

Nasazení mikrokontrolérů typu Arduino pro oblast bezpečnosti

Petr Trávník

Bakalářská práce
2023



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav bezpečnostního inženýrství

Akademický rok: 2022/2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Petr Trávník**
Osobní číslo: **A18197**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Bezpečnostní technologie, systémy a management**
Forma studia: **Kombinovaná**
Téma práce: **Nasazení mikrokontrolérů typu Arduino pro oblast bezpečnosti**
Téma práce anglicky: **Deployment of Arduino-type microcontrollers in the field of security**

Zásady pro vypracování

1. Seznamte se s platformou Arduino a jejími možnostmi.
2. Popište základní bezpečnostní technologie.
3. Proveďte průzkum trhu s cílem identifikovat možnosti využití Arduino v bezpečnostních aplikacích.
4. Navrhněte možné využití Arduino v bezpečnosti.
5. Demonstrujte využití na modelovém příkladu.

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

Seznam doporučené literatury:

1. VODA, Zbyšek. Průvodce světem Arduina. Bučovice: Martin Stříž, 2015. ISBN 978-80-87106-90-7.
2. SELECKÝ, Matúš. Arduino: uživatelská příručka. Přeložil Martin HERODEK. Brno: Computer Press, 2016. ISBN 978-80-251-4840-2.
3. BURDA, Karel. Základy elektronických zabezpečovacích systémů. Brno: Akademické nakladatelství CERM, 2017. ISBN 9788072049677.
4. LUKÁŠ, Luděk a kolektiv. Bezpečnostní technologie, systémy a management I. 1. Zlín: VerBuM, 2011. ISBN 978-80-87500-05-7.
5. LUKÁŠ, Luděk a kolektiv. Bezpečnostní technologie, systémy a management II. 1. Zlín: VerBuM, 2012. ISBN 978-80-87500-19-4.
6. LUKÁŠ, Luděk a kolektiv. Bezpečnostní technologie, systémy a management III. 1. Zlín: VerBuM, 2013. ISBN 978-80-87500-35-4.
7. VALOUCH, Jan. Projektování bezpečnostních systémů. Vyd. 1. Ve Zlíně: Univerzita Tomáše Bati ve Zlíně, 2012, 152 s. ISBN 978-80-7454-230-5. Dostupné také z: <http://dspace.k.utb.cz/handle/10563/18663>

Vedoucí bakalářské práce: **Ing. Lukáš Králík, Ph.D.**
Ústav bezpečnostního inženýrství

Datum zadání bakalářské práce: **16. prosince 2022**

Termín odevzdání bakalářské práce: **5. června 2023**



doc. Ing. Jiří Vojtěšek, Ph.D. v.r.
děkan

Ing. Jan Valouch, Ph.D. v.r.
ředitel ústavu

Ve Zlíně dne 16. prosince 2022

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;
- 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 – 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 2. 6. 2023

.....
Petr Trávník, v. r.

ABSTRAKT

Hlavním účelem této bakalářské práce je zhodnotit možnosti nasazení mikrokontrolérů typu Arduino v bezpečnosti. Výsledným produktem je pak vytvoření funkčního modelu bezpečnostního zařízení pracujícího na platformě Arduino. Práce je rozdělena na teoretickou a praktickou část. V teoretické části je představeno Arduino. Součástí práce je i stručná teorie popisující druhy bezpečnostních technologií a systémů. V praktické části jsou navržena reálná využití Arduina v bezpečnosti a zkonstruován ukázkový model.

Klíčová slova: mikrokontrolér, Arduino, systém kontroly vstupu, detektory, senzory, programování

ABSTRACT

The main purpose of this Bachelor thesis is to assess options of Arduino microcontrollers' use in safety. Creating a safety device's functional prototype (model) working on Arduino's platform is going to be the final outcome. The thesis is divided into a theoretical and practical part. Arduino is presented in the theoretical part. A brief theory describing types of the security technologies and systems is also the part of this thesis. Arduino's real use in safety is designed in the practical part and a sample model is constructed here.

Keywords: microcontroller, Arduino, access control system, detectors, sensors, programming

Rád bych poděkoval své rodině. Jak manželce za trpělivost, obětavost a nedocenitelnou podporu při vytváření práce, tak synovi, kterému během toho času „chyběl“ otec.

Velké díky patří samozřejmě také všem, kteří se jakkoliv na práci podíleli. Ať už se jednalo o připomínky k jazykové a slohové úpravě, tak i těm co se účastnili testování vytvořeného modelu.

V neposlední řadě bych rád poděkoval vedoucímu práce panu Ing. Lukáši Králíkovi, Ph.D., za jeho obětavost, připomínky, nápady a celkový přínos k práci.

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

Motto:

„Jediná věc, která stojí mezi vámi a vaším cílem jsou ty kecy o tom, jak to nezvládnete, které si neustále namlouváte.“

Jordan Belfort

OBSAH

ÚVOD	8
I. TEORETICKÁ ČÁST	9
1 ARDUINO	10
1.1 HARDWARE.....	10
1.1.1 Arduino desky v oficiální distribuci.....	10
1.1.2 Klony.....	12
1.2 PERIFERIE.....	14
1.2.1 Shieldy.....	14
1.2.2 Snímače.....	15
1.3 SOFTWARE.....	16
1.3.1 Arduino IDE.....	16
1.3.2 Tinkercad.....	17
1.3.3 Fritzing.....	17
1.4 SHRNUÍ.....	19
2 BEZPEČNOSTNÍ TECHNOLOGIE A SYSTÉMY	20
2.1 POPLACHOVÝ ZABEZPEČOVACÍ A TÍSŇOVÝ SYSTÉM.....	20
2.1.1 Plášťová ochrana.....	20
2.1.2 Prostorová ochrana.....	21
2.1.3 Předmětová ochrana.....	21
2.1.4 Perimetrická ochrana.....	21
2.1.5 Tísňové systémy.....	22
2.1.6 Shrnutí.....	22
2.2 ELEKTRICKÁ POŽÁRNÍ SIGNALIZACE.....	22
2.3 ELEKTRONICKÉ SYSTÉMY KONTROLY VSTUPU.....	23
2.3.1 Čipové identifikační prvky.....	23
2.3.2 Magnetické karty.....	24
2.3.3 Optické identifikační prvky.....	25
2.3.4 Biometrika.....	25
2.4 SHRNUÍ.....	25
3 ARDUINO V BEZPEČNOSTI	26
3.1 SENZORY.....	26
3.1.1 Teplota, vlhkost a tlak.....	26
3.1.2 Plyn a kvalita ovzduší.....	27
3.1.3 Vzdálenost, pohyb a vibrace.....	28
3.1.4 Světlo, barva a optika.....	30
3.1.5 Proud a napětí.....	31
3.1.6 Hmotnost.....	31
3.1.7 Biometrika.....	32
3.2 SIGNALIZACE.....	33
3.2.1 Optická.....	33
3.2.2 Akustická.....	33
3.2.3 Přenosem.....	34

3.3	DOPLŇKY	35
3.4	SHRnutí.....	36
II. PRAKTICKÁ ČÁST.....		37
4	NÁVRHY PRAKTICKÉHO VYUŽITÍ ARDUINA PRO OBLAST BEZPEČNOSTI.....	38
4.1	MĚŘENÍ PROUDOVÉHO ODBĚRU.....	38
4.2	OTŘESOVÉ ČIDLO	40
4.3	SENZOR PLAMENE A SENZOR HOŘLAVÝCH PLYNŮ	41
4.4	DETEKTOR POHYBU PIR.....	44
4.5	SYSTÉM KONTROLY VSTUPU.....	45
4.5.1	Displej	45
4.5.2	Klávesnice a systém kontroly vstupu pomocí hesla.....	46
4.5.3	RFID.....	47
4.6	SHRnutí.....	49
5	PROJEKT „KRABICE“	50
5.1	VÝROBA UKÁZKOVÉHO MODELU.....	50
5.2	SCHÉMA ZAPOJENÍ.....	52
5.3	ZDROJOVÝ KÓD UKÁZKOVÉHO MODELU.....	53
5.4	NÁVOD.....	57
5.5	SPOTŘEBA	59
5.6	SHRnutí.....	61
ZÁVĚR		62
SEZNAM POUŽITÉ LITERATURY.....		64
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK		68
SEZNAM OBRÁZKŮ		70
SEZNAM TABULEK.....		72
SEZNAM PŘÍLOH.....		73

ÚVOD

Bezpečnost je důležitým aspektem našeho každodenního života, ať už jde o ochranu našich domovů, pracovišť nebo veřejných prostor. V dnešní době moderních technologií existuje nespočet variant jak zlepšit bezpečnost prostředí pomocí různých systémů. Jednou z možností může být platforma Arduino.

Díky internetu je možné sdílet informace a poznatky s celým světem a realizovat tak projekty podle vlastních přání. Vytvořené prototypy je možné pak využít v robotice, chytré domácnosti a v neposlední řadě v oblasti bezpečnosti.

Arduino v oblasti bezpečnosti je velmi širokým tématem a může být pojato různě. V teoretické části tedy bude popsáno, co je vlastně to Arduino, jaké jsou možnosti pořízení základního hardwaru, potřebných periférií a v neposlední řadě softwaru, který pro práci s Arduinem lze využít. Je také důležité si vysvětlit, co je myšleno touto oblastí bezpečnosti, jak je možné ji rozdělit a zmínit některé základní definice v tomto odvětví. Při průzkumu trhu pak do tohoto rozdělení začlenit jednotlivé prvky a zjistit pořizovací náklady.

Člověk může přecítit nespočet teorie a přesto danou problematiku nepochopit do hloubky, je proto vhodné si teorii převést do praxe. Z tohoto důvodu budou v praktické části bakalářské práce vybrány některé senzory, ukázány jejich možnosti zapojení, použití a naprogramování. Okomentované zdrojové kódy jednotlivých aplikací budou umístěny v přílohách na konci práce. Pro účely prezentace bude zkonstruován komplexní model obsahující dílčí ukázky. Zmíněno bude pár zajímavých řešení použitých při programování výsledného projektu a vytvořen jednoduchý návod k použití.

Cílem této bakalářské práce je poskytnout čtenářům ucelený přehled o možnostech využití Arduina v oblasti bezpečnosti a inspirovat je k vytváření vlastních projektů a řešení.

I. TEORETICKÁ ČÁST

1 ARDUINO

Arduino je otevřená hardwarová a softwarová platforma založená na mikrokontrolérech, která byla vyvíjena od roku 2005 italskými výzkumníky. Cílem jejich práce bylo vytvořit snadný nástroj pro rychlou tvorbu prototypů elektronických zařízení. [1] [2]

Arduino oproti jiným mikrokontrolérům vyniká jednoduchým integrovaným vývojovým prostředím založeném na jazyku Processing, které je možné s jednoduchými editory provozovat na běžných operačních systémech, jako například Microsoft Windows, Linux a Mac OS. Nahrávání zjednodušuje bootloader a samotná komunikace s PC pak probíhá přes převodníky sériovou linkou USB, kterou lze monitorovat. [1] [2]

Díky své jednoduchosti a nízkým pořizovacím nákladům byly dalším předpokladem vedoucímu k masivnímu rozšíření Arduina licenční podmínky a to Open-source, což znamená otevřené všem. Vzhledem k tomu, že se jedná hlavně o hardwarové řešení, týkají se Arduina projekty Open-source hardware a Open-source software. U software jde o sdílení zdrojových kódů, vývojového prostředí a u hardwarového zase o poskytnutí podkladů, schémat zapojení sloužící k možnosti výroby vlastního hardware. Lidé tedy mohli tato softwarová a hardwarová Open-source data studovat, vylepšovat a dále zdokonalovat. Došlo tak k vytvoření obsáhlých komunit. [1] [2]

1.1 Hardware

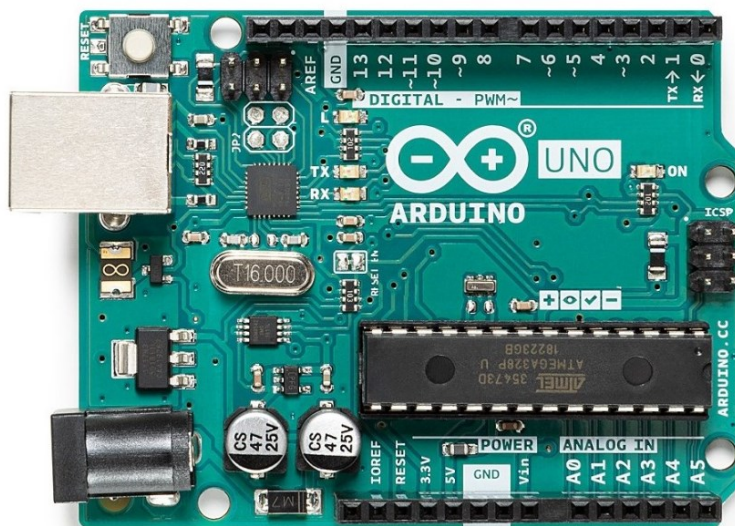
Základem platformy je mikrokontrolér, tedy integrovaný obvod obsahující procesní jednotku, paměti a periferní rozhraní. Většina originálních zařízení nese procesory od výrobce Atmel z rodiny AVR, tedy 8 bitové mikročipy s harvardskou architekturou a redukovanou instruktážní sadou (Reduced Instruction Set Computer - RISC), která vychází z předpokladu, že 80 % operací vyžaduje asi jen 20 % instrukcí. Mikroprocesor je tedy vybaven jen základními jednoduchými a snáze proveditelnými mikroinstrukcemi. Devízou harvardské architektury je to, že paměť programu a instrukcí jsou fyzicky zvlášť odděleny a nesdílí tak adresní a datové sběrnice. [1] [3] [4] [5]

1.1.1 Arduino desky v oficiální distribuci

Arduino desky se vyrábějí v různých provedeních, jsou rozlišeny velikostí, výkonem, analogovými a digitálními vstupy/výstupy, funkcemi a další. Portfolio nabízených mikrokontrolérů od Arduina je v dnešní době již poměrně obsáhlé. Budou zde zmíněny a zdůrazněny

hlavní vlastnosti několika mikrokontrolérů v nabídce. Mezi nejznámější a základní patří Arduino Uno, které bude přibliženo více, pak Arduino Mega, Arduino Nano a Arduino Mini. [1] [6]

Arduino Uno (obr. 1) je nejzákladnější a nejrozšířenější deska od Arduina, v dnešní době dostupná ve třetí revizi. Je možné ji pořídit mimo základní možnost připojení skrze USB, také s možností připojení přes Wi-Fi, Bluetooth, ethernet nebo pro konkrétní aplikace bez programovacího převodníku. Arduino Uno je osazeno zmíněným procesorem firmy Atmel, konkrétně ATmega328P o taktovací frekvenci 16 MHz. Deska má 14 digitálních vstupů/výstupů a z toho 6 s pulsně šířkovou modulací (Pulse Width Modulation – PWM). Dále 6 analogových vstupů, zejména pro připojení různých senzorů. Množství, druh pinů a jejich rozmístění je také patrné z obrázku (obr. 1) základní desky Arduino Uno. [1] [6] [7]



Obrázek 1 Deska Arduino Uno [7]

Technické specifikace Arduino Uno:

- | | |
|----------------------------|------------|
| - Mikrokontrolér | ATmega328P |
| - Taktovací frekvence | 16 MHz |
| - Operační napětí | 5 V |
| - Doporučené napájení | 7-12 V |
| - Limitní napájení | 6-20 V |
| - Digitální vstupy/výstupy | 14 |
| - PWM výstupy | 6 |

- Analogové vstupy	6
- Paměť Flash	32 kB z toho 0,5 kB pro bootloader
- EEPROM	1 kB
- SRAM	2 kB
- Rozměry	68,6 / 53,4 mm
- Váha	25 g
- Cena	718,- Kč [7]

Arduino Mega 2560 – Obsahuje výkonnější čipy, 54 digitálních vstupů/výstupů a 16 analogových. Rozměry desky jsou tedy větší. Tento model se hodí již zkušeným uživatelům, kteří potřebují pracovat s více vstupy/výstupy a větším výkonem, nebo pro konkrétní projekty tyto vlastnosti vyžadující. Pořizovací cena je 1098,- Kč. [1] [6] [8]

Arduino Micro je pojmenováno dle typu konektoru pro připojení USB Micro na rozdíl od standardního USB B u předchozích modelů. Jako jedno z mála je osazeno čipem ATmega32U4 obsahující převodník. Díky tomu je možné posílat příkazy přímo do PC, pro který se může tvářit jako myš, klávesnice a podobně. Toto je možné i s deskami bez čipu s integrovaným převodníkem, ale je vyžadováno přeprogramování převodníku, což nemusí být zrovna nejjednodušší. Oproti verzi Uno má 20 digitálních a 12 analogových vstupů/výstupů a je možné jej pořídit v přepočtu za 507,- Kč [1] [6] [8]

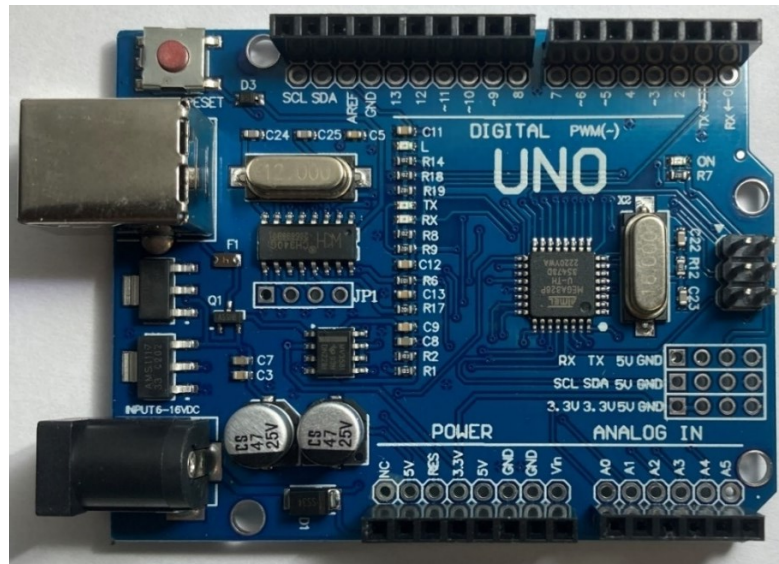
Arduino Nano je téměř nejmenší verze Arduina. Menší je verze Mini, která na rozdíl od Nano neobsahuje USB port pro připojení, a je proto nutné použití externího USB-Serial převodníku. Nano však výkonem nezaostává za většími, a to díky procesoru ATmega328. Ani počtem vstupů a výstupů se od předchozích moc neodlišuje, konkrétně má 22 datových a 8 analogových vstupů/výstupů. Svými rozměry nachází využití například v různých chytrých vypínačích, dálkových ovladačích a místech, kde je vyžadována miniaturizace. Má i sympatickou cenu 368,- Kč. [1] [6] [8]

1.1.2 Klony

Jak již bylo zmíněno, díky Open-source licenci vzniká nepřeberné množství mikrokontrolérů, které jsou navrženy, aby byly kompatibilní s originálními Arduino deskami. Jsou vyráběny jinými společnostmi nebo jednotlivci. Klony nabízejí často stejnou funkčnost za nižší ceny, ale mnohdy na úkor kvality a podpory u výrobce. Nejvíce těchto klonů vzniká asi

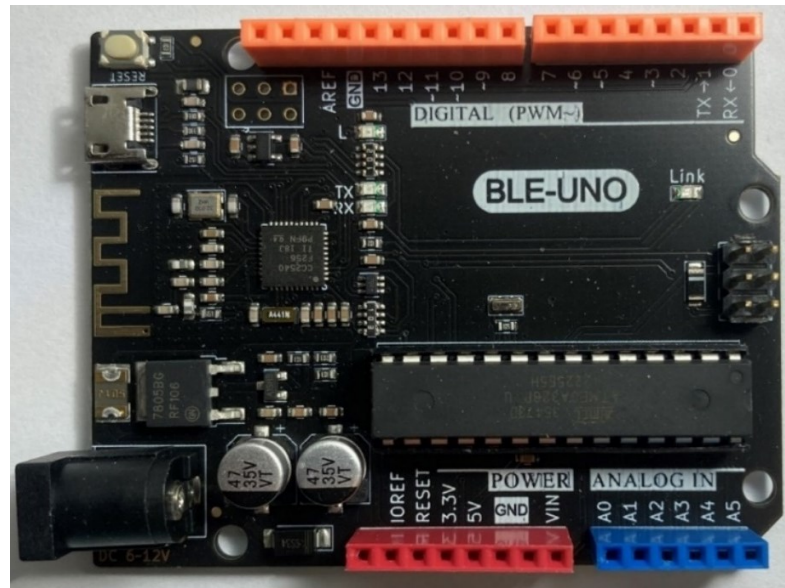
v Číně, ale i různé komunity si vytvářejí vlastní. Příkladem může být česká společnost Las-kaKit. [1] [8]

Klony svým tvarem a provedením mohou být přizpůsobeny konkrétním aplikacím, nebo mít vlastní originální design výrobce. Mnohdy ale mají barvu, rozměry a rozložení připomínající originální Arduino. Dva takovéto modely jsou použity dále pro potřeby této bakalářské práce. Jedním je blíže neidentifikovatelný klon UNO pořízený v Číně přes platformu www.aliexpress.com, kde se jeho cena v přepočtu pohybuje v rozmezí 80,- až 300,- Kč. Je osazen integrovaným procesorem ATmega328P U-TH (obr. 2). Vzhledem k tomu že se jedná o klon Arduino Uno Rev3, měli by parametry být totožné. Pozorováním lze s jistotou pouze konstatovat, že rozměry, počty vstupů a výstupů a způsob napájení opravdu totožné jsou.



Obrázek 2 Klon UNO

Dalším použitým klonem je BLE-UNO od výrobce Emakefun (obr. 3), který má již čip zasazený v patici, obdobně jako originální Arduino Uno Rev3. Je možné jej tedy použít pro vlastní modifikace. BLE-UNO disponuje navíc integrovaným čipem TI CC2540 pro připojení skrze Bluetooth a kabelem se připojuje skrze mnohem běžnější koncovku USB Micro. Jeho cena se v přepočtu na e-shopu www.aliexpress.com pohybuje okolo 300,- Kč. [9]



Obrázek 3 Klon BLE-UNO

Ať už má uživatel představu, na jakém projektu chce pracovat, nebo se do problematiky teprve chystá proniknout, může se díky dostupným informacím zorientovat. Existuje také možnost, nechat si poradit na obsáhlých fórech. Všechny, nejenom tyto desky, mají své specifické vlastnosti a jsou vhodné pro různé projekty. Také je možnost použít nějaký softwarový simulátor o kterých bude ještě zmínka. Pro začátečníky, seznámení s Arduinem a nejuніверzálnější použití má Arduino Uno. Z těchto důvodů je pro potřeby bakalářské práce zvolen klon této verze.

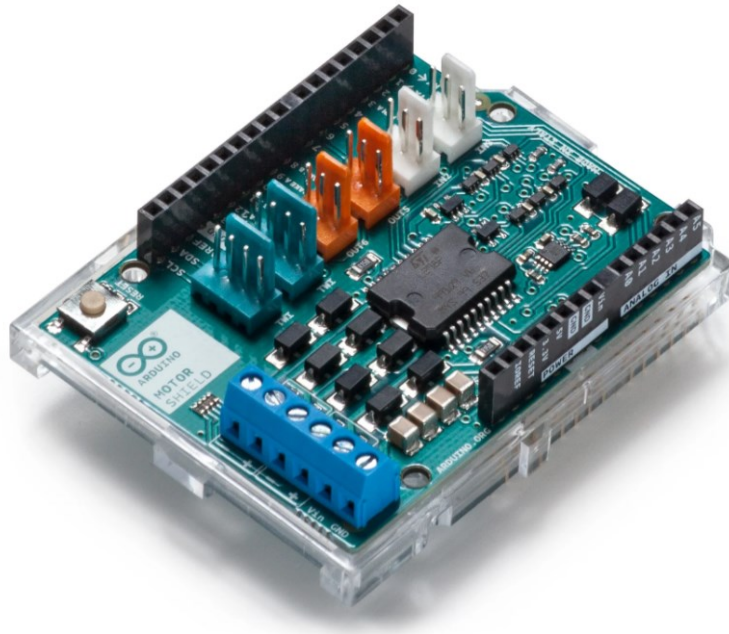
1.2 Periferie

Samotné Arduino toho v podstatě moc neumí a lze jej připojením různých periferních modulů, takzvaných Shieldů, rozšířit o další funkce. A pro samotné praktické použití pak na vstupy a výstupy připojit různé senzory, čidla, klávesnice, zobrazovací jednotky a spoustu dalšího. [2]

1.2.1 Shildy

Shildy jsou rozšiřující moduly nebo desky, které se připojují přímo na Arduino desku a rozšiřují její funkčnost. Shildy jsou snadno integrovatelné s Arduino deskami a využívají standardní piny pro připojení, což umožňuje vytváření složitějších projektů bez nutnosti pájení. Tyto moduly mají zpravidla stejnou velikost jako desky, pro které jsou určeny. Nasazují se „na ni“ a přidávají požadovanou funkci a dále zachovávají původní rozmístění pinu. Nebo

jen umožňují připojit různá zařízení pomocí konektorů a rozšiřují na každý konektor zemnicí a napájecí pin. Na níže uvedeném modelu (obr. 4), který zjednodušuje práci s motory, to je patrné. [1] [2] [10]



Obrázek 4 Arduino Motor Shield Rev3 [10]

Dále může jít o Shildy zabezpečující konektivitu, tedy třeba možnost připojení k Wi-Fi síti, Bluetooth, GSM/GPRS a ethernet, přes které je možno s Arduinem obousměrně komunikovat a otevírají se tak nové možnosti využití, například i ovládání přes mobilní telefon na dálku. Pokud je zapotřebí ukládat nějaká data, je možnost pořízení Shieldu, který desku rozšíří o slot na paměťovou kartu. [1] [2] [6] [8]

Bylo uvedeno pouze pár příkladů z nepřeberného množství Shieldů. Stejně jako samotná deska i Shildy nejsou výhradou oficiálního distributora, ale vznikají v jiných společnostech a samozřejmě také v Číně.

1.2.2 Snímače

Snímače neboli také senzory pro Arduino jsou další důležitou součástí mnoha projektů, které využívají Arduino desky. Snímače jsou jakýmsi propojením reálného světa s elektronickými zařízeními. Slouží k měření různých fyzikálních veličin, jako je teplota, vlhkost, světlo, zvuk, pohyb nebo tlak, které transformují na elektrický signál. Tyto snímače mohou být připojeny pomocí drátků přímo k Arduino desce nebo prostřednictvím Shieldu, kde může být

těchto senzorů více, například pro měření vlhkosti a teploty (obr. 5). K jednotlivým fyzikálním veličinám je většinou možnost využít různé snímače a je zapotřebí před jejich pořízením prostudovat katalogové listy a vybrat ten nejvhodnější.



Obrázek 5 Snímač vlhkosti a teploty

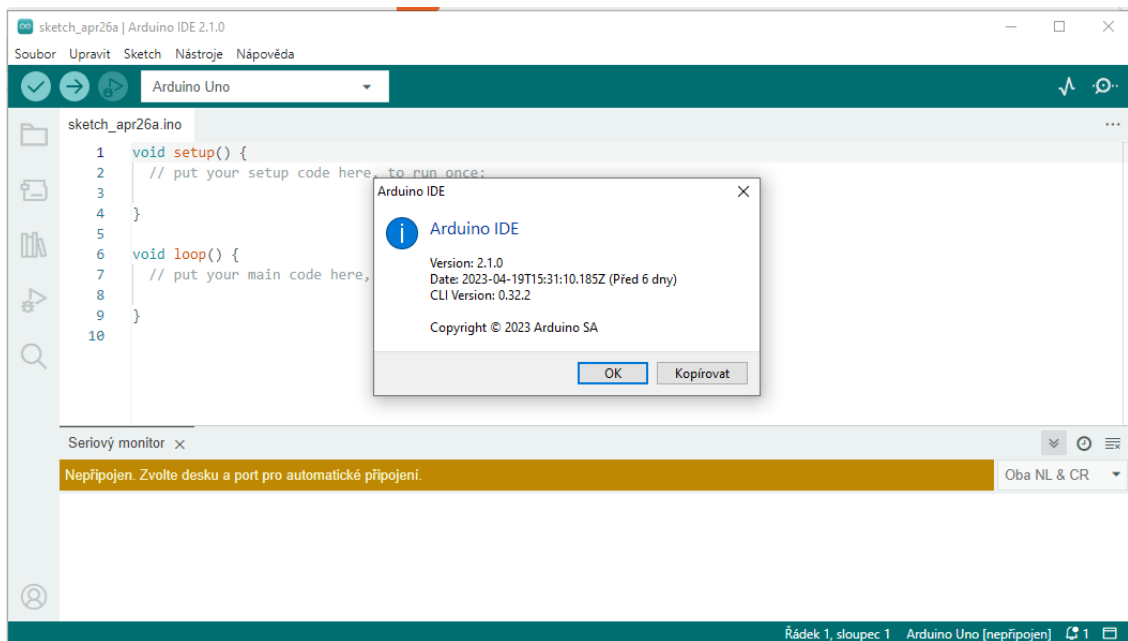
Při výběru Shieldů a snímačů pro projekt je zapotřebí kontrolovat kompatibilitu s Arduino deskou, zda jsou dostupné knihovny a ovladače. U kvalitnějších produktů jsou u výrobců tyto informace a materiály volně dostupné, včetně příkladů použití, návodů a lze se rozhodnout, zda budou vhodné pro daný projekt.

1.3 Software

Samotné Arduino je jen deska plná součástek a sama o sobě nic neumí. Pro jeho uvedení do provozu je zapotřebí jej naprogramovat, k čemuž slouží integrovaná vývojová prostředí (v anglickém znění Integrated Development Environment – IDE). Lze použít téměř jakékoliv IDE podporující jazyk Wiring k programování hardwarových prvků. Úkolem IDE je zejména zefektivnit tvorbu kódu pomocí nápověd, automatického formátování kódu, jeho barevné zvýrazňování a v neposlední řadě kontrola chyb programu. Mezi nejznámější IDE můžeme zařadit Arduino IDE, Visual Studio Community, Atmel Studio, CodeBender, Mixly, MagicBlock a spoustu dalších. [1] [2] [9]

1.3.1 Arduino IDE

Nejrozšířenějším integrovaným vývojovým prostředím pro platformu Arduino je přímo od výrobce Arduino Software, aktuálně ve verzi Arduino IDE 2.1.0 (obr. 6). Na stránkách k projektu www.arduino.cc je dostupné pro platformy Windows, Mac OS X a Linux. IDE vzniklo z výukového prostředí Processing v jazyce Java a stejně jako celý projekt je Open-source, takže zdrojové kódy můžeme najít přímo na GitHub. [1] [2] [4]



Obrázek 6 Software Arduino IDE 2.1.0

Arduino Software má také cloudovou verzi vývojového prostředí Arduino Editor, dostupnou přímo ze stránek projektu. Velkou výhodou je integrace všech možných knihoven v jejich nejaktuálnější verzi a díky cloudovému řešení, možnost pracovat odkudkoliv. Pokud není k dispozici internet, je zapotřebí využít předchozí off-line verzi Arduino IDE, v opačném případě Arduino Editor plně nahrazuje instalovanou verzi. [4]

1.3.2 Tinkercad

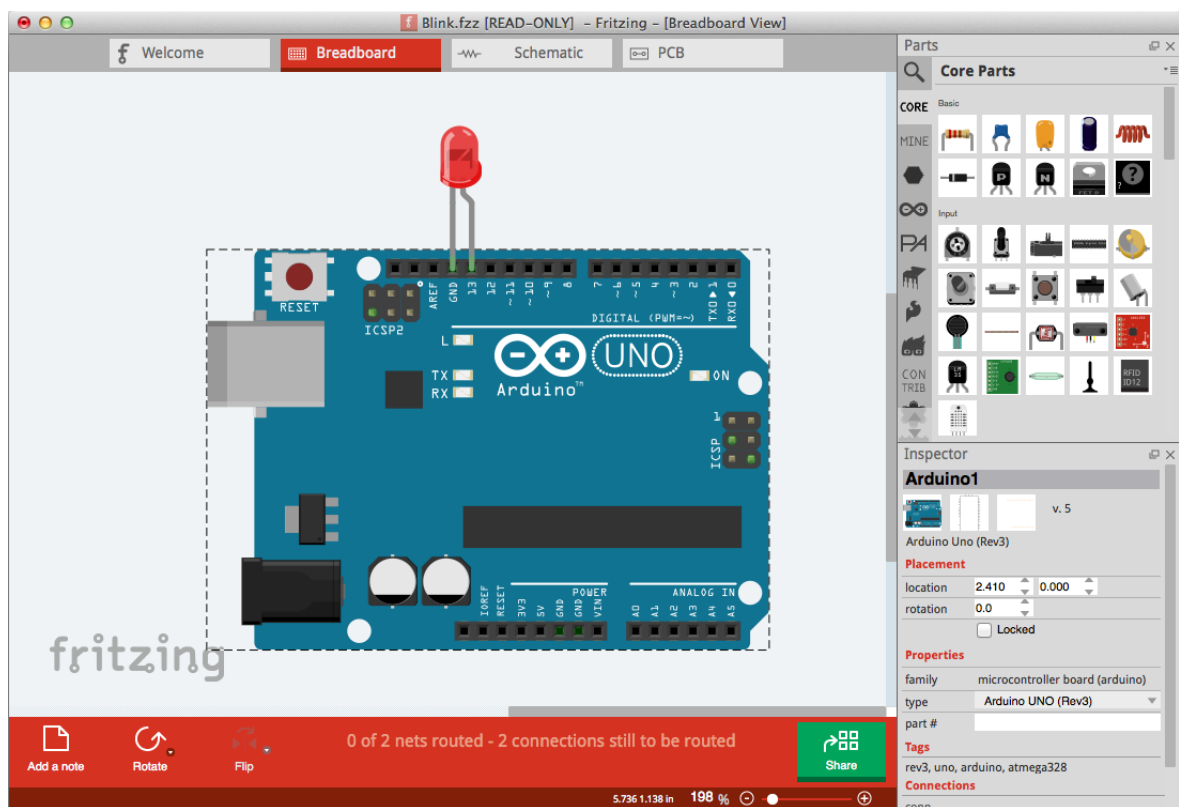
Online řešení, které umožňuje vytváření simulačních projektů na platformě Arduino, je software Tinkercad od společnosti Autodesk. Po registraci je možné procházet vytvořené projekty a dále je upravovat, nebo si vytvořit projekt vlastní. Propojení, vytváření obvodu je poměrně snadné a intuitivní. Po kliknutí na tlačítko „kód“, je možno Arduino naprogramovat a spustit simulaci. [11]

Velkou nevýhodou je poměrně strohá databáze modulů, čidel a senzorů. Využití nachází spíše pro studijní účely, pro vyzkoušení si některých programových kódů či projektových řešení, před jejich fyzickým pořízením.

1.3.3 Fritzing

Mocným nástrojem je pak software Fritzing (obr. 7). Oproti například Tinkercadu obsahuje v knihovně velké množství senzorů, desek, Shieldů a jiných součástek. Dostupné jsou řešení

oficiální od velkých zahraničních výrobců a největších čínských společností. Jedná se tedy o Arduino, Intel, SparkFun, Seeed Studio, Snootlab, Dagu, Atlas Scientific a další. Pokud by software náhodou neobsahoval námi požadovanou součástku, je velmi jednoduché si ji nainportovat, protože většina Arduino tvůrců Fritzing využívá. Mimo grafický návrh zapojení je možné k programu také připojit fyzické zařízení a obdobně jako v Arduino IDE jej programovat, včetně integrace sériového monitoru. Výhodou je pak možnost na základě grafického návrhu vytvořit schematické zapojení, nebo si vytvořit PCB desku (Printed Circuit Board – deska tištěných spojů). [12]



Obrázek 7 Software Fritzing [12]

Na první pohled by se mohlo zdát, pokud se pokusíte stáhnout produkt z oficiálních stránek, že jde o placený software, ale opak je pravdou. Jedná se o Open-source projekt umístěný na GitHubu a hlavními představiteli jsou němečtí autoři. Zkompilovanou verzi je možno tedy stáhnout z oficiálních stránek, zaplatit a podpořit tak autory v dalším vývoji. Aktualizace opravdu stále probíhá a například od května roku 2022 je možné využít program jako simulátor, obdobně jako Tinkercad. Pokud si nepřejeme podpořit autory, je možné si z GitHubu stáhnout zdrojové kódy, zkompilovat a produkt využívat zdarma. [12] [13]

1.4 Shrnutí

Existuje mnoho softwarových řešení týkajících se Arduina, mimo výše uvedených které byly využívány při práci. Je možné narazit například na simulátory, nebo vývojové prostředí jako jsou Mixly, CodeBlocks, Simuino a další. Některá řešení jsou „zdarma“, některá licencována a zpoplatněna.

Hardwarových řešení byl v práci přiblížen jen malý vzorek. Ve skutečnosti existuje více originálních řešení, či nespočetně více neoficiálních řešení z čínské distribuce. Je tedy jen na uživateli, co si koupí. Pro nezkušeného začínajícího uživatele je vhodné využití simulátorů, případně pořízení neoficiální verze Arduino Uno. Po zabřednutí do problematiky, porozumění a vůli dále pracovat v této oblasti se otevírají neskutečné možnosti jak Arduino využívat.

2 BEZPEČNOSTNÍ TECHNOLOGIE A SYSTÉMY

Bezpečnostní technologie a systémy jsou klíčovým prvkem ochrany majetku, informací a osob v soukromém i veřejném sektoru. V dnešní době se stále více spoléháme na technologie, které nám pomáhají zlepšit bezpečnost a snižovat rizika. Pro potřeby této bakalářské práce, jejímž cílem je zjistit, kde lze nejčastěji využít platformy Arduino, budou přiblíženy pojmy poplachové zabezpečovací a tísňové systémy (dále jako PZTS), elektrická požární signalizace (dále jako EPS) a elektronické systémy kontroly vstupu (dále jen SKV).

2.1 Poplachový zabezpečovací a tísňový systém

Poplachový zabezpečovací a tísňový systém anglicky Intruder and Hold-up Alarm System (I&HAS) je soubor zařízení, technologií a postupů navržených tak, aby chránily majetek před nebezpečím, jako jsou vloupání, požáry, záplavy nebo úrazy. Tento systém zahrnuje detektory, senzory, ovládací panely, komunikační zařízení a další prvky, které spolupracují na identifikaci hrozeb a rychlé reakci na ně. Mezi nejzákladnější prvky PZTS patří poplachové zabezpečovací systémy, dle dřívějších definic také detektory narušení. Jednoduše řečeno se zabývá detekcí a signalizací přítomnosti, vniknutí nebo pokusu o vniknutí narušitele do sledovaných prostor. Poplachový tísňový systém pak slouží k úmyslnému vyvolání poplachového stavu a přivolání tak například pomoci. Při zajišťování fyzické bezpečnosti se dělí stupně ochrany na perimetrickou, plášťovou, prostorovou a předmětovou ochranu. [14] [15] [16]

2.1.1 Plášťová ochrana

Plášťová ochrana je souhrn opatření realizovaných na plášti chráněného objektu, například budovy. Cílem je zajistit detekci narušení, či pokusu o narušení tohoto pláště, aneb proniknutí například oknem, dveřmi a zdí. Prvky pro tuto detekci jsou zpravidla umístěny vevnitř objektu a může se jednat o magnetické kontakty, senzory na ochranu skleněných ploch, mechanické kontakty, vibrační senzory, poplachové fólie, drátové senzory, či rozpěrné tyče. Nejslabším článkem pláště objektu bývají okna, nejběžnějším prvkem plášťové ochrany jsou tedy senzory tříštění skla a magnetické kontakty sledující jejich otevření. Do plášťové ochrany lze také zařadit dohledové systémy, laicky kamerové, a další prvky. Převážná většina těchto prvků se vyznačuje širší detekční charakteristikou a kratším dosahem. [14] [16] [17]

2.1.2 Prostorová ochrana

Prostorová ochrana se zabývá odhalením pohybu narušitele uvnitř chráněného objektu, a jsou tedy realizována ve vnitřních prostorách. Opět jsou využívány dohledové systémy, mechanické zábranné systémy, zejména senzory pohybu a jiné detektory narušení. Senzory pohybu je možno rozdělit na aktivní a pasivní, dle způsobu detekce narušení. Pasivní infračervené (anglicky Passive InfraRed - PIR) senzory, sledují fyzikální změny v prostředí, konkrétně infračervené záření – teplotu a detekují změny v tomto prostředí (pohyb osoby s jinou teplotou než stacionární okolí) a můžou tak vyhodnotit narušení. Aktivní senzory si vytvářejí vlastní detekční pole a sledují změny v tomto prostředí, například různé mikrovlnné a ultrazvukové senzory. Případně může jít o kombinaci těchto sensorů v jednom zařízení a spojení výhod a nevýhod jednotlivých technologií. [14] [16] [17]

2.1.3 Předmětová ochrana

U předmětové ochrany je zaměření na zabránění zcizení, či neoprávněnou manipulaci s chráněnými aktivy – předměty, jako mohou být umělecká díla, cenné papíry, patenty, historické artefakty a jiné cenné objekty. Pro tuto ochranu jsou typické zabezpečené vitríny, závěsové senzory, polohové senzory, kapacitní senzory, seizmické senzory a další. U této ochrany lze také využít sensorů primárně určených například pro prostorovou ochranu či jiné. [14] [16] [17]

2.1.4 Perimetrická ochrana

Perimetrická ochrana se zabývá problematikou zabezpečení hranice pozemku chráněného objektu, nebo-li katastrální hranici prostoru. Může jít jak o přírodní překážky jako vegetace, vodní tok a podobně, nebo o umělé bariéry, jako ploty, zdi a jiné. Cílem jako u ostatních ochran je zpomalení a odhalení narušitele. Vzhledem k použití ve venkovním prostředí, jsou na tyto detektory kladeny vysoké nároky. Důležitá je odolnost vůči povětrnostním vlivům, ale i odolnost vůči planým poplachům, která bývá největší potíží. Vzhledem k různorodým druhům tvarů a způsobů vytvoření perimetru je spousta prvků a možností k jeho ochraně. Mimo dohledové systémy, které lze stejně jako u ostatních ochran také použít, se můžeme například setkat se světelnými závorami, laserovými bránami, zemními tlakovými hadicemi, šterbinovými kabely, perimetrickými PIR detektory a další. [14] [16] [17]

U perimetrické ochrany nemusí jít pouze o detekování narušitele (zloděje), ale může zde být zakomponováno například pomocí laserových bariér hlídání bazénu před malými dětmi, které by mohli do něj spadnout.

2.1.5 Tísňové systémy

Tato zařízení umožňují uživatelům rychle a snadno aktivovat poplach v případě nouze, například při přepadení nebo zdravotním problému. Můžou být ve formě tlačítek na veřejných budovách, pro přivolání pomoci například ze zdravotních důvodů. Nebo se může jednat o osobní bezdrátová tlačítka pro nemocné a seniory, kterými mohou přivolat pomoc. Případně také může jít o automatické systémy, které vyvolají tísňové hlášení nezávisle na obsluze. [17]

2.1.6 Shrnutí

Veškeré detektory napříč všemi druhy ochrany by také měly splňovat další kritéria. Měly by mít detekci sabotáže, to znamená odhalit, zda se je nesnaží někdo rozebrat, pozměnit, poškodit s nimi či zakrýt jejich pole působení (Antimasking). Vhodným prvkem je také automatické testování funkce, případně i manuální test, zda vše funguje tak, jak má. Dále by měly být odolné vůči chybové funkci a vyvolání planých poplachů. Všechny tyto parametry a kvalita detektorů bývá zohledněna také v ceně produktu a je jen na nás, kterou variantu zvolíme. U výběru a způsobu zabezpečení bychom samozřejmě měli zohlednit, zda cena zabezpečení nepřevýší cenu chráněného objektu. Asi stodoletu se senem nezabezpečíme laserovými závorami a nedělní obrázek dítěte kapacitním polem. Celkově je poplachový zabezpečovací a tísňový systém navržen tak, aby poskytoval ochranu a bezpečnost pro majetek a osoby v případě různých hrozeb a nebezpečných situací. [14] [16] [17]

2.2 Elektrická požární signalizace

Elektrická požární signalizace má za úkol rychle a spolehlivě určit místo požáru, vyhlásit požární poplach, kontaktovat oprávněnou osobu či hasičský záchranný sbor, aktivovat a řídit evakuační systém a další požárně bezpečnostní zařízení, jakými mohou být samočinné hasební zařízení (sprinklery, odvětrání a podobně), zpřístupnění klíčového trezoru, automatické odemčení dveří a další. Srdcem celého systému je pak ústředna EPS, která tyto činnosti zabezpečuje, tedy přijímá vstupní informace různých hlásičů, jak automatických (detektory kouře a tepla, detektory plynu), tak z manuálních tlačítek. Tyto informace zpracovává a provede výše uvedené činnosti. [14] [16] [17]

U autonomních elektrických požárních systémů je vyžadována vysoká spolehlivost, což obnáší zdlouhavé certifikace a testování, které obnáší nemalé finanční náklady. Pro aplikace v EPS je tedy zcela nevhodné použití technologie Arduino. Navíc licenční podmínky této platformy to přímo zakazují. [2] [14]

2.3 Elektronické systémy kontroly vstupu

Systémy kontroly vstupu mají za úkol řídit a evidovat přístup do zabezpečených prostor či objektů na základě jasně přidělených přístupových práv každému uživateli, kterého identifikuje, dle požadovaného stupně oprávnění, časového harmonogramu, personální politiky či další. Hlavní funkcí SKV je omezit přístup neoprávněným osobám do zájmového objektu. Tyto systémy také umožňují sledování pohybu osob v objektu, což může být užitečné pro řešení bezpečnostních incidentů či vyšetřování. Další funkcí může být evidence docházky zaměstnanců, což může být důležité pro účely plánování a řízení lidských zdrojů. Není to ovšem primární funkcí a neplést si je s docházkovým systémem. Docházkové systémy bývají jednodušší a využívají se zpravidla při vstupu do firmy. Existuje několik identifikačních prvků, které se v SKV používají. Mezi nimi můžeme najít čipové, magnetické, optické a biometrické. [14]

2.3.1 Čipové identifikační prvky

Asi nejběžnějším představitelem jsou čipové karty, ale může jít i o různé přívěsky. Výhodou karty je, že na ni lze umístit fotografii a jiné osobní údaje k identifikaci osoby fyzickou os-
trahou. Čipy mohou být pasivní, tedy bez vlastního zdroje energie. Velmi vzácně se využívají aktivní, s baterií, ale ty se uplatňují spíše v jiných odvětvích, než je identifikace osob. V této oblasti se nejvíce rozšířilo používání identifikace na rádiové frekvenci (dále jako RFID – Radio Frequency IDentification). [14]



Obrázek 8 Ukázka multifunkční identifikační karty [18]

RFID technologie využívá rádiové frekvence pro komunikaci mezi čtečkou a RFID „tagem“. Zpravidla ve formě karty (obr. 8) nebo přívěsku. Přístupová čtečka komunikuje s čipem na kartě tak, že svým elektromagnetickým polem dodá kartě dostatek energie pro funkci čipu a může tímto způsobem identifikovat „tag“, neboli uživatele. Na základě získaných informací pak řídicí jednotka provede autentizaci a danou osobu vpustí, či nikoliv, popřípadě provede záznam do docházkového systému. [14]

2.3.2 Magnetické karty

Magnetické karty jsou typické černým pruhem přes šířku karty, jak je patrné z ukázky kombinované karty (obr. 8). Jedná se o magnetický pásek, kde pomocí zmagnetizování ukládáme binární 1 a binární 0 v případě nezmagnetizování. Čtení pak probíhá mechanickým přejetím karty ve čtečce. Princip čtení je obdobný jako u čárového kódu. Nejčastěji se s touto technologií lze setkat u bankovní karty, která se používá pro otevření dveří bankomatu mimo pracovní dobu pobočky. [14] [19]

2.3.3 Optické identifikační prvky

Tento prvek funguje na principu 2D a QR kódu. Tyto kódy jsou čteny opticky a příkladem může být opět uvedená karta (obr. 8). Velkou nevýhodou je snadné okopírování takového kódu (stačí například vytisknout uvedený příklad), a proto se téměř nepoužívá, v žádném případě pak u bezpečnostních aplikací. [19]

2.3.4 Biometrika

Biometrické systémy kontroly vstupu využívají jedinečných fyziologických nebo behaviorálních charakteristik jednotlivce, jako jsou otisky prstů, rozpoznání obličeje, sítnice a duhovky oka, hlasový vzor a další. Biometrické údaje je téměř nemožné zfalšovat a nabízejí tak vyšší úroveň bezpečnosti. Tyto systémy jsou ale mnohdy náročnější na zpracování a příjem identifikačních dat a tím pádem dražší. [14] [19]

SKV nese podle způsobu použití spoustu výhod. Lze ušetřit za výměnu zámku v případě ztráty klíče (u ztráty identifikačního prvku, se smaže záznam v databázi). Díky tomuto systému dosáhneme zvýšení bezpečnosti, kdy je omezen přístup neoprávněných osob do objektu, a předcházíme tak vandalismu, úniku citlivých informací a snižuje se riziko krádeže. Je umožněno snadno měnit oprávnění uživatelů, například při změně pracovní pozice. V případě krizových situací dává přehled o počtu osob a jejich přibližné poloze a další.

Ideální variantou je kombinace různých identifikačních prvků dohromady. V některých případech je to dokonce vyžadováno. Například pro vstup do zabezpečených oblastí stupně utajení „tajné“, je zapotřebí přiložit identifikační kartu a zároveň zadat PIN na klávesnici čtečky. Systémy kontroly vstupu jsou důležitým nástrojem pro zajištění bezpečnosti objektů a ochranu citlivých informací. Díky široké škále technologií a funkcí mohou být tyto systémy přizpůsobeny potřebám různých typů organizací a prostředí.

2.4 Shrnutí

Oblast bezpečnostních technologií a systémů je velmi rozsáhlou oblastí a zmíněna byla jen velmi malá část. Jenom pro uvedené části je k dispozici několikero knih, skript, norem a mnohostránkových materiálů, více či méně odborných, sloužících pro orientaci v problematice. A snad pro každou z těchto částí, můžou být psány samostatné závěrečné práce při studiu oborů, které nabízí nejedna univerzita. Cílem této kapitoly, bylo spíše přiblížit neznalým čtenářům, že něco takového existuje a seznámit je se zavedenými termíny. Pro další studium pak můžou využít zdroje uvedené ke konci práce.

3 ARDUINO V BEZPEČNOSTI

V komerční sféře a zejména v bezpečnosti jsou často vyžadovány certifikace. Arduino má krátký životní cyklus a certifikace se tak nevyplatí. Navíc Arduino není možné používat tam, kde závisí lidský život a zdraví na jeho funkčnosti. Příkladem tak může být EPS. Nicméně je to vynikající nástroj na pochopení funkce některých prvků využívaných v bezpečnosti, jejich zkoušení, testování a hlavně výuku. Na základě získaných poznatků pak lze sestavit vlastní hardware, provést certifikaci a začít prodávat. Jde případně využít pro osobní účely k zabezpečení, protože i necertifikované, ne stoprocentně funkční zabezpečení, je lepší než žádné. [2]

V první kapitole bylo rozebráno samotné Arduino jakožto řídicí prvek, který sám o sobě nenabízí velkého využití. Je zapotřebí využít různých přídavných senzorů, plošných spojů, Shieldů a podobně. Některé z těchto komponent, které jsou využitelné v bezpečnosti, budou přiblíženy v této kapitole.

3.1 Senzory

Senzory, tedy čidla na plošných spojích s potřebnými součástkami pro jejich funkčnost je možné rozdělit do kategorií dle jejich snímaných fyzikálních veličin a v jednotlivých odstavcích si přiblížit konkrétní výrobky.

3.1.1 Teplota, vlhkost a tlak

V této kategorii se nachází nepřeberné množství teplotních čidel od součástkových přes kuchyňské po zapouzdřené a vodotěsné. Specialitou jsou pak čidla snímající teplo v infračerveném pásmu. Zvláštní kategorii tvoří čidla měřící vzdušnou vlhkost, či atmosférický tlak. [20]

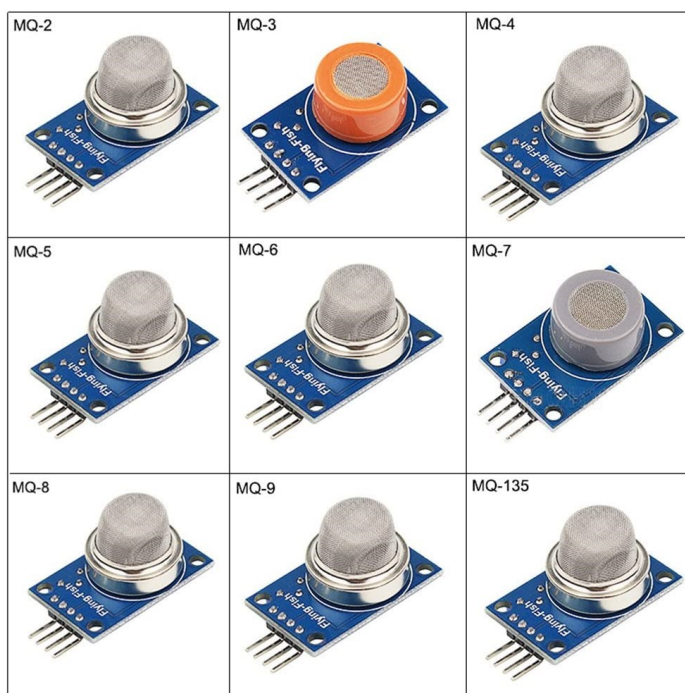


Obrázek 9 Teplotní čidlo DHT11 a infračervený teploměr MLX90615 [20]

V bezpečnosti lze pak využít zejména teplotní čidla jako jeden z prvků pro detekci požáru. Infračervené teploměry (obr. 9 vpravo) lze pořídit okolo 500,- Kč a obyčejná teplotní čidla (obr. 9 vlevo) zhruba od 50,- Kč. [20]

3.1.2 Plyn a kvalita ovzduší

Laicky by se mohlo zdát, že kvalitu ovzduší v bezpečnostních aplikacích není nutné řešit, ale patří sem detekce různých plynů (ať hořlavých či jedovatých). Sledování kvality ovzduší je tedy žádoucí pro prevenci před vznikem požáru či otrávením. Hlavními představiteli v této kategorii jsou čidla MQ-2 až 9 a 135 zobrazených níže (obr. 10). [21]



Obrázek 10 Čidla plynů [22]

Jak je z obrázku patrné (obr. 10), čidla vypadají téměř všechny stejně. Dělí se jen podle toho, jaké plyny detekují (tab. 1).

Tabulka 1 Čidla plynů [23]

Označení	Detekované plyny
MQ-2	Kouř, hořlavé plyny
MQ-3	Etylalkohol (C_2H_5OH)
MQ-4	Metan (CH_4)

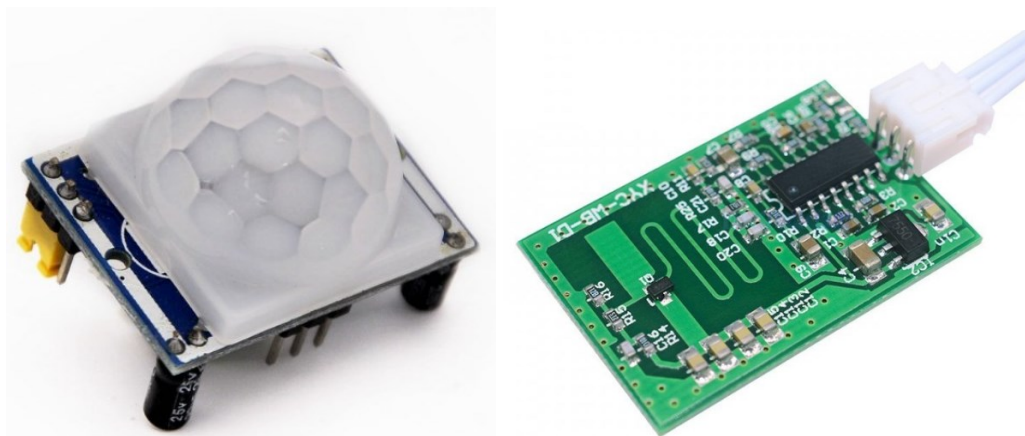
Označení	Detekované plyny
MQ-5	Metan (CH ₄), propan-butan (LPG)
MQ-6	Propan-butan (LPG)
MQ-7	Oxid uhelnatý (CO)
MQ-8	Vodík (H ₂)
MQ-9	Metan (CH ₄), oxid uhelnatý (CO)
MQ-135	Benzen (C ₆ H ₆), čpavek (NH ₃)

Většina těchto čidel snímá i ostatní hořlavé plyny, ale se sníženou citlivostí. Tato čidla lze pořídit okolo 60,- Kč. Případně lze pořídit sofistikovanější a komplexnější řešení s optickou detekcí. Pořizovací náklady u takových řešení jsou pak ve stovkách až tisících korun. [21]

3.1.3 Vzdálenost, pohyb a vibrace

V této části lze čidla rozdělit na snímače pohybu okolí, snímače pohybu „sebe sama“ a měřiče vzdálenosti. Jednotlivé kategorie nalézají různé způsoby využití v bezpečnosti.

Snímače pohybu okolí jsou zejména PIR čidla, která sledují změny v infračerveném pásmu (teploty) ve sledovaném prostoru a mikrovlnné detektory pracující na principu Dopplerova jevu. Kombinací těchto čidel je pak možné sestavit vlastní detektory pohybu, využitelné zejména v prostorové a perimetrické ochraně. [24]



Obrázek 11 PIR detektor HC-SR501 a mikrovlnný detektor XYC-WB-D1 [24]

Modul pyroelektrické detekce HC-SR501 (obr. 11 vlevo) lze pořídit za 38,- Kč a na stejném obrázku vpravo mikrovlnný detektor pohybu Doppler radar XYC-WB-D1 pak za 68,- Kč. [24]

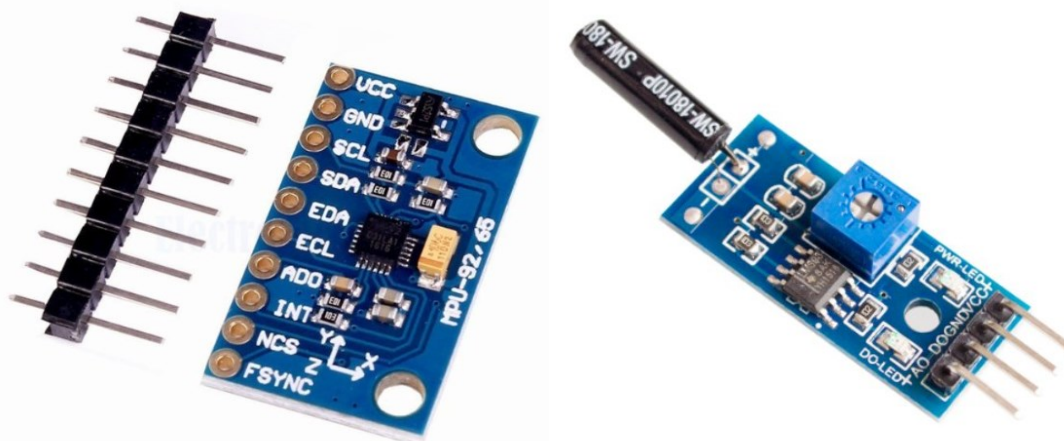
Měřič vzdálenosti je vhodným naprogramováním možné použít také pro sledování pohybu jako předešlé senzory. Vhodnější aplikací by však mohlo být jejich využití v rámci předmětové ochrany u stacionárních objektů, kdy senzor bude daný předmět „pozorovat“ a vyhodnotí jeho pohyb. V této skupině se nachází zejména čidla ultrazvuková, laserová a infračervená. Všechny pracují na principu vyslání do prostředí nějakého prvku, který se zpravidla odrazí od překážky a vrátí zpět do čidla, které na základě časové prodlevy vyhodnotí vzdálenost. [24]



Obrázek 12 Měřič vzdálenosti ultrazvukový HC-SR04 a laserový VL53L1X [24]

Uvedené měřiče vzdálenosti (obr. 12) je možné pořídit opět za relativně nízké ceny. V případě ultrazvukového měřiče HC-SR04 jde o 38,- Kč a u laserového VL53L1X I2C o trochu více a to 228,- Kč. [24]

Vibrační čidla detekující pohyb „sebe sama“ pracují na principech, které se liší podle druhu pohybu. Gyroskopy sledují svůj náklon, akcelerometry zrychlení či zpomalení a elektronické kompasů otáčení v ose. Zde se přímo nabízí využití v oblasti předmětové ochrany. Může jít o použití jak na samotném chráněném předmětu nebo schránce, ve které je předmět umístěn. Tato čidla lze také využít pro zjištění sabotáže na jiná zařízení například pokus o otočení kamery nebo rozebírání bezpečnostního prvku jako třeba PIR detektoru. [24]

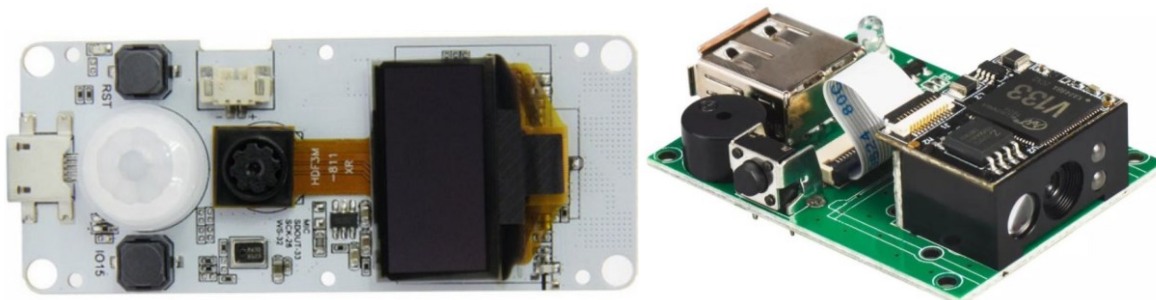


Obrázek 13 3-osý gyroskop, akcelerometr a magnetometr MPU-9250 SPI/IIC a vibrační snímač SW1801P [24]

U kombinovaného modulu gyroskopu, akcelerometru a magnetometru MPU-9250 (obr. 13 vlevo) je cena 458,- Kč, v případě verze MPU6050 bez magnetometru 68,- Kč a vibrační snímač SW1801P v témže obrázku vpravo stojí 18,- Kč. [24]

3.1.4 Světlo, barva a optika

Hlavním představitelem v této kategorii jsou kamery. Hodí se pro použití v bezpečnosti, například pro prostorovou ochranu. Můžeme z nich sestavit různé fotopasti. Při využití je nutné použít dalšího modulu z důvodu malé paměti Arduina. Buď jej lze doplnit o modul pro vložení paměťové karty nebo nějaký komunikační modul, který bude záznam posílat například na server, popřípadě kombinace obojího. Můžeme zde nalézt čtečku čárových kódů pro využití u docházkového systému. Také sem patří různá čidla pro měření intenzity osvětlení, které najdou využití spíše v chytré domácnosti než v bezpečnosti. [25]

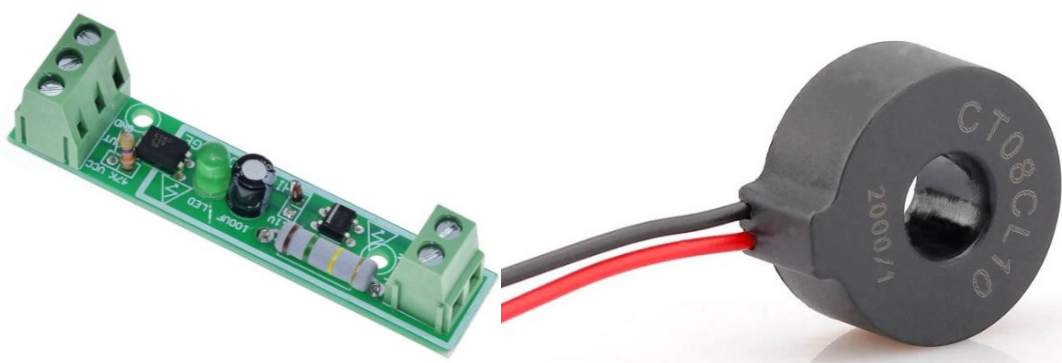


Obrázek 14 LilyGO TTGO T-Camera ESP32 s mikrofonem a PIR a čtečka čárových kódů YHDAA YHD-M800D [25]

U kamery LilyGO TTGO T-Camera ESP32 (obr. 14 vlevo) je pořizovací cena 1058,- Kč. Jak jsme zmínili je nutné Arduino doplnit o další Shieldy, například komunikační na připojení do ethernetu za 648,- Kč. Připočteme-li zároveň cenu desky, dostáváme se do částek, kdy lze pořídit již kompletní, případně certifikované, zařízení. Čtečku čárových kódů YHDAA YHD-M800D lze pak pořídit za 1088,- Kč, což jako v předešlém případě není výhodnou koupí. [25]

3.1.5 Proud a napětí

V této části se nachází prvky pro měření proudu a napětí, které se využijí zejména pro monitorovací systémy a kontrolu funkčnosti jiných zařízení, případně ke sledování stavu záložních zdrojů napájení. Najdou zde využití zejména Hallovy senzory, které detekují magnetické pole. A jak je známo, čím větší proud protéká, tím intenzivnější je magnetické pole. Různým naprogramováním takových senzorů pak lze sledovat například pokles, či úplnou ztrátu proudu vedoucího k bezpečnostním prvkům a upozornit tak na poruchu v zařízení.

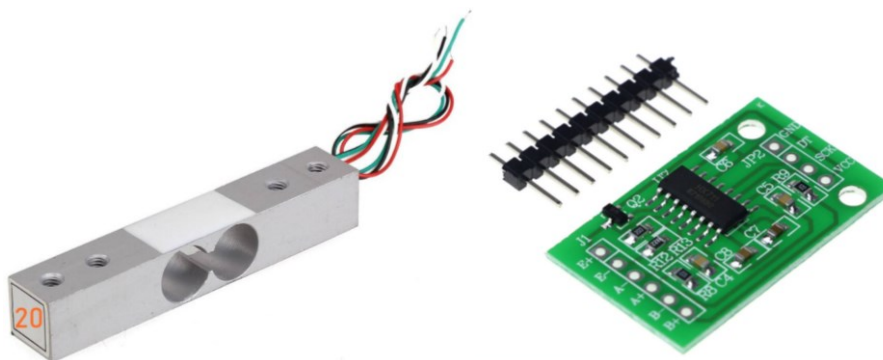


Obrázek 15 Snímač střídavého napětí 230 V a cívka pro měření proudu CT08CL10 [26]

Snímač střídavého napětí 230 V lze pořídit za 58,- Kč a o trochu dráže pak snímací cívku pro měření proudu CT08CL10 za 68,- Kč. [26]

3.1.6 Hmotnost

V nejednom filmu bylo při krádeži drahých předmětů využito hmotnostních kontrol, tedy když někdo něco ukradl, nahradil to stejně těžkým předmětem. Samozřejmě jde jenom o film, ale u předmětové ochrany toho můžeme docílit právě instalováním čidla pro hlídání hmotnosti do podložky předmětu. Čidla pracují na principu změny odporu podle toho, jak velká tíha působí na jejich kovové jádro. [27]

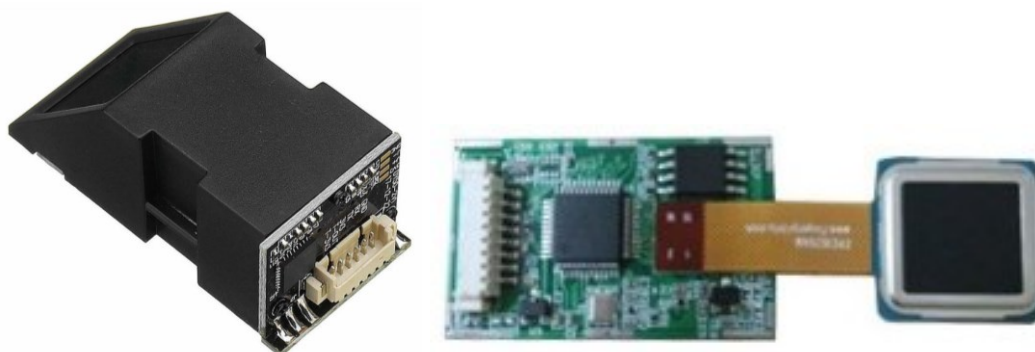


Obrázek 16 Vážicí senzor GUANG CE YZC-131 a převodník HX711 [27]

Čidla se prodávají podle toho, jak velkou hmotnost předpokládáme měřit. V nabídce jsou různé varianty od jednoho po sto kilogramů. K těmto čidlům je zapotřebí dokoupit také převodník, který stojí 28,- Kč. Samotný vážicí senzor GUANG CE YZC-131 (obr. 16) pak stojí 128,- Kč. [27]

3.1.7 Biometrika

Pro využití v dohledových a docházkových systémech můžeme využívat biometriku, v této části se nacházejí čtečky otisků prstů. Může jít o čtečky optické, které prst vyfotí a porovnají s uloženým vzorkem, nebo kvalitnější a také dražší kapacitní.



Obrázek 17 Čtečka otisků prstů optická DY50 a kapacitní FPC1020AM [28]

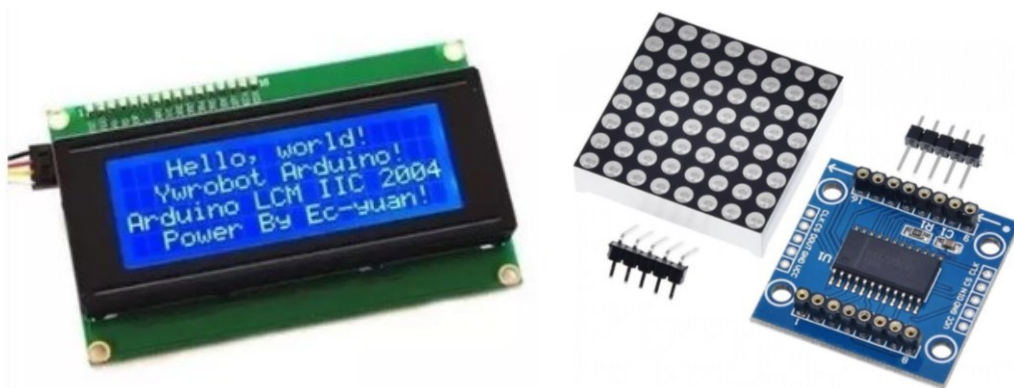
V případě obyčejné čtečky otisku prstů DY50 (obr. 17 vlevo) je cena stanovena na 398,- Kč, sofistikovanější kapacitní čtečku FPC1020AM s pamětí až pro 500 vzorků otisků prstů (obr. 17 vpravo) pak pořídíme za 1198,- Kč. [28]

3.2 Signalizace

Detekuje-li senzor nějaké narušení, je zapotřebí, aby o tom dokázal také nějakým způsobem informovat. Může to provést akusticky, opticky, nebo danou informaci přenést do jiného systému, kde lze provádět například další vyhodnocení.

3.2.1 Optická

Opticky je možné informovat pomocí různě barevných LED, žároveček, blesků, ale při praktickém využití například majákem. V neposlední řadě lze informaci, co konkrétního se stalo, uvést v textové podobě na displeji, kde je uživatelsky přívětivější zobrazení. Technologií u displejů je několik od maticového LED, přes LCD, TFT, OLED až po E-ink. Liší se samozřejmě cenou a je zapotřebí vybrat vhodný v závislosti na požadavcích na zobrazované informace. [29]



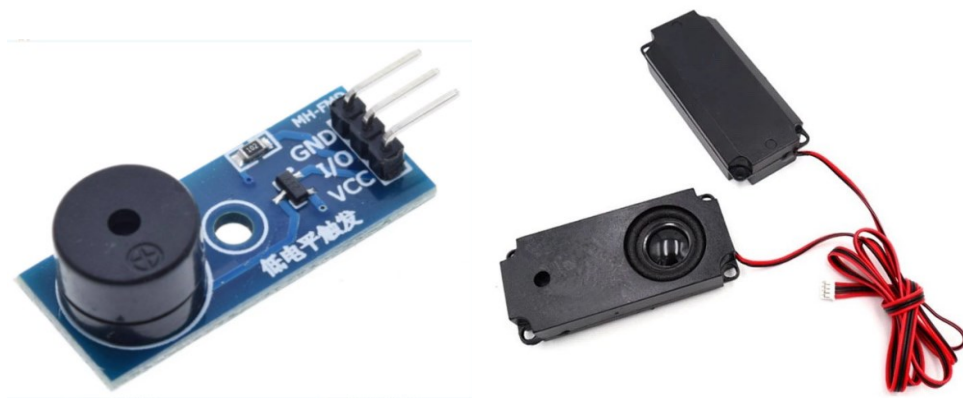
Obrázek 18 LCD displej 2004 a maticový 8x8 LED displej MAX7219 [29]

Maticový LED displej MAX 7219 (obr. 18 vpravo) s cenou 68,- Kč je možno použít například k zobrazení kódu chyby. Může také rozsvěcovat jednotlivé diody, které ve spojení s legendou určí, o jaké narušení se jedná, případně rozblikat celý displej a simulovat tak maják. Programování těchto displejů je poměrně náročnější. LCD displej 2004 s integrovaným převodníkem I²C (obr. 18 vlevo) s cenou 208,- Kč využijeme právě na přímé zobrazení informace, co se stalo. U většiny aplikací by displej neměl chybět. [29]

3.2.2 Akustická

Nejenom optická signalizace je žádaná, ale u většiny prvků bezpečnosti je potřebná i akustická. Ať už se jedná o vyplašení narušitele sirénou, nebo upozornění uživatele na poruchu

či narušení. Důležitá je také signalizace provedení některých úkolů, například akustické potvrzení načtení identifikační karty. V nabídce jsou různé bzučáky, reproduktory, ale také se nabízí možnost přes relé připojit sirénu.

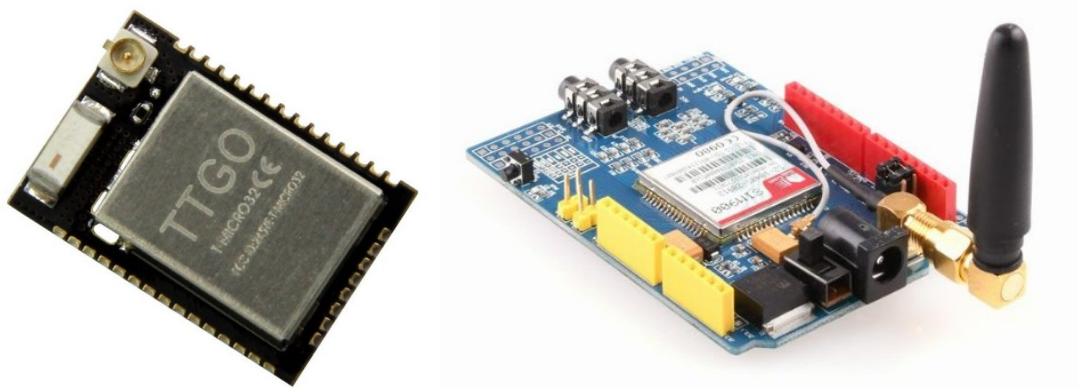


Obrázek 19 Bzučák a stereo reproduktor 5 W [30]

Bzučák lze pořídit na plošném spoji, jak je uvedeno (obr. 19) za 38,- Kč, varianty samostatných bzučáků jsou již od 4,- Kč. Výkonné 5 W stereofonní reproduktory pak lze koupit za 138,- Kč. [30]

3.2.3 Přenosem

V případě komplexnějšího a praktičtějšího řešení je potřeba senzory propojit, ideálně pak řídit a sbírat informace na jednom místě, třeba na domácím PC. K tomu je zapotřebí Arduino rozšířit o možnost přenášet získaná data k tomuto jednomu řídicímu prvku – ústředny, která může být opět realizována Arduinem, nebo třeba serverem. K přenosu je možno využít bezdrátových technologií jako Wi-Fi, Bluetooth, IrDA, GSM a v neposlední řadě dnes moderní IoT, nebo metalickým vedením po ethernetu.



Obrázek 20 Modul Wifi s Bluetooth LilyGO a GPRS-GSM SIM900 Shield [31]

Na rozdíl od senzorů se u těchto komunikačních modulů pohybuje pořizovací částka v řádech stokorun a cena se odvíjí podle použité technologie, kvality a univerzálnosti použití. Zobrazený modul Wi-Fi kombinovaný s Bluetooth LilyGO TTGO Micro-32 V2.0 CE (obr. 20 vlevo) lze pořídit za 238,- Kč a rozšiřující shield GPRS-GSM SIM900 (obr. 20 vpravo) pak za 688,- Kč. [31]

3.3 Doplnky

K tomu, aby mohlo zařízení fungovat, je zapotřebí mu také dodat nějakou energii. Je několik variant, jak Arduino napájet. Je možnost jej připojit na AC-DC zdroj napájený zpravidla z veřejné sítě střídavého napětí 230 V, které změní na stejnosměrné pro napájení Arduina, aneb takzvané „trafo“. K napájení lze také využít jednorázových baterií nebo nabíjecích akumulátorů, které lze zkombinovat se solárním panelem (obr. 21 vpravo). Velice záleží na konkrétních aplikacích a požadavcích využití. Také je vhodné mít sestavu pod kontrolou a využít tak různých ovládacích prvků, jako jsou klávesnice (obr. 21 vlevo), tlačítka, ale také například dálkové ovladače.



Obrázek 21 Maticová klávesnice 4x4 a solární panel 5 V 4 W [32] [33]

Takovou klávesnici s maticí 4x4 lze pořídit za 108,- Kč a solární panel pro napájení, nebo spíše dobíjení akumulátoru ve vyobrazeném provedení za 538,- Kč. [32] [33]

Mezi další prvky, které je nutné pořídit, tak abychom mohli Arduino „zapojit“, patří různé vodiče nebo-li propojovací kabely. U složitějších aplikací najdou uplatnění nepájivá kontaktní pole a v případě trvalého využití je možno si s cuprexitovou deskou vytvořit vlastní tištěný spoj. Je možné pořídit také různé obaly, jak univerzální, tak určené již ke konkrétním

aplikacím. Pokud někdo disponuje technologií 3D tisku či gravírovacího laseru, může si takovéto boxy vyrobit. Určitě je spousta dalších součástek a prvků, které je možné zde uvést, ale většinou přijdou na mysl, až jsou zapotřebí. [8]

3.4 Shrnutí

V této části práce byly přiblíženy konkrétní prvky pro připojení k Arduino, které lze nějakým způsobem využít v bezpečnosti. Některá řešení jsou poměrně lukrativní, jak svým využitím, tak pořizovací cenou. Některá, opět díky ceně, lze využít spíše jen pro výukové a studijní účely – hraní si. Obecně pro většinu případů lze konstatovat, že výše uvedené senzory a příslušenství mají velmi nízké pořizovací náklady, jak bylo u příkladů uvedeno a určitě minimálně pro výukové účely není k „zahození“ si některá řešení pořídit a sestavit.

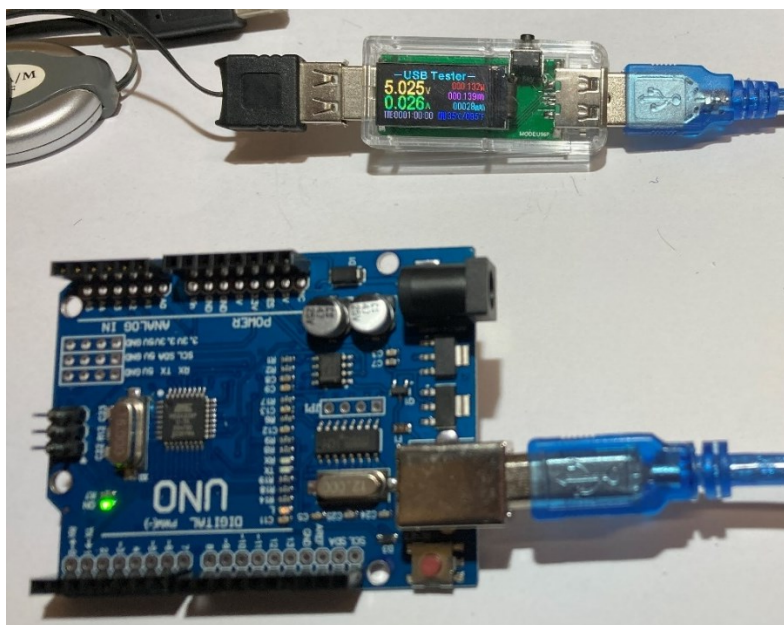
II. PRAKTICKÁ ČÁST

4 NÁVRHY PRAKTICKÉHO VYUŽITÍ ARDUINA PRO OBLAST BEZPEČNOSTI

V této části práce budou navrženy konkrétní aplikace, vysvětlena funkce a technické parametry jednotlivých použitých senzorů. Uvedená zapojení budou také změřena na proudový odběr. Zdrojové kódy se pak nachází v přílohách na konci práce, pokud by si to chtěl někdo vyzkoušet. Na začátku budou popsány jednodušší senzory jako vibrační, kouřové čidlo, detektor pohybu PIR, až po složitější SKV pomocí hesla a RFID technologie.

4.1 Měření proudového odběru

Většina bezpečnostních prvků je v praxi napájena z rozvodné sítě 230 V, ale v případě výpadku elektrické energie je chod zálohován z baterií. Je tedy nutné vést v patrnosti proudový odběr těchto zařízení, na jehož základě se určí kapacita baterie pro zabezpečení chodu systému při výpadku. V této části budou jednotlivá řešení měřena také na odběr proudu. K tomuto měření je využito měřicí zařízení USB Tester U96P (obr. 22), které bylo pořízené přes platformu www.aliexpress.com. Výrobce udává pro chod zařízení odběr menší než 3 mA, proto je tato hodnota již ze zobrazených údajů odečtena. Vzhledem k původu měřicího přístroje budou výsledky orientační a pro přesnější hodnoty by bylo zapotřebí provést laboratorní měření s patřičnými certifikovanými přístroji. [34]



Obrázek 22 Měření odběru Arduina pomocí USB Testeru U96P

Pro ověření funkce a provedení prvotního měření, byla připojena samotná deska Arduino přes USB k měřicímu zařízení a do PC (obr. 22). Zobrazený odběr takto připojené desky činil převážně v rozmezí 26 – 31 mA. Pro zjištění kvality měření a jeho ověření bylo provedeno testování o délce zhruba hodinu. Během delšího procesu kolísal jak testovaný proud, tak také napětí z ne zcela stabilního zdroje (PC-USB) a přesné výpočty tak není možno realizovat. Je-li předpokladem, že uvedené zařízení počítá správně, tak při aplikaci jednoduchého vzorce pro výkon, který využijeme pro zpětný přepočít (1), nám vychází odběr na 27,8 mA.

$$P = U \cdot I \rightarrow I = \frac{P}{U} = \frac{0,139}{5} \cong 27,8 \text{ mA} \quad (1)$$



Obrázek 23 Měření odběru Arduina napájeného 9 V baterií

Pro další testování využijeme i možnost připojení zařízení pomocí krokosvorek na baterii. Testování bylo provedeno přes 9 V baterii, které během zkoušení (10 minut) klesalo napětí z 8,2 V na 7,9 V (již slabší baterie). Podle uvedených údajů na USB Testeru (obr. 23), tedy odběru 37 mWh po 10 minutách je vypočten výkon podle vzorce (2). Pak je pomocí vzorce (3) vypočítána hodnota proudového odběru zhruba na 27,6 mA.

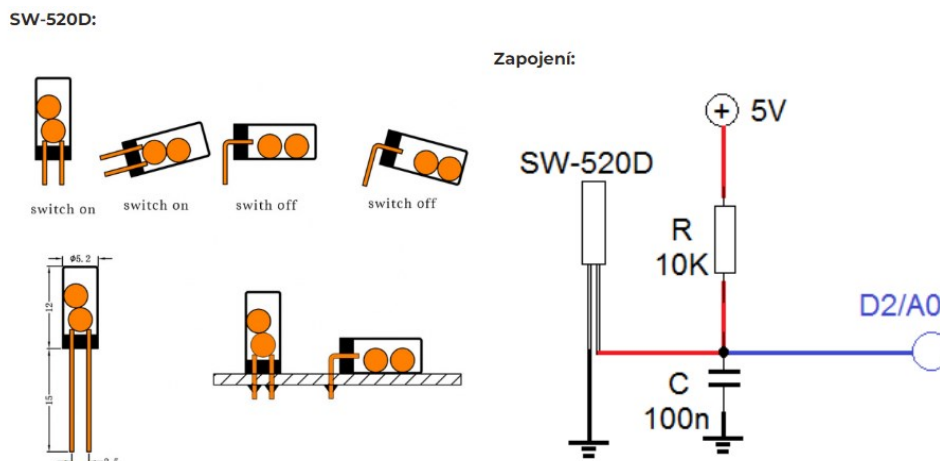
$$E = P \cdot t \rightarrow P = \frac{E}{t} = \frac{0,037}{\frac{1}{6}} = 0,222 \text{ W} \quad (2)$$

$$I = \frac{P}{U} = \frac{0,222}{8,05} \cong 27,6 \text{ mA} \quad (3)$$

Z výše provedeného měření a testování, lze odvodit, že pro účely hrubého přehledu je možné s tolerancí hodnoty odečítat přímo z USB Testeru. Při dalších měřeních bude počítáno s proudovým odběrem Arduina po zaokrouhlení 28 mA.

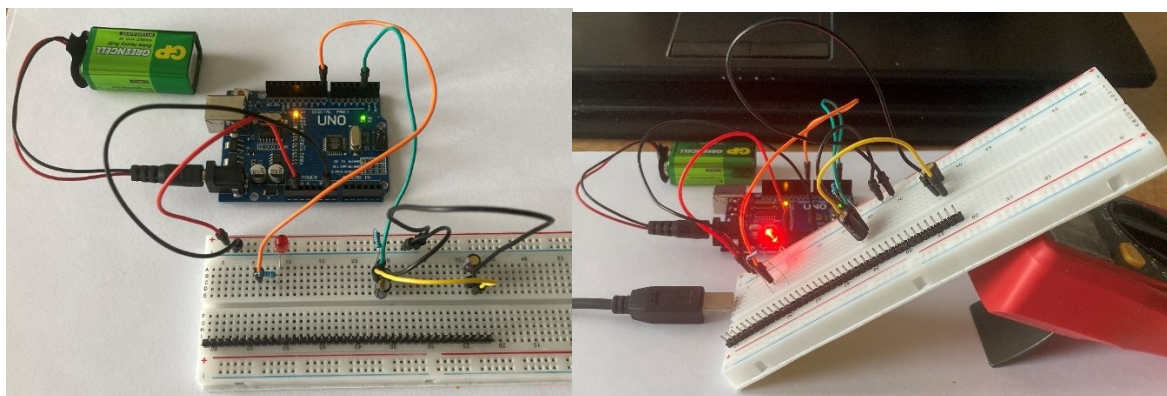
4.2 Otřesové čidlo

Pro toto řešení je využito otřesové čidlo SW-520D, které je principiálně velmi primitivní. Jedná se o dva kontakty přivedené do pouzdra, ve kterém jsou dvě kovové kuličky, které tyto kontakty spojí, jak je zobrazeno níže (obr. 24 vlevo).



Obrázek 24 Princip funkce otřesového čidla SW-520D a schéma zapojení [35]

Dle schématu zapojení (obr. 24 vpravo) je na vstupním pinu Arduina (D2) nízká úroveň (low nebo také 0 V) jsou-li kontakty kuličkou spojeny – obvod je uzemněn. Pokud je čidlo „rozpojeno“, bude pak na vstupním pinu (D2) vysoká úroveň (high, nebo také 5 V). Dioda pak signalizuje, je-li s čidlem pohnuto. Nebo-li když se rozpojí kuličky, dioda se rozsvítí (viz. praktická ukázka obr. 25). Jednoduchou změnou ve zdrojovém kódu, který je uveden v příloze PI, lze funkci čidla otočit pro případ, že výchozí poloha by byla s rozpojenými kontakty.



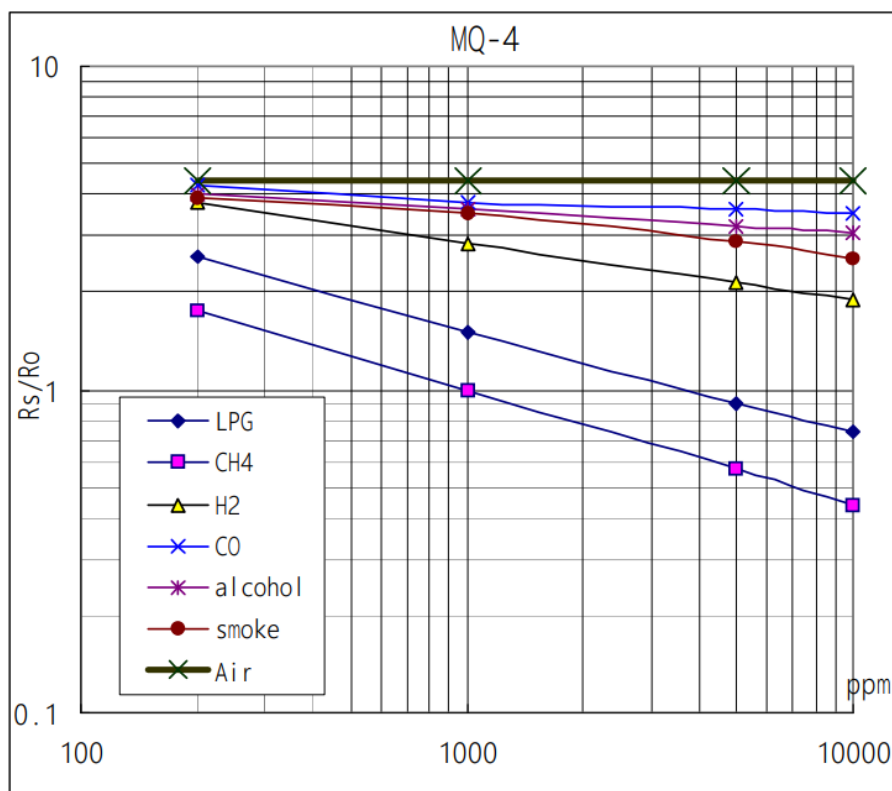
Obrázek 25 Ukázka zapojení a funkce otřesového čidla

Nastavením prodlevy (delay) ve zdrojovém kódu je možné regulovat citlivost. Při nastavení 10 ms reaguje LED při sebemenším otřesu, u finálního nastavení 100 ms je již zapotřebí větší rány, nebo trvalého rozpojení kontaktů. Pro praktické využití je zapotřebí správné umístění, nebo raději využít jiné čidlo, které pracuje na jiném, vhodnějším principu. Příkladem může být pružinka, která se při otřesu dotkne vodivé stěny, a uzavře tak obvod.

Vzhledem k principu čidla, kdy jde jen o prostý spínací kontakt, nemá smysl řešit proudový odběr tohoto zapojení, protože bychom dostali zhruba hodnotu odběru desky v provozu tedy 28 mA.

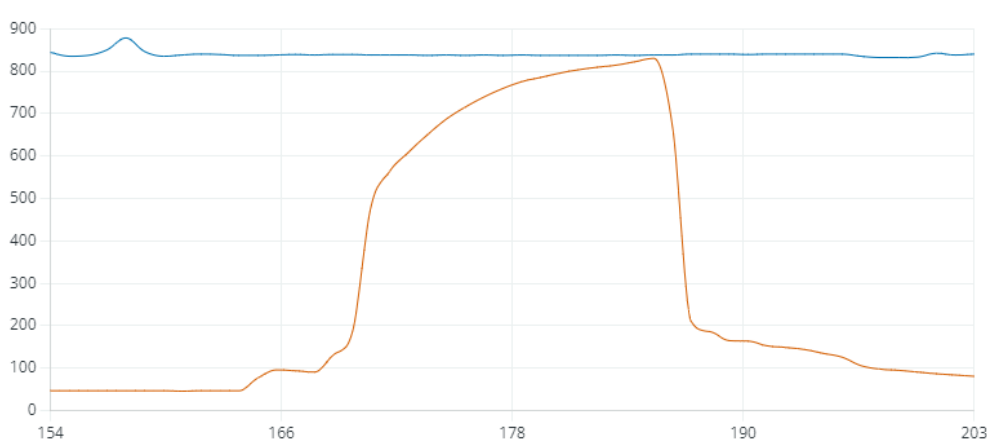
4.3 Senzor plamene a senzor hořlavých plynů

Senzor pro hořlavé plyny, zejména metan, využívá jako detekční prvek oxid cínčitý (SnO_2). Tento oxid má nízkou vodivost v „čistém“ vzduchu a s rostoucí koncentrací hořlavého plynu se vodivost úměrně zvyšuje. Podle počtu částic sledovaného plynu na milion ostatních (parts per million – ppm) je z grafu (obr. 26) patrné, jak se pro jednotlivé plyny mění odpor čidla. Pro „čistý“ vzduch je odpor stálý (zelená) a pro ostatní plyny se křivka s počtem částic strmě mění. Čidlo je určeno hlavně pro detekci metanu (CH_4), který má křivku nejstrmější (fialovo-modrá).



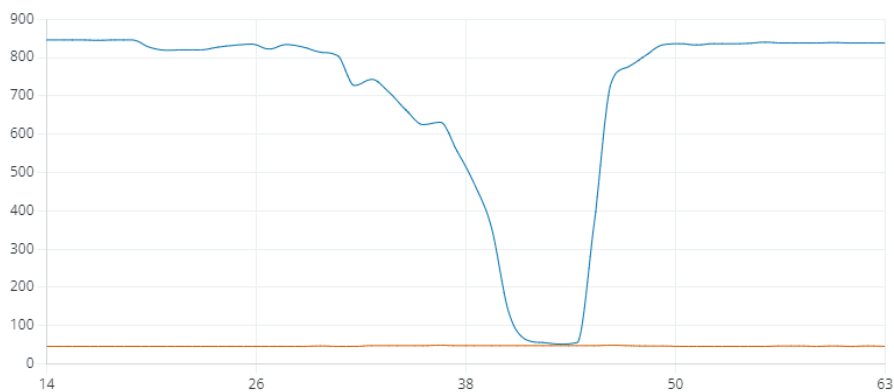
Obrázek 26 Detekční charakteristika čidla MQ-4 [36]

Plošný spoj obsahuje již převodník analogového signálu na digitální a Arduino tak pracuje již s údaji o koncentraci metanu ve vzduchu. Senzor je také osazen regulací (trimerem) pro nastavení spínané hodnoty pro digitální výstup. Ve výchozím nastavení je nastaven cca na hodnotu 300, jak bylo zjištěno při pozorování sériové komunikace při testování čidla. Zkoušení proběhlo tak, že byl na senzor nasměrován zapalovač a puštěn plyn. Křivka změny je pak znázorněna oranžovou barvou v grafu (obr. 27). [36]



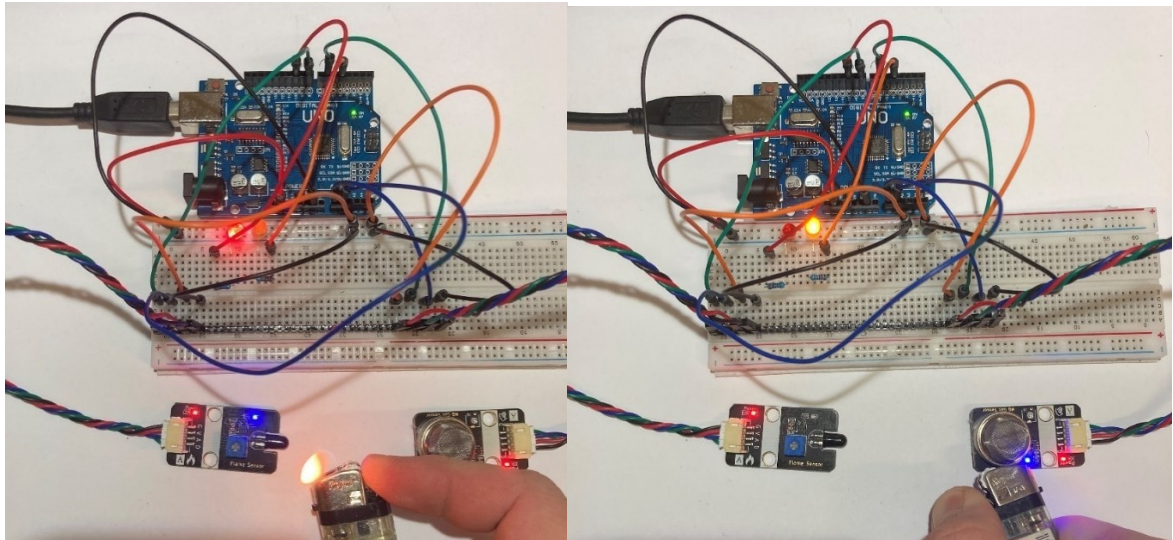
Obrázek 27 Graf analogového výstupu senzoru hořlavých plynů

Senzor plamene využívá pro detekci plamene infračervenou diodu, která snímá světlo o vlnové délce v rozsahu 760 – 1100 nm. Při hoření – požáru dochází právě k velmi intenzivnímu infračervenému záření. Čím více intenzivní záření je, tím více se sníží hodnota analogového výstupu. Na výstupu ze sériového monitoru zobrazeného v grafu (obr. 28) je princip patrnější. Pro testování bylo zhruba ve vzdálenosti 1 m od detektoru škrtnuto zapalovačem (počáteční výkyv) a pak byl přibližován až k čidlu a následně vypnut, průběh je pak znázorněn modrou křivkou. [37]



Obrázek 28 Graf analogového výstupu senzoru plamene

Hlavním čidlem je fotodioda YG1006 zatavená v černém epoxidu. Díky tomu je citlivá na infračervené záření. Celý senzor není odolný vůči ohni a je zapotřebí jej držet v dostatečné vzdálenosti od plamene, aby nedošlo k jeho poškození. [38]



Obrázek 29 Ukázka funkce senzoru plamene a hořlavých plynů

Zapojení obou senzorů bylo realizováno zároveň a zdrojový kód daného testu je pak v příloze PII. Jak pro plamen, tak pro detekci hořlavých plynů, proběhl test pomocí zapalovače, jak je vyobrazeno výše (obr. 29). Senzor pro detekci plamene je připojen na digitální vstup D8 a analogový A1. Detektor hořlavých plynů je pak napojen na digitální D6 a analogový A0. Sensory jsou přizpůsobeny platformě Arduino a připojení je tedy přímé, bez dalších komponent. Pro signalizace jsou použity diody umístěné za 220 Ω rezistorem. Pro detektor hořlavých plynů slouží žlutá dioda na digitálním výstupu D4 a pro senzor plamene červená dioda na digitálním výstupu D10. Oba senzory jsou uzemněny a napájeny napětím 5 V z desky Arduina.

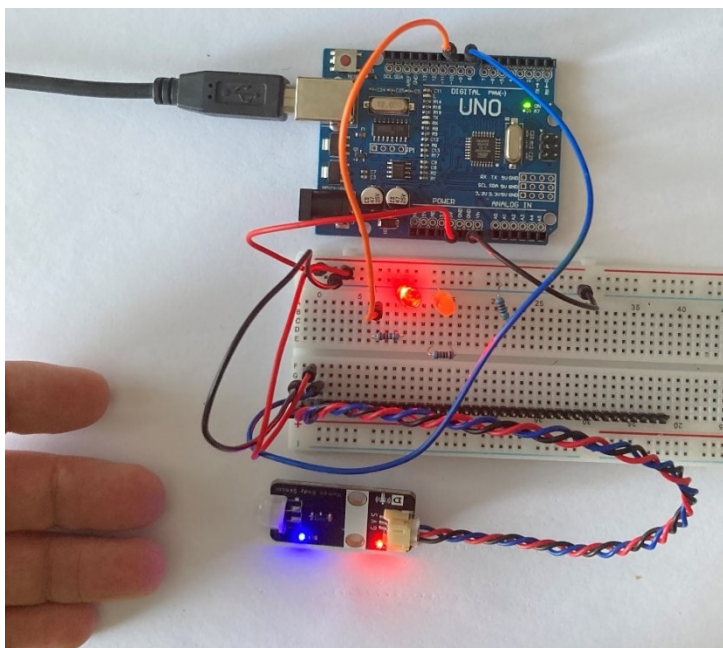
Tabulka 2 Výsledky měření proudového odběru – detektor hořlavých plynů

Měřená sestava	Proudový odběr
Kouřový senzor	128 – 134 mA
Celkový odběr	156 – 162 mA

Proudový odběr detektoru hořlavých plynů je dle prodejce 160 mA. Naším měřením byly zjištěny nižší hodnoty, které jsou uvedeny v tabulce (tab. 2). Z principu funkce čidla senzoru plamene lze odběr zanedbat, pouze námi použité provedení zvýšilo odběr o červenou diodu (asi 2 mA) a v případě detekce o modrou diodu, která má z principu technologie odběr o něco vyšší.

4.4 Detektor pohybu PIR

Pro tento test bylo použito čidlo HC-SR505. Pyroelement, tedy čidlo, je hlavním prvkem detektoru, který přijímá infračervené záření ze svého okolí. Začne-li se toto prostředí dynamicky měnit, tyto změny čidlo zachytí a detektor je pak vyhodnotí jako pohyb osoby. Na statické předměty tedy nereaguje a bere je za výchozí stav prostředí. Na základě výše zmíněné funkce je podmínkou pro správné fungování, aby pohybující se osoba měla jinou teplotu než snímané stacionární prostředí. [14] [16]



Obrázek 30 Ukázka detektoru pohybu

Tento detektor vrací pouze logickou hodnotu 0 či 1 na základě toho, zda detekoval pohyb osoby či nikoliv. Zapojení je tedy velmi jednoduché a naprogramování také. Samozřejmě záleží na konkrétní aplikaci. Zdrojový kód je uveden v příloze PIII, kdy na základě jednoho vstupu D8 z čidla, rozsvěcujeme diodu připojenou na D10 jako výstup (obr. 30).

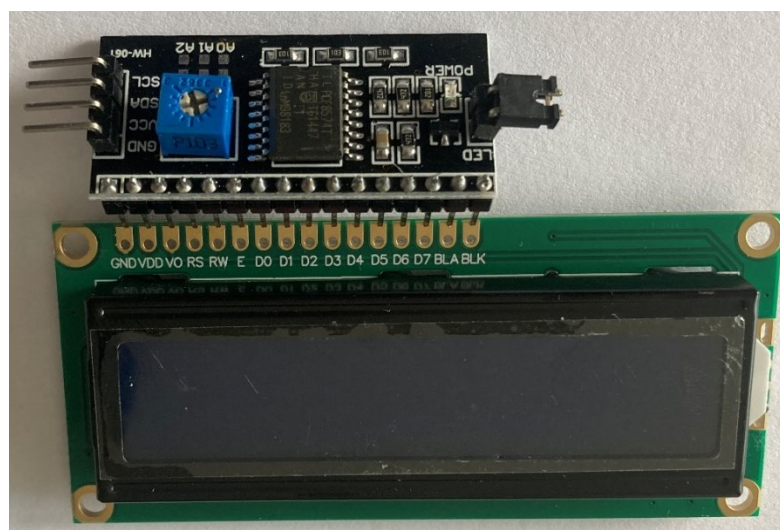
U měření proudového odběru PIR detektoru, obdobně jako u senzoru plamene, nebylo v rozsahu měřicího zařízení, kromě rozsvícení LED, naměřeno žádných zvýšených hodnot. Takový byl i původní předpoklad, protože se v obou případech jedná o pasivní čidla. Navíc je výrobcem uváděn proudový odběr menší jak $60 \mu\text{A}$. [8]

4.5 Systém kontroly vstupu

Díky možnostem Arduina se nám otevírá nepřeborné množství variací pro jeho využití. Jednou z nich může být jeho použití pro systémy kontroly vstupu. V této části práce budou přiblíženy jedny z mála možností, jak lze pro tyto systémy Arduino využít.

4.5.1 Displej

Je mnoho přijatelných řešení, jak zobrazit informace z Arduina. Mimo základní výpis do sériového monitoru ve vývojovém prostředí, který vyžaduje připojené PC, je možné informace zobrazit přímo na displeji připojeného k zařízení. Jak již bylo zmíněno, programování bývá složitou záležitostí a obsadí mnoho pinů Arduina. Například displej LCD 1602 (obr. 31), využitý při této práci, vyžaduje v základním připojení 6 datových pinů Arduina. Toto řešení je velmi nepraktické, a proto byl vyvinut převodník (Inter-Integrated Circuit – I²C), který piny LCD displeje převede na pouhé dva. Konkrétně na SDA – datový kanál a hodinový signál – SCL. [1] [2] [39]



Obrázek 31 LCD 1602 s I²C převodníkem

Pro použití displeje s I²C převodníkem, je nutno implementovat externí knihovnu do našeho programu. Pokud neznáme adresu zařízení, je možné si v diskuzích dohledat jednoduchý prográmeček I2C Scanner pro zjištění této adresy. [2]

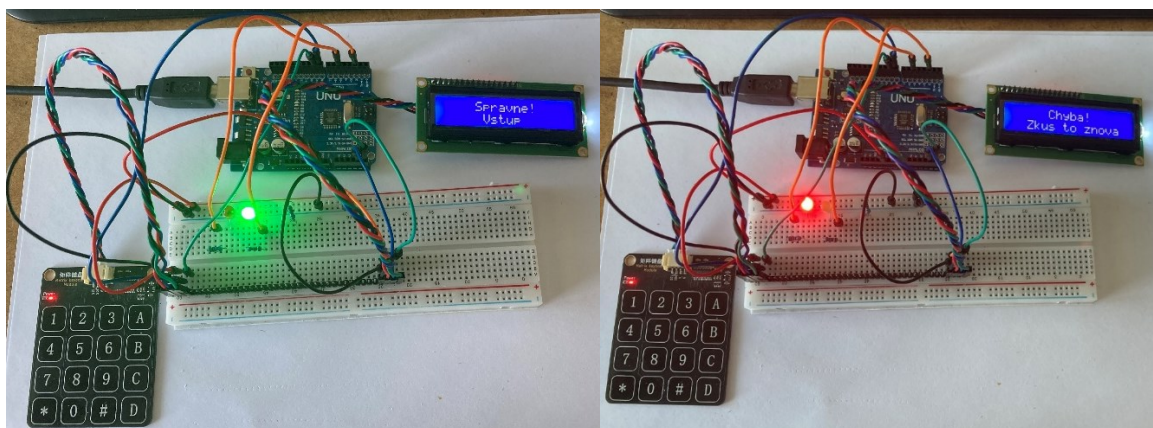
Tabulka 3 Výsledky měření proudového odběru – LCD

Měřená sestava	S podsvícením	Bez podsvícení
LCD displej	23 mA	4 mA
Celkový odběr	51 mA	32 mA

Výsledky měření proudového odběru displeje jsou zobrazeny výše (tab. 3). Měření bylo provedeno pouze s připojeným displejem k samotnému zařízení. Odběr LCD displeje byl pak vypočten odečtením odběru Arduina (28 mA) od celkového odběru. Měřeny byly dva nejčastější stavy displeje, a to s aktivovaným podsvícením a bez spuštěného podsvícení.

4.5.2 Klávesnice a systém kontroly vstupu pomocí hesla

Jedním z možných řešení u systému kontroly vstupu je přístup pomocí hesla. V této části bude takovýto případ přiblížen. K tomuto testu byla využita kapacitní klávesnice TTP229, která vyžaduje taktéž implementování knihovny do zdrojového kódu. Standardní 4x4 maticová klávesnice vyžaduje připojení 8 pinů a proto, obdobně jako u displeje, je využit I²C převodník. Celé zapojení s ukázkou funkce je na obrázku níže (obr. 32).



Obrázek 32 Systém kontroly vstupu pod heslem (povolen / nepovolen vstup)

Výchozí stav sestavení je zhasnutý displej a reakce klávesnice pouze na stisk symbolu „*“. Po stisku je uživatel vyzván k zadání hesla. Zadávané znaky jsou na LCD displeji postupně maskované. Pro potvrzení hesla je zapotřebí stisknutí znaku „#“, následně dojde k vyhodnocení. Při shodě s předdefinovaným heslem je uživatel informován o možném vstupu a systém je uveden do výchozího stavu. Při neshodě je uživatel vyzván k opakovanému zadání hesla

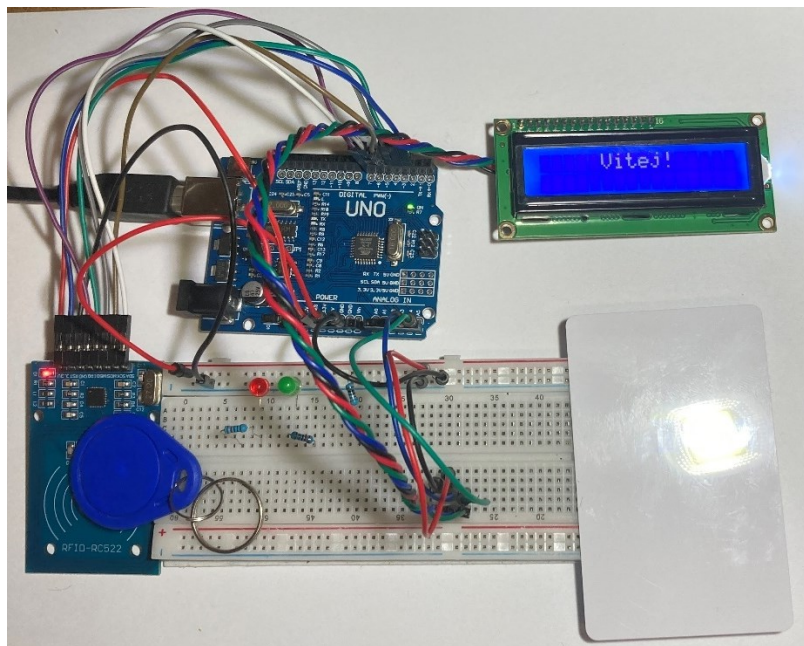
a je nutné opět začít stiskem „*“. Hlavním problémem při tvorbě programu je často chybná funkce v dostupných online řešeních. Například je možné zadávat heslo bez stisku „*“. Nebo pokud je heslo zadáno delší, ale počáteční znaky se shodují, je zadání vyhodnoceno jako správné. Vyskytly se i další problémy, které byly odstraněny. Vysvětlený a okomentovaný zdrojový kód tohoto řešení je pak v příloze PIV. Toto řešení bude dále rozvíjeno v další kapitole a klávesnice bude hlavním ovládacím prvkem celého projektu.

Pozorováním byl zjištěn opět malý odběr kapacitní klávesnice způsobený spíše diodou signalizující zapnutí. Prodejce také uvádí proudový odběr v rozmezí 2,9 – 9 μA . [8]

4.5.3 RFID

System kontrolly vstupu pomocí hesla je pohodlným prvkem ovládání (například vrat), ale pokud je dlouhodobě používán jen jeden kód, je často z opotřebování klávesnice po delším používání patrné, o jaký kód jde. Také hrozí, že tento kód může být snadno odpozorovatelný. Kromě časté změny pinu je možné zvýšit zabezpečení přidáním dalšího autentizačního prvku. Jednou z možností, již dříve zmíněnou, může být doplnění systému o RFID čtečku. Samozřejmě nemusí jít jenom o doplnění jiné technologie, ale lze ji využít i samostatně, jak bude přiblíženo dále.

K testování byla využita čtečka RFID MFRC-522 (obr. 33), která pracuje na kmitočtu 13,56 MHz. Čtečka vyšle impuls do transpondéru, ve formě přívěsku nebo karty, které obsahují čip a anténu, takzvaný RFID tag. Zde se přijatou energií nabije kondenzátor a naladí se na stejnou frekvenci. Anténa přijímače pak přijme 128 bitů dat, která posílá dále ke zpracování do Arduina. RFID čtečka, jako jedna z mála, z nepřeberného množství připojitelných zařízení, vyžaduje nižší napájení v rozmezí 2,5 – 3,3 V, aby nedošlo k přehřátí a následnému zničení zařízení. Arduino má jeden z napájecích výstupů 3,3 V, ke kterému je čtečku nutné připojit. [2] [8]



Obrázek 33 Ukázka RFID čtečky

MFRC-522 využívá pro připojení k Arduinu osm pinů, z čehož dva pro napájení a šest datových, z toho jeden datový pro přerušení (Interrupt ReQuest - IRQ) není nutné zapojovat. Načtený RFID tag posílaný do Arduina je určen k dalšímu zpracování v programovém kódu, pro naše testování uveden v příloze PV. Pro lepší přehlednost je v kódu pracováno s tagem v hexadecimálním tvaru. Nejprve je nutné zjistit tento tag a uložit, či opsat si jej pro další činnosti. Dalším příložením je pak možné data s uloženými porovnávat a na základě vyhodnocení povolit či zamítnout aktivitu například vstup. [2] [8]

Čtečka pracuje v rádech desítek milisekund, ale námi použité zařízení k měření proudového odběru má rozlišení 500 ms. Není tedy možné s dostupnými prostředky přesněji určit proudový odběr a je lepší vycházet z hodnot uváděných výrobcem, viz tabulka (tab. 4). [34]

Tabulka 4 Proudový odběr RFID MFRC-522 [8]

Stav	Proudový odběr	Jednotka
Pracovní proud	13-26	mA
Klidový proud	10-13	mA
Spací proud	<80	μA
Špičkový proud	<30	mA

Při pokusu o měření USB Testerem, byl odběr proudu celého projektu s rozsvíceným displejem 76 mA. Je-li na základě dřívějších měření odečten odběr Arduina, který je 28 mA a displeje 51 mA, vychází pak spotřeba samotného RFID modulu na 25 mA. Při používání a testování delším jak 10 minut byla podle dříve uvedených vzorců vypočtena hodnota okolo 20 mA, což by zhruba odpovídalo střední hodnotě pro pracovní proud uvedený výrobcem (tab. 4).

4.6 Shrnutí

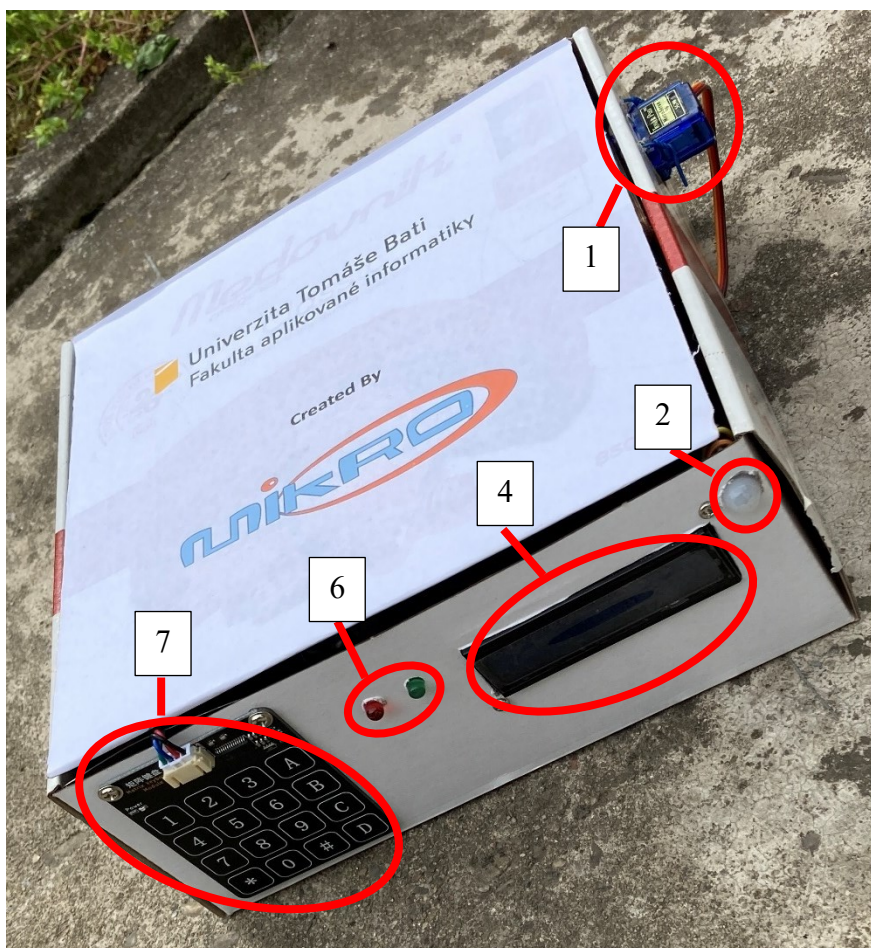
V této kapitole byla navržena některá možná řešení využití Arduina v bezpečnosti. Na trhu je velké množství senzorů, které otevírají rozsáhlé možnosti použití nejenom v bezpečnosti, ale zejména v robotice. Dalším studiem a pokusy je možné vytvořit nové aplikace. Je také možné použít výše uvedené testy pro jiný účel, než bylo zmíněno.

5 PROJEKT „KRABICE“

V této kapitole bude popsána vyrobená „krabice“, do které jsou zakomponovány téměř všechny předešlé prvky. Tento projekt může být zmenšenou verzí případného využití pro domácnost, kancelář, nebo také jako zabezpečená vitrína a podobně. Postupně zde budou rozebrány části z programového kódu, znázorněno zapojení jednotlivých prvků k Arduino a sepsán krátký návod k použití. Zaměření bude hlavně na využití pro SKV, kde nalézá Arduino velkého uplatnění.

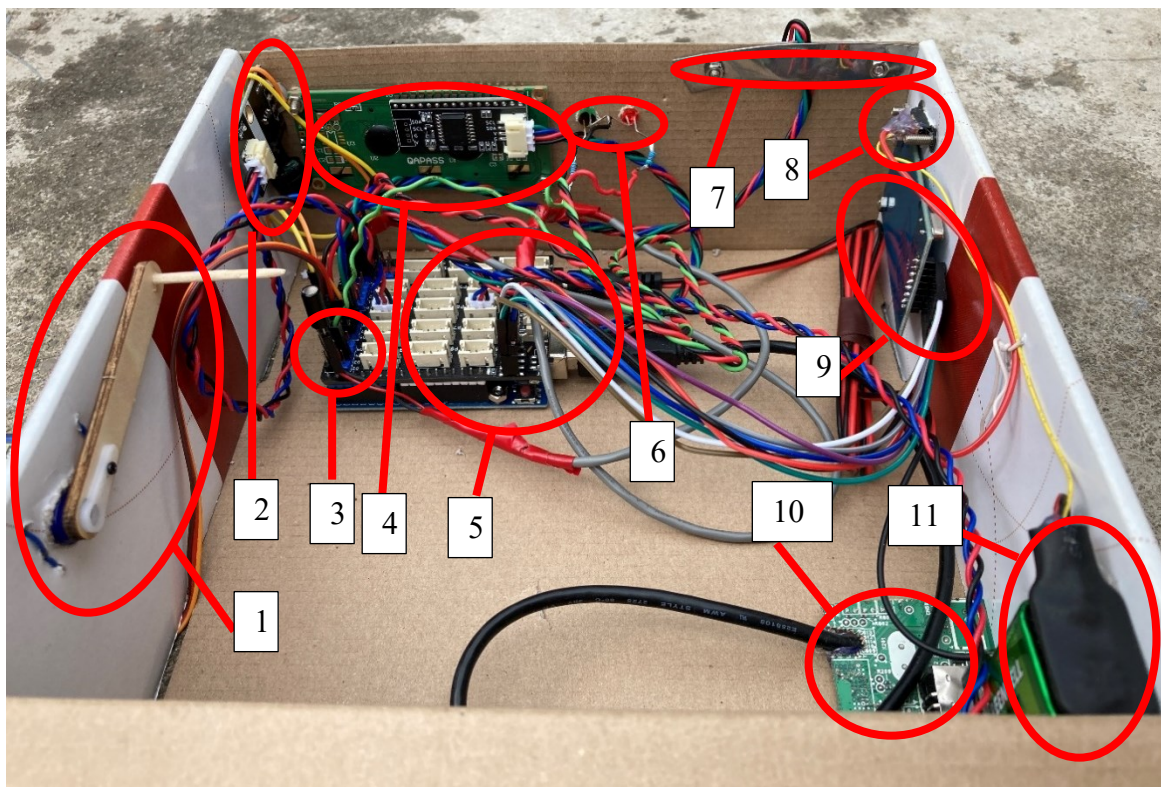
5.1 Výroba ukázkového modelu

Pro projekt byla zvolena krabice od medovníku, která byla dostatečně velká pro všechny komponenty. Výhodou je, že je vyrobena z pevného kartonu, který nezpůsoboval deformace, ale zároveň umožnil lehkou úpravu za pomoci nože a vrtáků.



Obrázek 34 Vnější pohled

Z vnějšího pohledu (obr. 34) je patrné, že byl vyřezán otvor pro displej, vyvrtány díry pro LED, PIR, servomotor a pro uchycovací šroubky. Z levé části je pak zabudován vypínač pro vypnutí/zapnutí „krabice“ od přívodu napětí z 9 V baterie a konektor pro připojení programovacího kabelu USB-B, tyto komponenty nejsou na prvním obrázku vidět.



Obrázek 35 Vnitřní pohled

Z vnitřního pohledu (obr. 35) je pak patrné rozmístění jednotlivých komponent v projektu. Na obou obrázcích jsou komponenty označeny a očíslovány a v tabulce (tab. 5) popsány.

Tabulka 5 Popis komponent projektu (obr. 34 a obr. 35)

P. č.	Komponenta	Popis
1	Servomotor	Servomotor a prodloužené rameno pro otevření krabice
2	PIR + bzučák	PIR detektor pro snímání pohybu v okolí „krabice“ a špatně viditelný bzučák pro akustickou signalizaci.
3	Otřesové čidlo	Otřesové čidlo pro detekci manipulace s „krabicí“
4	LCD displej	LCD displej - výstup pro komunikaci s uživatelem

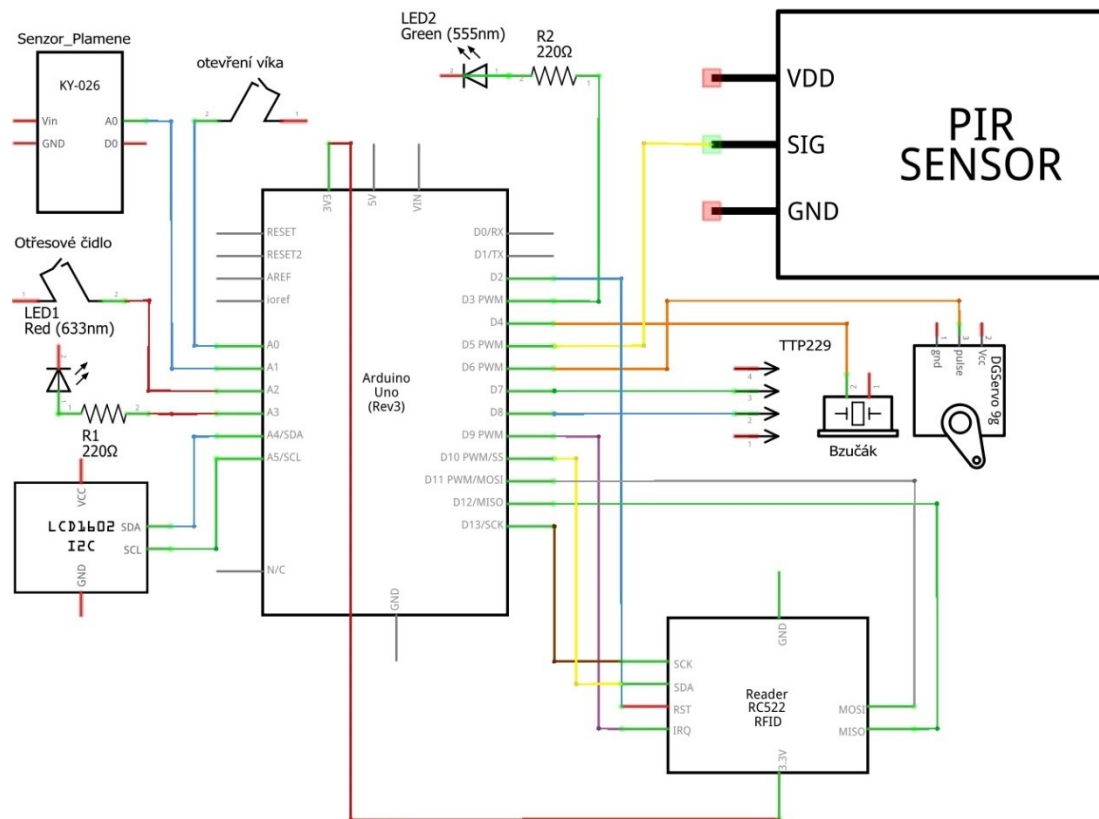
P. č.	Komponenta	Popis
5	Arduino	Základní řídicí jádro projektu – klon Arduino UNO doplněn o Shield - Motor Expansion Board (výrobce Okidea)
6	LED	Červená a zelená LED pro optickou signalizaci
7	Klávesnice	Klávesnice - vstup pro komunikaci s uživatelem
8	Přepínač	Vypnutí/zapnutí zařízení od napájení
9	RFID	RFID čtečka pro autentizaci a autorizaci uživatele
10	USB-B	Prodloužení programovacího připojení k Arduino
11	Napájení	9 V baterie pro zabezpečení napájení projektu

Ještě na obrázcích není vyobrazen detektor plamene, který je umístěn na zadní stěně vedle 9 V baterie. Dále chybí čidlo otevření víka, které bylo přidáno v pozdější fázi projektu. Také je patrná špatná poloha PIR detektoru, který by z principu funkce měl být ve vodorovné ose o 90° otočen. Pro potřeby projektu, kdy slouží převážně k aktivaci podsvícení displeje, bylo zvoleno spíše designové řešení umístění.

5.2 Schéma zapojení

Ke grafickému znázornění zapojení celého projektu bylo využito softwaru Fritzing. Zapojení jednotlivých komponent k Arduino – montážní desce, je uvedeno v příloze PVI. V tomto zobrazení bylo využito nepájivé pole pro propojení kladných a záporných vodičů všech zařízení. V hotovém projektu bylo nepájivé pole nahrazeno rozšiřujícím Shieldem Motor Expansion Board. Níže uvedené blokové schéma zapojení (obr. 36), je zobrazeno bez napájecích vodičů, pro lepší orientaci ve schématu. U uvedeného zapojení je kladen důraz na vyobrazení připojení komponent k jednotlivým pinům Arduina, které bylo využíváno při programování projektu. Ze schématu je patrné, že pro „krabici“ bylo použito téměř všech pinů Arduina. Při zapojení celé sestavy docházelo k nadměrnému pohasínání displeje při dosažení limitu napájení Arduina. V předchozí kapitole byl u detektoru hořlavých plynů naměřen proudový odběr okolo 130 mA. Z těchto důvodů byl z projektu detektor hořlavých plynů vyjmut a uvolnil se tak jeden analogový pin A0. Na tento volný pin bylo připojeno další

otřesové čidlo, o které byl projekt rozšířen. Jeho funkcí je hlídat otevření víka mimo dovolený stav.



fritzing

Obrázek 36 Schéma zapojení bez napájení a uzemnění [12]

Jak bylo v předchozí kapitole zmíněno, je nutné na rozdíl od ostatních komponent napájet RFID čtečku napětím 3,3 V, proto je toto napájení ve schématu znázorněno. U obou zapojení, jak v příloze uvedeného, tak u výše zobrazeného byla snaha dodržet barevné značení vodičů odpovídající barvám v „krabici“.

5.3 Zdrojový kód ukázkového modelu

Účelem této části bakalářské práce není popisovat základní struktury programu, ke kterým je spousta literatury v tištěné podobě a zejména na internetu. Proto zde budou rozebrány jen důležité části programového kódu. Budou nastíněna použitá vlastní řešení a vysvětleny složitější postupy v programovém kódu.

Hlavní funkcí programu je smyčka „loop()“. Pro efektivní funkci zařízení je žádoucí, tuto smyčku „přerušit“ další smyčkou, k čemuž lze využít příkazu „while“. V případě tohoto řešení, je nutné vytvořit globální proměnnou, která bude mimo základní smyčku, aby bylo možné s její pomocí řídit tok programu. K tomu stačí využití 8 bitového datového typu. V našem příkladu je využit „boolean“, který nabývá logických hodnot „true(1)/false(0)“ a je přiřazen k funkci, která tyto hodnoty vrací, podle požadavku na pokračování v běhu programu. Příkladem je funkce „vstup_pod_heslem()“, která posláním návratové hodnoty „false“ vyčkává na zadání celého hesla uživatelem. Po zadání hesla vrátí hodnotu „true“ a program pokračuje dále v základní smyčce.

Kód 1 Řízení běhu programu

```
void loop() {
...
  while (nacteno == false) {
    nacteno = vstup_pod_heslem();
    trvale();
  }
}
...
boolean vstup_pod_heslem() {
  if (podmínka) {
return false;
} else {
return true;
}
}
```

Při práci s Arduinem je velmi časté využívání funkce „delay()“ například pro přečtení informací na displeji. Při použití tohoto příkazu zařízení nereaguje na žádné vstupy a jeho využití pro bezpečnostní aplikace je tedy nevhodné. Pro řešení tohoto problému byla vymyšlena dvě možná řešení:

Kód 2 Vlastní prodleva

```
void my_delay(float i) {
  i = nasobek * i;
  for (float t = 0; t < i; t++) {
    trvale();
    standby = 0;
  }
}

... void my_delay(float i) {
...   for (float j = 0; j < i; j++) {
...     trvale();
...     delay(1);
...   }
... }
```

Součástí obou řešení je odkaz na funkci „trvale()“, která sdružuje jiné funkce, které je zapotřebí neustále hlídat. V našem případě to zahrnuje otřesové čidlo pro detekci manipulace se zařízením, hlídání podsvícení displeje a snímač plamene pro sledování požáru. Druhé řešení je vhodné spíše pro kratší pauzy, protože požadovanou hodnotu pauzy zvětšuje o dobu zpracování funkce „trvale()“. Využito bylo tedy první funkce, kdy přes sledování sériové linky se vstupní hodnotou 100 000 byla odečtena reálná doba trvání a prostým dělením byl pak vypočten násobek vstupní hodnoty tak, aby zhruba odpovídal jednotce milisekundy. Bohužel nejde o přesné hodnoty a s každou změnou funkce „trvale()“ je nutné provést nové měření a násobek aktualizovat. Ideální je také mít v trvalé smyčce kód co nejvíce optimalizovaný a jednoduchý, aby k velkým rozdílům nedocházelo.

Příkladem je kód pro sledování doby nečinnosti, který měl před optimalizací velký vliv na délku pauzy. Po zdlouhavém testování a ladění je ve stavu, kdy má na chod minimální účinek, ale stále funguje dle požadavku. To znamená, že i když není podsvícení aktivní, tak se při detekování pohybu zapne. Zároveň pokud je kdekoliv při používání zařízení překročena doba nečinnosti, je podsvícení vypnuto. Aktivace proběhne opět pohybem nebo stiskem klávesy, která vyvolá libovolnou akci. V celém projektu všechny uživatelem vyvolatelné akce obsahují funkci „my_delay()“ nebo „reset_lcd()“, ve kterých je implementováno vynulování počítadla pro nečinnost. Pro optimálnější chod bylo zakomponováno před přetečením použité proměnné „unsigned int“, tedy dosažení hodnoty 65535, nastavení hodnoty na limit pro nečinnost zvětšené o jedna. Tím nedochází k opětovnému zavolání funkce pro zhasnutí displeje. Použité řešení je uvedeno níže. [2]

Kód 3 Sledování činnosti kvůli podsvícení LCD po optimalizaci

```
...
  if (standby == necinnost) {
    podsviceni(0);
  } else if (standby > necinnost) {
    if (digitalRead(PIR_D) == 1) {
      podsviceni(1);
    }
  }
  standby++;
  if (standby == 65535) {
    standby = necinnost + 1;
  }
...

```


Dalším zajímavým řešením je „zpomalení“ servomotoru. Jde o kombinaci cyklů, které lze využít pro prodloužení trvání požadované činnosti. Dá se aplikovat nejenom pro ovládání servomotoru, ale u všech funkcí volaných s parametrem u kterých je možné tento parametr dále dělit. Servomotor se otáčí o zadaný počet stupňů. Pro zpomalení je požadovaný úhel cyklem rozdělen na jednotky a po každém pootočení vložena pauza. Výsledná rychlost je pak ve stupních za čas. V uvedeném příkladu je také řešen směr otáčení.

Kód 4 Prodloužení funkce

```
void servo_stupne(int uhel) {
    int u = servo6.read();
    for (u; u <= uhel; u++) {
        servo6.write(u);
        my_delay(speed);
    }
    for (u; u >= uhel; u--) {
        servo6.write(u);
        my_delay(speed);
    }
}
```

Dále bylo zapotřebí řešit neustálé zhasínání LED při použití vestavěné funkce „tone()“, která z principu vnitřní funkce narušovala plynulý chod PWM, které je využito pro ovládání intenzity jasu zelené LED. Byla proto vytvořena nová funkce, která nemá tak stabilní frekvenci tónu jako vestavěná, ale pro účely upozornění, či spuštění sirény plně dostačuje a nenarušuje chod PWM.

V kódu (5) pro vlastní vytvoření zvuku je již zakomponováno hlídání, zda je zapnuté akustické upozornění projektu. Vypnutí zvuku bylo umístěno do této části, protože například vyvolání tísňe spoléhá pro zobrazení informací na dobu trvání výstrahy a tímto umístěním se čas nezmění, pouze se nerozkmitá piezoelektrický krystal v „bzučáku“.

Kód 5 Funkce „my_tone()“ [26]

```
void my_tone(float frekvence, float doba) {
    float pul_perioda = 500000 / frekvence;
    for (int i = 0; i < doba; i++) {
        if (bzucak == true) {
            digitalWrite(pip_D, HIGH);
        }
        delayMicroseconds(pul_perioda);
        digitalWrite(pip_D, LOW);
        delayMicroseconds(pul_perioda);
    }
}
```

V celém projektu je více řešení, které již ale využívají základní programovací syntaxi, a jde zejména o nalezení a logické vyřešení těchto problémů. Například ošetření, aby klávesnice reagovala pouze na předdefinované funkce a nebylo tak možné napsat heslo mimo zobrazený stav. Dále nutnost vymazat vložené heslo, které slouží pro porovnání, aby nedocházelo v určitých případech k opětovnému povolení vstupu bez vložení hesla. Přepínání mezi funkcemi a ošetření „zamrznutí“ zařízení při chybových hlášení. Zacyklení programu při stisku tlačítka, které nemá přidělenou funkci. A další řešení, vyvstávající při komunikaci s uživatelem.

Bylo uvedeno pár zajímavých řešení z programového kódu celého projektu. Mnoho dalších a praktické použití uvedených příkladů naleznete ve zdrojovém kódu v příloze PVII na konci této bakalářské práce. Kód je co nejvíce komentován pro orientaci případného „bastlíře“ (nadšence programování mikrokontrolérů).

5.4 Návod

Na levé části se nachází přepínač pro ovládání napájení zařízení, je označen slovem „Power“. Posunutím od sebe jej lze zapnout. Napájení zařízení je signalizováno zelenou LED vedle displeje a červenou LED na klávesnici. Na rozsvíceném displeji se zobrazí „Vítej! – Vyber funkci“.

Při nečinnosti dojde zhruba po půl minutě k vypnutí podsvícení. K opětovné aktivaci dojde pomocí pohybu, či stisku tlačítka, kterému je přiřazena akce (viz. dále). Ovládání podsvícení není vázáno pouze na výchozí obrazovku, může tak při nečinnosti dojít k jeho vypnutí kdykoliv.

V kterékoliv fázi práce s projektem může být také vyvolán poplach v případě detekce plamene senzorem umístěným na zadní stěně uvnitř modelu. Navíc, z důvodu aby nedocházelo k častým poplachům, je pouze červenou LED signalizován otřes, tedy manipulace se zařízením. Toto čidlo má simulovat sabotáž na zařízení. Zkoušku lze provést poklepáním shora na pravou boční stěnu „krabice“. Taktéž je po celou dobu hlídáno, zda se do zařízení nesnaží vniknout narušitel neoprávněným otevřením víka.

Víko lze otevřít dvěma způsoby. Prvním je vstup pomocí hesla (1 2 3), které je možné zadat po stisku hvězdičky – „*“. Zadání je nutné potvrdit křížkem – „#“. Při zadání špatného, nebo moc dlouhého hesla je uživatel na danou chybu upozorněn a zařízení se uvede do výchozího stavu. Je tedy nutné pro opakované zadání začít opět stiskem hvězdičky, nebo si vybrat jinou akci, například druhou možnost otevření „krabice“.

Druhý způsob otevření je pomocí RFID čtečky. K tomu abychom mohli otevřít víko pomocí RFID čtečky, je nutné nejprve registrovat nového uživatele. Po stisku tlačítka „B“ je uživatel vyzván k zadání administrátorského hesla (3 2 1), po zadání a potvrzení křížkem je zobrazena výzva k přiložení čipu. Kartu nebo přívěšek je potřeba přiložit z levé boční strany pod přepínač sloužící k zapnutí zařízení. Pokud vše proběhne v pořádku, je zobrazena informace o vytvoření uživatele a vypsán jeho identifikační kód. Pro následné otevření víka je zapotřebí stisknout „A“ a po přiložení čipu postupovat stejně jako při vstupu za pomoci hesla popsaného výše.

Dále je možné pomocí písmene „C“ provést zastřežení systému. Po jeho stisku, pokud nejsou narušeny vstupy (otevřené víko), začne odpočet pro opuštění zastřežované oblasti. Po zobrazení „Zastřeženo“ je při detekci pohybu PIR senzorem vyvolán poplach.

Pro odstřežení použijeme tlačítko „D“, které mimo jiné slouží k uvedení projektu do výchozího stavu.

Při zapnutí zařízení jsou deaktivována zvuková upozornění, včetně upozornění na poplarchy. Akustická oznámení zapneme, potažmo vypneme stiskem nuly – „0“.

Níže v tabulce (tab. 6) je popsáno, jaká funkce je přiřazena danému tlačítku. Výchozí heslo pro funkce, kde je vyžadováno, je uvedeno v závorce. Tato legenda je umístěna také na „krabici“ mezi LCD a klávesnicí.

Tabulka 6 Popis ovládacích prvků

Klávesa	Popis
5	Tísňové tlačítko
*	SKV – vstup pomocí hesla (123)
#	Potvrzení zadaného heslo
A	SKV – RFID, aktivace vstupu
B	RFID – registrace nové karty (321)
C	Zastřežení systému
D	Odstřežení systému a výchozí stav
0	Zapnutí/vypnutí zvuku

Projekt byl dlouhodobě testován, není tedy důvod předpokládat, že by neměl fungovat. Nastane-li ale situace, kdy by došlo k zacyklení, nereagování, či jinému nestandardnímu chování, je zapotřebí jej pomocí vypínače na boční straně vypnout a opět zapnout – provést restart. Výše uvedený návod lze najít ve zkrácené formě pro účely prezentace v příloze PVIII.

5.5 Spotřeba

Během provozování „krabice“ bylo odečteno několik údajů o spotřebě elektrické energie (E). Poté byla podle předchozích vzorců vypočtena průměrná hodnota výkonu (P) a proudového odběru (I) v uvedeném čase. První dva údaje uvedené v tabulce (tab. 7) byly odečteny za umělých (ideálních) podmínek. To znamená zakrytí PIR detektoru, aby nedocházelo k aktivaci podsvícení. Nejprve byl systém zastřežen na jednu hodinu, poté stejné měření proběhlo v odstřeženém stavu. Po těchto testech byla „krabice“ nasměrována na osobu sedící u PC a opět provedeno dvojí měření. Při zastřeženém režimu tak docházelo velmi často k vyvolání alarmu na základě pohybu. Následně se i rozsvítil displej. Při odstřeženém režimu docházelo pouze k aktivaci displeje. Každých dvacet minut (tedy čtyřikrát) bylo otevřeno a zavřeno víko za pomoci RFID čtečky a zadání hesla. Ještě při otevřené „krabici“ byl zapalovačem vyvolán poplach na detektoru plamene. Vše s aktivním akustickým oznámením.

Další měření (5. – 7. v tab. 7) bylo provedeno při používání zařízení. Převažující druh provozu je pak upřesněn ve sloupci „Stav“. Protože se jednalo o kontinuální měření, kdy byl čas vynulován a po uvedené době odečteny hodnoty, tak poslední dvě měření byla vypočtena odečtením měření 5. – 7 (tab. 7).

Tabulka 7 Údaje o spotřebě celého projektu

P. č.	Čas (hh:mm:ss)	Energie E (Wh)	Výkon P (mW)	Proud I (mA)	Stav
1.	1:0:0	0,356	356	71,2 (76)	Zastřeženo
2.	1:0:0	0,355	355	71 (78)	Odstřeženo
3.	1:0:0	0,467	467	93,4 (93)	Zastřeženo s poplarchy
4.	1:0:0	0,416	416	83,2 (95)	Odstřeženo s častým podsvícením displeje
5.	5:50:10	2,249	385,4	77,08	Klid
6.	19:35:57	8,169	416,8	83,36	Převážně klid
7.	38:20:15	16,349	490,4	98,08	Ladění a provoz + předešlé
8.	13:45:47	5,92	430,1	86,02	Převážně s podsvícením displeje
9.	18:44:18	8,18	436,5	87,3	Ladění + klid přes noc

Před samotným měřením bylo ještě v některých stavech provedeno aktuální odečtení hodnoty, tento údaj je uveden ve sloupci „Proud“ v závorce. Podle stavu, ve kterém se „krabice“ nacházela při odečtu, je vidět rozdíl mezi vypočtenými údaji. Samozřejmě hodnoty získané na základě delšího měření by měly být relevantnější. Celkově se dá říct, že proudový odběr celé sestavy je pod 100 mA, což je na počet připojených senzorů, způsob vytvoření projektu a ne zcela ideálně odladěný kód poměrně nízká hodnota. Kdyby byl kladen důraz na spotřebu, bylo by nutné doladit kód a mechanicky odpojit diody na senzorech. V případě odpojení diod by došlo k úspoře v řádu jednotek miliampér, na základě předchozích měření a počtu diod zhruba o 5 – 10 mA.

5.6 Shrnutí

Programování Arduina je poměrně zábavnou činností. Nutí k přemýšlení, hledání nových řešení, a tak k sebevzdělávání. Dobře fungující program, nejenom v případě Arduina, obnáší často zdoluhavé vytváření, testování a ladění v délce mnoha hodin. Laik je pak považuje za samozřejmá řešení, která přece nemohla dát tolik práce. S „krabicí“ si například hrál syn, který objevil ne jeden problém, který při programování nebyl zřejmý. V době odevzdání této bakalářské práce byly všechny známé problémy vyřešeny, ale to neznamená, že tam již další nejsou. Případné praktické využití by obnášelo dlouhé testování a revizi od jiných programátorů.

ZÁVĚR

Hlavním cílem této bakalářské práce bylo zhodnotit možnosti nasazení mikrokontrolérů typu Arduino v bezpečnosti. Neméně důležitou součástí je pak návrh a realizace modelu bezpečnostního zařízení, který je založen na platformě Arduino.

Softwarových a hardwarových řešení týkajících se Arduina existuje mnoho. Pro přiblížení bylo popsáno několik základních možností. V teoretické části byla rovněž rozebrána problematika bezpečnostních technologií, která byla s předešlou kapitolou v další části práce propojena. Z velkého množství senzorů bylo vybráno několik vhodných pro aplikace v oblasti bezpečnosti. Dalším průzkumem trhu je možné vytvořit spoustu variací jiného uplatnění.

V praktické části práce bylo vybráno pár senzorů využitelných v oblasti bezpečnosti, které byly na výběr z portfolia pořízených sad obou klonů Arduina. Tyto senzory byly blíže popsány, bylo provedeno samostatné zapojení a pořízena fotodokumentace. Téměř všechny zmíněné komponenty byly zakomponovány do jednoho projektu. Tento prototyp dostal pracovní název „krabice“. U uvedených zapojení, včetně výsledného projektu, je také změřena spotřeba elektrické energie. Proběhlo i testování funkčnosti prototypu, jež pomohlo ověřit spolehlivost systému. Zdrojové kódy jsou uvedeny v přílohách na konci práce a byly více či méně inspirovány z uvedených zdrojů. Pro kód celého projektu, uvedený v příloze PVII, již v mnoha případech zdrojová řešení nevyhovovala a došlo často k zásadním změnám.

Dle provedeného průzkumu trhu byla výsledná cena za komponenty použité v projektu odhadnuta na 700,- Kč. Pokud by byla započtena cena za naprogramování, již by se pořízení neoplatilo, i kdyby bylo počítáno s částkou 150,- Kč za hodinu práce. Proto by závěrečný projekt mohl sloužit spíše pro studijní zaměření, jako je programování mikrokontrolérů, nebo ukázka funkce některých bezpečnostních prvků při prezentacích školních projektů.

Informace zpracované v této práci mohou být přínosem pro širší veřejnost. Bezpečnostní systém založený na platformě Arduino by mohl nalézt uplatnění pro soukromé využití nebo aplikace, kde by cena pořízených certifikovaných prostředků, převýšila hodnotu chráněných aktiv. Dalším využitím by mohla být integrace do zařízení, které nedisponují detekcí sabotáže. Pokud by byl vyměněn servomotor za spínané relé, může mít potenciál využití pro otevírání zámku dveří či jiného využití v elektronickém systému kontroly vstupu.

Rozsah a pojetí práce bylo pojato poměrně ze široka. To mělo za následek, že jak v teorii, tak v praxi se dané téma probralo pouze povrchně. Aneb když si pořídíte produkt pět v jednom, nikdy nebude dosahovat kvalit pěti samostatných zařízení. Proto by bylo vhodné v projektu nadále pokračovat a směřovat jej na konkrétní aplikaci elektronického systému kontroly vstupu. Mohlo by být využito připojení k serveru, vytvoření databáze uživatelů a například zavedení možnosti ovládání přes mobilní telefon. Těmito úpravami by se vytvořilo komplexní zabezpečení, které by mohlo být využito pro téma další kvalifikační práce.

SEZNAM POUŽITÉ LITERATURY

- [1] VODA, Zbyšek. *Průvodce světem Arduina*. Vydání druhé. Bučovice : Martin Stríž, 2017. ISBN 978-80-87106-93-8.
- [2] SELECKÝ, Matúš. *Arduino: uživatelská příručka*. Přeložil Martin HERODEK. Brno : Computer Press, 2016. ISBN 978-80-251-4840-2.
- [3] HORÁK, Jaroslav. *Hardware: učebnice pro pokročilé*. Vydání druhé. Praha : Computer Press, 1998. ISBN 80-7226-122-3.
- [4] *Arduino - Home*. [Online] Arduino © 2023 [cit. 2023-04-25]. Dostupné z: <https://www.arduino.cc/>.
- [5] 8-Bit AVR® Core - Developer Help. *Home - Developer Help*. [Online] Microchip Technology, Inc. © 2021 [cit. 2023-04-25]. Dostupné z: <https://microchipdeveloper.com/8avr:avrcore>.
- [6] *Arduino Oficial Store: Boards Shields Kits Accessories*. [Online] Arduino S.r.l. - Partita IVA 09755110963. (c) 2021 [cit. 2023-04-25]. Dostupné z: <https://store.arduino.cc/>.
- [7] Arduino Uno rev3. *LaskaKit.cz: by Makers for Makers*. [Online] LaskaKit, © 2023 [cit. 2023-04-25]. Dostupné z: <https://www.laskakit.cz/arduino-uno-rev3--original/>.
- [8] *LaskaKit.cz: by Makers for Makers*. [Online] LaskaKit, © 2023 [cit. 2023-04-25]. Dostupné z: <https://www.laskakit.cz/>.
- [9] GitHub: Let's build from here. *emakefun-Ble-uno/BLE-Uno Operation Manual V.1.0.pdf at master · emakefun/emakefun-Ble-uno · GitHub*. [Online] GitHub, Inc. © 2023 [cit. 2023-04-25]. Dostupné z: <https://github.com/emakefun/emakefun-Ble-uno/blob/master/BLE-Uno%20Operation%20Manual%20V.1.0.pdf>.
- [10] Arduino Motor Shield Rev3. *Arduino Oficial Store*. [Online] Arduino S.r.l. - Partita IVA 09755110963. (c) 2021 [cit. 2023-04-25]. Dostupné z: <https://store.arduino.cc/products/arduino-motor-shield-rev3>.
- [11] Tinkercad: From mind to design in minutes. *Circuit design Smart Home's Kitchen*. [Online] Autodesk, Inc. © 2023 [cit. 2023-04-26]. Dostupné z: <https://www.tinkercad.com/things/7DPNu1dIAQ3-smart-homes-kitchen>.
- [12] Welcome to Fritzing. *Download Fritzing*. [Online] Fritzing GmbH, [cit. 2023-04-26]. Dostupné z: <https://fritzing.org/download/>.

- [13] GitHub: Let's build from here. *Fritzing · GitHub*. [Online] GitHub, Inc. © 2023 [cit. 2023-04-26]. Dostupné z: <https://github.com/fritzing>.
- [14] LUKÁŠ, Luděk. *Bezpečnostní technologie, systémy a management*. Zlín : Radim Bačuvčík - VeRBuM, 2015. ISBN 978-80-87500-05-7.
- [15] VALOUCH, Jan. *Projektování bezpečnostních systémů*. [Online] Vydání druhé, Zlín : Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, 2019 [cit. 2023-04-26]. ISBN 978-80-7454-858-1. Dostupné z: <https://digilib.k.utb.cz/handle/10563/45863>.
- [16] DRGA, Rudolf. *Technické prostředky bezpečnostního průmyslu*. Studijní výukový materiál. Zlín : Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, 2013.
- [17] IVANKA, Ján. *Systematizace bezpečnostního průmyslu*. [Online] Vydání páté, Zlín : Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, 2014 [cit. 2023-04-26]. ISBN 978-80-7454-410-1. Dostupné z: <https://digilib.k.utb.cz/handle/10563/27488>.
- [18] SCP ID Card - Customize Yours Now. *SCP Foundation ID Card*. [Online] SCP Foundation ID. © 2023 [cit. 2023-05-15]. Dostupné z: <https://scpidcard.com/product/scp-foundation-id-card>.
- [19] DRGA, Rudolf. *Elektronické bezpečnostní systémy: Přístupové systémy*. Studijní výukový materiál. Zlín : Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, 2013.
- [20] Teplotní čidla, čidla vlhkosti a tlaku. *LaskaKit.cz: by Makers for Makers*. [Online] LaskaKit, © 2023 [cit. 2023-04-27]. Dostupné z: <https://www.laskakit.cz/teplota--vlhkost-a-tlak/>.
- [21] Čidla kvality vzduchu a čidla plynu. *LaskaKit.cz: by Makers for Makers*. [Online] LaskaKit, © 2023 [cit. 2023-04-27]. Dostupné z: <https://www.laskakit.cz/plyn-a-kvalita-ovzdusi/>.
- [22] Amazon.com. *BONATECH 9 in 1 MQ Sensor Modules Kit Project Super Starter Kits for Gas Detection*. [Online] Amazon.com, Inc. © 1996-2023 [cit. 2023-04-27]. Dostupné z: <https://www.amazon.com/BONATECH-Arduino-Modules-Project-Detection/dp/B07D9H74LT>.

- [23] Distributor a obchod online - Transfer Multisort Elektronik. *Snímače plynu | Elektronické díly*. [Online] TME © 2023 [cit. 2023-04-27]. Dostupné z: https://www.tme.eu/cz/katalog/snimace-plynu_100526/.
- [24] Pohybová čidla, vzdálenost, vibrace. *LaskaKit.cz: by Makers for Makers*. [Online] LaskaKit, © 2023 [cit. 2023-04-27]. Dostupné z: <https://www.laskakit.cz/vzdalenost--pohyb-a-vibrace/>.
- [25] Světelná čidla a optika. *LaskaKit.cz: by Makers for Makers*. [Online] LaskaKit, © 2023 [cit. 2023-04-27]. Dostupné z: <https://www.laskakit.cz/svet--barva-a-optika/>.
- [26] Proudová čidla a napětí. *LaskaKit.cz: by Makers for Makers*. [Online] LaskaKit, © 2023 [cit. 2023-04-27]. Dostupné z: <https://www.laskakit.cz/proud-a-napeti/>.
- [27] Čidla na hmotnost. *LaskaKit.cz: by Makers for Makers*. [Online] LaskaKit, © 2023 [cit. 2023-04-27]. Dostupné z: <https://www.laskakit.cz/hmotnost/>.
- [28] Snímání srdečního tepu a biometrická čidla. *LaskaKit.cz: by Makers for Makers*. [Online] LaskaKit, © 2023 [cit. 2023-04-28]. Dostupné z: <https://www.laskakit.cz/biometrika-a-srdecni-aktivita/>.
- [29] Displeje. *LaskaKit.cz: by Makers for Makers*. [Online] LaskaKit, © 2023 [cit. 2023-04-28]. Dostupné z: <https://www.laskakit.cz/displeje/>.
- [30] Zvukové součástky a zdroje zvuku. *LaskaKit.cz: by Makers for Makers*. [Online] LaskaKit, © 2023 [cit. 2023-04-28]. Dostupné z: <https://www.laskakit.cz/zvuk-2/>.
- [31] Přenos a ukládání dat. *LaskaKit.cz: by Makers for Makers*. [Online] LaskaKit, © 2023 [cit. 2023-04-28]. Dostupné z: <https://www.laskakit.cz/prenos-a-ukladani-dat/>.
- [32] Ovládací prvky. *LaskaKit.cz: by Makers for Makers*. [Online] LaskaKit, © 2023 [cit. 2023-04-28]. Dostupné z: <https://www.laskakit.cz/ovladaci-prvky/>.
- [33] Solární panel 5V 4W s držákem na zeď. *LaskaKit.cz: by Makers for Makers*. [Online] LaskaKit, © 2023 [cit. 2023-04-28]. Dostupné z: <https://www.laskakit.cz/solarni-panel-5v-4w-s-drzakem-na-zed/?variantId=10752>.
- [34] Aliexpress.com. *Voltage Current Meter Ammeter Detector: Usb Voltage Current Meter Tester - 13 I Usb*. [Online] AliExpress.com © 2010-2022 [cit. 2023-04-30]. Dostupné z: https://www.aliexpress.com/item/4000000779795.html?spm=a2g0o.order_list.order_list_main.208.38851802edQVvk0.

- [35] Otřesové čidlo SW-520D. *LaskaKit.cz: by Makers for Makers*. [Online] LaskaKit, © 2023 [cit. 2023-04-30]. Dostupné z: <https://www.laskakit.cz/otresove-cidlo-sw-520d/>.
- [36] components101.com. *MQ-4.doc (www.hwsensor.com)*. [Online] Components101 © 2021 [cit. 2023-04-30]. Dostupné z: https://components101.com/sites/default/files/component_datasheet/MQ-4%20Methane%20Gas%20Sensor%20Datasheet.pdf.
- [37] Waveshare.com. *Flame Sensor*. [Online] Shenzhen: Waveshare Electronics. [cit. 2023-04-30]. Dostupné z: <https://www.waveshare.com/flame-sensor.htm>.
- [38] Sunrom Electronics. *Flame Sensor - 5mm IR Diode - YG1006 [5533]*. [Online] Ahmedabad, Gujarat, India: Sunrom Electronics © 2004-2023 [cit. 2023-04-30]. Dostupné z: <https://www.sunrom.com/p/flame-sensor-5mm-ir-diode-yg1006>.
- [39] laskkit_arduino_maxi_starter_kit_v1_4-1. *LaskaKit.cz: by Makers for Makers*. [Online] LaskaKit, © 2023 [cit. 2023-04-30]. Dostupné z: https://www.laskakit.cz/user/related_files/laskkit_arduino_maxi_starter_kit_v1_4-1.pdf.
- [40] EPSTEIN, Jenna. Using an MQ-4 Sensor with Arduino to Detect Methane Gas. In: *SENSING THE CITY – CPLN 571*. [Online] 2022-03-30 SENSING THE CITY Powered by WordPress, © 2023 [cit. 2023-04-29]. Dostupné z: <https://www.sensingthecity.com/configuring-an-mq-4-sensor-to-detect-methane-gas/>.
- [41] savion2047. How to combine a variety of coding in one Arduino code - #21 by savion2047. In: *Arduino Forum*. [Online] 2021-07-10 Arduino © 2020 [cit. 2023-04-29]. Dostupné z: <https://forum.arduino.cc/t/how-to-combine-a-variety-of-coding-in-one-arduino-code/883007/21>.
- [42] VODA, Zbyšek. TinyLab: Piezo bzučák. In: *Úvodní stránka - Bastlírna HWKITCHEN*. [Online] 2017-09-04, HWKITCHEN.CZ, © 2014-2023 [cit. 2023-05-04]. Dostupné z: <https://bastlirna.hwkitchen.cz/tinylab-piezo-bzucak/>.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AC-DC	Alternating Current – Direct Current
E	Energie (jednotka Wh)
EPS	Elektrická Požární Signalizace
GPRS	General Packet Radio Service
GSM	Groupe Spécial Mobile
I&HAS	Intruder and Hold-up Alarm System
IDE	Integrated Development Environment
IoT	Internet of Things
IrDA	InfraRed Data Association
IRQ	Interrupt ReQuest
LCD	Liquid Crystal Display
LED	Light-Emitting Diode
OLED	Organic Ligh-Emitting Diode
P	Výkon (jednotka W)
PC	Personal Computer
PCB	Printed Circuit Board
PIN	Personal Identification Number
PIR	Passive Infra Red
ppm	Parts per million
PWM	Pulse Width Modulation.
PZTS	Poplachové Zabezpečovací a Tísňové Systémy
QR	Quick Response
RFID	Radio Frequency Identification
RISC	Reduced Instruction Set Computer

SKV	System Kontroly Vstupu
TFT	Thin Film Transistor
USB	Universal Serial Bus
UTB	Univerzita Tomáše Bati ve Zlíně
Wi-Fi	Wireless Fidelity

SEZNAM OBRÁZKŮ

Obrázek 1 Deska Arduino Uno [7]	11
Obrázek 2 Klon UNO	13
Obrázek 3 Klon BLE-UNO	14
Obrázek 4 Arduino Motor Shield Rev3 [10]	15
Obrázek 5 Snímač vlhkosti a teploty	16
Obrázek 6 Software Arduino IDE 2.1.0	17
Obrázek 7 Software Fritzing [12]	18
Obrázek 8 Ukázka multifunkční identifikační karty [18].....	24
Obrázek 9 Teplotní čidlo DHT11 a infračervený teploměr MLX90615 [20]	26
Obrázek 10 Čidla plynů [22]	27
Obrázek 11 PIR detektor HC-SR501 a mikrovlnný detektor XYC-WB-D1 [24]	28
Obrázek 12 Měřič vzdálenosti ultrazvukový HC-SR04 a laserový VL53L1X [24]...29	
Obrázek 13 3-osý gyroskop, akcelerometr a magnetometr MPU-9250 SPI/IIC a vibrační snímač SW1801P [24].....	30
Obrázek 14 LilyGO TTGO T-Camera ESP32 s mikrofonom a PIR a čtečka čárových kódu YHDAA YHD-M800D [25]	30
Obrázek 15 Snímač střídavého napětí 230 V a cívka pro měření proudu CT08CL10 [26]	31
Obrázek 16 Vážicí senzor GUANG CE YZC-131 a převodník HX711 [27].....	32
Obrázek 17 Čtečka otisků prstů optická DY50 a kapacitní FPC1020AM [28].....	32
Obrázek 18 LCD displej 2004 a maticový 8x8 LED displej MAX7219 [29]	33
Obrázek 19 Bzučák a stereo reproduktor 5 W [30]	34
Obrázek 20 Modul Wifi s Bluetooth LilyGO a GPRS-GSM SIM900 Shield [31]	34
Obrázek 21 Maticová klávesnice 4x4 a solární panel 5 V 4 W [32] [33].....	35
Obrázek 22 Měření odběru Arduina pomocí USB Testeru U96P	38
Obrázek 23 Měření odběru Arduina napájeného 9 V baterií.....	39
Obrázek 24 Princip funkce otřesového čidla SW-520D a schéma zapojení [35].....	40
Obrázek 25 Ukázka zapojení a funkce otřesového čidla	40
Obrázek 26 Detekční charakteristika čidla MQ-4 [36].....	41
Obrázek 27 Graf analogového výstupu senzoru hořlavých plynů	42
Obrázek 28 Graf analogového výstupu senzoru plamene	42
Obrázek 29 Ukázka funkce senzoru plamene a hořlavých plynů.....	43

Obrázek 30 Ukázka detektoru pohybu.....	44
Obrázek 31 LCD 1602 s I ² C převodníkem.....	45
Obrázek 32 Systém kontroly vstupu pod heslem (povolen / nepovolen vstup).....	46
Obrázek 33 Ukázka RFID čtečky	48
Obrázek 34 Vnější pohled.....	50
Obrázek 35 Vnitřní pohled	51
Obrázek 36 Schéma zapojení bez napájení a uzemnění [12].....	53

SEZNAM TABULEK

Tabulka 1 Čidla plynů [23]	27
Tabulka 2 Výsledky měření proudového odběru – detektor hořlavých plynů	43
Tabulka 3 Výsledky měření proudového odběru – LCD.....	46
Tabulka 4 Proudový odběr RFID MFRC-522 [8]	48
Tabulka 5 Popis komponent projektu (obr. 34 a obr. 35)	51
Tabulka 6 Popis ovládacích prvků.....	59
Tabulka 7 Údaje o spotřebě celého projektu	60

SEZNAM PŘÍLOH

P I: ZDROJOVÝ KÓD OTŘESOVÉHO ČIDLA

P II: SENZORU PLAMENE A SENZORU HOŘLAVÝCH PLYNŮ

P III: PIR DETEKTOR

P IV: SYSTÉM KONTROLY VSTUPU – HESLO

P V: SYSTÉM KONTROLY VSTUPU – RFID

P VI: GRAFICKÉ ZNÁZORNĚNÍ ZAPOJENÍ PROJEKTU

P VII: ZDROJOVÝ KÓD CELÉHO PROJEKTU

P VIII: ZKRÁCENÝ NÁVOD K PROJEKTU

P IX: SAMOLEPKY PRO PROJEKT

PŘÍLOHA P I: ZDROJOVÝ KÓD OTŘESOVÉHO ČIDLA

```
// Otřesové čidlo
//-----

// Definování konstant a proměnných
const int LED = 8;      // číslo pinu LED
const int cidlo = 2;    // číslo pinu čidla
volatile int stav = 0;  // inicializace proměnné pro čidlo

// Nastavení vstupů/výstupů
void setup() {
  pinMode(LED,OUTPUT); // nastavení pinu pro LED jako výstup
  pinMode(cidlo,INPUT); // nastavení pinu čidla jako vstup
}

// Naprogramování funkce
void loop() {           // smyčka, opakuj stále
  stav = digitalRead(cidlo); // načti hodnotu čidla (stav)
  delay(100);           // počkej 10ms
  if(LOW == stav) {     // jsou-li kuličky v kontaktu (čidlo vodí) -> uzemněno
    digitalWrite(LED,LOW); // vypni LED
  } else {              // pokud čidlo rozpojí - signalizuj
    digitalWrite(LED,HIGH); // zapni LED
  }
}
}
[8]
```

PŘÍLOHA P II: SENZORU PLAMENE A SENZORU HOŘLAVÝCH PLYNŮ

```
// Senzor plamene a hořlavých plynů
//-----

// Definování konstant a proměnných pro senzor plamene
const int plamen_A = A1;      // číslo pinu senzoru plamene - analogový
const int plamen_D = 8;      // číslo pinu senzoru plamene - digitální
const int LED_R = 10;        // číslo pinu červené LED
volatile int hod_plamen_A = 0; // inicializace proměnné pro analogový senzor plamene
volatile int hod_plamen_D = 0; // inicializace proměnné pro digitální senzor plamene

// Definování konstant a proměnných pro kouřový senzor
const int kour_A = A0;        // číslo pinu kouřového senzoru - analogový
const int kour_D = 6;        // číslo pinu kouřového senzoru - digitální
const int LED_Y = 4;         // číslo pinu žluté LED
volatile int hod_kour_A = 0;  // inicializace proměnné pro analogový kouřový senzor
volatile int hod_kour_D = 0;  // inicializace proměnné pro digitální kouřový senzor

// Nastavení vstupů/výstupů
void setup() {
  pinMode(LED_R,OUTPUT);      // nastavení pinu pro červenou LED jako výstup
  pinMode(plamen_D,INPUT);    // nastavení pinu senzoru plamene jako vstup - digitální
  pinMode(plamen_A,INPUT);    // nastavení pinu senzoru plamene jako vstup - analogový
  pinMode(LED_Y,OUTPUT);      // nastavení pinu pro žlutou LED jako výstup
  pinMode(kour_D,INPUT);      // nastavení pinu kouřového senzoru jako vstup - digitální
  pinMode(kour_A,INPUT);      // nastavení pinu kouřového senzoru jako vstup - analogový
  Serial.begin(9600);         // spustit sériový monitor
}

// Naprogramování funkce
void loop() {
  hod_plamen_A = analogRead(plamen_A); // přečíst analogovou hodnotu z čidla plamene
  hod_plamen_D = digitalRead(plamen_D); // přečíst digitální hodnotu z čidla plamene
  hod_kour_A = analogRead(kour_A);      // přečíst analogovou hodnotu z kouřového čidla
  hod_kour_D = digitalRead(kour_D);     // přečíst digitální hodnotu z kouřového čidla
  Serial.println("Analogové: " + String(hod_plamen_A) + " , " + String(hod_kour_A));
  // napiš na sériový monitor Analogové hodnoty z obou čidel ve formátu plamen , kouřové
  Serial.println("Digitální: " + String(hod_plamen_D) + " , " + String(hod_kour_D));
  // napiš na sériový monitor Digitální hodnoty z obou čidel ve formátu plamen , kouřové
  Serial.println("----");              // oddělit další záznam
  if(hod_plamen_A < 760) {
    // je-li hodnota menší jak 760 (citlivost při testu), změnou lze regulovat citlivost
    digitalWrite(LED_R,HIGH);          // zapni červenou LED
  } else {
    // jinak
    digitalWrite(LED_R,LOW);           // vypni červenou LED
  }
  if(hod_kour_A > 400) {                // je-li hodnota větší jak 400 (odpovídá cca nastavení trimru
    // pro digitální výstup), změnou lze regulovat citlivost
    digitalWrite(LED_Y,HIGH);          // zapni žlutou LED
  }
}
```

```
} else {           // jinak
    digitalWrite(LED_Y,LOW); // vypni žlutou LED
}
delay(500);       // počkej 0,5s
}
```

[39] [40]

PŘÍLOHA P III: PIR DETEKTOR

```
// Detektor pohybu PIR
//-----

// Definování konstant a proměnných
const int PIR_D = 8;      // číslo pinu PIR senzoru
const int LED_R = 10;    // číslo pinu červené LED
volatile int detekce = 0; // inicializace proměnné pro detekci osoby

// Nastavení vstupů/výstupů
void setup() {
  pinMode(PIR_D,INPUT); // nastavení pinu PIR senzoru jako vstup
  pinMode(LED_R,OUTPUT); // nastavení pinu červené LED jako výstup
}

// Naprogramování funkce
void loop() {
  detekce = digitalRead(PIR_D); // přečíst digitální hodnotu z PIR
  if(detekce == 1) {           // je-li detekován pohyb
    digitalWrite(LED_R,HIGH); // zapni červenou LED
  } else {                     // jinak
    digitalWrite(LED_R,LOW);  // vypni červenou LED
  }
  delay(100);                  // odpočiň si
}
```

PŘÍLOHA P IV: SYSTÉM KONTROLY VSTUPU – HESLO

```
// Systém kontroly vstupu - heslo
//-----
// kód je zároveň připraven pro integraci dalších funkcí
//-----
// Import funkcí
#include <LiquidCrystal_I2C.h>
#include <EM_TTP229.h>

LiquidCrystal_I2C lcd(0x27,16,2); // adresace LCD displeje, počet znaků a řádků

// Definice proměnných a konstant
EM_TTP229 klavesnice; // vytvoření objektu klávesnice
int SCLPin = 9; // číslo datového pinu pro SCL klávesnice
int SDOPin = 8; // číslo datového pinu pro SDO klávesnice
int funkce = 0; // pomocná proměnná pro zvolení funkce

// Proměnné a konstanty pro vstup pod heslem
int pozice = 0; // pozice v řetězci pro kód
#define delka 3 // počet znaků hesla - mělo by být na začátku kódu, ale víc se to hodí sem
char heslo[delka] = {1,2,3};
// správné heslo pro vstup - změním li délku hesla, musím změnit i konstantu delka, podmínka
char vlozene_heslo[delka] = {0,0,0}; // zatím prázdný řetězec pro vkládání hesla

// Prvotní nastavení HW
void setup() {
    klavesnice.initTTP229(SCLPin, SDOPin); // vstupy klávesnice
    lcd.init(); // inicializace displeje
    lcd.noBacklight(); // zatím zhaslí, kvůli úspoře
    lcd.setCursor(0,0); // výchozí pozice kurzoru
    lcd.print(" Vitej! "); // úvodní zpráva
    Serial.begin(9600); // nastavení portu sériového monitoru
}

// Základní smyčka programu
void loop() {
    int klavesa = klavesnice.GetKey()+1;
    // přečtení hodnoty stisklé klávesy, +1 protože originál v knihovně je pro jedničku 0
    switch (funkce) {
        case 0: // zatím bez činnosti
            urceni_funkce(klavesa);
            // zavolám funkci pro určení činnosti s SKV s hodnotou stisklé klávesy
            break;
        case 10: // byla stisknuta * což je vstup pod heslem
            vstup_pod_heslem(); // spustím funkci pro vstup pod heslem
            break;
        default:
            break;
    }
}
```

```

    }
}
// pomocná funkce pro určení činnosti, nastaví proměnou pro funkci
void urceni_funkce(int docasna) {
    switch (docasna) {          // zjistím co bylo stisknuto
        case 10:                // byla-li stisknuta * aktivuj funkci pro vstup pod heslem
            funkce = 10;        // nastav hodnotu pro funkci 10 index pro funkci vstup pod heslem
            break;
        default:
            funkce = 0;
            break;
    }
}
// samotná funkce pro vstup pod heslem
void vstup_pod_heslem() {
    int spravnost = 0;          // pomocná proměnná pro určení správnosti hesla
    int i;                      // indexová dočasná proměnná
    int klavesa_heslo = klavesnice.GetKey()+1; // proměnná stisknuté klávesy pro heslo, jiná
                                                // jak v hlavní funkci aby to nefungovalo, než zmáčknou *
    if (klavesa_heslo < 17) { // klávesnice má 16 tl., if kvůli eliminaci náhodných ruchů
        switch (klavesa_heslo) {
            case 10:            // byla zmáčknuta hvězdička, začnu provádět přípravný proces
                pozice = 0;      // nastavím pozici hesla na první položku
                lcd.clear();     // vyčistím displej
                lcd.backlight(); // zapnu podsvícení
                lcd.setCursor(0,0); // nastavím kurzor na začátek
                lcd.print("Zadej sve heslo:"); // napíšu co se má dít
                delay(1000);     // delka max stisku a také prodleva před stiskem druhého tl.
                Break;
            case 12:            // byl stisknutý # pro potvrzení vloženého hesla
                for(i = 0; i < delka; i++) { // projedeme obě hesla podle nastavené délky
                    if(vlozene_heslo[i] == heslo[i]) {
                        // porovnáme jednotlivé hodnoty v řetězci se vzorem
                        spravnost ++;        // jsou-li shodné potvrdíme správnost
                    }
                }
                if(spravnost == delka && pozice <= delka) {
                    // je-li kompletně heslo správně a není-li delší, vpusti
                    lcd.clear();            // vyčisti displej
                    lcd.setCursor(0, 0);    // nastav kurzor na začátek
                    lcd.print("  Spravne!  "); // napiš
                    lcd.setCursor(0, 1);    // kurzor na druhý řádek
                    lcd.print("  Vstup    "); // co se stalo
                    reset_hesla();          // provedení resetu hesla pro další použití
                    delay(5000);
                    setup();                // zavolat výchozí nastavení pro další činnost
                    funkce = 0;
                    pozice = 0;
                }
            }
        }
    }
}

```



```

    } else { // když není heslo správně
        lcd.clear(); // vyčisti displej
        lcd.setCursor(0, 0); // nastav kurzor na začátek
        lcd.print(" Chyba! "); // oznam chybu
        lcd.setCursor(0, 1); // kurzor na druhý řádek
        lcd.print(" Zkus to znova "); // oznámení
        delay(2000); // čas na přečtení oznámení
        reset_hesla(); // reset hesla
        setup(); // zavolat výchozí nastavení pro další činnost
        pozice = 0;
        funkce = 0;
    }
    break;
default: // výchozí funkce switche - zadávání hesla
    vlozene_heslo[pozice] = klavesa_heslo;
    // začínáme od nuly a vložíme stisklé číslo do řetězce
    lcd.setCursor(pozice,1); // posuneme kurzor
    lcd.print("*"); // maskovaně znázorníme napsání číslice
    pozice ++; // posuneme pozici
    delay(1000); // delka max stisku a také prodleva před stiskem druhého tl.
    Break;
}
}
}
// Funkce pro reset hesla
void reset_hesla() {
    int i; // indexová dočasná proměnná
    for (i = 0 ; i < delka ; i++) { // cyklus pro projetí všech položek v hesle
        vlozene_heslo[i] = 0; // vyresetování vloženého hesla
    }
}
}
[2] [39] [41]

```

PŘÍLOHA P V: SYSTÉM KONTROLY VSTUPU – RFID

```
// Systém kontroly vstupu - RFID
//-----

#include <rfid.h>
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#define max_len 5
RFID rfid; // vytvoříme objekt typu RFID
LiquidCrystal_I2C lcd(0x27,16,2); // inicializace displeje (adresa,sloupců,řádků)
uchar uzivatel_id[max_len]; // pole pro autorizovaného uživatele
uchar moje_id[max_len]; // pole pro aktuálně načtený tag
boolean nacteno; // pomocná proměnná pro běh programu

// výchozí nastavení
void setup() {
  Serial.begin(9600); // spustit sériový monitor
  rfid.begin(14, 5, 4, 3, 6, 2); // rfid parametry pinů
  (IRQ_PIN,SCK_PIN,MOSI_PIN,MISO_PIN,NSS_PIN,RST_PIN) - IRQ_PIN není potřeba zapojovat
  rfid.init(); // inicializace RFID
  lcd.init(); // inicializace LCD
  lcd.backlight(); // zapnout podsvícení
  lcd.setCursor(0,0); // nastavit výchozí pozici kurzoru
  lcd.print(" Vitej! "); // zobrazit " Vitej! "
  delay(1000); // dej si pauzu na rozběhnutí
}

// funkce pro vypsání načteného tagu
void vypis_id() {
  int i; // pomocná proměnná pro cyklus
  if (nacti_id() == true) { // zavolej funkci nacti_id a pokud něco načte, pokračuj
    Serial.print("Cislo cipu: ");
    lcd.setCursor(0,1);
    lcd.print("ID: ");
    for (i = 0 ; i < max_len-1 ; i++) { // cyklus pro projítí pole
      Serial.print(moje_id[i],HEX); // vypsání hodnoty tagu v hexadecimálním tvaru
      lcd.print(moje_id[i],HEX);
    }
    Serial.println();
    delay(2000);
  } else { // jinak ukonči
    return;
  }
}

// funkce pro autorizaci uživatele do systému
boolean uloz_id() {
  int i;
  if (nacti_id() == true) { // zavolej načtení tagu a když něco bylo načteno pokračuj
```

```

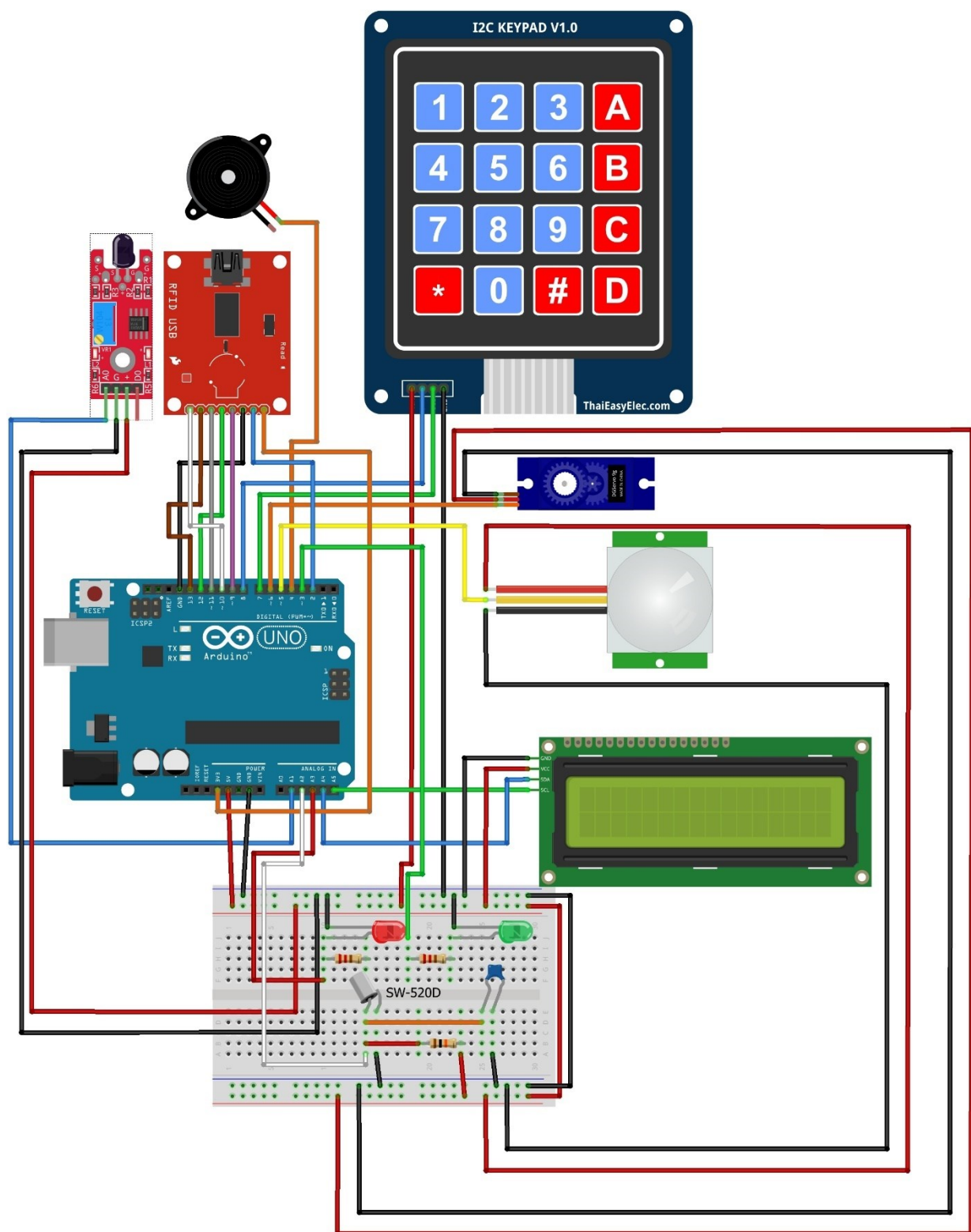
memcpy(uzivatel_id, moje_id, 5);           // zkopíruj načtené id do id uživatele
Serial.print("Vytvoren uzivatel s ID: ");
lcd.setCursor(0,0);
lcd.print("Vytvoren uzivatel");
lcd.setCursor(0,1);
lcd.print("s ID: ");
for (i = 0 ; i < max_len-1 ; i++) {       // vypiš info o konkrétním id
    Serial.print(uzivatel_id[i],HEX);
    lcd.print(uzivatel_id[i],HEX);
}
Serial.println();
delay(2000);
return true;
} else {
    return false;
}
}
// funkce pro načtení tagu
boolean nacti_id() {
    uchar status;                          // pomocná proměnná pro zjištění statusu čtečky
    uchar sken_id[MAX_LEN];                // celé načtené id
    int i;
    status = rfid.request(PICC_REQIDL, sken_id); // hledání čipu
    if (status != MI_OK) {                  // něco se pokazilo, odejdi
        return false;
    }
    status = rfid.anticoll(sken_id);
        // funkce pro předejití konfliktu která vrací 4 bajty sériového čísla karty
    if (status == MI_OK) {                  // pokud načtení proběhlo v pořádku
        memcpy(moje_id, sken_id,5);        // zkopíruj načtenou hodnotu do naší globální proměnné
        rfid.halt();                        // zastav čtečku
        return true;                        // návratová hodnota celé funkce - bylo provedeno
    }
    rfid.halt();
    return false;
}

// funkce pro kontrolu zda se jedná o povoleného uživatele
void kontrola() {
    if (nacti_id() == true) {               // pokud proběhlo načtení
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("    Vstup ");
        if (memcmp(uzivatel_id, moje_id,5) == 0) {
            // a pokud se shoduje načtené id s autorizovaným uživatelem
            Serial.print("Vstup");
            lcd.setCursor(0,1);
            lcd.print("    OK    ");        // povol vstup
        } else {                             // jinak vstup zamítni

```

```
    Serial.print("Neplatná karta");
    lcd.setCursor(0,1);
    lcd.print("    Zमितnut");
}
Serial.println();
delay(2000);
} else {
    return;
}
}
// samotný program
void loop() {
    //vypis_id();
    // zakomentováno, jen když požadují pro účely testování a nebo k dalšímu využití
    while (nacteno == false) { // dokud nezaregistruji uživatele opakuj,
        opětovné vyvolání registrace je přepsání hodnoty načteno na false
        nacteno = uloz_id(); // zavolej funkci pro autorizaci uživatele
            a změň hodnotu načteno na true - pokračování programu
    }
    kontrola(); // proved kontrolu načteného id a rozhodni zda je to autorizovaný uživatel
}
[2] [39]
```

PŘÍLOHA P VI: GRAFICKÉ ZNÁZORNĚNÍ ZAPOJENÍ PROJEKTU



fritzing

PŘÍLOHA P VII: ZDROJOVÝ KÓD CELÉHO PROJEKTU

```
// BP projekt "krabice"
//-----
// Petr Trávník
//-----
// Možnosti projektu jsou uvedeny v bakalářské práci
//-----

// Import funkcí
#include <LiquidCrystal_I2C.h> // knihovna pro displej
#include <EM_TTP229.h>         // knihovna pro klávesnici
#include <Servo.h>             // knihovna pro servo motorek
#include <rfid.h>               // knihovna pro RFID čtečku
//-----
//Definice
#define max_len 5             // maximální délka řetězce pro id tag
#define delka_hesla 3         // délka vstupního hesla
#define speed 20              // rychlost serva (větší číslo = pomalejší (ms/stupeň))
#define stisk 800             // prodleva mezi čtením stisku klávesy (ms)
#define necinnost 6000        // delka zhasnutí displeje po necinnosti (10000 = cca 52420ms)
#define citlivost 20          // citlivost otřesového čidla (pozorováním <10 téměř vždy, 10-25
                             // rozsah citlivost, >25 dlouhé rozpojení spíš převrácení)
#define cit_plamen 500        // citlivost detekce plamene, čím větší, tím citlivější
                             // (pro pokojové podmínky je ideální hodnota 750)
#define nas_val_f 2.8678      // násobek pro my_delay když víko false
#define nas_val_t 4.1511      // násobek pro my_delay když víko true
//-----
// Inicializace, objekty
LiquidCrystal_I2C lcd(0x27, 16, 2); //objekt lcd s parametry (adresa,počet znaků,počet řádků)
EM_TTP229 klavesnice;           // objekt klávesnice
RFID rfid;                       // objekt RFID čtečky 5117 - 5029
Servo servo6;                    // objekt servo motoru na pinu 6
//-----
// Definice proměnných a konstant
// SKV - heslo
byte funkce = 0;                 // pomocná proměnná pro zvolení funkce
byte pozice = 0;                 // pozice v řetězci pro kód (mimo kód kvůli smyčce)
char heslo[delka_hesla] = { 1, 2, 3 }; // správné heslo pro vstup - změním li délku hesla,
                                     // musím změnit i konstantu delka v definicích
char master_heslo[3] = { 3, 2, 1 }; // Administrátorské heslo pro nastavování
char vlozene_heslo[delka_hesla] = {}; // řetězec pro vkládání zadávaného hesla
char reset[delka_hesla] = {};     // trvale prázdné pole pro jednodušší reset hesla
int uzivatel = 0;                 // pomocná proměnná pro určení zadaného hesla
// SKV - rfid
uchar uzivatel_id[max_len];      // pole pro autorizovaného uživatele
uchar moje_id[max_len];          // pole pro aktuálně načtený tag
// Ostatní
float nasobek = nas_val_f;        // násobek pro funkci my_delay
```

```

boolean nacteno = false; // pomocná proměnná pro řízení toku programu
boolean bzucak = false; // pomocná proměnná pro ovládání ozvučení
boolean stav_vika = false; // pomocná proměnná pro sledování otevření víka
unsigned int standby = 0; // pomocná proměnná pro rozpoznání nečinnosti
// Piny z důvodu přístupu v programu
const byte PIR_D = 5; // číslo datového pinu pro PIR
const byte pip_D = 4; // číslo pinu pro bzučák
const byte plamen_A = A1; // číslo pinu senzoru plamene - analogový
const byte zelena_D = 3; // číslo pinu zelené diody - digitální
const byte cervena_A = A3; // číslo pinu červené diody - analogový
const byte otres_A = A2; // číslo pinu otřesového čidla - analogový
const byte otres2_A = A0; // číslo pinu otřesového čidla - analogový
//-----
// Prvotní výchozí nastavení HW
void setup() {
  Serial.begin(9600); // spustit sériový port v dané rychlosti
  pinMode(PIR_D, INPUT); // nastavení PIR senzoru jako vstup
  pinMode(pip_D, OUTPUT); // nastavení bzučáku jako výstup
  pinMode(plamen_A, INPUT); // nastavení senzoru plamene jako vstup
  pinMode(zelena_D, OUTPUT); // nastavení zelené diody jako výstup
  pinMode(cervena_A, OUTPUT); // nastavení červené diody jako výstup
  pinMode(otres_A, INPUT); // nastavení otřesového čidla jako vstup
  pinMode(otres2_A, INPUT); // nastavení otřesového čidla jako vstup
  servo6.attach(6); // přiřazení pinu pro servo motor
  servo6.write(10); // "vynuluj" polohu serva
  klavesnice.initTTP229(7, 8); // vstupy klávesnice (SCL,SDO)
  rfid.begin(9, 13, 11, 12, 10, 2); // rfid - parametry pinů
  //(IRQ_PIN,SCK_PIN,MOSI_PIN,MISO_PIN,SDA_PIN,RST_PIN)
  rfid.init(); // inicializace RFID čtečky
  lcd.init(); // inicializace LCD
  reset_lcd(1); // zobraz úvodní informace
  analogWrite(zelena_D, 30); // signalizace zapnutí - pomocí PWM na snížení jasu
  delay(1000); // dej si pauzu na rozběhnutí
}
//-----
// funkce pro detekci otevření víka
void viko() {
  for (int c = 1; analogRead(otres2_A) > 1000; c++) { // je-li detekováno otevření
    if (c >= 2000) { // je-li víko otevřeno
      analogWrite(cervena_A, 1023); // rozsvícení diody (max hodnota analogového pinu)
      reset_lcd(0); // reset zobrazení
      lcd.print(" Takhle NE! "); // informace o stavu
      lcd.setCursor(0, 1);
      lcd.print(" ZAVRIT! ");
      pipnuti(3); // vyvolání poplachu
      analogWrite(cervena_A, 0); // zhasnutí diody
      funkce = 13; // po zavření víka uvedení zařízení do původního sta-vu
    }
  }
}

```

```

}
//-----
// pomocná funkce pro určení násobky pro fci my_delay
void urci_stav_vika(boolean s) {
    stav_vika = s;          // nastav globální proměnnou stav_vika příchozí hodnotou
    if (s == true) {       // má-li být víko otevřeno
        nasobek = nas_val_t; // uprav násobek
    } else {               // má-li být víko zavřeno
        nasobek = nas_val_f; // nastav násobek
    }
}
//-----
// funkce pro detekci otřesení
void otres() {
    for (int c = 1; analogRead(otres_A) > 1000; c++) { // je-li detekován otřes
        if (c >= citlivost) { // dosažení citlivosti
            analogWrite(cervena_A, 1023); // rozsvícení diody
            delay(2);
            analogWrite(cervena_A, 0); // zhasnutí diody
            //Serial.println("otres"); // pro testování citlivosti
        } else {
            analogWrite(cervena_A, 0);
        }
        //Serial.println(c); // pro testování citlivosti
    }
}
//-----
//ovládání podsvícení displeje
void podsviceni(byte stav) { // pomocná funkce pro ovládání podsvětlení displeje
    if (stav == 1) { // vyžadují-li podsvícení
        standby = 0;
        lcd.backlight(); // zapni podsvícení
        //Serial.println("start-lcd"); // ladící funkce pro spočítání prodlevy
    } else {
        lcd.noBacklight(); // jinak vypni
        //Serial.println("konec-lcd"); // ladící funkce
    }
}
//-----
// ovládání serva
void servo_stupne(int uhel) {
    int u = servo6.read(); // dočasná proměnná pro zpomalení serva s aktuálním úhlem serva
    for (u; u <= uhel; u++) { // cyklus pro zpomalení serva - otevření ve stupních
        servo6.write(u); // zapiš úhel posunutý díky cyklu o 1 stupeň
        my_delay(speed); // pauza - rychlost
    }
    for (u; u >= uhel; u--) { // když poslední hodnota v servu byla větší (budu otáčet zpět)
        servo6.write(u);
        my_delay(speed);
    }
}

```



```

    }
}
//-----
//Reset LCD
void reset_lcd(byte i) {          // rozhodni úroveň resetu na základě vstupu
    lcd.clear();                 // vyčisti displej
    lcd.setCursor(0, 0);         // nastav kurzor na začátek
    podsviceni(1);               // nastavím hodnotu podsvícení
    if (i == 1) {                // pouze 2 varianty, v případě požadavku více, možno rozšířit o switch
        lcd.print("    Vitej! "); // navoď úvodní stav
        lcd.setCursor(0, 1);     // nastavte druhý řádek
        lcd.print("  Vyber funkci");
    }
}
//-----
//Vlastní funkce my_tone
void my_tone(float frekvence, float doba) {
    float pul_perioda = 500000 / frekvence; // pulperioda, potom 2x -> celý tón je peri-oda
    for (int i = 0; i < doba; i++) {        // délka tónu
        if (bzucak == true) {              // když je vyplí bzučák, nepípej
            digitalWrite(pip_D, HIGH);
        }
        delayMicroseconds(pul_perioda);
        digitalWrite(pip_D, LOW);
        delayMicroseconds(pul_perioda);
    }
}
//-----
// funkce pro pipnutí
void pipnuti(byte volba) { // volím si z předdefinovaných pípnutí
    switch (volba) {
        case 0: // něco špatně
            my_tone(500, 40);
            my_delay(200);
            my_tone(200, 40);
            break;
        case 1: // vše v pořádku
            my_tone(1000, 40);
            my_delay(200);
            my_tone(1500, 40);
            break;
        case 2: // jen potvrzení
            my_tone(1000, 40);
            break;
        case 3: // poplach
            for (byte r = 0; r < 5; r++) { // r je délka poplachu
                for (int i = 1000; i < 1400; i++) { // simulace sirény
                    my_tone(i, 1);
                    analogWrite(zelena_D, i - 1000); // pro optickou signalizaci
                }
            }
        }
}

```

```

    }
    for (int i = 1400; i > 1000; i--) {
        my_tone(i, 1);
        analogWrite(zelena_D, i - 1000);
    }
}
analogWrite(zelena_D, 30);
break;
case 4: // zastřežování systému - celkem 12,5s pro zpomalení změnit hodnoty ve „for“
//Serial.println("start první faze"); // pomocná fce pro určení délky
for (byte i = 0; i < 5; i++) { // 5x pípnout pomalu (10s)
    my_tone(1000, 40);
    my_delay(2000);
}
//Serial.println("start druhé faze"); // pomocná fce pro určení délky
for (byte i = 0; i < 5; i++) { // 5x zrychlit pípání (2,5s)
    my_tone(1000, 40);
    my_delay(500);
}
//Serial.println("konec zastřežení"); // pomocná fce pro určení délky
my_tone(1500, 40); // potvrzení zastřežení
my_delay(200);
my_tone(1000, 40);
break;
case 5: //stisk klávesy
    my_tone(2000, 30);
    break;
default:
    break;
}
}
//-----
// ověření o narušení vstupů
boolean naruseni() {
    if (servo6.read() > 15 || stav_vika == true) {
        reset_lcd(0);
        lcd.print(" Narusen vstup: ");
        lcd.setCursor(0, 1);
        lcd.print(" otevrene viko! ");
        my_delay(4000);
        return true;
    } else {
        return false;
    }
}
//-----
// funkce pro detekci plamene
void hori() {
    //Serial.println(analogRead(plamen_A)); // pro testování citlivosti

```

```

if (analogRead(plamen_A) < cit_plamen) { // jeli detekován oheň
    reset_lcd(0); // vymaž displej
    lcd.print("    HORI    "); // napiš že hoří
    pipnuti(3); // spusti akustickou výstrahu
    delay(3000); // počkej 3s
    reset_lcd(1); // navoď uvodní stav
}
}
//-----
// funkce pro trvale sledované senzory
void trvale() {
    otres(); // zavolám funkci pro detekci otřesu
    hori(); // zavolám funkci pro detekci plamene
    //ovládání podsvícení
    if (standby == necinnost) { // je-li doba pohotovosti rovna požadované doby vypnutí
        podsviceni(0);
    } else if (standby > necinnost) { // pokud displej nesvítí při pohybu zapni podsvícení
        if (digitalRead(PIR_D) == 1) { // je-li detekován pohyb při zhasnutém displeji
            podsviceni(1); // zapni podsvícení
        }
    }
    standby++;
    if (standby == 65535) { //před přetečením nastav hodnotu delší jak nečinnost
        standby = necinnost + 1;
    }
    if (stav_vika == false) { // je-li požadováno aby bylo zavřené víko
        viko(); // zjistí zda je opravdu zavřené
    }
}
//-----
// tísňový poplach
void tisen() {
    reset_lcd(0);
    lcd.print(" Byla vyvolana "); // oznámit opticky tiseň
    lcd.setCursor(0, 1);
    lcd.print("    TISEN    ");
    pipnuti(3); // oznámit akusticky tiseň
}
//-----
// funkce pro detekci pohybu
boolean PIR() {
    if (digitalRead(PIR_D) == 1) { // je-li pohyb vyvolej poplach a vrať hodnotu o de-tekci
        reset_lcd(0);
        lcd.print(" nekdo sem jde! ");
        pipnuti(3);
        return true;
    } else {
        return false;
    }
}

```

```

}
//-----
// samotná funkce pro vstup pod heslem
boolean vstup_pod_heslem() {
    int tlacitko_heslo = klavesnice.GetKey() + 1; // proměnná stisknuté klávesy pro hes-lo,
        // jiná jak v hlavní funkci aby to nefungovalo, než zmáčknou *
    if (tlacitko_heslo < 17) { //klávesnice má 16 tlačítek, if kvůli eliminaci náhodných ruchů
        switch (tlacitko_heslo) { // vybrání akce podle stisklé klávesy
            case 10: // byla zmáčknuta hvězdička, začnu provádět přípravný proces
                pipnuti(5);
                my_delay(stisk); // delka max stisku a také prodleva před stiskem druhého tla-čítka
                return false;
                break;
            case 12: // byl stisknutý # pro potvrzení vloženého hesla
                pipnuti(5);
                if (memcmp(vlozene_heslo, heslo, delka_hesla) == 0 && uzivatel != 99) {
                    // je-li kompletně heslo správně a není-li delší, vpusti
                    urci_stav_vika(true); // vypnout hlídání víka
                    reset_lcd(0); // vyčisti displej
                    lcd.print(" Spravne! "); // napiš
                    lcd.setCursor(0, 1); // kurzor na druhý řádek
                    lcd.print(" Vstup "); // co se stalo
                    servo_stupne(110); // otoč servo o 100 stupňů
                    memcpy(vlozene_heslo, reset, delka_hesla); // reset hesla pro další použití
                    my_delay(1000); // pauza na přečtení informací
                } else if (memcmp(vlozene_heslo, master_heslo, 3) == 0) {
                    // nebo pokud se jedná o heslo správce

                    reset_lcd(0);
                    lcd.setCursor(0, 1);
                    lcd.print(" Master ");
                    my_delay(3000);
                    memcpy(vlozene_heslo, reset, 3); // provedení resetu hesla pro další použití
                    return true; //ukonči smyčku a vrať se do loop
                    break;
                } else { // když není heslo správně
                    reset_lcd(0); // vyčisti displej
                    lcd.print(" Chyba! "); // oznam chybu
                    lcd.setCursor(0, 1); // kurzor na druhý řádek
                    lcd.print(" Zkus to znova "); // oznámení
                    my_delay(4000); // čas na přečtení oznámení
                    memcpy(vlozene_heslo, reset, delka_hesla); // reset hesla
                }
            }
        uzivatel = 0; //když je něco špatně, zruš funkci pro správce
        funkce = 0; // resetuj funkci pro další volbu
        return true;
        break;
    case 15: // bylo stisknuto D pro přidání RFID uživatele
        pipnuti(5);
        my_delay(stisk);

```

```

    return false;
    break;
default: // výchozí funkce switche - zadávání hesla
    pipnuti(5);
    if (pozice < delka_hesla) { // ověř zda nezadávat delší heslo
        vlozene_heslo[pozice] = tlacitko_heslo;
                                // začínáme od nuly a vložíme stisklé číslo do řetězce
        lcd.setCursor(pozice, 1); // posuneme kurzor na druhý řádek
        lcd.print("*");           // maskovaně znázorníme napsání číslice
        pozice++;                 // posuneme pozici
        my_delay(stisku); // delka max stisku a také prodleva před stiskem druhého tlačítka
        return false;
        break;
    } else { // zadali jsme více znaků
        reset_lcd(0);           // vyčisti displej
        lcd.print("    Chyba!    "); // oznam chybu
        lcd.setCursor(0, 1);    // kurzor na druhý řádek
        lcd.print("  Moc znaku "); // oznámení
        my_delay(2000);         // čas na přečtení oznámení
        memcpy(vlozene_heslo, reset, delka_hesla); // reset hesla
        reset_lcd(1);
        uzivatel = 0; //když je něco špatně, zruš funkci pro správce
        funkce = 0;
        return true;
        break;
    }
}
}
return false;
}
//-----
// funkce pro načtení tagu
boolean nacti_id() {
    uchar status; // pomocná proměnná pro zjištění statusu čtečky
    uchar sken_id[MAX_LEN]; // celé načtené id
    status = rfid.request(PICC_REQIDL, sken_id); // hledání čipu
    if (status != MI_OK) { // něco se pokazilo, odejdi
        funkce = 0;
        return false;
    }
    status = rfid.anticoll(sken_id); // funkce pro předejití konfliktu,
                                    // která vrací 4 bajty sériového čísla karty
    if (status == MI_OK) { // pokud načtení proběhlo v pořádku
        memcpy(moje_id, sken_id, 5); // zkopíruj načtenou hodnotu do naší globální pro-měnné
        rfid.halt(); // zastav čtečku
        funkce = 0;
        return true; // návratová hodnota celé funkce - bylo provedeno
    }
}
rfid.halt();

```

```

funkce = 0;
return false;
}
//-----
// funkce pro autorizaci uživatele do systému
boolean uloz_id() {
    if (nacti_id() == true) { // zavolej načtení tagu a když něco bylo načteno pokračuj
        memcpy(uzivatel_id, moje_id, 4); // zkopíruj načtené id do id uživatele
        pipnuti(2); // akustické potvrzení načtení
        reset_lcd(0);
        lcd.print("Vytvoren uzivatel"); // vypiš informaci s ID
        lcd.setCursor(0, 1);
        lcd.print("s ID: ");
        vypis_id();
        my_delay(4000);
        return true;
    } else {
        return false;
    }
}
//-----
// funkce pro vypsání načteného tagu
void vypis_id() {
    for (int i = 0; i < max_len - 1; i++) { // cyklus pro projítí pole
        lcd.print(moje_id[i], HEX);
    }
}
//-----
// funkce pro kontrolu zda se jedná o povoleného uživatele
boolean kontrola() {
    if (nacti_id() == true) { // pokud proběhlo načtení
        reset_lcd(0);
        lcd.print("    Vstup ");
        if (memcmp(uzivatel_id, moje_id, 4) == 0) {
            // pokud se shoduje načtené id s autorizovaným uživatelem
            pipnuti(1);
            lcd.setCursor(0, 1);
            lcd.print("    OK "); // povol vstup
            funkce = 10; // nastav funkci pro zadání hesla pro otevření
        } else { // jinak vstup zamítni
            pipnuti(0); // akustická signalizace chyby
            lcd.setCursor(0, 1);
            lcd.print("    Zamitnut ");
            my_delay(2000);
        }
        my_delay(2000);
        return true;
    } else {
        return false;
    }
}

```

```

    }
}
//-----
// funkce pro zastřežení systému
boolean zastrezeni() {
    switch (klavesnice.GetKey() + 1) { // detekce odstřežení a tísňe
        case 5: // detekce tísňového tlačítka
            tisen(); // spusti tiseň
            reset_lcd(0);
            lcd.print(" Zastrezeno "); // pokračuj v zastřeženém režimu
            analogWrite(zelena_D, 1);
            return false;
            break;
        case 13: // je požadavek odstřežení systému
            reset_lcd(0);
            lcd.print(" Odstrezeno "); // oznam odstřežení
            funkce = 13; // nastav funkci pro uvedení systému do výchozí pozice
            my_delay(2000); // prodleva pro přečtení
            return true; // ukonči střežení
            break;
        default:
            break;
    }
    if (PIR() == true) { // pokud byl spatřen pohyb, proved' poplach
        reset_lcd(0);
        lcd.print(" Zastrezeno "); // vrať potom informaci na displej o stavu systému
        analogWrite(zelena_D, 1);
    }
    return false; // jinak furt udržuj systém zastřežený
}
//-----
void my_delay(float i) {
    i = nasobek * i;
    for (float t = 0; t < i; t++) {
        trvale();
        standby = 0;
    }
}
//-----
// testovací funkce pro vlastní odezvu
void test_delay() { //výpočet pak 100000 / naměřená hodnota = násobek pro funkci my_delay
    // před měřením vynulovat násobky v my_delay
    // pro přesné měření zapnout na sériové lince časové razítko
    urci_stav_vika(false); // 34870ms -> nasobek = 2.8678;
    pipnuti(2);
    Serial.println("START");
    my_delay(100000); //test o délce 1min 40 sec = 100000ms
    Serial.println("KONEC");
    pipnuti(2);
}

```

```

my_delay(5000);
urci_stav_vika(true); // 24090ms -> nasobek = 4.1511;
Serial.println("2.START");
pipnuti(2);
my_delay(100000); //test o délce 1min 40 sec = 100000ms
Serial.println("2.KONEC");
pipnuti(2);
my_delay(5000);
}
//-----
// Základní smyčka programu
void loop() {
//test_delay(); // test pro určení násobku pro my_delay
int tlacitko = klavesnice.GetKey() + 1;
// přečtení hodnoty stisklé klávesy, +1 protože originál v knihovně je pro jed-ničku 0
trvale(); // sleduji senzory pro nepřetržité upozornění }
switch (funkce) {
case 0: // zatím bez činnosti
funkce = tlacitko; // přiřadím stisklé tlačítko funkci pro určení činnosti s krabicí
break;
case 5: // funkce pro tiseň
tisen(); // zavolám samotnou funkci
reset_lcd(1); // výchozí displej
funkce = 0; // resetuji proměnou pro další činnost
break;
case 10: // byla stisknuta * funkce pro vstup pod heslem
pozice = 0; // nastavím pozici hesla na první položku
reset_lcd(0); // vyčisti displej
lcd.print("Zadej sve heslo:"); // napíšu co se má dít
while (nacteno == false) {
nacteno = vstup_pod_heslem(); // spustím funkci pro vstup pod heslem
trvale(); // sleduji senzory pro nepřetržité upozornění
}
nacteno = false; // reset proměnné pro další využití
reset_lcd(1); // výchozí displej
break;
case 11: // byla stisknuta 0 pro aktivaci bzučáku
reset_lcd(0); // vyčisti displej
lcd.print(" Zvuk ");
lcd.setCursor(0, 1);
if (bzucak == false) { // jeli bzučák vypnutý -> zapni a obráceně a informuj
lcd.print(" ZAPNUT ");
bzucak = true;
} else {
lcd.print(" VYPNUT ");
bzucak = false;
}
}
pipnuti(5);
my_delay(4000); // prodleva pro přečtení

```



```

    reset_lcd(1);    // výchozí displej
    funkce = 0;     // resetuji proměnou pro další činnost
    break;
case 13: // bylo stisknut D což je reset do původního stavu a odstřežení
    reset_lcd(0);
    lcd.print("    NASTAVIT    ");
    lcd.setCursor(0, 1);
    lcd.print(" Puvodni stav");
    my_delay(3000);    // prodleva pro přečtení
    servo_stupne(10);    // nastavení výchozí polohy serva (zavření víka)
    analogWrite(zelena_D, 30); // vrácení původního jasu diody
    reset_lcd(1);    // výchozí displej
    funkce = 0;     // resetuji proměnou pro další činnost
    nacteno = false;    // reset proměnné pro další využití
    urci_stav_vika(false); // zapnout hlídání otevření víka
    break;
case 14:    // bylo stisknuto C pro zastřežení systému
    if (naruseni() == true) { // spusti funkci pro ověření narušení vstupů
        reset_lcd(1);    // výchozí displej
        funkce = 0;     // resetuji proměnou pro další činnost
        break;
    }
    reset_lcd(0);
    lcd.print("    Probiha    ");
    lcd.setCursor(0, 1);
    lcd.print("    Zastrezeni    ");
    pipnuti(4);
    reset_lcd(0);
    lcd.print("    Zastrezeno    ");
    analogWrite(zelena_D, 1); // nastavení minimálního jasu pro úsporu energie
    while (nacteno == false) {
        nacteno = zastrezeni(); // dokud není stisknuto D udržuj smyčku pro zastřežení
        trvale();    // sleduji senzory pro nepřetržité upozornění
    }
    if (funkce != 13) { // pokud dojde k ukončení smyčky, spíše ladící část programu
        funkce = 0;
        reset_lcd(1);
    }
    nacteno = false; // reset proměnné pro další využití
    break;
case 15:    // registrace nové karty
    pozice = 0;    // nastavím pozici hesla na první položku
    uzivatel = 99; // určím uživatele - správce
    reset_lcd(0);
    lcd.print(" Heslo spravce: "); // napíšu co se má dít
    while (nacteno == false) {
        nacteno = vstup_pod_heslem(); // spustím funkci pro vstup pod heslem
        trvale();    // sleduji senzory pro nepřetržité upozornění
    }
}

```

```

if (uzivatel == 99) { //pokud bylo vloženo heslo správně, zůstane uživatel 99
  nacteno = false; // reset proměnné pro další využití
  reset_lcd(0);
  lcd.print(" Prilozte CIP"); // napíšu co se má dít
  while (nacteno == false) { // dokud nezaregistruji uživatele, opakuj
    nacteno = uloz_id(); // zavolej funkci pro autentizaci uživatele
    trvale(); // sleduji senzory pro nepřetržité upozornění
  }
}
uzivatel = 0; // reset pro další použití
nacteno = false; // reset pro další použití
reset_lcd(1); // výchozí displej
break;
case 16: // autorizace autentizovaného uživatele
  reset_lcd(0);
  lcd.print(" Prilozte CIP");
  while (nacteno == false) {
    nacteno = kontrola(); // proved autorizaci uživatele
    trvale(); // sleduji senzory pro nepřetržité upozornění
  }
  nacteno = false; // reset pro další použití
  reset_lcd(1); // výchozí displej
  break;
default: // není-li definovaná funkce, nereaguj, jen signalizuj stisk klávesy
  funkce = 0;
  break;
}
}

```

[2] [8] [39] [40] [41] [42]

P VIII: ZKRÁCENÝ NÁVOD K PROJEKTU

Na levé části se nachází přepínač pro ovládání napájení zařízení, posunutím od sebe jej lze zapnout. Napájení zařízení je signalizováno zelenou LED vedle displeje a červenou LED na klávesnici. Níže v tabulce jsou popsány jednotlivé funkce projektu a způsoby jak je lze aktivovat.

Klávesa	Popis
5	Tísňové tlačítko
*	SKV – vstup pomocí hesla (123)
#	Potvrzení zadaného heslo
A	SKV – RFID, aktivace vstupu
B	RFID – registrace nové karty (321)
C	Zastřežení systému
D	Odstřežení systému a výchozí stav
0	Zapnutí/vypnutí zvuku

Tísňové tlačítko – vyvolá akustický poplach a vypíše informaci o vyvolání tísně. Toto tlačítko funguje ve výchozím a zastřeženém stavu.

Hvězdička – aktivuje vstup pomocí hesla (výchozí heslo: 1 2 3). Po správném zadání dojde k otevření krabice.

Křížek – slouží k potvrzení zadaného hesla, jak hesla uživatele, tak administrátorského hesla.

A – aktivace RFID čtečky, pro vstup pomocí čipu a zadání hesla uživatele pro vyšší stupeň zabezpečení (výchozí heslo: 1 2 3). Pro správnou funkčnost je nutné nejprve přidat uživatele pomocí další funkce „B“.

B – po zadání administrátorského hesla (výchozí heslo: 3 2 1), vyvolá výzvu k přiložení čipu a provede registraci nového uživatele se zobrazením ID v hexadecimálním tvaru.

C – stisk toho tlačítka vyvolá zastřežení systému, které předchází doba k opuštění prostoru. Po této časové prodlevě dojde k aktivaci senzoru, který v případě pohybu vyvolá poplach. Pokud zařízení není ve výchozím stavu, z důvodu narušených vstupů k zastřežení nedojde.

D – po stisknutí uvede zařízení do výchozího stavu jako po zapnutí zařízení, v případě zastřeženého systému slouží k odstřežení.

0 – nula slouží k aktivaci, či deaktivaci zvukového výstupu projektu. Ve výchozím stavu je akustické upozornění vypnuto



Created By



Tl.	Popis
5	Tišňové tlačítko
*	SKV - vstup pomocí hesla (123)
#	Potvrzení zadaného heslo
A	SKV - RFID, aktivace vstupu
B	RFID - registrace nové karty (321)
C	Zastřežení systému
D	Odstřežení systému a výchozí stav
0	Zapnutí/vypnutí zvuku



Power

I 0

Programovací konektor

USB – B

