpečovacího zařízení. Software byl vyvinut pro operační systém Windows v programovacím jazyce C# (vyslovováno C-Sharp), zapomocí IDE Visual Studio 2017.

### 5.1 Blokové schéma

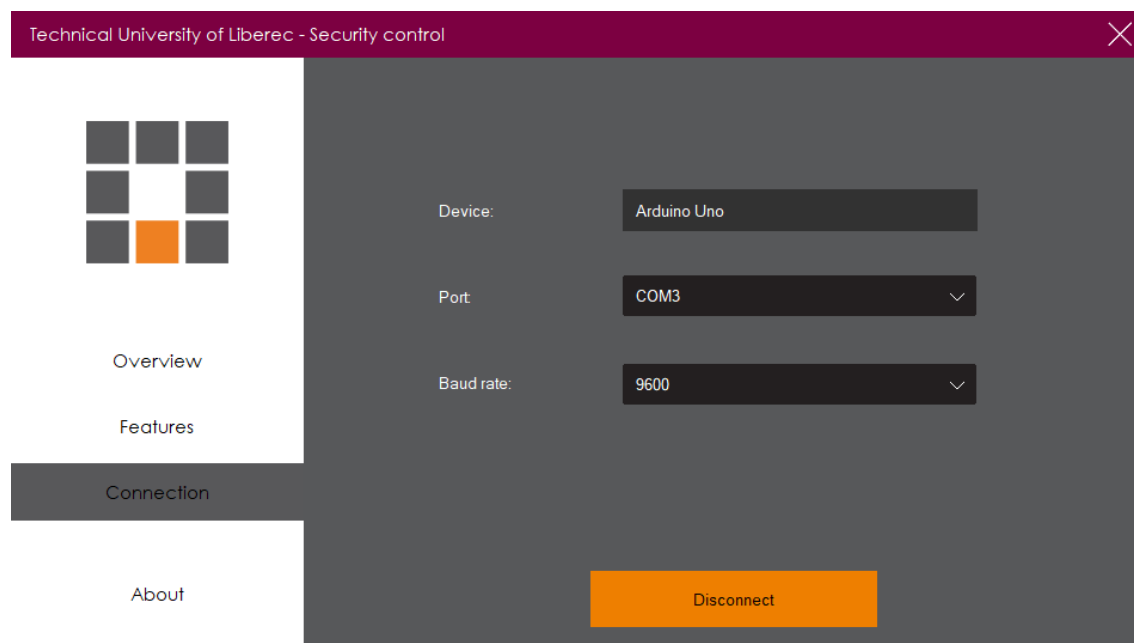


Obrázek 5.1: Blokové schéma objektového návrhu aplikace

Kód aplikace je rozdělen do více spolu bloků, které spolu vzájemně komunikují. Arduino Connector slouží k napojení zabezpečovacího zařízení a využívá API poskytované zabezpečovacím zařízením, díky čemuž poskytuje funkce pro jednoduché

zapisování a čtení do Arduina. Windows Forms poskytují grafické uživatelské rozhraní, základní logiku na propojení jednotlivých funkčních celků a notifikace pro uživatele. UserControls poskytují jednotlivé funkce aplikace, z nichž každá má své vlastní GUI, které se zobrazuje ve Windows Forms. Dále budou zmíněny pouze ty UserControly, které jsou důležité pro pochopení ovládní aplikace. Design byl inspirovaný manuálem jednotného vizuálního stylu TUL [19].

## 5.2 Spojení

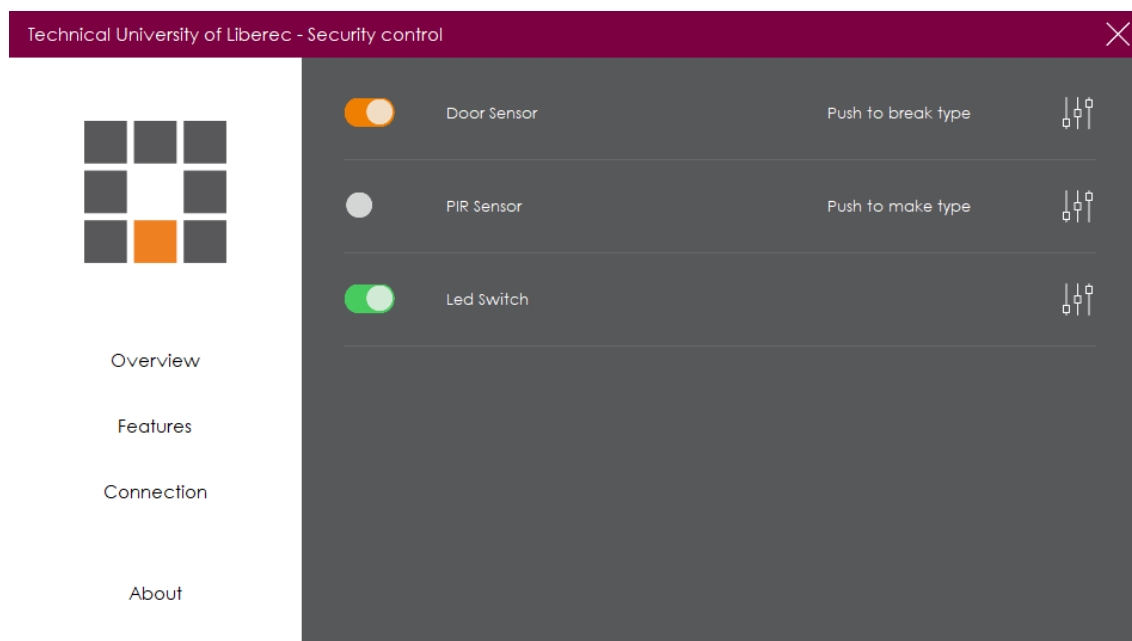


Obrázek 5.2: Okno s možností připojení/odpojení od řídicí jednotky

Spojení slouží k připojení či odpojení od zabezpečovacího zařízení. Za běhu aplikace na pozadí hlídá, zda je připojení stále navázáno. Při ztrátě spojení ihned informuje uživatele a upraví chování celé aplikace. Spojení rovněž hlídá, zda se neobjevilo nové zařízení k připojení, pokud ano, informuje uživatele a zobrazí nově dostupná zařízení.



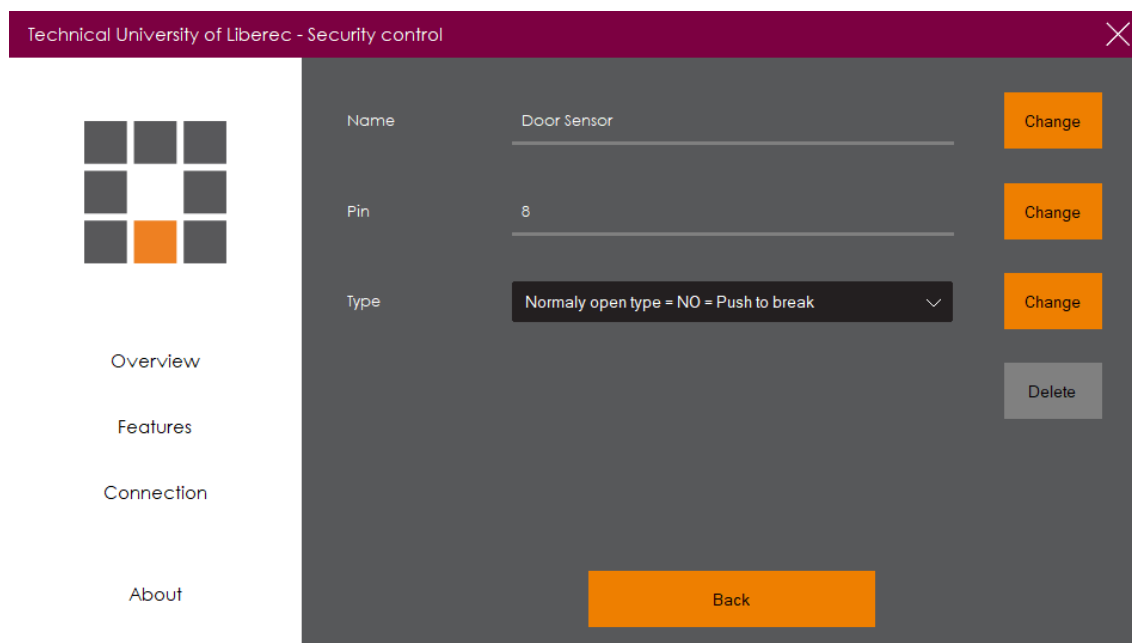
## 5.3 Přehled



Obrázek 5.3: Okno s přehledem jednotlivých komponent

Přehled slouží pro zobrazení všech dostupných spínačů a senzorů, které jsou v zabezpečovacím zařízení dostupné. Obsah se generuje dynamicky pod sebe, lze tedy vytvořit neomezený počet zobrazených zařízení. 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 měněna v objektech uvnitř aplikace, ale změny jsou posílány do zabezpečovacího zařízení, po obdržení odpovědi se změní stav v aplikaci a uživatel je upozorněn.

## 5.4 Nastavení senzoru



Technical University of Liberec - Security control

Overview

Features

Connection

About

Name: Door Sensor [Change]

Pin: 8 [Change]

Type: Normally open type = NO = Push to break [Change]

Delete

Back

Obrázek 5.4: Okno s nastavením komponenty (Senzor)

Nastavení senzoru slouží k změně interního nastavení uvnitř zabezpečovacího zařízení. Lze měnit název zařízení, pin na kterém se nachází a typ tohoto zařízení (rozpínací, spínací). Poslední funkcí je smazání zařízení.

## 5.5 Nastavení spínače

Nastavení je téměř identické s předchozím, lze měnit název, pin, a místo typu zařízení lze přenastavit jeho stav.

## 5.6 Funkce programu a nastavení

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

### Přehled funkcí:

- Přidat nový senzor
- Přidat nový spínač
- Znovu načíst všechna data ze zabezpečovacího zařízení

- Zobrazovat notifikace, pokud se změní stav senzoru (přepínač)
- Zobrazovat notifikace, pokud se senzor navrátí do výchozího stavu (přepínač)

## 5.7 Notifikace

Aplikace podporuje širokou škálu notifikací, upozornění a varovných hlášení, která slouží při upozornění uživatele na nějakou situaci. Například se jedná o změnu stavu pinů, chybové stavy řídicí jednotky, nebo odpojení zabezpečovacího zařízení.

### **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 řídicí 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

## 6 Mobilní aplikace

Hlavním účelem mobilní aplikace je monitorování zabezpečovacího zařízení, a není tedy možno měnit jakékoliv nastavení, k čemuž slouží desktopová aplikace. Aplikace byla vyvinuta pro mobilní operační systém Android 4.0.3 (Ice Cream Sandwich - API 15) a vyšší, v programovacím jazyce Java, zapomocí IDE Android Studio 2.3.2.

## 7 Závěr

## Literatura

- [1] 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.
- [2] Security alarm. Wikipedia [online]. [cit. 2017-05-23]. Dostupné z: [https://en.wikipedia.org/wiki/Security\\_alarm](https://en.wikipedia.org/wiki/Security_alarm)
- [3] Elektronické zabezpečovací systémy [online]. , 1 [cit. 2017-05-23]. Dostupné z: <http://www.ezasys.cz/elektronicke-zabezpecovaci-systemy/>
- [4] 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](https://cs.wikipedia.org/wiki/Elektronick%C3%A1_zabezpe%C4%8Dovac%C3%AD_signalizace)
- [5] Wiring [online]. [cit. 2016-05-08]. Dostupné z: <http://wiring.org.co/>
- [6] Visual Micro [online]. [cit. 2017-05-23]. Dostupné z: <http://www.visualmicro.com/>
- [7] Arduino introduction. Arduino [online]. [cit. 2017-05-23]. Dostupné z: <https://www.arduino.cc/en/Guide/Introduction>
- [8] Arduino libraries. Arduino [online]. [cit. 2016-05-08]. Dostupné z: <https://www.arduino.cc/en/Reference/Libraries>
- [9] Arduino programming language. Arduino [online]. [cit. 2016-05-08]. Dostupné z: <https://www.arduino.cc/en/Reference/HomePage>
- [10] ARDUINO LLC. Arduino Schematic [online]. 1 s. [cit. 2016-05-07]. Dostupné z: [https://www.arduino.cc/en/uploads/Main/Arduino\\_Uno\\_Rev3-schematic.pdf](https://www.arduino.cc/en/uploads/Main/Arduino_Uno_Rev3-schematic.pdf)
- [11] Arduino software. Arduino [online]. [cit. 2016-05-08]. Dostupné z: <https://www.arduino.cc/en/Main/Software>
- [12] Arduino source code. GitHub [online]. [cit. 2016-05-08]. Dostupné z: <https://github.com/arduino/Arduino/tree/1.6.8>

- [13] ATMEL CORPORATION. ATmega48A/PA/88A/PA/168A/PA/328/P Datasheet [online]. 2015, 660 s. [cit. 2016-05-07]. Dostupné z: [http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P\\_datasheet\\_Complete.pdf](http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf)
- [14] 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>
- [15] VODA, Zbyšek. Průvodce světem Arduina [online]. Vydání první. Bučovice: Martin Stríž, 2015 [cit. 2016-05-07]. ISBN 978-80-87106-90-7.
- [16] SIMCOM WIRELESS SOLUTIONS. 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](http://www.dfrobot.com/image/data/TEL0051/3.0/SIM908_AT%20Command%20Manua_V1.01.pdf)
- [17] SIMCOM WIRELESS SOLUTIONS. 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>
- [18] 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>
- [19] Manuál jednotného vizuálního stylu Technické univerzity v Liberci [online]. , 27 [cit. 2017-05-27]. Dostupné z: <http://www.ft.tul.cz/document/126>
- [20] 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>

## A Obsah přiloženého CD

Struktura a obsah adresářů je následující:

### **/Dokumentace**

Text bakalářské práce ve formátu pdf.

### **/Hardware**

Kompletní dokumentace s podklady k použitým sloučástkám, včetně fotografií.

### **/Software**

Vyvinutý software a zdrojové kódy, rozděleny do více kategorií.

### **/Software/Security (Firmware)**

Arduino aplikace, nahrané na desce zabezpečovacího zařízení.

### **/Software/SecurityControl (Nastavení a monitoring)**

Desktopová aplikace, sloužící k nastavení a monitorování zabezpečovacího zařízení.

### **/Software/SecurityViewer (Monitoring)**

Mobilní aplikace, sloužící k monitorování zabezpečovacího zařízení.