

Výukový model poplachového zabezpečovacího systému

František Kráčalík

Bakalářská práce
2020



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav bezpečnostního inženýrství

Akademický rok: 2019/2020

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **František Kráčalík**
Osobní číslo: **A17096**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Bezpečnostní technologie, systémy a management**
Forma studia: **Prezenční**
Téma práce: **Výukový model poplachového zabezpečovacího systému**
Téma práce anglicky: **An Educational Model of a Home Alarm**

Zásady pro vypracování

1. Vypracujte literární rešerši na dané téma.
2. Navrhněte komponenty poplachového zabezpečovacího systému vhodné pro výukový model.
3. Zvolte mikropočítačovou platformu pro řízení celého systému.
4. Vytvořte panel s mikropočítačem a vybranými komponentami.
5. Implementujte ukázkové programové vybavení pro zvolený mikropočítač.

Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam doporučené literatury:

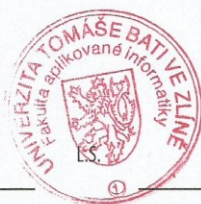
1. BANZI, Massimo. Getting started with Arduino. 2nd ed. Farnham: O'Reilly, 2011. ISBN 9781449309879.
2. BURDA, Karel. Základy elektronických zabezpečovacích systémů. Brno: Akademické nakladatelství CERM, 2017. ISBN 978-80-7204-967-7.
3. LADMAN, Josef. Elektronické konstrukce pro začátečníky. Praha: BEN – technická literatura, 2001. ISBN 80-730-0015-6.
4. MARGOLIS, Michael. Arduino cookbook. 2nd ed. Sebastopol, Calif.: O'Reilly, 2012, xx, 699 p. ISBN 1449313876.
5. PINKER, Jiří. Mikroprocesory a mikropočítače. Praha: BEN – technická literatura, 2004. ISBN 80-7300-110-1.

Vedoucí bakalářské práce:

Ing. Jan Dolinay, Ph.D.

Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce: 17. prosince 2019
Termín odevzdání bakalářské práce: 25. května 2020



doc. Mgr. Milan Adámek, Ph.D.
děkan

Ing. Jan Valouch, Ph.D.
ředitel ústavu

Ve Zlíně dne 7. prosince 2019

Prohlašuji, že

- beru na vědomí, že odevzdáním 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 bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk 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 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 připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování 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 bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem 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 bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 7. srpna 2020

.....
podpis diplomanta

ABSTRAKT

Cílem práce je navrhnout a sestavit jednoduchý model poplachového zabezpečovacího systému, který je založen na mikropočítačové platformě Arduino. Teoretická část se zabývá poplachovými zabezpečovacími systémy, mikropočítači a částečně nastiňuje i desku Arduino se softwarem. V praktické části jsou vybrány a popsány komponenty systému, jejich zapojení, umístění na desce a programové vybavení.

Klíčová slova: mikropočítač, Arduino, poplachový zabezpečovací systém, detektory, programování

ABSTRACT

The aim of the thesis is to design and build a simple model of alarm security system, which is based on the Arduino microcomputer platform. The theoretical part deals with alarm security systems, microcomputers and partially outlines the Arduino board with software. The practical part selects and describes the components of the system, their connection, location on the board and software.

Keywords: microcontroller, Arduino, alarm security system, detector, programming

Zde bych chtěl poděkovat Ing. Janu Dolinayovi, Ph.D. za odborné rady a konzultace při psaní mé bakalářské práce.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 POPLACHOVÝ ZABEZPEČOVACÍ SYSTÉM	11
1.1 SCHÉMA PZS.....	11
1.2 MOŽNOSTI ZAPOJENÍ PZS.....	12
1.2.1 Drátové systémy.....	12
1.2.2 Bezdrátové systémy	14
1.2.3 Hybridní systémy	15
1.3 REŽIMY SYSTÉMU.....	16
1.3.1 Zastřežený	16
1.3.2 Částečně zastřežený	16
1.3.3 Odstřežený.....	16
1.3.4 Servisní.....	16
1.4 PRVKY PZS	17
1.4.1 Prvky obvodové ochrany.....	17
1.4.2 Prvky plášťové ochrany	17
1.4.3 Prvky prostorové ochrany	18
1.4.4 Prvky předmětové ochrany	20
1.5 ÚSTŘEDNA PZS.....	21
1.6 KLÁVESNICE PZS.....	21
2 MIKROPOČÍTAČ	22
2.1 MIKROPROCESOR	22
2.2 ARCHITEKTURA.....	22
2.2.1 von Neumann	22
2.2.2 Harvardská	22
2.3 SBĚRNICE	23
2.3.1 Adresová sběrnice	23
2.3.2 Datová sběrnice	24
2.3.3 Řídicí sběrnice.....	24
2.4 INSTRUKČNÍ SADA	24
3 ARDUINO	25
3.1 VÝVOJOVÁ DESKA.....	25
3.2 VÝVOJOVÉ PROSTŘEDÍ	25
3.2.1 Arduino IDE.....	25
3.2.2 Programino	26
3.2.3 Visual Studio.....	27
3.2.4 Fritzing	28
II PRAKTICKÁ ČÁST	29

4	VÝBĚR KOMPONENTŮ	30
4.1	ARDUINO UNO R3	30
4.2	16X2 LCD DISPLEJ 1602 MODRÝ + I ² C PŘEVODNÍK	31
4.3	4X3 MATICOVÁ MEMBRÁNOVÁ KLÁVESNICE	32
4.4	PIR DETEKTOR HC-SR505.....	33
4.5	IR SNÍMAČ FC-51	34
4.6	MAGNETICKÝ KONTAKT SM-35.....	34
4.7	AKTIVNÍ BZUČÁK 5V.....	35
4.8	LED DIODA 3MM.....	35
5	ROZMÍSTĚNÍ A ZAPOJENÍ KOMPONENTŮ	36
6	PROGRAMOVÉ VYBAVENÍ.....	38
6.1	KNIHOVNY	38
6.2	DEKLARACE, INICIALIZACE PROMĚNNÝCH A KONSTANT	38
6.3	FUNKCE SETUP	40
6.4	FUNKCE SMYCKA	41
6.5	FUNKCE ODPOCET.....	46
6.6	FUNKCE DETEKTORY	47
	ZÁVĚR	51
	SEZNAM POUŽITÉ LITERATURY.....	52
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	55
	SEZNAM OBRÁZKŮ	57
	SEZNAM TABULEK.....	58
	SEZNAM PŘÍLOH.....	59

ÚVOD

Při pohledu na dnešní svět by se leckomu mohlo zdát, že ho už neřídíme my, ale tuto práci za nás dělají (mikro)počítače. Malé integrované obvody se postupem času staly obrovským hitem, bez kterého si dnes elektronická zařízení snad ani nedokážeme představit. Jejich obrovská univerzálnost a nízká cena v porovnání s odvedenou prací přímo nabízí, aby lidé přestali takovou práci dělat a začali stroje, zařízení či systémy obsluhovat. Nejinak je tomu i u zabezpečovacích systémů.

Tato bakalářská práce má za úkol navrhnout a prakticky zrealizovat výukový model malého poplachového zabezpečovacího systému. Tento model by měl sloužit studentům v předmětech pro programování mikropočítačů a PLC, aby snadněji pochopili, jak takový zabezpečovací systém v praxi funguje.

Při výběrů komponentů vznikla diskuze, jaká platforma by byla nejvhodnější. Po zvážení různých aspektů se vybrala platforma Arduino. Největší výhodou oproti ostatním má v open-source hardwarovém i programovém vybavení. Existuje několik kvalitních a volně dostupných knih zabývajících se programováním Arduina, možná stovky internetových stránek, kde přispívají svými nápady jak začátečníci ve světě Arduina, tak i výrazně pokročilí s odvážnými ukázkami. Dalšími výhodami jsou nízká cena i poměrně jednoduchý programovací jazyk wiring.

Volba použitých detektorů podléhala hlavně dvěma kritériím. První z nich byl jejich fyzikální princip. Je velmi vhodné, aby jednotlivé detektory nepracovaly na stejném principu, ale aby student dokázal pochopit, v čem je rozdíl oproti jiným senzorům a detektorům. Druhým kritériem byla kompatibilita Arduina s detektory, především napájecí napětí, velikost detektoru a vyzařovací charakteristika.

Programová část obsahuje řadu základních příkazů a cyklů, které by student s malou znalostí jazyka C/C++ měl porozumět. Smysl této části není ukázat složité a komplikované řešení, ale naopak, přesvědčit studenty o tom, že programování leč menšího systému nemusí být příliš složité.

I. TEORETICKÁ ČÁST

1 POPLACHOVÝ ZABEZPEČOVACÍ SYSTÉM

Poplachovým zabezpečovacím systémem (PZS) označujeme soubor elektronických prvků, jejichž účelem je detekce a signalizace neoprávněného vniku osob do střežených objektů nebo prostor, ale i střežení cenných předmětů před neoprávněnou manipulaci s nimi (např. sochy, trezory apod.). Neoprávněný vnik do střežených míst je zabezpečen pomocí detektorů. Ty mají nejčastěji za úkol sledovat určitý prostor či předmět a včas detekovat přítomnost pachatele. V případě narušení prostoru či předmětu tuto nežádoucí událost detektory odesílají do ústředny. Ústředna je hlavní „mozek“ celého systému zajišťující jeho řízení. [1] [2]

1.1 Schéma PZS

Celý systém není samozřejmě složen pouze z detektorů a ústředny. Obrázek 1 představuje soubor nejčastěji používaných prvků PZS.



Obrázek 1 Blokové schéma PZS [1]

- **Detektor** – včasně detekuje neoprávněný pohyb či manipulaci s předmětem na základě fyzikální změny, která při takové činnosti nastane. Tato fyzikální změna je vyhodnocena elektronikou uvnitř detektoru a odešle se do ústředny.
- **Ústředna** – z detektorů vyhodnocuje přijaté informace, zpracovává je a odesílá k obsluze. Ústředna autonomně ovládá celý PZS, zprostředkovává napájení komponentů (v případě bezdrátového napájení sleduje kapacitu baterie) i způsob přenosu informace.
- **Přenosové prostředky** – zajišťují přenos signálu z ústředny do místa vzniku nebo i opačným směrem.
- **Signalizační zařízení** – přijaté informace v podobě napětí se převádí do jiných podob signálu, např. optických či zvukových signalizací.

- **Doplňková zařízení** – pomocí nich lze systém ovládat nebo propojit PZS s jinými systémy nainstalovanými v objektu (Systém kontroly vstupu – provádět řízení přístupu a výstupu do/z objektu). Ovládání systému nejčastěji probíhá skrze klávesnici a displej, dnes je však trendem systém odstřežit/zastřežit i pomocí dálkového ovladače, RFID čipu, mobilní aplikace nebo přes počítač. [1] [2]

1.2 Možnosti zapojení PZS

Podle způsobu zapojení můžeme PZS rozdělit:

1.2.1 Drátové systémy

Smyčkové

Zařízení se propojují přes metalická vedení. Tyto kabely bývají většinou ve svazcích po párech. Využívá se jeden datový vodič pro komunikaci a zbylé mohou sloužit pro napájení zařízení. Komunikace datovým vodičem je realizována pouze ve směru od zařízení k ústředně. Ta měří odpor smyčky, kde je obvykle zapojeno několik rezistorů, a podle této velikosti pak rozhoduje, co se ve smyčce stalo, případně které zařízení spustilo hlášení. Smyčková zapojení lze taktéž rozdělit a to na několik typů:

- **Smyčka NC (Normally Closed)** – Detektory jsou ve smyčce zapojeny sériově. Smyčka je schopna reagovat na dva stavy. V klidovém je odpor smyčky téměř nulový, obvodem prochází proud. Při aktivaci se naopak odpor blíží nekonečnu, obvod je rozepnutý a tudíž jím nemůže proud procházet (v obou případech zanedbáváme odpor vodiče). Maximální počet detektorů v jedné smyčce a maximální celkový počet smyček udává výrobce.
- **Smyčka NC – jednoduše vyvážená** – Jedná se o vylepšenou verzi předchozí smyčky. Používá se zde tzv. zakončovací odpor (End Of Line – EOL), který chrání smyčku před sabotáží zkratem. Při klidovém stavu je odpor celé smyčky roven zakončovacímu odporu. Pokud odpor smyčky klesne pod určitou úroveň zakončovacímu odporu (obvykle 30 %), je vyhlášena sabotáž, pokud je odpor smyčky vyšší (blížící se nekonečnu), je vyhlášen poplach nebo sabotáž způsobena přerušením smyčky.

- **Smyčka NC – dvojitě vyvážená** – Dvojitým vyvážením se myslí přidání sabotážního kontaktu (tamperu). Smyčka je poté schopna vyhodnotit 4 různé stavy: sabotáž zkratem, klidový stav, stav poplachu, narušení detektoru a sabotáž přerušením smyčky.
- **Smyčka NC – trojitě vyvážená** – K poplachu a tamperu je sériově zařazen tzv. antimasking, který dokáže rozpoznat (ne)úmyslné zastínění detektoru. V případě aktivace tohoto kontaktu odpor celé smyčky je roven odporu paralelně zapojeného k antimaskingu a zakončovacímu odporu.
- **Smyčka NC – ATZ (Advanced Technology Zoning)** – Nejdokonalejší způsob smyčkového zapojení spočívá v použití ATZ. Výše uvedené smyčky v zapojení NC nedokáží při sepnutí poplachu zjistit, který detektor poplach vyvolal. U ATZ jsou však jednotlivé poplachové kontakty od sebe rozděleny pomocí různých hodnot odporů. Většinou se prvnímu detektoru ve smyčce přiřazuje nejnižší odpor k poplachovému kontaktu a se zvyšujícím se počtem detektorů stoupá i hodnota těchto odporů. Kvůli rozdílným odporům je pak ústředna schopna přesně určit ten detektor, který kontakt rozpojil a zvýšil celkový odpor ve smyčce. Je však také možnost používat menší počet odporů u více detektorů a rozdělit si je tak do zón. Jako příklad může posloužit rodinný dům, kde v obývacím pokoji budou detektory s jinou hodnotou odporu než v ložnici. Při rozepnutí kontaktu a průchodem proudu paralelně zařazeným odporem detektorem umístěným v ložnici bude výsledný odpor jiný, než kdyby se sepnul detektor v obývacím pokoji. Touto cestou jsme schopni snadno určit místnost v domě, kde se při zastřežení systému někdo nachází. Při sabotáži přerušením smyčky nejsme schopni určit místo přerušení a při rozepnutí tamperu určit detektor. [2] [3]

Sběrnice

Tyto systémy využívají digitální komunikaci mezi ústřednou a detektory. Pro komunikaci se používá jeden pár vodičů, který se zapojuje paralelně k detektorům. Druhý pár vodičů slouží pro napájení. Všechny detektory musí být vybaveny sběrnicevým komunikačním modulem.

Za velkou výhodu použití sběrnice lze považovat fakt, že ústředna jednoznačně identifikuje, který detektor byl aktivován nebo i v jaké časové posloupnosti se stavy na ústřednu odeslaly (sabotáž zkratem vedení, stav poplachu, rozepnutí tamperu apod.). Každý detektor má totiž svou vlastní adresu. Na tuto adresu přicházejí z ústředny tzv. datové rámce (Obrázek 2). Ty

začínají řadou bitů ADR, což je adresa daného detektoru. Poté následují bity DATA, které přenáší stavy z detektoru nebo příkazy z ústředny. Komunikace je ukončena bity CRC sloužící ke kontrole, zda se při přenosu datového rámce stala někde chyba. [1] [2]

Detektory, nebo spíše veškerá zařízení připojená k ústředně přes sběrnici, mají předem stanovený odběr proudu z ústředny. Ve spojení s dlouhým kabelovým vedením se může stát, že velký úbytek napětí na vedení způsobí nízké napájecí napětí u zařízení, která jsou příliš daleko od zdroje napájení. Řešení nabízí posilovací zdroj, jenž je schopen vyrovnat úbytky napětí na sběrnici. [2]



Obrázek 2 Datový rámec [2]

Kombinované

Veškerá zařízení PZS lze připojit skrze smyčky nebo sběrnice. Starší smyčková zařízení se připojují přímo do ústředny nebo pomocí koncentrátoru. Výhoda použití koncentrátorů spočívá ve spojení dvou různých typů zapojení v celém systému. Na něj jsou zařízení připojena pomocí smyček, komunikace mezi ústřednou a koncentrátorem pak probíhá přes sběrnici. V případě dostatečné velikosti ústředny, lze do koncentrátoru přímo připojit zařízení PZS. Pak se jedná o funkční sběrniceový systém se všemi výhodami. Nesmí se však zapomenout na zajištění dostatečného napájení i pro vzdálená zařízení. [4]

1.2.2 Bezdrátové systémy

Jejich použití není obvyklé u novějších staveb, kde se vedení kabelů často instaluje pod omítku nebo do instalačních lišt. Využití nacházejí především u historických budov, které mají kamenné zdivo, historické malby na stěnách, stropech apod. Výhodu přináší i při dočasném použití nebo při rozšíření systému. Nevýhoda spočívá v napájení zařízení. Ta musejí mít vlastní baterie. Ústředna sleduje napětí na bateriích. Pokud toto napětí klesne pod stanovenou hodnotu, tak se informace přenesou na ústřednu nebo na místě, kde se zařízení nachází, dojde k akustické signalizaci. Při velké vzdálenosti zařízení od ústředny je možné tento problém vyřešit použitím opakovače rádiového signálu. Používaná zařízení jsou stejná jako u drátových systémů a to:

- detektory rozbití skla,
- PIR detektory,

- magnetické kontakty,
- signalizační zařízení,
- ovládací zařízení atd.

V bezdrátové komunikaci mezi ústřednou a ostatními zařízeními se dělí přenos na jednosměrný a obousměrný. [1]

Jednosměrná komunikace

Jednosměrnou komunikaci představuje vysílač v detektoru a přijímač v ústředně. Dříve detektory neposílaly žádné kontrolní telegramy, takže v případě poničení nebo i odcizení ústředna o ničem nevěděla. Se zavedením kontrolních telegramů tento problém částečně zmizel. Každé vyslání takového telegramu totiž stojí detektor část energie, kterou bere z baterie. Obvykle probíhá kontrola jednou za několik hodin. U tohoto typu komunikace je také nevýhoda ve snadném napadení. Lze zjistit frekvenci i modulaci komunikace, pak stačí přijímač zahltnit stejným kmitočtem o podstatně vyšší intenzitě. [4]

Obousměrná komunikace

Změnou k lepšímu se ukázalo použít vysílač i přijímač jak do ústředny, tak i do detektoru. Duplexní přenos odstranil hlavní nedostatky předchozí komunikace. Ta probíhá souběžně na dvou frekvencích. V případě zahlcení těchto frekvencí jsou všechna zařízení systému schopna vyhledat jiné frekvence a pracovat na nich. Ústředna si při každém zastřežení zkontroluje připojená zařízení, může si také ověřit, zda přijatá informace o poplachu je pravdivá či nikoliv. V klidovém stavu nemají detektory žádný odběr energie pro kontrolní telegramy, ty se nemusí posílat. [4]

1.2.3 Hybridní systémy

Kombinací drátového a bezdrátového systému jsme schopni sestavit systém hybridní. Jeho výhoda spočívá v použití jak drátových, tak i bezdrátových zařízení zapojených do stejné ústředny. Při realizaci se totiž může stát, že se na určitá místa s vedením dostaneme komplikovaně nebo třeba vůbec. V případě rozšíření drátového systému tak odpadá nutnost tahání kabeláže. [1]

1.3 Režimy systému

System může pracovat v několika režimech. Mezi ty základní patří:

1.3.1 Zastřežený

Během tohoto režimu je celý systém v pohotovosti. Všechny detektory jsou aktivní a jsou schopné ohlásit na ústřednu různé stavy (sabotáž, poplach, atd.). [3]

1.3.2 Částečně zastřežený

Slouží k odpojení některé zóny za předpokladu, že zbylé zóny systému jsou plně funkční. Zóny mohou být např. v rodinném domě rozděleny podle místností či pater. Abychom se mohli volně pohybovat při spaní v prvním patře, tak můžeme zastřežit pouze přízemí. Nebo budou reagovat pouze detektory obvodové ochrany, detektory v interiéru domu budou vypnuty.

1.3.3 Odstřežený

Ústředna nereaguje na stavy detektorů. Může ale sledovat pokus o sabotáž všech zařízení (přerušování vedení, maskování) nebo přijímat stavy z požárních hlásičů. [3]

1.3.4 Servisní

Servisní režim slouží ke kompletnímu nastavení celého systému. Při tomto režimu nemůže dojít k vyhlášení poplachu. K základu patří nastavení času, přidání/odebrání zařízení, programování PGM nebo nastavení kódů na různých úrovních. Takové programování je u mnoha výrobců odlišné. Obecně se dá říci, že lze provádět pomocí:

- Klávesnice, což je starší způsob. Veškeré programování probíhá pomocí zadávání čísel či písmen na klávesnici, odpovědi jsou zobrazeny na LCD displeji. Technik obvykle potřebuje instalační manuál.
- Počítače a softwaru. Ústředna a počítač je obvykle propojena sběrnici USB či RS 232, lepší ústředny poté nabízí i komunikaci přes internet, telefonní linku apod. Programování probíhá v softwaru na počítači, někteří výrobci si software sami vyrábí, jiní používají sdílený s více výrobci. [3]

1.4 Prvky PZS

Jednotlivé prvky systému se třídí podle toho, kde se instalují. Umístění a směrování detektorů by se mělo provádět dle manuálu výrobce tak, aby byl detektor maximálně efektivní. Správnému výběru a instalaci také podléhá třída prostředí a stupeň zabezpečení.

1.4.1 Prvky obvodové ochrany

Perimetrickou (obvodovou) ochranou střežíme pozemek před neoprávněným vstupem cizích osob. Plocha pozemku může být velká, proto se prvky umisťují na předpokládaná místa narušení.

Pro svůj provoz tyto detektory můžeme rozdělit: [1]

Pasivní

Detektory v této skupině pasivně reagují na fyzikální změny v jejich blízkosti. Kolem svého okolí nevyzařují žádný signál a obvykle se těžko identifikují. Patří zde:

- plotové vibrační a tenzometrické detektory,
- mikrofonní kabely,
- diferenciální tlakové detektory,
- vláknové optické systémy. [5]

Aktivní

Vyzařují signál do prostoru, vytváří si okolí, kde jejich detekční schopnost dosáhne. Například pohyb pachatele ve střeženém prostoru je zaznamenán změnou vyzařovací charakteristiky. Ta se dá lehce zjistit, proto lze u aktivních detektorů určit tzv. mrtvé zóny. Patří zde:

- kapacitní čidla,
- infračervené závory a bariéry,
- štěrbinové kabely. [5]

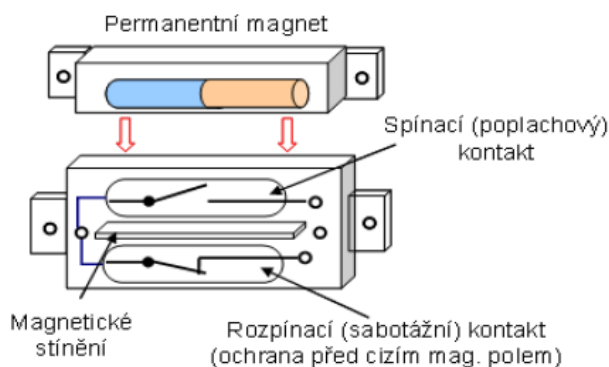
1.4.2 Prvky plášťové ochrany

Jejich úkolem je včasná signalizace pachatele při překonávání pláště budovy. Nejčastěji se jedná o otvorové výplně (dveře, okna, světlíky), avšak i stropy, podlahy nebo střechy zde můžeme zařadit. Detektory se dělí:

- kontaktní,
- destrukční,
- destrukčních projevů,
- tlaková akustická,
- bariérová. [1]

Magnetický kontakt

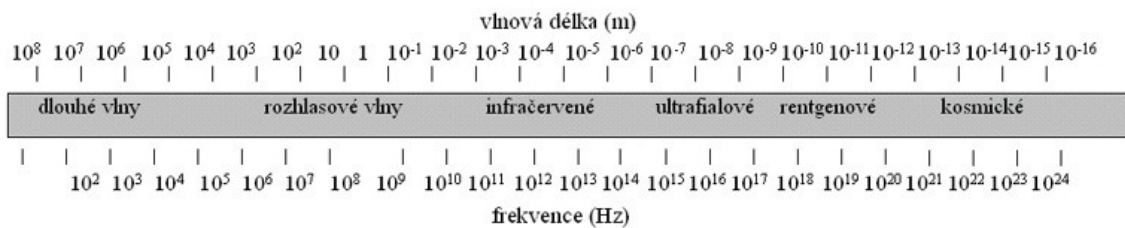
Používají se ke střežení pohyblivých částí pláště budovy (otevření oken, dveří, vrat). Detektor má dvě části: permanentní magnet připevněným na pohyblivou část a jazýčkový kontakt na pevné části. Jazýčkový kontakt se skládá ze dvou plíšků magneticky měkkého materiálu, ty jsou trvale zataveny do skleněné trubičky vyplněné obvykle argonem. Po přiložení magnetu, který kolem sebe vytváří magnetické pole, se jazýčky zmagnetizují, přiblíží se k sobě a vytvoří kontakt. Po oddálení magnetu naopak magnetické pole zanikne, jazýčky se rozpojí. Ústředna je schopna tyto dva stavy snadno identifikovat. Výhodné jsou kvůli snadné montáži a dlouhé životnosti. Nevýhodou v podobě přiložení jiného magnetu ke kontaktu a otevření dveří/oken bez rozepnutí jazýčků jsme schopni eliminovat použitím dvou jazýčkových kontaktů v jednom pouzdře. [1]



Obrázek 3 Magnetický kontakt se dvěma jazýčky [6]

1.4.3 Prvky prostorové ochrany

Jsou často označovány jako doplňkové prvky plášťové ochrany. Instalují se na přístupová nebo spojovací místa budovy tj. chodby, schodiště, dveře nebo i přímo do místností. Pohybové detektory mohou pracovat na různých fyzikálních principech a také v různém elektromagnetickém spektru.



Obrázek 4 Elektromagnetické spektrum, upraveno [7]

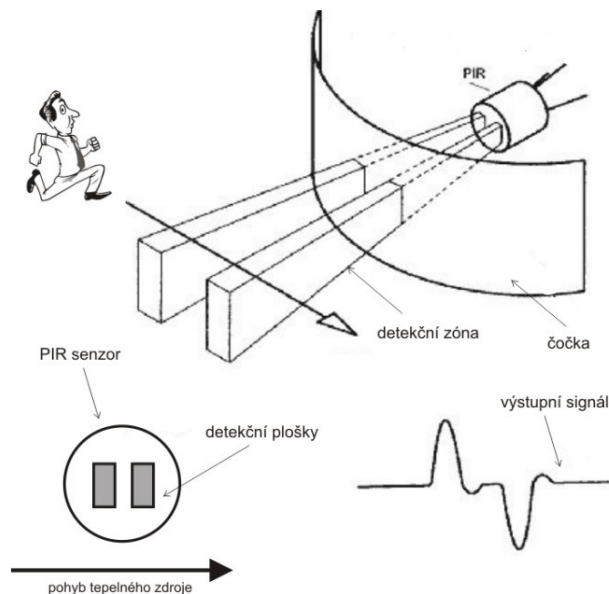
Můžeme je roztrždit:

- VKV detektory,
- mikrovlnné detektory,
- ultrazvukové detektory,
- pasivní a aktivní infračervené detektory,
- kombinované (duální) detektory. [1]

Nevýhodou těchto detektorů bývá vyšší náchylnost k falešným poplachům z důvodu proudění vzduchu v prostoru (topení, větrání). Citlivost je také dána materiály pohlcující ultrazvuk (polystyrén, koberec). Našly však své uplatnění v interiérech aut, kde se okolní teploty mohou výrazně měnit. [1]

PIR detektor

Pasivní infračervený detektor patří k nejpoužívanějším prvkům celého zabezpečovacího systému. Pracuje na principu detekce infračerveného záření lidského těla, které se pohybuje kolem 10 μm . Základem je pyroelektrický snímač využívající pyroelektrický jev. Snímač sleduje změny intenzity tepelného záření, to znamená, že když se v jeho zorném poli objeví osoba nebo předmět vyzařující teplotu v rozmezí jeho nastavení (nejčastěji teplota lidského těla 35° až 45°), je tento pohyb zachycen. V praxi se používají dva snímače v rozdílovém zapojení a Fresnelova čočka. Tato kombinace dokáže rozeznat pohyb osoby tím, že se postupně přerušují jednotlivé segmenty čočky. [1] [2]



Obrázek 5 Princip PIR detektoru [8]

1.4.4 Prvky předmětové ochrany

Slouží ke střežení vzácných, cenných předmětů jako jsou sochy, trezory či šperky. Nejčastěji je tedy můžeme nalézt v muzeích, peněžních ústavech galeriích atd. Do této skupiny detektorů řadíme:

- kontaktní,
- kapacitní,
- bariérová,
- trezorová,
- tlaková akustická,
- pro umělecké předměty (závěsové, váhové, polohové, optické). [1]

Optický detektor

Pracují podobně jako ultrazvukové detektory. LED dioda vysílá infračervené záření, jehož vlnová délka se pohybuje od 700 μm do 1 mm. Toto záření se odráží od chráněného předmětu a dopadá na přijímač tvořený polovodičovou fotodiodou. PN přechod vně fotodiody se při průniku světla otevře a vzroste závěrný proud. Při odcizení předmětu se do oblasti PN přechodu žádné záření nedostane, neprochází proud a je vyhlášen poplach. [2]

1.5 Ústředna PZS

Nejdůležitějším prvkem poplachového zabezpečovacího systému je ústředna. Základní deska ústředny se skládá z vyhodnocovací a řídicí jednotky a svorkovnice pro připojení veškerých ostatních zařízení. V krytu ústředny pak můžeme nalézt např. záložní baterii, expandér, transformátor síťového napájení nebo komunikační zařízení. Mezi základní funkce ústředny patří:

- příjem a vyhodnocení informací nebo stavů z detektorů,
- ovládání přenosových, signalizačních a záznamových zařízení,
- diagnostika systému,
- napájení všech prvků systémů vyjma těch bezdrátových. [9]

1.6 Klávesnice PZS

Ačkoliv ústředna je schopna řídit celý systém bez pomoci člověka, takový systém by však nebyl schopen nikdo obsluhovat. Pro ovládání nebo i programování ústředny se používá klávesnice. Umisťuje se na viditelná místa nejčastěji ke vstupním dveřím do objektu potažmo místnosti. Její hlavní činností je zastřežení/odstřežení objektu. S ústřednou může komunikovat bezdrátově či drátově, např. pomocí sběrnice. Dříve se používali LED klávesnice, které indikovaly přijaté informace z ústředny skrze různě barvené LED diody. Tyto typy vystřídaly LCD klávesnice se zabudovaným displejem, do nich se začal přidávat i čtečka karet. Nejmodernější typ pak představuje dotyková klávesnice s barevným displejem. Takové ovládací panely nabízí různé chytré funkce, např. měření teploty, změnu barev, intenzitu podsvícení apod. [3]

2 MIKROPOČÍTAČ

V dnešní době lze mikropočítač nalézt v téměř každém výpočetním zařízení. Takový jednočipový mikropočítač obsahuje tři základní bloky: mikroprocesor, operační paměť a vstupně/výstupní obvody zkonstruované do jednoho integrovaného obvodu. Sloučením bloků do podoby jednodité a malé, přesto velmi účinné součástky, se začali konstruktéři a programátoři předhánět v technologiích, jak zvětšit paměti či zrychlit periferní obvody tak, aby se nezvětšil objem samotného mikropočítače, někdy označovaného jako mikrokontrolér. [10]

2.1 Mikroprocesor

Mnoho lidí zaměňuje pojem mikropočítač za mikroprocesor, ale nejsou to ekvivalentní názvy. Mikroprocesor vykonává aritmetické a logické operace, které postupně provádí jako jednotlivé instrukce kódu. Tato základní jednotka mikropočítače obsahuje aritmeticko logickou jednotku Arithmetic Logic Unit (ALU), pracovní registry, vyrovnávací paměti a řídicí obvody – řadič. [10]

2.2 Architektura

Každý mikropočítač má svoji architekturu, která vysvětluje, jak probíhá komunikaci mezi bloky, jak jsou navrženy obvody bloků či atd. Používají se dvě různé architektury:

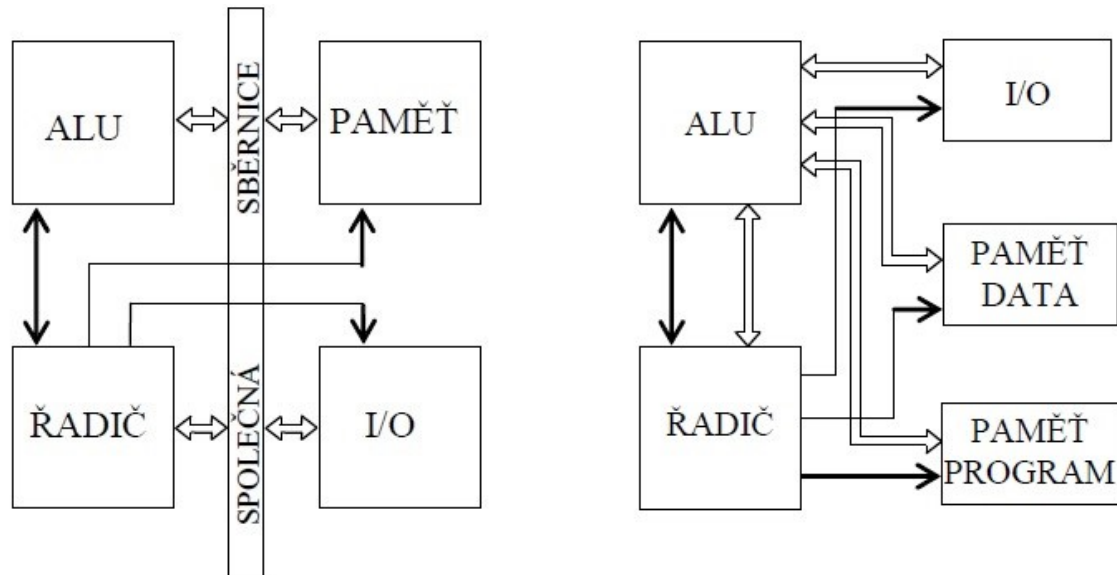
2.2.1 von Neumann

Moderní mikropočítače obvykle fungují na principu, který navrhl Američan John von Neumann. Mikroprocesor má 4 funkční bloky: ALU, řadič, paměť a vstupně/výstupní jednotky. V jedné paměti se nachází jak příkazy (instrukce), tak i hodnoty (operandy). Program se vykonává jako posloupnost příkazů, ve kterém se uvádí adresa operandu, takže program se při změně dat nemění. Instrukce probíhají tak, jak jsou zapsány do paměti. Lze tedy souvisle upravovat prostor pro data a program v paměti, ale za cenu pomalejšího procesu a možného přepsání programu pomocí dat. [11]

2.2.2 Harvardská

Základem je opět von Neumannova architektura, avšak velký rozdíl je v oddělené paměti pro program a pro data. Díky tomu je daleko bezpečnější pro programátory, protože jsou data čtena i ukládána ze samostatné paměti a tedy nemohou přepsat program. Paměti

nevyužívají společnou, nýbrž samostatnou sběrnici, takže procesy probíhají rychleji. Nevýhodou je, že prostor pro data nebo program může být zbytečně velký nebo naopak malý a zůstane nevyužitý. [11]



Obrázek 6 Architektura von Neumann (vlevo), Harvardská architektura (vpravo) [11]

2.3 Sběrnice

Mikroprocesor je spojen s pamětí a ostatními periferiemi prostřednictvím vodičů. Mikropočítače používají tři různých skupin vodičů, nazýváme je sběrnice (anglicky BUS). Každá sběrnice slouží při komunikaci k něčemu jinému. Po sběrnících se informace přenáší paralelně nebo sériově. Vyjma níže uvedeného rozdělení se na ně můžeme dívat jako na vnitřní mikropočítačové, které propojují mikroprocesor s ostatními částmi mikropočítače, vnitřní mikroprocesorové, sloužící pro propojení registrů uvnitř mikroprocesoru a vnější mikroprocesorové pro připojení dalších přídatných zařízení. Nejklasičtější dělení sběrnice však vypadá následovně: [11]

2.3.1 Adresová sběrnice

Mikroprocesor při čtení nebo k zápisu dat potřebuje adresy těchto dat. Tato sběrnice tedy slouží pro přenos adres. Počet bitů je pak roven počtu bitů té adresy, kterou mikroprocesor umí vytvořit a určuje maximální využitelný adresovatelný prostor. Univerzální mikroprocesory nabízí většinou dva adresové prostory – jeden pro paměť a další pro vstupy a výstupy. [11]

2.3.2 Datová sběrnice

Datová sběrnice zabezpečuje přenos všech dat. Komunikace mezi dvěma bloky zajišťuje vysílač a přijímač. V jednom okamžiku může vysílat pouze jeden zdroj dat. Při kolizi by se totiž data ztratila, dokonce by se vysílací obvody mohly zničit. Pro tyto případy se přidávají do obvodů tzv. třístavové budiče.

Důležitou vlastností datové sběrnice je její šířka – tj. počet bitů, které lze zároveň po sběrnici přenést. Na základě tohoto údaje pak značíme mikroprocesory (obecně mikropočítače) jako osmibitové, šestnáctibitové atd. Dalšími vlastnostmi je taktovací frekvence a přenosová rychlost. [11]

2.3.3 Řídicí sběrnice

Funkce řídicí sběrnice spočívá v přenosu signálů, které řídí komunikaci mezi bloky. Signály přichází jen do bloků, jimž jsou pro to určeny. Jednotlivé vodiče obvykle přenáší jeden typ signálu od různých zdrojů. Přenos přes vodič probíhá při hodnotě HIGH. [11]

2.4 Instrukční sada

Instrukční sada je soubor aritmeticko logických funkcí, instrukcí pro řízení programu, pro přesun mezi jednotlivými registry mikroprocesoru, pro skoky mezi instrukcemi atd. Používají se dvě různé koncepce: CISC (Complex Instruction Set Computer) znamená kompletní instrukční sada, obsahuje velký počet instrukcí a těmto procesorům trvá déle zpracování strojové instrukce. RISC (Reduced Instruction Set Computer), tedy redukovaná instrukční sada, má omezený počet instrukcí. Důraz je totiž na rychlé zpracování, avšak délka programu v porovnání s CISC je delší. Většina instrukcí se provádí během jednoho taktu a mají pevně stanovenou délku. [10] [11]

3 ARDUINO

Arduino je volně dostupná platforma od firmy Atmel. Původním záměrem projektu bylo vytvořit jednoduchou a levnou učební pomůcku pro studenty na technicky zaměřených školách. Postupem času se však ukázalo, že je Arduino velmi všestranné a používají ho i pokročilí lidé v oblastech slaboproudé elektrotechniky i programování.

3.1 Vývojová deska

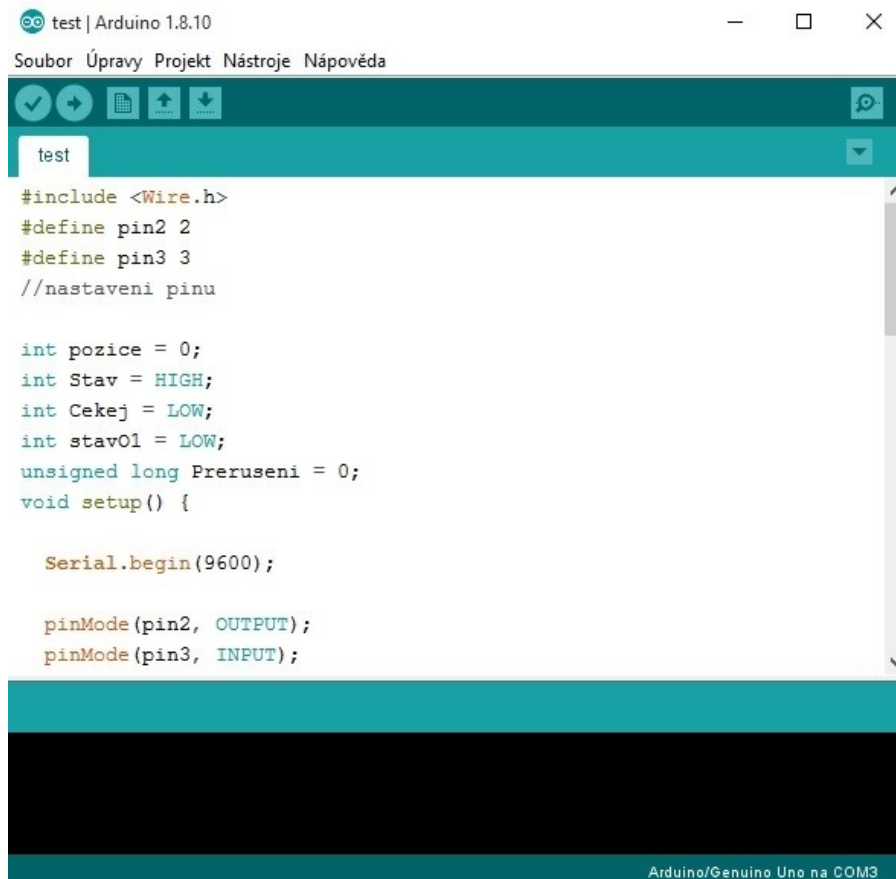
Základ Arduino tvoří vývojová deska. Vyrábí se několik typů těchto desek, které jsou od sebe rozlišné svou velikostí, počtem digitálních a analogových pinů, velikostí FLASH paměti, taktovací frekvencí, napájecího napětí atd. Na každé desce nalezneme napájecí zdroj 5V, port pro komunikaci s počítačem, mikropočítač a krystal. Jednotlivé desky jsou pak rozšířeny o další součásti. Po několika letech od začátku prodeje se začaly objevovat první kopie desek. Jelikož si Arduino zakládá na principu open-source, vývojáři z Číny převzali jejich nápady a začali ve velkém vyrábět a distribuovat do Evropy klony, které jsou i několikrát levnější než originální desky. K nim se dá připojit tzv. shield, což je rozšiřovací modul se vstupními i výstupními prvky jako je displej, potenciometr, LED diody, relé a další. Je však potřeba ho vybrat přesně, především kvůli rozmístění pinů. Výše zmíněné prvky se dají samozřejmě připojit také samostatně pomocí vodičů. Arduino nabízí svým zákazníkům nespočet možných typů čidel a senzorů, kupříkladu senzor teploty vody, tlaku, plynů, kapalin, barev, čidla pohybu, překážek, snímač otisku prstu. [12]

3.2 Vývojové prostředí

Druhou neméně důležitou částí je software, ve kterém se deska a její komponenty programují. Existuje několik typů prostředí nabízející funkce jak pro začátečníky, tak i pro pokročilé či zkušené programátory. Vhodný výběr prostředí pak výrazně zjednoduší práci s Arduinem.

3.2.1 Arduino IDE

Pro začátečníky je nejlepší použít Arduino IDE. Obsahuje jednoduchý textový editor, jenž podporuje programovací jazyky C a C++. Pro zjednodušení programování je k dispozici množství knihoven, ty lze zdarma stáhnout ze stránek Arduina a jejich instalace je pak velmi snadná. Prostředí nabízí snadné ovládání ve více než 30 jazycích světa. Je možné ho provozovat na běžných operačních systémech jako Windows, Mac OS X i Linux.



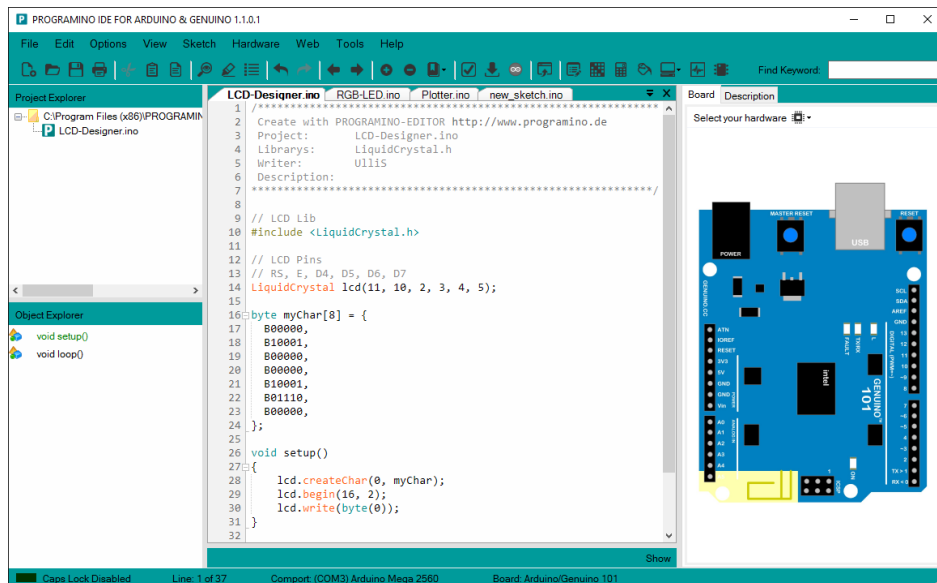
Obrázek 7 Vývojové prostředí Arduino IDE

Na horní bílé liště se nachází základní funkce pro ovládání celého prostředí. Pod lištou jsou tlačítka pro ovládání textového editoru – přeložení kódu, nahrání do desky, vytvoření nového souboru, otevření již vytvořeného souboru, uložení souboru a sériový monitor.

3.2.2 Programino

Programino IDE je alternativa Arduino IDE s několika funkcemi navíc. Lze jej bez potíží použít s deskami Arduino nebo kompatibilními mikropočítači. Podporuje jazyk C i C ++, nicméně nabízí podporu i pro další jazyky jako HTML, Javascript atd. Programino disponuje výhodou v zobrazení vývojové desky přímo v softwaru, takže si programátor nemusí např. pamatovat čísla pinů. Zmíněné výhody však tlumí fakt, že software nelze zdarma stáhnout.

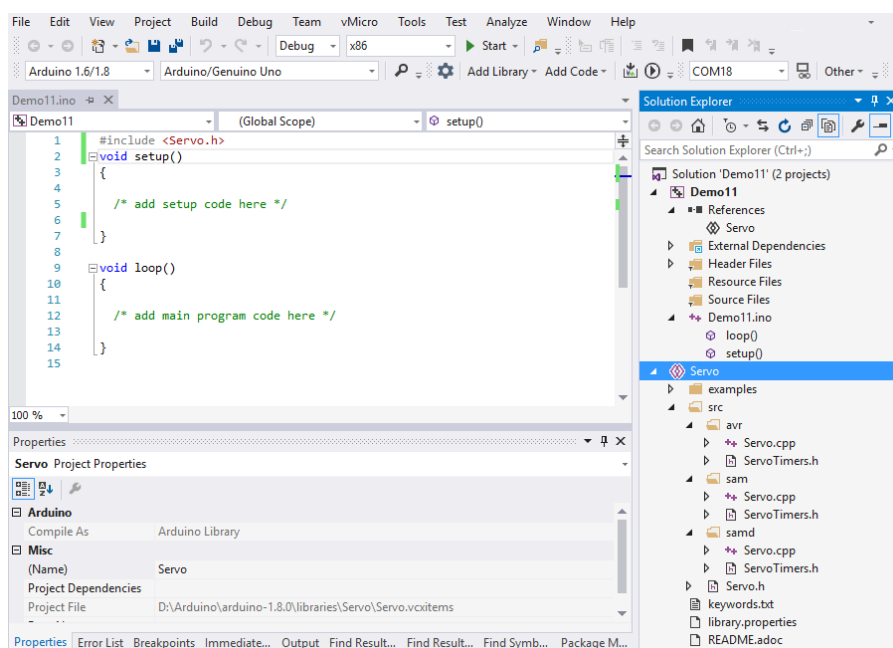
[13]



Obrázek 8 Vývojové prostředí Programino [13]

3.2.3 Visual Studio

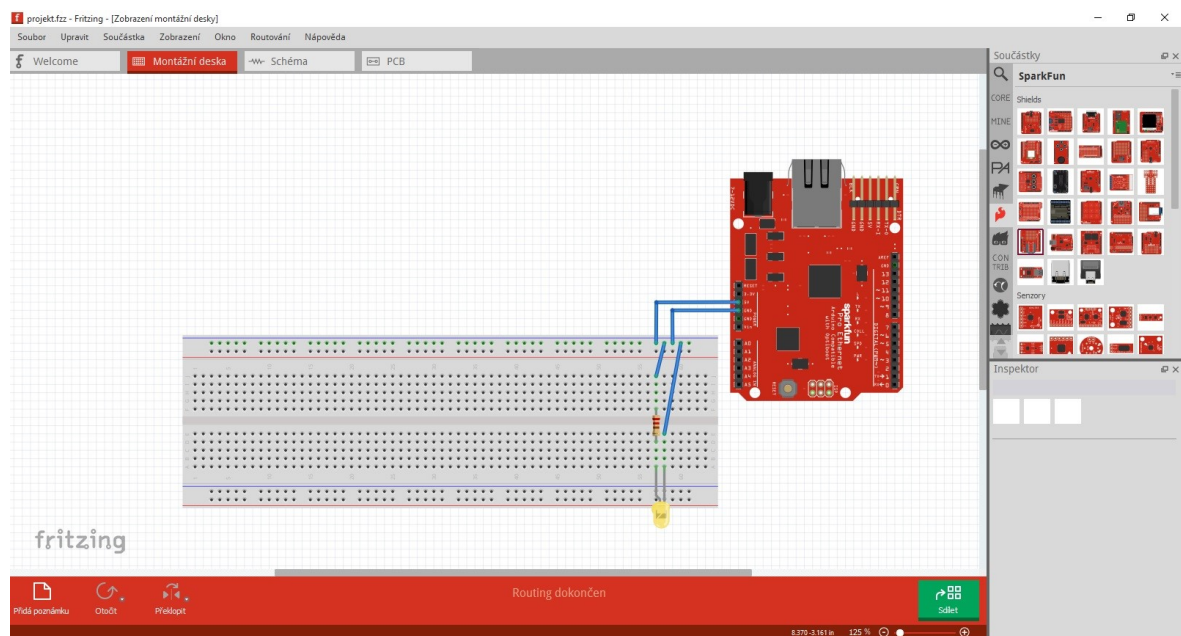
Pro zkušené programátory, kteří píšou složitější kódy, je tu Visual Studio od společnosti Microsoft. Je zcela zdarma a existuje několik verzí pro systémy Linux, Windows a Mac OS X. Podporuje téměř všechny hlavní programovací i webové jazyky. Z Visual Studio Marketplace pak lze zdarma nebo za úhradu stáhnout frameworky pro podporu stovek dalších. Při práci s Arduinem je toto prostředí výhodné zejména pro možnost krokovat program, ale přidání potřebné knihovny již vyžaduje potřebné znalosti. [14]



Obrázek 9 Vývojové prostředí Microsoft Visual Studio [14]

3.2.4 Fritzing

Pro sestavení funkčního obvodu na platformě Arduino je možné využít open-source software Fritzing. Ten disponuje obrovským množstvím použitelných součástek i platform jako Arduino, Raspberry či SparkFun. Volně dostupné knihovny pak počet komponentů v programu mohou rozšířit. Uživatelské rozhraní nabízí plynulé přepínání mezi třemi režimy: montážní deska, schéma a PCB (z anglického Printed Circuit Board, česky deska plošných spojů). V režimu montážní desky uvidí uživatel všechny součástky tak, jak budou vypadat ve skutečnosti a lze je rovnou mezi sebou propojit. Schéma slouží k zobrazení těchto součástek v podobě normalizovaných schématických značek. V PCB si lze součástky a jejich vývody osadit na desku plošných spojů i rozmístit vodivých drah po desce.



Obrázek 10 Software Fritzing

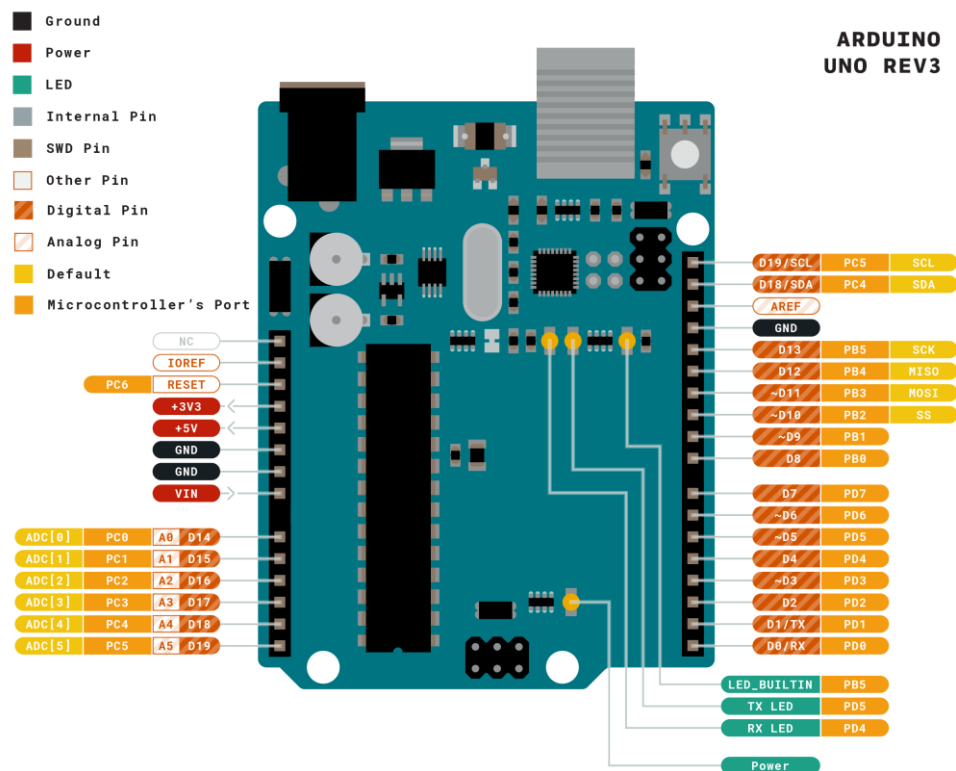
II. PRAKTICKÁ ČÁST

4 VÝBĚR KOMPONENTŮ

Při výběru komponentů je nutné zvážit několik aspektů, které souvisí s funkčností, ovládáním či možností změnit nastavení celého poplachového zabezpečovacího systému. Vhodným výběrem také studenti mohou získat znalosti, proč se některé detektory používají v praxi častěji než jiné.

4.1 Arduino Uno R3

Základem každého elektronického nebo elektrického systému musí být nějaký počítač, rozvaděč atd. V tomto případě je ústředna PZS nahrazena mikropočítačovou vývojovou deskou Arduino, konkrétně typem UNO R3 založeným na ATmega328. Tato verze má dostatečný počet pinů, rychlost flash paměti i USB konektor pro nahrání programu. Pro rozsáhlejší aplikace slouží verze MEGA a DUE. Deska v tomto případě slouží jako „mozek“, výpočetní jednotka. Přijímá stavy či informace z jednotlivých detektorů, např. jazýčkového kontaktu, PIR detektoru, ale i třeba senzoru teploty a vlhkosti. Nahráný program v desce pak má za cíl tyto stavy správně vyhodnotit a jednat dle programu. Obvykle se tímto způsobem ovládají výstupy (výpis na displej, sepnutí relé, zvuk sirény, rozsvícení/zhasnutí LED diod apod.).



Obrázek 11 Schéma pinů Arduino Una R3, upraveno [15]

Tabulka 1 Technické parametry – procesor ATmega 328
a Arduino Uno R3 [16]

Parametr	Hodnota
Frekvence procesoru	až 20 MHz
FLASH paměť	32 KB
EEPROM paměť	1 KB
SRAM paměť	2 KB
8 bitový časovač	2x
16 bitový časovač	1x
USART	2x
I2C	2x
SPI	1x
Digitální vstupy/výstupy	14x
PWM výstupy	6x
Analogové vstupy	6x
Vstupní napětí	6 – 12 V
Výstupní napětí	3,3 V, max. proudový odběr 180mA 5 V, max. proudový odběr 800mA

4.2 16x2 LCD displej 1602 modrý + I²C převodník

Pro výpis stavů slouží v jednotlivých programovacích prostředích sériový monitor neboli konzole. Pokud však model nebude připojen k počítači nebo jej uživatel nebude využívat primárně pro model, je vhodné si stavy vypsát na displej. Displej totiž zjednodušuje obsluhu modelu. Předpokládá se, že obsluha nebude mít problém přečíst znaky o velikost 5mm, proto byl vybrán dvouřádkový LCD displej o 16 znacích na jeden řádek. Intenzitu osvětlení lze nastavit trimrem. Hlavním kritériem byla nízká cena a základní použití. Ve větším provedení či při použití jiné technologie (TFT, OLED) by tyto vlastnosti nebyly příliš využity a ztrácely by význam. Velkou výhodou tohoto displeje je velmi snadné zapojení. Oproti levnější verzi stejného typu není potřeba připojovat minimálně 6 datových vodičů a dva napájecí vodiče.

Pomocí sběrnice I2C, která ke komunikaci používá piny SDA (Serial Data) a SCL (Serial Clock) a dva napájecí vodiče, jsme schopni ušetřit počet vodičů i digitální vstupy (piny) na desce, protože Arduino Uno R3 má pro SDA a SCL své vlastní piny určené pro tuto sběrnici.

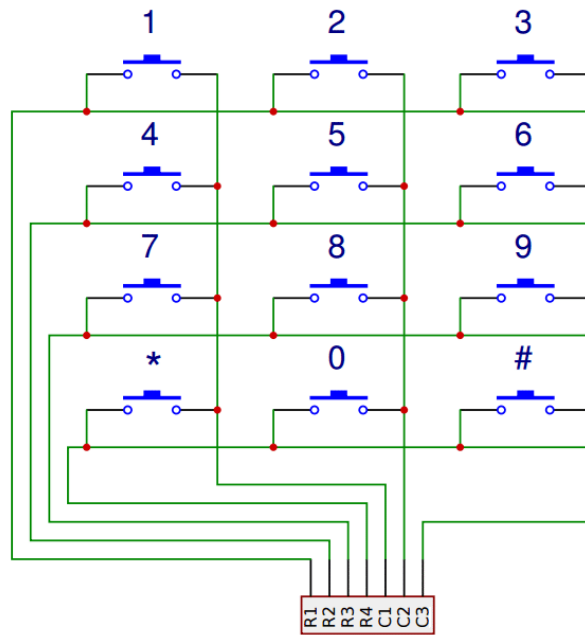
Tabulka 2 Technické parametry – 16x2 LCD displej 1602 modrý
+ I²C převodník [17]

Parametr	Hodnota
Rozlišení	2 řádky, 16 znaků
Barva	modrá
Napájení	5 V
Rozměry modulu	80x35x11 mm
Velikost zobrazovací plochy	64,5x16 mm
Řadič	HD44780

4.3 4x3 Maticová membránová klávesnice

Pro ovládání modelu slouží klávesnice. Těch na výběr není takové množství jako u ostatních komponentů. Nakonec byla zvolena membránová klávesnice, protože je několikanásobně levnější a konektory oproti tlačítkové nemusí být zapájené.

Jak klávesnice funguje, naznačuje Obrázek 15. Pod každým tlačítkem se nachází spínač. Tyto spínače jsou propojeny ve sloupcích a řádcích, tj. jeden sloupec má 4 spínače zapojené sériově, jeden řádek má 3 spínače. Pokud není stisknuto žádné tlačítko, tak sloupce jsou ve stavu HIGH a řádky v LOW. Při stisku tlačítka se stav sloupce změní na LOW a Arduino hledá řádek tak, že je postupně přepíná do stavu HIGH, až nalezne ten, kde je stav LOW. Tím přesně identifikuje stisknuté tlačítko.



Obrázek 12 Princip maticové klávesnice, upraveno [18]

Tabulka 3 Technické parametry – 4x3 Maticová membránová klávesnice [19]

Parametr	Hodnota
Počet tlačítek	12
Kontaktní odpor	500 Ω
Izolační odpor	100 M Ω
Velikost	77x69 mm

4.4 PIR detektor HC-SR505

Tento velmi často používaný detektor v praxi patří také mezi trojici použitých detektorů na výukovém modelu. Lze ho sehnat v několika různých provedeních s umístěním na stěnu, strop či jen volně položeným a připojeným k ústředně. Roli při výběru také hraje snímaný úhel a vzdálenost, na kterou je schopen zaznamenat pohyb, a jeho rozměry. V početné škále nabízených typů byl zvolen mini PIR detektor pohybu HC-SR505, jelikož veškeré parametry včetně nízké ceny odpovídají požadavkům.

Tabulka 4 Technické parametry – PIR detektor HC-SR505 [20]

Parametr	Hodnota
Napájení	4,5 – 20 V
Detekční dosah	cca 3 m
Detekční úhel	pod 100°
Provozní teplota	-20° až +80°
Rozměr	10x23 mm

4.5 IR snímač FC-51

V nabídce prvků pro předmětovou ochranu se jako vhodný jeví IR snímač FC-51. Velkou výhodou je možnost nastavení vzdálenosti detekce předmětu pomocí trimru.

Tabulka 5 Technické parametry – IR snímač FC-51 [21]

Parametr	Hodnota
Napájení	3,3 – 5 V
Detekční dosah	2 – 30 cm
Detekční úhel	35°
Rozměry	32x14mm
Komparátor	LM393

4.6 Magnetický kontakt SM-35

Třetím detektorem na výukovém modelu je prvek plášťové ochrany magnetický kontakt. Na trhu se vyskytuje v nepřeberném množství od různých výrobců, v různých barvách, použitých materiálech apod. Typ SM-35 představuje základní provedení takového detektoru. Jazyčkový kontakt připevněný na desce neobsahuje přívodní vodiče, které někdy bývají zapouzdřené uvnitř kontaktu. Při zapojení do Arduina je vhodné používat vodiče, jejichž průměr je dimenzován na piny Arduina tak, aby nedocházelo k vypadávání a ztrátě kontaktu.

Tabulka 6 Technické parametry – Magnetický kontakt SM-35 [22]

Parametr	Hodnota
Detekční dosah	31 cm
Tamper	ne
Rozměry	65x15x12 mm
Montáž	povrchová pomocí šroubů nebo lepidla

4.7 Aktivní bzučák 5V

Součástí poplachového zabezpečovacího systému je aktivní bzučák mající funkci sirény. Oscilační obvod začne vydávat zvuk, když je k němu připojeno stejnosměrné napětí. Pasivní bzučák vyžaduje střídavý signál k vytvoření zvuku, protože uvnitř bzučáku není oscilační obvod.

Tabulka 7 Technické parametry – Aktivní bzučák 5V [23]

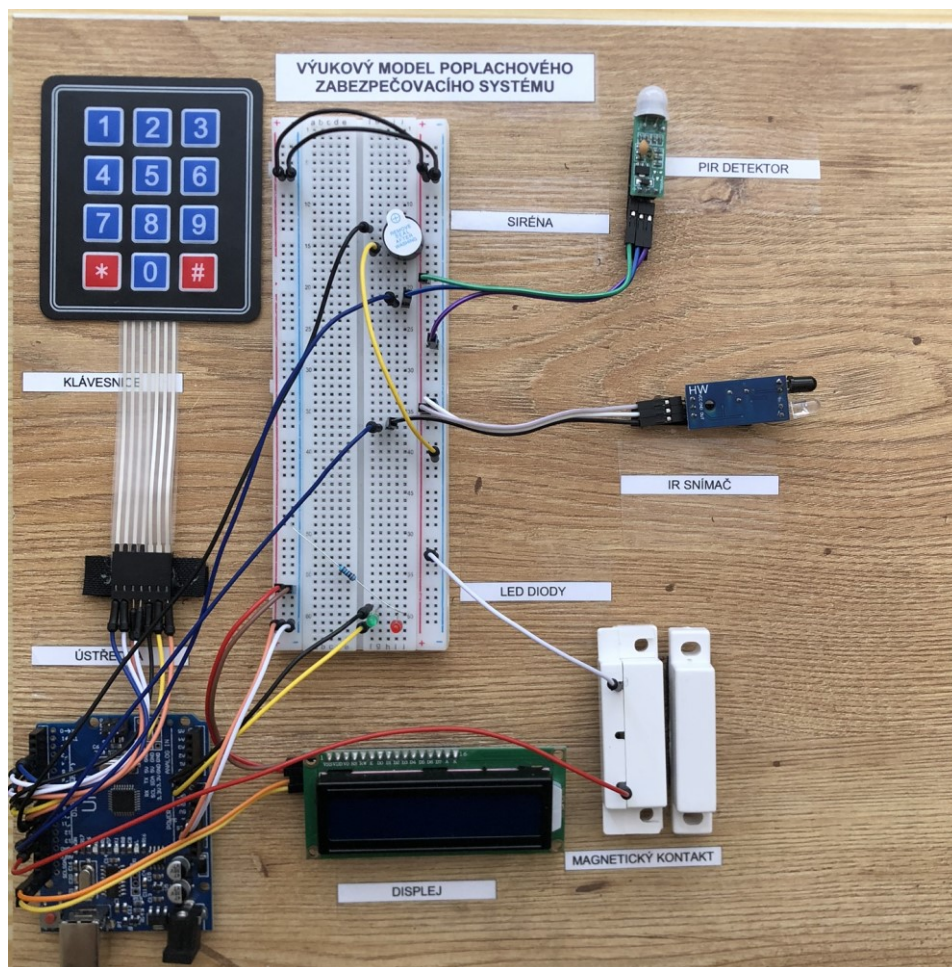
Parametr	Hodnota
Napájení	5 V
Proudový odběr	do 25 mA
Frekvence	2300 +- 500 Hz
Rozměry	12x9,5 mm

4.8 LED dioda 3mm

K optické signalizaci zastřežení nebo spuštění poplachu jsou zde 2 LED diody v různém barevném provedení.

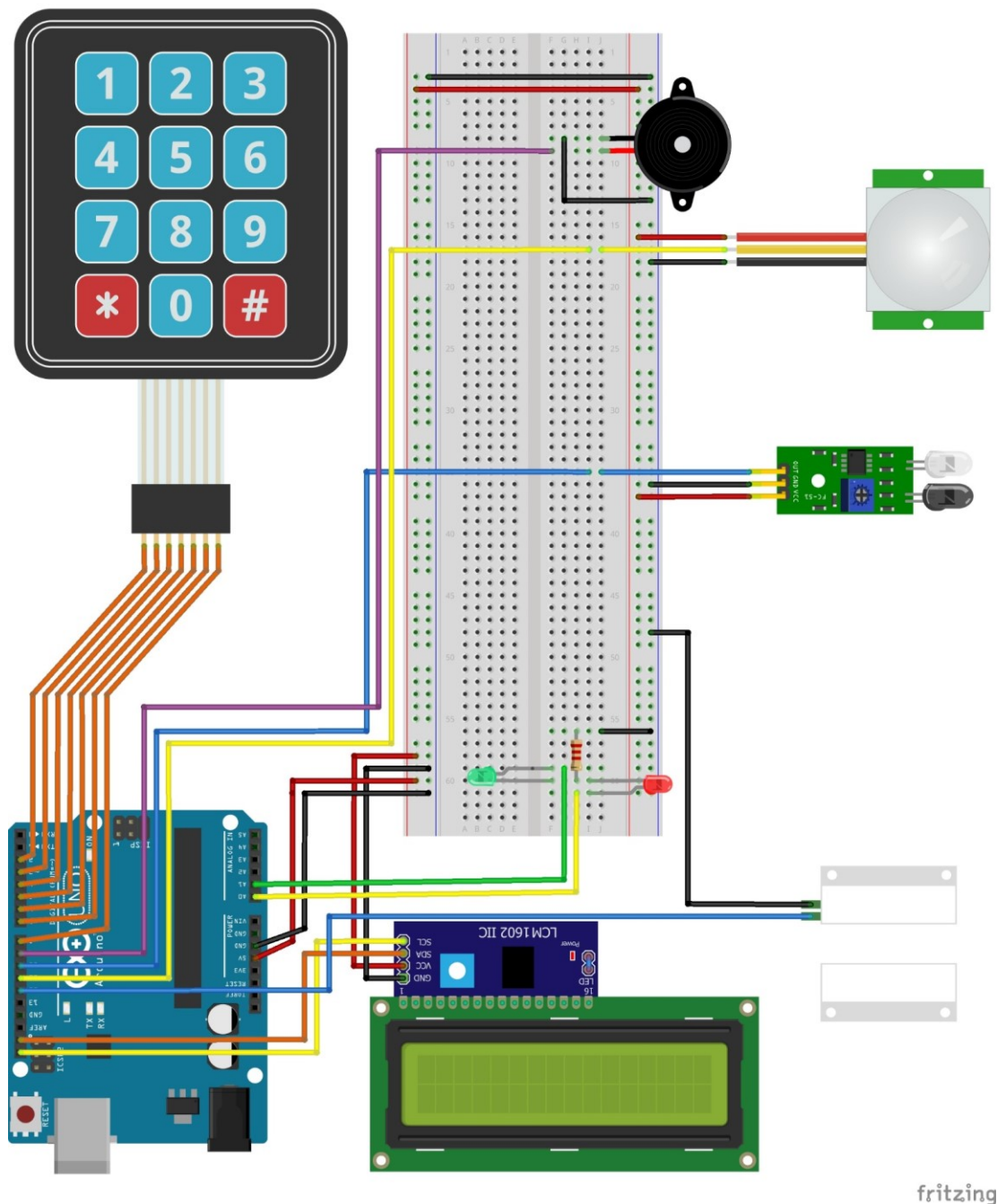
5 ROZMÍSTĚNÍ A ZAPOJENÍ KOMPONENTŮ

Všechny komponenty systému jsou rozmístěny na sololitové desce o rozměrech 30 x 30 cm. Na Obrázku 16 lze toto rozmístění názorně vidět. Ústředna systému se nachází v levém dolním rohu. Lze ji napájet přes USB kabel typu B nebo konektorem DC 5,5/2,1 mm pro 9V. Je otočena tak, aby se přívodní kabel zbytečně nepletl mezi ostatní komponenty. Nad ústřednou je zapojena membránová klávesnice. Ve středu desky leží nepájivé kontaktní pole kvůli tomu, aby při změně zapojení neleželi na desce dlouhé kabelové propojky. V poli jsou navíc zapojeny součástky, které nelze snadno připevnit na desku. Pod nepájivým kontaktním polem je displej. Napravo od pole jsou rozmístěny detektory. PIR detektor byl záměrně umístěn a natočen směrem od obsluhy, aby neustále nezachytával pohyb kolem něj. IR snímač má před sebou na desce volnou plochu pro chráněný předmět. Magnetický kontakt s permanentním magnetem u pravého rohu desky vyplňuje místo vedle displeje. Jednotlivé komponenty mají u sebe štítky pro snadnější rozpoznání.



Obrázek 13 Výukový model PZS

Na Obrázku 17 je přehledně znázorněno zapojení systému pomocí softwaru Fritzing. Barvy vodičů jsou upraveny tak, jak je tomu obvyklé v praxi – červená pro napájení, černá pro uzemnění, zbylé barvy pro komunikaci. Veškeré komponenty napájí Arduino pěti volty buď ze svorkovnice v nepájivém poli nebo z digitálních/analogových vstupů. LED diody jsou zapojeny tak, že každá má svůj vlastní pin z Arduino (A0, A1), ale výstup s odporem 2,2kΩ mají společný.



Obrázek 14 Grafické schéma zapojení

6 PROGRAMOVÉ VYBAVENÍ

V této kapitole je vysvětleno programového vybavení celého poplachového zabezpečovacího systému. Celý zdrojový kód byl vytvořen ve vývojovém prostředí Arduino IDE, jelikož se běžně využívá v předmětech, kde se mikropočítače typu Arduino Uno a jiné programují. Je rozdělen na několik částí, které jsou postupně seřazeny od direktiv, deklarací a inicializací proměnných až po samotné funkce.

System funguje následujícím způsobem: po zapnutí je nejprve vyžadováno zadání správného hesla, které je aktuálně nastaveno na „1234“. V případě zadání špatného hesla se cyklus neustále opakuje. Pokud je heslo správné, spustí se desetisekundový odpočet pro konfiguraci systému. Pak jsou aktivovány detektory a rozsvítí se zelená dioda na znamení zastřežení systému. Při sepnutí některého detektoru do stavu poplachu se volá funkce pro zadání hesla, zelená dioda zhasne a rozsvítí se červená. Pro snadnější pochopení systému i samotné funkce má každá (vyjma funkce loop ()) svůj vývojový diagram, kde jsou příkazy řazeny do bloků.

6.1 Knihovny

Moduly připojené k Arduino (např. klávesnice, displej, váhový sensor) velmi často používají knihovny. Ty mají své opodstatnění při zjednodušení kódu – stačí jeden nebo pár řádků k tomu, abychom dostali požadovaný výsledek (výpis na displej, kalibrace váhy). Přitom nemusíme psát mnohdy dlouhé a složité řádky kódu. Pro vložení knihoven se používá direktiva `#include`. Knihovna **Wire.h** jde zde kvůli komunikaci po sběrnici I2C, **LiquidCrystal_I2C.h** pro displej a **Keypad.h** pro klávesnici. Použité knihovny jsou na internetu volně dostupné třeba na GitHubu. [24] [25]

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Keypad.h>
```

Obrázek 15 Inicializace knihoven

6.2 Deklarace, inicializace proměnných a konstant

Pro deklaraci proměnných nebo konstant můžeme použít dvě možnosti. Jedna z nich je pomocí `#define` v hlavičce programu. Syntaxe zápisu vypadá následovně: `#define pinPIR 11` přiřazuje pin číslo 11 na Arduino konstantě pinPIR. Pomocí `#define` lze

také deklarovat konstanty příkazu switch, konstanta musí být celočíselná. Je vhodné, aby se tato čísla nerovnal obsazeným pinům na Arduinu.

```
#define pinPIR 11
#define ZADNY 14
#define poplachPIR 15
#define poplachIR 16
#define poplachMAG 17
```

Obrázek 16 Deklarace pomocí define

Ta druhá možnost je **int var**. Na prvním místě se zapisuje datový typ (int, char, boolean atd.), na druhém místě název proměnné. Do ní nemusí být ihned vložena hodnota, ale je to vhodné, jelikož by se do ní při startu programu mohla dostat náhodná hodnota, což by mohlo způsobit zmatek. Tomuto kroku, kdy se přiřazuje proměnné hodnota, se říká inicializace. Například **const int Magnet = 9**; znamená, že pin číslo 9 vysílá data do proměnné Magnet. Datový typ jednoznačně určuje její velikost při překladu programu. Významný rozdíl oproti define je v syntaxi i v místě uložení (const int se ukládá do paměti flash, zatímco pouze int do SRAM, které není mnoho).

Tato možnost deklarace může být ještě rozšířena o další instrukce, např. const nebo unsigned. **Const var** představuje proměnnou, kterou nelze v průběhu programu měnit, do **unsigned var** zase nelze vložit záporné hodnoty.

```
int Stav = ZADNY;
const int Magnet = 12;
const int Sirena = 9;
int Cervena = A0;
int Zelena = A1;

int UrceniDetektoru = 0;
const int DelkaTonu1 = 100;
const int DelkaTonu2 = 1000;
const int DelkaTonu3 = 5000;
const int Frekvence = 3000;

int i = 0;
String ZadaneHeslo = "";
String SpravneHeslo = "1234";
boolean ZadejHeslo = true;
```

Obrázek 17 Deklarace pomocí datového typu

Pro navázání komunikace mezi klávesnicí a Arduinem byla použita knihovna Keypad.h. První dva příkazy deklarují proměnné se čtyřmi řádky a třemi sloupci. Následuje pole definující, které znaky se odešlou při stisknutí konkrétního tlačítka na klávesnici. Znaky jsou rozloženy tak, jak se objevují na klávesnici. Další dva řádky nastavují čísla pinů pro řádky a sloupce tak, jak jsou zapojeny do Arduina (Kapitola 4.3). Poslední příkaz je vytvoření instance. Tato část kódu byla převzata z [26], proto zde nejsou upraveny názvy proměnných. Pouze došlo k úpravě čísel pinů, které jsou jinak zapojeny.

Inicializace displeje probíhá přes knihovnu LiquidCrystal_I2C.h. V objektu s názvem lcd jsou zadány tři parametry: nastavení adresy I²C, počet znaků a počet řádků.

```
const byte ROWS = 4;
const byte COLS = 3;
char keys[ROWS][COLS] =
{
    {'1', '2', '3'},
    {'4', '5', '6'},
    {'7', '8', '9'},
    {'*', '0', '#'}
};
byte rowPins[ROWS] = {3, 8, 7, 5};
byte colPins[COLS] = {4, 2, 6};
Keypad kpd = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

Obrázek 18 Deklarace a inicializace klávesnice a displeje

6.3 Funkce setup

Funkce setup vykonává příkazy uvnitř složených závorek jen jednou po startu programu. Tím se myslí buď nahrání nového programu do Arduina, stisknutí tlačítka reset nebo připojení Arduina ke zdroji napájení. V programu musí vždy existovat, ačkoliv nemusí obsahovat nějaké inicializační příkazy jako např. nastavení sériové komunikace, displeje, pinů apod. Používá datový typ void, protože nevrací žádnou hodnotu, zároveň do ní nic nevstupuje.

První tři příkazy inicializují piny pro jednotlivé detektory, další dva pro diody. Příkaz **pinMODE** má dva 2 parametry – prvním je číslo pinu, druhým je funkce (vstup nebo výstup). Proměnné pinPIR náleží hodnota 11, která je deklarována direktivou define výše, a funguje jako vstup (INPUT). Další způsob je: **pinMode(10, INPUT);**, kde číslo 10 je číslo pinu, INPUT opět vstup. Tento způsob zápisu není příliš vhodný, jelikož není tak přehledný,

jako když se číslo pinu uloží v datovém typu a ten pak vstupuje do příkazu pinMODE. Třetí způsob nastavuje pin jako INPUT_PULLUP. To znamená, že když je detektor vypnutý, je na pinu hodnota HIGH. V případě magnetického kontaktu musí být použit. V čipu Atmega jsou zabudovány 20K pullup rezistory, k nim se lze dostat právě pomocí tohoto příkazu. Příkaz **pinMode(Zelena, OUTPUT)**; nastavuje proměnnou Zelena jako výstup.

Příkaz **lcd.begin()**; inicializuje rozhraní na displej. Jako parametry se uvádí počet sloupců a řádků, k tomu však došlo již v objektu výše, proto zde parametry nejsou.

Pokud chceme využít možnosti výpisu na sériovou linku, ve funkci setup musí být příkaz **Serial.begin(9600)**; Číslo 9600 udává rychlost přenosu dat v bitech za sekundu (baud) pro sériový přenos dat. Je možné ho v programu změnit, avšak i v sériovém monitoru vpravo dole musí být změněno. Poslední příkaz vypíše do sériového monitoru text ve dvojitých uvozovkách.

```
void setup()
{
  pinMode(pinPIR, INPUT);
  pinMode(10, INPUT);
  pinMode(Magnet, INPUT_PULLUP);
  pinMode(Zelena, OUTPUT);
  pinMode(Cervena, OUTPUT);
  lcd.begin();
  Serial.begin(9600);
  Serial.println("Zadej heslo: ");
}
```

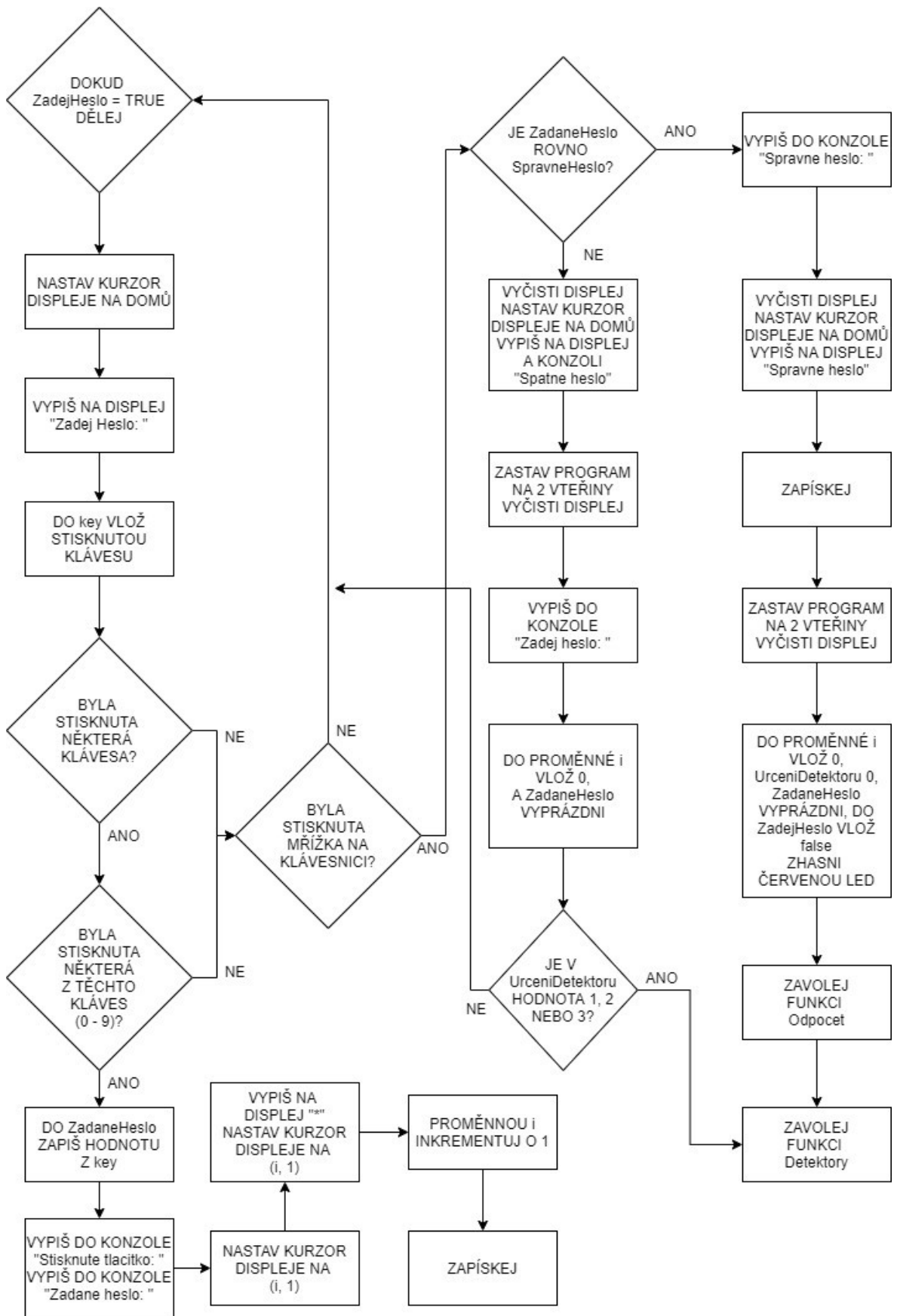
Obrázek 19 Funkce setup

6.4 Funkce Smycka

Jakmile se provede funkce setup(), následuje ji funkce loop(). Příkazy uvnitř této funkce se provádí pořád dokola v nekonečné smyčce. Bývá obvykle nejdůležitější částí programového vybavení. V tomto případě však obsahuje pouze jeden příkaz, proto nemá vlastní kapitolu a je o ní zmínka zde. **Smycka()**; volá funkci Smycka.

Při prvním pohledu na vývojový diagram funkce Smycka (Obrázek 23) se může zdát, že je funkce příliš složitá. Přitom opak je pravdou. Cyklus **while** kontroluje proměnnou ZadejHeslo, zda se v ní nachází hodnota true. Pokud je toto tvrzení pravdivé, tak se kurzor displeje nastaví do levého horního rohu a zleva doprava vypíše „Zadej heslo: “ –

lcd.print("Zadej heslo:");. Do lokální proměnné `key` vložíme hodnotu stisknuté klávesy. Podmínka `key != NO_KEY` se nevykoná, pokud nebylo stisknuto žádné tlačítko, které je nastaveno v `Keypad.h` jako `NULL char` (char s hodnotou 0). Když je podmínka splněna, tak se opět pomocí další podmínky snaží program zjistit, které tlačítko (pouze však čísla 0 až 9) bylo stisknuto. Proměnná `key` předá toto číslo další proměnné, viz. **`ZadaneHeslo += key;`**. Následuje série příkazů, které stisknuté číslo i zadávané heslo zobrazí v konzoli, na displeji je však místo čísla i hesla zobrazen symbol hvězdičky (**`lcd.print("*");`**) a jeho pozice je na druhém řádku displeje (**`lcd.setCursor(i, 1);`**). Následuje inkrementace proměnné `i`. Příkaz **`tone(Sirena, Frekvence, DelkaTonu1);`** provede krátké „pípnutí“ na znamení stlačení klávesy. Parametry se mohou napsat přímo dovnitř příkazu nebo se deklarují v hlavičce programu, což bývá přehlednější při možné úpravě.



Obrázek 20 Vývojový diagram funkce Smyčka

```
void loop()
{
  Smycka();
}

void Smycka ()
{
  while (ZadejHeslo)
  {
    lcd.home();
    lcd.print("Zadej heslo:");

    char key = kpd.getKey();

    if (key != NO_KEY)
    {
      if (key == '0' || key == '1' || key == '2' || key == '3' ||
          key == '4' || key == '5' || key == '6' || key == '7' ||
          key == '8' || key == '9' )
      {
        ZadaneHeslo += key;
        Serial.print("Stisknute cislo: ");
        Serial.println(key);
        Serial.print("Zadane heslo: ");
        Serial.println(ZadaneHeslo);
        lcd.setCursor(i, 1);
        lcd.print("*");
        i = i + 1;
        tone(Sirena, Frekvence, DelkaTonu1);
      }
    }
  }
}
```

Obrázek 21 Funkce Smycka, první část

Když se při zadávání hesla stiskne na klávesnici mřížka, obsluha systému tím dává najevo, že zadané heslo by mělo být shodné s tím pravým. Tuto skutečnost ověřuje další podmínka.

V případě úspěchu se provedou následující příkazy: na sériový monitor se vypíše hláška spolu se správně zadaným heslem, na displej se vypíše jen hláška, nikoliv heslo, proměnné `i` a `ZadaneHeslo` se vynulují, respektive vyprázdní, spustí se tón pro úspěšné zadání hesla, program se pozastaví, aby nápis na displeji šel přečíst, pak se displej vyčistí, `UrceniDetektoru` se vynuluje, `ZadejHeslo` se změní na `false`, aby program poté vyskočil

ven z cyklu, červená dioda zhasne a volají se dvě následující funkce (Odpocet a Detektory).

Pokud zadané heslo neodpovídá pravému heslu, tak následuje: výpis hlášky na sériový monitor, výpis hlášky na displej s časovou prodlevou 2 vteřin, vyprázdnění a vynulování dvou proměnných, vyčištění displeje a podmínka, která při splnění logického výroku uvnitř závorek, zavolá funkci Detektory (funkce Odpocet je tím pádem vynechána).

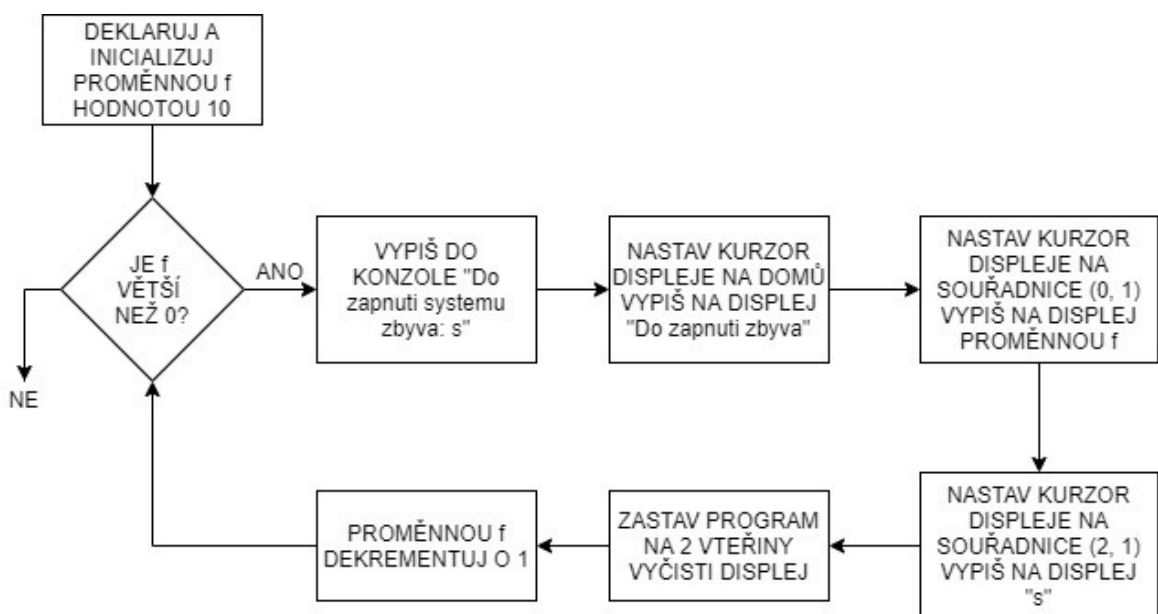
```
if (key == '#')
{
  if (ZadaneHeslo == SpravneHeslo)
  {
    Serial.print("Spravne heslo: ");
    Serial.println(ZadaneHeslo);
    lcd.clear();
    lcd.home();
    lcd.print("Spravne heslo");
    i = 0;
    ZadaneHeslo = "";
    tone(Sirena, Frekvence, DelkaTonu2);
    delay(2000);
    lcd.clear();
    UrceniDetektoru = 0;
    ZadejHeslo = false;
    digitalWrite(Cervena, LOW);
    Odpocet();
    Detektory();
  }
  else
  {
    Serial.println("Spatne heslo");
    lcd.clear();
    lcd.home();
    lcd.print("Spatne heslo");
    delay(2000);
    Serial.println("Zadej heslo: ");
    ZadaneHeslo = "";
    i = 0;
    lcd.clear();
    if (UrceniDetektoru == 1 || UrceniDetektoru == 2 || UrceniDetektoru == 3)
    Detektory ();
  }
}
}
```

Obrázek 22 Funkce Smyčka, druhá část

6.5 Funkce Odpocet

Tato funkce je zde kvůli konfiguraci systému. Po zadání správného hesla má obsluha 10 vteřin na to, aby všechny detektory byly připraveny k vyhlášení poplachu.

Prvním příkazem se deklaruje a inicializuje proměnná *f*, která je typu **unsigned int**. V cyklu **for** se kontroluje podmínka, zda je *f* větší než 0. Při splnění podmínky se do sériového monitoru a na displej postupně vypisuje zbývající čas. S každým proběhnutím přes cyklus se proměnná *f* sníží o „jedničku“ do doby, než není splněna podmínka. Pak se program vrací zpět funkce Smyčka a je volána funkce Detektory.



Obrázek 23 Vývojový diagram funkce Odpocet

```

void Odpocet ()
{
    unsigned int f = 10;
    for (f; f > 0; f--)
    {
        Serial.println((String)"Do zapnutí systému zbyva: "+f+" s");
        lcd.home();
        lcd.print("Do zapnutí zbyva");
        lcd.setCursor(0, 1);
        lcd.print(f);
        lcd.setCursor(2, 1);
        lcd.print("s");
        delay(1000);
        lcd.clear();
    }
}
  
```

Obrázek 24 Funkce Odpocet

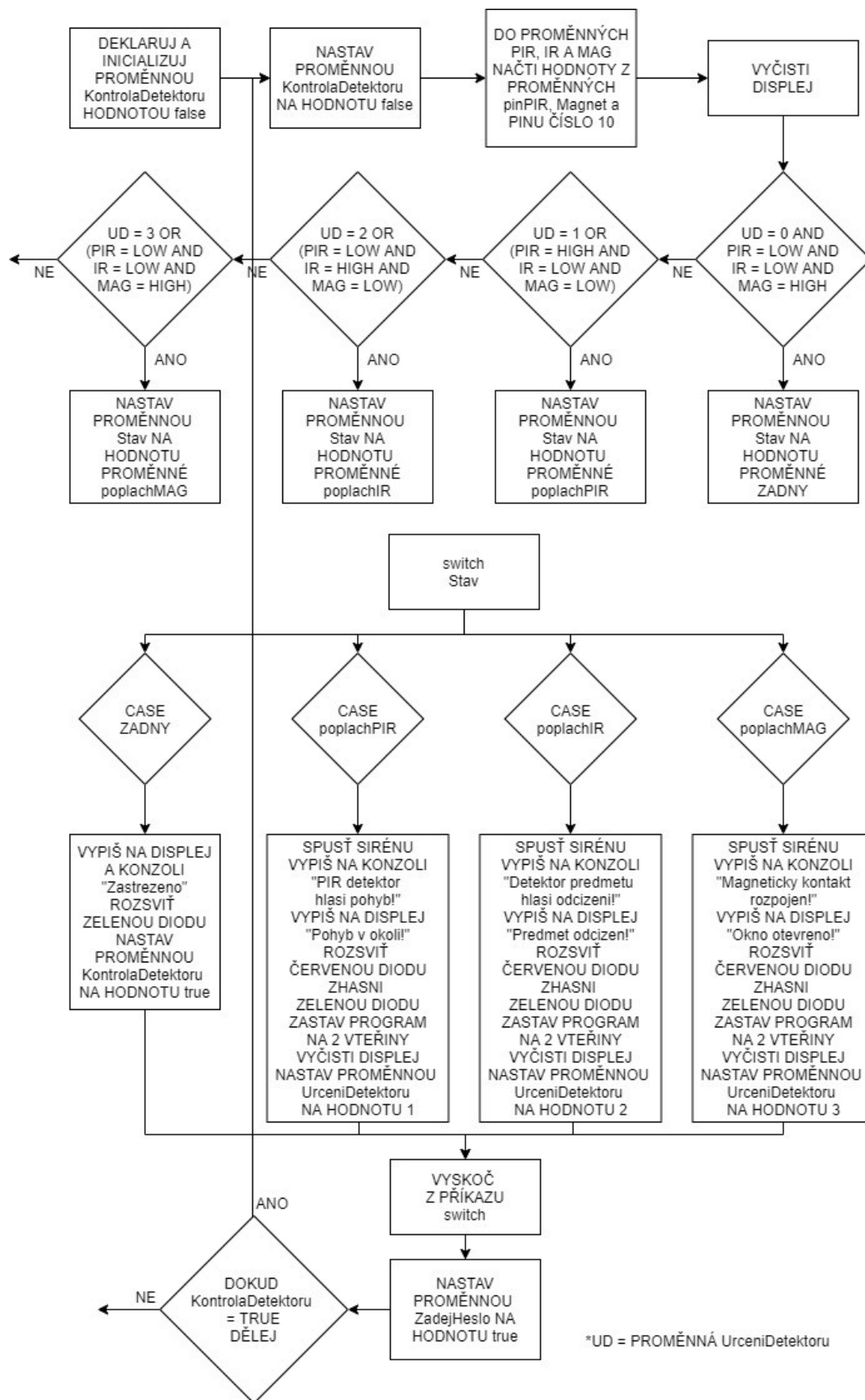
6.6 Funkce Detektory

Poslední částí programového vybavení je funkce Detektory, ve které se sledují stavy z detektorů a taky zde probíhá vyhodnocení těchto stavů.

Prvním příkazem se deklaruje a inicializuje lokální proměnná typu boolean `KontrolaDetektoru` na stav `false`. Cyklus `do-while` se provede alespoň jednou. Pak se do tří znovu lokálních proměnných pomocí funkce `digitalWrite` načtou hodnoty z detektorů. Parametrem v kulatých závorkách je buď proměnná nebo číslo pinu na Arduinu. Po nastavení cursoru displeje do levého horního rohu se díky čtyřem po sobě jdoucím podmínkám do proměnné `Stav` dostane hodnota deklarovaná v hlavičce programu, jiná možnost nikdy nenastane.

Příkaz `switch` zde výrazně zjednodušuje orientaci ve funkci Detektory. Princip je velmi prostý: podle toho, jaká hodnota je v proměnné `Stav`, se provede příslušná část „`case`“. Zbylé části jsou vynechány, pokud se před nimi nachází příkaz `break` říkající programu, ať „vyskočí“ ze switche a pokračuje dále. Občas se za posledním „`case`“ uvádí ještě příkaz `default`, což značí stav, který neodpovídá žádné hodnotě „`case`“ a vykonají se příkazy uvnitř `default`.

Proměnná `ZadejHeslo` se přepne do `true`, aby se mohla vykonat funkce `Smycka`. Cyklus `do-while` kontroluje stav v proměnné `KontrolaDetektoru`. Pouze v `case ZADNY` se tato proměnná změní do stavu `true`, aby cyklus mohl znovu vykonat příkazy, jinak by skončil a program by se vrátil na funkce `loop`. V ostatních `case` se proměnná nezmění, to značí, že některý detektor byl sepnut. Po provedení již známých příkazů (výpis na konzoli a displej, rozezvučení sirény atd.) si program zapamatuje detektor, jenž spustil poplach. Při zadání špatného hesla pro odkódování se ověří, jestli funkce `Smycka` byla zavolána z důvodu vyhlášení poplachu některého z detektorů (ověření probíhá ve funkci `Smycka`). Pokud se tato podmínka potvrdí, přes další podmínku (již ve funkci Detektory) se program dostane opět do `case`, kde byl vyhlášen poplach.



Obrázek 25 Vývojový diagram funkce Detektoru

```
void Detektory()
{
  boolean KontrolaDetektoru = false;
  do
  {
    KontrolaDetektoru = false;

    int PIR = digitalRead(pinPIR);
    int IR = digitalRead(10);
    int MAG = digitalRead(Magnet);
    lcd.home();

    if (UrceniDetektoru == 0 and PIR == LOW and IR == LOW and MAG == LOW)
    {
      Stav = ZADNY;
    }
    else if (UrceniDetektoru == 1 or (PIR == HIGH and IR == LOW and MAG == LOW))
    {
      Stav = poplachPIR;
    }
    else if (UrceniDetektoru == 2 or (PIR == LOW and IR == HIGH and MAG == LOW))
    {
      Stav = poplachIR;
    }
    else if (UrceniDetektoru == 3 or (PIR == LOW and IR == LOW and MAG == HIGH))
    {
      Stav = poplachMAG;
    }
  }
}
```

Obrázek 26 Funkce Detektory, první část

```
switch (Stav)
{
  case (ZADNY):
    Serial.println("Zastrezeno");
    lcd.print("Zastrezeno");
    digitalWrite(Zelena, HIGH);
    KontrolaDetektoru = true;
    break;
  case (poplachPIR):
    tone(Sirena, Frekvence, DelkaTonu3);
    digitalWrite(Cervena, HIGH);
    digitalWrite(Zelena, LOW);
    Serial.println("PIR detektor hlasi pohyb!");
    lcd.print("Pohyb v okoli!");
    delay(2000);
    lcd.clear();
    UrceniDetektoru = 1;
    break;
  case (poplachIR):
    tone(Sirena, Frekvence, DelkaTonu3);
    digitalWrite(Cervena, HIGH);
    digitalWrite(Zelena, LOW);
    Serial.println("Detektor predmetu hlasi odcizeni!");
    lcd.print("Predmet odcizen!");
    delay(2000);
    lcd.clear();
    UrceniDetektoru = 2;
    break;
  case (poplachMAG):
    tone(Sirena, Frekvence, DelkaTonu3);
    digitalWrite(Cervena, HIGH);
    digitalWrite(Zelena, LOW);
    Serial.println("Magneticky kontakt rozpojen!");
    lcd.print("Okno otevreno!");
    delay(2000);
    lcd.clear();
    UrceniDetektoru = 3;
    break;
}
ZadejHeslo = true;
}
while (KontrolaDetektoru);
}
```

Obrázek 27 Funkce Detektory, druhá část

ZÁVĚR

Cílem této práce bylo vytvořit hardwarové i softwarové vybavení pro výukový model PZS určený do předmětů pro programování mikropočítačů a PLC.

V teoretické části je popsán poplachový zabezpečovací systém, především jeho možnosti zapojení, rozdělení prvků a u použitých detektorů i jejich fyzikální princip tak, aby tyto zákonitosti byly v několika větech snadno pochopitelné. Dále je tato část zaměřena na krátký popis mikropočítačů a i samotné platformy Arduino, která je použita v praktické části.

Praktická část začíná optimální volbou výběru použitých komponentů systému jak z hlediska fyzikálního principu, tak i kompatibility s deskou Arduino. Díky tomu mohl být celý systém upevněn na sololitovou desku s rozměry 30 x 30 cm. Samotný návrh byl graficky zpracován v programu Fritzing, kde lze součástky nejen propojit vodivými spoji, ale i rozložit tak, jak jsou upevněny na desku. Rozmístění komponentů na sololitu je uspořádáno tak, aby na jedné straně byly detektory a na straně druhé zbylé komponenty. Všechny komponenty vyjma nepájivého pole a klávesnice jsou podlepeny suchým zipem proto, aby šlo změnit nejen programová část, ale i část hardwarová. Studenti si mohou bez použití šroubováků, klíčů a podobných nástrojů komponenty přemístit či odpojit a sejmut z desky.

Samotná obsluha systému je nenáročná a snadno pochopitelná. Ke většímu přehledu v programové části dopomáhají i vývojové diagramy, na kterých lze funkci celého systému graficky ukázat a lépe popsat než přímo ve zdrojovém kódu.

V příloze práce lze mimo hlavního programu nalézt i menší programy pro ověření funkčnosti jednotlivých detektorů tak, jak jsou zapojeny v celém systému. Tyto programy jsou i součástí přiloženého CD.

SEZNAM POUŽITÉ LITERATURY

- [1] UHLÁŘ, Jan. Technická ochrana objektů. II. díl, Elektrické zabezpečovací systémy. Praha: Policejní akademie České republiky, 2001, 205 s. ISBN 8072510762.
- [2] BURDA, Karel. Základy elektronických zabezpečovacích systémů. Brno: Akademické nakladatelství CERM, 2017, 123 s. ISBN 9788072049677.
- [3] DRGA, Rudolf. Elektronické bezpečnostní systémy: Poplachové zabezpečovací a tísňové systémy. Studijní výukový materiál. Zlín, 2013.
- [4] Bezpečnostní systémy: Základní funkce a rozdělení ústředí EZS. Studijní materiály - SŠEaS [online]. 2010 [cit. 2020-07-30]. Dostupné z: <http://studijni-materialy.sseas.cz/bezpecnostni-systemy/zakladni-funkce-a-rozdeleni-ustreden-ezs/>
- [5] HALOUZKA, Kamil. Perimetrické zabezpečovací systémy [online]. [cit. 2020-07-30]. Dostupné z: https://moodle.unob.cz/pluginfile.php/18075/mod_resource/content/2/10_Perimetrick%C3%A9%20zabezpe%C4%8Dovac%C3%AD%20syst%C3%A9my.pdf
- [6] HALOUZKA, Kamil. Elektrická zabezpečovací signalizace, vstupní systémy, biometrická kontrola vstupu [online]. [cit. 2020-07-30]. Dostupné z: <https://docplayer.cz/7632990-Tema-elektricka-zabezpecovaci-signalizace-vstupni-systemy-biometricka-kontrola-vstupu-ing-kamil-halouzka-ph-d-kamil-halouzka-unob.html>
- [7] MIKROVLNY - BEZPEČNÁ TECHNOLOGIE: Spektrum záření. In: MYKOS MIKROVLNY [online]. [cit. 2020-07-30]. Dostupné z: <http://www.mykos.cz/bezpecnost-mikrovlny/>
- [8] ŠIMÍČEK, Jan. Detekce pohybu objektu PIR senzory v polární konfiguraci. Praha, 2015. Bakalářská práce. ČVUT v Praze, Fakulta elektrotechnická. Vedoucí práce Ing. Jan Zábojník.
- [9] LUKÁŠ, L.: Bezpečnostní technologie, systémy a management V.: Teorie a praxe ochrany majetku a fyzické bezpečnosti. Zlín: VerBum, 2015, 368 s. ISBN 978-80-87500-67-4
- [10] PINKER, Jiří. Mikroprocesory a mikropočítače. Praha: BEN - technická literatura, 2004, 159 s. ISBN 8073001101.

- [11] PALACKÝ, Petr. MIKROPOČÍTAČOVÉ ŘÍDICÍ SYSTÉMY I. Ediční středisko VŠB – TUO, 2007, 123 s. ISBN ISBN 978-80-248-1494-0.
- [12] CO JE TO ARDUINO? Arduino.cz [online]. [cit. 2020-07-30]. Dostupné z: <https://arduino.cz/co-je-to-arduino/>
- [13] PROGRAMINO IDE for Arduino. PROGRAMINO [online]. [cit. 2020-07-30]. Dostupné z: <https://programino.com/>
- [14] Arduino IDE for Visual Studio. In: VisualMicro [online]. [cit. 2020-07-30]. Dostupné z: <https://www.visualmicro.com/page/Arduino-development-plugin-for-Visual-Studio.aspx>
- [15] ARDUINO UNO REV3. In: Arduino [online]. [cit. 2020-07-30]. Dostupné z: <https://store.arduino.cc/arduino-uno-rev3>
- [16] Arduino UNO R3, ATmega328P, Klon. LASKARDUINO.CZ [online]. [cit. 2020-07-30]. Dostupné z: <https://www.laskarduino.cz/arduino-uno-r3--atmega328p--klon/>
- [17] 16x2 LCD displej 1602 modrý + I2C převodník. LASKARDUINO.CZ [online]. [cit. 2020-07-30]. Dostupné z: <https://www.laskarduino.cz/16x2-lcd-displej-1602-i2c-prevodnik/>
- [18] PATTABIRAMAN, Krishna. HOW TO SET UP A KEYPAD ON AN ARDUINO. Circuit Basics [online]. [cit. 2020-07-30]. Dostupné z: <https://www.circuitbasics.com/how-to-set-up-a-keypad-on-an-arduino/>
- [19] Arduino 4x3 Maticová membránová klávesnice. LASKARDUINO.CZ [online]. [cit. 2020-07-30]. Dostupné z: <https://www.laskarduino.cz/arduino-4x3-maticova-membranova-klavesnice/>
- [20] Arduino Mini PIR detektor pohybu HC-SR505. LASKARDUINO.CZ [online]. [cit. 2020-07-30]. Dostupné z: <https://www.laskarduino.cz/arduino-mini-pir-detektor-pohybu-hc-sr505/>
- [21] Arduino IR optický infra snímač FC-51. LASKARDUINO.CZ [online]. [cit. 2020-07-30]. Dostupné z: <https://www.laskarduino.cz/arduino-ir-opticky-infra-snimac-fc-51/>
- [22] SM-35 - bílá. VARNET [online]. [cit. 2020-07-30]. Dostupné z: <https://www.varnet.cz/zbozi/0701-050-sm-35-bila>
- [23] Aktivní bzučák 5V. LASKARDUINO.CZ [online]. [cit. 2020-07-30]. Dostupné z: <https://www.laskarduino.cz/aktivni-bzucak-5v/>

[24] LiquidCrystal_I2C.h. GitHub [online]. [cit. 2020-07-30]. Dostupné z: https://github.com/andresWeitzel/LiquidCrystal_I2C.h

[25] Keylock. GitHub [online]. [cit. 2020-07-30]. Dostupné z: <https://github.com/Zprite/Keylock>

[26] Keypad Tutorial. Arduino Playground [online]. [cit. 2020-07-31]. Dostupné z: <https://playground.arduino.cc/Main/KeypadTutorial/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

PLC	Programmable Logic Controller
PZS	Poplachový zabezpečovací systém
RFID	Radio Frequency Identification
NC	Normally Closed
EOL	End Of Line
ATZ	Advanced Technology Zoning
ADR	Address
CRC	Cyclic Redundancy Check
PGM	Programmable Output
LCD	Liquid Crystal Display
USB	Universal Serial Bus
RS	Recommended Standard
Hz	Hertz
VKV	Velmi krátké vlny
f	frekvence
v	rychlost pohybu narušitele
m	metr
s	sekunda
c	rychlost ultrazvuku v prostředí
V	volt
A	ampér
PIR	Passive Infra Red
R	elektrický odpor
Ω	ohm
l	délka vodiče

S	příčný průřez vodiče
kB	kiloByte
LED	Light-Emitting Diode
ALU	Arithmetic Logic Unit
CISC	Complex Instruction Set Computer
RISC	Reduced Instruction Set Computer
IDE	Integrated Development Environment
HTML	Hypertext Markup Language
PCB	Printed Circuit Board
EEPROM	Electrically Erasable Programmable Read-Only Memory
SRAM	Static Random Access Memory
USART	Universal Synchronous / Asynchronous Receiver and Transmitter
SPI	Serial Peripheral Interface
PWM	Pulse Width Modulation
TFT	Thin-Film Transistors
OLED	Organic Light-Emitting Diode
I ² C	Inter-Integrated Circuit
SDA	Serail Data
SDC	Serail Clock
DC	Direct Current
IR	Infra Red

SEZNAM OBRÁZKŮ

Obrázek 1 Blokové schéma PZS	11
Obrázek 2 Datový rámec	14
Obrázek 3 Magnetický kontakt se dvěma jazýčky	18
Obrázek 4 Elektromagnetické spektrum, upraveno	19
Obrázek 5 Princip PIR detektoru	20
Obrázek 6 Architektura von Neumann (vlevo), Harvardská architektura (vpravo)	23
Obrázek 7 Vývojové prostředí Arduino IDE	26
Obrázek 8 Vývojové prostředí Programino	27
Obrázek 9 Vývojové prostředí Microsoft Visual Studio	27
Obrázek 10 Software Fritzing	28
Obrázek 11 Schéma pinů Arduina Una R3, upraveno	30
Obrázek 12 Princip maticové klávesnice, upraveno	33
Obrázek 13 Výukový model PZS	36
Obrázek 14 Grafické schéma zapojení	37
Obrázek 15 Inicializace knihoven.....	38
Obrázek 16 Deklarace pomocí define.....	39
Obrázek 17 Deklarace pomocí datového typu.....	39
Obrázek 18 Deklarace a inicializace klávesnice a displeje.....	40
Obrázek 19 Funkce setup.....	41
Obrázek 20 Vývojový diagram funkce Smyčka	43
Obrázek 21 Funkce Smyčka, první část.....	44
Obrázek 22 Funkce Smyčka, druhá část	45
Obrázek 23 Vývojový diagram funkce Odpocet	46
Obrázek 24 Funkce Odpocet.....	46
Obrázek 25 Vývojový diagram funkce Detektory	48
Obrázek 26 Funkce Detektory, první část	49
Obrázek 27 Funkce Detektory, druhá část.....	50

SEZNAM TABULEK

Tabulka 1 Technické parametry – procesor ATmega 328 a Arduino Uno R3	31
Tabulka 2 Technické parametry – 16x2 LCD displej 1602 modrý + I ² C převodník	32
Tabulka 3 Technické parametry – 4x3 Maticová membránová klávesnice	33
Tabulka 4 Technické parametry – PIR detektor HC-SR505	34
Tabulka 5 Technické parametry – IR snímač FC-51	34
Tabulka 6 Technické parametry – Magnetický kontakt SM-35	35
Tabulka 7 Technické parametry – Aktivní bzučák 5V	35

SEZNAM PŘÍLOH

Příloha P I: CD ROM