



## ZADÁNÍ BAKALÁ SKÉ PRÁCE

<b>Název:</b>	Inteligentní bezpečnostní systém garáže
<b>Student:</b>	Miroslav Vá a
<b>Vedoucí:</b>	Ing. Martin Da hel
<b>Studijní program:</b>	Informatika
<b>Studijní obor:</b>	Bezpečnost a informa ní technologie
<b>Katedra:</b>	Katedra počíta ových systém
<b>Platnost zadání:</b>	Do konce letního semestru 2018/19

### Pokyny pro vypracování

Seznamte se sou asnými pí ístupy, které eší problematiku elektronického zabezpečení objekt . Pro garáž navrhnete a implementujete zabezpečovací systém, dle těchto pokynů :

1. Dle analýzy zvolte vhodnou vývojovou desku na platformě Arduino, zvažte možnost budoucího rozšíření komunikace wifi/ethernet.
2. Analyzujte cenově dostupná díla a senzory vhodné pro monitorování garáže (vyberte max. 3 typy).
3. Navrhnete systém s ohledem na bezpečnost a odolnost vůči chybám pí m ení.
4. Navrhnete a implementujete vnit ní logiku systému, který se bude nacházet v n jakém stavu (nap . odem eno, zam eno, narušení) dle vyhodnocení nam ených hodnot.
5. Navrhnete komunika ní rozhraní, pomocí kterého se budou pí edávat informace o aktuálním stavu garáže.
6. Navržený systém otestujte.
7. Testováním analyzujte, jak rozeznat cílené útoky od nespolehlivosti celého systému, pí padn zvažte r zné typy útok , vytvořte profil úto níka (náhodný zlod j nebo n kdo, kdo se sofistikovan snaží pí ekonat zabezpečení).

### Seznam odborné literatury

Dodá vedoucí práce.

prof. Ing. Róbert Lórencz, CSc.  
vedoucí katedry

doc. RNDr. Ing. Marcel Ji ina, Ph.D.  
d kan

V Praze dne 17. íjna 2017



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
KATEDRA POČÍTAČOVÝCH SYSTÉMŮ



Bakalářská práce

## **Inteligentní bezpečnostní systém garáže**

*Miroslav Váňa*

Vedoucí práce: Ing. Martin Daňhel

7. května 2018





---

## Poděkování

Chci upřímně poděkovat všem, kteří mi věnovali čas a poskytli pomoc při psaní této bakalářské práce. Především děkuji svému vedoucímu Ing. Martinu Daňhelovi za odborné vedení, trpělivost a cenné rady, které mi pomohly práci zkompletovat. Dále chci srdečně poděkovat své rodině za podporu během celých mých studií.



---

# Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 7. května 2018

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2018 Miroslav Váňa. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Váňa, Miroslav. *Inteligentní bezpečnostní systém garáže*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018.

---

# Abstrakt

Cílem práce je přinést cenově dostupný, spolehlivý a bezpečný systém na fyzické zabezpečení a monitorování garáže. V návrhu systému je řešeno předávání informací o aktuálním stavu garáže na server. Bezpečnost a spolehlivost systému je zaměřena především na problémy s výpadkem senzorů, možnosti bezpečné komunikace se serverem a odolnost proti cíleným útokům. Vytvořeného cíle je dosaženo pomocí vývojové desky Yún na platformě Arduino s využitím různorodých senzorů a programovacího jazyka C++. Pro odhalení bezpečnostních rizik byl využit model STRIDE. Výsledkem se stal systém schopný odhalit vniknutí do objektu a odesílat výsledky ethernetovým rozhraním pomocí REST API.

**Klíčová slova** Inteligentní monitorovací systém, zabezpečení garáže, spolehlivost senzorů, Arduino Yún



---

# Abstract

The goal of this thesis is to provide an affordable, reliable and secure system for physical security and monitoring of a garage. In the design of the system it is solved the transfer of information about the current state of the garage to a server. Safety and reliability of the system are primarily focused on sensor failure, secure server communication, and resistance to targeted attacks. The set goal is achieved through the Yún development board on the Arduino platform using a variety of sensors and C++ programming language. The STRIDE model was used to detect security risks. As a result, the system is able to detect object intrusion and send out the results via the Ethernet interface using REST API.

**Keywords** Intelligent monitoring system, garage security, sensor reliability, Arduino Yún





---

# Obsah

Úvod	1
Cíle práce	3
<b>1 Úvod do problematiky</b>	<b>5</b>
1.1 Současné zabezpečovací systémy . . . . .	5
1.2 Platforma Arduino . . . . .	7
1.3 Spolehlivost . . . . .	8
<b>2 Analýza a výběr součástek</b>	<b>11</b>
2.1 Analýza vývojových desek . . . . .	11
2.2 Analýza senzorů . . . . .	15
2.3 Výběr čidel . . . . .	19
<b>3 Návrh systému</b>	<b>23</b>
3.1 Základní návrh . . . . .	23
3.2 Logika systému . . . . .	24
3.3 Spolehlivost systému . . . . .	25
3.4 Bezpečnost systému . . . . .	26
3.5 Komunikace . . . . .	28
3.6 Schéma zapojení . . . . .	30
3.7 Implementace . . . . .	31
3.8 Náklady . . . . .	34
<b>4 Testování</b>	<b>35</b>
4.1 Magnetické spínače . . . . .	35
4.2 Čidla kouře a plynu . . . . .	36
4.3 PIR . . . . .	37
4.4 Celý systém . . . . .	38
4.5 Problém s pamětí . . . . .	39

<b>5</b>	<b>Profil útočníka</b>	<b>41</b>
5.1	Matka příroda . . . . .	41
5.2	Náhodný zloděj . . . . .	41
5.3	Cílený útok . . . . .	42
5.4	Protiopatření . . . . .	42
	<b>Závěr</b>	<b>43</b>
	<b>Literatura</b>	<b>45</b>
A	Instalační příručka	49
B	Fotografie systému	51
C	Členění kódu	55
D	Seznam použitých zkratk	57
E	Obsah příloženého média	59

---

## Seznam obrázků

1.1	GSM alarm . . . . .	5
1.2	Modulární zabezpečovací systém . . . . .	6
1.3	Paralelní spolehlivostní model . . . . .	9
1.4	Sériový spolehlivostní model . . . . .	10
1.5	Spolehlivostní model systému . . . . .	10
2.1	Arduino Uno WiFi . . . . .	11
2.2	Arduino Tian . . . . .	12
2.3	Arduino Ethernet Rev3 . . . . .	12
2.4	Arduino Yún – schéma . . . . .	13
2.5	Arduino Yún – deska . . . . .	14
2.6	Infračervené záření . . . . .	16
2.7	Mikrovlnná čidla . . . . .	17
2.8	Magnetický spínač . . . . .	18
2.9	PIR Sparkfun SE-10 . . . . .	19
2.10	PIR HC-SR505 . . . . .	19
2.11	Me PIR . . . . .	20
2.12	PIR HC-SR501 . . . . .	20
2.13	Čidlo kouře a plynu MQ2 . . . . .	21
2.14	Magnetický spínač MC-38 . . . . .	21
2.15	Magnetický spínač AL28 . . . . .	22
2.16	Magnetický spínač SA-211 . . . . .	22
3.1	Základní návrh systému . . . . .	23
3.2	Stavový diagram . . . . .	24
3.3	Chyby při měření . . . . .	25
3.4	Schéma zapojení . . . . .	30
3.5	Načítání dat . . . . .	31
3.6	Obsluha stavu zamčeno . . . . .	32
3.7	Ukázka komunikace . . . . .	33

3.8	Implementace REST API . . . . .	33
4.1	Test čidla a plynu – graf . . . . .	37
B.1	Fotografie systému 1 . . . . .	51
B.2	Fotografie systému 2 . . . . .	52
B.3	Fotografie systému 3 . . . . .	52
B.4	Fotografie systému 4 . . . . .	53
B.5	Fotografie systému 5 . . . . .	53
B.6	Fotografie systému 6 . . . . .	53

---

## Seznam tabulek

2.1	Porovnání čidel . . . . .	15
3.1	Komunikační protokol . . . . .	28
3.2	Náklady . . . . .	34
4.1	Test magnetických spínačů . . . . .	35
4.2	Test čidla kouře a plynu . . . . .	36
4.3	Test PIR a spínačů . . . . .	38



---

# Úvod

Bezpečnost je v dnešní době velmi používaným termínem a není tomu tak nadarmo. Každý z nás je neustále obklopen různými informačními technologiemi, které se snaží přinášet užitek ve všech odvětvích. Odvrácenou stranou mince je však častá náchylnost ke zneužití díky nedokonalostem způsobených nesprávným a neodborným přístupem k těmto technologiím. Mou motivací pro volbu oboru a této práce je především fakt, že na špatně navržené systémy může doplatit téměř kdokoli využívající aktuální technologie. Ať už se jedná o slabiny v chytrých telefonech, špatně zabezpečená auta, nebo napadnutelné servery. Závažnost těchto slabin je často obrovská a není náhodou, že některé z nich obletí v médiích celý svět.

Touto prací chci obohatit materiály k zabezpečení objektů a demonstrovat na jednoduchém příkladu, že systém může být funkční i bezpečný. Celkově chci tímto tématem přispět k tvorbě kvalitnějších IT řešení. To všechno jsou důvody, díky kterým jsem si vybral návrh inteligentního bezpečnostního systému garáže, ve kterém se zaměřím nejen na sestavení funkčního produktu, ale také na otázky spojené s jeho spolehlivostí a bezpečností.

Rozsah zadání je velmi široký a již v úvodu bych chtěl zmínit, že zde nebude prostor pro detailní rozbor všech tematických okruhů. Výstupem práce tedy nebude pokrytí celé bezpečnostní problematiky, nebo vytvoření nejspolehlivějšího systému. Primárním záměrem je věnovat vytyčeným tématům dostatečnou pozornost a vyzdvihnout důležité oblasti.

Přínosem této práce má být zejména vytvoření cenově dostupného a snadno rozšiřitelného systému. Ten může v konečném důsledku působit jako konkurence pro zavedené produkty zabývající se ochranou objektů. Hlavní předpokládaná výhoda mého systému má být právě v bezpečnosti a nízké ceně.

*„One of the main cyber-risks is to think they don't exist. The other is to try to treat all potential risks.“ — Stephane Nappo*





---

## Cíle práce

Mým primárním cílem při tvorbě této práce je navrhnout, otestovat a odladit systém pro monitorování garáže, který bude v co největší míře eliminovat hrozby a zároveň bude poskytovat přijatelné použití pro uživatele. K dosažení tohoto cíle je potřeba splnit mnoho menších úkonů.

Startovním krokem je vypracování teoretické části práce, kde se budu zabývat přehledem současných znalostí a poznatků k danému tématu. Vzhledem k povaze práce se zaměřím především na již existující řešení systémů. Dále uvedu základní informace o platformě Arduino. Následně provedu jednoduchou analýzu spolehlivosti uvažovaného systému.

V navazující části provedu analýzu. Nejprve si budu muset zjistit dostatek informací o vývojových deskách a dostupných senzorech. Na základě těchto informací vyberu vhodnou vývojovou desku na platformě Arduino. Následně díky analýze zvolím tři vhodné a cenově dostupné senzory.

Mezi požadavky na systém patří možnost komunikace na bázi protokolů WiFi nebo Ethernet. Vytvořený monitorovací systém bude posílat informace o aktuálním stavu garáže na centrálu pro možnost dalšího zpracování informací. Bude tedy nutné navrhnout jednoduchý protokol, pomocí kterého se budou informace odesílat. Cílem této práce je také vypracování bezpečnostních opatření, která předchází běžným hrozbám. V rámci zajištění spolehlivosti systému bude nutné si poradit například s výpadkem senzoru a zároveň poskytnout uživateli informace o těchto událostech.

Kvůli širokému rozsahu práce bude systém v mnoha ohledech co nejjednodušší a nepředpokládám, že dojde k realizaci příliš komplexní logiky, nebo návrhu složitého schématu komunikace. Z těchto důvodů není součástí práce tvorba přístupového systému, ani informování uživatele pomocí SMS nebo e-mailu. Přístupový systém bude jednoduše simulován tlačítkem a ponechán jako varianta pro budoucí rozšíření.



# Úvod do problematiky

Prvním krokem k tvorbě zabezpečovacího systému je zhodnocení současného stavu podobných produktů. Poté následují informace o platformě Arduino. Úvod do problematiky uzavírá podkapitola se stručným přehledem pojmů z oblasti spolehlivosti. Ta je jedním ze základních požadavků na systém.

## 1.1 Současné zabezpečovací systémy

V této podkapitole jsou přiblížena aktuální komerční řešení a následně je uveden směr, kterým se bude praktická část této práce ubírat. Jako ukázka slouží produkty českého dodavatele zabezpečovacích systémů JABLOTRON ALARMS<sup>1</sup>.

### 1.1.1 GSM alarmy

Ve většině případů se jedná o řešení „vše-v-jednom“ kombinující více ochranných prvků s možností odeslat upozornění ve formě SMS, či e-mailu [1]. Často



Obrázek 1.1: GSM alarm EYE-02 s kamerou a pohybovým detektorem [1].

<sup>1</sup>Největší společnost ze skupiny JABLOTRON GROUP působící v Jablonci nad Nisou od roku 1990. Zaměřuje se na systémy a služby pro ochranu majetku a zdraví.

## 1. ÚVOD DO PROBLEMATIKY

---

se kombinuje spojení kamery a pohybového detektoru. Velkou výhodou tohoto řešení je snadná montáž a instalace při nízké ceně (řádově tisíce Kč).

### 1.1.2 Modulární systémy

Modulární systémy poskytují komplexní řešení a možnost zabezpečení celého domu, či jednotlivých místností jako garáže. Lze tak nainstalovat mnoho samostatných detektorů napojených do ovládacího modulu [2]. Tím je zajištěna vysoká modularita, rozšiřovatelnost a snadná výměna jednotlivých detektorů. Pro ilustraci je uveden obrázek 1.2 zahrnující ovládací modul a posléze mnoho připojitelných detektorů. Mezi ně patří například detektor kouře, pohybu, otevření, nebo také siréna a modul klávesnice.



Obrázek 1.2: Modulární zabezpečovací systém Jablotron v rodinném domě [2]. Všechny žluté komponenty jsou připojeny do jednoho ovládacího modulu.

### 1.1.3 Zhodnocení

Záměrem této práce je zhotovit systém, který bude dostatečně přizpůsobitelný, ale zároveň s ním nebude problém pokrýt blok garáží. Celý tento blok může být hromadně monitorován pomocí serveru, na který se data odešlou. To otevírá spoustu možností pro další rozšíření práce směrem k IoT (Internet of Things). Například sledovat stav garáže online, odemknout dveře pomocí aplikace nebo odesílat data z případné kamery. Stejně tak se nabízí mnoho možností jak systém rozšířit dalšími senzory, moduly nebo přístupovým systémem. Pro zachování rozumného rozsahu je práce vymezena pouze na vytvoření základního systému. To v podobě zabezpečení jedné garáže a navázání komunikace se serverem.

## 1.2 Platforma Arduino

Zadání práce definuje zhotovení systému právě na platformě Arduino. Ta nabízí široké možnosti využití při snaze zachovat uživatelsky nenáročné prostředí. Je tedy vhodná pro tvorbu různorodých projektů od domácího teploměru, přes quadrokoptéru po webový server. Hlavním zdrojem informací pro tuto podkapitolu je kniha Exploring Arduino [3].

### Otevřená platforma

Otevřená platforma, neboli open-source je značným důvodem popularity Arduina. Veškeré podklady jako specifikace, návrhy desek, schémata a zdrojový kód jsou dostupné z oficiálních zdrojů. Díky tomu mnoho výrobců využívá Arduino jako základ pro návrh vlastního hardwaru. Jsou zde také výrobci, kteří pouze přeprodávají klony Arduino zařízení a tlačí tak cenu velmi nízko.

### Atmel mikrokontrolér

MCU (mikrokontrolér) je srdcem všech doposud představených vývojových desek. MCU má v paměti zkompileovaný kód, kterým se mohou ovládat veškeré periferie. Většinou tedy analogově digitální převodníky (ADCs), vstupní a výstupní (I/O) piny a komunikační sběrnice včetně I<sup>2</sup>C a SPI. Nedílnou součástí desek je kromě MCU také 16, či 8 MHz keramický oscilátor.

### Napájení

Samotné vývojové desky jsou ve většině případů napájené 5V z USB. Další možností je přivést 6–20V ze stejnosměrného (DC) zdroje na pin s názvem  $V_{in}$ . To umožňuje snadné napájení z běžně dostupných 9V alkalických baterií. Díky zabudovaným regulátorům jsou pro součástky a periferie k dispozici 5V a 3,3V napájecí piny. Maximální hodnota stejnosměrného proudu na I/O pinech je 40 mA [4]. Běžně doporučovaná hranice pro  $V_{cc}$  pin je 200 mA. Případně se může lišit dle použitého zdroje napájení [5].

### Bootloader

Je firmware uvnitř Atmel MCU, který umožňuje nahrát do Arduina nové programy bez nutnosti použít externí hardwarový programátor [6]. Předtím, než jsou programy zavedeny do MCU proběhne několik kroků [7]. Nejprve předzpracování do C++ kódu, který je následně předán kompilátoru (avr-gcc). Kompilátor převede zdrojový kód do strojového, ten je následně slinkován s knihovnamí. Výsledkem je jeden hexa soubor, který se (většinou) pomocí USB a bootloaderu zavede do MCU. Bootloader nicméně zabírá často cenné programové místo (flash paměť) a motivací k použití externího programátoru může být právě zisk přibližně 4 KB paměti navíc.

### 1.3 Spolehlivost

Tato podkapitola obsahuje základní definice použitého názvosloví, které jsou převážně převzaté z odborné literatury zabývající se spolehlivostí elektronických systémů [8]. Dále je zde úvod do spolehlivostních modelů, na základě kterého bude navržen blokový model systému.

#### 1.3.1 Názvosloví a definice

**Defekt** – je nedovolená odchylka od požadovaného technického nebo fyzikálního stavu výrobku takové povahy, že buď způsobí poruchu nebo vede k nepřijatelnému zvýšení rizika vzniku poruchy.

**Porucha** – jev, při kterém dojde u komponenty (čidlo, popř. jiný podpůrný HW) k ukončení schopnosti plnit požadovanou funkci podle technických předpisů definovaných výrobcem komponenty. Jinými slovy se jedná o model defektu na logické úrovni.

**Spolehlivost** – je komplexní vlastnost, která může zahrnovat různé ukazatele. Např. bezporuchovost, životnost, udržovatelnost, dostupnost a to buď jednotlivě nebo v kombinaci. V této práci je klíčovým ukazatelem **bezporuchovost**. Neboli pravděpodobnost bezporuchového provozu v daném časovém intervalu.

**Redundance** – jedná se o širší pojem zálohování komponent (čidel) systému, či dokonce celého systému za účelem zvýšení celkové spolehlivosti. Pro tuto práci je **klíčové použití redundance čidel, kde se na základě jejich snížené kvality předpokládá možnost defektů nebo poruch**.

**Selhání systému** – za selhání systému se považuje stav, kdy systém nedokáže dále vykonávat požadovanou funkci. Podle rozsahu funkčního omezení lze rozlišit kritické a nekritické selhání.

- Kritické selhání systému – selhání řídicí desky (Arduina).
- Nekritické selhání systému – porucha jedné z neredundantních komponent (např. bzučáku).
- Neselhání systému – porucha jednoho z redundantních čidel (např. magnetického spínače).

**Chyba** – rozdíl mezi správnou a skutečnou hodnotou nějaké veličiny, zjištěný měřením nebo pozorováním. Chyba je většinou důsledek poruchy.

**Systém odolný proti poruchám** – je systém, který zůstane ve funkčním stavu i v případě, že jedna nebo více jeho částí selže v důsledku poruchy.

„Odolnost vůči poruchám byla donedávna výsadní vlastností systémů používaných v několika privilegovaných oborech, např. v kosmonautice, letectví nebo ve vojenské technice. S poklesem ceny, rozměrů a energetické náročnosti elektronických systémů však odolnost proti poruchám postupně proniká do řady dalších aplikačních oblastí, takže již zdaleka není ničím výjimečným.“ [8]

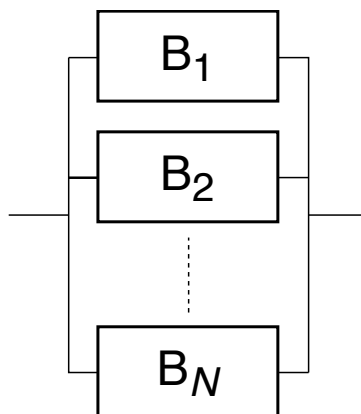
Cílem této práce není určit celkovou intenzitu poruch ani jinak vyčíslvat spolehlivost. Cílem je pouze navrhnout nejvhodnější použitelný spolehlivostní model a na základě něj vytvořit funkční prototyp zařízení.

### 1.3.2 Spolehlivostní modely

Vyvíjené zařízení lze nejlépe popsat spolehlivostními modely s nezávislými prvky [8]. Jedná se o blokové modely, jejichž bloky (prvky systému) mohou představovat určitý funkční celek systému. Celý systém je většinou možno logicky rozdělit podle toho, jak jsou bloky zapojeny. Lze rozlišit až tři typy těchto modelů – paralelní, sériový a kombinovaný.

#### Paralelní

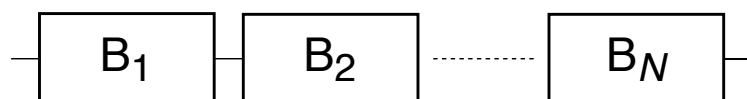
V tomto případě je systém tvořen redundantními (ne nutně stejnými) nezávislými bloky.



Obrázek 1.3: Příklad obecného paralelního modelu s  $N$  prvky.

#### Sériový

Systém je tvořen sérií navzájem nezávislých bloků. Z hlediska spolehlivosti je to v zásadě každý systém, který sám o sobě není redundantní a skládá se z několika částí. V případě poruchy jakékoli části celý systém selže.

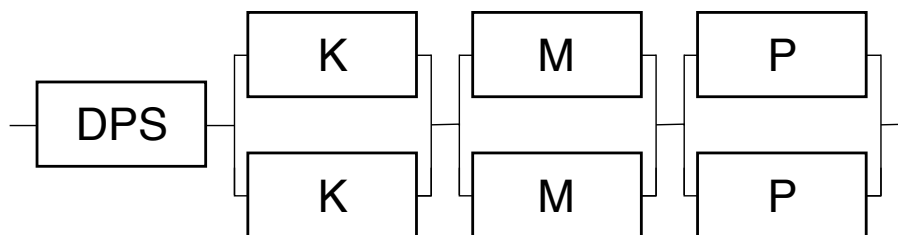
Obrázek 1.4: Příklad obecného sériového modelu s  $N$  prvky.

### Kombinovaný

Ve složitějších případech může být spolehlivostní model systému vytvořen kombinací sériového a paralelního spojení prvků. Příkladem kombinovaného spolehlivostního modelu je tento systém. Jeho spolehlivostní model je zachycen na obrázku 1.5.

#### 1.3.3 Zhodnocení

Záměrem tohoto systému je využít redundance čidel pomocí paralelního modelu a zajistit tak co nejlepší odolnost systému proti poruchám. Horní limit pro počet redundantních čidel udává počet pinů na vývojových deskách. S ohledem na možnost budoucího rozšiřování systému se předpokládá použití dvou čidel z každé skupiny (redundance). Na následujícím obrázku 1.5 je znázorněn kombinovaný model systému.



Obrázek 1.5: Návrh kombinovaného **spolehlivostního modelu pro tento systém**. Zkratky K, M a P označují čidla kouře a plynu, magnetické spínače a pohybové senzory. Tedy čidla, která byla zvolena v analýze senzorů – 2.2.

Tímto přístupem se vytvářený systém odlišuje od běžně dostupných komerčních řešení (viz 1.1). Ta často využívají několik druhů čidel v sériovém modelu, ale nepoužívají redundanci ve smyslu konkrétních komponent. U těchto systémů může dojít ke kritickému selhání při poruše kteréhokoliv čidla nebo komponenty.



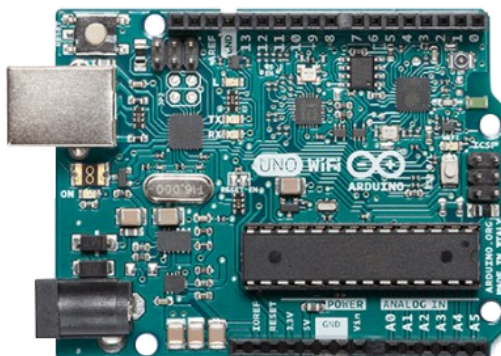
## Analýza a výběr součástek

### 2.1 Analýza vývojových desek

Základním kamenem celého systému je právě vývojová deska. Ta spojuje všechny použité součástky, vyhodnocuje stav garáže a komunikuje s okolím. V následujících podkapitolách jsou popsány hlavní výhody a nevýhody Arduino desek použitelných pro účely této práce. Nutnou podmínkou pro uvedení desky v této analýze je přítomnost alespoň jedné z technologií pro komunikaci se serverem (Ethernet nebo WiFi). Mezi další primárně sledované atributy patří cena, dostupnost, počet pinů a také velikost paměti.

#### 2.1.1 Arduino Uno WiFi

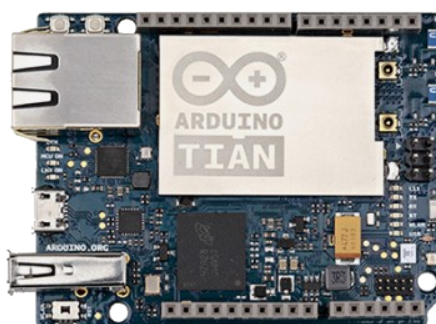
Populární Arduino Uno rozšířené o WiFi technologii. Obsahuje MCU ATmega328P a modul ESP8266 zajišťující WiFi komunikaci. Uno WiFi je osazeno 14 digitálními a 6 analogovými piny. Velikost flash paměti je 32 KB [9]. Cena u českých dodavatelů se pohybuje kolem 1 000,- Kč. Nízká cena se tak stává hlavní výhodou Uno WiFi.



Obrázek 2.1: Arduino Uno WiFi [9]

### 2.1.2 Arduino Tian

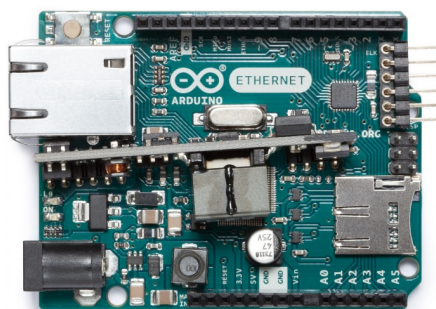
Aktuálně (září 2017) nejdražší vývojová deska z oficiální nabídky Arduina [10]. Její cena se na našem trhu pohybuje kolem 3 000,- Kč. Tian spojuje klasický mikrokontrolér na architektuře ARM Cortex a mikroprocesor Atheros využívající MIPS architekturu, na které lze spustit Linuxové distribuce. Na desce se nachází 20 digitálních I/O pinů. Mikrokontrolér disponuje velikostí flash paměti 256 KB a v mikroprocesoru se nachází dokonce 4 GB paměti typu eMMC [11]. Výhodou je nativní podpora WiFi, Ethernetu a vyššího výkonu mikročipů. Nevýhodou je horší dostupnost na českém trhu a vysoká cena.



Obrázek 2.2: Arduino Tian [11]

### 2.1.3 Arduino Ethernet Rev3

Srdcem Arduina Ethernet je MCU ATmega328 se stejným množstvím analogových a digitálních pinů jako u Uno WiFi. Tedy 14 digitálních a 6 analogových [12]. Jak název napovídá, deska podporuje technologii Ethernet, WiFi však nikoliv. Cena začíná na částce 1 400,- Kč a modul je k dostání i s PoE<sup>2</sup> rozšířením. V ČR je však tato deska špatně dostupná.



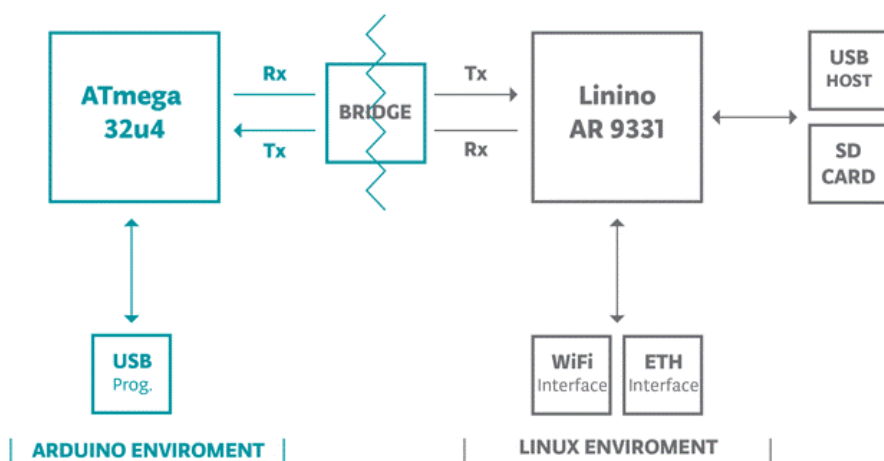
Obrázek 2.3: Arduino Ethernet Rev3 s PoE [13]

---

<sup>2</sup>Power over Ethernet – napájení přes ethernetový kabel.

### 2.1.4 Arduino Yún

Arduino Yún je parametry velmi podobné desce Tian, ale narušil od ní je levnější a také dostupnější. Cena se pohybuje okolo 1 800,- Kč. Primární využití má tato deska nacházet především v oblasti IoT projektů. Po vzoru Arduina Tian také využívá výhody dvou mikročipů. Je osazena slabší verzí mikroprocesoru Atheros, který je distribuován s Lilino OS, nebo OpenWrt Linux<sup>3</sup>. Jako mikrokontrolér byl použit ATmega32U4 poskytující flash paměť o velikosti 32 KB a dostatek I/O pinů. Konkrétně 20 digitálních a 12 analogových [14]. S okolím je možné komunikovat pomocí WiFi a Ethernet připojení. Vývojová deska je k dispozici také s rozšířením v podobě PoE modulu. Výhodou Yún je přívětivá cena při zachování dobrých parametrů.



Obrázek 2.4: Schéma znázorňující komponenty a návrh Arduina Yún. Komunikaci mezi čipy ATmega32u4 a AR9331 zajišťuje knihovna Bridge. Zdroj: [www.arduino.cc/en/Guide/ArduinoYunLin](http://www.arduino.cc/en/Guide/ArduinoYunLin).

### 2.1.5 Arduino Primo

Deska osazená třemi MCU s podporou mnoha technologií. První MCU má integrované BLE (úsporné bluetooth) a NFC. Druhý mikrokontrolér se stará o komunikaci mezi MCU. Poslední mikrokontrolér zajišťuje WiFi. Zajímavou funkcí desky je hardwarová podpora AES instrukcí. Počtem pinů je na tom stejně jako Arduino Uno WiFi, tedy 14 digitálních a 6 analogových [15]. Uživatel tedy dostane do rukou spoustu možností za cenu kolem 1 800,- Kč. Zásadní nevýhodou je nedostupnost na našem trhu a nerozšířenost této desky.

<sup>3</sup>Záleží na konkrétní verzi desky.

### 2.1.6 Výběr vývojové desky

Tato podkapitola se zaměřuje na výběr ideální vývojové desky v poměru cena/výkon s ohledem na dostupnost. Jako nejhorší varianta se jeví Arduino Primo. To díky své špatné dostupnosti a poměrně vysoké ceně. Arduino Tian se stává druhou zamítnutou deskou, jejíž největší nevýhoda spočívá v ceně převyšující 3 000,- Kč. Tři tisíce totiž dělají předběžnou částku, do které by se měly vejít veškeré výdaje včetně senzorů a kabelů.

Další desky Arduino Ethernet a Uno WiFi zajišťují komunikaci pouze přes jednu technologii, což je jejich největší slabinou. Bez většího důrazu na finance se jeví jako ideální deska Arduino Yún, protože podporuje WiFi i Ethernet a disponuje dobrou výbavou. V případě, že chceme ušetřit peníze se stačí rozhodnout nad drátovou, nebo bezdrátovou komunikací, protože cenovka i výbava je u Arduina Ethernet i Uno WiFi velmi podobná. Nakonec se pro práci **vybrala deska Arduino Yún**, protože cenový skok mezi ostatními kandidáty (Uno WiFi a Ethernet) není tak propastný a Yún poskytuje **znatelně víc možností** a tedy větší volnost při tvorbě systému.



Obrázek 2.5: Vybraná vývojová deska – Arduino Yún [14].

## 2.2 Analýza senzorů

Klíčovým krokem pro vznik celého systému je volba správných typů čidel, ze kterých se v příští podkapitole 2.3 vyberou konkrétní modely. V nadcházející analýze je uveden přehled aktuálně dostupných typů čidel, které jsou alespoň částečně vhodné pro monitorování objektů (garáže). Právě výběr správné skupiny čidel do velké míry ovlivní, jak spolehlivý bude celý systém. Nejprve je uvedeno proč a jaké 3 typy čidel jsou v práci použity. V podkapitolách jsou následně tato vybraná čidla blíže popsána. Zdrojem informací pro tvorbu této tabulky byla především kniha o moderních senzorech [16]. Dále také zkušenosti autora z různými druhy senzorů z předmětu BI-ARD<sup>4</sup> a tvorby maturitní práce na téma Detektor barev [17].

Čidlo	Výhody	Nevýhody
Pohybu	Zajistí do velké míry hlavní aspekt bezpečnosti. Tedy narušení objektu osobou.	Náchylnost na chybu a správné umístění.
Zvuku	Jednoduché a levné. Může navýšit spolehlivost systému.	Snadno ovlivnitelné nežádoucím hlukem.
Světla	Podpůrné čidlo pro snížení chybovosti systému.	V objektu s oknem bude ve dne nízká účinnost.
Vzdálenosti	Schopnost zaměřit se na konkrétní věc, nebo část monitorovaného objektu.	Jedno čidlo nemusí pokrýt celý sledovaný objekt.
Kouře a plynu	Lze včas zareagovat na krizovou situaci.	Nemusí stačit pouze 1 čidlo pro celý objekt.
Optická závora	Možnost jednoduše sledovat pozici vybraného objektu.	Větší rozměry a cena kvalitnějších čidel.
Spínač (magnetický)	Velmi levné a jednoduché. Snadné hlídání oken a dveří.	Pracná instalace. Nepříliš univerzální.

Tabulka 2.1: Porovnání různých typů čidel pro monitorování garáže.

Dle tabulky se jako **nezbytné součásti systému jeví senzory<sup>5</sup> pohybu a kouře**. Oba by měly poměrně dobře a účinně zajistit ochranu proti specifické kategorii hrozeb. Poslední druh senzoru by měl podpořit spolehlivost a správný úsudek systému v nebezpečné situaci.

Čidla zvuku a světla sice mohou správný úsudek podpořit, ale snadno tomu může být i naopak. Z důvodu nejisté spolehlivosti se v této práci využívat nebudou. Čidla vzdálenosti jsou poměrně levná i přesná, ale jakákoliv překážka nebo přemístění senzoru vede k chybám. Pro praktické nasazení tedy nejsou vhodná díky předpokládaným komplikacím a špatné univerzálnosti. Poslední

<sup>4</sup>Předmět Interaktivní aplikace s Arduinem využívající vývojovou desku Esplora.

<sup>5</sup>Senzor, neboli čidlo.

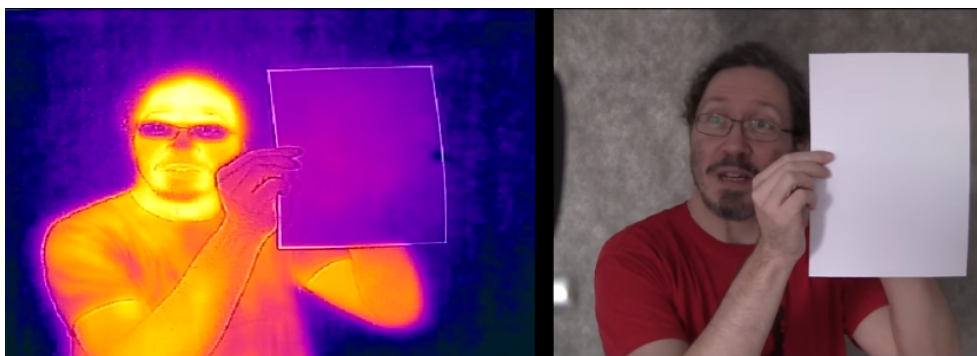
dva druhy čidel jsou optická závora a spínač. Obě čidla slouží zejména na monitorování oken a dveří. Optické závory fungující na delší vzdálenost (nad 50 cm) jsou dost drahé a nepraktických rozměrů. Naproti tomu spínače lze díky malým rozměrům umístit téměř kamkoliv a jejich cena je velmi nízká. Proto je jako **poslední z kategorie čidel vybrán právě magnetický spínač**.

### 2.2.1 Čidla pohybu

Čidlo pohybu je stěžejní bod celého systému zajišťující prostorovou ochranu. K dispozici je mnoho technologií a stejně tak i velké množství jednotlivých senzorů. Prostorová čidla se dělí na pasivní a aktivní. **Pasivní** ve svém střeženém prostoru zaznamenávají fyzikální změny způsobené útočníkem. **Aktivní** naopak „emitují do okolí jistou formu energie (například elektromagnetické pole) a detekují změny v prostředí způsobené touto emisí“ [18]. Následující podkapitoly popisují nejpoužívanější technologie a jejich vlastnosti.

#### PIR čidla

PIR neboli Passive InfraRed jsou čidla, která poměrně spolehlivě odhalí lidský pohyb [16]. Čidla „jsou založena na principu zachycení změn vyzařování v infračerveném pásmu kmitočtového spektra elektromagnetického vlnění. K detekci je použit materiál vykazující pyroelektrický jev. Detekční prvek je měnič, který je schopen detekovat pouze změny záření dopadající na detektor. Obraz střeženého prostoru v infračerveném pásmu je prostřednictvím optiky transformován na plochu senzoru“ [19]. To vše si lze v jednoduchosti představit jako infrakameru. Důležitou podmínkou je správné nasměrování a umístění čidla. Pokud bude senzor otočen na místo, kde dochází k častým nebo prudkým změnám teplot, nemusí fungovat dobře. Není ho tedy vhodné umístit blízko topení a oken. Problémem stejně tak mohou být i prudké vlivy ventilace, nebo vyzařování světla (slunce, reflektory).



Obrázek 2.6: **Infračervené záření**, zdroj: <https://youtu.be/fpx7hsoYEt4>.

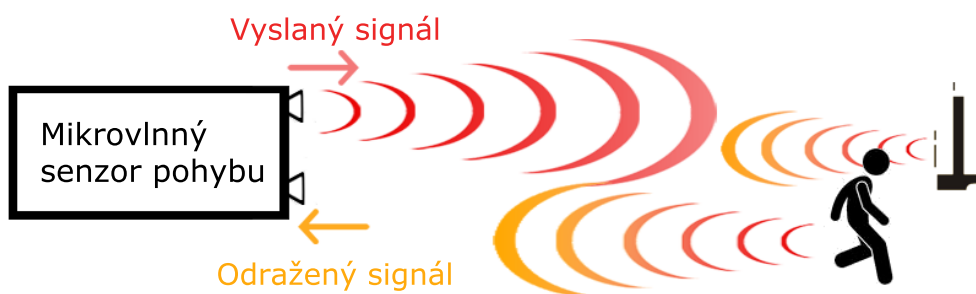


### Ultrazvuková čidla

Využívají část spektra nad pásmem kmitočtů slyšitelných lidským uchem. Jelikož do prostoru vysílají energii, řadí se mezi aktivní čidla pohybu. Základním prvkem je vysílač, který vysílá vlnění o konstantním kmitočtu. Odražené vlnění od překážek v uzavřeném prostoru je snímáno přijímačem. Není-li prostor narušen, je vlna přijata ve stejné fázi jako vlna vyslaná. Pokud se v hlídaném prostoru pohybuje těleso, dojde ke změně fáze přijatého vlnění a tím k poplachu. Princip spočívá v aplikaci Dopplerova jevu [20]. Nevýhodou je množství předmětů vydávající zvuk v oblasti pracovního kmitočtu čidla a nutnost synchronizace při instalaci více čidel najednou.

### Mikrovlonná čidla

Fungují na stejném fyzikálním principu jako ultrazvuková čidla, pouze ke své činnosti využívají elektromagnetického vlnění. Čidlo měří, jak dlouho trvá, než se vyslané vlnění vrátí zpátky. Po umístění a zapnutí se senzor uvede do „normálního“ stavu. Tento stav reprezentuje dobu odezvy v objektu, kde nedochází k žádnému pohybu. Při vstupu člověka nebo jiného objektu do snímaného prostoru dojde ke změně zpoždění signálu a tím k detekci pohybu. Mikrovlonné čidlo je vhodné i pro pokrytí většího prostoru, ale není schopno rozlišit, zda jde o pohyb člověka, zvířete, nebo věci [16]. Dále není vhodné používat tato čidla v blízkosti velkých kovových objektů [19].



Obrázek 2.7: Vysílaný signál **mikrovlonného čidla** je modifikován pohybem osoby nebo objektu. Tím dochází k rozpoznání pohybu osob i věcí.

### Shrnutí čidel pohybu

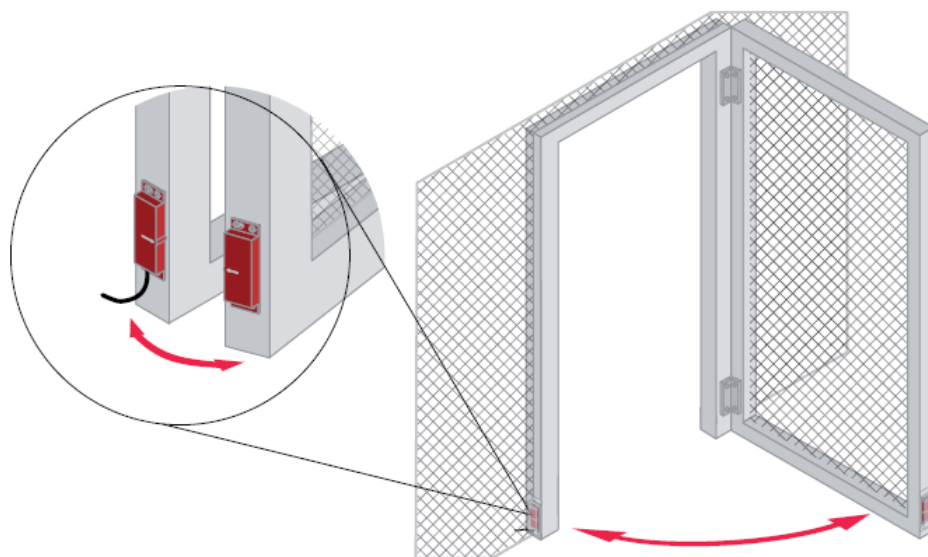
Dle zjištěných informací se mikrovlonná a ultrazvuková čidla využívají především na pokrytí větších prostor, ve kterých by muselo být nasazeno více PIR čidel. Oproti PIR senzorům může být mikrovlonné a ultrazvukový senzor ovlivněn pouhým pádem předmětu v monitorovaném objektu. Z těchto důvodů se jako výhodnější čidlo pro detekci osob v garáži jeví PIR.

### 2.2.2 Čidla kouře a plynu

Kategorie čidel kouře a plynu není tak rozmanitá jako je tomu u čidel pohybu. S Arduinem se používají především čidla ze série MQ. Ty mají uvnitř elektrochemický snímač, který je zpravidla citlivý na řadu plynů a kouř [21]. Čidla mohou být často kalibrována pomocí trimru. Kromě detekce kouře a plynu se některá čidla zaměřují na snímání kvality vzduchu. U MQ čidel se předpokládá použití v místnostech s pokojovou teplotou a proto jsou ideálním kandidátem na instalaci v prostoru garáže.

### 2.2.3 Magnetické spínače

Kategorii vybraných čidel uzavírají spínače, které se ukázaly jako ideální podpůrný prvek pro monitorování dveří, či oken. Používají se jako spolehlivá kontrola správného zavření. Samotný magnetický snímač se skládá ze senzoru (čtecí hlavy) a aktuátoru (permanentní magnet). Ve čtecí hlavě je většinou umístěno několik jazýčkových kontaktů pracujících na principu Reedova spínače [22]. Čtecí hlava reaguje na magnet v rozsahu zhruba jednoho cm. Vyrábí se spousta variant. Od levných po průmyslové spínače splňující přísné normy. V souvislosti s magnetickými spínači se také často uvádí pojem plášťová ochrana. Prvky plášťové ochrany se používají pro detekci otevření nebo zničení prostupů pláště budovy (oken, vrat a dveří).



Obrázek 2.8: Magnetický spínač přidělaný na rámu dveří [22].



## 2.3 Výběr čidel

Z již stanovených kategorií z předchozí podkapitoly Analýza senzorů budou zvoleny co nejvhodnější konkrétní modely čidel.

### 2.3.1 PIR čidla

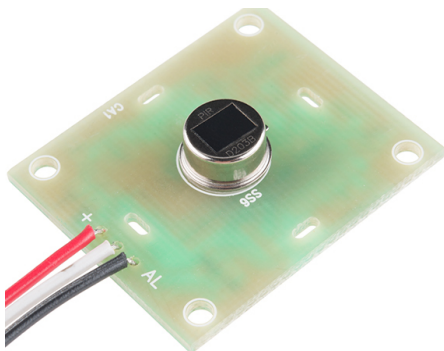
PIR čidla mají poměrně hojné zastoupení. Následující přehled popisuje pět běžně dostupných PIR čidel s možností 5V napájení. V první řadě jsou sledovány údaje jako dosah snímání, cena a délka sepnutí.

#### HC-SR501

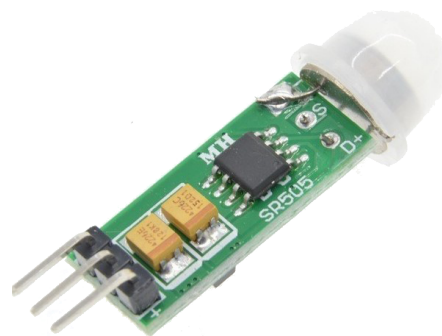
Cenově dostupné čidlo, jehož hlavní výhodou je velmi nízká cena začínající v zahraničí od 22,- Kč za kus. Dle výrobce senzor snímá přibližně pod úhlem  $120^\circ$  až do vzdálenosti 7 metrů [23]. Pomocí trimru lze nastavit délku sepnutí při detekci a zároveň i citlivost snímání. Délku sepnutí je možné nastavit od 3 sekund po 5 minut. Citlivost snímání upravuje maximální vzdálenost, na kterou je možné rozpoznat pohyb. U tohoto čidla lze nastavit 3–7 metrů. Čidlo je napájeno z 5–20V zdroje a má klidový odběr proudu do 50  $\mu\text{A}$ . Existuje nespočet různých obdob, ale HC-SR501 je jedno z nejpoužívanějších PIR čidel pro platformu Arduino [24]. Nevýhodou je 60 sekundová inicializace čidla, v průběhu které nelze získat věrohodná data.

#### Sparkfun SE-10

Stejně jako v předchozím případě se jedná o jednoduché čidlo pohybu. Oproti HC-SR501 je čidlo dražší a na českém trhu je k dostání od 290,- Kč. Výhodou je však velmi rychlá inicializace v čase 3 sekund. Dalším důležitým parametrem je vzdálenost, na kterou čidlo rozpozná pohyb. Zde jsou to 3 metry, bez možnosti tento dosah upravit. Délka sepnutí čidla není regulovatelná a z dokumentů výrobce není jasné na jak dlouho k sepnutí dojde [25].



Obrázek 2.9: Sparkfun SE-10 [25]



Obrázek 2.10: HC-SR505 [26]

### HC-SR505

HC-SR505 je menší kolega předešlého čidla HC-SR501. V porovnání s ním disponuje menším dosahem i úhlem detekce [26]. Důraz je u této součástky kladen především na velikost, ale pokulhávací tu jiné vlastnosti. Není zde například možné upravovat detekční vzdálenost nebo délku sepnutí. Cena je v řádu desítek Kč. Senzor si lze prohlédnout na obrázku 2.10.

### Me PIR

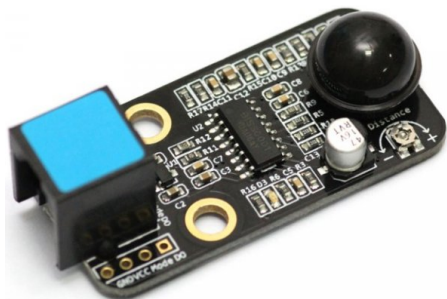
Jedná se o modul, který lze připojit pomocí RJ11 [27]. Vlastnosti jsou podobné jako u prvního zmiňovaného senzoru (HC-SR501), ale cena se pohybuje kolem 200,- Kč. Hlavní výhodou oproti levným PIR jsou dva předpřipravené otvory na montáž. Společně s konektorem RJ11 lze čidlo snadno upevnit a připojit na delší vzdálenost. Me PIR je k vidění na obrázku 2.11.

### B&B Thermo-Technik

Čidlo s označením PIR-T1-M1-L0 spadající do vyšší cenové kategorie díky ceně převyšující 600,- Kč za kus. Zajímavou vlastností je možnost analogového čtení hodnot. Dále také možnost úpravy citlivosti měnící dosah od 4 do 8 metrů [28].

### Zvolené PIR

Nejlepší parametry se uvádí u čidla od firmy B&B, které však bylo uvedeno spíše pro zajímavost. To proto, že systém v této práci se zaměřuje na cenově dostupná čidla a cenovka oproti nejlevnějšímu čidlu je 30× vyšší. Model HC-SR505 má výhodu pouze v malých rozměrech, což v rámci této práce není přednostně důležité. U čidla Me PIR je výhodou snadná instalace, nicméně ostatními parametry konkurenci nepřevyšuje. Rozhodnutí tedy spočívá mezi čidly HC-SR501 a Sparkfun SE-10. Díky zajímavým parametrům, širokým možnostem úpravy citlivosti a velmi nízké ceně je nakonec vybráno právě čidlo HC-SR501. K prohlédnutí je na obrázku 2.12.



Obrázek 2.11: Me PIR [27]



Obrázek 2.12: HC-SR501 [23]

### 2.3.2 Čidla kouře a plynu

V této kategorii není zdaleka tak velká konkurence a různorodost použitých technologií jako u pohybových čidel. Následuje proto pouze krátký souhrn používaných čidel. Zástupce této kategorie je na obrázku 2.13.

#### MQ-2

Čidlo je citlivé na kouř a dalších 6 plynů jako jsou LPG a propan. Obvod lze napájet pomocí 5V. Výhodou je dlouhá životnost a rychlá reakce senzoru [29]. Po připojení senzoru k elektrické síti se musí nechat zahřát po dobu 3 minut. Pokud k zahřátí nedojde, může senzor hlásit chybné výstupy. Cena je od 30,- Kč v zahraničí a od 90,- Kč v ČR.

#### MQ série

Existuje mnoho dalších čidel z MQ série, které se zaměřují na detekci určitého spektra plynů. Série čítá téměř 20 různých modelů [21].

#### Zvolené čidlo kouře a plynu

Velmi rozšířeným a cenově dostupným čidlem je právě MQ-2. Pro tuto práci je naprosto vyhovující a není potřeba uvádět detaily podobných čidel.

### 2.3.3 Magnetické spínače

U tohoto typu čidel není příliš velký rozdíl v parametrech a liší se především kvalitou konstrukce a cenou, případně výchozím stavem (NO, nebo NC) sepnutí kontaktu. Dané označení znamená, že bez působení magnetu na čtecí hlavu je NO kontakt rozepnutý a NC sepnutý. Cena všech uvedených spínačů je včetně magnetu a montážních šroubů, které jsou součástí balení.



Obrázek 2.13: MQ2 [29]



Obrázek 2.14: MC-38 spínač [30]

### MC-38

Velmi levný spínač, který je možné zakoupit od 30,- Kč. Jedná se o typ s NO kontaktem [30]. Jeho výhodou je nízká cena. Nevýhodou je nedostupnost na českém trhu společně s nulovým zdrojem informací od výrobce. Ke spínači nelze najít datasheet a lze se tak spolehnout pouze na web prodejců. Obrázek spínače si lze prohlédnout na předchozí straně na obr. 2.14.

### AL28

Kvalitnější a také dostupnější čidlo. Poskytuje krytí IP67 a funguje na stejném principu jako předchozí čidlo MC-38. Tedy při oddálení magnetu se kontakt rozepeje [31]. Senzor je dostupný za cenu 90,- Kč.

### ALF-MC04

Masivní kovový magnetický NO spínač určený i do náročných podmínek [32]. Funguje spolehlivě i při instalaci na kovová vrata nebo dveře. Výhodou je dobrá dostupnost. Cena se pohybuje kolem 200,- Kč za kus.

### Zvolený magnetický spínač

Pro komerční nasazení by volba připadala zřejmě na čidlo ALF-MC04, které disponuje skvělými vlastnostmi. Pro účely této práce bude však plně **dostačující druhé čidlo AL28**, které je sice plastové, ale poskytuje částečné krytí proti vodě a prachu.

### Komplikace s dostupností

Mezitím než došlo k samotné objednávce všech čidel, nastala komplikace s dostupností již zvoleného čidla AL28. V důsledku toho byla **objednána čidla od firmy Jablotron s označením SA-211**. Ta sice neposkytují krytí IP67, ale mají velmi malé rozměry a lze je snadno umístit i do menších prostor. Maximální vzdálenost pro sepnutí je dva centimetry [33].



Obrázek 2.15: AL28 [31]



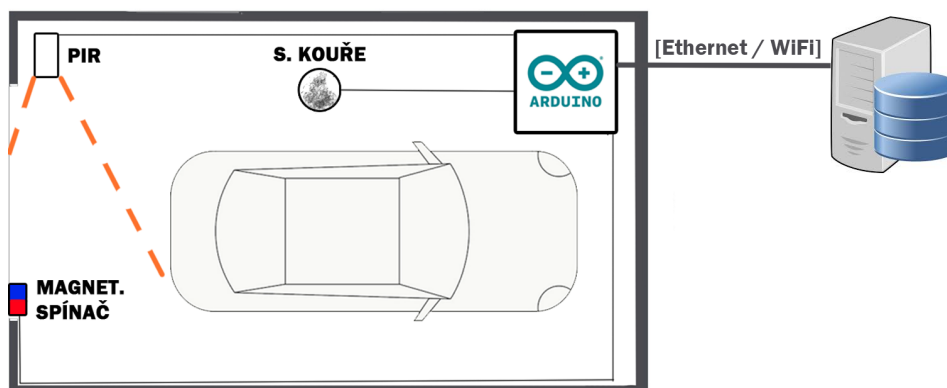
Obrázek 2.16: SA-211 [33]

## Návrh systému

Základním krokem vedoucím k sestavení funkčního systému je správný návrh. Jeho součástí je vytvoření logiky systému, která definuje v jakých stavech se může systém nacházet. Již v průběhu návrhu je důležité věnovat dostatečnou pozornost spolehlivosti a bezpečnosti (viz podkapitoly 3.3 a 3.4). Dále musí být navržen komunikační protokol – podkapitola 3.5. Završením této kapitoly je návrh schématu zapojení (3.6) a samotná implementace v C++.

### 3.1 Základní návrh

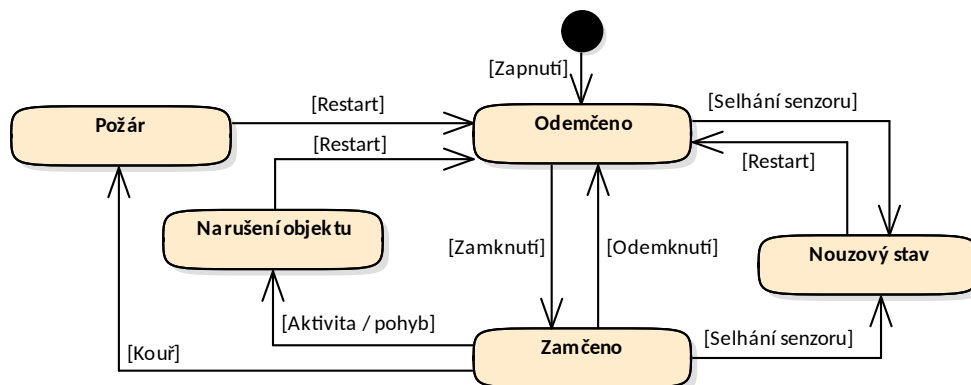
Účelem této podkapitoly je poskytnout úvod do praktické části práce a nastínit, jak přibližně bude systém vlastně vypadat. Díky předešlé analýze 2.2 je jasné, že pro monitorování garáže bude použit senzor kouře, PIR čidlo a magnetický spínač. Srdcem všeho je právě Arduino Yún, které přijímá data ze senzorů. Tato data následně zpracuje, vyhodnotí a přes komunikační rozhraní odešle závěry, či naměřená data na server.



Obrázek 3.1: **Základní návrh systému v prostoru garáže.** Zdroje: ikona Arduino: [www.arduino.cc](http://www.arduino.cc), ikona serveru: [www.lemerg.com/628256.html](http://www.lemerg.com/628256.html).

## 3.2 Logika systému

Logika systému definuje stavy, ve kterých se systém může nacházet a akce, které musí nastat pro přechod do jiného stavu. Vše je znázorněno na stavovém diagramu 3.2, jehož jednotlivé stavy budou nyní detailněji popsány.



Obrázek 3.2: Stavový diagram garážového systému skládající se z pěti stavů. Událost zamknutí, odemknutí a restartu je simulována tlačítkem.

### 3.2.1 Odemčeno

Po připojení Arduina k napájecímu zdroji dojde nejprve k prvotní inicializaci senzorů a nastavení pinů. Následně se systém nachází právě ve stavu *Odemčeno*. Jedná se o výchozí stav, do kterého lze přejít několika způsoby. Zapnutím systému, restartem v jednom z nouzových stavů, nebo nejběžnějším způsobem – odemknutím garáže. V odemčeném stavu může z pohledu diagramu dojít pouze ke dvěma událostem. Buď k uzamčení garáže (pomocí tlačítka), nebo k odhalení nefunkčnosti senzoru a předání řízení do *Nouzový stav*. Oproti jiným stavům systému zde nedochází k žádné významné činnosti a ve většině případů se pouze čeká na signál k uzamčení garáže.

### 3.2.2 Zamčeno

Nejčastěji je systém právě ve stavu *Zamčeno*. V ideálním případě jsou tyto první dva stavy jediné, ve kterých se systém kdy nachází. Odemknutí garáže je znovu simulováno pomocí tlačítka. V praxi to však může být čtečka čipových karet nebo jiné komplexní řešení, se kterým bude zabezpečovací systém komunikovat. Při detekci aktivity, či kouře a plynu dojde okamžitě k přechodu do stavu narušení objektu, respektive požáru.

### 3.2.3 Požár a narušení objektu

Primárním účelem těchto stavů je co nejrychleji informovat o nebezpečné situaci a poté nadále poskytovat zprávy o aktuálním dění v garáži. Pro tyto účely jsou zřízeny jednoduše čitelné metody s výstupy o aktuálním stavu senzorů.

### 3.2.4 Nouzový stav

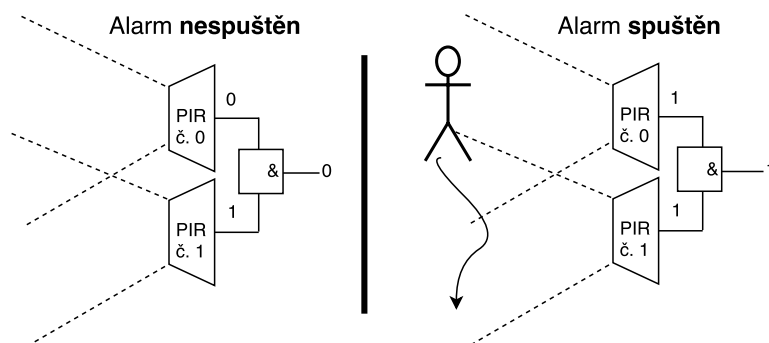
V nouzovém stavu se systém nachází pouze v případě, že došlo k selhání některého ze senzorů kouře a plynu. Případně pokud došlo k rozpojení některých částí obvodu. Tímto stavem tedy systém signalizuje pravděpodobnou nutnost odborného zásahu za účelem výměny nebo opravy obvodu.

## 3.3 Spolehlivost systému

Jelikož je cílem práce dosáhnout odolnosti proti chybám způsobených nepřesným měřením, jsou zde popsány metody pro eliminaci chyb a detekci selhání.

### 3.3.1 Chyby při měření

Jak dosáhnout toho, aby se celý systém choval spolehlivě a například nedetekoval falešně pohyb v prázdné místnosti? Ideálně lze tohoto cíle dosáhnout použitím kvalitních senzorů, u kterých se výrobce zaručí k požadované spolehlivosti a k chybám nedojde. Zde je však situace úplně jiná a s použitím nejlevnějších součástek se předpokládají komplikace. Pro zamezení falešných poplachů se použije více totožných PIR senzorů současně – redundance. Předpokladem je, že PIR budou monitorovat stejný prostor. Lze uvažovat, že pokud došlo k pohybu v tomto prostoru, projeví se to na všech čidlech. Naopak v případě, kdy došlo k sepnutí pouze jednoho z instalovaných čidel, bude se hodnota považovat za chybnou.



Obrázek 3.3: **Eliminace chyb při měření s PIR.** Levá strana zobrazuje chybné měření u PIR č. 1 s vyhodnocením alarmu jako nespuštěn. Vpravo je skutečný pohyb, který musí ke spuštění alarmu zaznamenat obě PIR.

#### 3.3.2 Výpadek čidla

U **magnetických spínačů** lze problém s výpadkem čidla vyřešit pomocí pull-up rezistorů. Spínače jsou za normálních okolností při uzamčení garáže sepnuty. Narušení obvodu s magnetickými spínači, ať už z jakéhokoliv důvodu, však vede ke spuštění alarmu. Poroucháním čidla, či přerušením drátů vedoucích od magnetických spínačů vznikne stejný poplach, jako kdyby došlo k otevření dveří.

Softwarovou úpravou u **čidla kouře a plynu** je možné předcházet falešnému spuštění alarmu. To pomocí průběžného testování, zda na analogovém pinu nejsou příliš nízké nebo vysoké hodnoty. Při odpojení senzoru od GND se na analogovém pinu skokově objeví obrovské hodnoty, kterých ani v běžném prostředí nelze dosáhnout. Při odpojení  $V_{cc}$  pinu se naopak objeví téměř nulové hodnoty, na které se v běžném provozu detektoru nelze dostat.

Výpadek **PIR čidla** nelze určit jistě ani jedním z předchozích způsobů. Pro spolehlivé odhalení poruchy by muselo dojít ke sledování jak napájecích pinů, tak digitálního kabelu. To by vedlo k vyčerpání téměř všech I/O pinů u Arduina Yún. V případě tohoto systému je použito více faktorů prostorové ochrany a není zcela nezbytné zjistit výpadek PIR ihned. Informace o této problematice byly čerpány z [34].

#### 3.3.3 Výpadek elektřiny

Výpadek elektřiny lze řešit více způsoby. Jedním z nich je použití integrovaného obvodu pro přepnutí napájení z baterie v případě, že hlavní zdroj selže. Toto řešení je však poměrně náročné, kvůli nutnosti navrhnout si vlastní specializovaný obvod. Další možností je zakoupit vývojovou desku ve verzi s PoE rozšířením a napájet Arduino přes Ethernet [35]. Tím se tedy přesouvá zodpovědnost za výpadek elektřiny na servery, které často mívají UPS právě pro eliminaci takovýchto výpadků. Ani jedna z těchto možností však v práci nebude využita. To především kvůli omezenému rozpočtu a rozsahu práce. Případný výpadek napájení může být monitorován na straně serveru, který při dlouhotrvající odmlce v komunikaci podnikne náležité kroky.

### 3.4 Bezpečnost systému

Arduino se zaměřuje především na uživatelský komfort a přívětivé prostředí. To však přináší mnoho negativ z pohledu bezpečnosti. V analýze [36] vývojové desky Yún bylo objeveno mnoho zranitelností. Úvod této podkapitoly je věnován bezpečnosti integrovaných obvodů ATmega32u4 a Atheros AR9331 zmíněných v podkapitole 2.1.4.

Jedno z míst, kde byl kladen důraz na komfort místo bezpečnosti je WiFi. Ve výchozím nastavení OpenWrt dochází k automatickému zapnutí WiFi hot-



spotu. Zapnutí nastane v případě, kdy se Yún nepodaří do jedné minuty připojit k přednastavené WiFi síti.

V analýze Yún bylo dále objeveno a zneužito několik zásadních zranitelností. Yún neposkytuje kontrolu selhání paměti, ani bezpečnostní prvky jako ASLR, nebo ochranu zásobníku kanárky<sup>6</sup>. Proto je možné zneužít přetečení zásobníku k DoS, nebo získání vyšších privilegií v Linuxovém systému. Některým zranitelnostem lze zamezit manuální instalací novějších (neoficiálních) verzí systému, knihoven a zakázáním nepoužívaných služeb v Linux systému.

Identifikované zranitelnosti však stojí na předpokladu, že je k zařízení fyzický nebo vzdálený přístup. Na problém s horší bezpečností se lze podívat z jiného pohledu. Jak co nejvíce omezit veškerý přístup k zařízení a přenechat bezpečnost na nadřazeném systému. Na základě této myšlenky se bude odvíjet další návrh systému včetně podkapitoly 3.5 s názvem komunikace.

### 3.4.1 Hrozby a útoky

Tato část volně vychází z klasifikačního schématu STRIDE [37], které napomáhá identifikaci hrozeb<sup>7</sup>. Pro zachování rozumné délky práce bude uveden pouze stručný přehled závažných bezpečnostních kategorií. K nim budou uvedena opatření proti některým hrozbám, či jiná bezpečnostní zlepšení systému.

#### Fyzický útok

Příkladem fyzického útoku je přerušení kabelu, či jiná snaha poničit systém a jeho komponenty. Hrozbou spočívající v přerušení kabelu se může zabývat server komunikující s Arduinem. Pokud neobdrží odpověď po delší časový úsek, může ohlásit narušení nebo jinak informovat o tomto incidentu. Poničení komponent systému hrozí pouze v tom případě, kdy má útočník fyzický přístup. To už by měl systém dávno ohlásit stavem narušení.

#### Ovládnutí systému

Do konfigurace Yún je možné přistupovat několika způsoby. Typicky pomocí SSH připojení po zadání platného uživatelského jména a hesla. Další možností je přistoupit pomocí prohlížeče do webového rozhraní nazvaného LuCi. Zde je samozřejmě také nutné zadat platné přístupové heslo. V tomto webovém rozhraní lze striktně omezit přístup a upravit další síťová nastavení. Možností, jak obejít princip přihlašování, je odposlouchávání komunikace. Zde stačí využít dočasně vygenerovaný autentizační řetězec a vydávat se za již autentizovanou stranu. Pro uskutečnění tohoto útoku je nutný přístup na server komunikující s Arduinem. Možnou ochranou před těmito útoky může být úplné zamezení přístupu, či šifrování veškeré komunikace.

---

<sup>6</sup>ASLR a kanárci jsou systémová, resp. softwarová opatření proti přetečení bufferu.

<sup>7</sup>Hrozba (threat) je potencionální příčina nežádoucího incidentu, který může mít za následek poškození systému nebo organizace [38].

#### Spoofing a tampering

Pojmy spoofing a tampering označují podvržení identity, respektive pozměnění dat. Tvoří dvě z celkových šesti kategorií pro klasifikaci hrozeb v modelu STRIDE. Obě kategorie úzce souvisí s již zmíněnými hrozbami. Pro uskutečnění útoku na systém by útočník opět musel získat fyzický přístup a proto nebudou hlouběji popisovány.

#### 3.4.2 Bezpečná komunikace

Jak již bylo zmíněno, Arduino Yún obsahuje mikroprocesor Atheros AR9331, na kterém běží openWRT. Díky tomu je k dispozici mnoho nástrojů. Od Python skriptování po kryptografické nástroje jako OpenSSL nebo odlehčené PolarSSL pro vestavné systémy [39]. V rámci této práce probíhá komunikace pouze v lokální síti. Jedinou možností, jak by mohl útočník odposlouchávat data, či zfalšovat komunikaci, spočívá ve fyzickém přístupu k zařízení, Ethernetu nebo zařízení komunikujícím s Yún. Z těchto důvodů se v aktuální verzi šifrovaná komunikace nenachází. Její implementace by byla na úkor důležitějších bodů této práce. Pokud by Arduino komunikovalo se vzdáleným serverem na internetu, byla by situace úplně jiná.

### 3.5 Komunikace

Pojem komunikace má v této podkapitole více významů. Jako první je stanoven komunikační protokol, na jehož návrhu bude realizována komunikace se serverem. Následně bude rozhodnuto, po jakém komunikačním rozhraní se budou data přenášet. Detaily přibližující implementační technologie použité pro realizaci komunikace se nachází v části 3.7.4.

#### 3.5.1 Komunikační protokol

Zde je stanoven jednoduchý textový protokol definující pravidla komunikace mezi Arduinem a serverem. Jednotlivé příkazy jsou uvedeny v tabulce 3.1 a následně blíže popsány.

Příkaz	Stručný popis
state	Aktuální stav systému.
values	Vypiš stručně hodnoty.
valuesHelp	Nápověda pro příkaz values.
debug	Vypiš hodnoty s popisem.
time	Čas od poslední změny stavu.
commands	Seznam všech příkazů.

Tabulka 3.1: Jednoduchý komunikační protokol

**State**

Po odeslání příkazu *state* obdrží odesílatel informaci o aktuálním stavu garáže. Stav odpovídají návrhu logiky systému v části 3.2. Jedinou změnou je pojmenování stavů v angličtině<sup>8</sup>, protože celý program je psán anglicky.

**Values**

```
PIR1_D,...;S&G1_A,...;S&G1_D,...;MS1_D,...;
```

<sup>9</sup>

Příkazem *values* se dotazuje zkrácený výpis aktuálně naměřených hodnot na všech čidlech. Formát výpisu je navržen s ohledem na snadné přidávání a odebrání senzorů v systému. Bližší informace o výpisu jsou v popisu následujícího příkazu *valuesHelp*.

**ValuesHelp**

*ValuesHelp* poskytuje nápovědu o formátu odpovědi z příkazu *values*. Hodnoty stejné skupiny čidel jsou odděleny čárkou a skupiny jsou od sebe odděleny středníkem. Písmeno za podtržítkem značí, zda se jedná o digitální, nebo analogovou hodnotu. Poslední středník má ukončovací význam.

**Debug**

Podrobněji rozepsaný výpis aktuálně naměřených hodnot lze získat pomocí příkazu *debug*. Slouží především pro testovací a ladící účely.

**Time**

*Time* je příkaz, který vrací čas od poslední změny stavu. Odpověď je ve formátu HH:MM:SS.

**Commands**

*Commands* je jednoduchý příkaz na vypsání všech dostupných příkazů. Každý z nich je na novém řádku.

**3.5.2 Komunikační rozhraní**

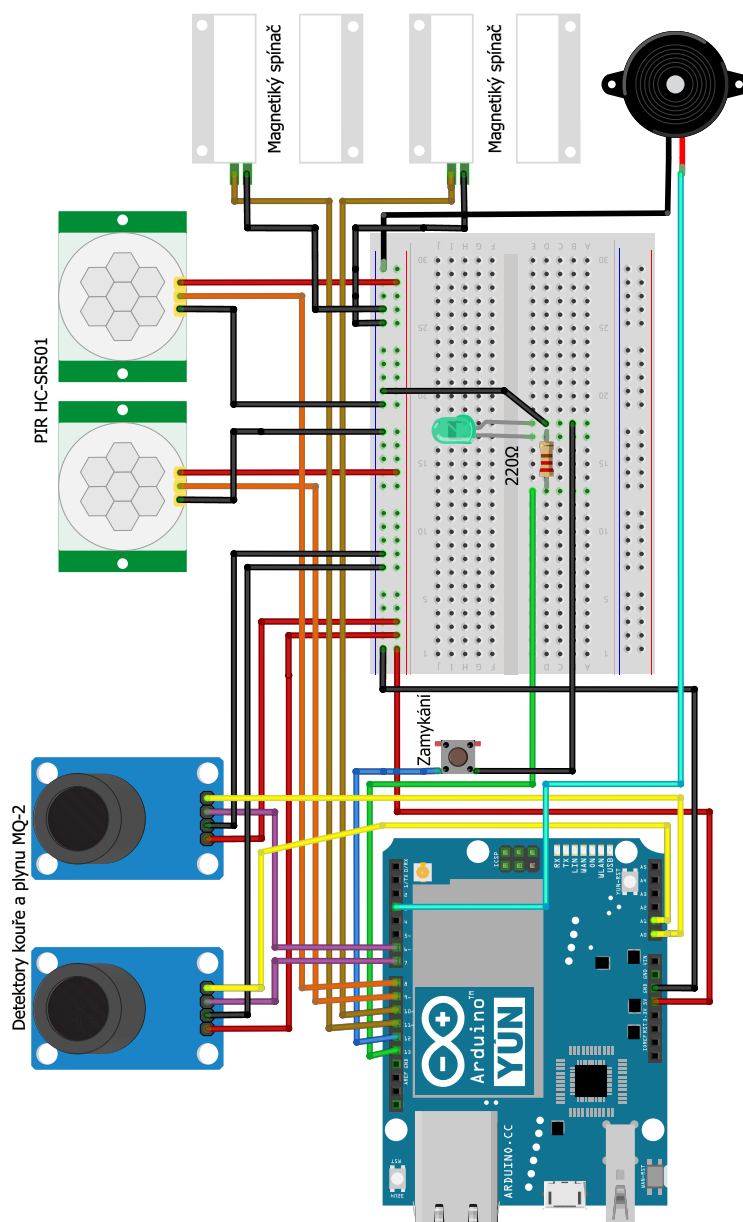
Díky univerzálnosti vývojové desky Yún přichází na řadu rozhodnutí pro WiFi nebo Ethernet. Volba pro jednu z těchto technologií je nakonec poměrně přímočará. Hlavním ohledem je navýšení bezpečnosti systému. Pro pracnější možnosti odposlechu a narušení komunikace bude upřednostněna drátová komunikace, tedy Ethernet. Nevýhoda spočívá v náročnější instalaci a možnosti přerušit fyzicky spojení. Výhoda WiFi spočívá především ve vyšším komfortu. To však na úkor nižší spolehlivosti přenosu a již zmíněné bezpečnosti.

<sup>8</sup>Tedy LOCKED, UNLOCKED, FIRE, DISRUPTION a EMERGENCY.

<sup>9</sup>Formát odpovědi po zavolání příkazu *values*.

## 3.6 Schéma zapojení

Kompletní schéma zapojení modelované pomocí open-source programu Fritzing v režimu „montážní deska“. Fotografie z reálného zapojení systému jsou v příloze B.



Obrázek 3.4: **Schéma zapojení** zobrazující finální propojení komponent. Pro testování systému je zde mimo detektorů také zelená dioda a piezoelektrický bzučák. Spínací tlačítko simuluje zamykání a odemykání garáže.

## 3.7 Implementace

Zdrojový kód je pro vyšší přehlednost členěn do 6 souborů a čítá přes 600 řádků. Bližší informace o členění kódu jsou v příloze C. Program je psán s ohledem na jednoduchost, rozšiřovatelnost a udržitelnost. Dostupná flash paměť mikroprocesoru ATmega32u4 je využita programem z 62%. Dynamická SRAM paměť je využita z 64%.

### 3.7.1 Vývojové prostředí

Výchozí vývojové prostředí (Arduino IDE) je pro psaní větších programů poměrně špatné. Jednou možností, jak zde strukturovat kód, je psaní programu do více souborů s koncovkou *.ino*. Ty se v Arduino IDE zobrazí podobně jako záložky v prohlížeči. Výhodou Arduino IDE je opravdu jednoduché prostředí, ve kterém není těžké se zorientovat. Mezi přidané hodnoty bohužel patří pouze zvýraznění syntaxe a nástroje pro práci s knihovny. Při snaze využít CLion IDE společně s pluginem pro Arduino se, bohužel, v rozumném čase nepodařilo překonat prvotní chyby. Teprve poté bylo nalezeno vývojové prostředí PlatformIO poskytující více možností. Podporuje například debugování, inteligentní doplňování kódu nebo automatické formátování kódu. Seznam vývojářských nástrojů včetně přehledu IDE je k dispozici na oficiálních stránkách Arduina<sup>10</sup>.

### 3.7.2 Načítání dat

Snadná udržitelnost zdrojového kódu je podpořena použitím dvouvrstvé architektury. Veškeré konstanty (kromě některých textů) jsou umístěny ve vyhrazeném souboru *constants.h*. Pro přidání, nebo odebrání některého čidla v systému je tak potřeba minimum úprav. Jedním z příkladů takto navržených metod je analogové a digitální načítání dat. Pro ilustraci je uvedena část zdrojového kódu na obrázku 3.5.

```
void digitalReadToArray( const int * pins, int numOfPins, bool * resultArray ){
    for(int i =0; i < numOfPins; ++i)
        resultArray[i] = digitalRead(pins[i]);
}
void analogReadToArray( const int * pins, int numOfPins, int * resultArray ){
    for(int i =0; i < numOfPins; ++i)
        resultArray[i] = analogRead(pins[i]);
}
```

Obrázek 3.5: **Uložení dat** do pole na základě parametrů určujících jaké piny se mají číst. Rozlišeno je analogové a digitální čtení.

<sup>10</sup><https://playground.arduino.cc/Main/DevelopmentTools>

#### 3.7.3 Stavový manager

V první části programu dojde k nastavení pinů a jednotlivých komponent. Poté probíhá zhruba tříminutová inicializace, při které dojde k ustálení hodnot PIR čidel společně s detektory kouře a plynu. Řízení programu je poté předáno do nekonečné smyčky, ve které probíhá obsluha v závislosti na aktuálním stavu. Na obrázku 3.6 je zobrazena obsluha nejsložitějšího stavu – *Zamčeno*.

```
void stateManage(){
    switch (state) {
        case LOCKED:
            if( isGarageOnFire() ) {
                changeNextState(FIRE);
                tone(BUZZER_PIN, BUZZER_FREQUENCY);
                Serial.println("FIRE - gas escape or fire.");
            }
            else if( isThereMotion() ) {
                changeNextState(DISRUPTION);
                tone(BUZZER_PIN, BUZZER_FREQUENCY);
                Serial.println("DISRUPTION - motion.");
            }
            else if( !areDoorsClosed() ) {
                changeNextState(DISRUPTION);
                tone(BUZZER_PIN, BUZZER_FREQUENCY);
                Serial.println("DISRUPTION - opened doors.");
            }
            else if( isChangeOfState() ) {
                changeNextState(UNLOCKED);
                Serial.println("UNLOCK - lock possible after 3 sec.");
            }
            break;
    }
}
```

Obrázek 3.6: **Obsluha stavu zamčeno** probíhá v nekonečné smyčce. Neustále je kontrolováno, zda nedošlo k nebezpečné události nebo stisku tlačítka.

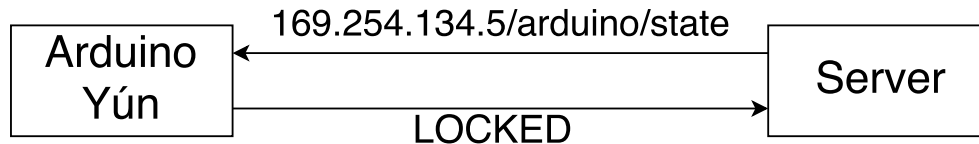
#### 3.7.4 REST komunikace

Po připojení Ethernet kabelu a autentizaci k Yún je možné využít REST<sup>11</sup> API poskytované Yún knihovnamí. V tuto chvíli je z pohledu komunikačního protokolu serverem Arduino, nicméně serverem se stále bude označovat zařízení sbírající data. Server po odeslání požadavku ve tvaru *<IP adresa Yún>/arduino/<příkaz protokolu>* obdrží od Arduina zprávu, kterou už může

---

<sup>11</sup>REST, neboli Representational state transfer je datově orientovaný architektonický styl zajišťující jednotný přístup ke zdrojům pomocí metod GET, POST, PUT a DELETE [40].

libovolně vyhodnotit. Například zavolat hasiče po obdržení stavu *Požár*. Požadavky se odesílají pomocí jednoduchých HTML volání. Pro ty je možné využít webový prohlížeč nebo například nástroj cURL.



Obrázek 3.7: **Ukázka komunikace** s konkrétní IP adresou a požadavkem na Arduino. V odpovědi HTML požadavku je vrácen aktuální stav garáže.

K tvorbě REST API je využita třída `BridgeClient` z Arduino knihovny `Bridge`. Díky této třídě je možné jednoduše číst a odpovídat na příchozí HTML požadavky. Pro zpracování požadavků je využita metoda `whitelisting`<sup>12</sup>.

```

void processClientAndStop(BridgeClient client){
    String command = client.readString(); // read the command
    command.trim(); // remove whitespaces

    if (command == "state" )
        getState(client);
    else if (command == "values")
        getValues(client);
    else if (command == "debug")
        debug(client);
    else if (command == "time")
        timeFromStateChange(client);
    else if (command == "valuesHelp")
        getValuesHelp(client);
    else if (command == "commands")
        commands(client);
    else
        client.print(F("Wrong command"));
    client.stop();
}
  
```

Obrázek 3.8: **Implementace jednoduchého REST API** pomocí třídy `BridgeClient`. Po odeslání odpovědi dojde k ukončení spojení.

<sup>12</sup>Jakýkoliv požadavek, který není explicitně povolen je zamítnut. Tím se předchází neočekávaným situacím.

### 3.8 Náklady

Celková útrata za systém činí 2 562,- Kč. Jedním z požadavků v zadání práce je vybrat cenově dostupná čidla a senzory. Za tyto položky bylo celkem utraceno 458,- Kč. Nejdražší částí celého systému se stala vývojová deska Arduino Yún. Při objednání některých dílů ze zahraničí by se celková cena zřejmě dostala pod 2 tisíce Kč. Další z možností, jak snížit celkovou cenu, je navrhnout vlastní DPS (desku plošných spojů), nebo vybrat levnější vývojovou desku.

<b>Položka</b>	<b>Cena (v Kč)</b>
Arduino Yún	1 774,-
Magnetické spínače (2x)	142,-
PIR (2x)	118,-
Senzor kouře (2x)	198,-
Nepájivé pole	50,-
Propojovací kabely	160,-
Ethernet a USB kabel	80,-
Ostatní	40,-
<b>Celkem</b>	<b>2 562,-</b>

Tabulka 3.2: Náklady za veškeré komponenty



## Testování

V této kapitole jsou uvedeny poznatky, komplikace a veškeré informace související s testováním systému. To jak jednotlivých čidel v počáteční fázi, tak i celého systému jako celku. Všechna čidla byla zakoupena po dvou až třech kusech, aby mohla být použita redundantně a zvýšila se tak celková spolehlivost systému. Díky tomu testování často probíhalo souběžně na více čidlech.

### 4.1 Magnetické spínače

Při testování magnetických spínačů nedošlo k žádnému problému, či rozporu skutečnosti s dokumentací. Nejnáročnější částí tak zřejmě bylo správné napájet pin koncovky. Spínače spolehlivě a bez kolísání detekují silnější magnet do vzdálenosti 2–3 cm. S dodávaným (slabším) magnetem je vzdálenost přibližně 2 cm. V dokumentaci je uváděna vzdálenost 19 mm za ideálních podmínek. Tento parametr však ovlivňuje nejen použitý magnet, ale také okolní feromagnetické materiály.

#### 4.1.1 Test spínačů

Magnetická čidla byla testována na správné rozpoznání přiloženého magnetu. Při testu došlo k opakovanému přiložení magnetu a naměření hodnot. Logika hodnot je obráceně díky pull-up rezistoru. Prvotní test magnetických spínačů je zaznamenán v tabulce 4.1. Naměřené výsledky u již kompletního systému jsou k dispozici v tabulce 4.3.

Měření č.	1	2	3	4	5	6
Spínač 1	0	0	0	0	0	0
Spínač 2	0	0	0	0	0	0

Tabulka 4.1: **Test magnetických spínačů.** Nulová hodnota značí přiložený magnet. Všechna měření tedy dopadla dle očekávání.

## 4.2 Čidla kouře a plynu

U zvolených senzorů MQ-2 popsaných v kapitole 2.3.2 je k dispozici analogový i digitální výstup. Digitální výstup je možné regulovat potenciometrem zabudovaným přímo na detektoru. Stačilo tedy zjistit jaká je mezní hodnota v čistém ovzduší, nastavit na potenciometru rezervu a detekovat hodnoty na digitálním výstupu. Při mnohačetném testování několika různých MQ-2 senzorů tento způsob fungoval spolehlivě. Problém nastal při testování v úplně jiném prostředí. Rozdíly v teplotě a vlhkosti najednou způsobily zvýšenou detekci na přítomnost plynu a kouře. Nepomohl ani delší časový úsek na ustálení hodnot. Na základě těchto poznatků byl systém upraven tak, aby se automaticky adaptoval na aktuální ovzduší.

**Adaptace v závislosti na ovzduší** se ukázala být pracná. Analogové hodnoty detektorů jsou po ustálení dost odlišné v závislosti na kusu. To i v případě, kdy se nachází v klidovém stavu při naprosto stejných podmínkách. Při testování plynem ze zapalovače se znatelně liší i maximální naměřené hodnoty. V důsledku těchto znalostí je pro správné fungování systému nutné zkalibrovat každé čidlo zvlášť. Systém byl proto přepracován na analogové vyhodnocování, při kterém se aktuální data porovnávají s referenční hodnotou. Referenční hodnota se stanoví po ustálení systému (třech minutách od zapnutí).

### 4.2.1 Testovací měření 1

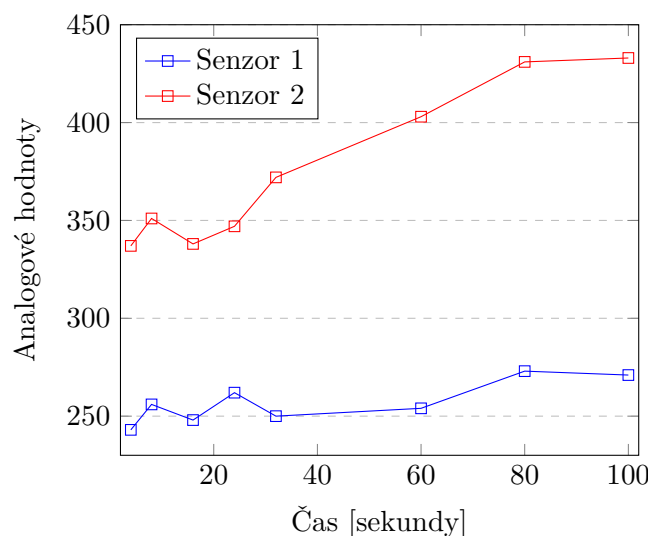
Čidla kouře a plynu byla testována pomocí butanu z klasického zapalovače. V tabulce 4.2 je přehled naměřených hodnot v čase od puštění plynu. Je zde vidět velký rozdíl v počátečních klidových i maximálních hodnotách. To poukazuje na nepříliš velkou kvalitu součástek. Až při posledním měření došlo po 60 sekundách k překročení limitu a vyhodnocení situace jako únik plynu.

Čas [s]	4	8	16	24	32	44	60
<b>Senzor 1</b>	244	249	242	249	247	251	253
<b>Senzor 2</b>	341	343	354	363	397	404	<b>413</b>

Tabulka 4.2: **Test čidla kouře a plynu.** Do malého uzavřeného objektu byl pouštěn plyn po dobu 10 sekund. Analogový záznam probíhal 60 sekund.

### 4.2.2 Testovací měření 2

Plyn byl pouštěn po dobu 20 sekund. Minimální a maximální měřitelná analogová hodnota je 0, respektive 1023. V grafu 4.1 je krásně vidět propastný rozdíl mezi naměřenými hodnotami. Rozdíl často převyšuje 10% celého analogového pásma. K překročení bezpečných hodnot došlo u senzoru 2 na hodnotě 403 po 60 sekundách.



Obrázek 4.1: **Druhé testovací měření čidel kouře a plynu.** Zobrazuje analogové hodnoty čidel v průběhu testu. Plyn byl pouštěn po dobu 20 sekund.

## 4.3 PIR

Už při počátečním testování jednotlivých senzorů HC-SR501 (popsaných v části 2.3.1) se ukázalo, že PIR budou problematická.

### 4.3.1 Prvotní testování

Celkem byly objednány tři senzory a po prvních testech se zdálo, že čidla jsou naprosto nepoužitelná. Při testování nastalo mnoho komplikací najednou a bylo časově náročné zjistit v čem je problém. Špatně odhalitelnou komplikaci způsoboval nejprve nekvalitní kabel, který bylo nutné vyměnit. Ani poté však nevypadaly výstupy jednotlivých senzorů smysluplně.

V procesu testování bylo vystřídáno několik místností a vyzkoušeno odlišné umístění senzorů. Senzory byly testovány všechny zároveň, ale i jednotlivě. První senzor nedetekuje pohyb téměř nikdy. Druhý je naopak příliš citlivý a dochází k falešným detekcím. Teprve třetí zvládá alespoň do určité míry odhalit pohyb a nepůsobit přemíru falešných poplachů. Na základě tohoto prvotního testu bylo vyhodnoceno, že jsou čidla vadná nebo nekvalitní a zamýšlelo se zakoupení jiných PIR.

### 4.3.2 Druhá šance

Nakonec se pro systém použila přece jen PIR čidla HC-SR501. Spolehlivějšího fungování bylo dosaženo pomocí softwarové úpravy, o které je psáno v části 3.3.1. Další úpravou vedoucí k lepší spolehlivosti bylo prodloužení inicializač-

ního času na ustálení hodnot. V datasheetu [23] je doporučeno počkat jednu minutu, nicméně až dvě minuty se ukázaly být dostačující. I přes tato zlepšení občas k falešným poplachům dochází. Stejně tak se vyskytuje problém, kdy čidlo detekuje nejen pohyb lidský, ale i jakéhokoli předmětu. To i při nastavení nejnižší citlivosti. Výsledky z testování PIR čidel jsou uvedeny v tabulce 4.3.

## 4.4 Celý systém

Pro jednoduchost a dobrou přenositelnost se celý systém umístil do krabice od bot. Otvírání vrat hlídané magnetickými spínači je simulováno zvednutím víka krabice. Pohybová PIR čidla mají vyříznuté otvory a snímají okolní prostor (mimo krabici). Obrázky systému si lze prohlédnout v příloze B.

### 4.4.1 Test PIR a magnetických spínačů

Tento test byl proveden na finální verzi systému a jeho účelem je ověřit správnou reakci (je simulováno narušení útočником). PIR byla nasměrována ke zdi vzdálené přibližně tři metry. Testování PIR probíhala pohybem osoby ve snímaném prostoru. Magnetické spínače byly testovány nadzvednutím víka krabice. Střídavě v místech kde jsou spínače instalovány, nebo souběžně otevřením celého víka. Výsledky v přívětivých<sup>13</sup> podmínkách jsou k dispozici v tabulce 4.3. Výsledek „OK“ značí správné vyhodnocení. V průběhu testování došlo k **jednomu falešnému poplachu**.

Měření číslo	PIR	Magnetický spínač
1	OK	OK
2	OK	OK
3	OK	OK
4	OK	OK
5	OK	OK
6	OK	OK
7	OK	OK
8	OK	OK
9	OK	OK
10	OK	OK
<b>Úspěšnost</b>	<b>100%</b>	<b>100%</b>

Tabulka 4.3: **Test PIR a spínačů na již kompletním systému.** Výsledek OK značí správnou reakci systému na pohyb osoby nebo otevření víka. Nemí zde promítnut falešný poplach, který v průběhu testování nastal kvůli PIR.

---

<sup>13</sup>Bez prudkých změn teplot, proudění vzduchu s eliminací slunečního záření.

#### 4.4.2 Falešné poplachy

Při stejných podmínkách, jako v předchozí části 4.4, byla otestována **četnost falešných poplachů**. Systém byl uzamčen a po dobu 45 minut bylo sledováno, zda bude falešně vyhodnocovat pohyb. Hned po 2 minutách od uzamčení došlo k falešnému poplachu kvůli PIR. Systém byl následně restartován a znovu uzamčen. V následujících 43 minutách už k žádnému falešnému poplachu nedošlo. Pro jistotu se nakonec provedl skutečný pohyb před snímačem, který byl ihned detekován.

#### 4.4.3 Funkčnost na delší vzdálenost

Při reálném nasazení systému do garáže by samozřejmě narostla i délka kabelů vedoucích k jednotlivým sensorům. Na magnetických spínačích byl proto proveden test funkčnosti pomocí 5 m kabelů. Spínače naprosto bez problémů nadále zvládaly odhalit přiblížení a oddálení magnetu. Funkčnost systému tak zůstává neporušena i při použití delších propojovacích kabelů.

### 4.5 Problém s pamětí

Při průběžném rozšiřování a upravování zdrojového kódu nastal zásadní problém. Yún z neznámých důvodů přestávalo komunikovat přes sériovou linku. Docházelo k častým výpadkům v komunikaci a pomocí USB nešlo nahrát nový program. Program bylo naštěstí možné přehrát přes WiFi, nebo Ethernet. Po několika pokusech se ukázalo, že problém nastává pouze u specifické verze zdrojového kódu pro zabezpečení garáže.

Odhalit přesnější příčinu se povedlo díky vývojovému prostředí Arduino IDE (použitého dočasně namísto PlatformIO). To při nahrání programu ohlašovalo upozornění na vyčerpání téměř 85% z volné SRAM globálními proměnnými. Mnoho pomocných výpisů tak způsobilo zahlcení SRAM o velikosti pouhých 2,5 KB. Pro vyřešení tohoto problému bylo nutné změnit přístup k některým proměnným a konstantám.



---

## Profil útočníka

V této podkapitole jsou představeny úvahy spojené s možnými útoky na systém. Úvahy jsou rozděleny do třech skupin. Prvním jsou nečekané události způsobené přírodními vlivy. Druhá a třetí skupina popisuje náhodný a cílený útok. Závěrem této podkapitoly je představeno jedno z možných protiopatření.

### 5.1 Matka příroda

Pomyslným útočníkem může být matka příroda. Při poryvech větru může dojít k pohybu vrat nebo dveří o milimetry až centimetry. Tím může dojít k rozpojení kontaktu spínače a ke spuštění alarmu. Doporučení zde spočívá v provedení kvalitní instalace čidel a co největším zamezení nežádoucího pohybu zamčených dveří a vrat. Například bezpečnostní dveře a vrata poskytují kvalitní těsnění, které by mělo tomuto problému předcházet. Obecně není možné tento problém řešit softwarově.

### 5.2 Náhodný zloděj

Uvažujme zloděje, jako náhodného kolemjdoucího, který je vybaven nástroji na překonání zámku vrat. Zároveň uvažujme, že je systém umístěn v garáži, do které jsou pouze jedna přístupová vrata.

Na základě provedených testování lze předpokládat, že k chybnému spuštění alarmu rozhodně nedojde častěji jak jednou za třicet minut. Při správné instalaci čidel do objektu by se nemělo stát, že zloděj sepne čidla pohybu pouze jednou. Předpokládejme, že náhodný zloděj pronikne do objektu. Je velmi pravděpodobné, že magnetické spínače nebo čidla pohybu ohlásí kritický stav po delší časový úsek. Z nasbíraných dat by mělo být odhadnutelné, zda se jedná o chybnou událost, nebo skutečné narušení. Stejně tak by ze záznamu analogových hodnot na detektoru kouře a plynu mělo být rozpozna-

telné, zda došlo k jednorázovému chybnému měření, či skutečně došlo k požáru nebo úniku plynu.

### 5.3 Cílený útok

V tomto případě se předpokládá útočník, který má částečné, nebo úplné znalosti o instalovaném systému. Úvahy spojené s cíleným útokem budou rozděleny právě do těchto dvou skupin – úplná a částečná znalost systému.

#### 5.3.1 Částečná znalost systému

V případě částečných znalostí si lze představit útočníka jako dobrého (bývalého) kamaráda, který si je vědom zabezpečení, ale přesné rozmístění komponent nezná. Díky kombinaci různých a navíc redundantních senzorů je pravděpodobnost překonání systému velmi malá. Pohybová PIR čidla lze do jisté míry oklamat, ale magnetické spínače nikoli (pokud neznáme jejich přesné umístění).

#### 5.3.2 Úplná znalost systému

Úplnou znalostí systému může disponovat například technik, který do garáže systém instaloval. Pokud by se takovému útočnickovi podařilo bez povšimnutí překonat fyzické zabezpečení (zamčená vrata), mohl by připevnit své vlastní magnety do míst, kde jsou čtecí hlavy magnetických spínačů. Kdyby se útočnickovi tento postup vydařil, stále ještě musí překonat pohybová PIR. Z principu funkčnosti PIR (viz 2.2.1) je teoreticky možné oklamat senzor odstíněním vyzařovaného tepla z lidského těla. Pro tento účel může posloužit několik vrstev oblečení a pomalý pohyb.

Aktuálně je však problémem použitých PIR čidel nadměrná citlivost. Proto je velmi pravděpodobné, že pouhá změna teplot proudícího vzduchu po otevření vrat způsobí spuštění alarmu.

### 5.4 Protiopatření

Jedna z možností, jak vylepšit systém a provést opatření právě proti vniknutí zloděje, je nainstalovat kameru. Ačkoliv Arduino vývojové desky nepatřím k těm nejvýkonnějším, jejich výkon je dostačující pro použití jednoduché kamery. Ta by mohla pořídit krátký záznam nebo snímek v případě, že dojde ke spuštění alarmu. Tím by bylo mnohem snazší odhalit falešné popluchy. Nejobvyčejnější VGA kamery spolupracující s Arduinem [41] jsou k dostání od 60,- Kč.



---

## Závěr

Na základě definovaných požadavků jsem **shrnul nejdůležitější teoretická témata** této práce a navrhl základní spolehlivostní model celého systému. Následně jsem vytvořil analýzu vývojových desek a vybral základní pilíř této práce – Arduino Yún. V dalším kroku jsem provedl **komplexní analýzu čidel**, ve které jsem vybral tři kategorie senzorů. Prostorovou ochranu jsem zajistil pomocí PIR a magnetických spínačů. Protipožární ochrany jsem dosáhl čidlem kouře a plynu. Z důvodu požadované spolehlivosti jsou **jednotlivé druhy čidel redundantní**. Při výběru čidel jsem narazil na první větší problém. Spočíval ve velmi slabé, nebo žádné oficiální dokumentaci k levným, ale přitom velmi rozšířeným součástkám.

**Nejnáročnější, ale zároveň i nejzajímavější** částí práce se pro mě stala **praktická část**. Návrh systému z cenově dostupných komponent mě přinutil k implementaci různých SW opatření. Po prvním testování čidel se zdálo, že jsou PIR naprosto nepoužitelná. Nakonec se mi pomocí delšího inicializačního času a SW eliminace chyb podařilo dosáhnout obstojné spolehlivosti. Systém umí také ohlašovat výpadky některých čidel.

V průběhu práce jsem zjistil, že **platforma Arduino není příliš bezpečná**. K systému jsem se snažil co nejvíce omezit fyzický přístup a použít pouze drátovou komunikaci v místní síti. Podrobnější identifikaci hrozeb pomocí modelu STRIDE jsem shledal příliš obsáhlou. Proto jsem se rozhodl použít STRIDE pouze jako opěrný bod bezpečnosti. Při implementaci systému jsem navrhl logiku o pěti základních stavech. Prvotní komunikaci jsem realizoval pomocí sériové linky. Poté jsem navrhl jednoduchý textový protokol a implementoval ho pomocí REST API. To lze použít přes Ethernet rozhraní.

Za **největší problémy** při realizaci systému bych označil hned několik témat. Zejména zdlouhavý proces objednávky komponent a nekonzistentnost čidel, kterou jsem do velké míry eliminoval návrhem a SW úpravami. Mezi další úskalí patřilo nekomfortní Arduino IDE, které jsem nakonec vyměnil za PlatformIO a neočekávané problémy s pamětí. Kvůli vyčerpání téměř celé SRAM u ATmega32U4 nechtělo Yún komunikovat přes sériovou linku a ne-

bylo možné nahrát nový program. Tento problém se mi podařilo překonat až nahráním programu přes WiFi.

Myslím, že vytvořený systém **splnil očekávání** a může být skvělým **základem pro další vývoj** bezpečnostního systému garáže. Jedna z cest, kterou se lze ubírat, je IoT. Systém by mohl například odesílat data do cloudu nebo upozorňovat uživatele na e-mail. Další možností pro rozvoj je tvorba přístupového systému, přidání kamery nebo GSM modulu. Z mého pohledu se jako nejprínosnější úprava nabízí přidání kamery a pořízení kvalitnějších PIR čidel.

Práce pro mne byla obrovským přínosem a umožnila mi vyzkoušet si vlastní návrh komplexnějšího systému. Příležitost spojit dohromady mnoho oborových znalostí a vše si prakticky vyzkoušet je pro mě cennou zkušeností.

---

## Literatura

- [1] *GSM alarm* [online]. 2017 [cit. 2017-10-8]. Dostupné z: <https://www.jabloshop.cz/eye-02-gsm-bezpecnostni-kamera>
- [2] *Jablotron - alarm do domu* [online]. 2017 [cit. 2017-10-8]. Dostupné z: <https://www.jablotron.com/cz/produkty/alarmy/alarm-do-domu/>
- [3] BLUM, Jeremy. *Exploring Arduino: tools and techniques for engineering wizardry* [online]. Indianapolis, IN: Wiley, 2013 [cit. 2018-03-20]. ISBN 978-111-8549-360. Dostupné z: <https://ebookcentral.proquest.com/lib/cvut/detail.action?docID=1318682>
- [4] *Vývojová deska Arduino Yún* [online]. 2018 [cit. 2018-3-9]. Dostupné z: [https://www.arduino.cc/en/Main/ArduinoBoardYun?utm\\_source=platformio&utm\\_medium=piohome](https://www.arduino.cc/en/Main/ArduinoBoardYun?utm_source=platformio&utm_medium=piohome)
- [5] *Proudové limity* [online]. 2018 [cit. 2018-3-10]. Dostupné z: <https://electronics.stackexchange.com/questions/67092/how-much-current-can-i-draw-from-the-arduinios-pins>
- [6] *Bootloader Development* [online]. 2018 [cit. 2018-3-9]. Dostupné z: <https://www.arduino.cc/en/Hacking/Bootloader>
- [7] *Build Process* [online]. 2018 [cit. 2018-3-9]. Dostupné z: <https://github.com/arduino/Arduino/wiki/Build-Process>
- [8] HLAVIČKA, Jan a kolektiv. *Číslicové systémy odolné proti poruchám*. Vydavatelství ČVUT Praha, 1992. ISBN 80-010-0852-5.
- [9] *Arduino Uno WiFi* [online]. 2017 [cit. 2017-09-20]. Dostupné z: <https://store.arduino.cc/arduino-uno-wifi>
- [10] *Arduino boards* [online]. 2017 [cit. 2017-09-30]. Dostupné z: <https://store.arduino.cc/arduino-genuino/arduino-genuino-boards-modules?dir=desc&order=price>

- [11] *Arduino Tian* [online]. 2017 [cit. 2017-09-16]. Dostupné z: <https://store.arduino.cc/arduino-tian>
- [12] *Arduino Ethernet Rev3 without PoE* [online]. 2017 [cit. 2017-09-24]. Dostupné z: <https://store.arduino.cc/arduino-ethernet-rev3-without-poe>
- [13] *Arduino Ethernet Rev3 with PoE* [online]. 2017 [cit. 2017-10-21]. Dostupné z: <https://store.arduino.cc/arduino-ethernet-rev3-with-poe>
- [14] *Arduino Yún* [online]. 2017 [cit. 2017-09-20]. Dostupné z: <https://store.arduino.cc/arduino-yun>
- [15] *Arduino Primo* [online]. 2017 [cit. 2017-10-21]. Dostupné z: <https://store.arduino.cc/arduino-primo>
- [16] FRADEN, Jacob. *Handbook of modern sensors: physics, designs, and applications*. 3rd ed. New York: Springer, ©2004. ISBN 03-870-0750-4.
- [17] VÁŇA, Miroslav. *Detektor barev*. Česká Lípa, 2015. Maturitní práce. Střední průmyslová škola. Vedoucí práce Tomáš Racháč.
- [18] HARTH, Petr. *Laboratorní úloha elektronického zabezpečovacího systému komunikujícího přes síť ethernet*. Brno, 2010. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, ústav telekomunikací. Vedoucí práce Karel Burda.
- [19] KORTÁN, Jaroslav. *Elektronický zabezpečovací systém rodinného domu*. Praha, 2013. Diplomová práce. České vysoké učení technické v Praze, Fakulta elektrotechnická, katedra mikroelektroniky. Vedoucí práce Miroslav Husák.
- [20] *Doppler Effect* [online]. 2018 [cit. 2018-3-10]. Dostupné z: <https://www.grc.nasa.gov/www/k-12/airplane/doppler.html>
- [21] *MQ Gas sensors* [online]. 2017 [cit. 2017-10-9]. Dostupné z: <https://playground.arduino.cc/Main/MQGasSensors>
- [22] *Magnetické bezpečnostní spínače pro dveře a kryty* [online]. 2017 [cit. 2017-10-24]. Dostupné z: <https://automatizace.hw.cz//bezpecnost-stroju-komponenty/magneticke-bezpecnostni-spinace-pro-dvere-a-kryty.html>
- [23] *HC-SR501 Datasheet* [online]. 2017 [cit. 2017-10-8]. Dostupné z: <http://www.datasheetpdf.com/pdf/775434/ETC/HC-SR501/1>
- [24] *Top 50 Arduino sensors by sales* [online]. 2018 [cit. 2018-03-31]. Dostupné z: [https://www.matthewb.id.au/IoT/top\\_50\\_Arduino\\_sensors.html](https://www.matthewb.id.au/IoT/top_50_Arduino_sensors.html)

- 
- [25] *Sparkfun PIR datasheet* [online]. 2018 [cit. 2018-3-11]. Dostupné z: <https://www.sparkfun.com/products/13285>
- [26] *HC-SR505 Mini PIR Motion Sensor* [online]. 2017 [cit. 2017-10-23]. Dostupné z: <https://www.elecrow.com/hcsr505-mini-pir-motion-sensor-p-1382.html>
- [27] *Me PIR Motion Sensor* [online]. 2017 [cit. 2017-10-8]. Dostupné z: <http://store.makeblock.com/me-pir-motion-sensor>
- [28] *Thermo-Technik PIR datasheet* [online]. 2018 [cit. 2018-3-11]. Dostupné z: <http://www.produktinfo.conrad.com/index.php?bl=CS&q=172500>
- [29] *MQ2 Gas sensor* [online]. 2017 [cit. 2017-10-10]. Dostupné z: [http://wiki.seeed.cc/Grove-Gas\\_Sensor-MQ2/](http://wiki.seeed.cc/Grove-Gas_Sensor-MQ2/)
- [30] *Magnetický spínač MC-38* [online]. 2017 [cit. 2017-10-26]. Dostupné z: <http://www.cybronyx.com/jual-MC-38-Door-Window-Sensor-Magnetic-Switch-Home-Alarm-System.html>
- [31] *Magnetický spínač AL28* [online]. 2017 [cit. 2017-10-25]. Dostupné z: <http://www.growzone.cz/Magneticke-cidlo-na-okna-nebo-dvere-N0-d3702.htm>
- [32] *Magnetický spínač ALF-MC04* [online]. 2017 [cit. 2017-10-25]. Dostupné z: <https://eletur.cz/magneticke-kontakty-masivni-povrchovy-kovovy>
- [33] *Magnetický spínač SA-211* [online]. 2018 [cit. 2018-3-11]. Dostupné z: <https://www.jabloshop.cz/sa-211-det-zavrt-miniaturni-s-kabelem-extra-melky>
- [34] *Detect disconnected pin* [online]. 2018 [cit. 2018-3-13]. Dostupné z: <https://arduino.stackexchange.com/questions/14647/how-can-i-detect-a-disconnected-pin>
- [35] *Arduino Yún with POE* [online]. 2018 [cit. 2018-03-13]. Dostupné z: <https://store.arduino.cc/arduino-yun-with-poe>
- [36] ALBERCA, Carlos. *Security analysis and exploitation of arduino devices in the internet of things* [online]. New York, USA: ACM Press, 2016. 437-442 [cit. 2018-03-25]. ISBN 9781450341288. Dostupné z: <http://dl.acm.org/citation.cfm?doid=2903150.2911708>
- [37] *STRIDE Threat Model* [online]. 2018 [cit. 2018-03-13]. Dostupné z: [https://msdn.microsoft.com/en-us/library/ee823878\(v=cs.20\).aspx](https://msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx)

## LITERATURA

---

- [38] ISO/IEC 27005:2008. *Information technology – Security techniques – Information security risk management*. 1. 2008.
- [39] *Securing communications with HTTPS on the Arduino Yun* [online]. 2018 [cit. 2018-03-19]. Dostupné z: [http://hypernephelist.com/2014/08/19/https\\_on\\_arduino\\_yun.html](http://hypernephelist.com/2014/08/19/https_on_arduino_yun.html)
- [40] MASSÉ, Mark. *REST API design rulebook*. Sebastopol, CA: O'Reilly, 2012. ISBN 978-144-9310-509.
- [41] *Camera Chip OV7670 datasheet* [online]. 2018 [cit. 2018-05-13]. Dostupné z: <https://www.voti.nl/docs/OV7670.pdf>

---

# Instalační příručka

Tato příručka popisuje dvě jednoduché varianty instalace. První z nich je naprosto platformě nezávislá, druhá podporuje většinu běžně používaných OS. Předpokládá se, že Arduino Yun je ve výchozím nastavení. Pro restartování WiFi konfigurace lze využít tlačítka na vývojové desce. Instalační příručka nepokrývá pokročilejší síťová nastavení vedoucí k vyšší bezpečnosti. To z důvodu vyšší složitosti a neuniverzálnosti.

## Příprava:

1. Zapojte Arduino Yún a komponenty dle schéma zapojení z části 3.6.
2. Připojte Yún k napájení pomocí microUSB kabelu.

## A) Nahrání HEX souboru

Nahrání HEX souboru je možné provést na libovolné platformě přes webové rozhraní.

1. Proveďte část příprava.
2. Připojte se z počítače na WiFi hotspot který Arduino vytvoří.
3. Zkontrolujte zda máte na Vašem Arduinu Yún nahranou poslední stabilní verzi OpenWRT (verze 1.5.3 k 5.4.2018).
  - Kontrolu lze provést ssh připojením (`ssh root@192.168.240.1`) a provedením příkazu `cat /etc/arduino/openwrt-yun-release`. Pokud soubor neexistuje, používáte Lilino OS místo OpenWRT. Pokud existuje proveďte příkaz `cat /etc/banner` nebo `uname -r`.
  - Instalaci OpenWRT lze provést pomocí tohoto návodu: <https://www.arduino.cc/en/Tutorial/YunSysupgrade>.

4. Otevřete webový prohlížeč a do adresního řádku **zadejte „yun.local“ nebo IP adresu** WiFi/Ethernet rozhraní. Defaultní IP adresa WiFi rozhraní je **192.168.240.1**. Detailnější informace o nalezení nebo nastavení IP adresy jsou na stránce <https://www.arduino.cc/en/Guide/ArduinoYun#toc16>.
5. Zadejte heslo. Ve výchozím nastavení je to „arduino“.
6. Nahrajte HEX soubor umístěný na médiu – `medium:\impl\Main.hex`.
7. Ověřte funkčnost pomocí sériové linky (Arduino IDE – Serial Monitor) nebo zadáním `192.168.240.1/arduino/debug` do adresního řádku prohlížeče. Při využití REST rozhraní je nutno počkat 3 minuty od nahrání programu nebo zapnutí.

## B) Instalace pomocí Arduino IDE

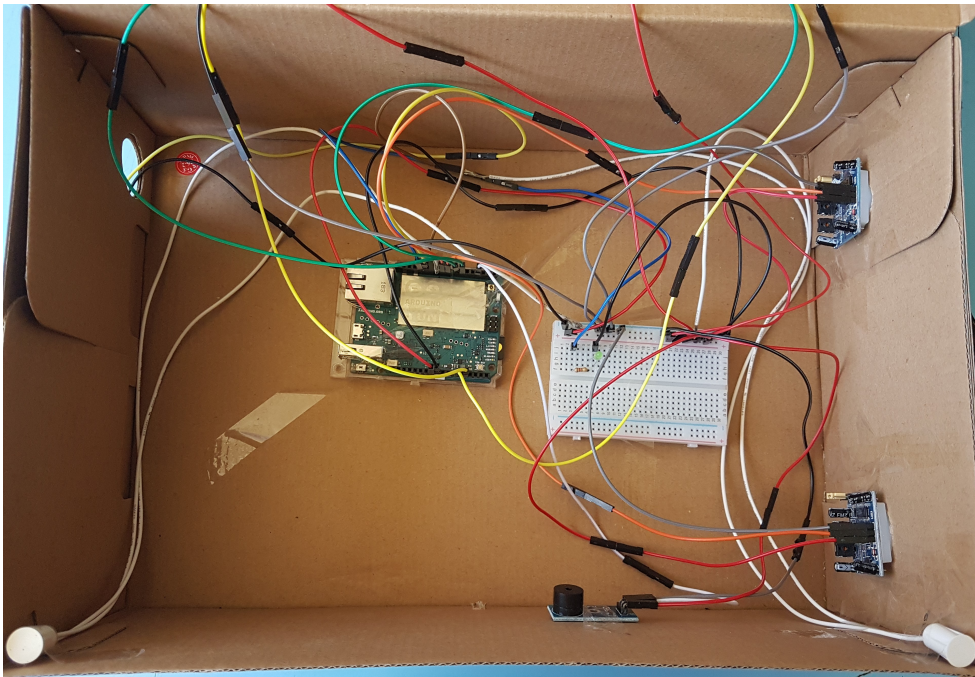
1. Proveďte část příprava.
2. Nainstalujte Arduino IDE z oficiálních stránek <https://www.arduino.cc/en/Guide/HomePage>.
3. Nakonfigurujte Arduino IDE a ověřte funkčnost komunikace s Yún nahráním testovacího programu Blink. Viz návod rychlý start <https://www.arduino.cc/en/Guide/ArduinoYun#toc1>.
4. V Arduino IDE otevřete `medium:\impl\Main\Main.ino`. Zdrojový kód je také dostupný z repozitáře<sup>14</sup>.
5. V Arduino IDE nainportujte knihovny obsažené v souboru `libraries.zip`. Cesta: Sketch, Include Library, Add .ZIP Library, vyberte z média soubor `medium:\impl\libraries.zip`.
6. Proveďte upload a otevřením serial monitoru ověřte funkčnost (první tři minuty probíhají inicializační výpisy).

---

<sup>14</sup><https://github.com/vanamir2/YunGarageSystem>



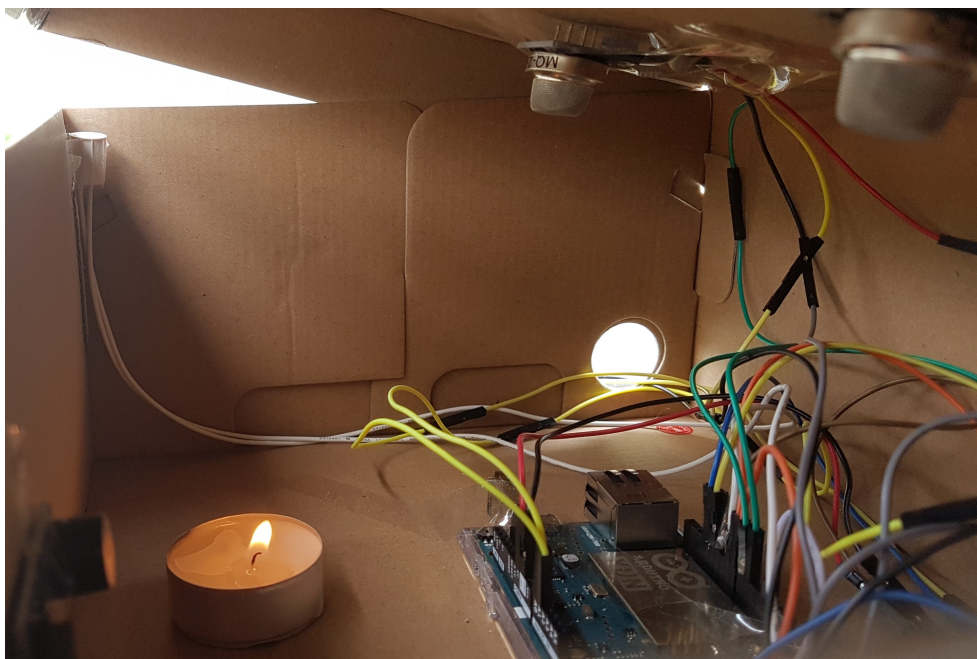
## Fotografie systému



Obrázek B.1: Fotografie testovacího zapojení v krabici. Na pravém boku jsou PIR. Dole jsou bílé magnetické spínače a mezi nimi bzučák. Nechybí samozřejmě Arduino Yún a nepájivé pole.

## B. FOTOGRAFIE SYSTÉMU

---



Obrázek B.2: Detail senzorů kouře a plynu při testování protipožární ochrany.



Obrázek B.3: Detail Arduina Yún.





Obrázek B.4: PIR monitorující okolní prostor. Vlevo si lze všimnout mírně vyčnívajícího tlačítka pro zamykání a odemykání systému.



Obrázek B.5: Detail tlačítka



Obrázek B.6: Detail PIR



---

## Členění kódu

Jak již bylo zmíněno v podkapitole 3.7, zabývající se implementací, zdrojový kód je členěn do šesti souborů. Velikost paměti je u Arduina často omezujícím faktorem, především u komplexnějších projektů. Z tohoto důvodu není zdrojový kód strukturován do jednotlivých tříd a tím pádem nelze sestrojít diagram tříd. Popis jednotlivých souborů je uveden v této příloze, konkrétní metody jsou komentovány přímo ve zdrojovém kódu.

### **Main.ino**

Výchozí soubor ve kterém jsou definovány základní metody *setup()* a *loop()*, které jsou symbolem pro Arduino platformu. Hlavním účelem Main.ino je importovat, nastavit a inicializovat celý systém, aby byl připraven ke spuštění. Po inicializaci systému je řízení předáno do metody *loop()*, která využívá metod definovaných v dalších souborech – *SystemManager.ino* a *REST.ino*.

### **REST.ino**

Jak již název napovídá, v tomto souboru jsou definovány metody poskytující REST API. Kromě zpracování požadavků je zde definován také komunikační protokol z podkapitoly 3.5.

### **SystemManager.ino**

V tomto souboru je implementován stavový automat odpovídající návrhu z podkapitoly 3.2 – Logika systému.

### **Utility.ino**

Utility.ino obsahuje podpůrné metody a makra zaměřující se především na několik vybraných kategorií.

- Snadnou práci s časem

- Práci s tlačítkem (push-button)
- Výpis dat do řetězců
- Debug metody

### **sensors.h**

Veškeré metody pro práci se senzory (např. čtení dat) jsou definovány v tomto souboru. Jedná se o header (.h) soubor, protože soubory s koncovkou .ino poskytují pouze vizuální oddělení. Ve skutečnosti jsou totiž všechny soubory s koncovkou .ino sloučeny do jednoho velkého .ino souboru, ze kterého se provádí spouštění programu (v tomto případě Main.ino).

Pro odstranění potíží se závislostí jsou metody pracující se senzory umístěny do souboru s koncovkou .h. Díky tomu je možné importovat sensors.h na libovolné místo.

### **constants.h**

Soubor určený výhradně pro konstanty a výčtový typ (enum). Jsou zde definovány například piny senzorů, některé řetězce nebo časové konstanty.

## Seznam použitých zkratek

- ADC** Analog-to-Digital Converter
- AES** Advanced Encryption Standard
- API** Application Programming Interface
- ARM** Advanced RISC Machine
- ASLR** Address Space Layout Randomization
- BLE** Bluetooth Low Energy
- ČR** Česká Republika
- DoS** Denial of Service
- DPH** Daň z Přidané Hodnoty
- DPS** Deska Plošných Spojů
- eMMC** embedded MultiMediaCard
- GUI** Graphical User Interface
- GND** Ground
- GSM** Global System for Mobile communications
- HTML** HyperText Markup Language
- HW** Hardware
- I/O** Input/Output
- I<sup>2</sup>C** Inter-Integrated Circuit
- IDE** Integrated Development Enviroment

**IoT** Internet of Things

**IP** Internet Protocol

**IP67** International Protection marking

**IT** Information Technology

**IR** Infrared Radiation

**LPG** Liquefied Petroleum Gas

**MCU** MicroController Unit

**MIPS** Microprocessor without Interlocked Pipeline Stages

**MS** Magnetic Switch

**NC** Normaly Closed

**NFC** Near Field Communication

**NO** Normaly Opened

**OS** Operační Systém

**PIR** Passive InfraRed

**PoE** Power over Ethernet

**PWM** Pulse-Width Modulation

**REST** Representational State Transfer

**RJ** Registered Jack

**S&G** Smoke and Gas

**SMS** Short Message Service

**SPI** Serial Peripheral Interface bus

**SRAM** Static Random Access Memory

**SSH** Secure Shell

**SW** Software

**UPS** Uninterruptible Power Supply

**USB** Universal Serial Bus



---

## Obsah přiloženého média

	readme.txt	.....	stručný popis obsahu média
	impl	.....	zdrojové kódy implementace
	thesis	.....	práce a její zdroje
		thesis.pdf	..... text práce ve formátu PDF
		thesis.tex	..... zdrojová forma práce – $\text{\LaTeX}$
		images	..... obrázky práce
		attatchement	..... přílohy práce