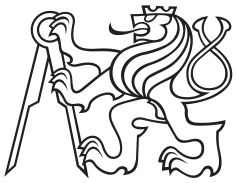
Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra radioelektroniky

Embedded zařízení pro ukládání hesel

Embedded Password Storage Device

Jan Sedlák

Vedoucí práce: Ing. Ondřej Nentvich, Ph.D.
Studijní program: Elektronika a komunikace
Květen 2024

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Sedlák** Jméno: **Jan** Osobní číslo: **507277**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra radioelektroniky**

Studijní program: **Elektronika a komunikace**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Embedded zařízení pro ukládání hesel

Název bakalářské práce anglicky:

Embedded Password Storage Device

Pokyny pro vypracování:

Proveďte rešerši možných variant zařízení pro ukládání dat, zejména se poté orientujte na ukládání přístupových údajů a jejich šifrování. Následně navrhněte a realizujte elektronické zařízení pro správu hesel a klíčů s využitím mikrokontroléru. Pro autorizaci přístupu ke správě záznamů využijte zadání uživatelského přístupového hesla nebo otisk prstu. Zadávání a předávání hesel s mikrokontrolérem realizujte s pomocí grafické aplikace.

Seznam doporučené literatury:

[1] NOVIELLO, Carmine. Mastering STM32. 2nd. Leanpub, 2022.
[2] KLEIDERMACHER, David a Mike KLEIDERMACHER. Embedded Systems Security. Elsevier, 2012. ISBN 978-0-12-386886-2.
[3] MAREŠ, Martin a Tomáš VALLA. Průvodce labyrintem algoritmů. Praha: CZ.NIC, 2017. ISBN 978-80-88168-19-5.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Ondřej Nentvich, Ph.D. katedra radioelektroniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **24.01.2024** Termín odevzdání bakalářské práce: **24.05.2024**

Platnost zadání bakalářské práce: **21.09.2025**

Ing. Ondřej Nentvich, Ph.D.	doc. Ing. Stanislav Vítek, Ph.D.	prof. Mgr. Petr Páta, Ph.D.
podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

. Datum převzetí zadání

Podpis studenta

Poděkování

Tímto bych rád poděkoval panu Ing. Ondřeji Nentvichovi, Ph.D. za jeho ochotu, nápomoc a předání mnoha zkušeností při vedení této práce. Dále chci poděkovat pražské laboratoři STMicroelectronics za poskytnutí vývojového kitu. A v neposlední řadě také mé rodině za její podporu v průběhu mého studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 21. května 2024

Abstrakt

Tato bakalářská práce se zabývá problematikou bezpečné práce s hesly a jejich ukládání pomocí správců hesel. Poskytuje přehled zabezpečení hesel od šifrování, hardwarového zabezpečení až po nejrůznější způsoby autentizace uživatele. Cílem práce je provést rešerši na ukládání přístupových údajů a jejich šifrování, následně navrhnout a realizovat elektronické zařízení pro správu hesel s využitím mikrokontroléru. Pro realizaci zařízení byl vybrán vývojový kit s LCD displejem a kapacitní dotykovou vrstvou, umožňující snažší ovladání grafické aplikace. Pro zabezpečení celého systému byl použit, mimo hlavního hesla, také senzor otisku prstu.

Klíčová slova: embedded systém, heslo, otisk prstu, šifrování, mikrokontrolér, správce hesel

Abstract

This bachelor thesis deals with the issue of secure password handling and storage using password managers. It provides an overview of password security from encryption, hardware security to various methods of user authentication. The aim of the paper is to conduct a research on access data storage and encryption, then design and implement an electronic password management device using a microcontroller. For the implementation of the device, based on the predefined requirements, a development kit with LCD touch screen was selected to facilitate the operation of the graphical application. For the security of the whole system, apart from the master password, a optical fingerprint sensor was also used.

Keywords: embedded system, password, fingerprint, encryption, microcontroller, password manager

Obsah

Obrázky

Tabulky

Úvod

Digitální bezpečnost se v současné době stala klíčovým tématem digitálního prostředí, kde je ochrana osobních údajů a citlivých informací stále stěžejnější. Mezi základní prvky ochrany digitální identity patří správa hesel, která v posledních letech nabývá na významu v důsledku růstu počtu online služeb a aplikací, které vyžadují přihlášení. Tato bakalářská práce se věnuje problematice bezpečné práce s hesly a jejich bezpečnému ukládání pomocí správců hesel, s poskytnutím přehledu moderních technik zabezpečení hesel.

Práce zkoumá způsoby zabezpečení hesel zahrnující nejen šifrování a hardwarové zabezpečení, ale také různé způsoby autentizace uživatele. Cílem této práce je provést rešerši v oblasti ukládání přístupových údajů a jejich šifrování a následně navrhnout a realizovat elektronické zařízení pro správu hesel s přívětivým grafickým rozhraním.

Tato práce přináší nejen teoretický přehled moderních metod zabezpečení hesel, ale také praktickou implementaci v podobě správce hesel. Jejím cílem je přispět k lepší ochraně digitální identity uživatelů prostřednictvím efektivních a uživatelsky přívětivých nástrojů pro správu hesel.

Kapitola 1

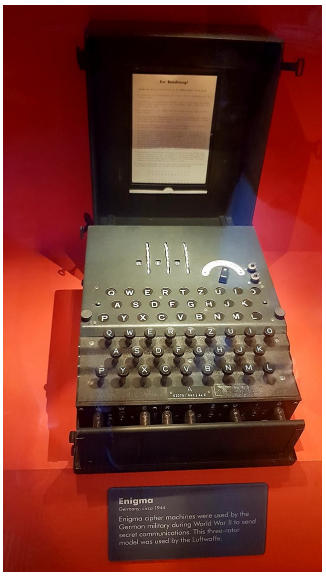
Teoretická část

Hesla jsou součástí našich životů od nepaměti. Za celou dobu jejich historie existovala v nejrůznějších podobách. Počátky používání hesel sahají až do říše římské, kde římští vojáci používali tzv. *watchwords*. Slova, která byla známá pouze užší skupině a sloužila pro autentizaci vojáka. Tato metoda dokázala spolehlivě odhalit nepřítele. Podobným způsobem byla ve 20. letech minulého století používána slova při koupi zakázaného zboží, například alkoholu či drog. Dnes bychom tento způsob mohli přirovnat ke slangu [1]. Dlouhý vývoj má za sebou také širování. Za zmínku stojí Caesarova šifra, jedná se o jednoduchý šifrovací algoritmus založený na posunu znaků, písmen. Jednotlivé znaky zprávy jsou posunuty o přesný krok. Poud je zpráva zašifrována s krokem 1, všechna písmena *a* se stanou *b*. Tuto šifru používal již Julius Caesar při posílání tajných zpráv pro své spojence.

S první digitalizací se společnost setkala v 60. letech 20. století na prestižním MIT. Při vývoji (CTSS)[1], později známý jako Multics, přišel doktor Ferdinand Corbató s jednoduchým a přímočarým řešením, jak ochránit soubory před použitím jinými uživateli, tedy pomocí hesla - unikátního sledu znaků, který zná pouze autorizovaný uživatel [2].

V dnešní době je kladen veliký důraz na kybernetickou bezpečnost pro přístup k nejrůznějším souborům nebo službám. Z tohoto důvodu je důležité vytvářet bezpečná hesla a v ideálním případě mít pro každý přístup jedno unikátní. Studie provedená v roce 2023 společností NordPass uvádí, že se průměrný počet hesel na uživatele výrazně zvýšil, a to o 25 %. Za jednu z příčin nárůstu je v několika posledních letech považována i globální pandemie COVID-19. V tuto dobu došlo k technologickému rozmachu. Hesla jsou vyžadována na webových stránkách, sociálních sítích a dalších službách. Průměr se tak vyšplhal na 100 hesel na uživatele [3]. Pro porovnání, dle studií v roce 2008, po nástupu webu 2.0, činil průměr 21 hesel [4]. Řešením problému s velkým počtem přilašovacích údajů jsou správci hesel.

[1] Compatible Time-Sharing System - umožňuje přístup několika uživatelů najednou



(a) : Enigma [5]

(b) : M-209 [6]

Obrázek 1.1: Nejznámější šifrovací stroje druhé světové války

1.1 První šifrování

První pokusy zbavit se čistého textu nalezneme v době 2. světové války. Jedním z šifrovacích strojů používaných americkou armádou byl přístroj s označením M-209 uvedený na obrázku 1.1b. Skládal se ze šesti rotorů s 26, 25, 23, 21, 19 a 17 pozicemi, délka periody klíče tak činí 101405850. Pro zajímavost, dnešní zařízení jsou schopna tuto periodu vyřešit za setinu vteřiny [7]. Na německé straně se využíval šifrovací stroj Enigma, který je pro ilustraci na obrázku 1.1a. Stejně jako M-290 fungoval na bázi rotorů. Její rozšifrování hrálo podstatnou roli ve vítězství spojenců.

Stroj Enigma byl jednou z prvních mechanizovaných metod šifrování textu pomocí iterační šifry. Používal řadu rotorů, které pomocí elektřiny, žárovky a reflektoru umožňovaly obsluze zprávu buď zašifrovat, nebo dešifrovat. Původní poloha rotorů, nastavená při každém šifrování a založená na předem připraveném vzoru, který zase vycházel z kalendáře, umožňovala používat stroj i v případě, že byl kompromitován. Když byla Enigma v provozu, při každém dalším stisknutí klávesy se rotory měnily v zarovnání oproti nastaveným pozicím tak, že pokaždé vzniklo jiné písmeno. Se zprávou v ruce zadával operátor jednotlivé znaky do stroje stisknutím klávesy podobné psacímu stroji. Rotory se vyrovnaly a písmeno se rozsvítilo, čímž obsluha zjistila, o jaké písmeno se skutečně jedná. Podobně by při šifrování stiskl operátor klávesu a rozsvícené písmeno by bylo šifrovaným textem. Neustále se měnící vnitřní tok elektřiny, který způsoboval změnu rotorů, nebyl náhodný, ale vytvářel polyalfabetickou šifru, která mohla být při každém použití jiná [8].

1.2 Šifrování

Jak bylo zmíněno v úvodu teoretické kapitoly 1, za zrod digitálních hesel považujeme 60. léta 20. století. Vše nastartoval Fernando Corbató, když se podílel na vývoji zabezpečení dat uživatelů používajících sdílený systém Compatible Time-Sharing System (CTSS).

Ukládání dat jako čistého textu se ukázalo býti nebezpečné, proto lidé přišli s hešováním hesel. Hešovací funkce jsou jednosměrné funkce, které transformují vstupní řetězec dat libovolné délky na výsledek o pevné délce, ten nazýváme **heš**.

Pro označení šifrovacího algoritmu za bezpečný je nutné, aby splňoval tato kritéria [9]:

- Pro útočníka musí být nemožné vygenerovat zprávu se stejným specifickým hešem.
- Pro útočníka musí být nemožné vytvořit dvě zprávy, které by vyprodukovaly stejný heš.

1.2.1 Symetrické a asymetrické šifrování

Symetrické šifrování
Symetrické šifrování, nazývané také jako šifrování tajným klíčem, používá pouze jeden klíč, *sdílený klíč*. Ten se používá jak pro šifrování, tak i pro dešifrování zprávy. Z tohoto důvodu je kladen vysoký důraz na předání klíče, to musí proběhnout zcela bezpečně [10].

Symetrické šifrování má vysokou rychlost a používá se v mnoha moderních počítačových systémech k ochraně dat. Díky jeho rychlosi je tato metoda považována za velice bezpečnou. Symetrické šifrování je používáno například u šifrovacího standardu AES, který nahradil předchůdce DES.

Asymetrické šifrování
Asymetrické šifrování využívá, na rozdíl od symetrického, privátní a veřejný klíč. Tyto dva klíče spolu matematicky souvisí a jsou vygenerovány pro vzájemné použití. Privátní klíč musí zůstat pouze v držení vlastníka. Veřejný klíč se sdílí pro ostatní uživatele.

Díky použití dvou šifrovacích klíčů je matematicky nerealizovatelné pro kohokoli použít veřejný klíč k získání klíče privátního.

Primární výhodou asymetrického šifrování je absence předání tajného klíče. Veřejný klíč může být k dispozici celému světu, protože samotný je nepoužitelný [10].

Proces šifrování s použitím asymetrického šifrování

- Odesílatel zprávy použije veřejný, volně dostupný, klíč pro zašifrování zprávy.
- Příjemce rozšifruje zprávu pomocí svého vlastního privátního klíče. Pouze privátní klíč, související s veřejným klíčem, který zprávu zašifroval, je schopen ji dešifrovat.

Délky klíčů
Další rozdíl mezi symetrickým a asymetrickým šifrováním je spjat s délkou klíčů. Ta je měřena v bitech a souvisí s mírou zabezpečení. V symetrických schématech jsou klíče vybírány náhodně a jejich délka je v závislosti na požadované úrovni zabezpečení obvykle nastavena na 128 nebo 256 bitů [11].

V asymetrickém šifrování ale musí mezi veřejným a soukromým klíčem existovat matematický vztah, což znamená, že spolu sdílí matematický vzorec. Vzhledem k tomu, že by tento vzorec mohli útočníci potenciálně zneužít k prolomení šifrování, musí být asymetrické klíče mnohem delší, aby poskytovaly stejnou úroveň zabezpečení. Rozdíl v délce klíče je tak výrazný, že k podobné úrovni zabezpečení, kterou najdete u 128bitovového symetrického klíče, budete potřebovat 2048bitový asymetrický klíč [11].

1.2.2 Data Encryption Standard

S postupným zdokonalováním útoků bylo zapotřebí inovovat také bezpečnostní opatření. UNIX udělal několik bezpečnostních vylepšení. Jedním z nich byla adaptace standardu Data Encryption Standard, zkráceně DES. Jedná se o šifrovací systém vydaný roku 1975 Národním úřadem pro standardizaci (NBS), dnes známým jako Národní institut standardizace a technologií (NIST).

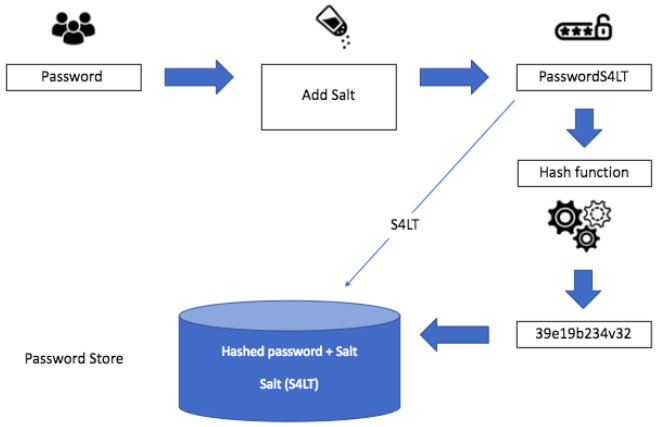
Jeho nevýhodou byla malá rychlost v softwarové implementaci. Na druhou stranu byl extrémně rychlý na komerčních čipech, které byly vytvořeny přímo pro DES. Aby se zabránilo použití takových čipů k prolomení hesla, systém UNIX implementaci DES upravil jednu z vnitřních tabulek používaných algoritmem tak, aby byla závislá na náhodném 12bitovém čísle. To by nutilo útočníky k vytvoření vlastních čipů DES, což samozřejmě mohli udělat, ale na úkor velmi vysoké ceny [12].

1.2.3 Key stretching

Jedná se o techniku vylepšující bezpečnost šifrovacích klíčů transformací původního klíče na klíč vylepšený. U běžných hesel spočívá v použití několika iterací hešovací funkce. Zjednodušeně, vytváříme heš heše, tento postup opakujeme dle počtu iterací.

1.2.4 Salted Passwords

Salted hesla jsou založena na metodě key stretching. K heslu je přidána takzvaná sůl. Jde o krátký sled náhodných znaků, který je přidán ještě před šifrováním nebo hešováním. Sůl v podobě čistého textu je uložena spolu s hešem a použita při autentizaci pro ověření hesla. Na obrázku 1.2 můžete vidět blokový diagram znázorňující tuto šifrovací metodu [12].



Obrázek 1.2: Diagram Salted passwords [13]

1.2.5 Message Digest 5

Message Digest 5, známý taky pod zkratkou MD5, je široce používaná jednosměrná 128bitová hešovací funkce, která byla vytvořena profesorem Ronaldem Rivestem v roce 1992 na americkém MIT. Tento hešovací algoritmus byl původně navržen pro autentizaci digitálních podpisů.

Operace v MD5 se provádějí tak, že se nejprve zpráva či text doplní tak, že se k jejich délce přidá 64bitová hodnota a inicializuje se čtyřslovný MD buffer (A,B,C,D), každý po 128 bitech. Poté MD5 zpracovává zprávu v blocích po 16 slovech (512 bitů) pomocí 4 kol 16bitových operací na blok zprávy a vyrovnávací paměti a výstup přidá ke vstupu vyrovnávací paměti, čímž vytvoří novou hodnotu vyrovnávací paměti. Výstupní hodnota, heš, se stane konečnou hodnotou vyrovnávací paměti o délce 128 bitů [9].

Dle požadavků z úvodu kapitoly 1.2, MD5 již není považován za bezpečný šifrovací algoritmus.

1.2.6 Advanced Encryption Standard

Advanced Encryption Standard (AES) je v současné době nejrozšířenější symetrickou šifrou. V některých průmyslových standardech je její použití povinné a v mnoha dalších, komerčních i nekomerčních, je používána coby nejbezpečnější způsob zabezpečení dat. NIST započal vývoj této šifry v roce

1997, kdy bylo oznámeno vyvinutí nástupce pro tehdy nejpoužívanější standard DES, který se stával zranitelnějším vůči útokům hrubou silou. Proti AES se doposud neobjevil žádný způsob, mimo útoku hrubou silou, jak tuto šifru prolomit. Šifra byla vyvinuta kryptografy Vincentem Rijmenem a Joanem Daemenem.

Advanced Encryption Standard byl vytvořen pro vládu USA s dobrovolným a bezplatným použitím ve veřejných nebo soukromých, komerčních nebo nekomerčních programech, které poskytují šifrovací služby. Nevládní organizace, které se rozhodnou AES používat, však podléhají omezením vytvořeným limitacím ze strany USA.

Jak AES funguje?
Šifra obsahuje tři šifry různé délky. Klíče jsou o délce 128, 192 a 256 bitů. Dle délky jsou jednotlivé šifry pojmenovány AES-128, AES-192 a AES-256 [14].

Jelikož se jedná o symetrickou šifru, k šifrování i dešifrování se využívá jeden tajný klíč. Tento klíč musí znát jak odesílatel, tak příjemce zprávy, viz kapitola 1.2.1.

Vláda klasifikuje informace do tří kategorií: **důvěrné**, **tajné** a **přísně tajné**. Informace s důvěrným a tajným označením mohou být šifrovány pomocí všech délek klíče. Informace přísně tajné vyžadují použití pouze 192bitového nebo 256bitového klíče.

Algoritmus AES definuje několik transformací, které musí být provedeny. Prvním krokem je převod vstupních dat do pole, na kterém jsou transformace aplikovány a několikrát opakovány dle počtu iterací. Jejich počet je pro každou délku jiný, 10 pro 128bitovu délku, 12 pro 192bitovou délku a 14 pro 256bitovou délku. První transformací je záměna dat pomocí substituční tabulky. Druhá transformace posune datové řádky. Třetí promíchá sloupce. Poslední transformace se provádí na každém sloupci pomocí jiné části šifrovacího klíče. Delší klíče potřebují více iterací pro dokončení.

Rozdíl mezi AES a DES
Oba tyto standardy používají symetrické šifrování, ale AES je více matematicky efektivní. Hlavní benefit AES spočívá v délce šifrovacího klíče a možnosti její volby. Délka klíče je primárním faktorem ovlivňujícím dobu nutnou pro prolomení komunikace. Proto je AES exponenciálně bezpečnější než DES, který využívá klíč o délce 56 bitů. Rychlost AES je ideální pro použití v aplikacích, firmwarech a hardwaru, který vyžaduje nízkou latenci nebo vysokou propustnost [14].

1.3 Správce hesel

Vzhledem k aktuální situaci jsou správci hesel velice žádaným společníkem v každodenním životě. Jedná se o technologický nástroj pomáhající uživateli vytvářet, ukládat a spravovat přístupové údaje k online účtům [15]. Přístup je zabezpečen *hlavním heslem*. Jedná se o jediné heslo, které je nutné si zapamatovat.

Veškeré uživatelské informace jsou uloženy do zašifrované databáze. Pro šifrování je typicky použit Advanced Encryption Standard 256 (AES 256). Šifrování používá symetrický klíč, kterým lze přistupovat k datům. Další informace ke standardu jsou popsány v kapitole 1.2.6. V následující tabulce jsou uvedeny nejčastěji používaní správci hesel různých typů.

Tabulka 1.1: Nejpoužívanější správci hesel

Typ	Název
Lokální	KeePassXC
Lokální	Password Safe
Lokální	Enpass
Cloudový	Dashlane
Cloudový	1Password
Cloudový	Keeper
Prohlížeč	Google Chrome
Prohlížeč	Mozilla Firefox
Prohlížeč	Microsoft Edge

1.3.1 Používání správců hesel

Použitím několika stejných či slabých hesel se vystavujeme riziku jejich prolomení a možnému zneužití nebo prolomení u dalších služeb. Užitím správců hesel toto riziko výrazně snižujeme.
Evropské agentura sítí a informací (ENISA) doporučuje pro správnou práci s hesly [16]:

- Kombinaci několika znaků v hesle (malá a velká písmena, speciální znaky).
- Používání dlouhých hesel. Do délky 9 znaků lze heslo prolomit v řádu sekund. Při použití více než 14 znaků již nehrozí jeho prolomení pomocí aktuálně dostupné výpočetní techniky.
- Využití unikátního hesla pro každou webovou službu.
- Používat náhodně vygenerovaná hesla s využitím co největší znakové sady.

- Vyvarovat se však nepraktických náhodně vygenerovaných frází.

Různé webové služby vyžadují odlišné požadavky pro tvorbu hesla. Správci hesel byli vytvořeni za účelem usnadnění práce s hesly. Hlavní funkcí je ukládání uživatelských informací, není to však zdaleka vše, co tyto programy nabízí.

1.3.2 Funkcionality

Mimo ukládání přihlašovacích údajů do šifrované databáze poskytují správci hesel také další funkce. Ty z nich dělají účinný nástroj. Mezi nejběžněji poskytované funkce patří [17]:

Automatické doplnění. Správce je schopen dle aktuální URL adresy doplnit nejen přihlašovací údaje, ale také ostatní informace, například adresu trvalého bydliště, telefonní číslo či platební údaje.

Synchronizace. Možnost přihlášení k uživatelskému účtu z různých zařízení a platforem. Tuto funkcionalitu využívají primárně cloudové služby a správci v prohlížečích.

Ochrana. Uživatel dostane upozornění, pokud bylo heslo nalezeno na seznamu uniklých hesel. Pokud uživatel nepoužívá správce hesel a chce si zkontrolovat, zda nebylo prolomeno heslo do jeho e-mailu, lze využít tuto webovou stránku: `https://haveibeenpwned.com/`. Při vytváření nových záznamů je hlídána síla a četnost hesla v uživatelské databázi.

Generování. Silným nástrojem je možnost vygenerovat libovolně silné heslo dle aktuálního požadavku služby [17].

Záloha. Správci často umožňují zálohovat uživatelská data. V případě cloudových manažerů hesel je poskytována záloha přímo od poskytovatele.

1.3.3 Rizika

V online světě není možné být 100 % v bezpečí. V této souvislosti je třeba zmínit, že používání správců hesel s sebou nese rizika, se kterými by měl být každý uživatel seznámen.

Vše na jediném místě. Jak je již uvedeno v kapitole 1.3, data jsou ukládána do jediné zašifrované databáze. Při prolomení hlavního hesla získá útočník přístup ke všem údajům.

Záloha nemusí být vždy k dispozici. Dojde-li k poškození serveru, jediným způsobem zisku ztracených dat je záloha přímo od poskytovatele. Takové riziko nelze snížit ani vlastní zálohou na nechráněném harddisku či špatně zabezpečenou cloudovou službou [18].

Ne všechna zařízení jsou dostatečně zabezpečena. Málo zabezpečená zařízení jsou hlavním cílem hackerů. Pokud je zařízení infikováno škodným malwarem, hrozí odhalení hlavního hesla a získání plného přístupu do správce hesel kyberzločinci. Pro ochranu se doporučuje používat antivirové programy [18].

Nepoužití biometrického ověření. Biometrika je způsob ověření s vysokou úrovní zabezpečení. Umožňuje-li zařízení a správce hesel použití této varianty, je vhodné ji využít [18].

Zapomenutí hlavního hesla. Toto riziko pramení z uživatelského chování, kdy si uživatel zvolí špatně zapamatovatelné hlavní heslo [18].

Nepoužití dvoufázového ověření. Nepoužitím dvou a vícefázového ověření umožňuje uživatel útočníkovi výrazně snazší krádež přístupu. Volbou kvalitního poskytovatele správce hesel dokážeme výše zmíněná rizika minimalizovat. Ne všechna však pramení ze správce samotného. Dalším důležitým faktorem je chování a odpovědnost uživatele.

Rizika se mění dle typu použitého správce. Na trhu je nalezneme v různých podobách. **Lokální**, program přímo v počítači uživatele, **cloudové**, služba dostupná jako internetová aplikace, a **správce v prohlížečích**.

1.3.4 Lokální správci hesel

Lokální správce hesel je offline desktopová aplikace, zpravidla s grafickým uživatelským rozhraním (GUI). Ukládání dat probíhá přímo na lokální disk zařízení, chytrého telefonu či počítače. Data jsou obvykle šifrována a uložena do databáze, pro přístup je vyžádováno hlavní heslo, které může být kombinováno s dalším způsobem autentifikace jako je biometrika nebo dvoufázové ověření.

Přenos dat mezi zařízeními je možný pomocí USB disku, HID zařízení nebo další možností, pokud správce tuto metodu povoluje. Přenos tímto způsobem může narušit integritu přenášených dat a hrozí vystavení databáze bezpečnostnímu riziku. Pro lokální ukládání zašifrovaných dat není třeba použití internetu, čímž se snižuje riziko spojené s přenosem dat po síti a zneužití dat po cestě, například útokem *Man in the middle*. Pro export celé nebo části databáze je nejčastěji použit formát CSV. Díky tomuto formátu je možné exportovaná data otevřít i v jiných databázových či tabulkových aplikacích, např. Microsoft Excel, případně v obyčejném textovém editoru. Přenos dat tímto způsobem představuje bezpečnostní riziko zneužití, zejména, když je databáze dlouhodobě uložena v čitelné, nezabezpečené podobě.

Lokální správce poskytuje uživateli plnou kontrolu nad uloženými daty a jejich zálohami [19]. Je vhodné využít funkci automatického zálohování, které umožňuje provést zálohu jedenkrát za uživatelem zvolený časový úsek a na konkrétní bezpečné místo.

Tato volba správce s sebou přináší i negativa. Jedním z nich je limitovaný přístup. Program samotný je vázaný na zařízení, na kterém je nainstalován. Není možné se k datům dostat bez přítomnosti autorizované osoby u daného zařízení. Rizikem je i ztráta dat. Toto riziko zahrnuje i možnost poruchy zařízení či jeho krádež [19].

1.3.5 Cloudové služby

Cloudové služby jsou online verzí desktopových správců hesel. Uživatel má k dispozici webové uživatelské rozhraní či mobilní aplikaci. Jejich hlavní (ne)výhodou oproti lokálním programům je možnost přihlášení do databáze z libovolného zařízení, které umožňuje připojení k internetu. I cloudové služby, stejně jako ostatní typy správců, využívají k autentifikaci hlavní heslo. Zásadním rozdílem je místo uschování zašifrovaných dat, která jsou uložena na vzdáleném serveru poskytovatele [20]. Při každém přístupu jsou data přenášena pomocí webového prohlížeče internetem zabezpečenou metodou, zpravidla v zašifrované formě.

Jednou z výhod, kterou cloudové služby disponují, je automatická synchronizace. Data uložená na serveru jsou vždy aktualizována a připravena k použití na libovolném zařízení. Dalším benefitem jsou frekventované zálohy dat. Zálohy zašifrovaných uživatelských dat probíhají zcela automaticky bez nutnosti zásahu uživatele. Při poškození systému je možné ztracená data obnovit. Díky této funkci se výrazně snižuje riziko jejich ztráty [19].

Při výběru toho typu správce je vhodné brát v potaz i nevýhody, které s sebou cloudové řešení přináší, jako je například závislost na internetu. Internetový přístup je hlavním faktorem pro užívání. Pokud dojde k jeho výpadku na straně uživatele nebo poskytovatele, k datům se nelze dostat. I zařízení poskytovatele není 100% zabezpečené. Na druhou stranu jsou hesla často ukládána ještě lokálně, takže riziko ztráty se snižuje díky decentralizaci dat. Důležité je vybírat spolehlivé služby s vynikajícími výsledky v oblasti bezpečnosti. Vzhledem ke skutečnosti, že ve většině případů jsou servery ve vlastnictví třetích stran je o to zásadnější výběr důvěryhodného poskytovatele [18].

1.3.6 Správci hesel v prohlížečích

Většina prohlížečů jako Google Chrome, Microsoft Edge, Mozilla Fireox, Safari či Opera používá správce hesel zabudovaného přímo v prohlížeči. Způsob ukládání hesel jednotlivých prohlížečů je závislý na použitém operačním systému, na kterém je program spuštěn. Oproti ostatním typům správců hesel mají jen omezené funkce [21]. Oproti citovanému zdroji již v současné době nabízí hlídání síly hesla nebo varování před podezřelým přihlašováním a synchronizaci přístupových údajů mezi zařízeními. Na druhou stranu stále nenabízí komplexní generátory náhodných hesel.

Na operačním systému Microsoft Windows prohlížeče Microsoft Edge a Chrome používají jako výchozí šifrování Windows Data Protection API (DPAPI)[2]. Pro šifrování je použito uživatelské heslo do účtu Windows. Dešifrování je umožněno pouze, pokud je daný uživatel přihlášen do operačního systému [21].

Na OSX operačních systémech se Safari a Chrome spoléhají na službu OSfX Keychain[3]. Podobně jako u DPAPI je i zde výchozí heslo nastaveno jako uživatelské přihlašovací heslo do systému [21].

Prohlížeč Mozilla Firefox má správce hesel zabudovaný přímo v prohlížeči. Narozdíl od zbylých zmíněných prohlížečů si uživatel může zvolit vlastní hlavní heslo. Pro šifrování lokálních dat využívá prohlížeč šifrovací algoritmus 3DES. Pokud uživatel nezvolí vlastní hlavní heslo, data jsou šifrována pomocí klíče, který lze nalézt v lokálním souboru *key4.db*. Jako funkci pro odvození klíče používá prohlížeč hešovací funkci SHA-1, která již není považována za bezpečnou, a to z důvodu dnešní výpočetní síly grafických karet [22].

1.4 Nabídka na trhu

Před realizací je vhodné poukázat na aktuální nabídku na trhu a porovnat ji s vlastním řešením. Na trhu nalezneme spousty softwarových správců hesel, někteří byli zmíněni v tabulce 1.1. Pokud se uživatel rozhodne pro softwarové řešení, dostane se mu široké a kvalitní nabídky. Relevantnější pro tuto práci jsou nabízená hardwarová zařízení. Zde je nabídka značně omezená.

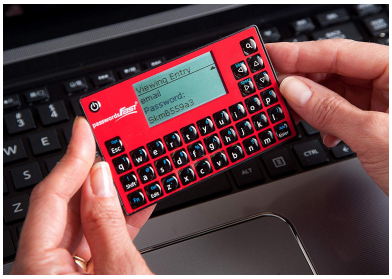
1.4.1 Tokeny

Velmi rozšířené jsou hardwarové tokeny, jejichž fungování je popsáno v podkapitole 1.6.3. Jejich využití nalezneme jako jeden z možných způsobů autentizace. Token se pomocí USB vsune do počítače a provede ověření. Za speciální varianty můžeme považovat hardwarové peněženky k uschovávání kryptoměn. Tato varianta byla jednou z motivací k vytvoření této bakalářské práce. Kromě klasických tokenů existují i varianty, které mají implementovánu paměť pro vícero přihlášení. Paměť je velice omezená, pro 10 až 30 unikátních hesel, můžeme ji tak považovat v dnešní době, kdy má každý průměrně 100 hesel, za nedostačující. Uživatel by potřeboval několik takových zařízení pro uspokojení svých potřeb. Mezi běžně dostupné výrobce patří například OnlyKey, GoTrust nebo u variant kryptopeněženek společnost Ledger či český výrobce Trezor.

[2] http://msdn.microsoft.com/en-us/library/ms995355.aspx
[3] http://pdox.ca/osxkeychain

1.4.2 Hardwarový správce hesel

Na trhu nalezneme i správce jako samostatné embedded zařízení. Tato zařízení mají, narozdíl od tokenů, větší displej a klávesnici pro komunikaci se zařízením. Disponují vlastním firmwarem, který je zpravidla jednoduchý a uživatelsky nenáročný. Největším záporem je běžně nekvalitní úroveň zabezpečení. Jedná se o nevhodně použité řešení pro ukládání dat nebo nezabezpečenou komunikaci mezi periferiemi, která je tak jednoduše čitelná po nabourání do systému. Na následujících obrázcích 1.3 až 1.5 jsou příklady běžně používaných hardwarových správců hesel.



Obrázek 1.3: PasswordFast [23]

PasswordFast využívá pro šifrování dat AES-256. Vstup do databáze, která umožní uschovat až 125 hesel, je zabezpečen jedním hlavním heslem. Délka hesla může být až 32 znaků.

Obrázek 1.4: PasswordSafe [23]

PasswordSafe umožňuje uložit až 400 přístupových údajů, které jsou opět zabezpečené pouze jediným hlavním heslem.

Obrázek 1.5: Mooltipass [25]

Mooltipass je oproti předešlým řešením daleko komplexnější. Uložená hesla jsou šifrována standardem AES-256 a přístup k nim je zabezpečen pomocí hlavního hesla. Bluetooth modul umožňuje připojení k libovolnému zařízení a následné doplňování údajů do přihlašovacích formulářů. Pro správu databáze je k dispozici, na rozdíl od předchozích, uživatelská aplikace.

1.5 Problematika zabezpečení embedded zařízení

Nedílnou součástí dnešního technologického ekosystému jsou také embedded zařízení. I ta potřebují zabezpečit. Velká část zabezpečení embedded zařízení se zaměřuje na jejich software. Zde se využívají firewally, které filtrují síťový provoz a blokují cizí uživatele. V rámci softwaru nalezneme také implementované způsoby analyzované v předešlých podkapitolách. V této je věnována pozornost možnostem zabezpečení po hardwarové stránce [28].

1.5.1 Izolace

Mezi základní pilíře hardwarového zabezpečení patří hardwarová izolace, to znamená, že vše, co je třeba zabezpečit, je nutné seskupit do jedné části hardwaru a limitovat přístup pouze pro ostatní nutné komponenty. Tímto zamezíme přístupu zbývající, potenciálně nezabezpečené části hardwaru. Ta k izolovaným částem nemůže přistupovat jinak než prostřednictvím přesně definovaných velmi úzkých, rozhraní, která vynucují bezpečnostní omezení. Mimo hardwarové izolace existují také izolační nástroje [26].

Architektura procesoru
Vývojáři by pro své aplikace měli používat mikrokontroléry, které využívají hardwarově zaměřenou izolaci. Ta rozdělí běh aplikace na dvě různá prostředí. Zabezpečená oblast spravuje periferie, paměť a funkce, které jsou pro zabezpečení zařízení kritické. Zatímco nezabezpečená oblast je běžným, komplexním prostředím pro spouštění aplikací. V současných mikrokontrolérech se této izolace dosahuje dvěma způsoby, buď použitím vícejádrových procesorů, jako je PSoC 64 společnosti Cypress, nebo použitím technologie izolace jednoho jádra, jako je TrustZone společnosti Arm.

Vícejádrové mikrokontroléry dosahují hardwarové izolace tím, že jedno jádro mikrokontroléru je určeno pro zabezpečené a druhé pro nezabezpečené prostředí. Naproti tomu TrustZone disponuje pouze jedním jádrem, které přepíná mezi zabezpečeným a nezabezpečeným prostředím.

Ochrana paměťové jednotky
Ochrana paměťových jednotek (MPUs) je druhou bezpečnostní vrstvou v zabezpečených aplikacích bez ohledu na zvolenou primární vrstvu. Staly se běžným nástrojem užívaným v mikrokontrolérech z důvodu zvýšení robustnosti aplikace.

Tyto jednotky lze využít k vytvoření přístupových hranic pro jednotlivé úlohy programu. Hranice pak přiřazují úloze pouze paměť potřebnou v rámci jejího procesu. Pokud se úlohy z nějakého důvodu pokusí přistupovat k nepřidělené paměti, MPU detekuje porušení a vyvolá výjimku.

SMPU'S a PPU'S
Posledním nástrojem pro hardwarovou izolaci jsou ochranné jednotky sdílené paměti SMPUs a ochranné jednotky periferií PPUs. Jedná se o přídavnou vrstvu, kterou může vývojář využít pro sdílení paměťového obsahu. Jednotka pro ochranu sdílené paměti, SMPUs, dokáže rozlišovat mezi různými kontexty ochrany v rámci procesoru, včetně rozdílů mezi zabezpečenými a nezabezpečenými přístupy. Ty se využívají pro vývoj různých úrovní zabezpečení a přístupů v rámci aplikace. Periferní jednotka ochrany, PPUs, spravuje přístup k různým periferním blokům, jako jsou komunikační rozhraní UART, SPI nebo USB [28].

1.5.2 Hardwarový bezpečnostní modul

Bezpečné embedded zařízení nesmí postrádat bezpečnostní modul (HSM). Jedná se o hardwarové zařízení odolné proti neoprávněné manipulaci, které zabezpečuje kryptografické procesy generováním, ochranou a správou klíčů používaných pro šifrování a dešifrování dat. Moduly musí být testovány, validovány a certifikovány standardy **FIPS 140-3** [27] a **Common Criteria**[4]. FIPS 140-3 je nejaktuálnější verzí standardů pro validaci kryptografických modulů vydanou americkou vládou. Jedná se o nástupce staršího FIPS 140-2.

Proč používat bezpečnostní moduly?
Kryptografické operace jako šifrování a digitální podepisování jsou bezcenné, pokud nejsou dobře chráněny soukromé klíče, které používají. Útočníci jsou dnes mnohem sofistikovanější, pokud jde o jejich schopnost najít uložené nebo používané soukromé klíče. HSM chrání transakce, identity a aplikace. Zároveň excelují v kryptografickém zabezpečení klíčů, šifrování, dešifrování, ověřování a digitálního podepisování pro širokou škálu aplikací.

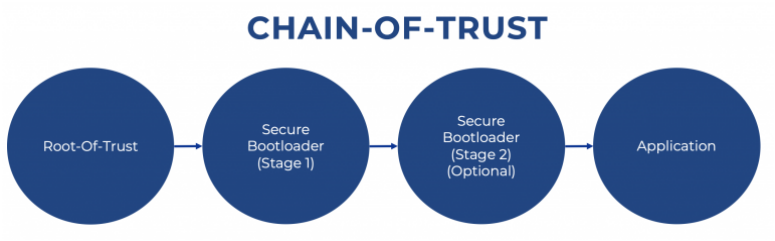
Bezpečnostní modul jako služba
Moduly HSM se vyskytují i jako služby založené na předplatném, kdy zákazník může bezpečnostní modul užívat v cloudu k vytváření, přístupu a ochraně svých citlivých dat. Služba obvykle poskytne stejnou úroveň ochrany jako lokální implementace a zároveň umožní větší flexibilitu.

1.5.3 Root Of Trust

Při spuštění embedded systému a následnému spuštění bootovacího procesu chceme, abychom si byli jisti, že naše zařízení naběhne s legitimním softwarem. Problém, se kterým se potýká mnoho systémů, spočívá v určení, zda první kód, který se v zařízení spustí, je skutečně náš autentický kód. Systém

[4] https://www.commoncriteriaportal.org/index.cfm

může být úspěšně spuštěn, ale když naběhne jako první škodlivý malware, vystavujeme zbytek aplikace nedůvěře. A právě zde Root Of Trust (ROT) zajistí, že bootujeme do správného softwaru. Celý proces je znázorněn na obrázku 1.6.



Obrázek 1.6: Chain of Trust [24]

Root Of Trust
Root Of Trust je základem, na kterém závisí všechny bezpečné operace systému. Osahuje klíče používané pro kryptografické funkce a umožňuje bezpečné spouštění systému. Bezpečnost je zajištěna již v samotném návrhu ROT. Nejbezpečnější implementací je ta hardwarová, která je imunní vůči útokům malwaru. V hardwarové formě může mít podobu samostatného modulu nebo je implementován jako bezpečnostní modul přímo v procesoru. V takovém případě jsou kryptografické klíče předvypáleny do paměti a jsou použity pro ověření digitálního podpisu ROT. Aby byla zajištěna důvěryhodnost bootovacího systému, musí být každý obsažený kód podepsán pomocí klíčů od důvěryhodných zdrojů, většinou jde o samotné výrobce modulů nebo čipů. Certifikáty důvěryhodnosti jsou spravovány organizací certifikačních autorit, která zároveň vede záznamy o vydavateli klíče, podpisovém algoritmu a datu ověření. Díky tomu je možné doložit pravost klíče [29].

Secure bootloader/ Measured bootloader
Měřený boot funguje jako pojistka proti kompromitovanému zařízení. Při bootování dochází k vyhodnocení každého kroku bootovacího procesu, zda neobsahuje malware. Pokud je krok vyhodnocený jako bezpečný, uloží ho. V opačném případě, kdy narazí na něco podezřelého, vrací se k předešlému bezpečnému kroku, dokud nenajde bezpečnou metodu dalšího postupu a neobejde malware. V krajních případech, kdy je zařízení fatálně poškozeno, se bootovací proces zacyklí, díky tomu nedojde k implementování škodlivého softwaru a data zůstanou v bezpečí [30].

Aplikace
Po úspěšném absolvování bezpečnostních kontrol ROT a secure bootloaderů může systém spustit finální aplikaci bez rizika porušení integrity citlivých dat.

Proč použít Root of trust?
Zaručuje integritu bootování správného firmwaru, dokáže odhalit přítomnost

malwaru a zabránit jeho spuštění a díky tomu zvyšuje bezpečnost systému. Dokáže také ochránit před činnostmi, jako je klonování zařízení nebo nahrávání neautorizovaného firmwaru.

1.6 Způsoby autentizace uživatele

Ověření uživatele při přihlášení již dávno není pouze o zadání hesla. Ačkoli tento tajný řetězec znaků je stále základním stavebním kamenem autentizace, nalezneme dnes několik alternativ pro způsob přihlášení. Ve většině případů se přidávají jako další možný způsob autentizace. Možná je i celková náhrada hesla jiným řešením, například *otiskem prstu, sítnice či 3D snímkem obličeje.*

1.6.1 Biometrika

Biometrika je druhou nejběžnější metodou autentizace. Zakládá se na fyzičnu uživatele, kdy se měří a analyzují otisky prstů, vzory duhovky nebo vzhled obličeje. Dnes se jedná o běžně používanou metodu odemykání chytrých telefonů.

Pokud je biometrika správně implementována, zaručuje vysokou míru zabezpečení, jelikož je spojena s fyzickou přítomností osoby. Otisk prstu, rysy obličeje nebo jiné části těla zůstávají beze změny v průběhu života. Přestože jsou techonologie na vysoké úrovni, nejsou dokonalé a 100% spolehlivé a existují případy, kdy útočník získal detailní sken obličeje a dokázal do zařízení proniknout [31].

Mezi nejznámější autentifikátory toho typu řadíme například **Windows Hello** nebo **Face ID** a **Touch ID** od společnosti Apple.

1.6.2 Single sign-on

Single Sign-On (SSO) je způsob autentifikace, kdy zaměstnanec obdrží jedinou sadu autorizačních údajů, pomocí které se přihlašuje do různých aplikací nebo webových stránek. Každý zaměstnanec má účet s poskytovatelem identity (IdP), pro aplikace se jedná o důvěryhodný zdroj. Poskytovatel služby běžně neukládá zadané heslo, ale pouze otisk uživatele. Poskytovatel identity sdělí aplikaci či webové stránce prostřednictvím souboru cookie nebo tokenů, že se uživatel ověřil jeho prostřednictvím [32].

Single Sign-On snižuje počet údajů, které si musí uživatel pamatovat, a posiluje tím bezpečnost. Zároveň není nutností se pokaždé opětovně přihlašovat. Tato metoda ověřování však znamená, že pokud dojde k narušení dat IdP, mohou hekři získat přístup k více účtům pomocí jediné přihlašovací sady. Single Sign-On také vyžaduje velkou počáteční časovou investici pro IT oddělení na nastavení a připojení k různým aplikacím a webovým stránkám [32].

1.6.3 Tokeny

Token, v tomto případě bezpečnostní token, je zařízení, mezi které patří chytré telefony, USB disky nebo chytré karty. Autentizace založená na bázi tokenů umožňuje uživateli ověření totožnosti pomocí výše zmíněných fyzických zařízení. Tokeny mohou být součástí několikanásobného ověření (MFA) nebo je možné je použít jako náhradu hesla. Při ověřování na základě tokenu se uživatelé znovuověřují jednou za předem stanovený čas, aby se omezilo neustálé přihlašování [32]. Pro odcizení přihlašovacích údajů je nutná krádež fyzického zařízení.

1.6.4 Dvoufázové a vícefázové ověření

Dvoufázové nebo vícefázové ověření vyžaduje mimo běžného hesla ještě jeden nebo více doplňujících autentizačních faktorů. Tyto způsoby mohou být libovolné z výše zmíněných, nebo lze využít ověření jednorázovým kódem zaslaným přes SMS či e-mail.

Technická doporučení pro implementaci autentizace uživatele poskytuje například americký National Institut of Standards (NIST). Ve své publikaci SP 800-63B [33] vydává směrnici s technickými požadavky pro implementaci ověřovacích mechanismů u digitálních služeb používaných federálními úřady.

Kapitola 2
Hardware

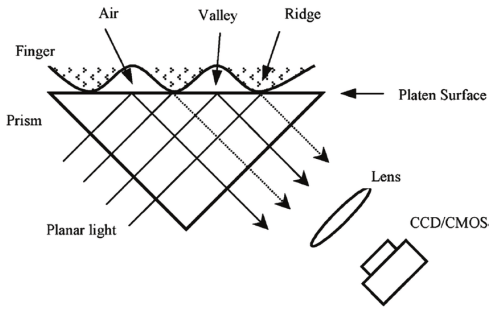
Náplň této kapitoly je věnována hardwarové části prototypu. Díky rešerši se podařilo získat povědomí o aktuální nabídce embedded správců hesel na trhu. Zároveň bylo možné využít jednotlivé produkty jako vzor pro návrh vlastního zařízení. Před samotným plánováním jsem vytyčil nosné pilíře, na kterých byl následně prototyp stavěn. Pro zachování jednoduchosti ovládání vlastního GUI je zapotřebí dostatečně velikého dotykového displeje. Aby byl zaručen plynulý chod aplikace je, důležitý výběr kvalitního a výkonného mikrokontroléru s dostatečnou pamětí. Posledním pilířem je zabezpečení systému. Pro zajištění dvoufázového ověření uživatele jsem se rozhodl pro hlavní heslo a biometriku, přesněji otisk prstu.

2.1 Senzor otisků prstů

Senzor otisků prsů je modul, který je v embedded systémech implementován jako biometrický bezpečnostní prvek. Díky dnes již vyspělé technologii se jedná o velice kvalitní a spolehlivé zařízení. Existuje několik senzorů otisků prstů, kde každý pracuje na odlišném principu. Jedná se konkrétně o princip **optický**, **kapacitní** a **ultrazvukový**.

2.1.1 Optický princip

Jedná se o nejrozšířenější senzor otisků prstů, zejména u chytrých telefonů. Využívá nejstarší metodu pro snímání a následné porovnání. Princip funguje na bázi optického snímání, viz obrázek 2.1, a pořízení dvoudimenzionální fotografie. Poté je využit algoritmus pro detekci unikátních vzorů povrchu prstu, hřebenů (ridge) a údolí (valley), který zároveň rozeznává světlé a tmavé oblasti snímku. Stejně jako kamery, i senzor má své rozlišení. Čím vyšší rozlišení, tím jemnější detaily dokáže čtečka rozeznat, a zvýšit tak bezpečnost. Oproti kamerám mají optické senzory velmi vysoký počet diod na palec, pro snazší zachycení detailů zblízka. Aby snímek po přiložení prstu nebyl tmavý, obsahují skenery také pole LED diod, které fungují jako blesk v době skenování. Vzhledem k faktu, že jsou pořizovány pouze 2D snímky, lze optickou čtečku obelhat při použití kvalitní fotografie.

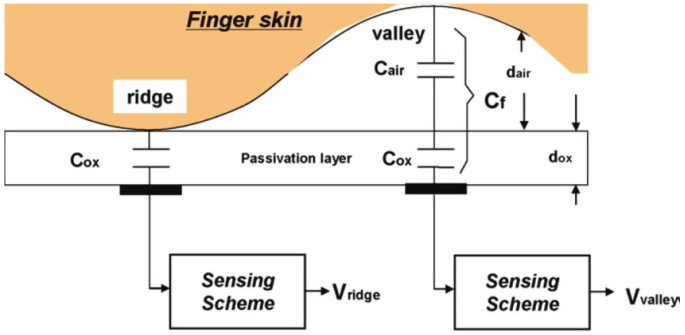


Obrázek 2.1: Princip fungování optického senzoru [34]

2.1.2 Kapacitní princip

Požadavky na zvýšení bezpečnosti vedly k použití kapicitních senzorů, které místo pořízení 2D fotografie využívají pro sběr dat pole kapacitorů. Princip kapacitního senzoru je znázorněn na obrázku 2.2. Přiložením prstu k senzoru se mění kapacita mezi prstem a deskou, respektive v místě hřebene (ridge) a údolíčka (valey). Vyčtením všech kapacitorů se poté složí v otisk prstu. Po zachycení se tato digitální data analyzují a hledají se v nich charakteristické a jedinečné atributy otisků prstů. Ty pak lze uložit pro pozdější porovnání. Díky principu založenému na změně kapacit nelze kapacitní senzor oklamat pomocí fotografie. V úvahu nepřipadá ani protéza, protože různé materiály zaznamenají mírně odlišné změny náboje na kondenzátoru. Jediné skutečné bezpečnostní riziko plyne z hacknutí hardwaru nebo softwaru.

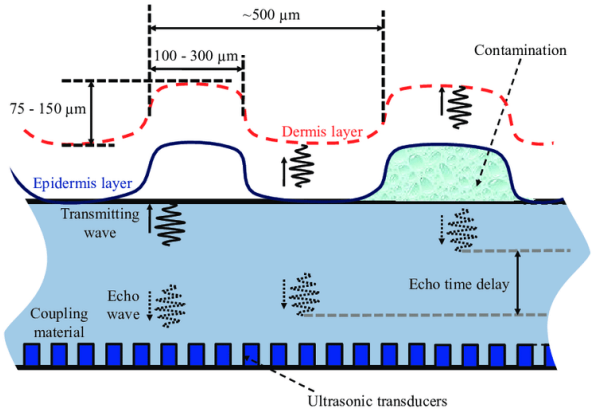
Vytvoření dostatečně velkého pole těchto kondenzátorů, obvykle stovek, ne-li tisíců, v jednom skeneru umožňuje vytvořit velmi detailní obraz hřebenů a údolí otisku prstu pouze z elektrických signálů. Stejně jako u optického skeneru je výsledkem většího počtu kondenzátorů vyšší rozlišení skeneru. To do určité míry zvyšuje úroveň zabezpečení. Přesto je výroba snímače s vysokou hustotou kapacitorů výrazně dražší.



Obrázek 2.2: Princip fungování kapacitního senzoru [34]

2.1.3 Ultrazvukový princip

Aktuálně nejnovější technologie snímání otisků prstu je pomocí ultrazvukového senzoru. Dnes se s ním můžeme setkat ve většině špičkových telefonů. Hardware ultrazvukového senzoru se skládá z ultrazvukového vysílače a přijímače. Ultrazvukový pulz je vysílán proti prstu, který je položen nad senzorem. Některé pulzy jsou absorbovány a některé se naopak odrazí zpět do snímače v závislosti na hřebenech, údolích a dalších unikátních detailech otisku prstu. Pro výpočet intenzity vracejícího se ultrazvukového impulzu na různých místech skeneru se používá senzor, který dokáže detekovat mechanické napětí. Skenování po delší dobu umožňuje získat detailnější údaje otisku. Výsledkem je podrobná 3D reprodukce naskenovaného otisku prstu. Díky 3D povaze této techniky snímání je ještě bezpečnější alternativou kapacitních skenerů.



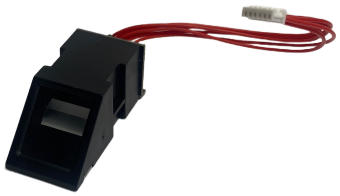
Obrázek 2.3: Princip fungování ultrazvukového senzoru [34]

2.1.4 Senzor DY50

Pro účely mého prototypu jsem z finančních a praktických důvodů použil čtečku otisků prstů DY50. Jedná se o jednoduchý biometrický optický snímač otisků prstů, s pamětí až 127 otisků prstů. Je plně kompatabilní s Arduinem, které bude třeba do projektu implementovat pro vyhodnocení. Komunikace s mikrokontrolérem probíhá pomocí sběrnice Universal asynchronous receiver-transmitter (UART). V následující tabulce 2.1 je uveden stručný výčet parametrů senzoru.

Tabulka 2.1: Tabulka základních vlastností DY50

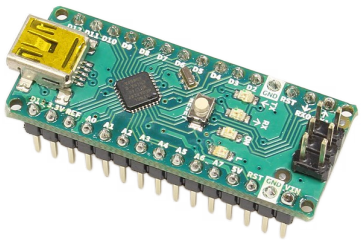
Napájení [V]	3,6 až 6
Pracovní proud [mA]	120 až 140
Teplota [°C]	-20 až 50
Režim shody	1:1



Obrázek 2.4: Použitý optický senzor otisků prstů, DY50

2.2 Arduino Nano

Arduino Nano s rozměry 45 x 18 mm je zmenšenou verzí Arduina Una. Deska je osazena mikrokontrolérem ATmega328, který je vybaven 22 digitálními vstupy/výstupy, z nichž 6 lze použít pro generování signálu pulsně-šířkové modulace (PWM) a 8 pinů lze použít jako analogové vstupy. Systém je taktován hodinovým signálem 16 MHz, má 32 kB paměti Flash a 2 kB paměti SRAM.



Obrázek 2.5: Arduino Nano [35]

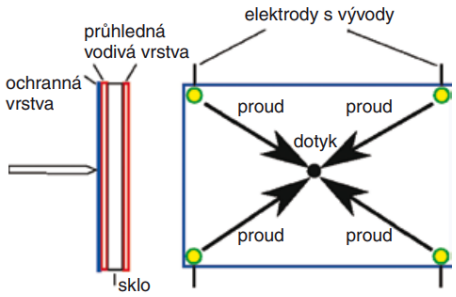
2.3 Grafický dotykový displej

Druhou důležitou komponentou je grafický dotykový displej. Jde o elektronické zařízení, které lze použít jako vstupně-výstupní periferii. Aplikace zobrazena na dospleji je uživatelem ovládána pomocí prstu nebo stylusu. Díky jednoduchosti ovládání se tak jedná o ideální náhradu klávesnice a myši. Displejů existuje celá řada. Při výběru je důležité vedle ceny a technických parametrů vybrat také, na jakém principu dotykový panel pracuje. Jedná se o princip **kapacitní**, **rezistivní**, **infračervený** a **s povrchovou akustickou vlnou (SAW)**.

2.3.1 Kapacitní dotyková vrstva

Kapacitní dotykový panel je potažen materiálem, který uchovává elektrické náboje. Při dotyku panelu displeje se do místa dotyku přivede malé množ-

ství náboje. Obvody umístěné v každém rohu panelu měří náboj a odesílají informace do mikrokontroléru ke zpracování. Uživatel musí pro interakci s kapacitním dotykovým panelem použít prst, na rozdíl od rezistivních panelů a panelů s povrchovou vlnou (SAW). Princip získání souřadnic z kapacitního panelu je vidět na obrázku 2.6. Kapacitní dotykové obrazovky nejsou ovlivňovány vnějšími vlivy a mají dobrou čitelnost.

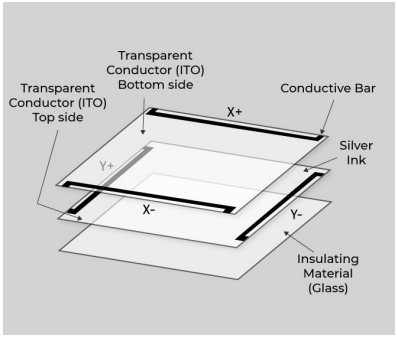


Obrázek 2.6: Princip kapacitního displeje [36]

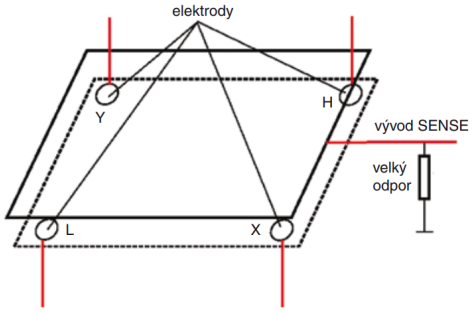
2.3.2 Rezistivní princip

Rezistivní dotykový panel se skládá z několika vrstev. Dvě vrstvy panelu jsou potaženy rezistivní vrstvou. Na tuto vrstvu je vždy na dvou protilehlých stranách nanesen proužek tvořící elektrodu. Na jedné folii vlevo a vpravo, na druhé pak nahoře a dole. Mezi vrstvami jsou malé průhledné izolační body, které zabraňují za normálních okolností styku obou vrstev. Dotyk v příslušném bodě způsobí styk těchto dvou vrstev, který je následně vyhodnocen. V tomto případě se jedná o 4-vodičový rezistivní panel.

Existují také 5-vodičové panely, ty se od předchozího typu liší uspořádáním elektrod. Na jedné vrstvě jsou čtyři elektrody označované jako Y, H, L a X. Horní vrstvu pak tvoří jediná elektroda *Sense*, která je připojena přes rezistor s vysokou hodnotou odporu na zem. Na obrázku 2.7 jsou znázorněné oba principy rezistivních panelů.



(a) : 4-vodičový dotykový panel [37]

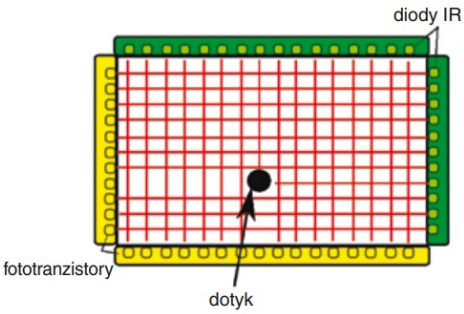


(b) : 5-vodičový dotykový panel [36]

Obrázek 2.7: Principy rezistivních dotykových panelů

2.3.3 Infračervený princip

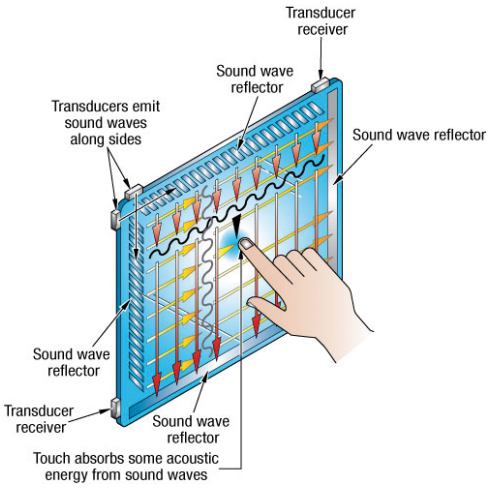
Infračervené dotykové obrazovky využívají matici infračervených paprsků, které jsou vysílány světelnými diodami (LED) s fototranzistorovým přijímacím koncem. Infračervené paprsky tvoří velkou mřížku paprsků, která pokrývá povrch překryvné vrstvy. Jakmile se povrchu dotkne prst nebo jiný neprůhledný nástroj, světelné paprsky se rozptýlí. Toto přerušení poskytuje zařízení vstupní informace o poloze místa dotyku. Princip je znázorněn na obrázku 2.8.



Obrázek 2.8: Princip infračerveného displeje [36]

2.3.4 Surface acoustic wave

Technologie povrchových akustických vln (SAW) využívá ultrazvukové vlny, které procházejí přes panel dotykové obrazovky. Při dotyku panelu je část vln absorbována a nedostane se tudíž k přijímači, jak lze vidět na obrázku 2.9. Změna ultrazvukové vlny zaznamená polohu dotyku a odešle tuto informaci do mikrokontroléru ke zpracování. Principiálně se jedná o měření doby šíření signálu. Panely dotykových obrazovek s povrchovou akustickou vlnou jsou ze všech tří typů nejpokročilejší, ale vnější vlivy je mohou snadněji poškodit.

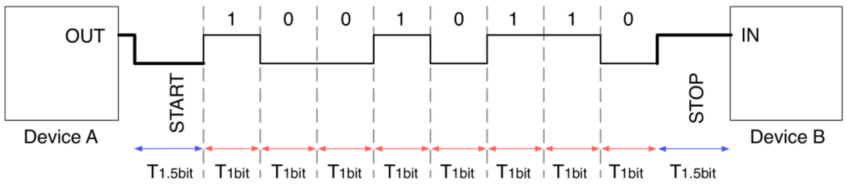


Obrázek 2.9: Princip displeje s povrchovými akustickými vlnami [39]

2.4 Potřebné periferie

2.4.1 UART

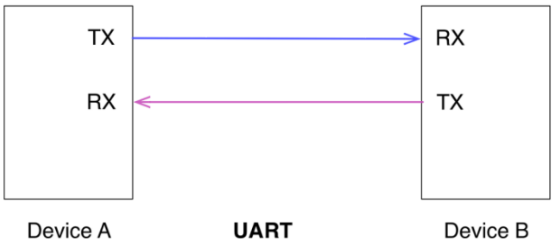
Universal asynchronous receiver-transmitter je rozhraní s komunikačním protokolem používaný v embedded systémech pro komunikaci periferií s mikrokontrolérem v režimu point-to-point nebo point-to-multipoint. Zde záleží na použitém rozhraní. Jelikož senzor DY50 je připojený přímo, budu používat režim point-to-point jako je například rozhraní RS-485. V tomto případě se jedná, jak název napovídá, o asynchronní komunikaci s konfigurovatelnou rychlostí přenosu dat. Přenos je znázorněn na obrázku 2.10.



Obrázek 2.10: Přenos dat UART [41]

Asynchronní znamená, že se neposílá hodinový signál CLK, který by synchronizoval výstup bitů, ale je třeba data synchornizovat stejnou rychlostí na obou stranách a začátek přenosu je zpravidla uveden jedním start bitem a ukončen stop bitem.

Přenos je realizován pomocí dvou signálů, **transmitter (Tx)** a **reciever (Rx)**. Jeho schéma zapojení je znázorněno na obrázku 2.11. Z toho plyne, že připojením dvou zařízení napřímo mohou obě zařízení kominukovat současně, tj. full-duplex režim. Data jsou přenášena ve formě paketů. Jeden paket se skládá ze start bitu, datového rámce, paritního bitu a stop bitu. Paket je znázorněn na obrázku 2.12.

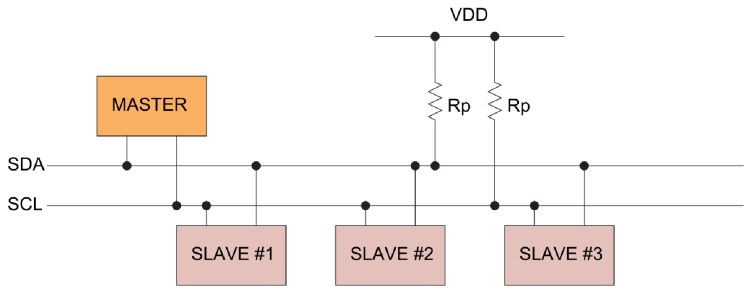


Obrázek 2.11: Schéma zapojení UART [41]

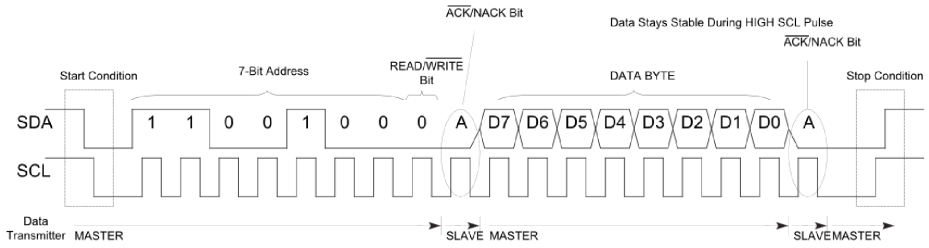
Obrázek 2.12: Ukázka paketu [40]

2.4.2 I2C

Inter-Integrated Circuit (I2C) je dvouvodičové sériové rozhraní, někdy označované jako TWI (Two Wire Interface). Obousměrná sběrnice je snadno implementovatelná a umožňuje jednoduchou komunikaci mezi jednotlivými periferiemi. Komunikce probíhá pomocí datového vodiče **SDA**, hodinového vodiče **SCL** a společnou zemí. Komunikační protokol I2C se řídí hierarchií master-slave, přičemž master je definován jako zařízení, které taktuje sběrnici, adresuje slave a zapisuje nebo čte data do a z registrů slave. Podřízené jednotky jsou zařízení, která odpovídají pouze na dotaz od masteru prostřednictvím své jedinečné adresy. Proto je nezbytné zabránit duplicitě adres mezi podřízenými zařízeními. Podřízené jednotky nikdy neiniciují přenos dat. Standardní rychlost přenosu dat je 100 kbit/s, zatímco rychlost přenosu v rychlém režimu je 400 kbit/s. Datové pakety I2C jsou uspořádány po 8 bitech, které obsahují adresu slave, číslo registru a přenášená data. Zapojení sběrnice a přenos dat po sběrnici jsou znázorněny na následujících obrázcích 2.13, respektive 2.14.



Obrázek 2.13: Zapojení pomocí sběrnice I2C [43]



Obrázek 2.14: Přenos dat po sběrnici I2C [43]

Přenos začíná splněním *startovací podmínky*, tedy změnou stavu SDA z logické jedničky na logickou nulu a současně hodinový signál SCL zůstává v

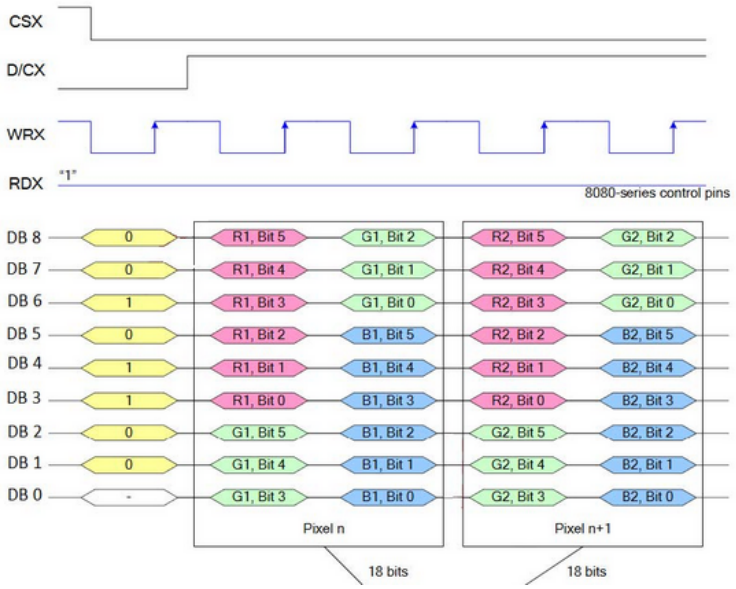
logické jedničce. Následuje přenos 8 bitů, kde prvních 7 je *adresa zařízení*, se kterým chce master komunikovat, a zbylý bit signalizuje, zda se bude číst nebo zapisovat. Pro potvrzení úspěšného přenosu každého bajtu se posílá *acknowledge* bit, který definujeme jako logickou nulu u SDA v průběhu 9. pulsu hodinového signálu SCL. Poté se posílá 8 bitů dat a následuje opět potvrzovací bit. Pokud při přenosu dojde k chybě, špatná slave adresa, slave je zaneprázdněn nebo nedokáže přijímat další data, vysílá se v rámci 9. hodinového pulsu *not acknowledge* bit, který ponechá SDA v logické jedničce. Celá komunikace je ukončena, když master vyvolá *stop podmínku*, změní stav SDA z aktivního stavu do klidového, v rámci I2C se jedná o logickou jedničku, zatímco SCL zůstává v klidu.

2.4.3 SDIO

Secure Digital Input/Output (SDIO) je rozhraní navržené jako rozšíření stávajícího standardu SD karet, které umožňuje připojení různých periferií k zařízení pomocí standardního řadiče SD.

2.4.4 Rozhraní RGB

Rozhraní RGB (Red Green Blue) je často používané pro kontrolu velkoplošného LCD displeje s vysokým rozlišením. Výhodou tohoto rozhraní je, že R,G,B data jsou přímo zapisována do displeje bez nutnosti použití Graphics Random Access Memory (GRAM). Naopak nevýhodou je, že kontrola LCD displeje je komplexnější a vyžaduje více datových vodičů. Na obrázku 2.15 je znázorněn přenos dat RGB666, barvy jsou tedy přenášeny v 6bitové formě.



Obrázek 2.15: Přenos barev RGB rozhraní [46]

2.5 STM32H7B3I-DK

Vzhledem k výše zmíněným požadavkům jsem se rozhodl použít vývojový kit STM32H7B3I-DK od společnosti STMicroelectronics, který je na obrázku 2.16. Deska disponuje dotykovým panelem, výkonným mikrokontrolérem s dostatečnou pamětí pro grafické aplikace a vysokou variabilitou rozhraní pro připojení nejrůznějších periferií, viz tabulka 2.2.

Tabulka 2.2: Přehled použitelných periferií

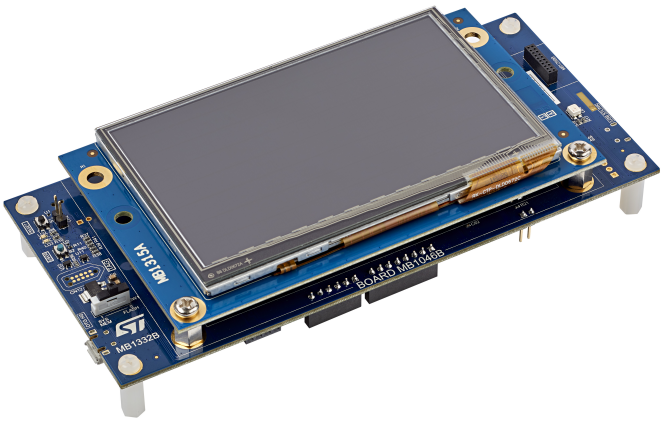
Periferie	Počet
I2C	4
SPI	6
SDMMC	2
USART	5
UART	5
USB HS/FS OTG	1
16bitový AD převodník	2
uživatelské LED	2

2.5.1 Mikrokontrolér

Vývojový kit je osazen mikrokontrolérem STM32H7B3LIH6Q. Zařízení s řadou STM32H7B3xx jsou založena na vysoce výkonném jádru Arm Cortex-M7 s 32bit Reduced Instruction Set Computer (RISC) architekturou operující až do frekvence 280 MHz. Všechny důležité parametry jsou vypsány v tabulce 2.3.

Tabulka 2.3: Technické parametry vývojového kitu

Mikroprocesor	ARM Cortex-M7
Taktovací frekvence	až 280 MHz
L1 cache	16 KB datové a 16 KB instrukční cache
Paměť flash	2 MB
RAM	1,4 MB
SRAM	1,18 MB uživatelské, 4 KB záložní

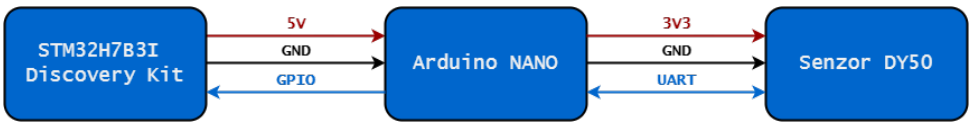


Obrázek 2.16: STM32H7B3I Discovery kit

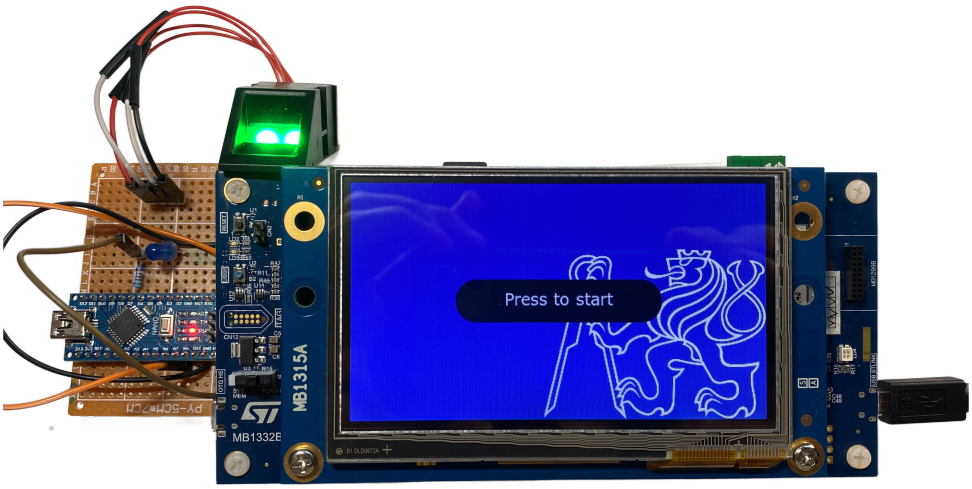
2.6 Finální realizace

Finální prototyp správce hesel se skládá ze tří bloků. Těmi jsou vývojová deska STM32H7B3I s vlastním firmwarem, senzor otisků prstů a Arduino Nano. Výsledná realizace je zobrazena na obrázku 2.18. Senzor je propojen s Arduinem pomocí sběrnice UART, přes kterou probíhá komunikace se senzorem otisků prstů. Arduino se senzorem jsou napájené z vývojové desky, která disponuje 5V výstupem pro napájení periferií. Zdrojem napájení celého zařízení může být počítač nebo powerbanka.

Arduino v tomto zapojení má za úkol vyhodnotit načtený otisk prstu a při shodě odeslat signál do vývojového kitu, který jej následně zpracuje a vpustí uživatele do aplikace. Úspěšná identifikace je doprovázena rozsvícením modré LED na PCB s Arduinem. V případě nenalezení shody v databázi otisků je tento stav indikován rozsvícením červené LED. Blokové schéma zapojení je znázorněno na obrázku 2.17.



Obrázek 2.17: Blokové schéma zapojení prototypu



Obrázek 2.18: Finální podoba prototypu

Kapitola 3
Software

Tato kapitola je věnována implementaci fimrwaru prototypu. K vytvoření grafického rozhraní jsem se rozhodl použít grafický vývojový designer TouchGFX, od společnosti STMicroelectronics. Designer umožňuje vývojáři vygenerovat projekt od základu a pro vyzkoušení fungování aplikace obsahuje také simulátor. Použitý vývojový kit je plně kompatabilní a podporuje snadnou implementaci aplikace.

3.1 TouchGFX

TouchGFX je aplikace z balíku X-Cube, obsahuje vše pro úplnou implementaci grafického uživatelského prostředí (GUI) pro hardware založený na mikrokontrolérech STM32. TouchGFX se skládá ze tří hlavních částí.

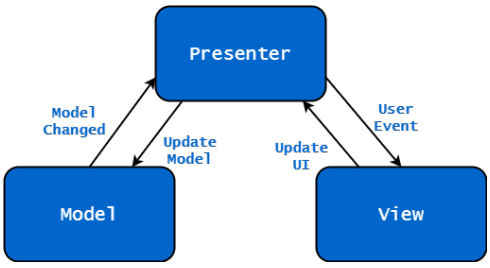
- **Designer** - Program umožňující vývojáři pohodlně vytvořit grafickou aplikaci.
- **Generator** - STM32CubeMX plugin, který umožňuje nakonfigurovat vlastní abstraktní vrstvu pro hardware s STM32.
- **Engine** - C++ framework, který řídí GUI aplikaci. Zpracovává aktualizace obrazovek, událostí a časování. Poskytuje tak maximální výkon s minimální zátěží procesoru a jednotlivých pamětí.

3.1.1 Architektura

Architektura aplikace je známá jako Model-View-Presenter (MVP). Tato struktura umožňuje vývojáři oddělit kód do několika částí, kdy každá má vlastní odpovědnost. Schéma struktury je na obrázku 3.1. Kód je jednodušší, má snazší údržbu a můžeme ho snadněji opakovaně použít. Jelikož logika, presenter, je oddělena od vizuální vrstvy, je snazší jednotlivé vrstvy testovat odděleně. MVP se skládá ze tří hlavních tříd:

- **Model** - Definuje data, která se zobrazí v uživatelském rozhraní.
- **View** - Pasivní rozhraní, které zobrazuje data z modelu a spojuje uživatelské příkazy (události) s presenterem.

- **Presenter** - Chování je ovlivněno na základě dat z tříd model a view. Získává z nich data a zároveň je formátuje pro zobrazení zpět pomocí třídy view.

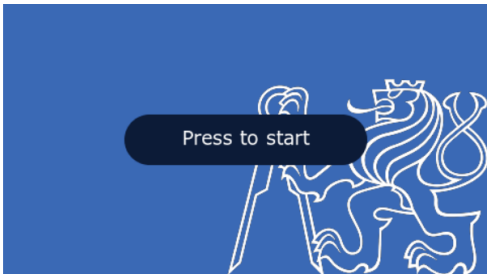


Obrázek 3.1: Architektura Model-View-Presenter

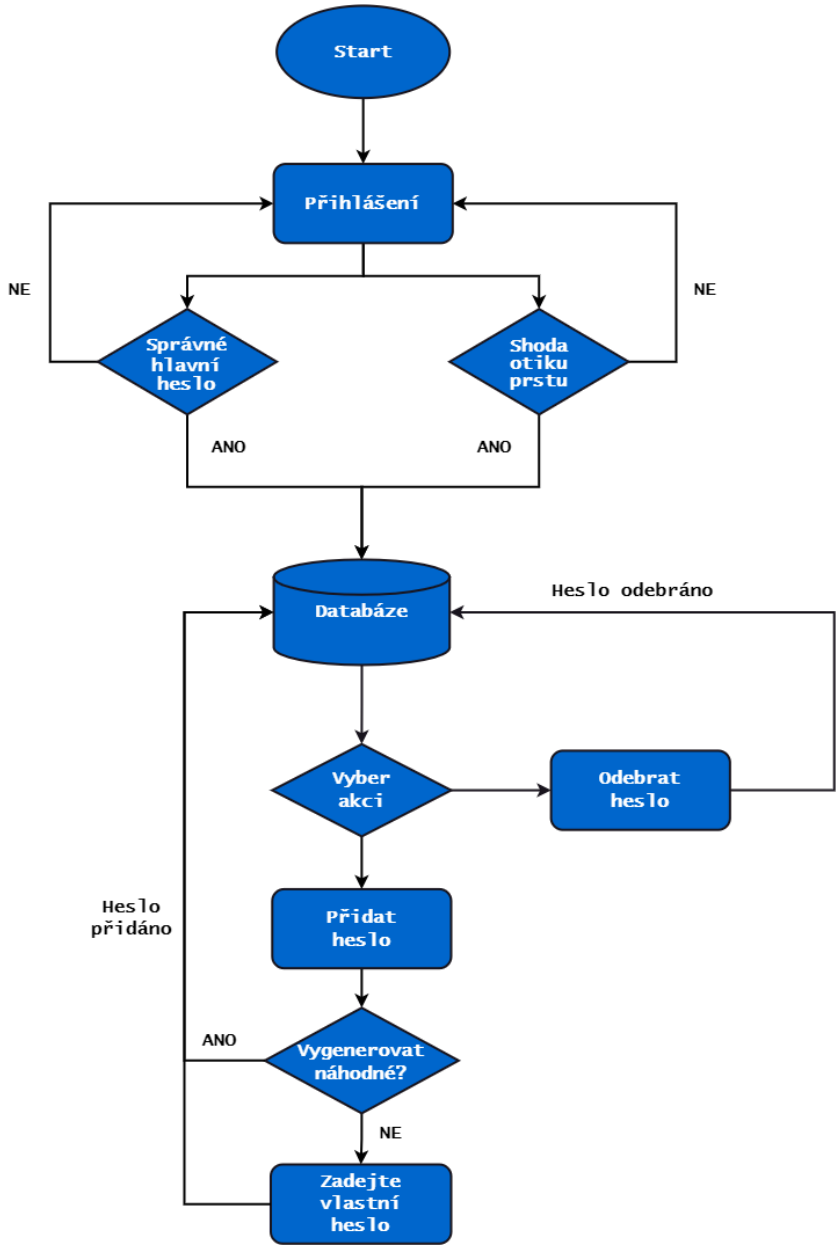
V TouchGFX je komunikace s backend systémem, částí, která není viditelná pro uživatele, prováděna pomocí třídy Model. Backend aplikace je sofwartová část, která získává události z uživatelského rozhraní a zároveň jiné události posílá zpět do uživatelského rozhraní. Pojmem událost (označováno jako event) je myšleno například meření ze senzoru. Uživatel má možnost umístit backend v podobě separátního úkolu na stejný mikroprocesor jako aplikaci, na jiný, separátní procesor nebo na cloudový modul. Na umístění prakticky nezáleží, dokud je možná komunikace se zbytkem systému, ale dobrou praxí je mít události ve zdrojovém souboru, který souvisí s daným modulem.

3.2 Vlastní firmware

Hlavním cílem je vytvořit funkční aplikaci s jednoduchým a intuitivním ovládáním. Chod firmwaru je znázorněn na vývojovém diagramu na obrázku 3.3. Po spuštění aplikace přivítá uživatele úvodní obrazovka, která je na obrázku 3.2 s tlačítkem pro vstup do přihlašovací části.



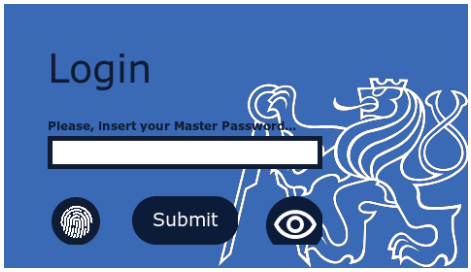
Obrázek 3.2: Uvítací obrazovka



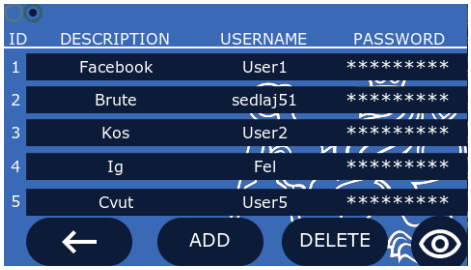
Obrázek 3.3: Vývojový diagram aplikace

Aplikace je zabezpečena hlavním heslem a otiskem prstu, uživatel má tedy možnost výběru, který způsob přihlášení zvolí. Na obrázku 3.4 dole jsou vidět piktogramy, které slouží pro přihlášení pomocí otisku prstu, tlačítko *Submit* pro ověření hesla, které uživatel zadá do řádku nahoře nebo piktoram oka pro zobrazení či zakrytí zadaného hesla.

Po úspěšném přihlášení se uživatel dostane do databáze, kde nalezne přehled uložených přístupových údajů, viz obrázek 3.5. Hesla zakrytá v databázi lze odkrýt dvěma způsoby, kliknutím na samotné heslo nebo tlačítkem s ikonou *oka* v pravém dolním rohu, které odhalí všechna hesla.

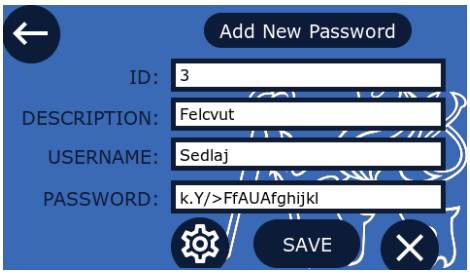


Obrázek 3.4: Přihlášení do aplikce

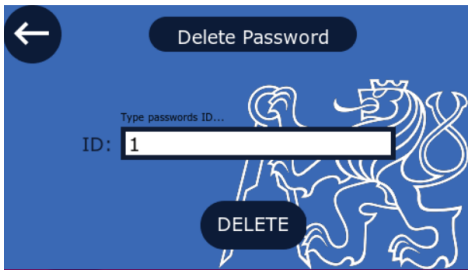


Obrázek 3.5: Databáze uložených hesel

Správa samotné databáze je velice snadná. Uživatel má možnost hesla přidávat a odebírat. Pro přidání hesla zadává uživatel číselný identifikátor (ID), popis, přihlašovací jméno a samotné heslo. To může být náhodně vygenerováno pomocí zabudovaného generátoru náhodných hesel, který je dostupný pod tlačítkem s ozubeným kolečkem. Odebrání záznamů funguje na základě zadání ID hesla, které má být odstraněno. Na následujících obrázcích jsou přidány snímky obrazovek pro přidání 3.6a a odebrání hesla 3.6b.



(a) : Přidání hesla



(b) : Odebrání hesla

Obrázek 3.6: Obrazovky pro přidání a odebrání hesla

Závěr

Použití správců hesel se v dnešní době zdá býti nepostradatelným uživatelským nástrojem. S průměrem 100 hesel na uživatele je prakticky nemožné si všechna hesla zapamatovat. Použití správců s sebou přináší mnoho výhod. Hlavní výhodou je možnost ukládat přístupové údaje na jediné bezpečné místo, které je přístupné pod jedním hlavním heslem. Komplexnější správci disponují několika dalšími funkcemi, např. náhodným generováním hesel, automatickým doplňováním do přihlašovacích formulářů, synchronizací mezi zařízeními či poskytnutím zálohy.

Nejrozšířenějším typem jsou správci v prohlížečích. Jsou součástí každého webového prohlížeče, který je poskytuje zdarma. Pro náročnější uživatele doporučuji správce, kteří jsou poskytováni v podobě desktopové aplikace nebo rozšíření do prohlížeče. Nabízejí i komplexnější funkce. Hlídají, zda-li bylo nějaké z uložených hesel prolomeno. Umožňují generovat náhodná hesla podle potřeb uživatele. Poskytují bezpečné zálohy po určitém časovém úseku a umožňují snadný přenos databáze mezi zařízeními. Jednou z nevýhod je jejich zpoplatnění, možnost poruchy zařízení nebo jeho krádež. Posledním dvěma rizikům se však dá zodpovědným zálohováním zcela vyhnout.

Dalším cílem práce bylo vytvoření funkčního prototypu hardwarového správce hesel. Ten byl realizován na vývojovém kitu STM32H7B3I od společnosti STMicroelectronics. Pro autorizaci uživatele byl použit senzor otisků prstů DY50, jehož drivery jsou kompatibilní s mikrokontroléry ATmega. Za účelem vyhodnocení snímků bylo nutné přidat Arduino Nano. Pro komunikaci s kitem byl využit GPIO pin, jehož stav signalizuje shodu, či neshodu otisku s databází. V reálném světě je varianta GPIO pinu vysoce nebezpečná, proto byla použita pouze v rámci proof of concept.

Finálním výsledkem je funkční prototyp správce hesel, který disponuje dvoufázovým ověřením uživatele, generátorem náhodných hesel a možností ukládání přístupových údajů.

Zkratky

AD Analog-Digital. 30

AES Advanced Encryption Standard. 5, 7–9, 14, 15

CTSS Compatible Time-Sharing System. 3, 5

DES Data Encryption Standard. 5, 6, 8

DPAPI Windows Data Protection API. 13

ENISA European Union Agency for Cybersecurity. 9

GRAM Graphics Random Access Memory. 29

GUI Graphical user interface. 11, 21, 33

HID Human Interface Device. 11

HSM Hardwarový bezpečnostní modul. 16

I2C Inter-Integrated Circuit. viii, 28–30

IdP Identity Provider. 18

LCD Liquid Crystal Display. vi, 29

LED Light-Emitting Diode. 21, 26, 30, 31

MD5 Message Digest 5. 7

MFA Multi-Factor Authentication. 19

MPUs Memory Protection Units. 15

MVP Model-View-Presenter. 33

NBS National Bureau of Standards. 6

NIST National Institut of Standards. 6, 7, 19

PCB Printed Circuit Board. 31

PPUs Peripheral Protection Units. 16

PWM Pulse Width Modulation. 24

RAM Random Access Memory. 30

RGB Red Green Blue. 29

RISC Reduced Instruction Set Computer. 30

ROT Root Of Trust. 17

SAW Sonic Acoustic wave. 24–26

SCL Serial Clock. 28, 29

SDA Serial Data. 28, 29

SDIO Secure Digital Input/Output. 29

SMPUs Share Memory Protection Units. 16

SSO Single Sign-On. 18

TWI Two Wire Interface. 28

UART Universal asynchronous receiver-transmitter. viii, 23, 27, 30, 31

USB Universal Serial Bus. 13

Literatura

[1] *A Brief History of Passwords*. Online. Dashlane. 2021. Dostupné z: `https://ripleyprd.wpengine.com/a-brief-history-of-passwords`. [cit. 2024-02-01].

[2] *Computer password inventor dies aged 93*. Online. BBC News. 2019. Dostupné z: `https://www.bbc.com/news/technology-48988091`. [cit. 2024-02-01].

[3] ROWE, Adam. *Study Reveals Average Person Has 100 Passwords*. Online. Tech.co. 2023. Dostupné z: `https://tech.co/password-managers/how-many-passwords-average-person`. [cit. 2024-02-02].

[4] BUCKLAND, Matthew. *The web 2.0 password crisis*. Online. Thought Leader. 2008. Dostupné z: `https://thoughtleader.co.za/a-web-20-password-crisis/`. [cit. 2024-02-02].

[5] DANI, Carlo. *Cifrante Enigma in dotazione alla Luftwaffe*. Online. In: Wikimedia commons. 2018. Dostupné z: `https://commons.wikimedia.org/wiki/File:Enigma_machine_Luftwaffe.jpg`. [cit. 2024-05-01].

[6] NEWTON, Mike. *M-209 Tactical Cipher Machine*. Online. In: Wikimedia commons. Dostupné z: `https://commons.wikimedia.org/wiki/File:M-209.jpg`. [cit. 2024-05-01].

[7] GREGERSEN, Erik. *History of cryptology*. Online. Encyclopedia Britannica. Dostupné z: `https://www.britannica.com/topic/cryptology`. [cit. 2024-02-06].

[8] ELLIS, Scott R. A Cryptography Primer: Enigma. In: *Computer and Information Security Handbook*. 3rd. Elsevier, 2017, s. 35-58. ISBN 978-0-12-803843-7.

[9] SHACKLETT, Mary E. a LOSHIN, Peter. *MD5*. Online. Security. 2021. Dostupné z: `https://www.techtarget.com/searchsecurity/definition/MD5`. [cit. 2024-02-10].

[10] SHINDER, Littlejohn a CROSS, Michael. *Scene of the Cybercrime*. 2. Syngress, 2008. ISBN 978-1-59749-276-8.

[11] *Binance Academy*. Online. 2019, 2022-16-10. Dostupné z: https://academy.binance.com/cs/articles/symmetric-vs-asymmetric-encryption. [cit. 2024-02-10].

[12] Robert Morris and Ken Thompson. *Password security: a case history*. Commun. ACM 22. 1979. 594–597. DOI: 10.1145/359168.359172

[13] JUNG, Jason. *What are Salted Passwords and Password Hashing?* Online. Okta. 2021. Dostupné z: https://www.okta.com/blog/2019/03/what-are-salted-passwords-and-password-hashing/. [cit. 2024-02-17].

[14] COBB, Michael a BERNSTEIN, Corinne. *Advanced Encryption Standard (AES)*. Online. Security. 2021. Dostupné z: https://www.techtarget.com/searchsecurity/definition/Advanced-Encryption-Standard. [cit. 2024-02-09].

[15] KERNER, Sean Michael. *Password manager*. Online. Security. 2023. Dostupné z: https://www.techtarget.com/searchsecurity/definition/password-manager. [cit. 2024-02-02].

[16] *Authentication Methods*. Online. Dostupné z: https://www.enisa.europa.eu/topics/incident-response/glossary/authentication-methods[cit. 2024-02-02].

[17] *Top tips for staying secure online*. Online. NCSC.GOV.UK. 2018, 2021. Dostupné z: https://www.ncsc.gov.uk/collection/top-tips-for-staying-secure-online/password-managers. [cit. 2024-02-03].

[18] VALIAUGAITĖ, Inga. *Are password managers secure?* Online. Cybernews. 2024. Dostupné z: https://cybernews.com/best-password-managers/are-password-managers-safe/. [cit. 2024-02-03].

[19] UNIQKEY. *Cloud-based vs Local Password Storage: The Pros and Cons*. Online. The Uniqkey Blog. Dostupné z: https://blog.uniqkey.eu/cloud-based-vs-local-password-storage/. [cit. 2024-02-04].

[20] AZTECH IT SOLUTIONS. *Benefits of Using a Password Manager*. Online. AZtech IT Solutions. 2021. Dostupné z: https://www.aztechit.co.uk/blog/benefits-of-a-password-manager. [cit. 2024-02-04].

[21] MCCARNEY, Daniel. *PASSWORD MANAGERS: COMPARATIVE EVALUATION, DESIGN, IMPLEMENTATION AND EMPIRICAL ANALYSIS*. Diplomová práce. OTTAWA, ONTARIO, CANADA: CARLETON UNIVERSITY, 2013.

[22] PALANT, Wladimir. *Master password in Firefox or Thunderbird? Do not bother!*. Online. Almost Secure.

2018. Dostupné z: https://palant.info/2018/03/10/master-password-in-firefox-or-thunderbird-do-not-bother/. [cit. 2024-02-04].

[23] ILASCU, Ionut. *Some Hardware-based Password Managers Have Poor Security*. Online. BleepingComputer. 2019. Dostupné z: https://www.bleepingcomputer.com/news/security/some-hardware-based-password-managers-have-poor-security/. [cit. 2024-04-10].

[24] *5 ELEMENTS TO SECURE EMBEDDED SYSTEM — PART #2 ROOT-OF-TRUST (ROT)*. Online. Beningo Embedded Group. 2021. Dostupné z: https://www.beningo.com/5-elements-to-secure-embedded-system-part-2-root-of-trust-rot/. [cit. 2024-02-15].

[25] STEPHAN ELECTRONICS. *Tindie*. Online. Tindie. 2024. Dostupné z: https://www.tindie.com/products/stephanelec/mooltipass-mini-ble-authenticator/. [cit. 2024-05-01].

[26] RISTO, Avila. *How to Use the Best Security for Your Embedded System*. Online. Nenalezený vydavatel. 2021. Dostupné z: https://www.qt.io/embedded-development-talk/how-to-use-the-best-security-for-your-embedded-system. [cit. 2024-02-15].

[27] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. FIPS PUB 140-3, *SECURITY REQUIREMENTS FOR CRYPTOGRAPHIC MODULES*. DOI: 10.6028/NIST.FIPS.140-3

[28] *5 ELEMENTS TO SECURE EMBEDDED SYSTEMS — PART #1 HARDWARE BASED ISOLATION*. Online. Beningo Embedded Group. 2021. Dostupné z: https://www.beningo.com/5-elements-to-secure-embedded-systems-part-1-hardware-based-isolation/. [cit. 2024-02-15].

[29] KLEIDERMACHER, David a KLEIDERMACHER, Mike. *Embedded Security: Practical Methods for Safe and Secure Software and Systems Development*. 1. Elsevier, 2012. ISBN 978-0-12-386886-2.

[30] PRESS, Rambus. *Hardware Root of Trust: Everything you need to know*. Online. Rambus. 2023. Dostupné z: https://www.rambus.com/blogs/hardware-root-of-trust/. [cit. 2024-04-10].

[31] BBC. *Face ID iPhone X 'hack' demoed live with mask by Bkav*. Online. BBC News. 2017. Dostupné z: https://www.bbc.com/news/av/technology-41992610. [cit. 2024-02-05].

[32] KYLE JOHNSON, By:. *Use these 6 user authentication types to secure networks*. Online. Security. 2023. Dostupné

z: https://www.techtarget.com/searchsecurity/tip/Use-these-6-user-authentication-types-to-secure-networks. [cit. 2024-02-05].

[33] GRASSI, Paul A.; FENTON, James L. a BURR, William E. National Institute of Standards and Technology Special Publication 800-63B. *Nenalezený vydavatel*. 2017. DOI: 10.6028/NIST.SP.800-63b.

[34] YIPENG, Lu. *Piezoelectric Micromachined Ultrasonic Transducers for Fingerprint Sensing*. Disertační práce. Davis, California: UNIVERSITY OF CALIFORNIA DAVIS, 2015.

[35] MAKEMAGAZINDE. *Arduino nano*. Online. In: Wikimedia Commons. Dostupné z: https://commons.wikimedia.org/wiki/File:Arduino_nano_isometr.jpg. [cit. 2024-05-04].

[36] NOVÁK PH.D, Ing. Martin. Principy dotykových panelů a jejich využití v automatizaci. *Elektro*. 2009, č. 2, s. 46-48.

[37] *Resistive Touch Panel Construction and Working Principles*. Online. Riverdi. Dostupné z: https://riverdi.com/blog/resistive-touch-panel-construction-and-working-principles. [cit. 2024-04-12].

[38] *Advantages and Applications of Infrared Touch Display*. Online. Shenzhen Touch Think Intelligence Co.,Ltd. 2017. Dostupné z: https://www.touchtecs.com/industry-news/advantages-and-applications-of-infrared-touch-display.html. [cit. 2024-04-12].

[39] *Surface Acoustic Wave (SAW)*. Online. Canvys. 2010. Dostupné z: https://www.canvys.com/custom-touch-screen-monitors/touch-screen-surface-acoustic-wave-touch/. [cit. 2024-04-12].

[40] PEŇA, Eric a LEGASPI, Mary Grace. *UART: A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter*. Online. Analog Devices. 2020. Dostupné z: https://www.analog.com/en/resources/analog-dialogue/articles/uart-a-hardware-communication-protocol.html. [cit. 2024-03-22].

[41] NOVIELLO, Carmine. *Mastering STM32*. Leanpub, 2015. ISBN 979-8857191675.

[42] TECHTARGET, Contributor. *Token*. Online. WhatIs. Https://www.techtarget.com/whatis/definition/token. Dostupné z: https://www.techtarget.com/whatis/definition/token. [cit. 2024-02-15].

[43] AFZAL, Sal. *I2C Primer: What is I2C? (Part 1)*. Online. Analog Devices. Dostupné z: https://www.analog.com/en/resources/technical-articles/i2c-primer-what-is-i2c-part-1.html. [cit. 2024-04-13].

[44] *Cybernews*. Online. 2024. Dostupné z: https://cybernews.com/best-password-managers/are-password-managers-safe/. [cit. 2024-02-03].

[45] *BBC News*. Online. 2019. Dostupné z: https://www.bbc.com/news/technology-48988091. [cit. 2024-02-01].

[46] OPENSYSTEMS MEDIA. *Embedded Computing Design*. Online. Embedded Computing Design. 2021. Dostupné z: https://embeddedcomputing.com/technology/processing/interface-io/tft-lcd-parallel-interface-comparison-mcu-vs-rgb. [cit. 2024-05-01].