

Využití osobního počítače jako ústředny poplachového zabezpečovacího systému

Using Personal Computer as an Intruder Alarm System Control
Unit

Bc. Adam Hanáček

Diplomová práce
2012



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2011/2012

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Adam HANÁČEK**
Osobní číslo: **A10312**
Studijní program: **N 3902 Inženýrská informatika**
Studijní obor: **Bezpečnostní technologie, systémy a management**

Téma práce: **Využití osobního počítače jako ústředny
poplachového zabezpečovacího systému**

Zásady pro vypracování:

1. Zpracujte literární rešerši na dané téma.
2. Navrhněte a realizujte vhodný způsob propojení drátových detektorů typu NO a NC s počítačem.
3. Vytvořte programové vybavení ve zvoleném programovacím jazyku, které bude zajišťovat funkce poplachové ústředny.
4. Ověřte funkci vytvořeného systému.
5. Zhodnotte vytvořené řešení, jeho výhody a nevýhody, a navrhněte další možnosti jeho zdokonalení.

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. PINKER, Jiří. Mikroprocesory a mikropočítače. 1. vyd. Praha: BEN – technická literatura, 2004, 159 s. ISBN 80-730-0110-1.
2. MANN, Burkhard. C pro mikrokontroléry: ANSI-C, kompilátory C, spojovací programy – linkery, práce s ATMELE AVR a MSC-51, příklady programování v jazyce C, nástroje pro programování, tipy a triky. Vyd. 1. Praha: BEN, 2003, 279 s. ISBN 80-730-0077-6.
3. CATSOULIS, John. Designing embedded hardware. 2nd ed. Sebastopol: O'Reilly, 2005, 377 s. ISBN 05-960-0755.
4. MATOUŠEK, David. Číslicová technika: základy konstruktérské praxe. 1. vyd. Praha: BEN – technická literatura, 2001, 207 s. ISBN 80-730-0025-3.
5. MARTINEK, Radislav. Senzory v průmyslové praxi. 1. vyd. Praha: BEN – technická literatura, 2008, 199 s. ISBN 80-730-0114-4.
6. NAGEL, Christian. C 2008: programujeme profesionálně. Vyd. 1. Brno: Computer Press, 2009, 1126 s. ISBN 978-802-5124-017.
7. ARDUINO. Arduino Home Page [online]. 2011 [cit. 2012-01-17]. Dostupné z: <http://www.arduino.cc>
8. C Programming Guide. MICROSOFT. Microsoft MSDN [online]. 2012 [cit. 2012-01-17]. Dostupné z: [http://msdn.microsoft.com/en-us/library/67ef8sbd\(v=vs.80\).aspx](http://msdn.microsoft.com/en-us/library/67ef8sbd(v=vs.80).aspx)

Vedoucí diplomové práce:

Ing. Jan Dolinay, Ph.D.

Ústav automatizace a řídicí techniky

Datum zadání diplomové práce:

24. února 2012

Termín odevzdání diplomové práce:

15. května 2012

Ve Zlíně dne 24. února 2012

prof. Ing. Vladimír Vašek, CSc.
děkan

L.S.

doc. RNDr. Vojtěch Křesálek, CSc.
ředitel ústavu

ABSTRAKT

Teoretická část diplomové práce je zaměřena na poplachové zabezpečovací a tísňové systémy. Hlavní důraz je kladen na možné způsoby připojení detektorů k ústředně, popis jednotlivých částí a požadavky norem na poplachové zabezpečovací a tísňové systémy. V praktické části diplomové práce je vytvořen poplachový systém, který poskytuje možnost využít počítač nebo notebook jako prostředek pro přenos poplachového signálu, programování, ovládání a dohlížení nad systémem, čímž lze snížit cenu zabezpečovacích systémů. Další výhodou navrženého systému spočívá v možnosti připojení nejen detektorů a tísňových zařízení, ale také velkého množství požárních hlásičů.

Klíčová slova: Poplachový zabezpečovací a tísňový systém, C#, ústředna, normy

ABSTRACT

The theoretical part of diploma thesis is focused on Intruder and Hold-up Alarm Systems. The main emphasis is placed on possible ways of connecting the detectors to the central, the description of each parts and requirements of standards on Intruder and Hold-up Alarm Systems. The practical part of diploma thesis is focused on development of Alarm System, which provides the possibility to use a personal computer or laptop as a tool for programming, control and supervision the System, but also for transmission of an alarm signal, what helps to reduce the cost of the Security System. Next advantage of designed system lies in the possibility of connection not only detectors and emergency facilities, but also in connection of large number of fire alarms.

Keywords: Intruder and Hold-up Alarm System , C#, Central, Standards

Děkuji vedoucímu mé diplomové práce panu Ing. Janu Dolinayovi, Ph.D. za rady, trvalý zájem a čas věnovaný mé práci.

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD	10
I TEORETICKÁ ČÁST	11
1 POPLACHOVÉ ZABEZPEČOVACÍ A TÍŠŇOVÉ SYSTÉMY	12
1.1 ÚVOD DO POPLACHOVÝCH ZABEZPEČOVACÍCH A TÍŠŇOVÝCH SYSTÉMŮ	12
1.1.1 Poplachový zabezpečovací systém.....	12
1.1.2 Poplachový tísňový systém.....	13
1.1.3 Poplachový zabezpečovací a tísňový systém.....	13
1.2 ZÁKLADNÍ PRVKY POPLACHOVÉHO ZABEZPEČOVACÍHO A TÍŠŇOVÉHO SYSTÉMU.....	13
1.3 BLOKOVÉ SCHÉMA PZTS.....	14
1.4 HLAVNÍ ČÁSTI PZTS	14
1.4.1 Ústředna	14
1.4.1.1 Kabelové (drátové) ústředny	15
1.4.1.2 Bezdrátové ústředny	16
1.4.1.3 Hybridní ústředny.....	16
1.4.2 Tísňová zařízení a detektory	17
1.4.3 Indikační a ovládací zařízení.....	18
1.4.4 Signalizační a ostatní spínaná zařízení	18
1.4.5 Příjem poplachového signálu a poplachový přenosový systém	18
1.5 NEJPOUŽÍVANĚJŠÍ ZPŮSOBY PROPOJENÍ TÍŠŇOVÝCH ZAŘÍZENÍ S ÚSTŘEDNOU PZTS.....	19
1.5.1 Zapojení NC (normally closed – v klidu uzavřeno) se zakončovacím odporem	20
1.5.2 Zapojení NC (normally closed) bez zakončovacího odporu.....	21
1.5.3 Zapojení NO (normally opened – v klidu otevřeno) se zakončovacím odporem	21
1.5.4 Zapojení NO bez zakončovacího odporu.....	22
1.5.5 Kombinace NC (v klidu uzavřeno) a NO (v klidu otevřeno) se zakončovacím odporem	23
1.6 NEJPOUŽÍVANĚJŠÍ ZPŮSOBY PROPOJENÍ DETEKTORŮ S ÚSTŘEDNOU PZTS.....	24
1.6.1 Zapojení typu NC dvojitě vyvážená	24
1.6.2 Zapojení typu NC trojitě vyvážená.....	26
1.6.3 Zapojení typu NC zdvojení zóny.....	27
2 POŽADAVKY NOREM NA POPLACHOVÉ ZABEZPEČOVACÍ A TÍŠŇOVÉ SYSTÉMY	28
2.1 VYBRANÉ ZKRATKY POUŽÍVANÉ V POPLACHOVÝCH ZABEZPEČOVACÍCH SYSTÉMECH.....	28
2.2 STUPNĚ ZABEZPEČENÍ I&HAS PODLE NORMY ČSN. EN 50131-7.....	30
2.3 POŽADAVKY NOREM NA ÚSTŘEDNY	30
3 ZÁKLADY PROGRAMOVÁNÍ V C#.....	34
3.1 HODNOTOVÉ A ODKAZOVÉ DATOVÉ TYPY	34
3.1.1 Proměnná	34
3.1.2 Hodnotové datové typy.....	34
3.1.2.1 Jednoduchý příklad použití hodnotového datového typu	35

3.1.3	Odkazové datové typy	36
3.1.4	Rozdíl mezi třídou a strukturou.....	36
3.1.5	Vysvětlení pojmů public a static	37
3.1.6	Použití private, protected, internal u tříd a struktur.....	37
3.2	DELEGÁTI	37
3.3	ZÁKLADY VÍCEVLÁKNOVÉHO PROGRAMOVÁNÍ.....	38
3.3.1	Příklad jednoduchého vícevláknového programu	39
3.3.2	Synchronně pracující vlákna	40
3.3.3	Asynchronně pracující vlákna	40
II	PRAKTICKÁ ČÁST	41
4	ÚVOD A ZÁKLADNÍ STRUKTURA VYTVOŘENÉHO POPLACHOVÉHO ZABEZPEČOVACÍHO SYSTÉMU	42
4.1	ÚVOD	42
4.2	CELKOVÁ STRUKTURA FUNGOVÁNÍ.....	43
4.2.1	Základní popis programu v Arduinu	43
4.2.1.1	Struktura paměti EEPROM.....	43
4.2.1.2	Hlavní část programu.....	45
4.2.2	Komunikační protokol v nezastřeženém stavu.....	46
4.2.2.1	Požadavek	46
4.2.2.2	Příjem a zpracování	47
4.2.2.3	Odpověď	47
4.2.2.4	Příjem odpovědi	47
4.2.2.5	Požadavky a odpovědi	48
4.2.3	Komunikační protokol v zastřeženém stavu	49
4.2.4	Způsob připojení detektorů	51
5	MANUÁL PRO PROGRAMOVÁNÍ	52
5.1	NABÍDKA “HLAVNÍ“	53
5.2	NABÍDKA “ZASTŘEŽIT“	54
5.3	NABÍDKA “KOMUNIKACE“	55
5.4	NABÍDKA “PROGRAMOVÁNÍ VSTUPŮ A VÝSTUPŮ“	55
5.4.1	Programování analogových vstupů	56
5.4.1.1	Nastavení časové aktivace u analogového vstupu.....	57
5.4.2	Programování digitálních vstupů/výstupů	58
5.4.2.1	Nastavení časové aktivace u digitálních vstupů/výstupů.....	59
5.4.3	Časová aktivace	59
5.5	NABÍDKA “PROGRAMOVÁNÍ ZÁKLADNÍHO NASTAVENÍ“	60
5.5.1	Programování základního nastavení.....	60
5.5.2	Změna hesla pro přístup k paměti, e-mailu a hesla k e-mailu.....	61
5.5.3	Změna hesel pro zastřežení a odstřežení.....	61
5.5.3.1	Přiřazení výstupů k daným heslům.....	62
5.5.4	Přehledný náhled nastavení.....	62
5.6	NABÍDKA “NASTAVENÍ ČASU“	63
5.7	POSTUP PŘI PROGRAMOVÁNÍ SUBSYSTÉMU	63
5.7.1	Připojení Arduina	63
5.7.2	Nastavení výstupů	63
5.7.3	Nastavení vstupů	63

5.7.4	Uložit program do Arduina	64
5.7.5	Přidat vstupy a výstupy k danému heslu.....	64
5.8	PRAKTICKÝ PŘÍKLAD PŘIPOJENÍ POPLACHOVÉHO DETEKTORU K NAVRŽENÉMU SYSTÉMU	64
6	DALŠÍ MOŽNOSTI ZDOKONALENÍ A POROVNÁNÍ S BĚŽNÝM POPLACHOVÝM SYSTÉMEM.....	66
6.1	MOŽNÉ ZPŮSOBY ZASTŘEŽENÍ NEBO ODSTŘEŽENÍ SYSTÉMU	66
6.1.1	Využití klávesnice u počítače.....	66
6.1.2	Využití klávesnice připojené k prodlužovacímu kabelu u počítače	67
6.1.3	Využití zařízení připojeného ke vstupu Arduina.....	67
6.2	FINANČNÍ OHODNOCENÍ.....	67
6.2.1	Finanční náklady základních prvků při využití poplachového zabezpečovacího systému popsaného v praktické části.....	68
6.2.2	Finanční náklady základních prvků při použití běžných zabezpečovacích systémů a cenové porovnání s navrhnutým systémem.....	68
6.3	VÝHODY A NEVÝHODY NAVRŽENÉHO SYSTÉMU V POROVNÁNÍ SE SOUČASNÝMI POPLACHOVÝMI ZABEZPEČOVACÍMI SYSTÉMY	70
	ZÁVĚR	71
	CONCLUSION.....	73
	SEZNAM POUŽITÉ LITERATURY	75
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	77
	SEZNAM OBRÁZKŮ	78
	SEZNAM TABULEK	80
	SEZNAM PŘÍLOH	81

ÚVOD

V současné době roste nutnost chránit majetek, život a zdraví. K tomu slouží ochrana fyzická, režimová, ochrana mechanickými zábrannými prostředky, ale také ochrana poplachovým zabezpečovacím a tísňovým systémem, který chrání tím způsobem, že vyšle informaci o narušení objektu zpravidla na poplachové přijímací centrum nebo mobilní telefon. Poplachový zabezpečovací a tísňový systém (dále jen PZTS) bývá složen z detektorů, tísňových zařízení, ústředny, poplachové přenosové trasy, signalizačního zařízení, ovládacího zařízení a zařízení pro příjem poplachového signálu a napájení. Jednotlivé prvky PZTS jsou popsány v teoretické části diplomové práce.

Při volbě PZTS je možné pro přenos informace mezi prvky použít systém bezdrátový nebo kabelový (drátový). Drátový přenos se pak dělí na sběrníkový nebo analogový. Výhodou drátového propojení je výrazně nižší riziko falešných poplachů, ovšem nevýhodou je nutnost instalace kabeláže. Jestliže je při volbě poplachového zabezpečovacího systému hlavní prioritou maximální bezpečnost přenosu a minimální riziko sabotáže, pak je vhodné zvolit sběrníkový systém, který je ovšem nejdražší. Pokud je však prioritou jednoduchost montáže, pak je vhodné zvolit bezdrátový systém, a v případě upřednostnění ceny, je vhodné zvolit analogový drátový systém.

Na možné způsoby snížení ceny zabezpečovacích systémů je zaměřena praktická část diplomové práce, kde je navržen systém, který využívá osobní počítač nebo notebook nejen pro programování systému, ale také pro zastřežení či odstřežení, kontrolu funkčnosti systému a pro vyvolání poplachu pomocí internetu. Tím se sníží cena nejen za poplachovou ústřednu, ale také za klávesnici a poplachovou přenosovou cestu. K systému je dále možné připojit také požární hlásiče a tísňová zařízení. Díky velkému množství zón lze jeho prostřednictvím ovládat i nepoplachové aplikace.

Při vývoji systému byl kladen důraz na variabilitu, jednoduchost ovládání, ale především na maximální ochranu proti jakékoli sabotáži.

I. TEORETICKÁ ČÁST

1 POPLACHOVÉ ZABEZPEČOVACÍ A TÍŠŇOVÉ SYSTÉMY

Kapitola je zaměřena na celkové pochopení poplachových zabezpečovacích a tísňových systémů a také na popis jednotlivých prvků, které PZTS obsahuje.

1.1 Úvod do poplachových zabezpečovacích a tísňových systémů

V terminologii se rozlišují dva základní typy ochran a to ochrana poplachovým zabezpečovacím systémem a poplachovým tísňovým systémem.

1.1.1 Poplachový zabezpečovací systém

Úkolem poplachového zabezpečovacího systému je ochrana života, zdraví a majetku osob, a to díky střežení zabezpečených prostorů. Jedná se tedy o soubor zařízení sloužících pro detekci narušení daného prostoru. Do PZS patří například magnetické kontakty (kde zdrojem magnetického pole zpravidla bývá magnet, který je umístěn na snímaném předmětu [1]), detektory, ústředna, ovládací zařízení, poplachový přenosový systém, signalizační zařízení, tísňové hlásiče, zapisovací zařízení nebo zamlžovací systém. Při detekci narušení chráněného prostoru vyšle detektor informaci o narušení ústředně, která aktivuje signalizační zařízení (optickou nebo akustickou signalizaci), zapisovací zařízení, zamlžovací systém nebo jakékoli jiné spínané zařízení. Především ovšem vyšle ústředna informaci o narušení na předem definované místo s pomocí poplachového přenosového systému. V současné době si může majitel střeženého objektu vybrat, jestli chce, aby informace přišla přímo jemu (například na mobilní telefon anebo pager) nebo využije možnost připojení na poplachové přijímací centrum. Je vhodné využít poplachové přijímací centrum, protože zde jsou k dispozici proškolení pracovníci, kteří přesně vědí, jak v případě narušení objektu postupovat, a jsou kdykoli připraveni k výjezdu. Nevýhodou této varianty je cena, kdy měsíční střežení stojí od 600Kč. V případě výjezdu se pak platí za každý výjezd minimálně 500Kč. K ceně se zpravidla připočítává poplatek, jehož výše závisí na lokalitě a dojezdové vzdálenosti. Dále si může zákazník připlatit za vyslání informace o narušení objektu, kontrolu doby zapnutí PZTS, za hlídání funkce odblokování pod nátlakem, případně i za výpisy poštou nebo e-mailem. Potom zde bývá ještě poplatek za využívání poplachového přenosového systému. Problémem ovšem je, že jen cena za střežení, výjezd zásahové služby a využívání poplachového přenosového systému je pro většinu lidí moc vysoká, proto si raději nechávají posílat poplachový signál zpravidla na pager nebo mobilní telefon. Jakmile ale dojde k narušení objektu, musí vzniklou situaci

řešit sami majitelé. Při zvolení zmíněného způsobu zabezpečení by si měl ovšem každý majitel uvědomit, že při narušení je třeba reagovat co nejdříve, což zase pro většinu majitelů není vždy možné.

Vhodnými doplňky poplachového zabezpečovacího systému jsou mechanické zábranné systémy, kamerové systémy, tísňové systémy a přístupové systémy.

1.1.2 Poplachový tísňový systém

Hlavní použití poplachových tísňových systémů spočívá v detekci přepadení. Při přepadení obsluha, přepadená osoba nebo svědek přepadení aktivuje tísňové zařízení, s jehož pomocí se informace o přepadení předá na příslušné místo. Tyto systémy se nejčastěji používají v bankách, obchodních domech, prodejnách a na benzinových pumpách. Podrobnější popis PTS je v kapitole 1.4.2.

1.1.3 Poplachový zabezpečovací a tísňový systém

Poplachový zabezpečovací a tísňový systém je spojení poplachového zabezpečovacího systému a poplachového tísňového systému. Systém tedy informuje nejen o narušení objektu, ale součástí je i detekce přepadení. PZTS je vhodné doplnit režimovým opatřením, fyzickou ostrahou a vhodnými mechanickými zábrannými prostředky.

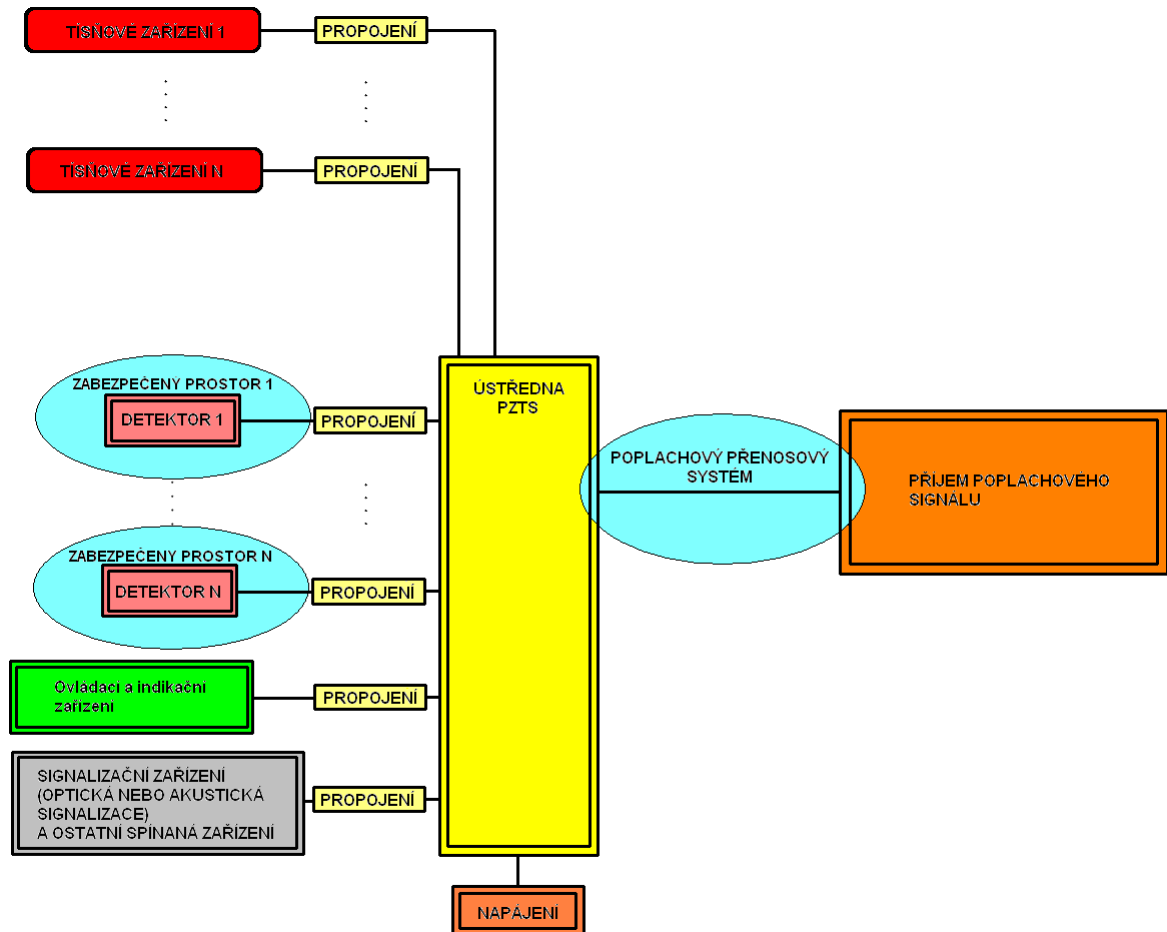
1.2 Základní prvky poplachového zabezpečovacího a tísňového systému

Mezi základní části PZTS patří:

- Ústředna
- Tísňové zařízení a detektory
- Ovládací zařízení
- Signalizační zařízení a ostatní spínaná zařízení
- Poplachový přenosový systém
- Zařízení pro příjem poplachového signálu
- Napájení

1.3 Blokové schéma PZTS

Na obr. 1 jsou znázorněny jednotlivé části PZTS, které budou dále vysvětleny.



Obrázek 1: Schéma PZTS

1.4 Hlavní části PZTS

1.4.1 Ústředna

Ústřednu lze považovat za nejdůležitější prvek poplachového zabezpečovacího systému.

Mezi základní úkoly ústředny patří:

- Příjem poplachových signálů od detektorů a tísňových hlásičů
- Napájení detektorů a tísňových hlásičů, potřebují-li napájet
- Příjem a vyhodnocení signálů z ovládacího zařízení
- Ovládání signalizačních a ostatních spínaných zařízení
- Ukládání základních údajů o poplachu do paměti

- Předání poplachového signálu do poplachového přenosového systému

Základní rozdělení ústředen:

- Kabelové (drátové)
- Bezdrátové
- Hybridní

1.4.1.1 Kabelové (drátové) ústředny

Kabelové ústředny se vyznačují tím, že propojení ústředny s detektory a klávesnicí je provedeno drátově. Nevýhodou je navýšení ceny za kabeláž, výhodou mnohem nižší riziko falešných poplachů.

Dělení:

- Analogové (smyčkové)
- S přímou adresací (sběrnice)
- Smíšené

Analogové ústředny PZTS

Každá analogová ústředna má určitý počet zón, nejčastěji jich je 6, 8 nebo 16 zón. Každá zóna tvoří vyhodnocovací smyčku, která bývá zakončena zakončovacím odporem, přičemž ústředna vyhodnocuje hodnotu odporu dané smyčky. Detektory se mohou na smyčku připojit sériově nebo paralelně v závislosti na použitém způsobu zapojení. Při poplachu pak detektor rozepne kontakt, který je připojen na danou smyčku, tím změní odpor smyčky a ústředna vyhlásí poplach. Teoreticky lze na každou zónu připojit libovolný počet detektorů, ale musí se počítat s tím, že čím víc detektorů bude na jedné smyčce, tím klesá možnost přesného zjištění pozice, kde byl poplach vyvolán. Obvykle se doporučuje připojit maximálně 10 detektorů na jednu zónu. U každé zóny lze použít tzv. zdvojení zóny (ATZ), čímž lze každou zónu rozdělit na dva subsystemy. Podrobné vysvětlení jednotlivých možností zapojení je popsáno v kapitole 1.5 a 1.6.

Hlavní výhodou analogových ústředen je velmi nízká cena za jednotlivé detektory a tísňové zařízení. Mezi nevýhody pak patří nemožnost přesné identifikace detektoru a vyšší náklady na propojení.

Ústředny s přímou adresací (sběrnice)

K propojení ústředny s detektory se používá sběrnice, kterou tvoří čtyři vodiče. Komunikace je datová a je zde možné přesně identifikovat pozici detektoru. Negativní stránkou sběrnice je to, že v každém okamžiku na ni může být připojen jen jeden zdroj dat. Nelze tak současně předávat data ze dvou zdrojů. [2]

Z hlediska bezpečnosti jsou však považovány sběrnice propojení za nejvhodnější. V praxi se ovšem využívá jen velmi zřídka, protože cena celkového zabezpečení je velmi vysoká.

Ústředny smíšené

Smíšené ústředny jsou kombinací sběrnice ústředny a analogové ústředny. K ústředně lze tedy připojit sběrnice detektory a lze také přikoupit koncentrátor, který slouží pro připojení analogových detektorů ke sběrnici.

1.4.1.2 Bezdrátové ústředny

Komunikace mezi ústřednou a jednotlivými prvky probíhá bezdrátově, a to s použitím frekvence buď 868MHz nebo 433 MHz. Komunikace je samozřejmě z důvodu bezpečnosti šifrovaná. Ústředna si periodicky hlídá připojené detektory a v případě, že některý detektor přestane komunikovat, ústředna vyhlásí poplach. Jestliže bude komunikace s určitým detektorem na dané frekvenci přerušena, tak se detektor za určitý čas automaticky přeladí na jinou frekvenci a přihlásí se ústředně na nové frekvenci.

Jednotlivé detektory jsou dražší oproti použití analogového propojení, ale na druhou stranu není třeba platit za kabeláž. Výhodou je i možnost zjištění přesné polohy detektoru a nevýhodou vyšší cena za jednotlivé prvky.

1.4.1.3 Hybridní ústředny

Hybridní ústředny jsou kombinací analogových drátových detektorů a bezdrátových detektorů. Jedná se o nejvariabilnější, ale z cenového hlediska nejdražší řešení.

1.4.2 Tísňová zařízení a detektory

Tísňová zařízení

Jak již bylo naznačeno v kapitole 1.1.2, poplachový tísňový systém slouží pro úmyslné vyvolání poplachu při přepadení (například přepadení banky, obchodu, benzinové pumpy a podobně). Poplachový signál se generuje pomocí tísňového zařízení a k aktivování může dojít buď kontaktně nebo bezkontaktně. Do kontaktních se řadí veřejné tísňové hlásiče, speciální tísňové hlásiče a kontaktní bankovkové detektory. Veřejné tísňové hlásiče se montují na viditelná místa a jsou určeny pro fyzickou ostrahu objektu při obchůzkové službě. Speciální tísňové hlásiče se používají pro skryté vyvolání poplachu, přičemž musí být skryté pro cizí osoby, ale dostatečně přístupné pro obsluhu. Typickým příkladem bezkontaktního tísňového hlásiče je bezkontaktní bankovkový detektor, který se používá v bankách a hlídá odejmutí poslední bankovky z příslušného místa.

Do PTS také patří osobní tísňové hlásiče, které jsou určeny pro strážníky vězeňské služby, lze je ale také uplatnit ve strážní službě nebo pro pracovníky v psychiatrických léčebnách. Osobní tísňový hlásič se neumísťuje na stěny, ale má ho daný pracovník připevněný na ruce či zavěšený jako přívěšek na krku nebo může být i volně v kapse.

Dále lze tísňové hlásiče využít u lidí s určitými zdravotními problémy, kteří mohou aktivací tísňového zařízení snadno přivolat lékařskou pomoc.

Detektory

Detektory jsou zařízení poplachového zabezpečovacího a tísňového systému, jejichž účelem je detekce narušení zabezpečeného prostoru.

Ochranu zabezpečeného prostoru lze rozdělit na předmětovou, prostorovou, plášťovou a perimetrickou. Do předmětové ochrany patří závěsové, polohové nebo kapacitní detektory, které mají za úkol jen zjistit odcizení daného předmětu. Prostorová ochrana slouží pro zjištění pohybu uvnitř v objektu. Nejčastěji používanými prvky prostorové ochrany jsou pasivní infračervené detektory, jejichž princip spočívá v zachycování změn v infračerveném pásmu kmitočtového spektra elektromagnetického vlnění. [3]

Často se ovšem používají také ultrazvukové detektory, mikrovlnné detektory nebo duální detektory, kde duální detektory jsou kombinací ultrazvukových a mikrovlnných detektorů. Hlavním úkolem plášťové ochrany je zjistit pokus o vloupání přes okna, dveře a stěny. Mezi klasické prvky plášťové ochrany pak patří magnetický kontakt, mechanický kontakt,

poplachové fólie, tapety, polepy, vibrační detektory, drátové detektory a detektory na ochranu prosklených ploch.

Zbývají prvky perimetrické ochrany. Obdobným názvem je venkovní obvodová ochrana a hlídají se venkovní prostory. Patří sem mikrovlnné bariéry, štěrbinové kabely, mikrofonické kabely, infračervené závory nebo bariéry a pasivní infračervené detektory určené pro perimetrickou ochranu.

1.4.3 Indikační a ovládací zařízení

V závislosti na použitém ovládacím prvku může ovládací zařízení sloužit například i pro nastavení nebo programování systému, ale hlavní úlohou je uvést systém do stavu střežení nebo klidu. Nejpoužívanějšími zařízeními pro zastřežení nebo odstřežení jsou čipové karty, klávesnice a v mnohých zemích se k zastřežení provede automaticky při uzamčení domu.

Indikační zařízení pak slouží k informování o zastřežení, odstřežení nebo poruchách systému.

1.4.4 Signalizační a ostatní spínaná zařízení

Pod pojmem signalizační zařízení se rozumí zařízení určené k informování o narušení objektu, a to buď opticky, akusticky nebo opticky i akusticky. Mezi nejrozšířenější signalizační zařízení patří houkačky, blikače a sirény.

Do ostatních zařízení si lze představit jakékoli spínané zařízení. Existují i systémy, které spínají elektrické zařízení - jako například televizi a rádio. Hlavním cílem je tedy vystrašení pachatele, jelikož si při sepnutí zmíněného zařízení může myslet, že v domě není sám. Dále zde patří i zamlžovací systémy, které v objektu vytvoří do několika sekund mlhu, a tím výrazně sníží viditelnost.

1.4.5 Příjem poplachového signálu a poplachový přenosový systém

Pro příjem poplachového signálu z ústředny se nejčastěji využívá mobilní telefon majitele nebo poplachové přijímací centrum. Výhody a nevýhody již byly zmíněny v kapitole 1.1.1. Poplachový přenosový systém pak slouží pro přenos poplachové informace od ústředny do

přijímacího zařízení. K vyslání se používá komunikátor, který bývá zpravidla zabudován přímo v ústředně. Lze využít radiové vysílání, které je sice velmi drahé, ale přenos probíhá v reálném čase a není nutné mít telefonní linku v objektu. Toto vysílání přenáší všechny informace, které je schopna poplachová ústředna poskytnout. Další možností je využití GSM komunikátoru, kdy není nutné mít v objektu telefonní linku, ovšem přenos dat je závislý na zatížení sítě, v současnosti operátoři nepodporují rozdělení SMS zpráv podle priorit a majitel platí za každou SMS. [4]

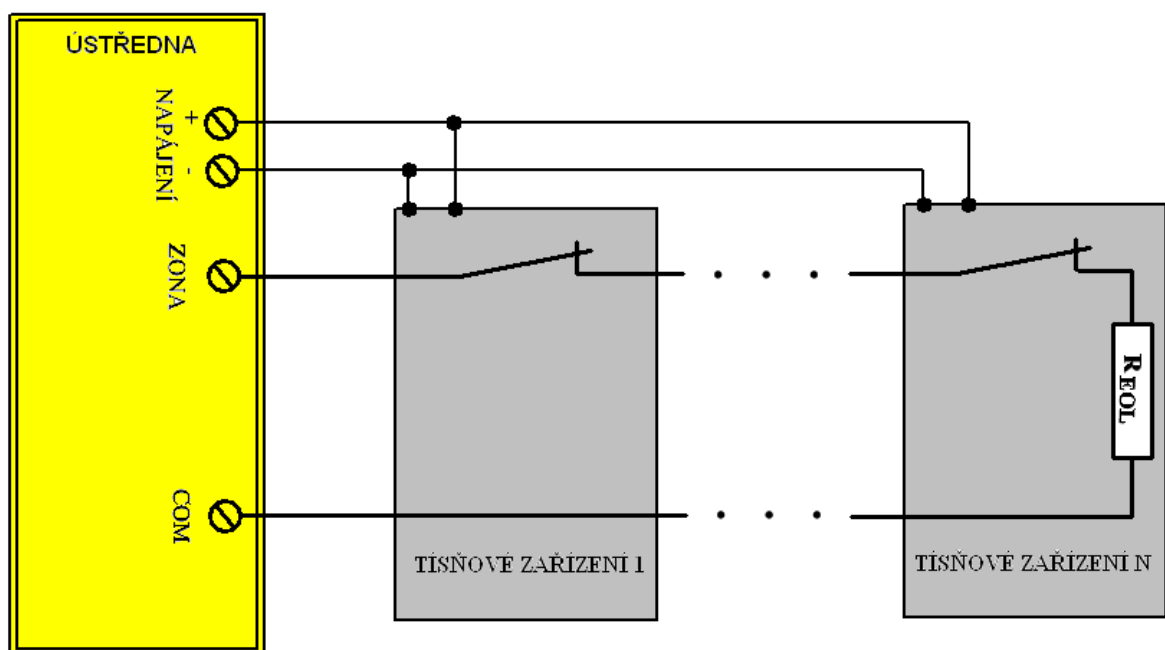
Při využití telefonní linky je přenos pomalejší, méně spolehlivý, platí se za každý telefonát a je nutností mít telefonní linku v objektu, ale obvykle má každá ústředna telefonní komunikátor již zabudovaný a není tedy potřeba ho dokupovat. Poslední možností je přenos pomocí internetu. Poplatky za měsíční připojení na poplachové přijímací centrum bývají obvykle velmi nízké, ale na druhou stranu musí mít objekt připojení k internetu.

1.5 Nejpoužívanější způsoby propojení tísňových zařízení s ústřednou PZTS

Tísňové zařízení lze zapojit do obvodu buď se zakončovacím odporem nebo bez něj. V případě použití obvodu se zakončovacím odporem bývá tento odpor označen zkratkou EOL (End Of Line) a jeho hodnota je zpravidla $1k\Omega$. Pomocí zmíněného odporu je možné zjistit, jestli případný pachatel zónu nepřerušil nebo nezkratoval. Je třeba ovšem podotknout, že není nutností tento EOL odpor mít, protože se u tísňových zařízení pokus o sabotáž nepředpokládá. Totéž platí i pro zapojení požárních hlásičů. Pokud ovšem použijeme zapojení se zakončovacím odporem, bývá zvykem umístit ho do posledního detektoru. Napájení je na obrázcích vyznačeno, ale většinou se jedná pouze o spínače, které jsou aktivovány manuálně a napájení nepotřebují. Princip vyhodnocení poplachu spočívá v měření hodnoty odporu na dané smyčce. Klidová a poplachová hodnota odporu závisí na zvoleném způsobu zapojení. V následujících kapitolách budou popsány způsoby, kterými lze připojit tísňové zařízení, ovšem obdobné zapojení lze použít i u analogových požárních hlásičů.

1.5.1 Zapojení NC (normally closed – v klidu uzavřeno) se zakončovacím odporem

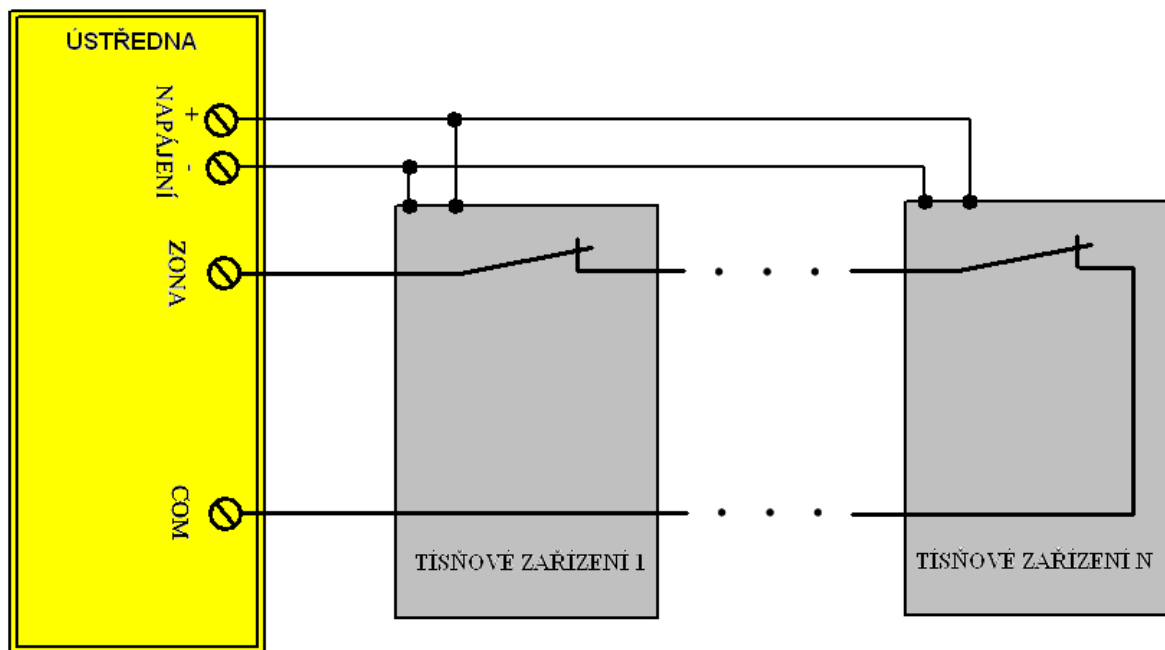
Princip vyhodnocení je jednoduchý. Ústředna měří hodnotu odporu mezi svorkou “ZONA“ a “COM“. Kontakty jednotlivých tísňových zařízení jsou připojeny sériově a jsou v klidovém stavu uzavřeny. Při aktivování tísňového hlásiče dojde k rozepnutí kontaktu, to způsobí výrazné zvýšení odporu dané smyčky, a ústředna vyhlásí poplach. Zapojení je znázorněno na obr. 2. Maximální počet tísňových hlásičů na jedné smyčce není stanoven, ale doporučuje se maximálně deset tísňových zařízení na jednu smyčku.



Obrázek 2: Zapojení pro tísňové zařízení typu NC se zakončovacím odporem

1.5.2 Zapojení NC (normally closed) bez zakončovacího odporu

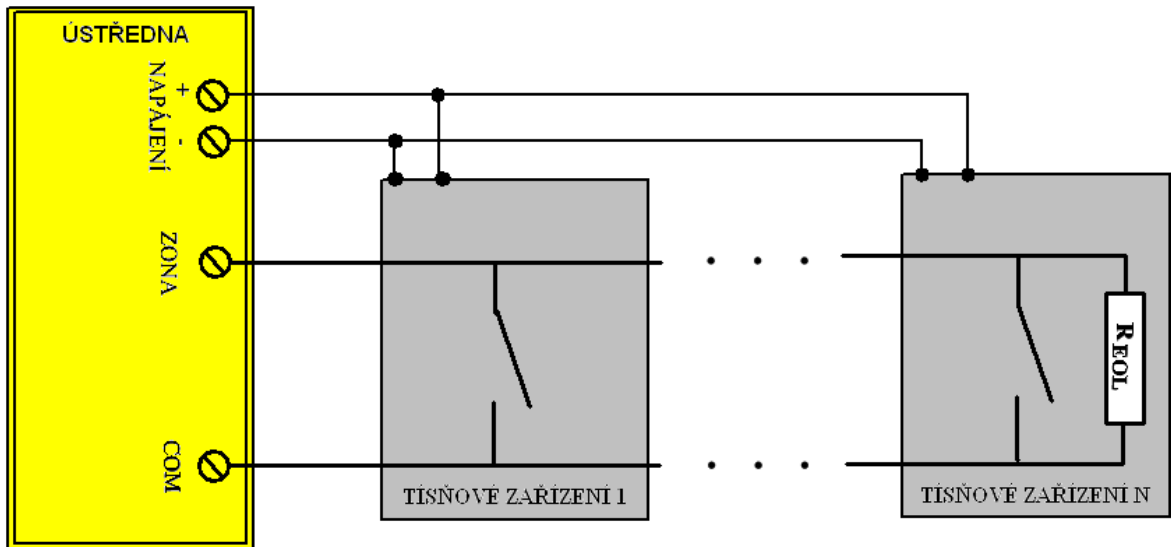
Zapojení je znázorněno na obr. 3. Princip funkce je stejný jako při zapojení NC se zakončovacím odporem. Jediný rozdíl je v chybějícím zakončovacím odporu.



Obrázek 3: Zapojení pro tísňové zařízení typu NC bez zakončovacího odporu

1.5.3 Zapojení NO (normally opened – v klidu otevřeno) se zakončovacím odporem

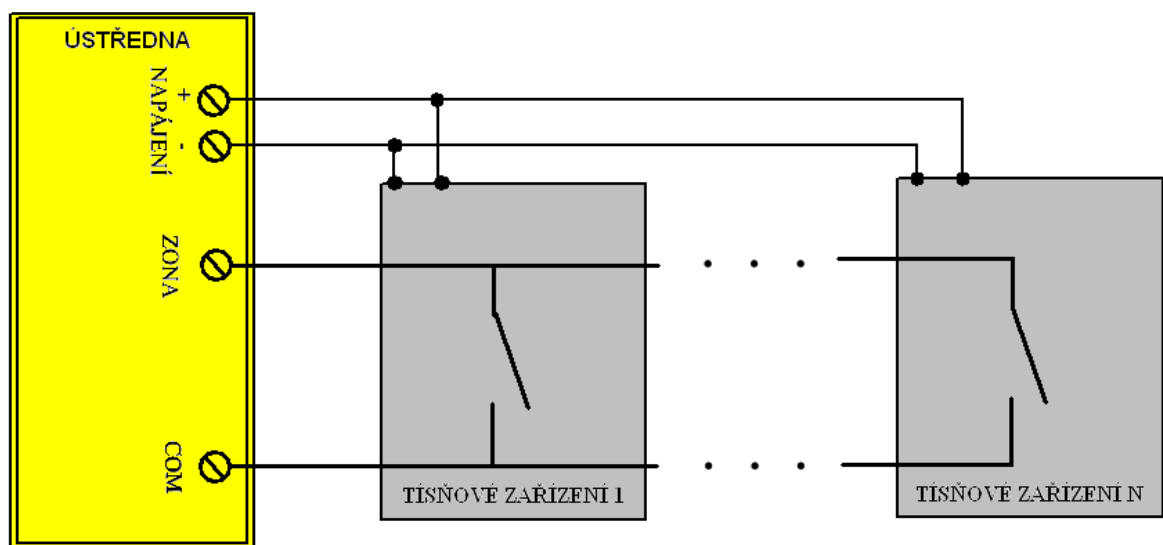
U zapojení NO jsou kontakty tísňových zařízení na smyčce připojeny paralelně, jsou v klidovém stavu rozepnuty a obvod je zakončen zakončovacím odporem. Při aktivaci některého z tísňových zařízení se kontakt uzavře, tím se sníží odpor smyčky, a ústředna vyhlásí poplach. Zapojení znázorňuje obr. 4.



Obrázek 4: Zapojení pro tísňové zařízení typu NO se zakončovacím odporem

1.5.4 Zapojení NO bez zakončovacího odporu

Princip funkce je vysvětlen v kapitole 1.5.3 a opět je jediný rozdíl v tom, že obvod nemá zakončovací odpor. Zapojení znázorňuje obr. 5.



Obrázek 5: Zapojení pro tísňové zařízení typu NO bez zakončovacího odporu

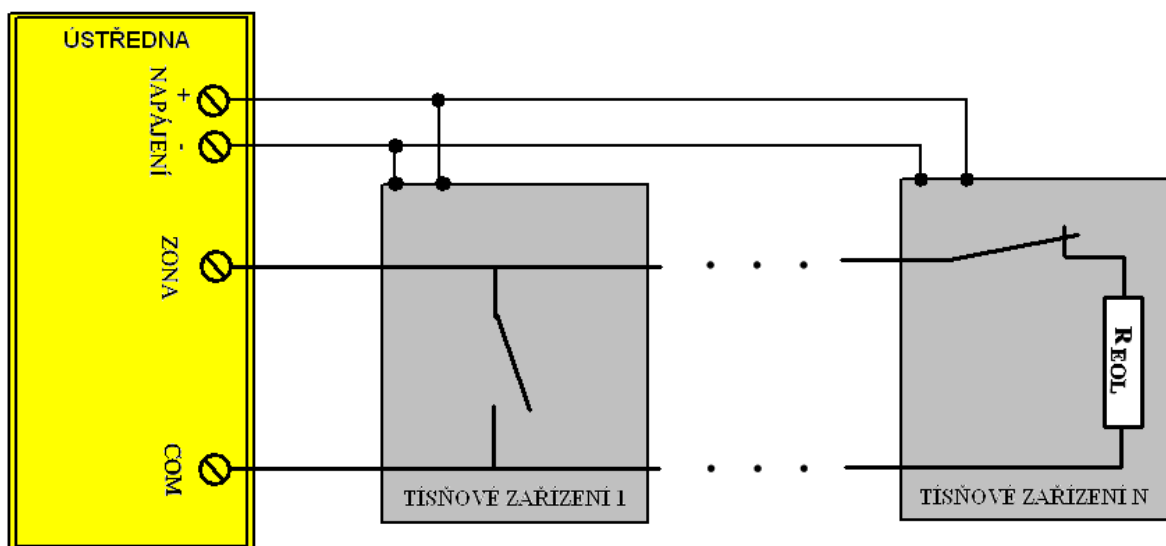
1.5.5 Kombinace NC (v klidu uzavřeno) a NO (v klidu otevřeno) se zakončovacím odporem

Obvod je připojen na svorky "ZONA" a "COM". Tísňové zařízení 1 je připojeno jako NO, tísňové zařízení N jako NC a zapojení obsahuje i zakončovací odpor. Ústředna si zde hlídá hodnotu zakončovacího odporu.

Kontakt tísňového zařízení 1 je v klidovém stavu rozeprt a při aktivaci se sepne. Tím se sníží odpor smyčky a ústředna vyhlásí poplach.

Kontakt tísňového zařízení N je naopak v klidovém stavu uzavřen a při aktivaci se rozeprne, čímž se zvýší odpor smyčky a vyhlásí se poplach.

Ze všech uvedených variant zapojení se tato kombinace používá nejméně. Znázornění lze nalézt na obr. 6.



Obrázek 6: Zapojení pro tísňové zařízení – kombinace NC a NO se zakončovacím odporem

1.6 Nejpoužívanější způsoby propojení detektorů s ústřednou PZTS

Detektory jsou připojeny k vyhodnocovací smyčce a stejně jako u tísňových hlásičů spočívá princip v měření hodnoty odporu dané smyčky. V klidovém stavu je odpor smyčky přibližně roven zakončovacímu odporu, který bývá $1k\Omega$. Musí zde však být určitá tolerance, protože hodnota odporu smyčky je závislá na teplotě, délce a šířce vedení. Používá se tolerance okolo 30%. Jednotlivé možnosti zapojení jsou probrány v následující kapitole. Každý detektor musí být připojen do obvodu se zakončovacím odporem, aby byla zajištěna ochrana proti sabotáži zkratováním a přerušením vedení. Zakončovací odpor bývá umístěn v posledním detektoru.

V současné době se rozlišují čtyři druhy sabotáží - sabotáž zkratováním vedení, přerušením vedení, narušením krytu detektoru a sabotáž zamaskováním scény. Sabotáží zkratováním vedení se pachatel snaží obejít ochranu proti sabotáži tak, že danou smyčku zkratuje. V sabotáži přerušením vedení se pachatel snaží obejít ochranu tím způsobem, že přeruší vedení na dané smyčce.

Před těmito dvěma pokusy o sabotáž je smyčka chráněna právě díky zakončovacímu odporu.

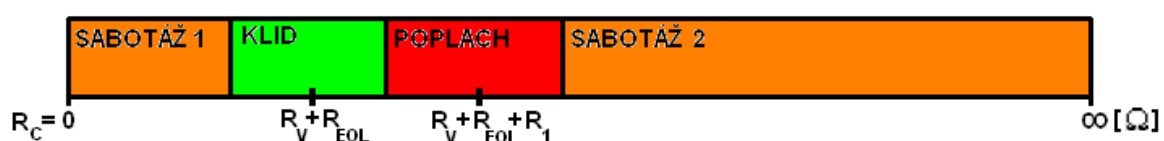
V případě, že se pachatel snaží dostat do vnitřních částí detektoru, tak se jedná o sabotáž narušením krytu detektoru, kde je ochrana zaručena kontaktem s názvem "TAMPER", který se při otevření krytu detektoru rozepne.

Jako poslední zbývá sabotáž zamaskováním scény, kdy se pachatel snaží potlačit schopnost detektoru zjistit pachatele v objektu tak, že buď umístí předmět před detektor, nebo na něj nanese vrstvu barvy například pomocí spreje. Zde se pak používá detekce zamaskování, která informuje ústřednu o narušení dalším kontaktem připojeným na smyčku s názvem "ANTIMASKING". Paralelně ke zmíněnému kontaktu se může přidat další odpor, díky kterému je možné rozlišit poplach vyvolaný kontaktem "TAMPER" a kontaktem "ANTIMASKING".

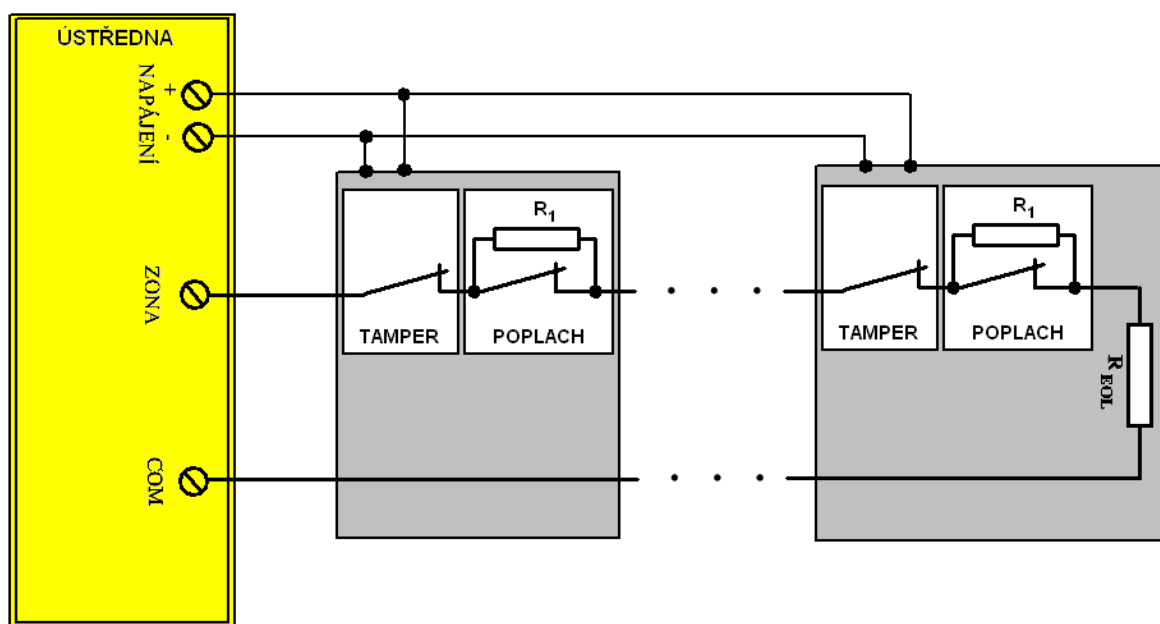
1.6.1 Zapojení typu NC dvojitě vyvážená

Zapojení je znázorněno na obr. 8, kde vidíme, že obvod obsahuje zakončovací odpor "R_{EO}L" a každý detektor má dva kontakty. Kontakt "TAMPER" a "POPLACH". Paralelně ke kontaktu "POPLACH" je připojen odpor "R₁", díky kterému ústředna rozliší, jestli byl poplach vyvolán z důvodu narušení prostoru nebo pokusu o sabotáž vniknutím do krytu

detektoru. Hodnota odporu “ R_1 “ je vždy $1k\Omega$. Jednotlivé poplachové stavy v závislosti na odporu smyčky znázorňuje obr. 7. Je-li celkový odpor smyčky “ R_C “ přibližně roven nule, pak jde o sabotáž zkratováním vedení, při hodnotě odporu smyčky, která se pohybuje okolo součtu zakončovacího odporu “ R_{EOL} “ a odporu vedení “ R_V “, jde o klidový stav. Jestliže platí $R_C \approx R_{EOL} + R_V + R_1$, vyhlásí ústředna poplach z důvodu narušení chráněného prostoru. Když celkový odpor smyčky R_C výrazněji převyšuje součet $R_{EOL} + R_V + R_1$, je vyhlášen poplach z důvodu sabotáže vniknutím do ústředny nebo přerušení vedení. Tyto dvě možnosti ústředna nerozliší.



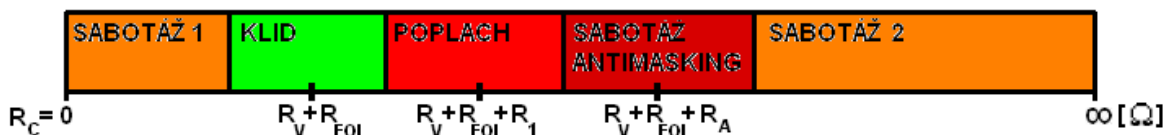
Obrázek 7: Hodnota odporu smyčky v závislosti na vyvolaném poplachu NC dvojitě vyvážená



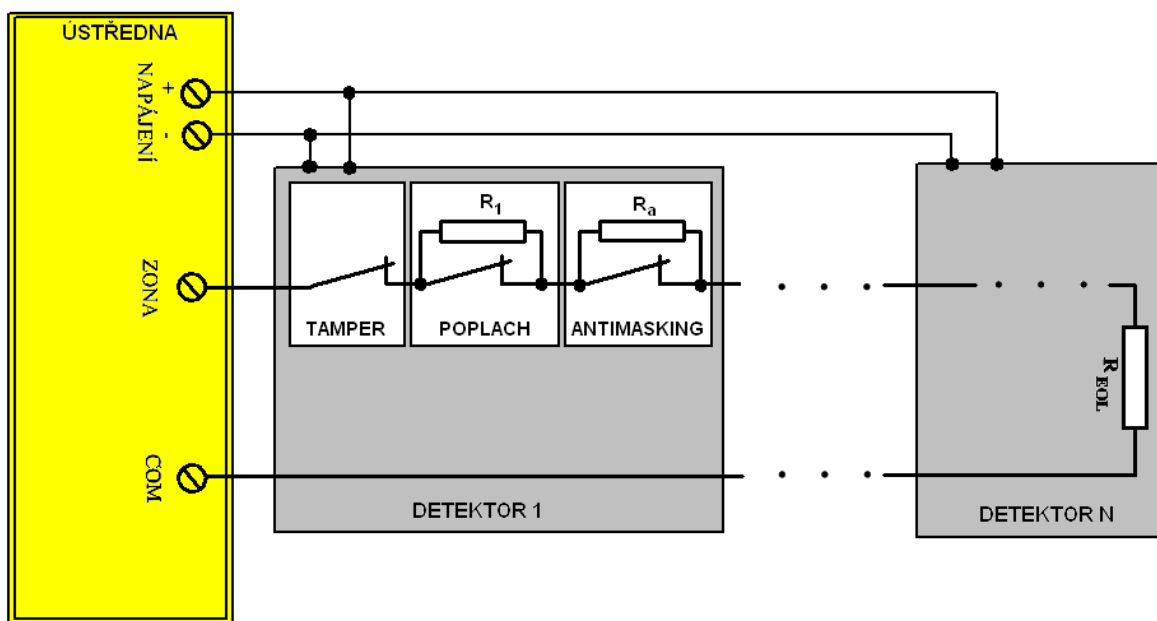
Obrázek 8: Zapojení typu NC dvojitě vyvážená

1.6.2 Zapojení typu NC trojitě vyvážená

Zapojení je velmi podobné předchozímu zapojení, ovšem každý detektor má navíc ještě kontakt "ANTIMASKING" pro detekci sabotáže zamaskováním scény a paralelně ke kontaktu "ANTIMASKING" je odpor " R_A ", pomocí něhož lze zmíněnou sabotáž odlišit od ostatních sabotáží. V případě, že narušitel zamaskuje scénu, bude celková hodnota smyčky $R_C \approx R_{EOL} + R_V + R_A$ (R_{EOL} je hodnota zakončovacího odporu, R_V odpor vedení a R_A hodnota odporu, který je paralelně ke kontaktu "ANTIMASKING"). Hodnota odporu " R_A " bývá $12k\Omega$. Samozřejmě lze také použít i předešlé zapojení tak, že odpor " R_A " nepřipojíme. Nevýhodou by pak ovšem byla nemožnost rozlišení sabotáže zamaskováním scény od sabotáže narušením krytu detektoru a sabotáže přerušением vedení. Zapojení znázorňuje obr. 10 a poplachové stavy v závislosti na odporu smyčky znázorňuje obr. 9.



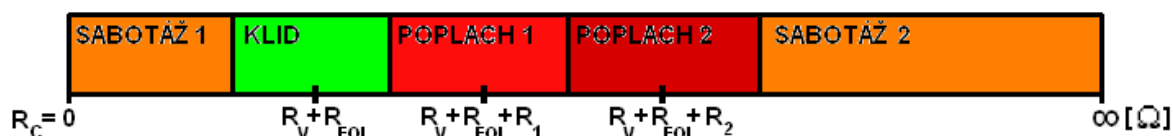
Obrázek 9: Hodnota odporu smyčky v závislosti na vyvolaném poplachu NC trojitě vyvážená



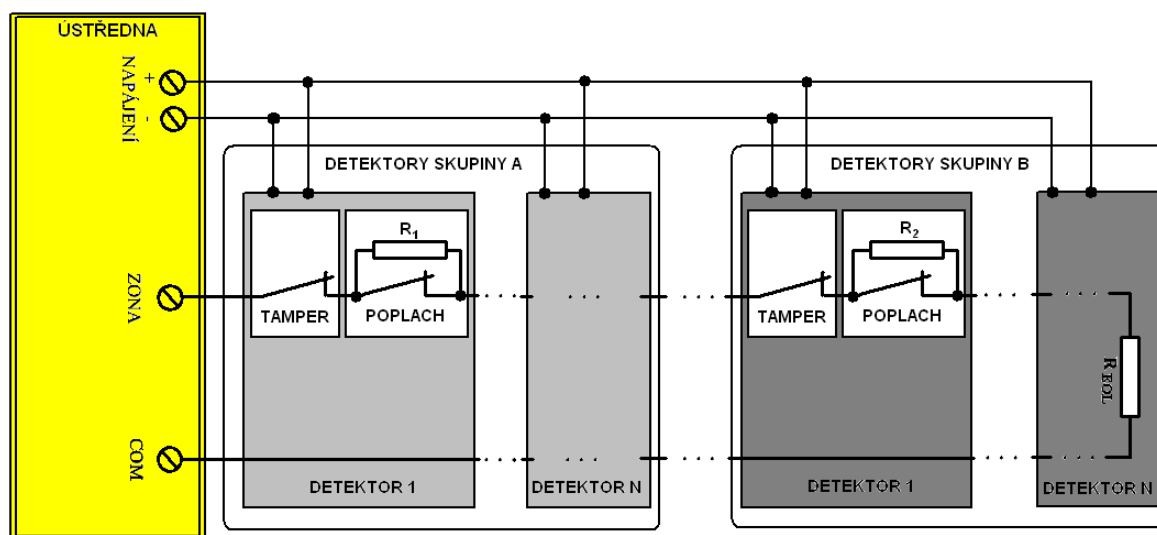
Obrázek 10: Zapojení typu NC trojitě vyvážená

1.6.3 Zapojení typu NC zdvojení zóny

V současné době je už u každé ústředny možné použít zdvojení zóny. Zapojení je znázorněno na obr. 12 a je podobné jako zapojení NC dvojitě vyvážené, ale s tím rozdílem, že se detektory rozdělí do dvou skupin. První skupina detektorů používá pro poplach, který je vyvolaný narušením prostoru pachatelem, již zmíněný odpor " R_1 " a druhá skupina detektorů používá odpor " R_2 ". Z toho vyplývá, že výhodou zapojení NC zdvojení zóny oproti zapojení NC dvojitě vyvážená je schopnost zjistit, která skupina detektorů vyhlásila poplach, a tím se zpřesní místo, kde byl poplach vyvolán. Při vyhlášení poplachu některým detektorem z druhé skupiny pak platí vztah $R_C \approx R_{EOL} + R_V + R_2$ (kde R_{EOL} je hodnota zakončovacího odporu, R_V odpor vedení a R_2 hodnota odporu, který je paralelně ke kontaktu "POPLACH" u některého z druhé skupiny detektorů). Hodnota odporu " R_2 " závisí na výrobci, ale naprostá většina ústředěn používá 2 k Ω nebo 2,2 k Ω . Hodnoty odporu smyčky v závislosti na vyvolaném poplachu jsou znázorněny na obr. 11.



Obrázek 11: Hodnota odporu smyčky v závislosti na vyvolaném poplachu NC zdvojení zóny



Obrázek 12: Zapojení typu NC zdvojení zóny

2 POŽADAVKY NOREM NA POPLACHOVÉ ZABEZPEČOVACÍ A TÍSŇOVÉ SYSTÉMY

Vysvětlení používaných zkratk

Norma ČSN EN 50 131 rozlišuje poplachové systémy pro detekci vniknutí a poplachové systémy pro detekci přepadení. [5]

Zkratkou IAS (Intruder Alarm System) se označují poplachové systémy pro detekci vniknutí do střeženého objektu. V českých textech se místo IAS používá zkratka PZS (poplachový zabezpečovací systém).

Zkratka HAS (Hold-up Alarm System) slouží pro pojmenování systémů pro detekci přepadení. V českých textech je nahrazena zkratka HAS zkratkou PTS a pojmenováním poplachový tísňový systém.

V případě, že daný poplachový systém obsahuje prvky pro detekci vniknutí i přepadení, používá se v odborných textech zkratka I&HAS (Intruder and Hold-up Alarm System).

Česká ekvivalentní zkratka je pak PZTS (poplachový zabezpečovací a tísňový systém).

2.1 Vybrané zkratky používané v poplachových zabezpečovacích systémech

Poplach (alarm) – výstraha na přítomnost nebezpečí pro život, majetek nebo okolní prostředí.

Poplachový signál nebo zpráva – signál nebo zpráva generovaná poplachovým bodem.

Poplachové přijímací centrum – obsluhované vzdálené středisko, do něž se přenášejí informace o stavu jednoho nebo více I&HAS.

Poplachový stav – stav I&HAS nebo jeho komponent, který je výsledkem odezvy systému na přítomnost nebezpečí.

Poplachový systém – elektrické zařízení, které reaguje na manuální podnět nebo automatickou detekci přítomnosti nebezpečí.

Poplachový přenosový systém – zařízení a síť používaná pro přenos informace mezi jedním nebo více I&HAS a jedním nebo více poplachovými přijímacími centry.

Ústředna – zařízení pro příjem, ovládání, indikaci a inicializaci následného přenosu informací.

Detektor – prvek, který je určen k vyslání poplachového signálu nebo zprávy jako odezvy na detekci abnormálního stavu, indikující přítomnost nebezpečí.

Příchodová/odchodová trasa – trasa, po níž je možné uskutečnit oprávněný příchod do střeženého prostoru nebo odchodu z něj.

Poplachový tísňový systém – poplachový systém, který poskytuje uživateli možnost úmyslného vyvolání poplachu.

Střežené prostory – část budovy nebo prostoru, kde může I&HAS detekovat pokus o vloupání aktivací tísňového zařízení.

Sabotáž – úmyslná nedovolená manipulace s I&HAS nebo jeho částmi.

Stav sabotáže – stav I&HAS, v němž byla detekována sabotáž.

Ochrana proti sabotáži – metody nebo prostředky I&HAS proti úmyslné nedovolené manipulaci.

Tísňové zařízení – zařízení, jehož aktivace způsobí generování poplachového tísňového signálu [6]

Komunikátor ve střeženém objektu – součást poplachového systému, které tvoří výstupní rozhraní pro přenos zpráv mezi I&HAS a poplachovou přenosovou sítí. [7]

Periodická komunikace – „Periodická“ znamená, že v předdefinovaném intervalu by se měla uskutečnit alespoň jedna zpráva pro ujištění, že je komunikace funkční. [8]

Expandér – elektronické zařízení, sloužící k rozšíření funkcí EZS. [9]

Planý poplach – každý poplach, který nevznikne vlivem přítomnosti či pohybu narušitele

Falešný poplach – poplach, který vznikne vadou elektronické součástky nebo jinou poruchou detektoru. [6]

Propojení – prostředky s jejichž pomocí jsou zprávy nebo signály přenášeny mezi komponenty I&HAS. [10]

Subsystém – část PZTS, která je schopna samostatného provozu. [11]

Výstražné zařízení – zařízení, které produkuje výstražný zvukový poplachový signál v odezvě na hlášení poplachu.

Detekce sabotáže – detekce úmyslného zásahu do I&HAS nebo jeho komponent.

Doplňkové ovládací zařízení – zařízení určené pro doplňkové účely.

Zabezpečený prostor – část budovy nebo prostoru, kde může být prostřednictvím I&HAS detekováno vniknutí, pokus o vniknutí nebo aktivace tísňového prostředku. [5]

Sabotáž – úmyslné zasahování s nedovolenou manipulací do PZS nebo jeho částí. [12]

2.2 Stupně zabezpečení I&HAS podle normy ČSN. EN 50131-7

Stupeň 1 – Nízké riziko

Předpokládá se, že narušitelé nebo lupiči mají malou znalost I&HAS a že mají k dispozici omezený sortiment snadno dostupných nástrojů.

Stupeň 2 – Nízké až střední riziko

Předpokládá se, že narušitelé nebo lupiči mají malou znalost I&HAS a že použijí základní sortiment nástrojů a přenosných přístrojů.

Stupeň 3 – Střední až vysoké riziko

Předpokládá se, že narušitelé nebo lupiči jsou obeznámeni s I&HAS a mají úplný sortiment nástrojů a přenosných elektronických zařízení.

Stupeň 4 – Vysoké riziko

Používá se, má-li zabezpečení prioritu před všemi ostatními hledisky. Předpokládá se, že narušitelé nebo lupiči jsou schopni nebo mají možnost zpracovat podrobný plán narušení nebo loupeže a mají kompletní sortiment zařízení včetně prostředků pro náhradu komponentů v I&HAS. [6]

2.3 Požadavky norem na ústředny

V tab. 1 jsou znázorněny požadavky normy na opakované neplatné pokusy o udělení oprávnění. Tabulka je rozdělena do čtyř stupňů zabezpečení a popisuje požadavky na maximální počet špatně zadaných kódů před prvním i dalším zablokováním vstupního zařízení, záznamy času zablokování vstupního uživatelského zařízení, požadavky na sabotážní signál a maximální počet pokusů před aktivací sabotáže.

	Stupeň 1	Stupeň 2	Stupeň 3	Stupeň 4
Zablokování vstupního(ch) uživatelského(ých) zařízení	V	V*	P	P
Maximální počet pokusů před prvním zablokováním vstupního(ch) uživatelského(ých) zařízení	10	10	10	3
Maximální počet dalších pokusů před zablokováním vstupního(ch) uživatelského(ých) zařízení	10	10	1	1
Záznam každého času zablokování vstupního(ch) uživatelského(ých) zařízení do paměti událostí	V	V	V	P
Sabotážní signál nebo zpráva	V	V*	V	P
Maximální počet pokusů před aktivací sabotáže	21	21	21	7
P = povinné V = volitelné				
* Pro stupeň zabezpečení 2 musí být splněn alespoň jeden z těchto dvou požadavků.				

Tabulka 1: Detekce opakovaných neplatných pokusů o udělení oprávnění [13]

Pro stupeň 1 je maximální možný počet pokusů zadání špatného kódu 10, před prvním zablokováním vstupního zařízení na určitou dobu. Počet pokusů před opětovným zablokováním je opět 10. Jestliže se zadá přístupový kód špatně 21 krát, pak musí být aktivována sabotáž.

Obdobné je to ve stupni 2. Jediný rozdíl je, že po 21. špatném zadání přístupového kódu musí být buď zablokováno vstupní uživatelské zařízení nebo vyslán sabotážní signál.

Ve stupni zabezpečení 3 je počet pokusů před prvním zablokováním 10 a počet dalších pokusů může být jen jeden. V případě, že se zadá přístupový kód špatně 21 krát, musí být zablokováno vstupní zařízení.

Nejpřísnější je stupeň 4, kdy se po 3. špatně zadaném kódu zablokuje vstupní uživatelské zařízení a počet dalších pokusů před opětovným zablokováním je jen jeden. Maximální možný počet špatně zadaných kódů je 7, přičemž pak musí být zablokováno vstupní zařízení a vyslán sabotážní signál nebo zpráva.

	Stupeň 1	Stupeň 2	Stupeň 3	Stupeň 4
Funkce monitorování běhu programu	V	V	P	P
Signál poruchy běhu programu	V	V	V	P
V = volitelné P = povinné				

Tabulka 2: Monitorování běhu programu [13]

Z tabulky 2 vyplývá, že stupeň zabezpečení 3 musí mít funkci monitorování běhu programu. Pro stupeň 4 je pak povinností i vyslání signálu poruchy běhu programu.

	Odkaz	Zpracování delší než	Zpracování a hlášení během	Minimum	Maximum
Signál vloupání	8.9	400 ms	10 s		
Tísňový signál	8.9	400 ms	10 s		
Sabotážní signál	8.9	400 ms	10 s		
Poruchový signál	8.9	10 s	10 s		
Signál zakrytí – zpracován jako vloupání	8.9 / EN 50131-1:2006, 8.4.5	400 ms	10 s		
Signál zakrytí – zpracován jako porucha	8.9 / EN 50131-1:2006, 8.4.5	10 s	10 s		
Signál podstatného snížení rozsahu pokrytí – zpracován jako vloupání	8.9 / EN 50131-1:2006, 8.4.6	400 ms	10 s		
Signál podstatného snížení rozsahu pokrytí – zpracován jako porucha	8.9 / EN 50131-1:2006, 8.4.6	10 s	10 s		
Aktivace zpoždění výstražného zařízení po dálkovém hlášení	8.6			0	10 min
Doba trvání akustické výstražné signalizace	8.6			90 s	15 min
Porucha vnějšího napájecího zdroje (EPS)	8.6	10 s	1 h ^a		
Kontrola chodu hlavního programu	8.4.3	10 s	30 s ^a		
Doba trvání indikace střežení po uvedení do tohoto stavu	8.3.3				^b
Doba trvání indikace stavu klidu po uvedení do tohoto stavu	8.3.4				30 s
Doba trvání uvádění do stavu klidu	8.3.4				45 s
^a Může být zrušena, pokud podmínky pominou během této doby zpoždění.					
^b Tato indikace je stanovena jako „časové omezení“, ale žádné aktuální omezení není v EN 50131-1 stanoveno. Časové omezení se nepoužívá u systémů stupňů 1 a 2 dle EN 50131-1:2006, 8.3.7, c).					

Tabulka 3: Časování [13]

Nejvyšší požadavky jsou kladeny na signál tísňový, sabotážní, signál vloupání a zakrytí detektoru, který je zpracován jako signál vloupání, a dále jsou nejvyšší požadavky kladeny na signál podstatného snížení rozsahu pokrytí, který je zpracován jako signál vloupání, kdy nesmí být doba zpracování vyšší než 400ms. Hlášení pak musí být uskutečněno do 10s.

Menší nároky jsou kladeny na poruchový signál a signál zakrytí, který je zpracován jako porucha, a dále jsou menší nároky kladeny na signál podstatného snížení rozsahu pokrytí, který je zpracován jako porucha, kde nesmí trvat zpracování signálu a zpracování hlášení déle než 10s.

Tabulka 3 dále znázorňuje, že zpracování signálu poruchy vnějšího napájecího zdroje a kontroly chodu hlavního programu nesmí překročit 10s. Zpracování hlášení poruchy vnějšího napájecího zdroje nesmí trvat déle než hodinu a zpracování hlášení chodu hlavního programu nesmí překročit 30s. Ovšem kontrolu chodu hlavního programu je povinné provádět jen u stupně zabezpečení 3 a 4.

Doba trvání akustické výstražné signalizace musí trvat 90 sekund až 15 minut.

Aktivace zpoždění výstražného zařízení po dálkovém hlášení musí trvat maximálně 10 minut.

Doba trvání indikace stavu klidu po uvedení do tohoto stavu nesmí překročit 30 sekund a doba trvání uvádění do stavu klidu nesmí přesáhnout 45 sekund.

Požadavky norem na ochranu proti sabotáži

V normě ČSN EN 50 131-1 ed. 2 je na straně 30 uvedeno, že ústředna, doplňkové ovládací zařízení, poplachový přenosový systém, výstražné zařízení a napájecí zdroj musí být chráněny před sabotáží ve všech stupních zabezpečení. Tísňová zařízení a detektory vniknutí musí být chráněny ve stupni druhém, třetím a čtvrtém. Zbývají rozvodné krabice, které musí být chráněny jen v třetím a čtvrtém stupni.

Na straně 31 jsou rozepsány sabotáže, které musí být detekovány. Před otevřením normálním způsobem musí být chráněny komponenty ve všech stupních zabezpečení. Odejmutí z montážní plochy musí být detekováno v druhém, třetím a čtvrtém stupni zabezpečení.

3 ZÁKLADY PROGRAMOVÁNÍ V C#

C# je moderní, objektově orientovaný a typově spolehlivý programovací jazyk, který vychází z programovacího jazyku C a C++. Jazyk je v práci popsán, protože je použit v praktické části.

Mezi klíčové vlastnosti C# patří:

- Plná podpora tříd a objektově orientovaného programování
- Konzistentní a dobře definovaná množina základních typů
- Automatické čištění dynamicky přidělované paměti
- Možnost označení tříd nebo metod uživatelsky definovanými atributy
- Úplný přístup k základní knihovně tříd prostředí .NET
- Ukazatele a bezprostřední přístup k paměti na vyžádání
- Možnost překladu kódu do aplikace nebo knihovny komponent prostředí .NET [14]

3.1 Hodnotové a odkazové datové typy

Pro práci s daty se v programu C# používají dva způsoby. Buď se jedná o práci s hodnotovými nebo odkazovými datovými typy. Při vytvoření hodnotového datového typu naalokujete místo v paměti a uložíte do něj hodnoty, s kterými pak přímo pracujete. Při vytvoření odkazového datového typu taky naalokujete místo v paměti, s kterým pracujete, ale v programu pracujete s odkazem na tuto paměť.

3.1.1 Proměnná

Proměnná se skládá z datového typu a identifikátoru. Použijeme-li tedy v programu jakýkoli datový typ, pak musíme k němu přiřadit i zmíněný příslušný identifikátor.

Pořadí je následující:

datový_typ identifikátor; [14]

3.1.2 Hodnotové datové typy

Hodnotové datové typy lze rozdělit na uživatelsky definované, vestavěné a výčtové.

Mezi uživatelsky definované se řadí struktury, které jsou označeny slovem `struct`. Struktury budou podrobněji vysvětleny dále.

Vestavěné hodnotové typy slouží pro práci s daty, přičemž těmito daty mohou být celá čísla, čísla v pohyblivé řádkové čárce, znaky nebo pravdivostní hodnoty. [15]

Jednotlivé vestavěné hodnotové typy včetně rozsahu hodnot a použití jsou znázorněny v tabulce 4 a 5.

Zkrácený zápis v C#	V souladu s CLS?	Typ v System	Rozsah	Význam
<code>sbyte</code>	Ne	<code>System.SByte</code>	-128 až 127	8bitové číslo se znaménkem
<code>byte</code>	Ano	<code>System.Byte</code>	0 až 255	8bitové číslo bez znaménka
<code>short</code>	Ano	<code>System.Int16</code>	-32 768 až 32 767	16bitové číslo se znaménkem
<code>ushort</code>	Ne	<code>System.UInt16</code>	0 až 65 535	16bitové číslo bez znaménka
<code>int</code>	Ano	<code>System.Int32</code>	-2 147 483 648 až 2 147 483 647	32bitové číslo se znaménkem
<code>uint</code>	Ne	<code>System.UInt32</code>	0 až 4 294 967 295	32bitové číslo bez znaménka
<code>long</code>	Ano	<code>System.Int64</code>	-9 223 372 036 854 775 808 až 9 223 372 036 854 775 807	64bitové číslo se znaménkem

Tabulka 4: Hodnotové datové typy 1 [16]

Zkrácený zápis v C#	V souladu s CLS?	Typ v System	Rozsah	Význam
<code>ulong</code>	Ne	<code>System.UInt64</code>	0 až 18 446 744 073 709 551 615	64bitové číslo bez znaménka
<code>char</code>	Ano	<code>System.Char</code>	U0000 až Uffff	Jediný 16bitový znak Unicode
<code>float</code>	Ano	<code>System.Single</code>	$1,5 \times 10^{-45}$ až $3,4 \times 10^{38}$	32bitové číslo v pohyblivé řádkové čárce
<code>double</code>	Ano	<code>System.Double</code>	$5,0 \times 10^{-324}$ až $1,7 \times 10^{308}$	64bitové číslo v pohyblivé řádkové čárce
<code>bool</code>	Ano	<code>System.Boolean</code>	true nebo false	Reprezentuje pravdivost nebo nepravdivost
<code>Decimal</code>	Ano	<code>System.Decimal</code>	100 až 1028	96bitové číslo se znaménkem

Tabulka 5: Hodnotové datové typy 2 [16]

3.1.2.1 Jednoduchý příklad použití hodnotového datového typu

```
ushort promenna;
promenna = 10;
Console.WriteLine(promenna);
Console.ReadLine();
```

Na prvním řádku je zadán datový typ `ushort`, takže program očekává hodnotu v rozmezí 0-65535. Název proměnné je "promenna". Každá proměnná se musí před použitím inicializovat, což je provedeno na druhém řádku, kdy je do proměnné vložena hodnota 10. Třetí řádek už jen vypíše proměnnou na konzoli. Výstupem je samozřejmě pouze číslo 10.

Obdobně se pracuje se všemi hodnotovými typy, jediná výjimka je při vkládání typu char, kdy se musí znak, který chceme vložit do proměnné, dát mezi jednoduché uvozovky.

3.1.3 Odkazové datové typy

Odkazové datové typy lze rozdělit na uživatelské a vestavěné. Do vestavěných patří string a object. Označení string použijeme, chceme-li uložit textový řetězec. Object se označuje jako předek všech typů. To znamená, že lze instanci jakéhokoli typu převést do typu object. Použití typu object je obdobné jako při použití hodnotových datových typů. Při použití typu string je jedinou výjimkou to, že vkládaný textový řetězec musí být označen dvojitými uvozovkami. Uživatelské datové typy jsou třídy, pole, delegáti a rozhraní.

3.1.4 Rozdíl mezi třídou a strukturou

Třídy i struktury jsou v podstatě šablonami, podle nichž jsou tvořeny objekty. Každý objekt obsahuje jednak data a jednak metody, které slouží pro manipulaci s nimi. Třídou i strukturou se definuje, jaká data a jaké funkce může konkrétní objekt (nazývaný instancí) dotyčné třídy nebo struktury obsahovat. [14]

Hlavní rozdíl mezi třídou a strukturou spočívá v tom, že struktura se řadí mezi hodnotové a třída mezi odkazové datové typy. Obecně platí, že struktura se hodí spíše pro jednoduché datové struktury a třída pro složitější.

Struktura se tedy podobá třídě, vykazuje však následující odlišnosti:

- Třída je odkazovým (referenčním) typem, zatímco struktura je hodnotovým typem. Z toho vyplývá, že struktury se typicky používají k vyjádření jednoduchých typů.
- Třída plně podporuje dědičnost, zatímco struktura může dědit pouze ze třídy Object a je implicitně uzavřená.
- Třída může mít destruktory, zatímco struktura jej mít nemůže.
- Třída může definovat vlastní bezparametrický destruktory a inicializovat datové členy instancí, což struktura nedokáže. [15]

3.1.5 Vysvětlení pojmů public a static

Pokud je v programu uvedeno klíčové slovo public, je nutné k němu vždy přistupovat z instance typu. Vývojáři ovšem chtěli usnadnit práci programátora a to tím, že umožnili vytvořit prvek, ke kterému není třeba vytvářet instanci typu, ale lze k nim přistupovat na úrovni příslušné třídy nebo struktury. Právě k tomu slouží pojem static. Pro statické proměnné je životnost rovna celé době běhu programu. [17]

Zde platí ale podmínka, že statické struktury a třídy mohou operovat jen se statickými členy.

3.1.6 Použití private, protected, internal u tříd a struktur

Členy dané třídy nebo struktury musejí specifikovat svou úroveň "viditelnosti". Pokud definujete nějaký člen, aniž byste konkrétně specifikovali klíčové slovo určující úroveň jeho přístupnosti, automaticky bude privátní (private).

Modifikátory přístupu:

Public – Označí člen jako veřejný – je volně přístupný odkudkoli – tvoří rozhraní třídy.

Private – Označí metodu jako privátní, takže bude dostupná pouze v rámci třídy, která ji definovala.

Protected – Označí metodu jako chráněnou, takže se bude moci používat v rámci třídy, která ji definovala, a také ve všech odvozených třídách. Chráněné metody však nejsou přístupné z proměnné objektu.

Internal – Definuje metodu jako interní, takže je přístupná pro jakýkoli typ ze stejné assembly, ale nikoli mimo assembly. [16]

3.2 Delegáti

Delegát je typ definující podpis metody, takže instance delegátů mohou obsahovat a volat nějakou metodu nebo seznam metod, jež odpovídají danému podpisu. Deklarace delegáta se skládá z názvu a podpisu metody. Delegáti mohou vyvolávat více metod. [18]

Pro přidání metody k delegátovi slouží += a pro odebrání metody - =.

Příklad:

```
delegate void DELEGAT();
class SpustDelegata
{
    static void Main(string[] args)
    {
        DELEGAT delegat = null;
        delegat += new DELEGAT(CSharp);
        delegat += new DELEGAT(a);
        delegat += new DELEGAT(Delegat);
        delegat();
        Console.WriteLine();
        delegat -= new DELEGAT(a);
        delegat -= new DELEGAT(Delegat);
        delegat();
        Console.ReadLine();}

    public static void CSharp(){
        Console.Write("C# ");
    }
    public static void a() {
        Console.Write("a ");
    }
    public static void Delegat() {
        Console.Write("delegat.");
    }
}
```

V příkladu byl deklarován delegát s názvem “DELEGAT“. Při spuštění programu byla vytvořena jeho instancí s názvem “delegat“. Pomocí += byly přidány tři metody k delegátovi, následně byl delegát spuštěn a zobrazen výsledek na obrazovce. Poté se díky -= odebraly dvě metody, opět byl spuštěn delegát a nakonec byl znovu zobrazen výsledek na obrazovce. Jak lze očekávat obrazovka vypíše na prvním řádku větu “C# a delegate.“ a na druhém řádku jen slovo “C#”.

3.3 Základy vícevláknového programování

Operační systémy používají pro oddělení různých běžících aplikací procesy. Tyto procesy jsou striktně odděleny a mají svou vlastní přidělenou paměť, kterou nesdílí s ostatními procesy a každý proces může mít jedno nebo více vláken. Jestliže má proces více vláken, pak vlákna paralelně provádějí funkce v příslušném procesu a platí, že vlákna mezi sebou sdílí paměť.

V kapitole bude popsáno vícevláknové programování. Už z názvu vyplývá, že bude spuštěno více vláken v jednu chvíli. Všeobecně platí, že jeden procesor může v jedné chvíli

vykonávat jen jednu operaci. Velmi často je ale nutné, aby procesor zpracovával více operací v daný moment. Důvod je jednoduchý. Představme si situaci, kdy pověříme procesor, aby zpracovával náročné matematické operace. Kdybychom měli spuštěné jen jedno vlákno (tzv. hlavní vlákno) zjistili bychom, že program se jeví jakoby zamrzl a nereagoval by na žádné další příkazy do doby, než by matematickou operaci dokončil. Řešení zmíněného problému spočívá v tom, že pro matematicky náročnou operaci vytvoříme druhé vlákno a hlavní vlákno nám pak může sloužit pro vykonávání dalších příkazů. Procesor tedy bude chvíli vykonávat vedlejší a chvíli hlavní vlákno. V souvislosti s tímto je vhodné zmínit, že zde existují priority vláken. Pro nastavení priority se v C# používá následující příkaz "vlákno.Priority=ThreadPriority.prioritavlakna;". Obecně platí, že pokud procesor vykonává vlákno s vyšší prioritou, než jakou mají ostatní vlákna, pak musí ostatní vlákna počkat, dokud se zmíněné vlákno s vyšší prioritou nedokončí.

3.3.1 Příklad jednoduchého vícevláknového programu

```
class Program
{
    static void Main(string[] args)
    {
        Thread VLAKNO1 = new Thread(new
ThreadStart(vlakno1));
        Thread VLAKNO2 = new Thread(new
ThreadStart(vlakno2));
        VLAKNO1.Start();
        VLAKNO2.Start();
        Console.ReadLine();
    }
    public static void vlakno1()
    {
        for (int i = 0; i < 5; i++)
        {
            Console.WriteLine("Vlakno1");
            Thread.Sleep(1);
        }
    }
    public static void vlakno2()
    {
        for (int i = 0; i < 5; i++)
        {
            Console.WriteLine("Vlakno2");
            Thread.Sleep(1);
        }
    }
}
```

```
}
```

Z příkladu vidíme, že první vlákno spustí metodu “vlákno1“ a druhé vlákno spustí metodu “Vlákno2“. Ke spuštění dojde ve chvíli, kdy zadáme příkaz Start(). Výsledkem samozřejmě bude střídavý výpis vláken na obrazovku.

3.3.2 Synchronně pracující vlákna

Pojem synchronizace vláken se zavádí tehdy, chceme-li řídit přístup vláken k jednotlivým částem kódu (metoda, struktura, třída). Zde je vhodné zmínit příkaz lock, jehož použití zapříčiní, že k dané části kódu může v jednu chvíli přistupovat jen jedno vlákno. Další možností řízení přístupu je s pomocí příkazů Pulse a Wait, díky kterým lze zajistit, aby se pravidelně střídal přístup k danému kódu.

3.3.3 Asynchronně pracující vlákna

Jednoduchý příklad asynchronně pracujících vláken je uveden v předchozím příkladu. Pro asynchronní operace ovšem lze použít taky delegáty.

Runtime nabízí způsob, jakým lze libovolnou metodu volat asynchronně, přičemž se berou v úvahu přebírání návratových hodnot a odkazové a výstupní parametry zadané metodě. Jakmile kompilátor narazí na delegáta, pak vygenerovaná odvozená třída delegáta obsahuje tři klíčové metody (Invoke, BeginInvoke a EndInvoke).

Volání Invoke volá metodu synchronně a volající musí čekat než delegát skončí vykonávání.

Volání BeginInvoke volá delegáta se zadaným seznamem parametrů a pak se okamžitě vrací a volání je asynchronní. Volání EndInvoke přebírá návratovou hodnotu volané metody společně se všemi odkazovými a výstupními parametry, k jejíž změně mohlo dojít. [18]

Při programování v C# se můžete setkat s problémem, že pro přístup k ovládacím prvkům smí přistupovat jen to vlákno, které ho vytvořilo. Tím pádem z žádné vedlejší vlákno nesmí přistoupit k ovládacímu prvku. Řešením je použít již zmíněný příkaz Invoke, pomocí kterého se lze přesměrovat na autorské vlákno.

II. PRAKTICKÁ ČÁST

4 ÚVOD A ZÁKLADNÍ STRUKTURA VYTVOŘENÉHO POPLACHOVÉHO ZABEZPEČOVACÍHO SYSTÉMU

4.1 Úvod

Hlavní cíl praktické části spočívá ve vytvoření poplachového systému, který využije místo ústředny PZTS počítač či notebook. Výhody a nevýhody budou podrobně probrány v kapitole 6.3. Celý systém byl navrhnout tak, aby byl co nejvariabilnější, nejsrozumitelnější a na ovládání co nejjednodušší. Hlavní důraz byl však také kladen na maximální ochranu proti jakékoli sabotáži. V mé bakalářské práci v příloze III [19] je popsán způsob, jakým lze u současných analogových ústředen ochranu proti sabotáži obejít. Poplachový systém popisovaný v této diplomové práci však dokáže zmíněný postup detekovat a vyhlásí poplach. Jako rozhraní pro připojení analogových detektorů k počítači jsem zvolil Arduino Mega, což je programovatelný mikropočítač založený na uživatelsky jednoduchém hardware a software, který má 16 analogových vstupů a 54 digitálních pinů.

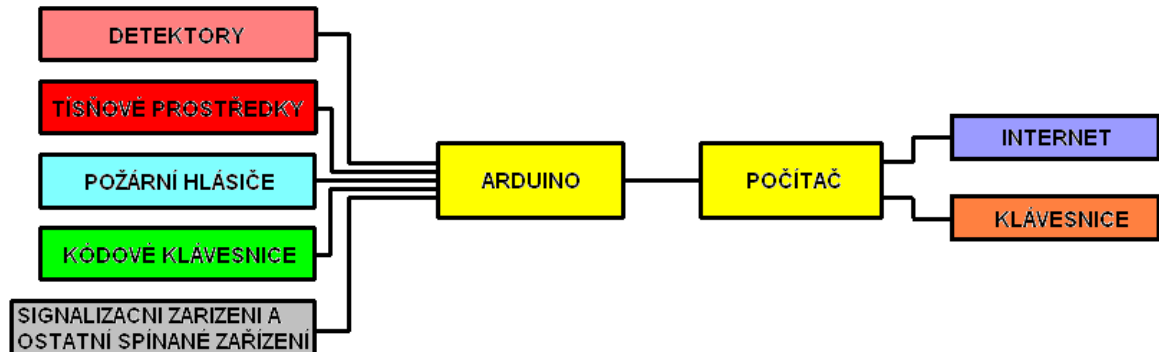
Je tedy využitelný jak pro snímání napětí pomocí vstupů, tak i pro ovládání světel, motorů a jiných spínaných zařízení. [20]

Digitální piny je možné nastavit na vstupy i výstupy a pomocí analogových vstupů lze měřit napětí v rozsahu 0-5V, které je díky 10-bitového A/D převodníku převedeno na číselnou hodnotu v rozsahu 0-1023. Při nastavení kteréhokoli digitálního pinu jako vstup jsme schopni rozeznat úroveň log 1, které odpovídá hodnota napětí přibližně 5V, a úroveň log 0, které odpovídá hodnota napětí přibližně 0V. Jestliže nastavíme některý z pinů 14-53 jako výstup, je na zmíněný výstup možné nastavit jen úroveň log 0 (0V) nebo úroveň log 1 (5V). U zbylých digitálních pinů nastavených jako výstupy pak lze využít i pulzně šířkovou modulaci, díky které můžeme zvolit jednu z 256 napěťových úrovní v rozsahu od 0V do 5V. U pinů 0 a 1 to ovšem není možné, protože jsou použity pro sériovou komunikaci. Pin 13 pak v programu slouží jako poplachový výstup v případě, že přestane fungovat komunikace Arduina s počítačem. Komunikaci počítače s Arduinem zprostředkovává USB rozhraní. K programování Arduina je použit jazyk C++ a program v počítači je vytvořen v C#.

4.2 Celková struktura fungování

Základní strukturu znázorňuje obr. 13. Počítač zde slouží pro zastřežení a odstřežení systému, programování ústředny, ukládání naprogramovaného nastavení ústředny, ukládání všech událostí, využívá také připojení k internetu jako poplachový přenosový systém, v případě využití zdvojení zóny ATZ rozezná ve které skupině detektorů byl vyhlášen poplach a v neposlední řadě pravidelně kontroluje dostupnost připojeného Arduina. Zmíněnou kontrolu dostupnosti lze ovšem vypnout jak ze strany počítače tak ze strany Arduina.

Arduino je zde využito pro připojení detektorů, požárních hlásičů, tísňových zařízení, ostatních spínaných zařízení a kódových klávesnic. Uvedení do stavu střežení nebo klidu je tedy možné jak s pomocí počítače, tak i ze zmíněné kódové klávesnice. Pro detektory jsou určeny analogové vstupy. Digitální vstupy/výstupy jsou určeny pro požární hlásiče, tísňová zařízení, kódové klávesnice a jakákoli jiná spínaná zařízení. Systém je možné rozdělit do čtyř subsystémů, kdy pomocí čtyř hesel je možné zastřežit nebo odstřežit určitou část subsystému.



Obrázek 13: Celková struktura poplachového zabezpečovacího systému

4.2.1 Základní popis programu v Arduinu

4.2.1.1 Struktura paměti EEPROM

Paměť se používá pro uchování dat po vypnutí příslušného zařízení. [21]

Do paměti EEPROM se ukládají základní data, která se po restartu nebo připojení elektrické energie nahrají do RAM paměti, čímž se zamezí nutnosti opětovného programování po výpadku elektrické energie. Jednotlivé data EEPROM se pak roztrídí do

příslušných struktur, ve kterých se nachází veškeré informace o naprogramování celého Arduina. V tabulce 6 je znázorněno rozložení EEPROM v závislosti na přidělených strukturách.

“Analog_Ovladani“ je pole 16-ti struktur, přičemž každá struktura náleží jednomu analogovému vstupu a obsahuje členy “Snimat“ (lze zde najít i informace o tom, jestli má být daná zóna aktivní), “Minimum“ (minimální klidová hodnota napětí), “Maximum“ (maximální klidová hodnota napětí) a “Casova_Aktivace“ (přidělená časová aktivace).

“Digital_Ovladani“ je také pole struktur, kdy každému digitálnímu vstupu/výstupu náleží jedna struktura, která obsahuje členy “Snimat“ (lze zde najít i informace o tom, jestli má být daná zóna aktivní), “Vstup_Nastaveni“ (informuje o nastavení zóny jako vstup a o klidové logické úrovni), “Vystup_Nastaveni“ (informuje o nastavení zóny jako výstup a o klidové logické úrovni) a “Casova_Aktivace“ (přidělená časová aktivace).

“Casova_Aktivace“ je pole 12 struktur. Má-li být analogový vstup nebo digitální vstup/výstup nastaven, musí mít ve své struktuře přidělenou jednu z 12 časových struktur.

Zmíněné struktury obsahují členy “Hodina_Deaktivace“, “Minuta_Deaktivace“, “Vterina_Deaktivace“, “Hodina_Aktivace“, “Minuta_Aktivace“ a “Vterina_Aktivace“, které slouží pro stanovení časového úseku, ve kterém nebude zóna aktivní i při zastřežení. Dále struktury obsahují členy “Minuta_Odchod“, “Vterina_Odchod“, “Minuta_Prichod“ a “Vterina_Prichod“ pro stanovení odchodového a příchodového času. Následuje člen “Aktiv_Zmena_Na“, “PWM“ a “Doba_Stavu2“.

Člen “Aktiv_Zmena_Na“ definuje změnu výstupu při poplachu, kdy se výstup změní na poplachový stav, nebo se změní aktuální stav příslušného výstupu, “PWM“ určí pulzně šířkovou modulaci a “Doba_Stavu2“ stanoví dobu poplachového stavu příslušných výstupů. V neposlední řadě zde lze najít i pole “Výstupy“, kde jsou uloženy výstupy, které se mají aktivovat při poplachu.

Poslední strukturou je struktura s názvem “Ostatni“, která obsahuje členy “DobaOpetovneReakce“, “Hlidani_Pc“, “Eeprom_Heslo“, “Heslo_Zastrez1“, “Heslo_Zastrez2“, “Heslo_Zastrez3“ a “Heslo_Zastrez4“. “DobaOpetovneReakce“ obsahuje časový interval, po který se má pravidelně kontrolovat komunikace s počítačem a pomocí “Hlidani_Pc“ lze zjistit, jestli je aktuálně spuštěné hlídání počítače a nastavení výstupu v případě chyby komunikace. Ve členech “Eeprom_Heslo“, “Heslo_Zastrez1“, “Heslo_Zastrez2“, “Heslo_Zastrez3“ a “Heslo_Zastrez4“ jsou uložena příslušná hesla pro

přístup k paměti EEPROM, zastřežení a zóny, které se mají při zastřežení aktivovat nebo deaktivovat.

Paměť EEPROM je ovšem využita i mezi 951-980B a 1000-4000B. Na pozicích 951-980B je uložen e-mail a heslo k e-mailu. Po každém připojení k počítači se nahraje do Arduina z počítače celá paměť EEPROM, a tím i e-mail, na který se poplachová informace posílá. Využití paměti nad 1000B spočívá v uložení všech poplachových stavů.

Adresa [B]	1-64	101-308	401-748	801-950
Struktura	Analog_Ovladani	Digital_Ovladani	Casova_Aktivace	Ostatni

Tabulka 6: Rozložení paměti EEPROM

4.2.1.2 Hlavní část programu

Hlavní část programu se nachází ve funkcích setup a loop. Zdrojový kód ve funkci setup se provede po každém zapnutí nebo restartu Arduina. Ihned po ukončení funkce setup se cyklicky opakuje kód ve funkci loop. Na obr.14 je možné vidět, že program pracuje se čtyřmi hlavními strukturami (Kontrola_Komunikace, ArduinoEeprom, Analog_Kontrola a Digit_Kontrola) a jednou třídou s názvem Čas. Ve funkci setup se příkazem “Seriál.begin(115200);“ nastaví rychlost přenosu po sériové lince. Následuje nastavení pinu 13 jako výstupní. Jedná se o poplachový výstup v případě, když přestane fungovat komunikace mezi počítačem a Arduinem. Následuje zkopírování paměti EEPROM, čímž se získají základní informace potřebné pro fungování celého programu bez nutnosti opětovnému programování při výpadku elektrické energie. Příkazem “Arduino.kontrola2();“ se nastaví implicitní hodnoty všech kódů. Nakonec je provedena inicializace proměnných ve strukturách Analog_Kontrola, Digit_Kontrola a Kontrola_Komunikace.

První příkaz ve funkci loop je “Cas.Cas();“, který slouží pro inicializaci času v Arduinu. Následuje kontrola analogových vstupů, digitálních vstupů/výstupů, kontrola dat na sériové lince a kontrola pravidelné komunikace s počítačem.

```

void setup() {
  Serial.begin(115200);
  pinMode(13, OUTPUT);
  ArduinoEeprom.Nahrej_Z_Pameti();
  ArduinoEeprom.kontrola2();
  Analog_Kontrola.Analog_Poprve_Zapnuto();
  Digit_Kontrola.Digit_Poprve_Zapnuto();
  Kontrola_Komunikace.Inicializuj();
}
void loop() {
  Cas.Cas();
  Analog_Kontrola.Analog_Pin_Kontrola();
  Digit_Kontrola.Digit_Pin_Kontrola();
  Kontrola_Komunikace.Kontrola_Komunikace();
  Kontrola_Komunikace.PcPoplach();
}

```

Obrázek 14: Základní popis programu v Arduinu

4.2.2 Komunikační protokol v nezastřeženém stavu

V nezastřeženém stavu platí, že Arduino odpovídá pouze na příkazy z počítače. Komunikaci znázorňuje obr. 15, ze kterého lze vyčíst, že počítač nejprve vyšle požadavek, Arduino ho přijme, zpracuje, odpoví na něj a počítač nakonec přijme odpověď. Pro komunikaci je zvolena maximální rychlost 115200 bd, aby byla doba přenosu dat co nejkratší. Všechny požadavky počítače a odpovědi Arduina v nezastřeženém stavu jsou znázorněny v tabulce 7 a 8.



Obrázek 15: Komunikace v nezastřeženém stavu

4.2.2.1 Požadavek

Požadavek se vždy skládá z identifikace a názvu funkce, kterou má Arduino spustit. Pro identifikaci slouží slovo “POCITAC“ a pro název funkce je použito následujících 6 bytů.

Jestliže požadavek obsahuje také data pro nahrání do paměti EEPROM, pak za identifikací a funkcí následuje heslo pro přístup k EEPROM, zmíněná data a vše je vždy zakončeno slovem “KONECKOM“, které pak Arduino následně zkontroluje pro ověření správnosti komunikace. Je zde ovšem také možnost změny času v Arduinu, kdy se první provede identifikace počítače, následuje název funkce a dva byty, které identifikují hodinu a minutu, na kterou se má čas změnit. Heslo k EEPROM ani zakončení se v příkazu nevyskytuje. V případě, že chceme změnit libovolné heslo, je hierarchie požadavku

následovná: identifikace počítače, název funkce, původní heslo, nové heslo a zakončení komunikace.

4.2.2.2 Příjem a zpracování

Program v Arduinu zde reaguje na písmeno "P" na sériové lince. Jakmile je písmeno přijato, čeká se 8ms, ve kterých se očekává příjem 13 bytů pro identifikaci počítače a zjištění funkce, která se má spustit. Jestliže spuštěná funkce očekává další data ze sériové linky, pak platí v případě příjmu od 3 do 11 bytů další zpoždění 10ms, v případě příjmu nad 11 bytů je zpoždění 20 ms. Zmíněné zpoždění je použito z toho důvodu, aby už všechna data stihly dorazit k Arduinu. Maximálně lze nahrát 100 bytů v jednom cyklu, pak ovšem probíhá i kontrola zakončení komunikace. Následuje zpracování požadavku a odpověď. Při zpracování ovšem může dojít k dalšímu zpoždění, které je způsobeno nahráváním dat.

4.2.2.3 Odpověď

Odpověď obsahuje identifikaci "ARDUINO", příslušnou spuštěnou funkci, oznámení o provedené změně nebo důvodu neprovedení změny a nakonec kontrolu ukončení s pomocí "KONECKOM".

4.2.2.4 Příjem odpovědi

Od vyslání požadavku čeká počítač zpravidla 250ms na odpověď. Jestliže do této doby odpověď nepřijde, je komunikace považována za chybnou. Jestliže se přenáší data pro zápis do EEPROM paměti do velikosti 11 bytů, pak počítač čeká 500ms a v případě, že je množství dat pro zápis do EEPROM paměti do 100 bytů, tak počítač čeká na odpověď 700ms.

Jiná situace ovšem nastává při mazání celé paměti v Arduinu, která je určená pro záznam událostí. Zde se čeká na odpověď 13,5s.

4.2.2.5 Požadavky a odpovědi

V tabulkách 7, 8 a 9 slouží zkratka “KK“ jako zkratka pro zakončení komunikace “KONECKOM“.

Úloha	Požadavek	Odpověď
Zjištění připojení Arduina	POCITACKDOJSI	ARDUINOMEGAKK
Zkopírování EEPROM do PC	POCITACNAHREJ	ARDUINONAHREJ+1000B+KK
Zjištění napětí na vstupech	POCITACZJISTI	ARDUINOZJISTI+68B+KK
Zjištění času v Arduinu	POCITACPOSCAS	ARDUINOPOSCAS+3B+KK
Změna času v Arduinu	POCITACNAHCAS+2B	ARDUINOOK+KK ARDUINOCHYBAKOMUNIKACE+KK
Načtení historie z Arduina	POCITACHISTOR	ARDUINOHISTOR+3000B+KK

Tabulka 7: Požadavky a odpovědi bez kontroly zakončení ze strany Arduina

Úloha	Požadavek	Odpověď
Změna hesel pro zastřežení	POCITACHESLOZ+12B+KK POCITACHESLO2+12B+ KK POCITACHESLO3+12B+ KK POCITACHESLO4+12B+ KK	ARDUINOOK+KK ARDUINOCHYBAKOMUNIKACE+KK
Nahrání paměti 0-800 [B]	POCITACEEPROM+heslo+100B+KK	ARDUINOHESLOUZEXISTUJE+KK ARDUINOSPATNEHESLO+KK
Nahrání paměti 801-980 [B]	POCITAEEPRO2+10B+KK	
Smazání historie v Arduinu	POCITACVYMAZH+6B+KK	ARDUINOSPATNEHESLO+KK ARDUINOOK+KK ARDUINOCHYBAKOMUNIKACE+KK

Tabulka 8: Požadavky a odpovědi s kontrolou zakončení ze strany Arduina

4.2.3 Komunikační protokol v zastřeženém stavu

Ve chvíli, kdy je program zastřežen, musí být program v počítači schopen v jednu chvíli přijímat data ze sériové linky a zároveň obsluhovat další příkazy pro zastřežení nebo odstřežení. Navíc program také umožňuje pravidelnou kontrolu komunikace. Jestliže program zacyklíme do jedné smyčky, tak nebude schopen obsluhovat další příkazy uživatele a bude se jevit, jakoby zatushl. Řešení je použití více vláken. V zastřeženém stavu program využívá jedno hlavní vlákno pro splnění příkazů pro zastřežení, odstřežení, spuštění nebo zastavení kontroly komunikace. Dále program využívá dvě vedlejší vlákna, kde jedno vedlejší vlákno zajišťuje pravidelnou kontrolu komunikace a druhé vlákno slouží pro příjem dat ze sériové linky. V programu je také použito časování ze strany Arduina i počítače, jinak by se totiž mohlo stát, že by vznikla chyba komunikace a to tak, že by byl v jednu chvíli vyslán požadavek na kontrolu komunikace a zároveň by uživatel mohl vyslat požadavek na zastřežení, odstřežení nebo zapnutí či vypnutí vzájemného hlídání. Obdobně by se mohlo stát, že by Arduino vyslalo odpověď na pravidelnou komunikaci a zároveň by některá zóna vyslala poplach. Program v C# očekává jen jednu odpověď nebo informaci o poplachu během 18 ms, z čehož vyplývá, že Arduino nesmí vyslat poplachový stav a příslušnou obslužnou informaci v kratším čase než je zmíněných 18ms, což je ošetřeno již zmíněným časováním. Jednotlivé funkce, používané v zastřeženém stavu jsou znázorněny v tabulce 9. Zpoždění Arduina je vždy 25ms před vysláním dat na linku a 75ms po vyslání. Program v C# si pravidelně kontroluje, jestli je linka volná a může vyslat informaci. Po vyslání jakéhokoli požadavku linka není volná a každý další požadavek pak čeká 250ms než může být vyslán. To umožní zpracování předchozího požadavku bez kolizí a je tím také zaručena dostatečná stabilita celého systému.

Posledním problémem je nemožnost vedlejšího vlákna přistupovat k ovládacím prvkům, jelikož je přístup povolen pouze z hlavního vlákna. Tento nedostatek byl vyřešen příkazem `Invoke`, který přesměruje vlákno na autorské.

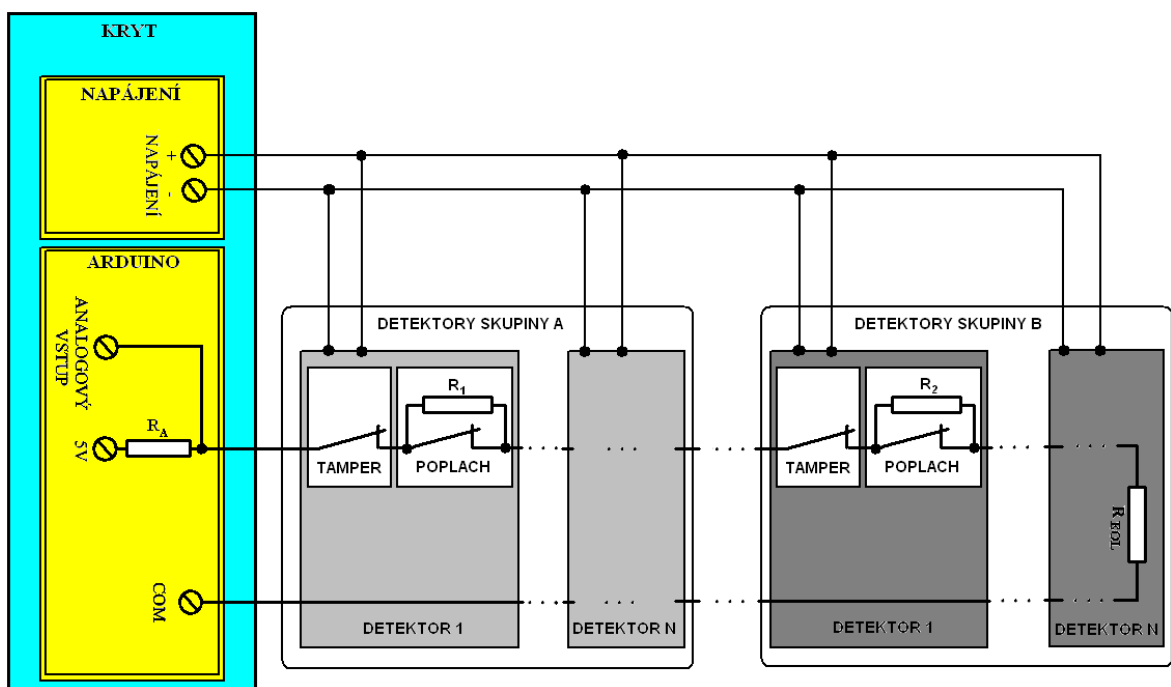
Úloha	Požadavek	Odpověď
Zapnout nebo vypnout hlídání	POCITACODEZV2+7b+KK	ARDUINOHLIDAMPORTTTKK ARUINONEHLIDAMPORTRK ARDUINOCHYBAKOMUNIKACEKK
Zastřežení nebo odstřežení	POCITACHLIDAT+6b+KK	ARDUINOZASTREZENO01KK ARDUINOODSTREZENO01KK ARDUINOZASTREZENO02KK ARDUINOODSTREZENO02KK ARDUINOZASTREZENO03KK ARDUINOODSTREZENO03KK ARDUINOZASTREZENO04KK ARDUINOODSTREZENO04KK ARDUINOSPATNEHESLOKK ARDUINOCHYBAKOMUNIKACEKK
Kontrola komunikace	POCITACODEZVAKK	ARDUINOODEZVAREAKCEKK

Tabulka 9: Požadavky a odpovědi v zastřeženém stavu

4.2.4 Způsob připojení detektorů

U klasických ústředěn spočívá princip vyhodnocení zóny v měření odporu smyčky. U Arduina je využit princip, který je znázorněn na obr. 16. Pomocí analogového vstupu se měří napětí na odporu R_A , jehož hodnota je $2k\Omega$.

Jakmile některý z detektorů rozpne kontakt, změní se napětí na odporu R_A a vyhlásí se poplach. Program je také schopný rozlišit, ve které skupině detektorů byl vyhlášen poplach, přičemž očekává hodnotu odporu $R_1=1k\Omega$ a $R_2=2k\Omega$ a hodnotu zakončovacího odporu $1k\Omega$. Obdobný princip lze využít pro všechna zapojení.



Obrázek 16: Způsob připojení detektorů ke smyčce typu ATZ

5 MANUÁL PRO PROGRAMOVÁNÍ

V této kapitole je podrobně vysvětleno, jakým způsobem se programuje a ovládá Arduino.

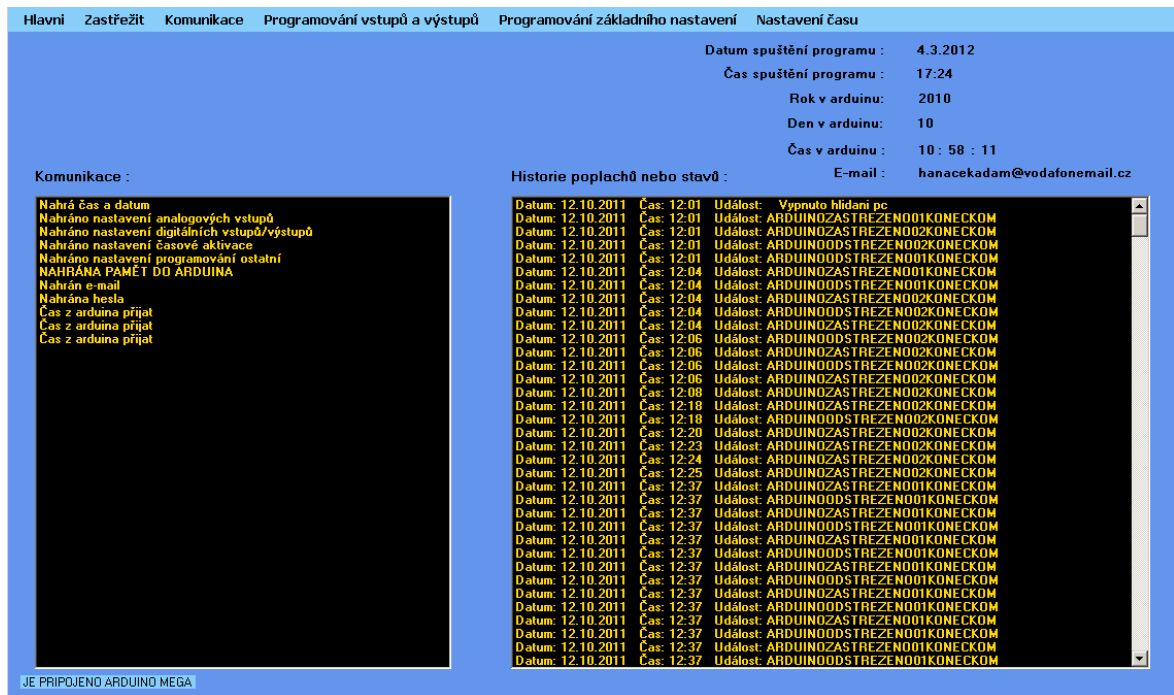
Po spuštění programu se otevře okno, které je znázorněno na obr. 17. Pokud je Arduino připojeno, načte se paměť EEPROM do počítače. Díky tomu je možné v programu upravovat aktuální nastavení Arduina. Lze však také připojit Arduino, vynulovat paměť a začít programovat od začátku. Po dokončení programování pak stačí pouze nahrát program do EEPROM paměti v Arduinu. Poslední možností je připojit Arduino, otevřít uložené nastavení a nahrát do Arduina.

Celý program se tedy nahrává do EEPROM paměti, ze které se po restartu nebo změně programu načtou aktuální data do RAM paměti a pomocí zmíněných dat funguje následně celý zabezpečovací systém. O připojení Arduina informuje text, který je zobrazen vlevo dole. Zároveň nás o aktuálním připojení informuje hlavní nabídka, která je bílá, jestliže Arduino není připojeno a modrá v případě, že je připojeno.

Při pohledu na hlavní okno vidíme dva listboxy. Listbox vlevo vypisuje již provedené operace a Listbox vpravo slouží jako výpis historie poplachů nebo stavů poplachového systému (zastřeženo, odstřeženo, spuštěna nebo zastavena funkce hlídání). Při náhledu na historii pak uvidíme datum s přesným časem poplachového stavu, danou událost, zónu poplachu a přesnou hodnotu poplachového stavu na dané zóně.

Dále lze v hlavním okně vpravo vidět datum a čas spuštění programu, datum a čas v Arduinu a v neposlední řadě je zde i e-mail, na který se posílají poplachové informace.

Jednotlivé položky hlavní rolovací nabídky budou popsány v další části kapitoly.



Obrázek 17: Hlavní okno

5.1 Nabídka “Hlavní“

Z obr. 18 je zřejmé, že v nabídce “Hlavní“ můžeme načíst historii zastřežení, odstřežení, hlídání a poplachů. Historie poplachů se ukládá do počítače i Arduina. Pro vymazání historie z Arduina je samozřejmě zapotřebí znát heslo pro přístup k paměti EEPROM.

Nabídka však především slouží pro ukládání a otevírání jednotlivých nastavení. Uložit nebo otevřít je možné buď programování vstupů a výstupů, programování základního nastavení nebo je zde i možnost otevření nebo uložení celého nastavení. Při výběru kterékoli možnosti se otevře okno, ve kterém zvolíme název souboru a pozici, kam se má soubor uložit nebo odkud se má otevřít. Při ukládání se místo všech hesel uloží jen hodnoty 255, aby nebylo možné přijít na žádné heslo.



Obrázek 18: Nabídka “Hlavní“

5.2 Nabídka “Zastřežit“

Pokud v hlavním okně klikneme na “Zastřežit“, objeví se nové okno, které je znázorněno na obr. 19. I zde se nachází listbox, který slouží pro zobrazení komunikace s Arduinem. Vpravo nahoře můžeme vidět dvě textové pole, které se využívají pro zastřežení a odstřežení. Jestliže chceme zastřežit pomocí níž položeného textového pole, musíme pro potvrzení kliknout na tlačítko “Zastřežit“. Výše položené textové pole slouží pro zastřežení nebo odstřežení z jiné místnosti, přičemž pro potvrzení stačí zmáčknout enter. O případném zastřežení, odstřežení a špatně zadaném heslu jsme informováni akusticky. Jestliže zadáme třikrát špatné heslo, zablokuje se klávesnice na 10s a uživatel musí čekat na odblokování. Ovšem po následném odblokování má už uživatel jen jeden pokus na odstřežení, protože je program v téhle části navrhnut tak, aby splňoval 4. stupeň zabezpečení dle ČSN EN 50131-1. Program má čtyři subsystemy, přičemž každý lze zastřežit 6-ti místným heslem. O aktuálních stavech subsystemů jsme informováni pomocí textů umístěných v pravé části okna. Vlevo dole pak můžeme spustit tzv. “hlídání“, což je spuštění vzájemné kontroly komunikace jak ze strany Arduina, tak ze strany počítače. Pro spuštění hlídání ovšem neslouží heslo pro zastřežení, ale heslo určené pro přístup k paměti EEPROM.

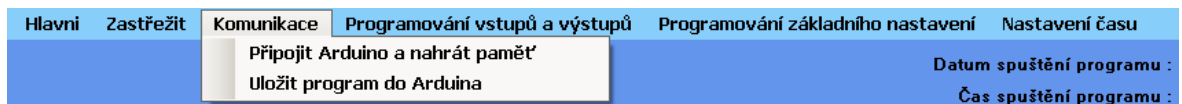
Jak si lze také všimnout, okno nelze minimalizovat, maximalizovat ani klasicky zavřít křížkem. Je to proto, že se spuštěním okna spustily také dvě vedlejší vlákna, která by se zavřením okna klasickým způsobem neukončily a komunikace by pak nefungovala správně. Okno lze tedy zavřít pouze tlačítkem ukončit.



Obrázek 19: Nabídka “Zastřežit“

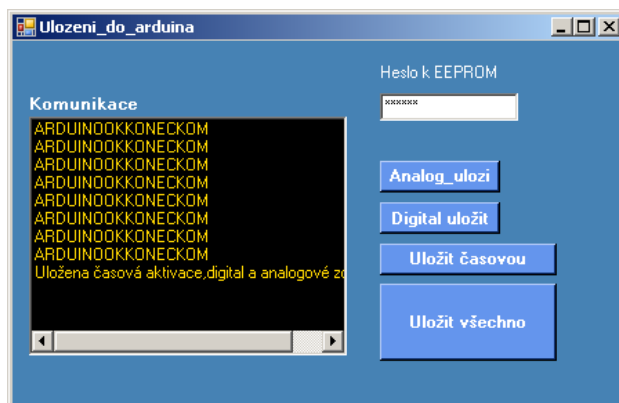
5.3 Nabídka “Komunikace“

Pomocí nabídky “Komunikace“ můžeme do Arduina uložit aktuální nastavení nebo připojit Arduino. Při připojení však zároveň dojde k nahrání EEPROM do počítače a tím se aktuální nastavení smaže. Jestliže Arduino nebylo připojeno před programováním, je proto vhodné si aktuální nastavení programu uložit ještě před připojením.



Obrázek 20: Nabídka “Komunikace“

Na obr. 21 je znázorněno okno, které se otevře kliknutím na “Uložit program do Arduina“. Pro uložení je nutné zadat heslo pro přístup k paměti EEPROM a kliknout na příslušné tlačítko. Je možné vybrat si uložení nastavení analogových vstupů, digitálních vstupů či výstupů, časové aktivace nebo uložení všech možností.

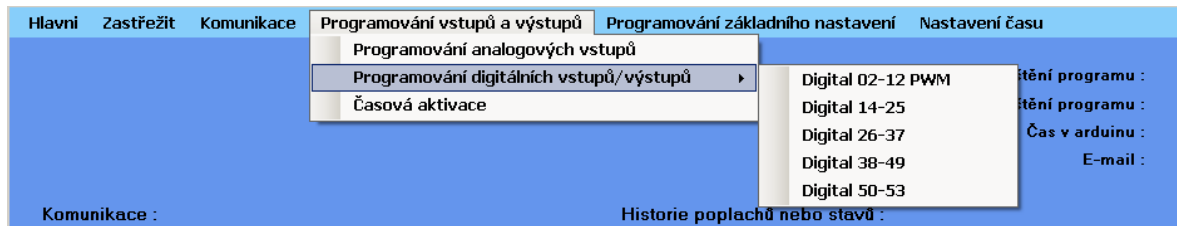


Obrázek 21: Uložení nastavení do Arduina

5.4 Nabídka “Programování vstupů a výstupů“

Tato nabídka slouží pro nastavení daných pinů jako vstup nebo výstup a přiřazení a nastavení příslušné časové aktivace. Na obr. 22 si lze všimnout, že digitální vstupy jsou rozděleny do 5 částí. Digitální piny 0 a 1 nebyly použity, protože jsou určeny pro komunikaci po sériové lince. Pin 13 také nemůžeme nastavit, protože je použit jako poplachový výstup při přerušení komunikace s počítačem. Ať už zadáme programování analogových vstupů nebo digitálních vstupů/ výstupu, vždy budeme vzápětí muset zvolit

časovou aktivaci. Těchto aktivací je celkem 12 a platí, že jakmile některou použijeme pro analogový vstup, nelze ho pak použít pro digitální vstup/výstup a naopak.



Obrázek 22: Nabídka “Programování vstupů a výstupů“

5.4.1 Programování analogových vstupů

Na obr. 23 je znázorněno programování analogových vstupů. U každé zóny je třeba zadat minimální a maximální hodnotu klidového stavu, který vychází z logiky zapojení. Vstupy měří napětí v rozsahu 0-5V, které jsou převedeny na číselné hodnoty od 0 do 255. Logika zapojení je vysvětlena v kapitole 4.2.4. Pro zjištění aktuální napěťové úrovně stačí kliknout na tlačítko “Nahrát aktuální hodnoty vstupu“, díky čemuž získáme zmíněnou hodnotu v rozmezí od 0 do 255 v dané chvíli a je možné zvolit toleranci, ve které bude klidový stav. Nakonec je třeba zadat časovou aktivaci u příslušných zón a kliknout na “ULOŽ“. Po kliknutí na “Zobrazit časovou aktivaci“ se otevře nové okno, kde lze nastavit základní vlastnosti dané zóny. Zmíněná časová aktivace je probrána v kapitole 5.4.1.1. Pro možnost smazání aktuálního nastavení slouží tlačítko “Vymazat_pole“, umístěné vpravo dole a pro přehledný náhled na časové aktivace stačí kliknout na “Všechny časové aktivace“.

Analogové vstupy

	Minimální hodnota	Maximální hodnota	Časová aktivace	
Zona 0:	148	200	NEZADÁNO	Zobrazit časovou aktivaci
Zona 1:	148	200	NEZADÁNO	Zobrazit časovou aktivaci
Zona 2:	148	200	NEZADÁNO	Zobrazit časovou aktivaci
Zona 3:	148	200	NEZADÁNO	Zobrazit časovou aktivaci
Zona 4:	148	200	NEZADÁNO	Zobrazit časovou aktivaci
Zona 5:	148	200	NEZADÁNO	Zobrazit časovou aktivaci
Zona 6:	148	200	NEZADÁNO	Zobrazit časovou aktivaci
Zona 7:	148	200	NEZADÁNO	Zobrazit časovou aktivaci
Zona 8:	148	200	NEZADÁNO	Zobrazit časovou aktivaci
Zona 9:	148	200	NEZADÁNO	Zobrazit časovou aktivaci
Zona 10:	148	200	11	Zobrazit časovou aktivaci
Zona 11:	148	200	11	Zobrazit časovou aktivaci
Zona 12:	148	200	11	Zobrazit časovou aktivaci
Zona 13:	148	200	11	Zobrazit časovou aktivaci
Zona 14:	148	200	11	Zobrazit časovou aktivaci
Zona 15:	148	200	11	Zobrazit časovou aktivaci

JE PŘIPOJENO ARDUINO MEGA

ULOŽENO

Obrázek 23: Programování analogových vstupů

5.4.1.1 Nastavení časové aktivace u analogového vstupu

Při nastavování vstupu u příslušné časové aktivace musíme zaškrtnout, že chceme nastavit vstup. Vlevo nahoře je možné nastavit odchodovou vteřinu a minutu, čímž docílíme zpoždění zóny, které majitel potřebuje pro odchod z daného objektu. Dále můžeme nastavit příchodové zpoždění, které poskytne čas na zadání kódu a odstřežení při příchodu do domu. Obě výše zmíněné zpoždění se používají jen tam, kde je zpoždění nutné, ostatní zóny se nastaví jako okamžité. Jedna z nejdůležitějších kolonek je “ZMENA NA“. V případě že chceme nastavit vstup, vždy zde musí být číslo 1 nebo 2. Pokud zadáme číslo 1, pak se při poplachu zapne snímání u vstupů nebo se sepne druhý stav. Pokud zadáme číslo 2, pak se změní aktuální stav. To umožní použít kódové klávesnice pro odstřežení objektu. Nakonec je už jen potřeba nastavit výstupy, které se mají aktivovat při poplachu, a kliknout na “ULOŽ“.

Další možností je nastavení časového úseku, ve kterém nebudou dané zóny aktivní. Tohoto nastavení dosáhneme zaškrtnutím podrobného nastavení a zadáním času deaktivace a aktivace zóny.

Obrázek 24: Nastavení vstupu u časové aktivace

5.4.2 Programování digitálních vstupů/výstupů

Jako vstup či výstup lze nastavit libovolnou digitální zónu, ovšem je třeba zadat, jestli bude v klidovém stavu úroveň logická 0 nebo logická 1, přičemž zadání jedné zóny jako vstup i výstup program nedovolí. Následně je nutné zvolit a nastavit časovou aktivaci pomocí tlačítka “Zobrazit časovou aktivaci“. Po dokončení programování stačí kliknout na “ULOŽ“. I v tomto případě máme možnost vymazat nastavení, nahrát aktuální hodnoty vstupu a ukázat nastavení všech časových aktivací. Programování všech ostatních digitálních vstupů/výstupů je obdobné.

Obrázek 25: Programování digitálních vstupů/výstupů

5.4.2.1 Nastavení časové aktivace u digitálních vstupů/výstupů

Pokud nastavujeme zónu jako vstup, je programování identické s programováním analogových vstupů. Jestliže programujeme zónu jako výstup, je třeba zadat příchodový a odchodový čas a dobu trvání stavu 2, což je časový interval v sekundách, po který bude trvat poplachový stav u příslušné zóny. Jestliže programujeme zónu 2-12, je nutné zadat i PWM, které slouží pro nastavení šířky impulzu. Výška impulzu (napětí) zůstává stejná.[22]

Obrázek 26: Časová aktivace pro digitální vstupy/výstupy

5.4.3 Časová aktivace

Kliknutí na časovou aktivaci umožní náhled a možnost změny nastavení u všech časových aktivací. Náhled na všechny časové aktivace znázorňuje obr. 27. Vlevo je zobrazeno, které časové aktivace jsou už použity a jestli jsou použity pro analogové vstupy nebo digitální vstupy/výstupy.

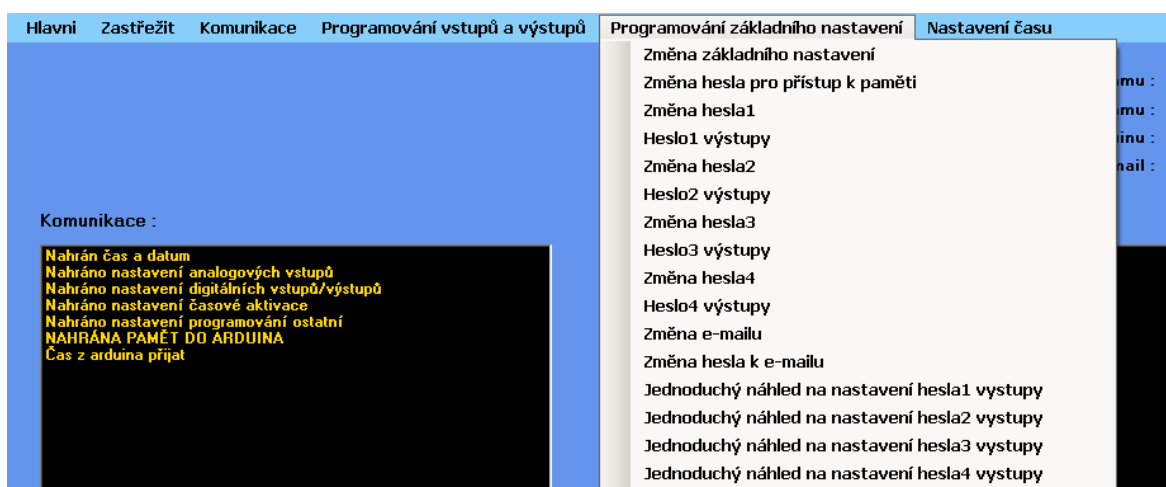
	HODINA DEAKTIVACE	MINUTA DEAKTIVACE	VTEŘINA DEAKTIVACE	HODINA AKTIVACE	MINUTA AKTIVACE	VTEŘINA AKTIVACE	MINUTA ODCHOD	VTEŘINA ODCHOD	MINUTA PRICHOD	VTEŘINA PRICHOD	ZMENA NA	PWM	DOBA STAVU 2
0: Digital	0	0	0	0	0	0	0	1	0	1	1	150	1
1: Nenastaven	0	0	0	0	0	0	0	0	0	0	0	0	0
2: Nenastaven	0	0	0	0	0	0	0	0	0	0	0	0	0
3: Nenastaven	0	0	0	0	0	0	0	0	0	0	0	0	0
4: Nenastaven	0	0	0	0	0	0	0	0	0	0	0	0	0
5: Nenastaven	0	0	0	0	0	0	0	0	0	0	0	0	0
6: Nenastaven	0	0	0	0	0	0	0	0	0	0	0	0	0
7: Nenastaven	0	0	0	0	0	0	0	0	0	0	0	0	0
8: Nenastaven	0	0	0	0	0	0	0	0	0	0	0	0	0
9: Nenastaven	0	0	0	0	0	0	0	0	0	0	0	0	0
10: Nenastaven	0	0	0	0	0	0	0	0	0	0	0	0	0
11: Analog	0	2	1	0	2	30	0	30	0	15	1	0	0

Obrázek 27: Všechny časové aktivace

5.5 Nabídka “Programování základního nastavení“

Nabídka slouží pro změnu základního nastavení, hesel, e-mailu a pro přiřazení jednotlivých výstupů k příslušným heslům. Jestliže uživatel heslo zapomeneme, stačí Arduino resetovat nebo odpojit napájení a hesla se změní na implicitní hodnoty. Implicitní hodnota hesla pro přístup k paměti je “000000“, hesla1 “111111“, hesla2 “222222“, hesla3 “333333“ a hesla4 “444444“.

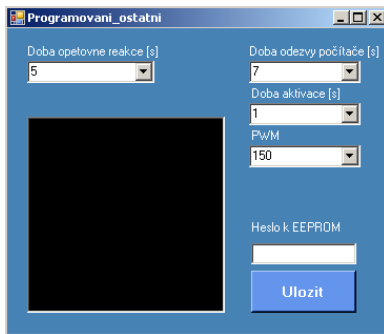
V nabídce „Programování základního nastavení“ se také nachází přehledné náhledy, díky kterým lze snadno zjistit, jestli jsme při programování udělali chybu.



Obrázek 28: Programování základního nastavení

5.5.1 Programování základního nastavení

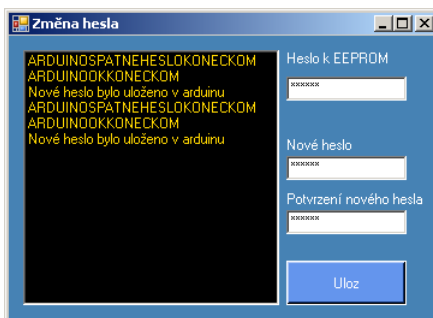
V programování základního nastavení je možné nastavit dobu opětovné reakce, odezvy počítače, dobu aktivace a PWM. Doba opětovné reakce nastaví časový interval neaktivity příslušné zóny po vyhlášení poplachu. V případě, že je v dané zóně vyhlášen poplach a narušitel je stále v objektu, tak bude poplachová informace vysílána nepřetržitě stále dokola. Abychom tomuto nepřetržitému vysílání zabránili, je možné nastavit zmíněnou dobu opětovné reakce. Další možností nabídky je doba odezvy počítače, která nastaví čas, podle kterého se periodicky vzájemně kontroluje dostupnost připojení mezi počítačem a Arduinem. Doba aktivace zase umožní nastavit dobu poplachového signálu v případě, že bude přerušena komunikace. Pro zmíněný účel je vyhrazen pin 13. Dále je v této nabídce také možné nastavit příslušné PWM. Nakonec stačí zadat heslo pro přístup k EEPROM a kliknout na tlačítko “Uložit“.



Obrázek 29: Programování základního nastavení

5.5.2 Změna hesla pro přístup k paměti, e-mailu a hesla k e-mailu.

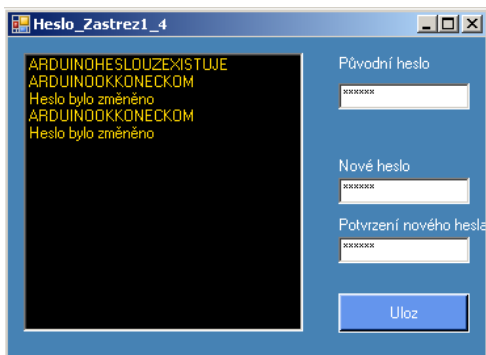
Změna je velmi jednoduchá - stačí zadat původní heslo k EEPROM, nové heslo, potvrzení nového hesla a kliknout na "Ulož". Na obr. 30 jsou znázorněny i odpovědi Arduina. E-mail zde slouží pro příjem poplachového signálu.



Obrázek 30: Změna hesla pro přístup k paměti a e-mailu

5.5.3 Změna hesel pro zastřežení a odstřežení

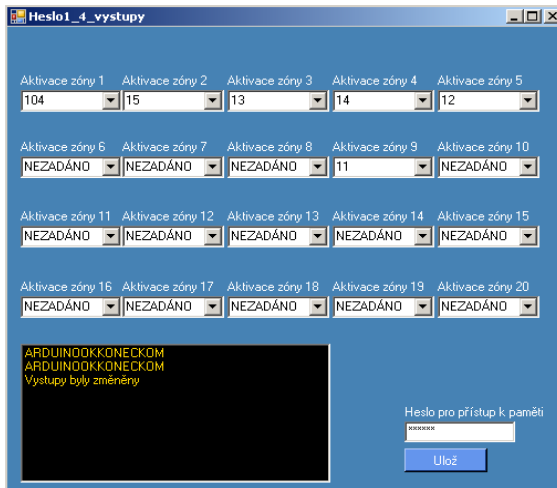
Změna se provede zadáním původního hesla, nového hesla, potvrzením nového hesla a kliknutím na "Ulož".



Obrázek 31: Změna hesel pro zastřežení a odstřežení

5.5.3.1 Přřazení výstupů k daným heslům

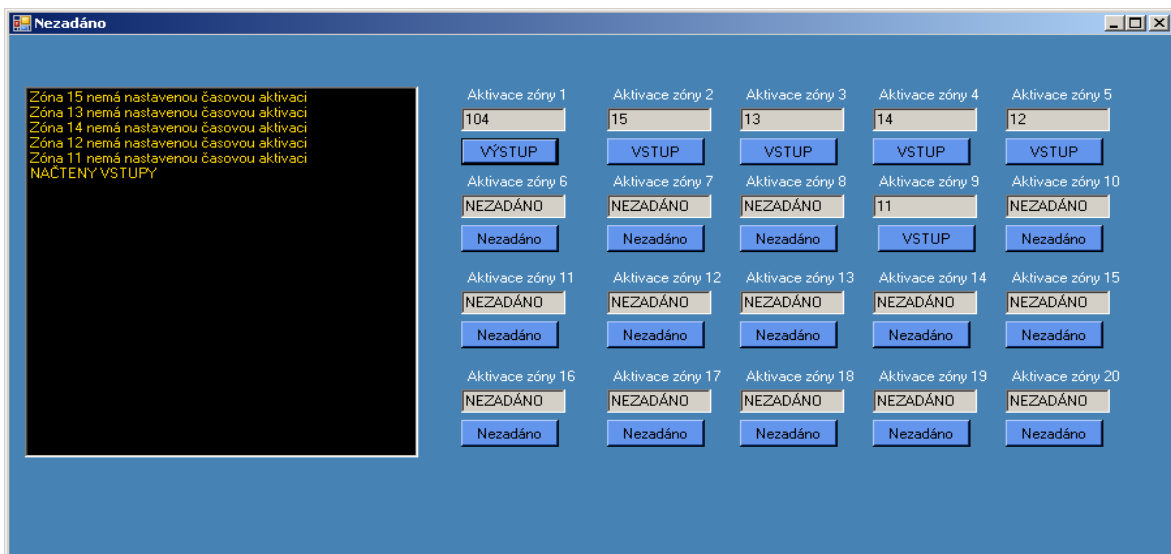
Jedná se o poslední úkon při programování jednotlivých zón. V následující tabulce zadáme vstupy nebo výstupy, které mají změnit svůj aktuální stav po zadání příslušného hesla. Pak už stačí jen zadat heslo pro přístup k paměti a kliknout na “Ulož”.



Obrázek 32: Přřazení výstupů k daným heslům

5.5.4 Přehledný náhled nastavení

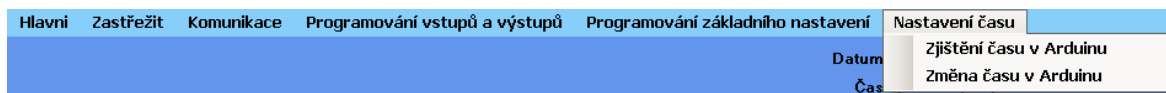
V přehledném náhledu můžeme snadno zjistit, jestli při programování nastala chyba. Jsou zde zobrazeny všechny zóny, které se aktivují po zadání hesla, přičemž lze u každé zóny vidět, jestli je nastavena jako vstup nebo výstup. Pokud klikneme na tlačítko u jakékoli zóny, objeví se příslušná časová aktivace. Na obr. 33 vlevo můžeme vidět případné chyby v nastaveném programu.



Obrázek 33: Přehledný náhled nastavení

5.6 Nabídka “Nastavení času“

Pokud klikneme na položku “Zjištění času v Arduino“, zobrazí se aktuální čas. Jestliže chceme čas změnit, stačí kliknout na položku “Změna času“ v Arduino, nastavit nový čas a nahrát čas.



Obrázek 34: Nabídka “Nastavení času“

5.7 Postup při programování subsystému

Při programování je nutné, aby se postupovalo přesně tak, jak je popsáno v následující kapitole, protože přidání výstupu u příslušné časové aktivace je možné až po nastavení daného výstupu jako výstup.

5.7.1 Připojení Arduina

Arduino lze připojit, pokud v položce “Komunikace“ klikneme na “Připojit Arduino a nahrát paměť“.

5.7.2 Nastavení výstupů

- Nejprve je třeba v položce “Programování vstupů a výstupů“ zadat “Programování digitálních vstupů/výstupů“ a následně kliknout na příslušnou pozici, kde se daný pin nachází.
- Pokračuje se nastavením daného pinu jako výstup s příslušnou klidovou logickou úrovní, zadáním časové aktivace a kliknout na “Ulož“.
- Vše se zakončí kliknutím na “Zobrazit časovou aktivaci“, zaškrtnutím výstupu, zadáním zpoždění, délky aktivace, pulzně šířkové modulaci a kliknutím na “Ulož“

5.7.3 Nastavení vstupů

- V položce “Programování vstupů a výstupů“ nejprve klikneme na “Programování analogových vstupů“
- U příslušných zón následně zadáme časovou aktivaci a klikneme na “Ulož“
- Nakonec klikneme na “Zobrazit časovou aktivaci“, zaškrtneme vstup, zadáme příchodový a odchodový čas, způsob změny při poplachu, výstupy, které se mají aktivovat a klikneme na “Ulož“

5.7.4 Uložit program do Arduina

- V položce “Komunikace“ vybereme “Uložit program do Arduina“
- Zadáme heslo pro přístup k EEPROM a klikneme na “Uložit všechno“

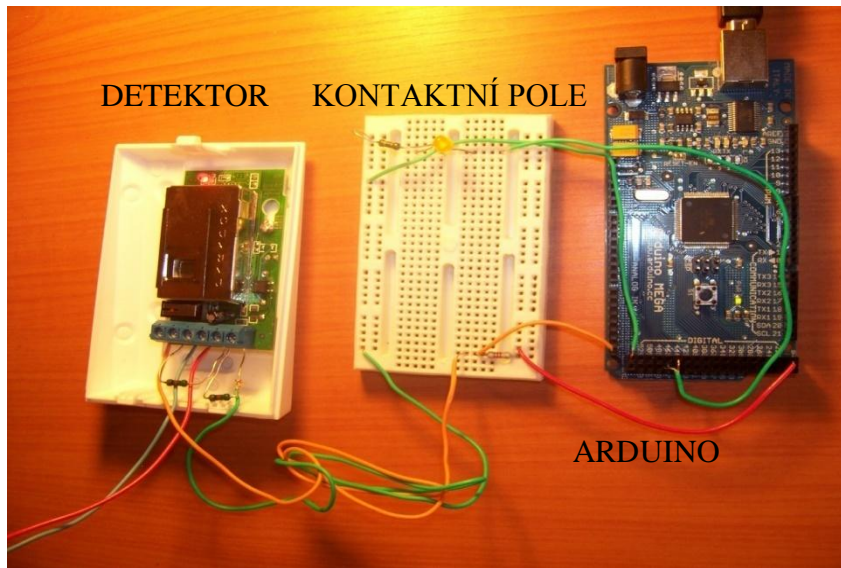
5.7.5 Přidat vstupy a výstupy k danému heslu

- V “Programování základního nastavení“ zvolíme “Heslo výstupy“
- Následně označíme zóny, které se aktivují nebo deaktivují po zadání hesla, zadáme heslo pro přístup k paměti a klikneme na “Ulož“

5.8 Praktický příklad připojení poplachového detektoru k navrženému systému

Při pokusu byl použit analogový PIR detektor PA-476 PRO PLUS od firmy Paradox Security Systems. Připojení detektoru k navrženému systému znázorňuje obr. 35 a vychází ze schématu, které je znázorněné v kapitole 4.2.4 obr. 16. Červený vodič je připojen k Arduinu na 5V a k odporu na kontaktním poli, jehož hodnota je $2k\Omega$. Pomocí oranžového vodiče připojeného ke zmíněnému odporu a k zóně 15 měří analogový vstup napětí příslušné smyčky. Smyčka dále pokračuje oranžovým vodičem přivedeným k detektoru ke kontaktu “Tamper“. Následují dva odpory o hodnotě $1k\Omega$. Odpor více vlevo slouží jako zakončovací odpor smyčky a pomocí druhého odporu je možné zjistit poplachový stav. Následuje zelený vodič připojený ke kontaktnímu poli, odkud je obvod pomocí dalšího zeleného vodiče uzemněn. Tím je smyčka ukončena.

K detektoru je dále připojen modrý a červený vodič pro napájení. Jako poplachový výstup byl zvolen digitální pin 44, přičemž poplach signalizuje led dioda. Digitální výstup i analogový vstup byl nastaven pomocí postupu, který je popsán v kapitole 5.7. Je zde nastaveno sekundové zpoždění zóny a klidový stav digitálního výstupu na úroveň log 1. Doba opětovné reakce je nastavena na 7 sekund. Následně dojde k zakrytí detektoru a zastřežení. Jakmile je stínění detektoru odkryto, vyšle se poplachový signál na SMS e-mail.



Obrázek 35: Připojení detektoru k poplachovému systému

6 DALŠÍ MOŽNOSTI ZDOKONALENÍ A POROVNÁNÍ S BĚŽNÝM POPLACHOVÝM SYSTÉMEM

Mezi hlavní výhody vytvořeného systému patří variabilita. Systém využívá 16 analogových vstupů a 51 digitálních vstupů/výstupů, u kterých lze nastavit délku poplachového signálu. Pro 12 digitálních výstupů je také možné nastavit pulzně šířkovou modulaci.

Jak už bylo zmíněno v kapitole 4.2, systém lze využít nejen k připojení poplachových detektorů, ale také k připojení tísňových zařízení a požárních hlásičů, kódových klávesnic a jakýchkoli spínaných zařízení. Celý systém lze naprogramovat tak, aby bylo možné řídit libovolným vstupem až 20 dalších vstupů nebo výstupů, které pak mohou ovládat libovolné výstupy, čímž lze provést zastřežení nebo odstřežení přímo ze vstupu Arduina. Možné způsoby zastřežení či odstřežení včetně doporučeného nastavení systému jsou popsány v následující kapitole. Ovšem ať již vybereme kterýkoli způsob, vždy je vhodné umístit Arduino do krytu se sabotážním kontaktem a záložní baterií. Maximální počet vstupů připojených k jednomu subsystému je 20. K výstupu je možné připojit libovolné spínané zařízení (například houkačky, blikáče, sirény, zamlžovací systém nebo i zařízení přenosového systému). Systém je navrhnout tak, aby byl pro poplachový přenosový systém využit počítač díky připojením k internetu, ovšem i přesto je lepší mít systém pro jistotu připojen ke dvěma přenosovým systémům, aby se minimalizovalo riziko nedoručení poplachové informace. Ve chvíli, kdy je vyhlášen poplach, vyšle počítač poplachovou informaci na e-mail, který je možné dále nastavit tak, že bude majitel informován prostřednictvím mobilního telefonu o každém novém e-mailu. Druhou variantou je využít SMS e-mail, díky kterému počítač posílá informaci přímo na mobilní telefon.

6.1 Možné způsoby zastřežení nebo odstřežení systému

Pro zastřežení a odstřežení lze využít přímo klávesnici u počítače, klávesnici připojenou k prodlužovacímu kabelu nebo libovolné zařízení připojené na vstup Arduina.

6.1.1 Využití klávesnice u počítače

Jestliže využijeme zastřežení pomocí klávesnice u počítače, musíme příslušnou zónu nastavit jako zpožděnou. Zde je vhodné nezadávat příliš dlouhý časový interval pro zadání kódu a umístit Arduino do místnosti, která je připojená k okamžité zóně. Zároveň je nutné spustit hlídání vzájemné komunikace mezi Arduinem a počítačem a nesmí se opomenout

připojit k výstupu u Arduina poplachové přenosové zařízení, které zajistí doručení poplachové zprávy při přerušení spojení s počítačem.

6.1.2 Využití klávesnice připojené k prodlužovacímu kabelu u počítače

Toto řešení považuji za nejvhodnější, jelikož bude klávesnice připojena k prodlužovacímu kabelu. Jestliže je třeba prodloužit kabel o více než 5 metrů, je ovšem nutné použít aktivní prodloužení, jinak by komunikace nemusela fungovat správně.

Využitím klávesnice připojené k prodlužovacímu kabelu u počítače získáme možnost nastavit jako zpožděnou pouze místnost s klávesnicí, přičemž počítač i Arduino zůstane připojeno k okamžité zóně. Nejvhodnější je použití numerické klávesnice, čímž zabráníme jakékoli nedovolené manipulaci s počítačem. Rezervní poplachová přenosová cesta zde není nezbytně nutná, protože v případě narušení místnosti s počítačem se okamžitě vyše poplachová zpráva. Ovšem i při zvolení tohoto řešení je vhodné využít náhradní poplachový přenosový systém.

6.1.3 Využití zařízení připojeného ke vstupu Arduina

Jestliže využijeme zařízení připojené přímo na vstup Arduina pro ovládání systému, tak lze zastřežení či odstřežení provést díky jakémukoli zařízení, které je schopné na svém výstupu generovat napětí minimálně 4 V. Tím se naskytne možnost ovládat systém i kódovými klávesnicemi, čipovými kartami, čipovými přívěšky nebo i dálkovým ovládáním. Program umožňuje vypnutí kontroly komunikace, takže v tomhle případě není nutné spojení s počítačem, ovšem je třeba si uvědomit, že se pak nebude kontrolovat běh Arduina, přijde se o poplachový přenosový systém využívající internet v počítači a nebude možná přesná identifikace místa poplachu. Z těchto důvodů je vhodné mít počítač připojen. Počítač i Arduino zde je v místnosti připojené k okamžité zóně. V případě připojeného počítače není náhradní poplachová přenosová cesta nezbytně nutná.

6.2 Finanční ohodnocení

Finanční ohodnocení je provedeno srovnáním ceny poplachového zabezpečovacího systému využívajícího klávesnici připojenou k prodlužovacímu kabelu s cenou běžného zabezpečovacího systému obdobných parametrů, která byla získána z ceníku firmy CLR s.r.o.

6.2.1 Finanční náklady základních prvků při využití poplachového zabezpečovacího systému popsaného v praktické části

Jednotlivé finanční náklady v závislosti na použitém způsobu zabezpečení jsou rozepsány v tabulce 10. Cena GSM komunikátoru značí možnost použití běžného komunikátoru, jehož cena se obvykle pohybuje od 500-2000 Kč. Zmíněný GSM komunikátor není nezbytně nutný při použití klávesnice připojené k prodlužovacímu kabelu nebo kódové klávesnice pro ovládání systému, ovšem z důvodu zvýšení zabezpečení náhradní přenosovou trasou byl zde také započítán k celkové ceně.

Zařízení	Klávesnice u počítače	Klávesnice připojená k prodlužovacímu kabelu	Kódová klávesnice
Arduino [Kč]	1230	1230	1230
GSM [Kč]	900	900	900
Prodloužení USB [Kč]	-	371	-
Numerická klávesnice [Kč]	-	126	-
Kódová klávesnice [Kč]	-	-	980
Celková cena [Kč]	2130	2627	3110
Počet analogových vstupů	16	16	16
Počet digitálních vstupů/výstupů	51	51	51

Tabulka 10: Finanční náklady na poplachový zabezpečovací systém s PC

6.2.2 Finanční náklady základních prvků při použití běžných zabezpečovacích systémů a cenové porovnání s navrženým systémem

Ceny jednotlivých komponentů jsou zprůměrovány v závislosti na daném výrobcí. Tabulka 12 znázorňuje celkový počet zón a výstupů u vybraných poplachových zabezpečovacích systémů. V tabulce 11 kolonka “Úspora při použití Arduina pro PZTS [%]“ znázorňuje, kolik procent z ceny ušetříme při použití poplachového systému navrženého v praktické

části, jestliže Arduino zastane jen funkci PZTS. Z tabulky je zřejmé, že při použití systému navrženého v praktické části lze ušetřit 77-85% z ceny za hlavní části poplachového systému v případě, že zastává jen funkci poplachové zabezpečovací a tísňové ústředny. Vzhledem k možnosti využití digitálních vstupů pro připojení požárních hlásičů je možné, aby navržený systém zastával funkci jak poplachové zabezpečovací tak i požární ústředny. Pak lze ušetřit 86-89% z ceny za hlavní části poplachového systému. Je třeba si ovšem uvědomit, že ve skutečnosti rozdíl není až tak velký, protože systém by bylo nutné certifikovat a v ceně navrženého systému také není započtena práce na vývoji softwaru.

	Jablotron JA80	Spectra SP5500	Digiplex EVO192	DSC POWER 1616
Ústředna [Kč]	1548	1788	3600	3828
Komunikátor [Kč]	6996	4434	4434	4260
Klávesnice [Kč]	2458	2280	5926	2117
Další vstupy [Kč]	744	1725	1725	599
Další výstupy [Kč]	606	1300	1558	1500
Cena PZTS celkem [Kč]	12352	11527	17243	12304
Úspora při použití Arduina pro PZTS [Kč]	9725	8900	14616	9677
Úspora při použití Arduina pro PZTS [%]	79	77	85	79
Cena PZTS+EPS celkem [Kč]	19252	18427	24143	19204
Úspora při použití Arduina pro PZTS+ EPS [Kč]	16625	15800	21516	16577
Úspora při použití Arduina pro PZTS + EPS [%]	86	86	89	86

Tabulka 11: Cena běžného poplachového zabezpečovacího systému včetně porovnání s poplachovým systémem navrženým v praktické části

	Jablotron JA80	Spectra SP5500	Digiplex EVO192	DSC POWER 1616
Celkový počet vstupů	14	13	16	14
Celkový počet výstupů	12	6	7	10

Tabulka 12: Celkový počet zón a výstupů u vybraného poplachového zabezpečovacího systému

6.3 Výhody a nevýhody navrženého systému v porovnání se současnými poplachovými zabezpečovacími systémy

Výhody:

- Nižší cena
- Vyšší komfort při programování
- Jednoduchost programování
- Velký počet vstupů i výstupů
- Možnost využití jako poplachové zabezpečovací i požární ústředny
- Možnost širokého využití i pro nepoplachové aplikace
- Lepší ochrana proti sabotáži poplachové zóny než u současných analogových ústředen
- Využití dvou poplachových přenosových cest ze zabezpečeného objektu
- Možnost vzájemné kontroly běhu Arduina i počítače

Nevýhody:

- Nutnost použití notebooku k programování
- Mimo možnosti vypnutí kontroly komunikace a použití kódové klávesnice nutnost použít počítač pro zastřežení a odstřežení
- Systém není certifikován
- Nemožnost bezdrátové nadstavby
- Nemožnost dalšího rozšiřování systému

ZÁVĚR

Teoretická část práce poskytuje celkový náhled na poplachový zabezpečovací a tísňový systém. Jsou zde podrobně probrány možné způsoby připojení detektorů, popis jednotlivých částí a v neposlední řadě také požadavky norem na současné poplachové zabezpečovací a tísňové systémy.

V praktické části bylo cílem navrhnout a realizovat poplachový zabezpečovací a tísňový systém, který by byl levnější než současné zabezpečovací systémy. Přínos práce tedy spočívá zejména v nižší ceně systému a tedy jeho větší dostupnosti široké veřejnosti. Snížení ceny je dosaženo využitím osobního počítače jako ústředny PZTS. Jako rozhraní pro připojení jednotlivých prvků PZTS k počítači bylo zvoleno Arduino Mega, na kterém je možné využít 16 analogových a 51 digitálních pinů. Počítač zde však neslouží pouze k programování a ovládání Arduina, ale vzhledem k možnosti využití připojení k internetu ho lze využít také jako poplachový přenosový systém. Při tvorbě systému byl kladen hlavní důraz na stabilitu, vysokou úroveň ochrany proti jakékoli sabotáži, vysokou variabilitu a jednoduchost programování pro uživatele. Vzhledem k tomu, že zmíněné Arduino Mega disponuje velkým množstvím vstupů, lze navržený systém využít nejen jako poplachový zabezpečovací a tísňový systém, ale také jako systém požární. Arduino pak pravidelně ověřuje komunikaci s počítačem, přičemž jestliže nastane poplach, vyšle počítači poplachový signál a zároveň aktivuje své výstupy. Navržený systém také umožňuje vypnout kontrolu komunikace s počítačem a připojení kódové klávesnice, díky čemuž může systém pracovat i autonomně bez počítače. Využití této možnosti však nepovažuji za vhodné, jelikož tak přijdeme o náhradní poplachový přenosový systém a o možnost kontroly běhu Arduina Mega počítačem.

V závěru práce lze nalézt cenovou kalkulaci, do které je zahrnuta také cena prodlužovacího kabelu ke klávesnici, cena samotné numerické klávesnice a cena náhradního poplachového přenosového zařízení připojeného k Arduinu Mega.

Z výsledku cenové kalkulace vyplývá, že při použití poplachového systému navrženého v praktické části je možné ušetřit 77-85% za hlavní části systému v tom případě, že bude navržený systém použit jen jako náhrada za ústřednu PZTS. V případě, že bude systém použit jako náhrada za PZTS i EPS, lze ušetřit až 86-89% z tržní ceny obdobného systému. Je však třeba zdůraznit, že do ceny navrženého systému není započtena certifikace, případný zisk z prodeje a cena za vývoj softwaru, tedy práce.

Mezi výhody navrženého systému patří nízká cena, jednoduchost programování, velký počet programovatelných vstupů a výstupů, možnost využití jako poplachové zabezpečovací i požární ústředny, možnost širokého využití i pro nepoplachové aplikace, velmi dobrá ochrana proti sabotáží poplachové zóny, využití dvou poplachových cest a možnost kontroly běhu Arduina Mega i počítače navzájem. Mezi nevýhody patří nutnost použití notebooku k programování, skutečnost, že systém není certifikován a nemožnost bezdrátové nadstavby a dalšího rozšiřování. Další nevýhodou může být nutnost použití počítače pro zastřežení a odstřežení poplachového systému v případě, že bude pro zmíněný účel použita klávesnice na notebooku či počítači nebo klávesnice připojená k prodlužovacímu kabelu.

Vzhledem ke komplexnosti, variabilitě a nízké ceně navrženého systému považuji cíl diplomové práce za splněný a pevně věřím, že bude možné výsledky práce využít i v praxi.

CONCLUSION

The Theoretical part of diploma thesis provides a general view of Intruder and Hold-up Alarm System. There are also analyzed possible ways of connection the detectors, description of individual parts and last but not least, the requirements of standards on current Intruder and Hold-up Alarm Systems.

The Practical part of diploma thesis was focused on design and implementation of Intruder and Hold-up Alarm System, which should have been cheaper than current security systems. The contribution of the thesis lies mainly in lower price of the system and therefore in its larger accessibility for general public. Price reduction is achieved by using a personal computer as a Central I&HAS. As an interface for connection of various components of I&HAS there was used Arduino Mega, which enables to use 16 analog and 51 digital pins. The computer is used not only for programming and control Andruino, but also as a Alarm Transmission System due to its possibility of internet connection. While the system was developing, the main emphasis was focused on stability, high level of protection against any sabotage, high variability and simplicity programming for users. Thanks to the fact, that mentioned Arduino Mega has large number of inputs, it is possible to use the developed system not only as a Intruder and Hold-up Alarm System, but also as a Fire Alarm System. Arduino periodically verifies communiacion with the computer and if any alarm occurs, Arduino sends an alarm signal to the computer and also activates its outputs. Developed system also allows to switch off control of the communication with the computer and connection of code keypad, what enables the system to work independently without computer control. However, I don't recommend this possibility, because we lose backup Alarm Transmission System and flow control Andruino Mega with the computer this way.

In conclusion of the thesis there is mentioned price calculation, which includes also the price of extension cable to the keypad, the price of numeric keypad and the price of backup alarm transmission device connected to Arduino Mega. The result of price calculation shows, that developed Alarm Security System can enable to save 77-85% for the main part of system in case it will be used as a replacement of Central I&HAS. In case the system will be used as a replacement of I&HAS and also as an Electronic Fire Alarm System, the developed system can save 86-89% of the market price of similar system. But I would like

to emphasise, that the price calculation of developed system doesn't contain the cost of certification, possible profit from sales and the cost of software development, i.e. work.

The advantages of developed system include low cost, simplicity of programming, large number of programmable inputs and outputs, the possibility of using not only as a Intruder and Hold-up Alarm System, but also as a Fire Alarm System, the possibility of using also for non-alarm applications, very good protection against sabotage of alarm zone, using 2 alarm ways and possibility of flow control Arduino Mega with the computer. The disadvantages include the need of using a laptop for programming, the fact, that the system isn't certified and impossibility of wireless and other extensions. Another disadvantage may be the need of using the computer for securing and unsecuring of alarm system in case, that we will use computer keyboard or keyboard connected to extension cable for the above purpose.

Due to complexity, variability and low cost of developed system I suppose the target of diploma thesis has been fulfilled and I strongly believe, that the thesis can be used in practice.

SEZNAM POUŽITÉ LITERATURY

- [1] MARTINEK, Radislav. *Senzory v průmyslové praxi*. 1. vyd. Praha: BEN - technická literatura, 2008, 199 s. ISBN 80-730-0114-4.
- [2] PINKER, Jiří. *Mikroprocesory a mikropočítače*. 1. vyd. Praha: BEN - technická literatura, 2004, 159 s. ISBN 80-730-0110-1.
- [3] MICHÁLEK, Bc. LIBOR. *KOMPLEXNÍ ZABEZPEČENÍ OBJEKTU*. Brno, 2011. Diplomová práce. VYSOKÉ UCENÍ TECHNICKÉ V BRNE.
- [4] MALÝ, Mgr. LUDĚK. *NÁVRH METODIKY ŘEŠENÍ ELEKTRONICKÉHO ZABEZPEČENÍ OBJEKTU*. Brno, 2008. Diplomová práce. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ.
- [5] ČSN. *EN 50131-1 ed. 2*. Praha: Úřad pro technickou normalizaci, 2007, 40 s.
- [6] ČSN. *EN 50131-7*. Praha: Úřad pro technickou normalizaci, 2010, 48 s.
- [7] ČSN. *EN 50131-1 ed. 2: ZMĚNA A1*. Praha: Úřad pro technickou normalizaci, 2010, 12 s.
- [8] ČSN. *EN 50131-1 ed. 2: ZMĚNA Z2*. Praha: Úřad pro technickou normalizaci, 2011, 20 s.
- [9] HORNÍK, Jan. *Model zabezpečení inteligentního domu*. Praha, 2010. Bakalářská. ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ.
- [10] KINDL, Ing. Jiří. *Projektování bezpečnostních systémů*. Zlín, 2007. ISBN 978-80-7318-554-1. Učební text vysokých škol. Univerzita Tomáše Bati.
- [11] ČERNÝ, JUDr. Josef a Ing. Ján IVANKA. *Systemizace bezpečnostního průmyslu I*. Zlín, 2006. ISBN 80-7318-402-8. Učební text vysokých škol. Univerzita Tomáše Bati.
- [12] STANISLAV KŘEČEK. *Příručka zabezpečovací techniky*. Blatná: Blatenská tiskárna, s.r.o, 2003. ISBN 80-902938-2-4.
- [13] ČSN. *EN 50131-3*. Praha: Úřad pro technickou normalizaci, 2010, 68 s.
- [14] *C# Programujeme profesionálně*. Brno: Computer Press, a.s, 2007, 1130 s. ISBN 80-251-0085-5.
- [15] HANÁK, Ján. *C# praktické příklady*. Praha: Grada, 2006, 288 s. ISBN 8024709880.

- [16] *C# a .NET 2.0 PROFESIONÁLNĚ*. Brno: Zoner Press, s.r.o, 2007, 1197 s. ISBN 80-86815-42-0.
- [17] MANN, Burkhard. *C pro mikrokontroléry: ANSI-C, kompilátory C, spojovací programy - linkery, práce s ATMEL AVR a MSC-51, příklady programování v jazyce C, nástroje pro programování, tipy a triky*. Vyd. 1. Praha: BEN, 2003, 279 s. ISBN 80-730-0077-6.
- [18] DRAYTON, Peter. *C# v kostce*. Praha: GRADA, 2003, 764 s. ISBN 8024704439.
- [19] *Způsoby zabezpečení drátových ústředn EZS proti sabotáži*. Zlín, 2010. Bakalářská práce. Univerzita Tomáše Bati ve Zlíně.
- [20] ARDUINO. Arduino Home Page [online]. 2011 [cit. 2012-01-17]. Dostupné z: <http://www.arduino.cc>
- [21] CATSOULIS, John. *Designing embedded hardware*. 2nd ed. Sebastopol: O'Reilly, 2005, 377 s. ISBN 05-960-0755.
- [22] MATOUŠEK, David. *Číslicová technika: základy konstruktérské praxe*. 1. vyd. Praha: BEN - technická literatura, 2001, 207 s. ISBN 80-730-0025-3.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

PZTS	Poplachový zabezpečovací a tísňový systém
PZS	Poplachový zabezpečovací systém
PTS	Poplachový tísňový systém
HAS	Hold-up Alarm System
IAS	Intruder Alarm System
I&HAS	Intruder and Hold-up Alarm System
NO	Normally opened
NC	Normally closed
EPS	Elektrická požární signalizace
GSM	Global system module
PPC	Poplachové přijímací centrum
V	Volt-jednotka napětí
Ω	Ohm-jednotka elektrického odporu

SEZNAM OBRÁZKŮ

Obrázek 1: Schéma PZTS.....	14
Obrázek 2: Zapojení pro tísňové zařízení typu NC se zakončovacím odporem.....	20
Obrázek 3: Zapojení pro tísňové zařízení typu NC bez zakončovacího odporu.....	21
Obrázek 4: Zapojení pro tísňové zařízení typu NO se zakončovacím odporem.....	22
Obrázek 5: Zapojení pro tísňové zařízení typu NO bez zakončovacího odporu	22
Obrázek 6: Zapojení pro tísňové zařízení – kombinace NC a NO se zakončovacím odporem.....	23
Obrázek 7: Hodnota odporu smyčky v závislosti na vyvolaném poplachu NC dvojitě vyvážená	25
Obrázek 8: Zapojení typu NC dvojitě vyvážená.....	25
Obrázek 9: Hodnota odporu smyčky v závislosti na vyvolaném poplachu NC trojitě vyvážená	26
Obrázek 10: Zapojení typu NC trojitě vyvážená	26
Obrázek 11: Hodnota odporu smyčky v závislosti na vyvolaném poplachu NC zdvojení zóny.....	27
Obrázek 12: Zapojení typu NC zdvojení zóny	27
Obrázek 13: Celková struktura poplachového zabezpečovacího systému	43
Obrázek 14: Základní popis programu v Arduinu	46
Obrázek 15: Komunikace v nezastřeženém stavu.....	46
Obrázek 16: Způsob připojení detektorů ke smyčce typu ATZ.....	51
Obrázek 17: Hlavní okno.....	53
Obrázek 18: Nabídka “Hlavní“	53
Obrázek 19: Nabídka “Zastřežit“	54
Obrázek 20: Nabídka “Komunikace“.....	55
Obrázek 21: Uložení nastavení do Arduina.....	55
Obrázek 22: Nabídka “Programování vstupů a výstupů“.....	56
Obrázek 23: Programování analogových vstupů	57
Obrázek 24: Nastavení vstupu u časové aktivace	58
Obrázek 25: Programování digitálních vstupů/výstupů	58
Obrázek 26: Časová aktivace pro digitální vstupy/výstupy	59
Obrázek 27: Všechny časové aktivace	59
Obrázek 28: Programování základního nastavení.....	60

Obrázek 29: Programování základního nastavení.....	61
Obrázek 30: Změna hesla pro přístup k paměti a e-mailu.....	61
Obrázek 31: Změna hesel pro zastřežení a odstřežení	61
Obrázek 32: Přiřazení výstupů k daným heslům	62
Obrázek 33: Přehledný náhled nastavení.....	62
Obrázek 34: Nabídka “Nastavení času“	63
Obrázek 35: Připojení detektoru k poplachovému systému	65

SEZNAM TABULEK

Tabulka 1: Detekce opakovaných neplatných pokusů o udělení oprávnění [13]	31
Tabulka 2: Monitorování běhu programu [13]	32
Tabulka 3: Časování [13]	32
Tabulka 4: Hodnotové datové typy 1 [16].....	35
Tabulka 5: Hodnotové datové typy 2 [16].....	35
Tabulka 6: Rozložení paměti EEPROM.....	45
Tabulka 7: Požadavky a odpovědi bez kontroly zakončení ze strany Arduina	48
Tabulka 8: Požadavky a odpovědi s kontrolou zakončení ze strany Arduina	48
Tabulka 9: Požadavky a odpovědi v zastřeženém stavu	50
Tabulka 10: Finanční náklady na poplachový zabezpečovací systém s PC	68
Tabulka 11: Cena běžného poplachového zabezpečovacího systému včetně porovnání s poplachovým systémem navrženým v praktické části	69
Tabulka 12: Celkový počet zón a výstupů u vybraného poplachového zabezpečovacího systému.....	70

SEZNAM PŘÍLOH

PŘÍLOHA PI: PŘIPOJENÍ DETEKTORU K POPLACHOVÉMU SYSTÉMU

PŘÍLOHA PI: PŘIPOJENÍ DETEKTORU K POPLACHOVÉMU SYSTÉMU

