

Návrh a implementace chytrého elektronického zámku s využitím platformy Arduino

Jakub Jatel

Bakalářská práce
2025



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav automatizace a řídicí techniky

Akademický rok: 2024/2025

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	Jakub Jatel
Osobní číslo:	A22192
Studijní program:	B0714A150006 Aplikovaná informatika v průmyslové automatizaci
Specializace:	Inteligentní systémy s roboty
Forma studia:	Prezenční
Téma práce:	Návrh a implementace chytrého elektronického zámku s využitím platformy Arduino
Téma práce anglicky:	Design and Implementation of a Smart Electronic Lock Using the Arduino Platform

Zásady pro vypracování

- Vytvořte literární rešerši na dané téma.
- Analyzujte potřebné požadavky na návrh chytrého elektronického zámku.
- Navrhněte a vytvořte chytrý elektronický zámek s využitím platformy Arduino.
- Otestujte a vyhodnoťte funkčnost zámku

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

Seznam doporučené literatury:

1. BLUM, Jeremy. Exploring Arduino: tools and techniques for engineering wizards. Indianapolis: Wiley, 2013. ISBN 978-1-118-54936-0.
2. BANZI, Massimo a SHILOH, Michael. Getting Started with Arduino. 3rd ed. Sebastopol: Maker Media, 2014. ISBN 978-1-4493-6333-8.
3. MONK, Simon. Programming Arduino: Getting Started with Sketches. 2nd ed. New York: McGraw-Hill, 2016. ISBN 978-1-259-64214-4.
4. NOURI, Hussein. Arduino for Embedded Systems. Switzerland: Springer, 2018. ISBN 978-3-319-77621-6.
5. MARGOLIS, Michael. Arduino Cookbook. 3rd ed. Sebastopol: O'Reilly Media, 2020. ISBN 978-1-4919-5518-4.
6. PATTERSON, Zachary a KNIGHT, Charles. Learning Arduino: A Beginner's Guide to Microcontroller Projects. United Kingdom: Packt Publishing, 2020. ISBN 978-1-80056-647-8.

Vedoucí bakalářské práce: **Ing. Václav Mach, Ph.D.**
Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce: **13. prosince 2024**
Termín odevzdání bakalářské práce: **27. května 2025**

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



prof. Ing. Vladimír Vašek, CSc. v.r.
ředitel ústavu

Ve Zlíně dne 13. prosince 2024

PROHLÁŠENÍ AUTORA BAKALÁŘSKÉ PRÁCE

Beru na vědomí, že

- odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., v platném znění bez ohledu na výsledek obhajoby;
- bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému a bude dostupná k nahlédnutí;
- jedno vyhotovení bakalářské práce v listinné podobě bude ponecháno Univerzitě Tomáše Bati ve Zlíně k uložení;
- 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;
- podle § 60 odst. 1 autorského zákona má Univerzita Tomáše Bati ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- podle § 60 odst. 2 a 3 mohu užít své dílo – bakalářskou práci – nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tj. k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- 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(a) samostatně a použitou literaturu jsem řádně citoval(a); v případě publikace výsledků budu uveden(a) jako spoluautor;
- odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou obsahově totožné.

Ve Zlíně, dne

.....
podpis autora

ABSTRAKT

Tato bakalářská práce se zaměřuje na vývoj elektronického zámku řízeného platformou Arduino. Cílem je vytvořit bezpečnostní systém, který umožní ovládat přístup pomocí digitálního kódu nebo RFID technologie. Systém bude sestaven z jednoduchých komponent, jako jsou servo motor a RFID čtečka. Jeho výhodou bude snadná instalace a ovládání. Systém bude navržen tak, aby poskytoval uživatelsky přívětivé ovládání a byl cenově dostupný pro domácí i kancelářské použití.

Klíčová slova: Arduino, Mikropočítač, RFID technologie,

ABSTRACT

This bachelor's thesis focuses on the development of an electronic lock controlled by the Arduino platform. The aim is to create a security system that allows access control using either a digital code or RFID technology. The system will be built using basic components such as a servo motor and RFID reader, offering the advantages of easy installation and operation. It will be designed to provide user-friendly functionality while remaining cost-effective for both home and office use.

Keywords: Arduino, Microcontroller, RFID technology

Děkuji vedoucímu mé bakalářské práce panu doktorovi Ing. Václavu Machovi Ph.D. za zachování chladné hlavy i v náročných chvílích a za cenné rady, které výrazně přispěly ke zlepšení návrhu mého projektu.

OBSAH

ÚVOD.....	11
I TEORETICKÁ ČÁST	12
1 VÝVOJOVÁ PLATFORMA ARDUINO	13
1.1 ROZDĚLENÍ ZÁKLADNÍCH DESEK.....	14
1.1.1 ČÁSTI DESKY	15
1.1.2 ARDUINO UNO R3	16
1.1.3 ARDUINO MIKRO.....	18
1.1.4 ARDUINO MEGA 2560 REV3	19
1.2 ARDUINO NANO ESP32.....	20
2 BEZPEČNOSTNÍ ZÁMKY	21
2.1 ELEKTRONICKÉ ZÁMKY	22
2.1.1 ZÁMKY NA KÓD.....	23
2.1.2 RFID ZÁMKY	24
2.1.3 BIOMETRICKÉ ZÁMKY	25
3 ROZŠÍŘUJÍCÍ MODULY (SHIELDY).....	26
3.1 MATRICOVÁ KLÁVESNICE	27
3.2 RFID MODUL	28
4 DOPLŇUJÍCÍ SOFTWARE	30
4.1 PROGRAM SOLIDWORKS	30
4.2 ANDROID STUDIO.....	31
4.2.1 PROGRAMOVACÍ JAZYK JAVA	32
4.2.2 KOTLIN	33
4.3 ARDUINO IDE.....	34
II PRAKTICKÁ ČÁST	35
5 NÁVRH ZÁMKU A SYSTÉMOVÉ ARCHITEKTURY.....	36
5.1 HARDWAROVÉ VYBAVENÍ	36
5.1.1 VÝVOJOVÁ DESKA LASKAKIT ESP32 DEVKIT	36
5.1.2 VÝVOJOVÁ DESKA ARDUINO NANO	38
5.1.3 SERVO MOTOR MG90S	39
5.1.4 RFID ČTEČKA RC522	40
5.1.5 NAPÁJENÍ	41
5.1.6 1 - BUTTON SHIELD	42
5.2 KONSTRUKCE MECHANISMU ZÁMKU.....	43
5.2.1 NÁVRH PROTOTYPU ZÁMKU	44

5.2.2	SCHRÁNKA MECHANISMU ZÁMKU.	45
5.3	SCHÉMA ZAPOJENÍ KOMPONENTŮ	48
6	PROGRAMOVÁ ČÁST ZÁMKU	50
6.1	POUŽITÉ KNIHOVNY	50
6.2	WEBOVÉ UŽIVATELSKÉ ROZHRANÍ	51
6.2.1	WEBOVÉ OVLÁDÁNÍ ZÁMKU A UŽIVATELSKÉ FUNKCE	53
6.2.2	ZMĚNA HESLA A ODSTRANĚNÍ REGISTRACE	54
6.2.3	INICIALIZACE HARDWARU NA DESCE ARDUINO NANO.....	56
6.2.4	ZPRACOVÁNÍ RFID KARTY A PŘÍKAZŮ Z ESP32.....	57
6.2.5	ROZPOZNÁVÁNÍ KARET	58
7	TESTOVÁNÍ A ANALÝZA VÝSLEDKŮ.....	59
7.1	TESTOVÁNÍ BEZPEČNOSTI	59
7.2	TESTOVÁNÍ RŮZNÝCH MATERIÁLŮ A VZDÁLENOSTÍ	60
7.2.1	OCHRANNÁ SCHRÁNKA	60
7.2.2	SKLO	60
7.2.3	DŘEVOTŘÍSKA	60
7.2.4	KOV.....	61
7.2.5	ZHODNOCENÍ VÝSLEDKŮ	61
7.3	TESTOVÁNÍ RYCHLOSTI WEBOVÉ STRÁNKY V ZÁVISLOSTI NA SIGNÁLU.....	62
8	PLÁNY DO BUDOUCNA	63
8.1	POROVNÁNÍ S KOMERČNĚ DOSTUPNÝMI ZÁMKY.....	64
	ZÁVĚR.....	65
	SEZNAM POUŽITÉ LITERATURY	66
	SEZNAM OBRÁZKŮ	70
	SEZNAM TABULEK.....	72
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	73
	SEZNAM PŘÍLOH.....	75
	PŘÍLOHA P I: NÁZEV PRVNÍ PŘÍLOHY	76

ÚVOD

Bezpečnost a přístupové systémy hrají v dnešním světě čím dál významnější roli. S rostoucí poptávkou po chytrých a snadno ovladatelných řešeních pro zabezpečení majetku roste i zájem o elektronické zámky, které nahrazují tradiční mechanické systémy. Elektronické zámky nabízejí nejen vyšší úroveň ochrany, ale také větší komfort a možnosti přizpůsobení konkrétním požadavkům uživatele. Zatímco dříve byla taková řešení doménou profesionálních firem, dnes je díky dostupným mikropočítačovým platformám možné podobné systémy navrhnout a realizovat i v rámci individuálního vývoje.

Tato bakalářská práce se zaměřuje na návrh a realizaci chytrého elektronického zámku, který je postaven na platformě Arduino, konkrétně na desce ESP32. Tato platforma byla zvolena díky svým pokročilým možnostem v oblasti bezdrátové komunikace, nízké spotřebě energie a dobré podpoře ve vývojové komunitě. Cílem bylo vytvořit zámek, který bude možné ovládat pomocí digitálního hesla prostřednictvím webového rozhraní, nebo přiložením RFID čipu. Systém byl navržen s důrazem na jednoduchost použití, bezpečnost, modulárnost a potenciál pro rozšíření do systémů chytré domácnosti.

V rámci práce byl kladen důraz nejen na softwarové řešení, ale i na mechanickou konstrukci zámku, která byla realizována pomocí 3D tisku. Dále bylo navrženo webové uživatelské rozhraní, které umožňuje správu přístupových údajů bez nutnosti připojení k internetu. Díky použití běžně dostupných komponent a otevřených technologií je výsledný systém cenově dostupný a zároveň přizpůsobitelný pro další vývoj.

I TEORETICKÁ ČÁST

1 VÝVOJOVÁ PLATFORMA ARDUINO

Arduino je open-source platforma založená na snadno použitelném hardwaru a softwaru. Společnost byla založena v roce 2005 v Itálii, jak je možné vidět na deskách. Jejím původním záměrem bylo vytvořit jednoduchou prototypovou platformu určenou z větší části jen pro studenty, ale díky rostoucí popularitě platformy se rozjel vývoj a různé spolupráce a začaly vycházet novější verze a v roce 2010 se prodalo více než 120 000 kusů. Usilují o zpřístupnění elektroniky a digitálních technologií široké veřejnosti. Proto udělali rozdělení 3 hlavních oblastí zaměření: Maker pro individuální inovátory, Education pro studenty a výuku a PRO pro firmy v oblasti IoT a Umělých inteligencí [7,9,11]



Obrázek 1: Arduino Uno [8]

Ve stejném roce také vyšla jedna z přelomových desek – deska Arduino Uno, která výrazně přispěla na popularitě mezi vývojáři. Díky své jednoduchosti a široké dostupnosti se stala základem tisíce projektů. Kromě Uno vyšla také deska Arduino Mega která byla vyvinuta jako variace desky Uno a byla určena pro pokročilejší aplikace, které vyžadovaly více pinů. Nabízela větší kapacitu paměti a vyšší flexibilitu a proto byla ideální na rozsáhlejší systémy ať už v robotice, automatizaci nebo IoT. Na fotce také můžeme vidět originální logo firmy Arduino, jejich logo symbolizuje nekonečné možnosti pro vývoj a tvořivost stejně tak jako jejich ikonická modro-zelená barva kterou můžeme vidět u všech oficiálních produktů. [7,8]

Mezi první větší spolupráce firmy Arduino patří Intel, Jejich spolupráce se vyplatila a přivedla na trh koncem roku 2013 první základní deska s integrovaným samostatným grafickým výstupem, byl osazen energeticky úsporným čipem Intel Quark což umožnilo Arduino využít v náročnějších a složitějších projektech. Otevřeli tím nové možnosti v oblasti vývoje hardware a softwaru a mělo to výrazný dopad na rozvoj platformy a jejího ekosystému. [10]

1.1 Rozdělení základních desek

Pro většinu Arduino desek je typické že jsou vybaveny 8-bitovými procesory od firmy Atmel což zajišťuje jednoduchost hardwarové platformy a umožňuje použití stejného programového vybavení napříč různými modely. Přičemž většina z nich, včetně jedné z nejpobulárnějších desek Arduino Uno, využívá procesor řady AVR ATmega. Programy těchto mikrokontrolerů jsou obvykle napsány v jazyce C nebo Assembly a programují se přes ICSP rozhraní. Jednou z nejdůležitějších vlastností je že je možné jej programovat jednoduše přes USB bez samotného programátoru, tuto možnost umožňuje bootloader Arduino který je nahrán do systému ATmega už při výrobě. [1]

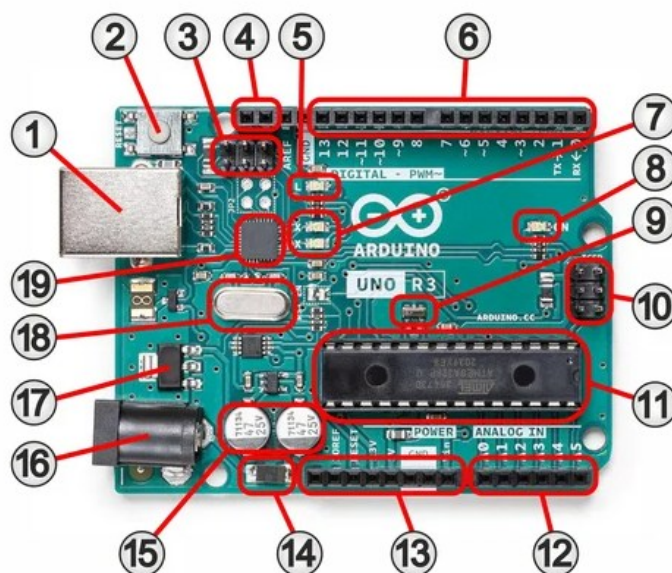
Typ desky Arduino	Typ procesoru	CPU	Analogové vstupy	Digitální vstupy	Flash paměť	Provozní napětí
Zero	ATSAMD21G18	32-bit	6	20	256 kB	3,3 V
Uno R3	ATmega328P	8-bit	6	14	32 kB	5 V
Due	ATmelSAM3X8E	32-bit	12	54	512 kB	3,3 V
Mega 2560	ATmega2560	8-bit	16	54	256 kB	5 V
Leonardo	ATmega32u4	8-bit	12	20	32 kB	5 V
Micro	ATmega32u4	8-bit	12	20	32 kB	5 V
Mini	ATmega328	8-bit	8	14	32 kB	5 V
Nano	ATmega328	8-bit	8	22	32 kB	5 V

Tabulka 1: Některé z nejznámějších Arduino desek a jejich parametry[12]

Arduino desky se liší jak typem procesoru tak i architekturou a ovlivňuje to jejich zaměření, UNO, Nano a Mega 2560 mají 8-bitové čipy a jsou ideální pro běžné většinou nenáročné projekty, jako je projekt se senzory nebo jen ovládání servomotorů. Na rozdíl od Zero a Due které mají výkonnější 32-bitové čipy a jsou vhodnější pro náročnější úlohy jako je například zpracovávání různých signálů. Provozní napětí je též velice důležité při výběru kompatibilních modulů, aby nedošlo k poškození externích periférií z důvodu nesprávného napětí.

1.1.1 Části desky

Každá deska se skládá z určitých částí, jejich počet je ovlivněn jejím zaměřením a samozřejmě velikosti desky samotné. Jedna z nejdůležitějších částí je USB konektor (1), slouží jako hlavní komunikační rozhraní mezi počítačem a deskou. Při chybě v programu a zaseknutí systému je možné resetovat systém manuálně na desce (2). ICSP rozhraní nám umožňuje sériové programování USB kontroléru (3). Sběrnice I²C přenáší data z připojeného zařízení „Master“ do zařízení „Slave“ pomocí vodičů SCL a SDA. Na desce též můžeme najít testovací LED, pomocí které můžeme testovat funkčnost desky a je možné ji ovládat z pinu 13 (5).



Obrázek 2: Části Arduino Desky [12]

Pro práci se vstupy a výstupy máme k dispozici sadu digitálních I/O pinů, z nichž 6 podporuje PWM výstup, což umožňuje simulaci analogových signálů (6). Správný přenos informací mezi počítačem a deskou signalizují LED RX a TX (7).

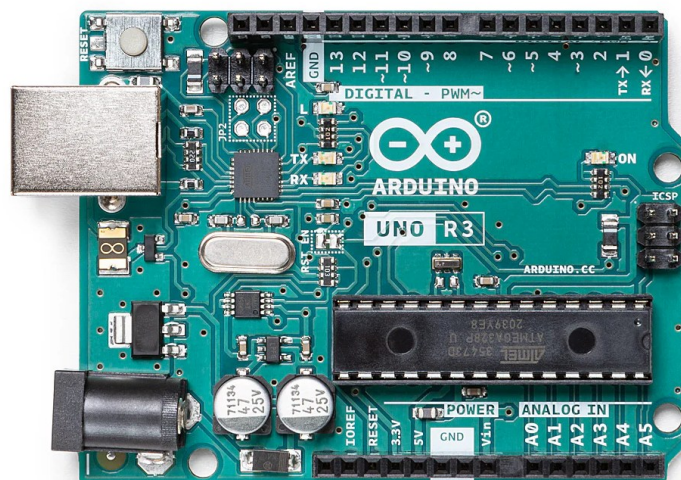
Poslední dioda na desce LED ON signalizuje napájení desky (8). Důležitým prvkem je krystalový oscilátor, krystal zajišťuje stabilitu oscilátoru v kontroléru a udržuje konstantní frekvenci (9).

Na pravé straně můžeme vidět další ICSP tentokrát ale pro samotný mikrokontrolér, ale většinou to není potřeba protože už je naprogramovaný z výroby (10). Nejdůležitějším prvkem na desce je samozřejmě mikrokontrolér, v případě této desky mikrokontrolér ATmega328P. Je složený z procesoru, periférií a paměti a slouží jako řídicí jednotka celé desky (11). Pro práci s analogovými signály je možné využít analogové vstupy, většinou použité na přenos informací z například: senzorů (12).

Napájení desky lze zajistit přes napájecí piny, které poskytují 3,3V a 5V výstupy (13). Usměrnovací dioda (14) chrání před nesprávným připojením zdroje, zatímco kondenzátory stabilizují napětí (15). Externí zdroj lze připojit přes napájecí konektor(16), přičemž stabilizátor napětí převádí vstup na stabilních 5V (17). USB rozhraní převádí signály do formátu kompatibilního s mikrokontrolérem (19), jehož stabilitu zajišťuje krystalový oscilátor (18). Tyto komponenty umožňují snadné programování a široké využití desky Arduino.[12]

1.1.2 Arduino Uno R3

Arduino UNO patří mezi nejlepší desky pro začátečníky. Do dnešní doby je jedna z nejznámějších a nejpoužívanějších desek. Taky je to první Arduino které začalo používat komunikaci pomocí USB. I když jsou si tyto dvě verze velmi podobné, Arduino UNO R3 přináší několik vylepšení.



Obrázek 3: Deska Arduino Uno R3 [11]

Novější verze přináší lepší resetovací obvod, toho bylo docíleno přidáním $1k\Omega$ pro zvýšení stability resetu. Má taky jiný USB převodník kdy starší Uno používalo FTDI čip a R3 používá ATmega16U2. Vylepšenou komunikaci mezi moduly (shieldy) a lepší podpora v novějších verzích Arduino IDE. [11]

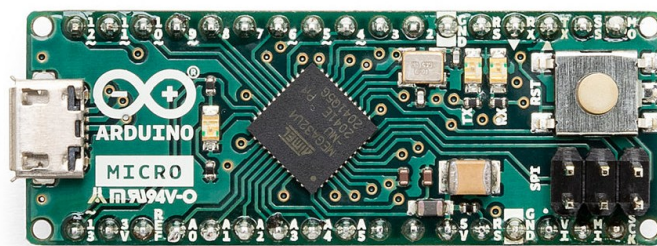
Podstatný rozdíl mezi procesorem ATmega328P a procesorem starší generace je nižší spotřeba energie v režimu spánku. Tato deska je po vzhledové stránce téměř totožná s původním modelem UNO. Zachovává stejné rozměry i rozmístění hlavních komponentů což znamená že všechny shieldy z původního modelu je možné použít i na Uno R3. Jediný viditelný rozdíl jsou nové piny SCL a SDA a jiný USB převodník kdy starší Uno používalo FTDI čip a R3 používá ATmega16U2. [11]

Procesor	ATmega328P
Provozní napětí	5V
Vstupní napětí (doporučené)	7-12V
Vstupní napětí (limit)	6-20V (většinou stejný u všech boardů)
Digitální I/O piny	14
PWM digitální I/O piny	6
Analogové vstupní piny	6
Maximální proud na jeden I/O pin	20 mA
SRAM	2 KB
EEPROM	1 KB
Flash paměť	32 KB (0,5KB použito pro bootloader)
Taktovací frekvence	16 MHz

Tabulka 2: Parametry desky Arduino Uno R3 [11]

1.1.3 Arduino Mikro

I když se jedná o jeden z nejmenších modelů v Arduino rodině, Arduino Micro dokazuje, že velikost není všechno. Deska má kompaktní rozměry přibližně 48×18 mm a váží okolo 13 gramů, což z ní činí ideální volbu pro projekty s omezeným prostorem, například v oblasti nositelné elektroniky nebo kompaktních řídicích systémů. Díky tomu se hodí na velkou škálu projektů s omezeným prostorem. [11]



Obrázek 4: Deska Arduino Micro [11]

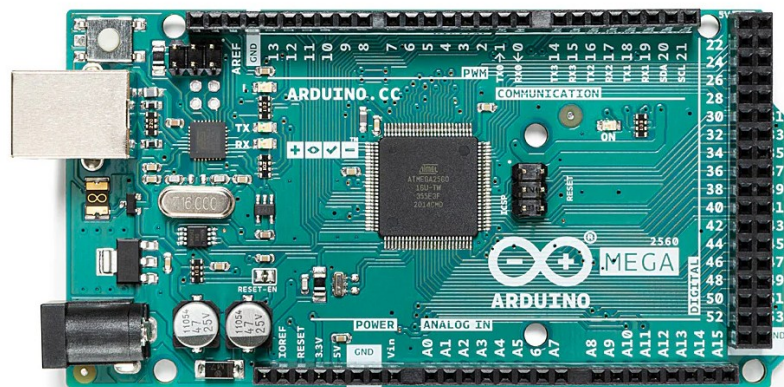
I přes jeho velikost si stále zachovává plnou funkčnost díky jeho výkonnému procesoru ATmega328 který můžeme najít v jeho větším bratříčkovi Arduino Uno. Jeho přednost oproti desce Arduino Mini je jeho USB rozhraní které na Mini desce nenajdeme. Tato vlastnost usnadňuje propojení mezi deskou a počítačem bez potřeby externího USB-to-serial převodníku. [11]

Procesor	ATmega32U4
Provozní napětí	5V
Vstupní napětí (doporučené)	7-12V
Digitální I/O piny	20
PWM digitální I/O piny	7
Analogové vstupní piny	12
Maximální proud na jeden I/O pin	20 mA
SRAM	2.5 KB
EEPROM	1 KB
Flash paměť	32 KB (4KB použito pro bootloader)
Taktovací frekvence	16 MHz

Tabulka 3: Parametry Desky Arduino Micro [11]

1.1.4 Arduino Mega 2560 rev3

Arduino Mega 2560 patří mezi nejvýkonnější desky které Arduino kdy přidalo na trh. Díky jejímu výkonu, větší paměti a velkému počtu vstupů a výstupů je ideální volbou pro náročnější projekty jako jsou robotika, automatizace a složitější datové zpracování. [11]



Obrázek 5: Arduino Mega 2560 rev3[11]

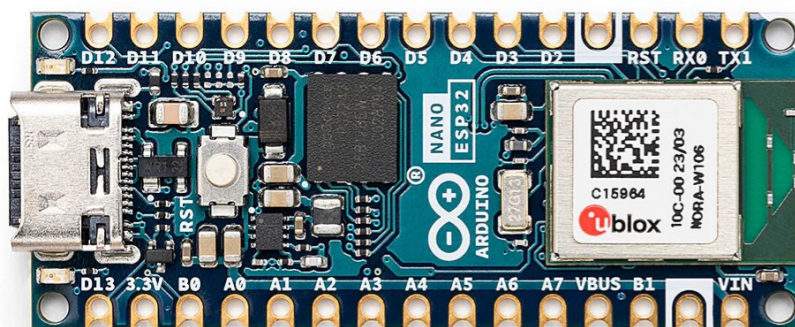
V průběhu vývoje prošla mnoha revizemi a její nejnovější model Arduino Mega 2560 rev3 přináší několik vylepšení. Oproti předchůdci obsahuje výkonnější USB převodník ATmega16U2. Další změna podobně jako u Uno R3 zvýšili odpor na Reset pinu pro stabilitu a má nový IOREF pin který umožňuje zjistit napájecí napětí desky. [11]

Procesor	ATmega2560
Provozní napětí	5V
Vstupní napětí (doporučené)	7-12V
Digitální I/O piny	54
PWM digitální I/O piny	6
Analogové vstupní piny	16
Maximální proud na jeden I/O pin	40 mA
SRAM	8 KB
EEPROM	4 KB
Flash paměť	256 KB (8KB použito pro bootloader)
Taktovací frekvence	16 MHz

Tabulka 4: Parametry desky Arduino Mega rev3. [11]

1.2 Arduino Nano ESP32

Při výběru součástek pro tuto práci jsem nakonec zvolil Arduino Nano ESP32 z důvodu integrovaných bezdrátových periférií jako je Wi-Fi a Bluetooth což umožňuje snadné připojení k bezdrátovým sítím a zařízení samotnému. Mezi další výhody patří jeho kompaktní rozměr, výkonný procesor ESP32-S3 který dosahuje taktovací frekvence až 240MHz a její podpora různých programovacích prostředí. Díky tomu se tato deska stala hlavním kandidátem pro vývoj mé bakalářské práce. [11]



Obrázek 6: Arduino Nano ESP32. [11]

Pro upřesnění v mé práci jsem použil model LaskaKit ESP32-DEVKit s mikrokontrolérem ESP32-WROOM. Byla zvolena kvůli nízké spotřebě energie a podpoře napájení z Li-Ion baterie pro možnost funkce i v místech bez možnosti připojení k elektrické síti[13]

Procesor	ESP32-S3
Provozní napětí	3,3V
Vstupní napětí (doporučené)	5V
Digitální I/O piny	14
PWM digitální I/O piny	16
Analogové vstupní piny	8
Maximální proud na jeden I/O pin	40 mA
SRAM	512 kB
ROM	384 kB
Flash paměť	128 Mbit (16MB)
Taktovací frekvence	240 MHz

Tabulka 5: Parametry desky Arduino Nano ESP32. [11]

2 BEZPEČNOSTNÍ ZÁMKY

Mechanické zábranné systémy hrají klíčovou roli při ochraně objektů před neoprávněným vstupem či krádežemi. Jejich hlavním úkolem je vytvořit fyzickou bariéru, která zabrání či aspoň znesnadní průnik neoprávněné osoby do prostoru chráněného těmito systémy. Mezi hlavní objekty co využívají tyto systémy patří domácnosti, firmy i státní instituce, kde je důležité zajistit bezpečnost majetku a citlivých informací.

Historicky se lidé snažili chránit svůj majetek různými způsoby například už ve starověkém Egyptě byly používány primitivní dřevěné zámky. Ve středověku se rozšířila výroba kovových, kombinačních zámků a trezorů, což vedlo k větší odolnosti proti zlodějům. S průmyslovou revolucí se objevily první sériově vyráběné bezpečnostní zámky, jako například cylindrický zámek, který vynalezl Linus Yale Jr. v 19. století a dodnes patří mezi nejpoužívanější typy. [14,15]



Obrázek 7: Příklad bezpečnostních dveří. [16]

Mezi základní prvky mechanických zábranných systémů dnes patří například bezpečnostní dveře, mříže na oknech, trezory nebo bezpečnostní zámky s vyšší odolností proti vylomení. V praxi se často kombinují s elektronickými systémy, například s alarmy nebo kamerovými systémy pro ohlášení neoprávněného vniknutí. Běžným příkladem hybridního zabezpečení je elektronický zámek s čtečkou otisků prstů, který kombinuje mechanickou ochranu s moderní technologií. Taková řešení jsou čím dál populárnější například v hotelích nebo firmách, kde se místo klasických klíčů využívají přístupové karty. [14]

V poslední době se zvedla poptávka po kvalitním zabezpečení hlavně v komerčním sektoru. Banky masivně investují do trezorových systémů a průmyslové podniky a různé firmy dost kombinují tradiční mechanické prvky s modernějšími chytrými technologiemi. Díky tomu jsou ty objekty dostatečně bezpečně chráněné vůči pokročilými metodami vloupání jako je lockpicking. [14]

Velmi důležitou roli v tom hrají normy a certifikace. Například normy ČSN EN 1627-1630 stanovují minimální odolnost těchto zabezpečovacích prvků vůči vloupání anebo norma EN1143-1 se zaměřuje na bezpečnost trezorů. Tyto normy velice napomáhají zákazníkům a uživatelům si vybrat produkt který splní jejich bezpečnostní požadavky. [14]

Celkově lze říci, že mechanické zábranné systémy zůstávají důležitou součástí bezpečnostních opatření a často tvoří první obrannou linii proti neoprávněnému vniknutí. V kombinaci s moderní elektronikou dokážou výrazně zvýšit úroveň ochrany a snížit riziko krádeží.

2.1 Elektronické zámky

Pro vyšší bezpečnost se v dnešní době instalují nejen lepší mechanické zámky ale též elektronické zámky. Většinou fungují na principu kombinace tradičního mechanického zámku jenž je ovládaný pomocí moderního elektronického rozhraní. Když je do systému zadán správný přístupový kód či heslo tak elektrický signál aktivuje malý motor nebo elektromagnet, který uvolní mechanismus v zámku a otevře dveře. Tento systém se může lišit v provedení například místo klávesnice na zapsání kódu může být použita RFID čtečka nebo biometrická čtečka otisku prstu nebo dokonce telefon s NFC technologií. Každá z těchto metod má své výhody i nevýhody [17]



Obrázek 8: Příklad elektronického zámku na RFID čip. [17]

2.1.1 Zámky na kód

Místo zámku nalezneme na dveřích či stěně vedle dveří integrovanou klávesnici, po zadání správného kódu se pošle signál do malého motorku či elektro magnetu, který uvolní zámkový mechanismus a to umožní otevření dveří. Je to populární volba vzhledem k její jednoduchosti, nevyžaduje fyzický klíč, u kterého hrozí možnost ztráty a snadné možnosti změny kódu. [17]



Obrázek 9: Elektronický zámek s kódem.

Mezi hlavní nevýhody patří že vás někdo může sledovat při zadávání kódu a získat tak možnost pro neoprávněný vstup. Tomuto se dá zamezit modernější technologií známou jako “falešný kód“, která umožňuje zadání několika falešných čísel či znaků před nebo po správném kódu čím se zabrání odpozorování správného hesla. Další nevýhoda není sice tak častá ale může se stát že při změně kódu se uživatel překlepne a může to znamenat trvalou ztrátu přístupu. V praxi se běžně používají v hotelech, kancelářských budovách nebo menších firmách. [17,19]

2.1.2 RFID zámky

Pracují na principu bezkontaktní identifikace kdy uživatel přiloží RFID čip, kartu, náramek či klíčenku k čtečce, která pomocí unikátního identifikačního čísla ověří oprávnění a uvolní zámek. Fungují velmi rychle a jednoduše se používají. Existují dva hlavní typy těchto systémů. První typ je “pasivní“ které nemají vlastní zdroj energie a aktivují se až po přiložení čipu či čtečky. Druhý typ je “aktivní“ který má vestavěnou vlastní baterii a funguje na delší vzdálenosti. Tato metoda je obzvláště oblíbená hlavně v budovách s vysokou návštěvností jako jsou hotely, posilovny, koupaliště či kanceláře. [18]



Obrázek 10: Elektronický zámek na RFID čip.

Největší nevýhoda je možnost kopírování či naklonování toho čipu či karty, tomu se dá ale zamezit pokročilejšími bezpečnostními protokoly, jako je šifrování AES nebo technologie rolling code, která při každém použití vygeneruje nový kód což znemožňuje zkopírování a naklonování přístupové karty. [18]

V dnešní době má i většina telefonů zabudované NFC, což umožňuje využití vlastního mobilního telefonu místo čipu pro bezpečnější ověření. NFC navíc často vyžaduje další ověření uživatele, jako například heslo telefonu nebo ověření pomocí biometrie (rozpoznání obličeje, otisk prstu). [18]

2.1.3 Biometrické zámky

Biometrické zámky využívají k ověření uživatele jeho vlastní unikátní osobní údaje jako jsou otisk prstu, rozpoznání obličeje nebo skenování duhovky. Tato technologie nabízí jednu z nejvyšších úrovní bezpečnosti jelikož biometrické údaje jsou pro každého jedince jedinečné a jejich padělání je extrémně obtížné. [20]

Mezi jejich největší výhody patří jejich rychlost, pohodlnost a hlavně bezpečnost, ale s tím taky přichází nevýhody jako jsou mnohem vyšší pořizovací náklady, náročnější údržba a jsou hodně citlivé na vnější faktory které ovlivňují jejich správnou funkci jako jsou špinavé či poraněné prsty v zámcích na otisk prstu. V rozpoznání obličeje sebemenší změny ve vzhledu, jako jsou vousy či změna účesu, mohou vyvolat problémy s identifikací. [20]

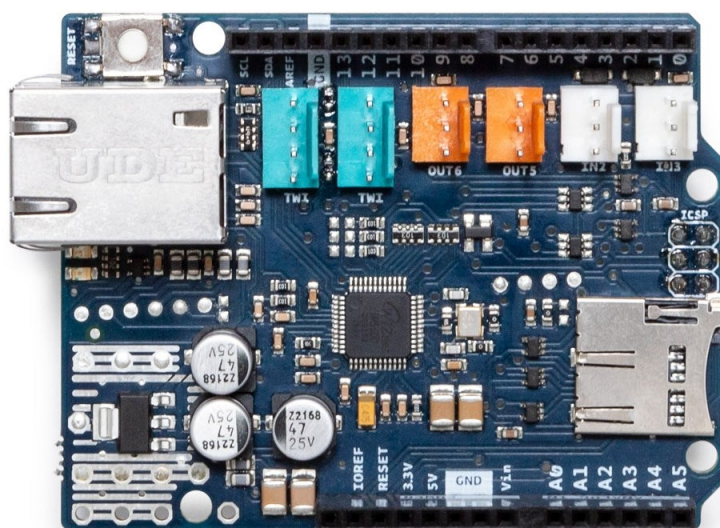


Obrázek 11: Sejf s biometrickým zámkem (otisk prstu).

Vzhledem k těmto nárokům se instalují hlavně v místech kde je kladen důraz na maximální bezpečnost, například v laboratořích, vládních institucích, novodobě i v bankách a datových centrech. Pro nejvyšší bezpečnost se často kombinuje více biometrických metod nebo je doplňují bezpečnostní prvky jako RFID technologie nebo PIN kódy. [20]

3 ROZŠIŘUJÍCÍ MODULY (SHIELDS)

Shieldy jsou přídatné moduly speciálně navržené pro rozšíření funkcionality vývojových desek Arduino. Připojují se přes hlavičkové piny Arduina, což umožňuje jednoduchou integraci nových funkcí bez nutnosti složitého zapojování vodičů a externích součástek. Každý z těchto modulů má svou určitou funkci jako je například připojení k internetu (Ethernet shield), bezdrátovou komunikaci (Bluetooth nebo Wi-Fi shield). [11,21]

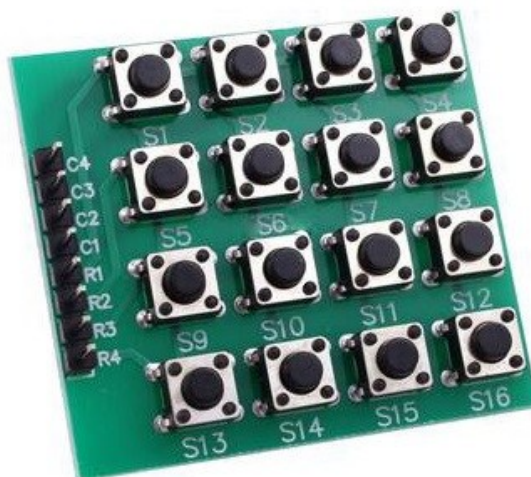


Obrázek 12: Arduino Ethernet Shield 2. [1]

K jejich připojení se využívají různé kombinace pinů na Arduino desce. Některé moduly využijí pouze několik digitálních či analogových pinů, takže zbývající můžou být volně použity pro další komponenty. Další moduly jako LCD shieldy s tlačítky můžou využít většinu pinů a znemožnit připojení dalších modulů. Proto některé z těchto shieldů obsahují průchozí konektory které umožňují připojení více modulů na stejná rozhraní. S touto možností je však nutné si dávat pozor na konflikt mezi jednotlivými moduly. [11,21]

3.1 Matricová klávesnice

Membránová klávesnice je vstupní zařízení, které se skládá z tlačítek uspořádaných do řádků a sloupců. V některých případech mohou být předem naprogramovaná a můžou mít už předem definované funkce, nebo mohou umožňovat vlastní přiřazení funkcí k jednotlivým tlačítkům. [11,22]



Obrázek 13: 4x4 Matricová klávesnice k Arduino. [22]

Funguje na principu řádkového a sloupcového skenování. Řídící mikrokontrolér aktivuje jednotlivé řádky a sleduje který sloupec vykazuje signál což pomáhá určit které z těch tlačítek bylo stisknuto. Tento způsob efektivně detekuje stisk tlačítka bez nutnosti používání velkého počtu pinů. [11,22]

Pin na klávesnici	Pin na Arduino	Funkce
ROW1	D9	První řádek
ROW2	D8	Druhý řádek
ROW3	D7	Třetí řádek
ROW4	D6	Čtvrtý řádek
COL1	D5	První sloupec
COL2	D4	Druhý sloupec
COL3	D3	Třetí sloupec
COL4	D2	Čtvrtý sloupec

Tabulka 6: Adresace pinů mezi klávesnicí 4x4 a Arduinem. [22]

3.2 RFID modul

Modul PN5180 slouží pro bezdrátovou identifikaci a práci s RFID i NFC technologiemi. Čtečka funguje na frekvenci 13,56 MHz a podporuje více standardů, jako jsou ISO 14443, ISO 15693 nebo NFC Forum Type 4. Využívá elektromagnetické pole k aktivaci pasivních RFID čipů, které následně vyšlou své unikátní identifikační číslo. [33,34]



Obrázek 14: RFID modul PN5180 [34]

Funguje na principu vytváření elektromagnetického pole o frekvenci 13.56 MHz a tím po přiložení indukuje napětí v pasivním RFID čipu či kartě. Tento aktivovaný čip vyšle své unikátní identifikační číslo, které se následovně odešle zpět do mikrokontroléru k dalšímu zpracování. [33]

PN5180 komunikuje přes rozhraní SPI a je známý svou odolností vůči elektromagnetickému rušení. Dokáže se automaticky přizpůsobit okolním podmínkám, což zajišťuje spolehlivou detekci i v náročnějším prostředí. Výhodou tohoto modulu je také možnost čtení i zápisu dat na kompatibilní čipy a karty. Součástí balení bývá testovací karta a čip podle standardu ISO 15693, které se dají využít pro ověření funkčnosti čtečky. [33,34]

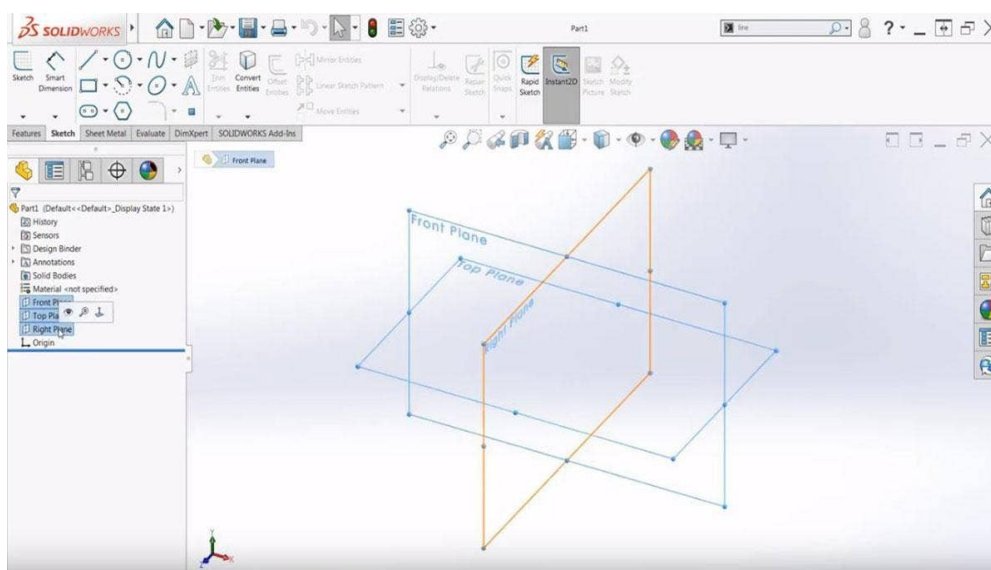
Pin na Arduinu	Název pinu	Funkce
1	Vcc	Napájení
2	RST	Reset
3	Ground	Uzemnění
4	IRQ	Přerušení
5	MISO/SCL/Tx	Příjem dat
6	MOSI	Odesílání dat
7	SCK	Hodinový signál pro SPI
8	SS/SDA/Rx	Výběr modulu

Tabulka 7: Adresace pinů mezi RFID modulem RC522 a Arduinem. [25]

4 DOPLŇUJÍCÍ SOFTWARE

4.1 Program Solidworks

Solidworks je pokročilý CAD software, který nabízí možnost vytváření a simulace 3D modelů, včetně návrhu mechanických a elektronických zařízení a sestav. Tento software je díky jeho flexibilitě a širokému spektru využití široce využíván v různých průmyslových odvětvích, jako je strojírenství, automobilový průmysl, elektronika a mnoho dalších. [26]

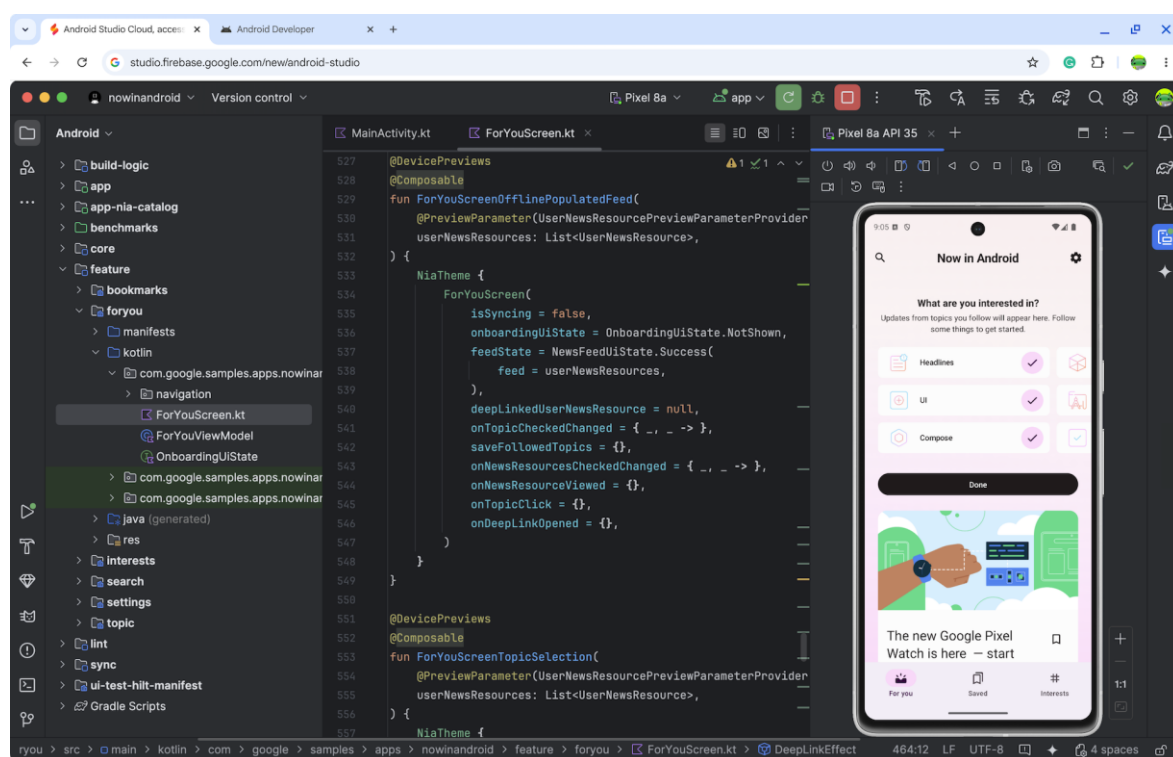


Obrázek 15: Solidworks User Interface

Uživatelé mají možnost navrhovat detailní modely součástek a následně je skládat do celkových sestav. SolidWorks poskytuje uživatelům širokou škálu nástrojů pro simulace, které jim umožňují otestovat, jak se jejich návrhy budou chovat v reálných podmínkách. Program neumožňuje pouze provádět mechanické analýzy, ale také nabízí simulace tepelného chování a dalších fyzikálních vlastností modelů, což je nezbytné pro optimalizaci návrhu před výrobou. Také se nabízí možnost návrhu desek plošných spojů, které lze též zahrnout do celkového návrhu. Tato možnost integrace zlepšuje efektivitu vývojového procesu a umožňuje testování spolupráce mezi mechanickými a elektronickými součástmi. Díky tomu se celý proces návrhu i výroby stává jednodušší a rychlejší, což vede k úspoře času a snížení nákladů. [26]

4.2 Android Studio

Android studio je vývojové prostředí pro vývoj aplikací na platformu Android. S touto platformou na trh přišel Google v roce 2013 kdy byl představen na Google I/O konferenci. Tato platforma nabízí vývojářům všechny potřebné nástroje pro vývoj aplikací. Mezi největší výhody Android Studia patří vestavěný Android Emulátor, díky kterému se dá aplikace rychle otestovat na různých zařízeních či verzích android. Tato funkce také výrazně usnadňuje ladění (debugging) aplikací. Také umožňuje jednoduchý import aplikace do telefonu, kde je možné dále testovat její funkčnost. [27]



Obrázek 16: Android Studio User Interface [27]

4.2.1 Programovací Jazyk Java

Java je jedním z nejznámějších a nejvíce používaných programovacích jazyků, který vznikl v roce 1995. Je to objektově orientovaný jazyk, který je oblíbený pro svou stabilitu a spolehlivost. Díky principu „napiš jednou, spust' kdekoli“ se aplikace napsané v Javě snadno přenášejí mezi různými platformami, aniž by bylo nutné je výrazně upravovat. Java se využívá nejen v mobilním vývoji (zejména na Androidu), ale i v serverových aplikacích, webových službách, bankovních systémech a podnikových softwarových řešeních. [27,28]



Obrázek 17: Oficiální logo Java

4.2.2 Kotlin

Kotlin je moderní programovací jazyk, který si v posledních letech získal velkou oblibu. Vyvinula ho společnost JetBrains a v roce 2017 ho Google oficiálně doporučil jako hlavní jazyk pro vývoj aplikací na Android. Kotlin se soustředí na jednodušší a přehlednější zápis kódu, což znamená, že vývojáři se vyhnou psaní opakujících se částí kódu a mohou se více soustředit na funkce aplikace. Další výhodou Kotlinu je jeho schopnost lépe chránit před běžnými chybami, jako je například tzv. „null safety“, která zabraňuje problémům způsobeným null hodnotami. [29,30]



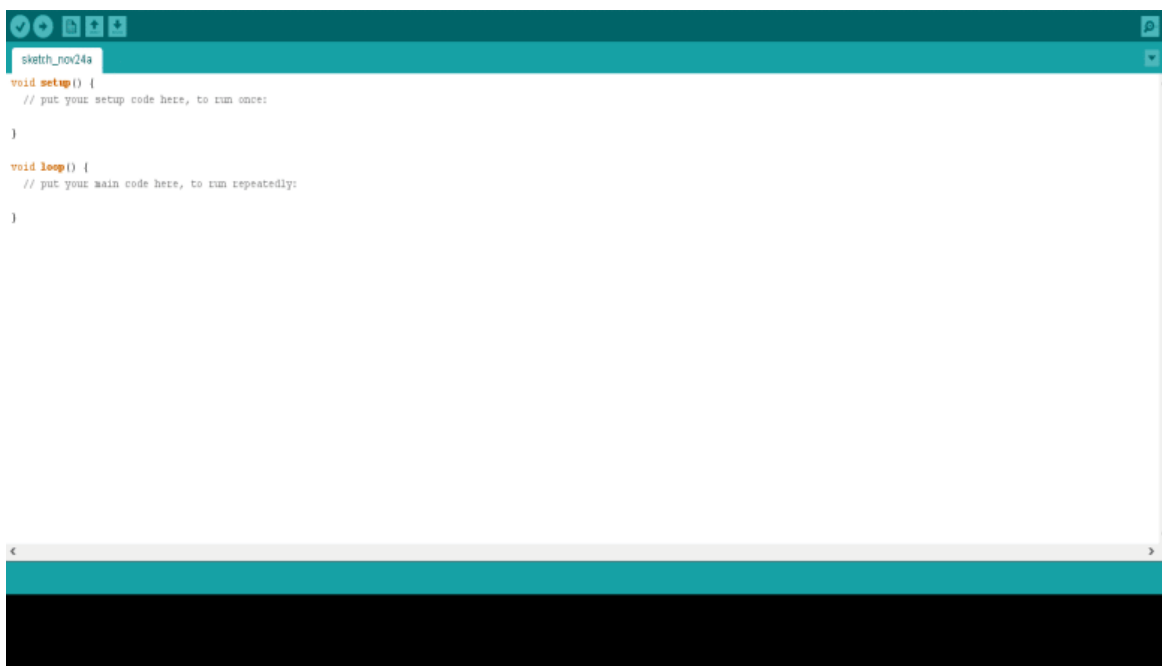
Obrázek 18: Kotlin logo

Kotlin je také plně kompatibilní s Javou, což znamená, že ho lze snadno začlenit do starších projektů napsaných v Javě. Vývojáři tak mohou přidávat nové funkce nebo postupně přepisovat části kódu. Díky těmto vlastnostem je vývoj s Kotlinem rychlejší, bezpečnější a méně náchylný k chybám. K tomu přispívá i moderní syntaxe, funkcionální prvky a velmi aktivní komunita, která jazyk neustále vylepšuje. [27,30]

4.3 Arduino IDE

Arduino IDE je vývojové prostředí, které je navrženo tak, aby bylo jednoduché na používání, což ho činí ideální pro začátečníky. I přesto je dostatečně silné pro pokročilejší uživatele. Umožňuje snadno programovat desky Arduino a pracovat s hardwarem. [2,11]

Používá jazyk wiring, což je jazyk postavený na C/C++, ale s jednodušší a srozumitelnější syntaxí, která je přizpůsobená pro práci s mikrokontroléry. Díky tomu, že wiring skrývá složitější details, mohou se uživatelé soustředit na samotnou logiku aplikace, aniž by se museli trápit technickými details. [3,11]



Obrázek 19: Arduino IDE User Interface

Arduino IDE je oblíbené i díky velké komunitě, která neustále vytváří nové knihovny a projekty, což usnadňuje práci s různými senzory, motory nebo jinými elektronickými komponentami. Tento jazyk a prostředí jsou tedy skvělé pro rychlý prototypování a experimentování s různými nápady. [2,3]

II PRAKTICKÁ ČÁST

5 NÁVRH ZÁMKU A SYSTÉMOVÉ ARCHITEKTURY

Cílem návrhu bylo vytvořit chytrý elektronický zámek s využitím platformy Arduino. Pro tento projekt bylo zvoleno Arduino Nano ESP32, díky kterému se otevřely možnosti pro efektivnější a bezpečnější odemykání. Systém byl navržen s důrazem na jednoduchost použití pro koncové uživatele, nízkou spotřebu energie což je vhodná varianta pro bateriové napájení a možností rozšíření o další funkce v budoucnu ať už se jedná o další autorizační metody nebo propojení s celou chytrou domácností.

5.1 Hardwarové vybavení

Pro správnou funkci chytrého zámku bylo potřeba vybrat vhodné komponenty aby byl zajištěn spolehlivý a bezpečný provoz celého systému. Při výběru těchto jednotlivých součástek byl hlavně kladen důraz na kompaktní rozměry, nízká spotřeba energie, snadná dostupnost a dobrá kompatibilita s vybraným Arduinem. Tyto určité komponenty se společně starají o komunikaci s okolními zařízeními, autorizaci uživatele a samozřejmě fyzické otevření.

5.1.1 Vývojová deska LaskaKit ESP32 DevKit

Základ celého projektu je vývojová deska LaskaKit ESP32 DevKit, která je postavena na čipu ESP32-WROOM-32 od firmy Espressif. I když je tato deska na pohled podobná klasickému Arduino Nano, ve skutečnosti jde o výkonnější a modernější platformu vhodnou pro chytré projekty s důrazem na bezdrátovou konektivitu. Tato deska byla zvolena především kvůli výbornému poměru mezi výkonem, spotřebou, kompaktními rozměry a integrovanou podporu bezdrátové komunikace což jsou ideální parametry pro použití při vývoji chytrého zámku. [31]

Na rozdíl od klasických desek, jako je Arduino Uno nebo Nano, nabízí ESP32 dvoujádrový procesor běžící až na 240 MHz, což umožňuje současný běh více úloh – například ovládání serva, komunikaci přes Bluetooth a zpracování dat z RFID čtečky. Samotná deska LaskaKit ESP32 DevKit navíc podporuje přímé napájení z baterie přes konektor JST, což bylo pro projekt klíčové kvůli požadavku na mobilní a bezdrátový provoz. [31]



Obrázek 20: LaskaKit ESP32 DevKit [13]

Na rozdíl od klasických Arduino desek má LaskaKit ESP32 DevKit širší podporu periférií – obsahuje velké množství GPIO pinů, podporuje ADC, DAC, PWM, I²C, SPI i UART a je plně kompatibilní s Arduino IDE, což výrazně usnadňuje vývoj.

Procesor	ESP32 (Tensilica Xtensa LX6, dual-core)
Provozní napětí	3.3V
Vstupní napětí (doporučené)	5V (USB C)
Digitální I/O piny	32
Analogové vstupní piny	16
DAC	2× 8bit
Wi-Fi	802.11 b/g/n
Bluetooth	v4.2 BR/EDR a BLE
Flash paměť	4 MB
Taktovací frekvence	Až 240 MHz
Rozměry	53 x 26 mm

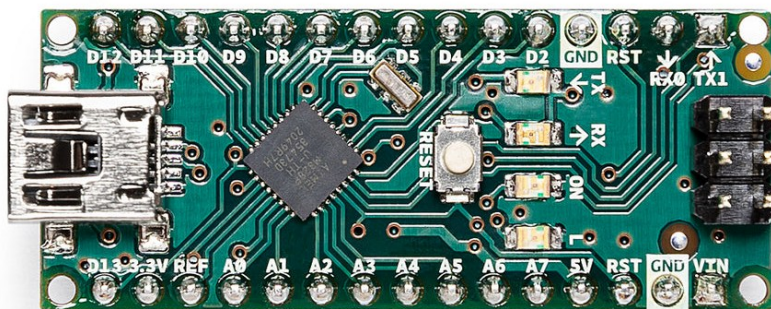
Tabulka 8: Parametry desky LaskaKit ESP32 DevKit [13,31]

ESP32 pin	Arduino Nano pin	Funkce
GPIO 17	D2	TX (ESP32 → Nano RX)
GPIO 16	D3	RX (ESP32 ← Nano TX)
GND	GND	Společná zem

Tabulka 9: Tabulka propojení ESP32 ↔ Arduino Nano (sériová komunikace)

5.1.2 Vývojová deska Arduino Nano

V rámci projektu byl kromě hlavní řídicí jednotky LaskaKit ESP32 DevKit využit také Arduino Nano, které sloužilo především pro ovládání servomotoru. Důvodem pro zařazení této druhé desky byla skutečnost, že ESP32 pracuje na napájecím napětí 3,3 V, zatímco běžná serva vyžadují pro správnou funkci řídicí signál na úrovni 5 V. [11]



Obrázek 21: Vývojová deska Arduino Nano [11]

Přímé připojení serva k výstupu ESP32 by tak mohlo vést k nestabilnímu chování nebo nespolehlivému řízení. Arduino Nano, které je plně kompatibilní s 5V logikou, proto plní roli mezičlánku mezi hlavní deskou a servem. Z ESP32 jsou do Nano přenášeny jednoduché povely například přes sériovou komunikaci nebo I²C sběrnici, na jejichž základě Arduino provádí konkrétní akce. Toto řešení zajistilo spolehlivé ovládání zámku bez nutnosti použití převodníků logických úrovní. [11]

Parametr	Hodnota
Procesor	ATmega328P
Provozní napětí	5 V
Vstupní napětí (doporučené)	7–12 V
Digitální I/O piny	22
PWM digitální I/O piny	6
Analogové vstupní piny	8
Maximální proud na I/O pin	20 mA
SRAM	2 KB
EEPROM	1 KB
Flash paměť	32 KB (z toho 2 KB bootloader)
Taktovací frekvence	16 MHz
Rozměry	45 × 18 mm

Tabulka 10: Parametry Vývojové Desky Arduino Nano [11]

5.1.3 Servo motor MG90S

Pro mechanické ovládání zámku bylo zvoleno servo MG90S, které se osvědčilo díky malým rozměrům, dostatečné síle a jednoduchému způsobu řízení. Jde o mikroservo vybavené kovovými převody, což zvyšuje jeho odolnost a prodlužuje životnost oproti běžným servům s plastovými částmi. Díky tomu je MG90S ideální pro použití tam, kde může docházet k většímu mechanickému zatížení – například při pohybu západky zámku.



Obrázek 22: Servo motor MG90S [32]

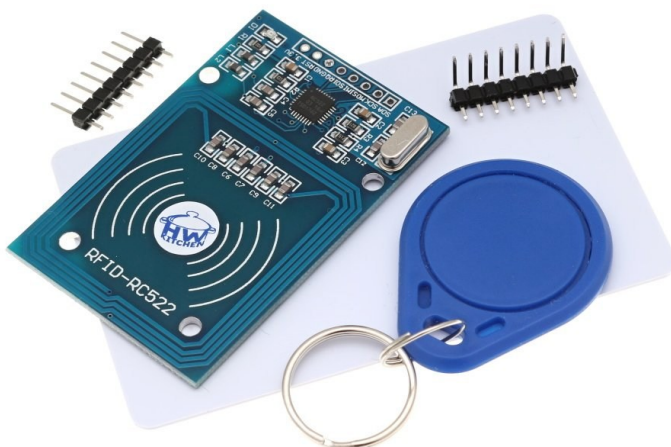
Servo pracuje s napětím v rozmezí 4,8 až 6 voltů a zvládne se otočit přibližně o 180 stupňů. Jeho poloha se nastavuje pomocí PWM signálu, kde délka signálu určuje požadovanou pozici. K připojení slouží tři vodiče: červený pro napájení, hnědý nebo černý pro zem a oranžový (nebo žlutý) pro řídicí signál, který se běžně připojuje přímo na jeden z výstupních pinů Arduino desky. [32]

Vodič (barva)	Signál	Pin na Arduino Nano
Červený	Napájení (+5V)	5V
Hnědý/Černý	Zem (GND)	GND
Oranžový	Řídicí signál	D6

Tabulka 11: Tabulka zapojení servomotoru MG90S

5.1.4 RFID čtečka RC522

Pro bezkontaktní autorizaci uživatele byla zvolena čtečka RC522, která nabízí moderní a spolehlivé řešení pro práci s RFID. Díky podpoře čtení i zápisu a široké kompatibilitě s různými typy čipů se hodí i pro náročnější aplikace, jako je právě chytrý zámek. Čtečka komunikuje přes SPI rozhraní, takže ji lze snadno připojit k mikrokontrolérům typu ESP32. Výhodou tohoto modulu je i vysoká odolnost vůči elektromagnetickému rušení a schopnost automaticky se přizpůsobit okolním podmínkám, což zaručuje stabilní a bezproblémový provoz i v méně ideálním prostředí. [24,25]



Obrázek 23: RFID čtečka RC522 [24]

Pin na MFRC522	Signál	Pin na Arduino Nano
VCC	Napájení (3.3V)	3.3V
GND	Zem (GND)	GND
RST	Reset	D9
SDA (SS)	Slave Select	D10
MOSI	Master Out	D11
MISO	Master In	D12
SCK	Hodinový signál	D13

Tabulka 12: Tabulka zapojení RFID čtečky MFRC522

RC522 je optimalizován pro práci s čipy na frekvenci 13,56 MHz podle standardu ISO/IEC 14443 typu A. Díky nízké spotřebě energie a kompaktním rozměrům se hodí pro použití v přenosných nebo vestavěných systémech. Modul umožňuje spolehlivé čtení UID kódu a také zápis základních dat do kompatibilních RFID tagů. Jeho obliba v komunitě tvůrců a vývojářů je podpořena bohatou dokumentací a množstvím dostupných knihoven pro různé platformy, včetně Arduino a ESP32. [24,25]

5.1.5 Napájení

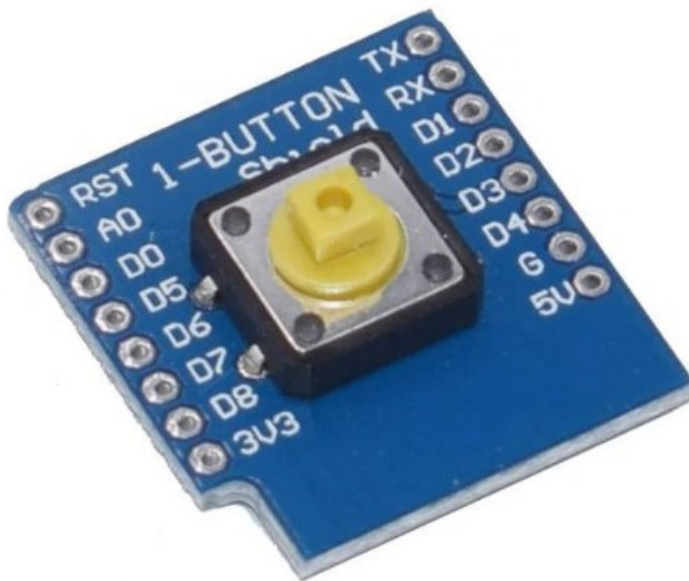
Systém je napájen přes USB-C konektor, který zároveň slouží i k programování. Jako záloha v případě výpadku napájení slouží malá Li-Po baterie s kapacitou 500 mAh, která umožní krátkodobý provoz například pro nouzové odemknutí. Díky jednoduchému zapojení a automatickému nabíjení je řešení praktické a spolehlivé.



Obrázek 24: Baterie GeB LiPol 503035 [35]

5.1.6 1 - Button Shield

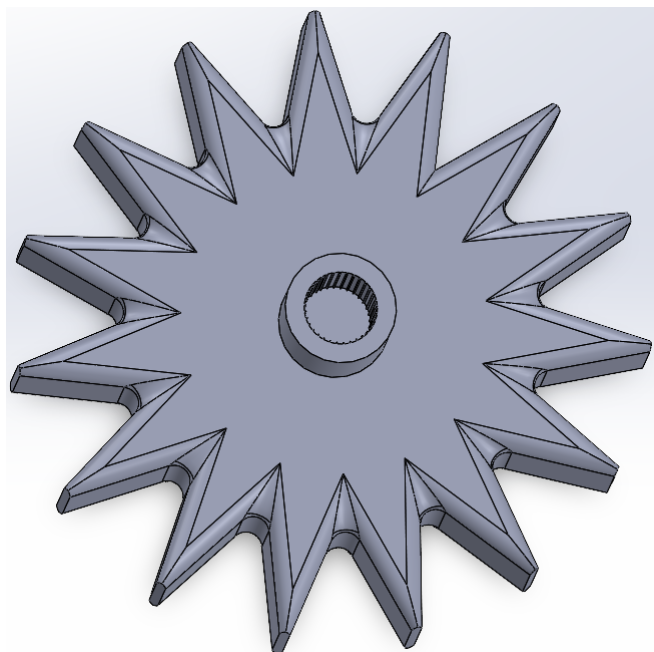
Do obvodu bylo přidáno samostatné tlačítko, jelikož vývojová deska LaskaKit ESP32 DevKit neobsahuje integrované BOOT tlačítko. a bez něj nebylo možné do mikrokontroléru nahrávat kód, což komplikovalo průběh vývoje a vyžádalo si úpravu zapojení.



Obrázek 25: Wemos D1 mini shield s tlačítkem [38]

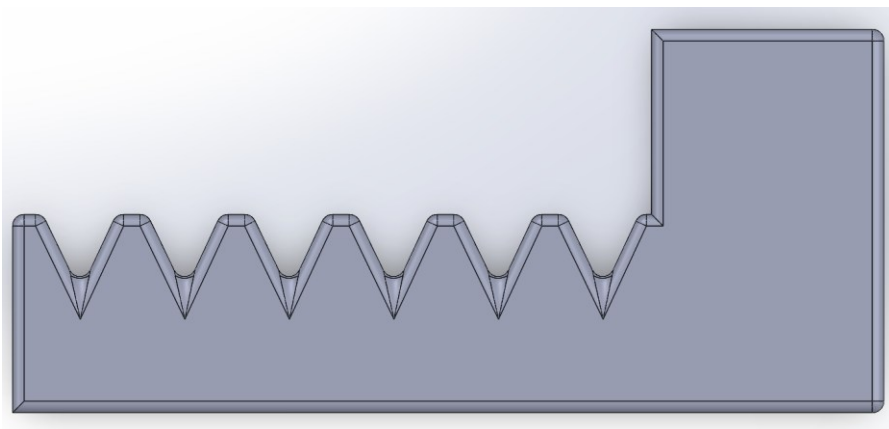
5.2.1 Návrh prototypu zámku

V samotném jádru tohoto zámku se nachází ozubené kolo které bylo navrženo tak aby použitý servomotor s rozmezím 0 až 180° nemusel zacházet až do úplných limitů při odemykání a umožnilo tak plynulý pohyb aniž by došlo k poškození samotného servomotoru.



Obrázek 27: 3D model ozubeného kola

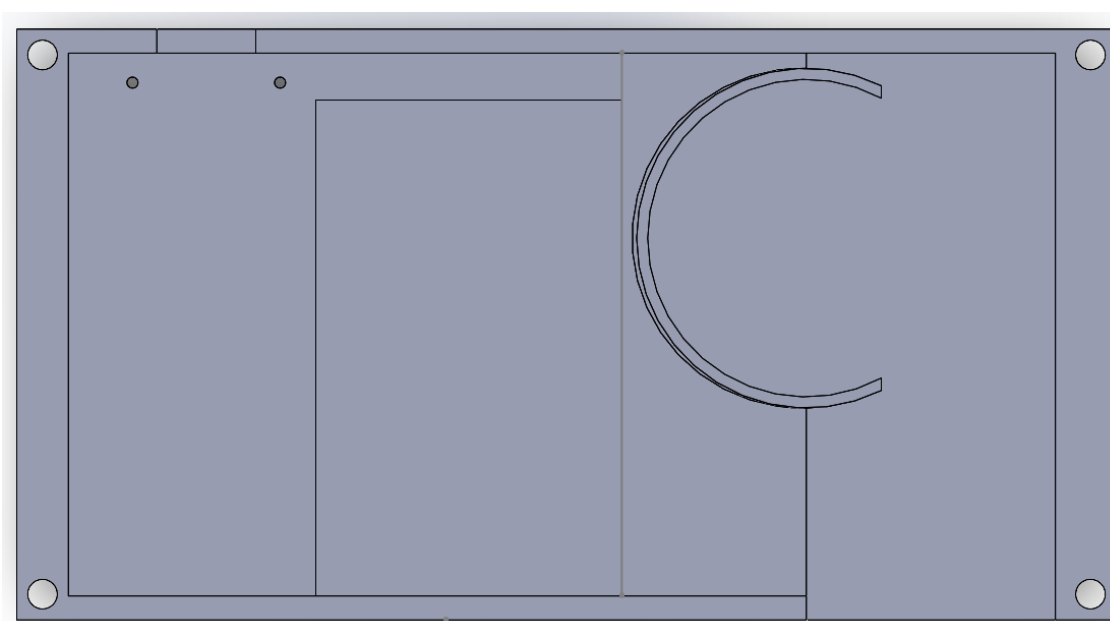
Celý mechanismus zámku je založen na klasickém západkovém systému. Samotná západka je klíčová součást zámku. Byla navržena tak aby do sebe zoubky z kola i západky zapadly bez problémů



Obrázek 28: Model výsuvné západky

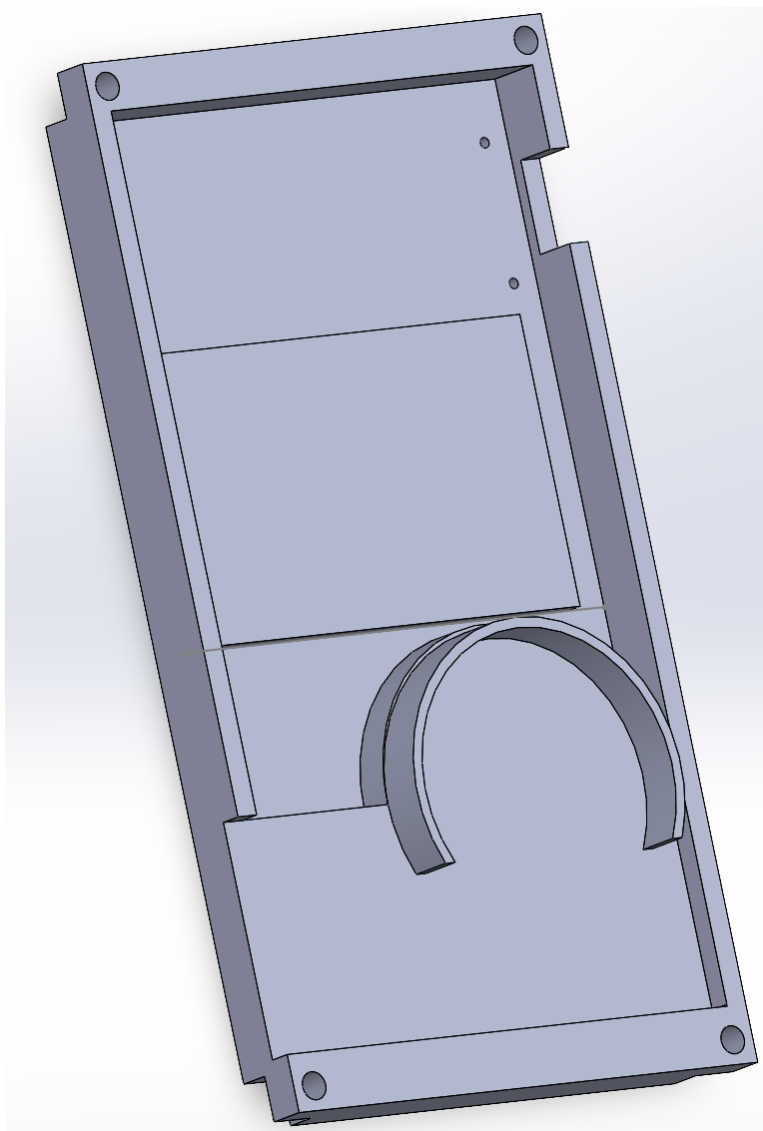
5.2.2 Schránka mechanismu zámku.

Všechn tento mechanismus je i s elektronickými komponenty uložen uvnitř ochranné schránky která tvoří vnější plášť celého zařízení. Je navržena tak aby se vlezly všechny komponenty bez problémů a jejich kabeláž nebyla poškozená při zavření schránky. Hlavní je samozřejmě zafixování komponentů aby se co nejvíce omezil možný pohyb ve schránce. Při silnějším zabouchnutí dveří by mohla západka vyskočit a zaseknout se a při pokusu o odemčení by se mohlo ozubené kolečko polámat nebo se také zaseknout o zaseknutou západku a poškodit servomotor.



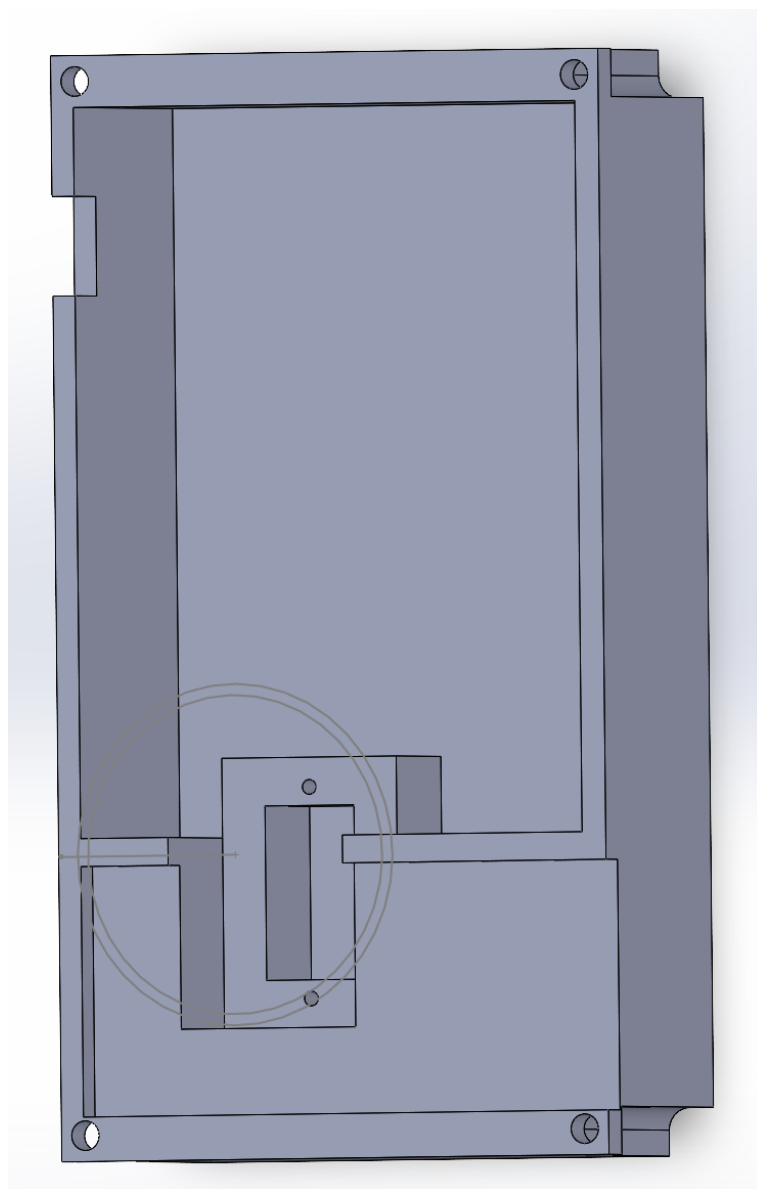
Obrázek 29: Spodní část ochranné schránky. Horní pohled

Na obrázku je zobrazena spodní část ochranné schránky. Pokud si tento model projdeme postupně zleva můžeme vidět otvory ty slouží pro fixaci Vývojové desky LaskaKit ESP32 DevKit. Na ní pomocí oboustranné pásky bude připevněna Vývojová deska Arduino Nano z důvodu ušetření místa. Nahoře nad těmito otvory je větší otvor který slouží pro napájení. Kvůli přidání druhé Arduino desky by bylo kvůli nedostatku místa tam umístit druhou baterii.



Obrázek 30: Spodní část ochranné schránky. Pohled z úhlu

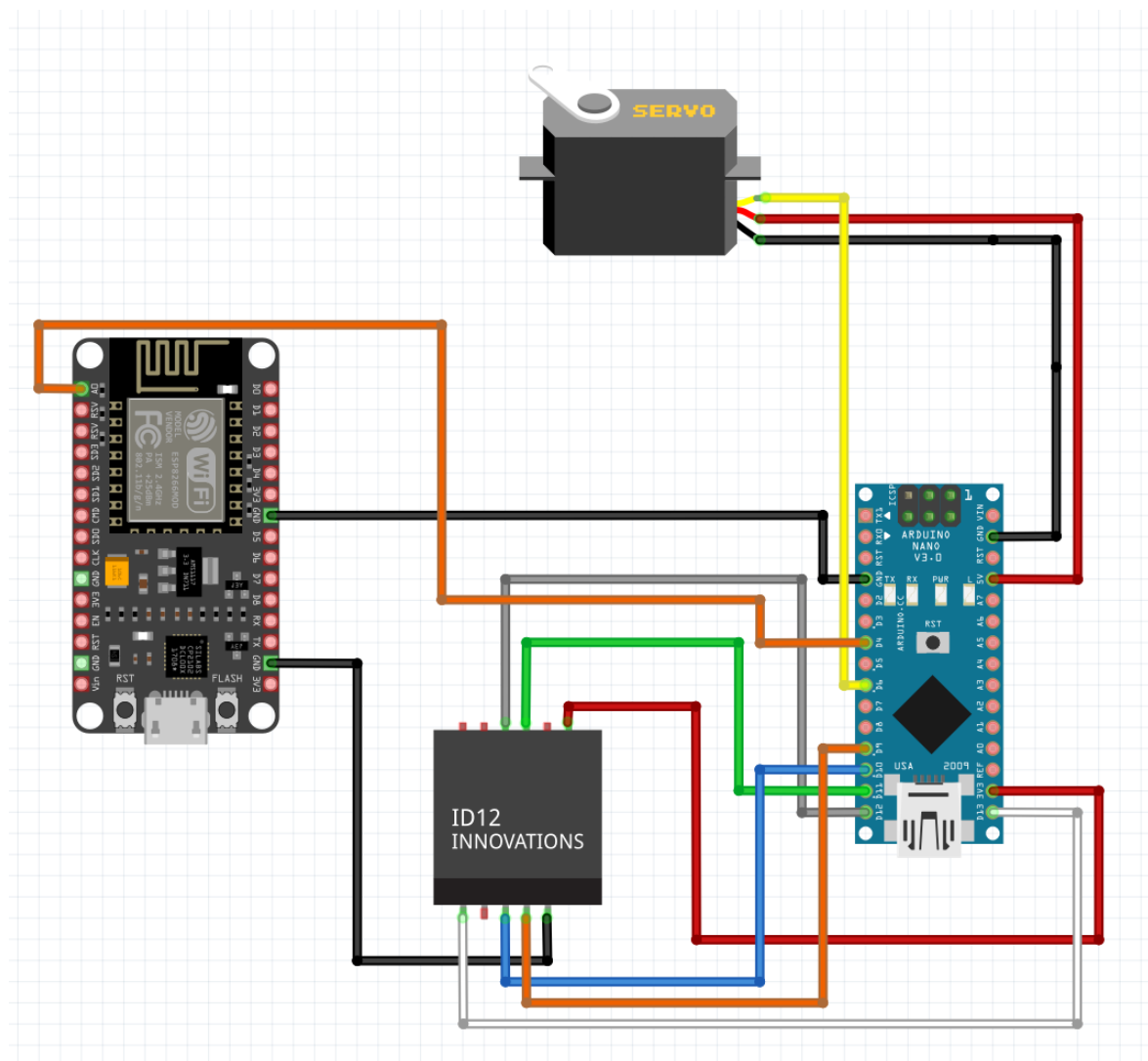
Poté tam je vyvýšená platforma pro RFID čtečku. Ta stejně jako Arduino Nano bude upevněna pomocí oboustranné pásky pro zamezení pohybu po celém modelu. Napravo je možné vidět ochranný půlkruh který slouží na zamezení pohybu ozubeného kolečka mimo jeho místo a díky tomu se zamezí poškození ať už servomotoru nebo samotné západky která patří přímo pod tento půlkruh a vysouvá se doprava při zamčeném stavu.



Obrázek 31: Spodní část ochranné schránky. Pohled z úhlu

Horní část ochranné schránky slouží jako víko celého zařízení a zároveň jako druhý díl, který uzavírá vnitřní prostor s elektronikou a mechanismem zámku. Je navržena tak, aby přesně doléhala ke spodní části a vytvořila pevný, kompaktní celek. Na spodu lze vidět vybrání pro západku, která zde má dostatek prostoru pro pohyb při zamykání a odemykání. Dále je zde připraven otvor pro servnomotor, který zajistí dostatečnou fixaci a omezí pohyb serva. Díky přesně definovaným vnitřním hranám a výřezům je zajištěno, že jednotlivé komponenty při zavření zůstanou na svém místě a nedochází ke kontaktu s víkem, což snižuje riziko mechanického poškození při provozu zařízení.

5.3 Schéma zapojení komponentů



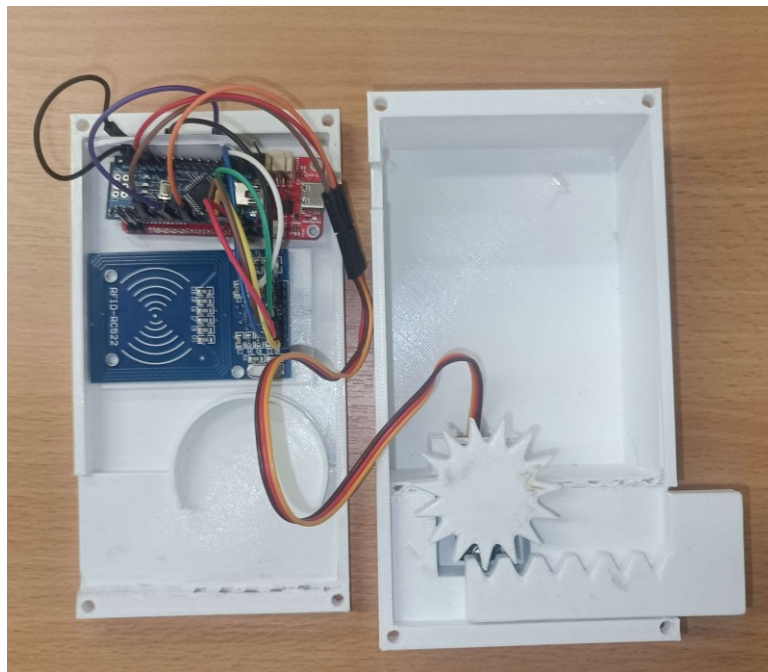
Obrázek 32: Schéma zapojení

Na obrázku je znázorněné schéma zapojení celého systému. RFID čtečka, konkrétně model MRC522, byla ve schématu nahrazena komponentem ID12, protože ve vizualizačním nástroji Fritzing není RFID čtečka MRC522 dostupná. Funkčně se však jedná o podobné zařízení, které slouží ke čtení dat z RFID čipů a karet.

Čtečka předává data Arduino Nano, které zajišťuje jejich vyhodnocení. Pokud je přiložen správný čip/karta, Arduino vyšle signál k servomotoru, který odemčie zámek.

ESP32 slouží hlavně pro bezdrátovou komunikaci. Umožňuje odesílat data do cloudu, na server nebo přijímat povely přes Wi-Fi. ESP32 a Arduino Nano jsou spolu propojeny pomocí sériové komunikace. Obě zařízení zároveň sdílí společnou zem, což je důležité pro správnou funkčnost celého obvodu.

5.4 Hotová sestava chytrého zámku



Obrázek 33: Kompletní sestava elektronického zámku

Na obrázku je vidět finální podoba chytrého zámku, rozložená na dvě části. Ve spodní části se nachází veškerá elektronika: RFID čtečka, mikrokontrolér a napájecí modul. Horní část pak obsahuje převodový mechanismus s ozubeným kolem a západkou, který pohání servomotor. Vše je zabudováno do pouzdra vytištěného na 3D tiskárně, které bylo navrženo přesně pro potřeby tohoto projektu.

6 PROGRAMOVÁ ČÁST ZÁMKU

Tato kapitola je věnována softwarové části, která zajišťuje komunikaci mezi všemi komponenty a zajišťuje funkčnost chytrého zámku. Kód byl psán v jazyku wiring ve vývojovém prostředí Arduino IDE. Tento program se dělí na dvě části z důvodu použití vícero Arduino desek. Hlavní část programu běží na desce Arduino ESP32 která obstarává hlavní funkce chytrého zámku jako komunikaci s uživatelem pomocí Arduino webu nebo výměnu dat s druhou deskou Arduino Nano. Na druhé desce je pak realizována hardwarová komunikace, konkrétně se servomotorem a RFID čipem.

6.1 Použité knihovny

V programu byly použity různé knihovny bez kterých by samozřejmě nefungovala komunikace s komponenty. Tyto knihovny jsou v podstatě hotové balíčky kódu které existují pro usnadnění práce programátorům místo toho aby museli veškerý kód psát od začátku. V této práci byla využita knihovna “Wifi.h“, Ta umožnila nastavit desku Arduino ESP32 jako přístupový bod. To znamená že je možné ovládat zámek bez potřeby internetu dokud je uživatel v dosahu Wi-Fi signálu který ESP32 vytváří.

Pro uživatelské rozhraní byla použita knihovna “WebServer.h“. Tato knihovna umožňuje vytvořit jednoduchý webový server přímo na desce. Tyto 2 knihovny spolu umožňují uživateli se připojit na vysílanou Wi-Fi a do prohlížeče na telefonu či počítači zadat programem určenou IP adresu, která ho poté přenesse na Arduino web.

Na webu poté je možná registrace a případný login do systému. Díky knihovně “Preferences.h“ si deska může přímo do vnitřní paměti ukládat data. Po registraci si deska zapamatuje zadané heslo, které je na login stránce poté použito pro odemčení zámku.

Další knihovna “Servo.h“. Slouží k snadnému ovládání servomotoru. Umožňuje přesné nastavení natočení servomotoru na určitou úroveň.

Knihovna “SPI.h“ zajišťuje komunikaci mezi Arduinem a čtečkou pomocí SPI rozhraní, které je běžné u rychlého přenosu dat mezi mikrokontrolérem a periferními zařízeními. Bez ní by Arduino nemohlo s čtečkou vůbec komunikovat.

Knihovna “MFRC522.h“ obsahuje funkce potřebné pro základní ovládání RFID čtečky MFRC522 – například její inicializaci, komunikaci přes SPI rozhraní, nebo kontrolu, zda je čtečka připravená načítat čipy. Tato knihovna umožňuje rozpoznat, kdy byla přiložena RFID karta nebo čip, a přečíst její jedinečné identifikační číslo podle standardu ISO/IEC 14443 typu A, který je nejběžněji využíván u běžných bezkontaktních karet. Díky této knihovně lze snadno implementovat ověření přístupu pomocí RFID technologií na platformě Arduino.

6.2 Webové uživatelské rozhraní

```
void handleRoot() {
    storedPassword = preferences.getString("pass", "");
    registered = storedPassword != "";

    if (!registered) {
        // Registrace
        String form = String(htmlStyle) + R"rawliteral(
            <h2>Register</h2>
            <form action="/register" method="POST">
                <input type="password" name="pass1" placeholder="Enter password"><br>
                <input type="password" name="pass2" placeholder="Confirm password"><br>
                <input type="submit" value="Register">
            </form>
        )rawliteral";
        server.send(200, "text/html", form);
    } else {
        // Přihlášení
        String form = String(htmlStyle) + R"rawliteral(
            <h2>Login</h2>
            <form action="/login" method="POST">
                <input type="password" name="loginpass" placeholder="Enter password"><br>
                <input type="submit" value="Login">
            </form>
        )rawliteral";
        server.send(200, "text/html", form);
    }
}
```

Obrázek 34: Kód pro zobrazení registračního a přihlašovacího formuláře

Při připojení na Wi-Fi a následnému zadání IP adresy se zobrazí jedna z těchto dvou možných webových stránek. To, která se z nich zobrazí záleží zda je uloženo nějaké heslo. Pokud ještě uživatel není zaregistrovaný tudíž ještě neuložil žádné heslo a je to jeho první přihlášení tak se zobrazí registrační stránka na které uživatel dvakrát zadá stejné heslo které se poté zapíše do systému. Jakmile je uloženo heslo tak se už bude zobrazovat přihlašovací stránka kde stačí heslo zadat jen jednou. Celé toto uživatelské rozhraní bylo vytvořeno v angličtině, protože je to nejrozšířenější jazyk a zajišťuje větší srozumitelnost hlavně z důvodu že Arduino web nepodporuje diakritiku.

```

const char* ssid = "Zamek";
const char* password = "12345678";

WebServer server(80);
Preferences preferences;

String storedPassword = "";
bool registered = false;

HardwareSerial SerialNano(1); // UART1: RX=16, TX=17 connected to Nano

```

Obrázek 35: Inicializace základních proměnných a nastavení komunikace

```

void setup() {
  Serial.begin(115200);

  pinMode(LOCK_SIGNAL_PIN, OUTPUT);
  digitalWrite(LOCK_SIGNAL_PIN, LOW); // Výchozí stav: zamčeno

  WiFi.softAP(ssid, password);
  Serial.println("Access Point started");
  Serial.print("IP address: "); Serial.println(WiFi.softAPIP());

  preferences.begin("smartlock", false);
  storedPassword = preferences.getString("pass", "");
  registered = storedPassword != "";

  server.on("/", handleRoot);

  server.on("/register", HTTP_POST, []() {
    String p1 = server.arg("pass1"), p2 = server.arg("pass2");
    if (p1 != p2) {
      server.send(200, "text/html", String(htmlStyle) + "<h3>Passwords do not match.</h3><a href=\"/\" class=\"button\">Try Again</a>");
    } else {
      preferences.putString("pass", p1);
      server.sendHeader("Location", "/"); server.send(303);
    }
  });

  server.on("/login", HTTP_POST, []() {
    String input = server.arg("loginpass");
    if (input == preferences.getString("pass", "")) {
      server.sendHeader("Location", "/loggedin"); server.send(303);
    } else {
      server.send(200, "text/html", String(htmlStyle) + "<h3>Wrong password.</h3><a href=\"/\" class=\"button\">Try Again</a>");
    }
  });
}

```

Obrázek 36: Ukázka funkce Setup() a zbylý kód registračního a přihlašovacího formuláře

Tady dochází k inicializaci všech potřebných částí systému ihned po spuštění zařízení. První se spustí sériová komunikace mezi deskou a počítačem a také druhou deskou. Po spuštění této komunikace se vytvoří vlastní Wi-Fi Access Point. To umožní uživateli se připojit. Po spuštění se do sériového monitoru vypíše IP adresa Arduino webu. Dále pak už je programová část registrační a přihlašovací stránky kde se kontroluje zda je uloženo nějaké heslo a správnost hesla. Po zadání špatného hesla v přihlášení nebo zadání dvou různých hesel v registraci se zobrazí pouze text že je heslo špatně a po správném zadání hesla je uživatel přesměrován do hlavního menu zámku.

6.2.1 Webové ovládání zámku a uživatelské funkce

```
// Přihlášení
void handleLoggedIn() {
    String page = String(htmlStyle) + R"rawliteral(
        <h2>Welcome!</h2>
        <a href="/lock" class="button">Lock Control</a><br>
        <a href="/changePASS" class="button">Change Password</a><br>
        <a href="/deletePASS" class="button">Delete Password</a><br>
        <a href="/" class="button">Logout</a>
    )rawliteral";
    server.send(200, "text/html", page);
}

// Ovládání zámku
void handleLockPage() {
    String page = String(htmlStyle) + R"rawliteral(
        <h2>Lock Control</h2>
        <form action="/open" method="POST"><input type="submit" value="Unlock" class="button"></form>
        <form action="/close" method="POST"><input type="submit" value="Lock" class="button"></form>
        <a href="/loggedin" class="button">Back</a>
    )rawliteral";
    server.send(200, "text/html", page);
}
```

Obrázek 37: Uživatelské rozhraní po přihlášení do systému

Po přihlášení se načte takzvaná homepage, tedy hlavní stránku, ve které jsou zobrazeny všechny dostupné možnosti zámku. Uživatel zde najde čtyři hlavní tlačítka: První je pro přechod na stránku která obstarává ovládání zámku. Druhé slouží jako stránka pro změnu hesla. Třetí je pro úplné odstranění hesla tudíž odstranění účtu uživatele, po odstranění se zobrazí zpět registrační stránka a nakonec čtvrté prosté tlačítko na odhlášení které poté navrátí uživatele zpět na přihlašovací stránku.

Stránka na ovládání zámku obsahuje 2 tlačítka. Jedno pro odemčení druhé pro zamčení. Při zmáčknutí jednoho z nich se odešle buď "HIGH" signál který otevře zámek nebo "LOW" který ten zámek zamče.

```
server.on("/loggedin", HTTP_GET, handleLoggedIn);
server.on("/lock", HTTP_GET, handleLockPage);

server.on("/open", HTTP_POST, []() {
    digitalWrite(LOCK_SIGNAL_PIN, HIGH); // odemkni
    delay(10000);
    digitalWrite(LOCK_SIGNAL_PIN, LOW);
    server.setHeader("Location", "/lock"); server.send(303);
});

server.on("/close", HTTP_POST, []() {
    digitalWrite(LOCK_SIGNAL_PIN, LOW); // zamkni
    server.setHeader("Location", "/lock"); server.send(303);
});
```

Obrázek 38: Posílání signálu z jedné desky do druhé

6.2.2 Změna hesla a odstranění registrace

```
// Změna hesla
void handleChangePassPage() {
    String page = String(htmlStyle) + R"rawliteral(
        <h2>Change Password</h2>
        <form action="/changePass" method="POST">
            <input type="password" name="old" placeholder="Current password"><br>
            <input type="password" name="new1" placeholder="New password"><br>
            <input type="password" name="new2" placeholder="Repeat new password"><br>
            <input type="submit" value="Change" class="button">
        </form>
        <a href="/loggedin" class="button">Back</a>
    )rawliteral";
    server.send(200, "text/html", page);
}

// Odstranění účtu/hesla
void handleDeletePassPage() {
    String page = String(htmlStyle) + R"rawliteral(
        <h2>Delete Password</h2>
        <form action="/deletePass" method="POST">
            <input type="password" name="old" placeholder="Current password"><br>
            <input type="submit" value="Delete" class="button">
        </form>
        <a href="/loggedin" class="button">Back</a>
    )rawliteral";
    server.send(200, "text/html", page);
}
```

Obrázek 39: Zobrazení stránky pro změnu a smazání hesla uživatele

Tato část kódu zajišťuje správu hesla uloženého v zařízení – konkrétně jeho změnu a úplné odstranění. Po přihlášení má uživatel možnost přejít na stránku, kde může heslo změnit. Nejprve musí zadat stávající heslo a následně nové heslo dvakrát pro ověření. Pokud staré heslo nesouhlasí nebo se nová hesla neshodují, systém na to upozorní pomocí jednoduché chybové hlášky. V opačném případě se nové heslo uloží a uživatel je přesměrován zpět na hlavní stránku.

```
server.on("/changePass", HTTP_GET, handleChangePassPage);
server.on("/changePass", HTTP_POST, []() {
    String oldp = server.arg("old"), n1 = server.arg("new1"), n2 = server.arg("new2");
    String cur = preferences.getString("pass", "");
    if (oldp != cur) {
        server.send(200, "text/html", String(htmlStyle) + "<h3>Wrong password.</h3><a href=\"/changePass\" class=\"button\">Try Again</a>");
    } else if (n1 != n2) {
        server.send(200, "text/html", String(htmlStyle) + "<h3>Passwords do not match.</h3><a href=\"/changePass\" class=\"button\">Try Again</a>");
    } else {
        preferences.putString("pass", n1);
        server.setHeader("Location", "/loggedin"); server.send(303);
    }
});

server.on("/deletePass", HTTP_GET, handleDeletePassPage);
server.on("/deletePass", HTTP_POST, []() {
    String oldp = server.arg("old");
    if (oldp != preferences.getString("pass", "")) {
        server.send(200, "text/html", String(htmlStyle) + "<h3>Wrong password.</h3><a href=\"/deletePass\" class=\"button\">Try Again</a>");
    } else {
        preferences.clear();
        server.setHeader("Location", "/"); server.send(303);
    }
});
```

Obrázek 40: Kód na zpracování požadavků na změnu a odstranění hesla

Druhá funkce umožňuje uživateli heslo úplně odstranit. Tento proces opět vyžaduje zadání stávajícího hesla, čímž se předchází neoprávněnému přístupu. Pokud je heslo zadáno správně, zařízení smaže všechny uložené údaje a systém se vrátí do výchozího stavu, jako by byl spuštěn poprvé. Tímto způsobem lze zařízení snadno „resetovat“ a připravit ho pro nového uživatele. Celkově tento mechanismus zajišťuje vyšší úroveň bezpečnosti a pohodlnou správu přístupu.

6.2.3 Inicializace hardwaru na desce Arduino Nano

```
#include <SPI.h>
#include <MFRC522.h>
#include <Servo.h>

#define SS_PIN 10
#define RST_PIN 9
#define SIGNAL_PIN 4 // D4 jako vstup

Servo myServo;
MFRC522 mfrc522(SS_PIN, RST_PIN);

byte allowedUID[] = {0x95, 0x5C, 0xC9, 0x01};
const byte allowedUIDLength = 4;

void setup() {
  Serial.begin(9600);
  myServo.attach(6);
  myServo.write(90);

  pinMode(SIGNAL_PIN, INPUT);
  SPI.begin();
  mfrc522.PCD_Init();
  Serial.println("Přilož kartu k čtečce...");
}
```

Obrázek 41: Inicializace servomotoru, sériové komunikace a RFID čtečky

V úvodu kódu dochází k definici pinů pro připojení RFID čtečky MFRC522 a k vytvoření potřebných objektů pro komunikaci a řízení zařízení. Deska Arduino Nano využívá digitální pin 4 a ESP32 využívá digitální pin 16 pro komunikaci mezi sebou. Pin 6 je využit pro připojení servomotoru. RFID čtečka MFRC522 je připojena k Arduino přes sběrnici SPI, přičemž pin 10 slouží jako čip select (SDA) a pin 9 jako reset.

Ve funkci `setup()` je inicializována sériová komunikace, nastaven servomotor do výchozí polohy a připraven objekt pro práci s čipem MFRC522. Následná smyčka `loop()` zajišťuje detekci přiložené RFID karty standardu ISO14443 a po jejím načtení vypíše její UID do sériového monitoru. Zároveň dojde k otevření servomotoru na pět sekund, čímž se simuluje odemčení zámku. Dále je zajištěno zpracování příkazů přijímaných z ESP32, které ovládají stav servomotoru, tedy otevření či zavření zámku podle přijatého signálu HIGH nebo LOW.

6.2.4 Zpracování RFID karty a příkazů z ESP32

```
void loop() {  
  // RFID čtečka  
  if (mfrc522.PICC_IsNewCardPresent() && mfrc522.PICC_ReadCardSerial()) {  
    Serial.print("UID karty: ");  
    for (byte i = 0; i < mfrc522.uid.size; i++) {  
      Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");  
      Serial.print(mfrc522.uid.uidByte[i], HEX);  
    }  
    Serial.println();  
  
    if (isUIDMatch(mfrc522.uid.uidByte, mfrc522.uid.size)) {  
      Serial.println("Karta schválena. Otevírám zámek...");  
      myServo.write(45);  
      delay(5000);  
      myServo.write(90);  
    } else {  
      Serial.println("Nepovolená karta.");  
    }  
  
    mfrc522.PICC_HaltA();  
  }  
}
```

Obrázek 42: Kód hlavní smyčky na načtení RFID karty a ovládání zámku

Hlavní smyčka programu zajišťuje detekci RFID karet a zároveň reaguje na příkazy odeslané z desky ESP32. Pokud je přiložena kompatibilní karta, načte se její UID a na krátkou dobu (5 sekund) dojde k otevření zámku pomocí servomotoru.

```
// Signál z ESP32  
if (digitalRead(SIGNAL_PIN) == HIGH) {  
  Serial.println("Signál z ESP32: Odemknout");  
  myServo.write(45);  
}  
else if (digitalRead(SIGNAL_PIN) == LOW){  
  Serial.println("Signál z ESP32: zamknout");  
  myServo.write(90);  
}
```

Obrázek 43: Kód hlavní smyčky na přijímání signálů z ESP32

Tato část kódu kontroluje zda přišel příkaz z ESP32. Pokud ano, podle typu signálu HIGH nebo LOW se zámek buď otevře, nebo zavře. Tato logika umožňuje ovládání zámku jak fyzicky pomocí karty, tak bezdrátově přes ESP32 z webového rozhraní.

6.2.5 Rozpoznávání karet

```
byte allowedUID[] = {0x95, 0x5C, 0xC9, 0x01};  
const byte allowedUIDLength = 4;
```

Obrázek 44: Ručně zapsané UID čipu

V programu je také implementováno rozpoznávání konkrétní karty na základě jejího UID. Po inicializaci systému dojde ke čtení UID každé přiložené karty a jeho porovnání s předem definovaným povoleným UID. Pokud dojde ke shodě, servomotor na krátkou dobu otevře zámek. V opačném případě je přístup zamítnut a zámek zůstává zavřený. Tímto způsobem je zajištěna základní úroveň autorizace uživatele pomocí RFID technologie. Dále je zachována možnost ovládat zámek i prostřednictvím sériové komunikace s ESP32, které může vysílat příkazy pro otevření nebo zavření zámku.

```
bool isUIDMatch(byte *uid, byte uidLength) {  
    if (uidLength != allowedUIDLength) return false;  
  
    for (byte i = 0; i < uidLength; i++) {  
        if (uid[i] != allowedUID[i]) return false;  
    }  
    return true;  
}
```

Obrázek 45: Boolová funkce na ověření přiložených karet

7 TESTOVÁNÍ A ANALÝZA VÝSLEDKŮ

V této kapitole se zaměřím na praktické ověření funkčnosti chytrého zámku s RFID čtečkou MFRC522. Cílem testování je zjistit, jak systém reaguje v různých situacích, které mohou nastat při běžném používání. Kromě ověření správného načítání autorizované RFID karty se budu také zabývat tím, zda čtečka dokáže kartu rozpoznat přes různé materiály, jako je plast, dřevo, sklo nebo kov. Tím chci zjistit, zda je možné instalovat čtečku skrytě – například pod kryt nebo za ochrannou vrstvu.

Dále plánuji otestovat, na jakou vzdálenost je možné kartu bezpečně přechytit, a jak rychle systém reaguje – tedy jak dlouho trvá, než se po přiložení karty servo otevře. Důležitou součástí bude i ověření bezpečnosti, například tím, že zkusím přiložit jiné RFID karty, které nejsou autorizované, a sleduji, jestli systém správně odmítne otevření zámku.

Chci také nasimulovat výpadek komunikace mezi Arduinem a ESP32, abych ověřil, že systém bude fungovat správně i bez připojení k externímu zařízení.

Výsledky jednotlivých testů budu zaznamenávat, vyhodnotím je a pokusím se z nich vyvodit závěry o tom, jak spolehlivý a bezpečný je tento způsob řízení přístupu v praxi.

7.1 Testování bezpečnosti

Kvůli omezenému počtu RFID karet jsem měl k dispozici jen jednu náhradní kartu, kterou jsem použil k otestování bezpečnosti systému. I když šlo jen o jednoduchý test, čtečka správně rozpoznala, že UID této karty neodpovídá uloženému, a zámek se neodemkl. Ověřil jsem tak, že systém reaguje pouze na autorizovanou kartu, což potvrzuje jeho základní bezpečnostní funkci.

```
Přilož kartu k čtečce...
UID karty:  08 79 15 C5
Nepovolená karta.
```

Obrázek 46: Výpis z Arduina do seriál monitoru

7.2 Testování různých materiálů a vzdáleností

V této části jsem testoval, jak různé materiály a vzdálenosti ovlivňují čtení RFID karet. Některý materiály, třeba kov, můžou signál blokovat nebo zeslabit, takže je důležitý vědět, jak to funguje v praxi. Takže jsem zkoušel, jestli karta projde skrz různé věci a jak daleko od čtečky se ještě dá spolehlivě přečíst.

7.2.1 Ochranná schránka

Ochranná schránka má tloušťku 7,5 mm a je vytištěná z PETG filamentu. Materiál je dost pevný a odolný, ale zároveň bylo potřeba zjistit, jestli neblokuje signál čtečky. Při testování se ukázalo, že i přes tuhle vrstvu dokáže čtečka kartu spolehlivě rozpoznat a zámek odemknout.

Bohužel se ale při testování objevil problém v designu vrchní strany ochranné schránky. Kvůli postavení servomotoru musela horní strana být vyšší a při odemykání z této strany nebyl čip ani zaznamenán.

Ze spodní strany však testování probíhá dobře a ke čtení dochází s větší rezervou. K přečtení dochází už kolem vzdálenosti 20 mm od čtečky.

7.2.2 Sklo

I přes 5mm skleněný panel dochází k rychlému a spolehlivému rozpoznání čipu. Signál čtečky není sklem nijak výrazně oslaben, což potvrzuje, že tento materiál nebrání komunikaci mezi kartou a čtečkou. Díky tomu by bylo možné umístit čtečku i za skleněnou výplň bez výrazného omezení funkčnosti zámku. Stejně jako u PETG filamentu je tam větší rezerva která se pohybuje též okolo 20 mm od čtečky

7.2.3 Dřevotříska

Na testování dřevotřísky jsem použil svůj pracovní stůl, jehož deska má tloušťku 18 mm. Když k tomu připočtu i ochrannou schránku, celková tloušťka materiálu, přes který musí signál projít, je přes 25 mm. Přesto se ukázalo, že čtečka dokáže kartu stále spolehlivě přečíst, což je dobrá zpráva pro možnost využití zámku i v silnějších dveřích

7.2.4 Kov

Testování přes kovovou překážku bylo trochu složitější, protože kov výrazně blokuje signál RFID čtečky. Použil jsem kovový plech o tloušťce 3mm, přes který se čtečka snažila přečíst kartu. Výsledek byl, že signál byl hodně oslabený a čtení karty nefungovalo spolehlivě nebo vůbec. To je důležité vědět, pokud by se zámek měl používat na dveřích s kovovým povrchem nebo za kovovými překážkami. V takovém případě bude třeba najít jiné řešení nebo čtečku umístit mimo celkovou konstrukci ochranné schránky.

7.2.5 Zhodnocení výsledků

Po otestování různých materiálů se ukázalo, že čtečka spolehlivě funguje skrz PETG, sklo i dřevotřísku, a to i na slušnou vzdálenost. To je dobrá zpráva, protože to znamená, že zámek lze dobře schovat nebo zabezpečit bez toho, aby se výrazně omezila jeho funkčnost. Na druhou stranu kov je pro čtečku problém – signál je rušený a karta se často nepřečte. Proto je potřeba při umístění zámku s kovem počítat a případně najít lepší řešení. Celkově ale můžu říct, že systém funguje spolehlivě a je připravený na praktické použití, když se správně zváží okolní podmínky.

Materiál	Tloušťka materiálu	Největší možná vzdálenost od čtečky
PETG/Plast	7.5 mm	Cca 24 mm
Sklo	5 mm	Cca 24 mm
Dřevotříska	18 mm	25.5 mm
Kov	3 mm	-

Tabulka 13: Testování různých materiálů

7.3 Testování rychlosti webové stránky v závislosti na signálu

Při testování funkčnosti webového rozhraní na platformě ESP32 byla sledována závislost jeho provozu na síle Wi-Fi signálu. Měření bylo prováděno v domácím prostředí, ve kterém se mezi zařízením a uživatelem nacházely různé překážky, například zdi či kusy nábytku. Tyto fyzické bariéry mohly výsledky měření částečně ovlivnit, a proto je možné, že ve volném prostoru by dosah signálu byl o něco větší.

První mírná degradace Wi-Fi signálu byla zaznamenána již ve vzdálenosti přibližně 8 metrů od ESP32, které fungovalo jako přístupový bod. I přes tuto mírnou oslabenou úroveň signálu však nebyly pozorovány žádné problémy s funkčností webového rozhraní. Stránky se načítaly rychle a plynule a všechny uživatelské požadavky byly zpracovávány bez obtíží.

Ve vzdálenosti přibližně 12 metrů, kdy byla úroveň Wi-Fi signálu indikována pouze dvěma čárkami, byla funkčnost rozhraní stále plně zachována. Webové stránky se načítaly bez zpomalení a systém reagoval na uživatelské vstupy bez výpadků, což svědčí o stabilním a spolehlivém připojení i při nižší intenzitě signálu.

Za hranicí přibližně 15 metrů, přičemž tato vzdálenost se nacházela za již zmíněnou stěnou, začaly být zaznamenávány první problémy s komunikací. Docházelo k občasným zpomalením načítání webových stránek a k nutnosti jejich opakovaného načtení. V této vzdálenosti se navíc zařízení začalo pravidelně odpojovat od Wi-Fi sítě, což dále snižovalo stabilitu a spolehlivost spojení. Přesto bylo zařízení stále schopno komunikovat, i když již s výrazně omezenou stabilitou.

Je ale nutné poznamenat, že i přes to, že měření bylo prováděno v interiérovém prostředí, kde se mezi zařízením a klientem nacházely různé fyzické překážky, jako jsou zdi, nábytek či další objekty, které mohly negativně ovlivnit kvalitu signálu, byly výsledky z hlediska funkčnosti systému stále uspokojivé. Navzdory těmto omezením se podařilo zajistit stabilní připojení a plynulé načítání webového rozhraní v běžně používaných vzdálenostech, což svědčí o dobré odolnosti a spolehlivosti Wi-Fi připojení v rámci testovaného prostředí.

8 PLÁNY DO BUDOUČNA

Do budoucna se počítá s několika vylepšeními, které by mohly celý systém udělat praktičtější a příjemnějším pro běžné používání.

Jedním z hlavních plánů je mobilní aplikace nebo software který by měl na starosti například správy uživatelů, historie odemčení nebo vzdálený přístup přes internet. Toto řešení by bylo efektivnější než současný způsob ovládání přes webové rozhraní, a navíc by umožnilo pohodlnější ovládání odkudkoli.

S tímto softwarem by pak bylo možné přidávat další zařízení a ovládat tak celou chytrou domácnost. Díky centralizované správě by uživatel mohl mít všechny prvky pod kontrolou v jedné aplikaci, což by výrazně zjednodušilo každodenní používání a zvýšilo komfort i bezpečnost domácnosti. S tím softwarem by také byla možnost rozšířit funkčnost systému o nastavení časového intervalu, během kterého bude možné odemknout zámek například pomocí RFID čipu. Tato funkce by zvýšila bezpečnost zařízení, například omezením přístupu pouze na pracovní dobu nebo jiné definované časové období.

Mezi možné budoucí rozšíření systému patří také zavedení dalších způsobů ovládání zámku, například prostřednictvím otisku prstu, zadáním PIN kódu v mobilní aplikaci nebo použitím dvoufázového ověření, které by mohlo výrazně zvýšit celkovou úroveň zabezpečení. Zajímavou funkcionalitou by byl také časově omezený přístup – například možnost vygenerovat jednorázový přístupový kód s platností na omezenou dobu (např. několik hodin). Takové řešení by mohlo být užitečné pro doručovatele, servisní techniky či krátkodobé návštěvy, které nepotřebují trvalý přístup.

Kvůli praktičnosti bylo uvažováno o oddělení RFID čtečky od samotného zámku. Během testování bylo zjištěno, že při použití kovových dveří nemusí být RFID čipy správně načítány, protože kov ruší signál mezi čipem a čtečkou. Z tohoto důvodu bylo doporučeno umístit čtečku na stěnu vedle dveří, kde by nebyl signál nijak rušen. Samotný zámek a řídicí elektronika by přitom zůstaly bezpečně uvnitř, což zajišťuje dostatečné zabezpečení. Toto řešení je technicky jednodušší a spolehlivější v reálném prostředí, kde podmínky pro bezdrátové čtení nemusí být ideální.

Ještě by bylo vhodné přidat do systému senzor, který by hlídal, jestli jsou dveře opravdu zavřené. Teď totiž celý zámek ovládá uživatel a v kódu přednastavené časovače, takže se může stát, že se dveře otevrou omylem nebo zůstanou otevřené. Díky takovému senzoru by bylo možné stav dveří kontrolovat automaticky, což by zvýšilo bezpečnost a spolehlivost celého zámku.

8.1 Porovnání s komerčně dostupnými zámky

Funkce / Model	Navržený zámek	Nuki Smart Lock 3.0	Tedee GO	Yale Linus
Ovládání přes mobilní aplikaci	Webové rozhraní	Ano	Ano	Ano
Vzdálený přístup	Ne	Ano	Ano	Ano
Ovládání pomocí RFID	Ano	Ne	Ne	Ne
Ovládání přes PIN nebo otisk	Heslo	PIN	Ne	PIN
Detekce otevřených dveří	Ne (plánováno)	Ano (dveřní senzor)	Ano	Ano
Napájení	USB / Baterie	Baterie	Baterie	Baterie
Přibližná cena	cca 500 Kč	~5000 Kč	~3500 Kč	~5500 Kč

Tabulka 14: Porovnání s různými dostupnými zámky

Přestože navržený systém zatím nedosahuje úrovně komerčních produktů z hlediska komfortu a funkcí, nabízí cenově dostupné řešení se základními možnostmi zabezpečení. Výhodou je možnost vlastní úpravy systému a rozšiřitelnost, kterou komerční zařízení často neumožňují. V budoucnu by díky plánovaným rozšířením mohl systém nabídnout podobný komfort jako dražší řešení, avšak s výrazně nižšími náklady.

ZÁVĚR

Cílem této práce bylo navrhnout a implementovat funkční chytrý elektronický zámek, který bude využívat platformu Arduino a nabídne moderní přístupové metody s důrazem na jednoduchost, bezpečnost a praktickou použitelnost. Vytvořený systém využívá mikropočítač ESP32 pro bezdrátovou komunikaci a správu uživatelského rozhraní a Arduino Nano pro ovládání hardwarových komponent, zejména servomotoru a zpracování RFID čipů.

Během vývoje byla navržena a realizována kompletní softwarová architektura včetně webového rozhraní, které umožňuje uživateli zaregistrovat heslo, přihlásit se, měnit ho, nebo přístup zcela odstranit. To vše bez nutnosti připojení k internetu – zařízení vytváří vlastní Wi-Fi síť. Pro ovládání zámku byl vyvinut jednoduchý mechanismus založený na servomotoru, který fyzicky ovládá západku. Všechny komponenty byly umístěny do kompaktní schránky vytvořené metodou 3D tisku, což usnadnilo testování a úpravy.

Systém byl následně podroben testování, které ověřilo jak funkčnost softwarových částí, tak mechanickou spolehlivost. Byly provedeny i bezpečnostní testy a zkoušky s různými materiály a vzdálenostmi RFID čtečky. Výsledky ukázaly, že řešení je stabilní, uživatelsky přívětivé a vhodné pro reálné nasazení například v domácnosti, kanceláři či jako přístupový systém pro menší zařízení.

Výhodou celého návrhu je jeho modularita a otevřenost – systém je snadno rozšiřitelný o další bezpečnostní funkce, jako je biometrické ověření, mobilní aplikace, nebo napojení na další prvky chytré domácnosti.

Tato práce ukazuje, že s využitím open-source nástrojů a běžně dostupných komponent lze navrhnout a realizovat funkční bezpečnostní systém, který svými schopnostmi může konkurovat komerčním řešením a zároveň je zcela přizpůsobitelný potřebám uživatele.

SEZNAM POUŽITÉ LITERATURY

- [1] BLUM, Jeremy. Exploring Arduino: tools and techniques for engineering wizardy. Indianapolis: Wiley, 2013. ISBN 978-1-118-54936-0.
- [2] BANZI, Massimo a SHILOH, Michael. Getting Started with Arduino. 3rd ed. Sebastopol: Maker Media, 2014. ISBN 978-1-4493-6333-8.
- [3] MONK, Simon. Programming Arduino: Getting Started with Sketches. 2nd ed. New York: McGraw-Hill, 2016. ISBN 978-1-259-64214-4.
- [4] NOURI, Hussein. Arduino for Embedded Systems. Switzerland: Springer, 2018. ISBN 978-3-319-77621-6.
- [5] MARGOLIS, Michael. Arduino Cookbook. 3rd ed. Sebastopol: O'Reilly Media, 2020. ISBN 978-1-4919-5518-4.
- [6] PATTERSON, Zachary a KNIGHT, Charles. Learning Arduino: A Beginner's Guide to Microcontroller Projects. United Kingdom: Packt Publishing, 2020. ISBN 978-1-80056-647-8.
- [7] PHGAME. *Začínáme s Arduinem*. PHGame [online]. 2013 [cit. 23. února 2025]. Dostupné z: https://phgame.cz/PHGame_serialy/serialy/zaciname-s-arduinem/
- [8] HW Kitchen. Arduino pod pokličkou: Jak funguje mikrokontroler. HW Kitchen [online]. 2021 [cit. 2025-02-23]. Dostupné z: <https://bastlirna.hwkitchen.cz/arduino-pod-poklickou-jak-funguje-mikrokontroler/>
- [9] Petus. Arduino, čili mikrokontroléry pro každého. Chiptron [online]. 17. ledna 2021 [cit. 2025-2-23]. Dostupné z: https://chiptron.cz/articles.php?article_id=275
- [10] Bez autora. Intel společně podpoří vzdělání a inovace v České republice. Česká škola [online]. Říjen 2013 [cit. 2025-2-23]. Dostupné z: <http://www.ceskaskola.cz/2013/10/intel-spolecne-podpori-vzdelani-a.html>
- [11] Arduino. "About Arduino." *Arduino* [online]. [cit. 2025-2-23]. Dostupné z: <https://www.arduino.cc/en/about>.
- [12] Conrad Electronic. "Arduino® » Praktický mikrokontrolér pro individuální spínací a řídicí úlohy." Conrad Electronic [online]. [cit. 2025-02-24]. Dostupné z: <https://www.conrad.cz/cs/clanky/elektromechanika/arduino.html>.

- [13] LaskaKit.cz. "LaskaKit ESP32-DEVKit." LaskaKit [online]. [cit. 2025-02-28]. Dostupné z: <https://www.laskakit.cz/laskakit-esp32-devkit/>
- [14] IVANKA, Ján. Mechanické zábranné systémy. Praha: Profesional Publishing, 2013. ISBN 978-80-7431-118-1.
- [15] Encyclopædia Britannica. Lock (security). Britannica [online]. [cit. 2025-02-28]. Dostupné z: <https://www.britannica.com/technology/lock-security>
- [16] ALSYKO. Mechanické zábranné systémy. ALSYKO [online]. [cit. 2025-02-28]. Dostupné z: <https://alsyko.cz/sluzby/mechanicke-zabranne-systemy/>
- [17] Prosimvstupte.cz. Jak fungují elektrické zámky. Prosimvstupte.cz [online]. [cit. 2025-02-28]. Dostupné z: <https://prosimvstupte.cz/jak-funguji-elektricke-zamky/>
- [18] Setro.sk. Elektronické zámky – poznáte ich výhody a použitie? Setro.sk [online]. [cit. 2025-03-10]. Dostupné z: <https://setro.sk/elektronicke-zamky-poznate-ich-vyhody-a-pouzitie/>
- [19] Baar Group. Elektronický zámek na kód TRON-CODE – popis a funkce. Baar Group [online]. [cit. 2025-03-12]. Dostupné z: <https://baargroup.cz/produkt/elektronicky-zamek-na-kod-tron-code/>
- [20] Anviz.com. Biometrická kontrola přístupu pomocí otisků prstů. Anviz.com [online]. [cit. 2025-04-01]. Dostupné z: <https://www.anviz.com/cs/blog/fingerprint-access-control.html>
- [21] Botland.cz. CAN-Bus Shield pro Arduino – SparkFun DEV-13262. Botland.cz [online]. [cit. 2025-04-01]. Dostupné z: <https://botland.cz/arduino-shield-komunikace/2452-can-bus-shield-pro-arduino-sparkfun-dev-13262-5904422369606.html>
- [22] ITnetwork.cz. Arduino 5. díl – Klávesnice. ITnetwork.cz [online]. [cit. 2025-04-01]. Dostupné z: <https://www.itnetwork.cz/hardware-pc/arduino/hardware/arduino-5-dil-klavesnice>
- [23] Drátek.cz. Arduino a klávesnice 4x4 (16 tlačítek). Drátek.cz [online]. [cit. 2025-04-01]. Dostupné z: <https://navody.dratek.cz/navody-k-produktum/arduino-klavesnice-4x4-16-tlacitek.html>

- [24] Circuit Digest. How to use RC522 RFID Reader With Arduino. Circuit Digest [online]. [cit. 2025-04-03]. Dostupné z: <https://circuitdigest.com/microcontroller-projects/interfacing-rfid-reader-module-with-arduino>
- [25] Components101. RC522 RFID Module Pinout. Components101 [online]. [cit. 2025-04-03]. Dostupné z: <https://components101.com/wireless/rc522-rfid-module>
- [26] SolidWorks Help. SolidWorks Documentation. Dassault Systèmes [online]. [cit. 2025-04-11]. Dostupné z: <https://help.solidworks.com>
- [27] ANDROID DEVELOPERS. Android Studio. Google, [online]. [cit. 2025-05-02]. Dostupné z: <https://developer.android.com/studio>
- [28] ECKEL, Bruce. Thinking in Java. 4th ed. Upper Saddle River: Prentice Hall, 2006. ISBN 0-13-187248-6.
- [29] DOSHI, Venkat Subramaniam. Kotlin Programming: The Big Nerd Ranch Guide. 1st ed. Pearson Education, 2018. ISBN 978-0135161630.
- [30] JETBRAINS. Getting Started with Kotlin. Kotlinlang.org, [online]. [cit. 2025-05-02]. Dostupné z: <https://kotlinlang.org/docs/getting-started.html>
- [31] ESPRESSIF SYSTEMS. ESP32-WROOM-32 Datasheet [online]. Version 3.9, 2023 [cit. 2025-05-08]. Dostupné z: https://www.laskakit.cz/user/related_files/esp32-wroom-32_datasheet_en.pdf
- [32] MG90S Micro Servo. [online]. [cit. 2025-05-08]. Dostupné z: <https://www.alldatasheet.com/datasheet-pdf/download/1132104/ETC2/MG90S.html>
- [33] NXP Semiconductors. PN5180 High-performance full NFC forum-compliant frontend datasheet [PDF]. [online]. [cit. 2025-05-08]. Dostupné z: https://www.laskakit.cz/user/related_files/pn5180-780120.pdf
- [34] LaskaKit.cz. RFID čtečka s vestavěnou anténou NFC/RF PN5180 ISO15693 – čtení i zápis [online]. [cit. 2025-05-08]. Dostupné z: <https://www.laskakit.cz/rfid-ctecka-s-vestavenou-antenou-nfc-rf-pn5180-iso15693-cteni-i-zapis>
- [35] LaskaKit.cz. Baterie Li-Pol 3.7V 500mAh (LiPo). Laskakit.cz [online]. [cit. 2025-05-08]. Dostupné z: <https://www.laskakit.cz/baterie-li-po-3-7v-500mah-lipo/>

- [36] Drátek.cz. IIC / I2C 5V na 3.3V obousměrný převodník logické úrovně [online]. [cit. 2025-05-08]. Dostupné z: <https://dratek.cz/arduino/1481-iic-i2c-5v-na-3.3v-obousmerny-prevodnik-logicke-urovne.html>
- [37] Drátek.cz. Nepájivé kontaktní pole ZY-170 W [online]. [cit. 2025-05-08]. Dostupné z: <https://dratek.cz/arduino/1424-nepajive-kontakti-pole-zy-170-w.html>
- [38] LaskaKit.cz. Wemos D1 mini shield s tlačítkem [online]. Laskakit.cz, [cit. 2025-05-09]. Dostupné z: <https://www.laskakit.cz/wemos-d1-mini-shield-s-tlacitkem/>
- [39] ECH Hardware. *Stainless Steel Mortise Lock* [online]. [cit. 2025-05-18]. Dostupné z: <https://www.echhardware.com/product-item/4585-stainless-steel-mortise-lock/>

SEZNAM OBRÁZKŮ

Obrázek 1: Arduino Uno [8]	13
Obrázek 2: Části Arduino Desky [12].....	15
Obrázek 3: Deska Arduino Uno R3 [11]	16
Obrázek 4: Deska Arduino Micro [11]	18
Obrázek 5: Arduino Mega 2560 rev3[11].....	19
Obrázek 6: Arduino Nano ESP32. [11]	20
Obrázek 7: Příklad bezpečnostních dveří. [16]	21
Obrázek 8: Příklad elektronického zámku na RFID čip. [17].....	22
Obrázek 9: Elektronický zámek s kódem.	23
Obrázek 10: Elektronický zámek na RFID čip.	24
Obrázek 11: Sejf s biometrickým zámek (otisk prstu).....	25
Obrázek 12: Arduino Ethernet Shield 2. [1]	26
Obrázek 13: 4x4 Matricová klávesnice k Arduino. [22].....	27
Obrázek 14: RFID modul PN5180 [34].....	28
Obrázek 15: Solidworks User Interface	30
Obrázek 16: Android Studio User Interface [27].....	31
Obrázek 17: Oficiální logo Java.....	32
Obrázek 18: Kotlin logo.....	33
Obrázek 19: Arduino IDE User Interface	34
Obrázek 20: LaskaKit ESP32 DevKit [13]	37
Obrázek 21: Vývojová deska Arduino Nano [11]	38
Obrázek 22: Servo motor MG90S [32].....	39
Obrázek 23: RFID čtečka RC522 [24].....	40
Obrázek 24: Baterie GeB LiPol 503035 [35].....	41
Obrázek 25: Wemos D1 mini shield s tlačítkem [38].....	42
Obrázek 26: Blueprint mechanického zámku [39]	43
Obrázek 27: 3D model ozubeného kola.....	44
Obrázek 28: Model výsuvné západky	44
Obrázek 29: Spodní část ochranné schránky. Horní pohled	45
Obrázek 30: Spodní část ochranné schránky. Pohled z úhlu	46
Obrázek 31: Spodní část ochranné schránky. Pohled z úhlu	47
Obrázek 32: Schéma zapojení.....	48
Obrázek 33: Kompletní sestava elektronického zámku	49
Obrázek 34: Kód pro zobrazení registračního a přihlašovacího formuláře	51

Obrázek 35: Inicializace základních proměnných a nastavení komunikace.....	52
Obrázek 36: Ukázka funkce Setup() a zbylý kód registračního a přihlašovacího formuláře	52
Obrázek 37: Uživatelské rozhraní po přihlášení do systému.....	53
Obrázek 38: Posílání signálu z jedné desky do druhé	53
Obrázek 39: Zobrazení stránky pro změnu a smazání hesla uživatele	54
Obrázek 40: Kód na zpracování požadavků na změnu a odstranění hesla	55
Obrázek 41: Inicializace servomotoru, sériové komunikace a RFID čtečky.....	56
Obrázek 42: Kód hlavní smyčky na načtení RFID karty a ovládání zámku.....	57
Obrázek 43: Kód hlavní smyčky na přijímání signálů z ESP32.....	57
Obrázek 44: Ručně zapsané UID čipu	58
Obrázek 45: Boolová funkce na ověření přiložených karet.....	58
Obrázek 46: Výpis z Arduina do seriál monitoru	59

SEZNAM TABULEK

Tabulka 1: Některé z nejznámějších Arduino desek a jejich parametry[12]	14
Tabulka 2: Parametry desky Arduino Uno R3 [11]	17
Tabulka 3: Parametry Desky Arduino Micro [11]	18
Tabulka 4: Parametry desky Arduino Mega rev3. [11]	19
Tabulka 5: Parametry desky Arduino Nano ESP32. [11]	20
Tabulka 6: Adresace pinů mezi klávesnicí 4x4 a Arduinem. [22].....	27
Tabulka 7: Adresace pinů mezi RFID modulem RC522 a Arduinem. [25]	29
Tabulka 8: Parametry desky LaskaKit ESP32 DevKit [13,31].....	37
Tabulka 9: Tabulka propojení ESP32 ↔ Arduino Nano (sériová komunikace)	37
Tabulka 10: Parametry Vývojové Desky Arduino Nano [11]	38
Tabulka 11: Tabulka zapojení servomotoru MG90S	39
Tabulka 12: Tabulka zapojení RFID čtečky MFRC522	40
Tabulka 13: Testování různých materiálů.....	61
Tabulka 14: Porovnání s různými dostupnými zámky.....	64

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

LED	Light-Emitting Diode
PWM	Pulse-Width Modulation
ICSP	In Circuit Serial Programming
IoT	Internet of Things
USB	Universal Serial Bus
SCL	Serial Clock
SDA	Serial Data
RX	Receive
TX	Transmit
SRAM	Static Random Access Memory
EEPROM	Electrically Erasable Programmable Read-Only Memory
ROM	Read-Only Memory
Wi-Fi	Wireless Fidelity
I ² C	Inter-Integrated Circuit
RFID	Radio Frequency Identification
AES	Advanced Encryption Standard
NFC	Near Field Communication
ČSN EN	Česká státní norma Evropská norma
SPI	Serial Peripheral Interface
UART	Universal asynchronous receiver-transmitter
UID	Unique Identifier
IDE	Integrated Development Enviroment
Li-Po	Lithium-Polymer
BLE	Bluetooth Low Energy
BR/EDR	Basic Rate / Enhanced Data Rate

DAC	Digital-to-Analog Converter
ADC	Analog-to-Digital Converter
GPIO	General Purpose Input/Output
JST	Japan Solderless Terminal
AP	Access Point
IP	Internet Protocol

SEZNAM PŘÍLOH

Příloha P I: Název první přílohy

PŘÍLOHA P I: NÁZEV PRVNÍ PŘÍLOHY

Obsah první přílohy.