

**Česká zemědělská univerzita v Praze**

**Provozně ekonomická fakulta**

**Katedra informačního inženýrství**



**Diplomová práce**

**Inventarizační zařízení založené na platformě Arduino a  
mikročipu ESP32**

**Bc. Vojtěch Šebesta**

© 2023 ČZU v Praze



# ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

## ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Vojtěch Šebesta

Informatika

Název práce

**Inventarizační zařízení založené na platformě Arduino a mikročipu ESP32**

Název anglicky

**Inventory device based on Arduino platform and ESP32 microchip**

---

### Cíle práce

Cílem teoretické části diplomové práce je pomocí studia odborné literatury podrobně popsat mikrokontrolér Arduino a mikročip ESP32. Pomocí dostupných zdrojů zhodnotit finanční náročnost provozu vlastního webového serveru a komerčních webhostingových služeb. Dále popsat principy inventarizace a jejího praktického využití v domácnosti. Nakonec definovat princip technologie 3D tisku a analyzovat dostupné CAD editory a další nezbytný software k tvorbě modelů.

Cílem praktické části je navrhnout mobilní hardwarové zařízení sloužící k inventarizaci (pomocí skenování QR a čárových kódů, a načítání RFID tagů). Získaná data ze zařízení se přenesou do online aplikace, kde budou uložena a zpracovávána pro přehledné zobrazení a editaci formou tabulek. Pouzdro na zařízení bude navrženo ve vhodném CAD editoru. Pro tisk budou porovnány technické vlastnosti různých filamentů (PLA, PETG, ABS) a vybere se ten, který nejvíce vyhoví požadavkům pro použití výrobku.

### Metodika

Nejprve bude popsán a vybrán vhodný přípravek Arduino, kde bude kladen důraz především na počet potřebných vstupů a výstupů, kompaktní velikost a spotřebu. Dále bude popsán mikročip ESP32 a bude porovnáno, zda je vhodnější alternativou použít čip samostatně, či již integrovaný na desce Arduino.

Poté bude navržen model, na základě kterého bude pomocí vhodných komponent výsledný výrobek sestaven. Při navrhování a sestavování bude brán zřetel na kompaktnost a mobilitu tohoto zařízení. Pro model se navrhne a vytiskne ochranný obal.

V další části bude vytvořena webová aplikace na vybrané platformě a softwarová výbava pro hardware.

Na závěr bude výrobek testován v modelovém prostředí.

**Doporučený rozsah práce**

60-80 stran

**Klíčová slova**

Inventarizační zařízení, Arduino, ESP32, 3D tisk, CAD, C++, PHP

---

**Doporučené zdroje informací**

KURNIAWAN, Agus. Internet of Things Projects with ESP32. Birmingham: Packt Publishing, 2019. ISBN 978-1-78995-687-0.

MONK, Simon. Programming Arduino. Getting Started with Sketches. United States: McGraw-Hill Education, 2016. ISBN 978-1-25-964163-3.

NOORANI, Rafiq. 3D printing: technology, applications, and selection. Boca Raton: CRC Press, 2018. ISBN 978-1-4987-8375-0.

---

**Předběžný termín obhajoby**

2022/23 LS – PEF

**Vedoucí práce**

Ing. Dana Vynikarová, Ph.D.

**Garantující pracoviště**

Katedra informačního inženýrství

---

Elektronicky schváleno dne 4. 11. 2022

**Ing. Martin Pelikán, Ph.D.**

Vedoucí katedry

---

Elektronicky schváleno dne 28. 11. 2022

**doc. Ing. Tomáš Šubrt, Ph.D.**

Děkan

V Praze dne 14. 03. 2023

## **Čestné prohlášení**

Prohlašuji, že svou diplomovou práci „Inventarizační zařízení založené na platformě Arduino a mikročipu ESP32“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 20.11.2023

---

## **Poděkování**

Rád bych touto cestou poděkoval Ing. Daně Vynikarové, Ph.D. za odborné vedení, trpělivost a cenné rady při konzultacích této práce. Dále bych rád poděkoval Ing. Michalu Kolačnému za odborné poradenství při volbě komponent.

# Inventarizační zařízení založené na platformě Arduino a mikročipu ESP32

## Abstrakt

Tato diplomová práce se zabývá návrhem zařízení usnadňující domácí inventarizaci a evidenci majetku. První část práce formuluje teoretická východiska. Je vysvětlen princip fungování mikrokontrolérů, spolu s představením platformy Arduino a ESP. Následně je představen proces inventarizace a formy evidence majetku, spolu s nástroji k tomu užívanými. Také je vysvětlen princip aditivní výroby v komparaci se subtraktivní výrobou. Jsou popsány nejběžnější formy aditivní výroby, tedy FDM, SLA a SLS tiskárny a poté jednotlivé materiály využívané k tisku na FDM tiskárnách. V praktické části práce jsou stanoveny požadavky na funkce zařízení. Na základě požadavků jsou vybrány vhodné komponenty a je vytvořen designový návrh finálního ochranného boxu. Následně jsou komponenty propojeny v modelovém prototypu a naprogramovány pro ověření jejich správné funkčnosti. Funkční komponenty jsou připájeny k řídicímu mikrokontroléru ESP32 a je vytvořena softwarová vybava v jazyce C++ a Wiring. Současně je vytvořena databáze spravující data o majetku, spolu s webovou aplikací nad ní pracující. Na základě celkových rozměrů celého řešení je vytvořen CAD model, který je vytištěn na FDM tiskárně.

**Klíčová slova:** mikrokontrolér, inventarizační zařízení, Arduino, ESP32, aditivní výroba, FDM, CAD, C++, PHP, HTML, HTTP

# **Inventory device based on Arduino platform and ESP32 microchip**

## **Abstract**

This Master's thesis deals with the design of a device that facilitates home inventory and property records. The first part of the thesis consists of the formulation of theoretical foundations. The principle of microcontroller operation is explained, along with an introduction to the Arduino and ESP platforms. Then, the inventory process and forms of property records are introduced, along with tools designed for this purpose. The principle of additive manufacturing is also explained in comparison to subtractive manufacturing. The most common forms of additive manufacturing, namely FDM, SLA and SLS printers are described, followed by individual materials used for printing on FDM printers. In the practical part of the thesis, requirements for the functionality of the device are defined. Based on these requirements, suitable components are selected and a design proposal for the final protective box is created. The components are then assembled into a model prototype and programmed to verify correct functionality. Functional components are soldered to the ESP32 control microcontroller and software is written in C++ and Wiring language. At the same time, a database is created to manage the property data and a web application is created to work over it. Based on the overall dimensions of the entire solution, a CAD model is created and printed on an FDM printer.

**Keywords:** microcontroller, inventory device, Arduino, ESP32, additive manufacturing, FDM, CAD, C++, PHP, HTML, HTTP



# Obsah

<b>1 Úvod.....</b>	<b>11</b>
<b>2 Cíl práce a metodika .....</b>	<b>12</b>
2.1 Cíl práce .....	12
2.2 Metodika .....	12
<b>3 Teoretická východiska .....</b>	<b>13</b>
3.1 Inventarizace .....	13
3.1.1 Možnosti evidence majetku .....	13
3.1.2 Dostupné identifikátory .....	14
3.2 Mikrokontrolér .....	18
3.2.1 Arduino .....	21
3.2.2 ESP32.....	23
3.3 Webový server .....	25
3.3.1 Provozní náklady webserveru .....	27
3.4 Aditivní výroba .....	29
3.4.1 Tvorba modelu .....	30
3.4.2 3D tisk.....	31
3.4.3 Tiskové filamenty .....	32
<b>4 Vlastní práce .....</b>	<b>35</b>
4.1 Návrh zařízení .....	35
4.1.1 Funkční požadavky .....	35
4.1.2 Volba komponent.....	36
4.1.3 Design modelu .....	41
4.2 Konstrukce zařízení a doprovodné webové aplikace .....	42
4.2.1 Sestavení prototypu.....	42
4.2.2 Kompletace řešení.....	50
4.2.3 Tvorba a tisk 3D modelu ochranného boxu.....	53
4.2.4 Tvorba menu .....	56
4.2.5 Tvorba databáze .....	58
4.2.6 Vývoj webové aplikace pro správu dat v databázi .....	60
4.2.7 Komunikace ESP32 s databází .....	62
4.3 Testování .....	66
<b>5 Výsledky a diskuse .....</b>	<b>67</b>
<b>6 Závěr.....</b>	<b>71</b>
<b>7 Seznam použitých zdrojů .....</b>	<b>73</b>
<b>8 Seznam obrázků, tabulek, grafů a zkratk.....</b>	<b>79</b>

8.1	Seznam obrázků .....	79
8.2	Seznam tabulek.....	80
8.3	Seznam úryvků kódu .....	80
8.4	Seznam použitých zkratek.....	81
<b>Přílohy</b>	.....	<b>83</b>

# 1 Úvod

Množství zásob a majetku, které se v běžné domácnosti vyskytují, stále roste. Udržet si ale aktuální přehled, kde, co a kolik majetku domácnost vlastní, se stává složitější. Je tedy běžné, že vznikají různé seznamy věcí. Například evidence knih v knihovně, včetně poznámek, komu byl jaký titul vypůjčen nebo vlastní zhodnocení knihy.

Možností, jak vést a spravovat archiv je mnoho. Nejběžnějším pomocníkem pro evidenci majetku v domácnosti je softwarový tabulkový procesor, nebo jeho papírová verze. Ty však mají společné nedostatky, kterými může být absence filtrování na základě parametrů, rychlé vyhledávání, nemožnost nahlížení z několika míst současně nebo složitá aktualizace údajů. Řešením těchto nedostatků mohou být komerčně dostupné produkty. Mezi takové produkty se řadí komplexní profesionální řešení. Tyto produkty se skládají z aplikace, která pracuje s uloženými daty a skeneru umožňujícím rychlou úpravu záznamů v databázi. Skenery načítají data z produktů s využitím technologií, jako jsou čárové kódy nebo RFID (identifikace na rádiové frekvenci) čipy. Nevýhodou jsou však vysoké pořizovací náklady, včetně nutnosti odborné instalace a servisu. Většinu domácností odradí zmíněné nedostatky od pořízení si takových systémů.

V této diplomové práci je navrženo cenově výhodnější mobilní zařízení srovnatelné s profesionálním zařízením. Dále je vytvořena webová aplikace pro správu záznamů majetku, který chce domácnost evidovat. Záznamy jsou uloženy v databázi a s využitím webové aplikace má mít uživatel možnost si všechna data vyexportovat a zálohovat na jiném úložišti. O místě provozu webové aplikace a databáze si je schopen rozhodnout uživatel sám. Může vybírat z placené webhostingové služby, pronájmu serveru, nebo provozu na vlastním domácím serveru. Návrh celého řešení, včetně vybraných komponent, je postaveno výhradně na open-source technologiích.

Teoretická část této práce se skládá z rešerše odborné literatury a uvedení do problematiky mikrokontrolérů a webových serverů. Další kapitoly teoretického úvodu vysvětlují principy aditivní výroby a inventarizace. V praktické části práce jsou vybrány a podrobně popsány jednotlivé komponenty a nástroje využité pro zhotovení inventarizačního zařízení. Výstup práce byl následně otestován na správě knih v domácí knihovně.

## **2 Cíl práce a metodika**

### **2.1 Cíl práce**

Hlavním cílem této diplomové práce je navrhnout a vyrobit zařízení pro inventarizaci majetku. Zařízení je vybaveno akumulátorem a komunikaci se serverem zajišťuje bezdrátová WiFi síť. Výrobek je chráněn ochranným boxem vytištěným na 3D FDM (tavené depoziční modelování) tiskárně. Uživatelská data jsou uložena v databázi, ke které má uživatel přístup s pomocí navržené webové aplikace nebo s využitím inventarizačního zařízení.

### **2.2 Metodika**

Diplomová práce je rozdělena do dvou částí, a to teoretické a praktické. Postup zpracování teoretické části práce spočívá v rešerši odborné literatury a dalších dostupných informačních zdrojích. Na základě těchto informací byla formulována východiska teoretické části práce.

V praktické části práce jsou popsány funkční požadavky, které má zařízení splňovat. Na základě těchto požadavků byly vybrány a zakoupeny jednotlivé komponenty pro stavbu zařízení. Po otestování vzájemné kompatibility bylo zařízení sestaveno a vybaveno vhodnou softwarovou výbavou. Ochranný box, navržený v CAD (počítačem podporované navrhování) editoru, byl vytištěn na 3D FDM tiskárně. Funkční zařízení bylo propojeno s databází, s jejímiž daty pracuje. S daty pracuje i webová aplikace, navržená pro pohodlnou správu dat i bez zařízení. Hotové zařízení bylo testováno správou a inventarizací knih v domácí knihovně.

Poznatky a výstupy, získané během návrhu, konstrukce a testování jsou na závěr shrnuty a diskutovány.

## **3 Teoretická východiska**

### **3.1 Inventarizace**

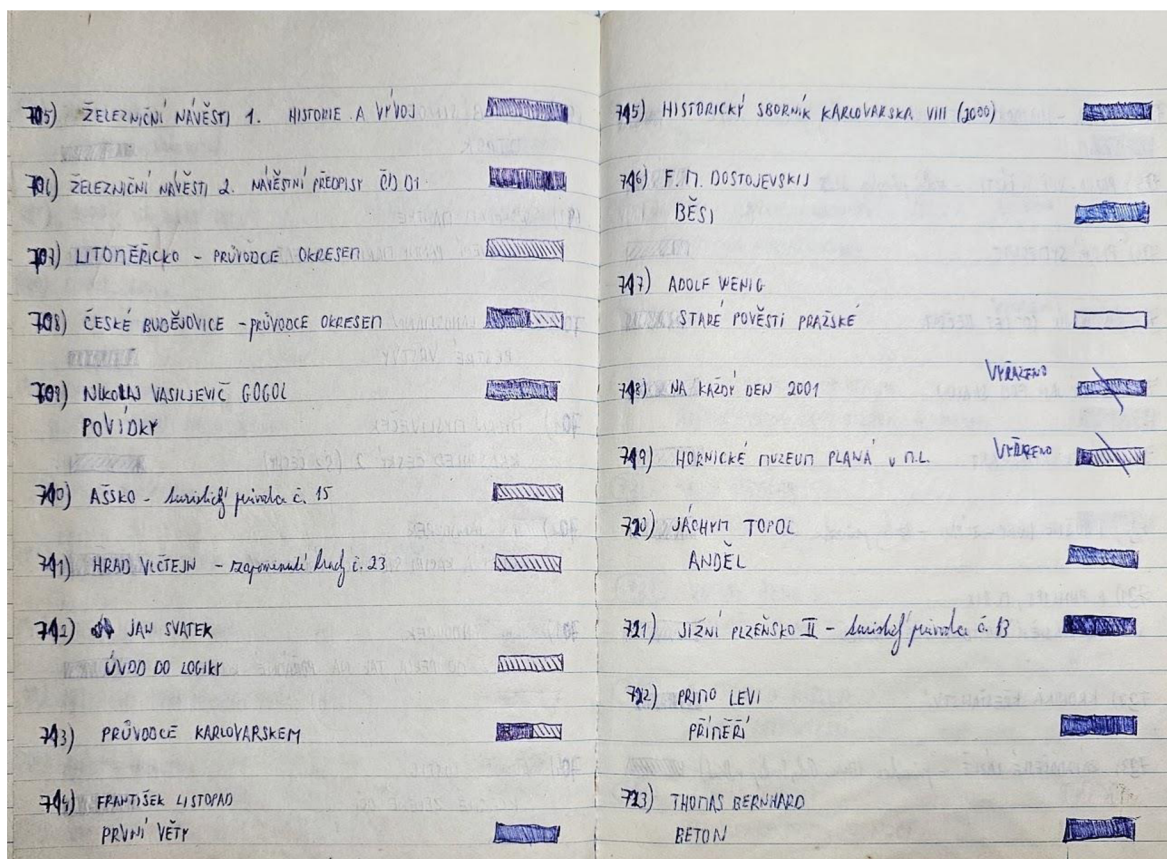
Inventarizace je proces, při kterém se ověřuje skutečný stav fyzického majetku, zboží a zásob oproti vedené evidenci. Proces zjišťování množství kusů nebo váhy skutečného stavu se nazývá inventura [1]. Přehled o stavu toho, kolik a jakých zásob podnik nebo domácnost má, je nezbytný ukazatel potřebný pro zvýšení efektivity a snížení nákladů. Například domácnost může před Vánoce zajímat, kolik kilogramů hladké mouky, cukru a másla má, aby byla schopna napéct požadované množství cukroví. Neméně důležité jsou i další informace o zásobách, jako je datum jejich expirace nebo údaje o výrobci. Firmám se pak díky přesnému přehledu skladových zásob daří mít zboží na skladech co nejméně. V ideálním případě je pak celý dodavatelský řetězec koordinovaný, aby žádný z jeho článků nemusel dlouho platit vysoké poplatky za uskladňování a správu zásob a zboží co nejpružněji odráželo proměnlivou spotřebitelskou poptávku. [2]

Tato práce se věnuje domácí inventarizaci, kde je přehled o stavu zásob také velice důležitý. Nejčastěji je sledován stav zásob potravin, drogistického vybavení, dílenských a staveních zásob a v neposlední řadě jsou sledovány údaje o stavu domácí knihovny. Například údaje o počtu knih, které byly zapůjčeny sousedovi zapsané pouze na papíru, bez možnosti automatického upozornění na expiraci výpůjční doby, bývají často zapomenuty nebo ztraceny. Tato problematika lze přenést na většinu majetku domácnosti.

#### **3.1.1 Možnosti evidence majetku**

Nástrojů pro pohodlnou evidenci zásob a majetku je mnoho, od těch nejtriviálnějších, jako jsou ručně psané záznamy na papír, přes jednoduché elektronické tabulky, až po komplexní systémy evidence majetku. Tužka a papír je, co se týče pořizovacích nákladů, velice výhodná. Ovšem s rostoucím počtem záznamů se stává neudržitelnou – rychlost vyhledávání a aktualizace dat přímo úměrně klesá s počtem řádků. Navíc je takový archiv náchylný na mechanické poškození. Elektronická tabulka vedená v nějakém z dostupných tabulkových procesorů odstraňuje většinu nedostatků papírové evidence, pokud je archiv správně nastaven. Navíc díky cloudovým řešením je možné k záznamům přistupovat a data editovat simultánně z několika zařízení. Ovšem rychlost přidávání a vyhledávání

jednotlivých položek je kvůli nutnosti manuálního zadávání stále pomalá a zdlouhavá. Komerční řešení na správu zásob a majetku odstraňují tento nedostatek. Využívají elektronické databáze, ve kterých se záznamy aktualizují pomocí nástrojů jako jsou ruční skenery a mobilní aplikace. Takovéto řešení by pro domácnost mohlo být ideální, ovšem komplexita řešení, jeho pořizovací a provozní náklady jsou natolik vysoké, že si domácnost pravděpodobně poradí jinak. Na následujícím obrázku 1 je fotografie inventarizační knihy.



Obrázek 1 – Ručně vedená inventarizační kniha o stavu rodinné domácí knihovny

### 3.1.2 Dostupné identifikátory

Manuální úpravu dat, ať s využitím psacích potřeb, nebo vypisováním na klávesnici si lze usnadnit využitím moderních technologií. Velký průlom provedli v roce 1949 Norman Joseph Woodland a Bernard Silvera, kteří zažádali o patent na „klasifikační přístroj a metodu.“ Zmíněný patent měl automatizovat proces nákupu potravin. Patent jim byl udělen až o tři roky později a metoda ihned zaujala řadu společností z průmyslového odvětví.

Norman Joseph Woodland a Bernard Silvera tak položili základ čárovému kódu, který v různých formách používají lidé po celém světě dodnes. [3]

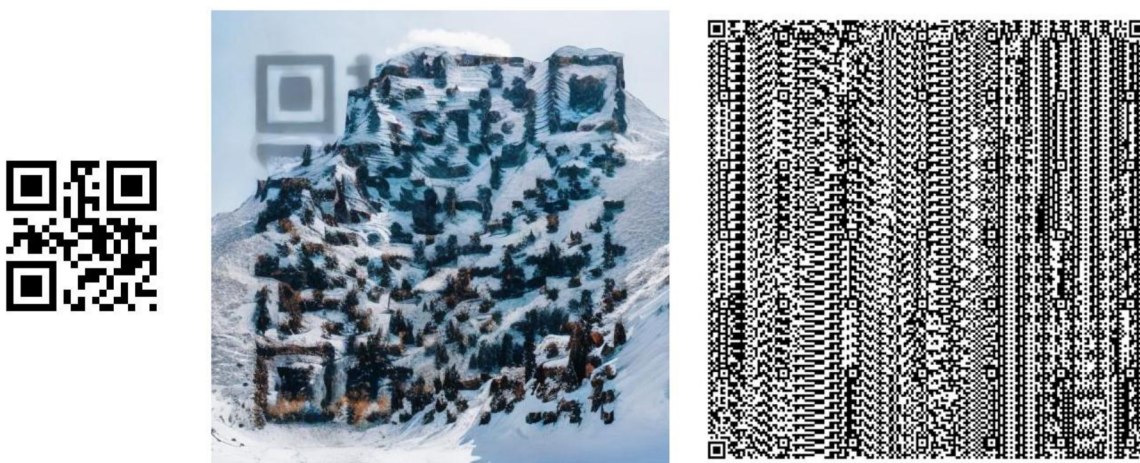
Čárové kódy se začaly v průmyslu objevovat koncem padesátých let dvacátého století. Vlastníkem patentu byla firma RCA, které se do jeho propadnutí v roce 1969 nepodařilo vyvinout provozuschopný komerční systém. Tato skutečnost vedla k tomu, že se po tak dlouhou dobu užitečný prostředek, jako jsou čárové kódy, nemohl rozšířit. Právě kvůli tomu se až začátkem sedmdesátých let o elektronické načítání produktů začala zajímat Národní asociace potravinářských řetězců Spojených států, která si kladla za cíl vytvořit jednotný kód potravinářských produktů. V roce 1971 svaz zaslal zadávací dokumentaci několika firmám, mezi kterými bylo i IBM. To nakonec v roce 1973 vyvinulo dodnes používaný standardizovaný jedenácti místní 1D kód U.P.C. (univerzální produktový kód) [4]. Prvním produktem, který byl s tímto kódem v roce 1974 uveden do prodeje, bylo balení deseti kusů žvýkaček firmy Wrigley s příchutí Juicy Fruit [5]. Na obrázku 2 je vygenerovaný čárový U.P.C. kód, který se vyskytuje na většině zboží v obchodech.



Obrázek 2 - U.P.C. čárový kód

Dle rozdílných potřeb, jako třeba délka kódu, velikost, nebo objem uložených dat vznikaly a nadále vznikají různé druhy čárových kódů. Například v Evropě vznikl známý 1D kód EAN-13 (evropský artiklový kód verze 13), který obsahuje informace o zemi registrace výrobku, kód výrobce, kód výrobku a kontrolní sekvenci pro ověření správnosti načtených dat. Množství informací, které bylo do kódů potřebné vtěsnat se ale zvyšovalo. Proto byla v roce 1994 Japonci představena nová technologie 2D kódů, přesněji QR (rychlá odpověď) kódový systém. QR kód může ve své největší velikosti, tedy verzi 40, kdy čtvercová matice obsahuje 177 řádků i sloupců, pojmut 7 089 čísel nebo 4 296 alfanumerických znaků. Naopak nejmenší možný QR kód má rozměry 21x21 bodů a pojme 41 číslic a 25 alfanumerických znaků [6]. Kromě zvýšené kapacity kódů jsou populární i pro svoji přesnost, i když je vytištěná plocha kódu velice malá, což dovoluje zmenšovat etikety, případně umístit QR kódy i na malá zařízení, jako jsou třeba farmaceutické implantáty. Jedním z benefitů je korekce chyb při načítání kódu, kterou algoritmy ve čtečkách zvládají

do 30 % poškození nebo znečištění skenované plochy kódu. Na obrázku číslo 3 jsou k nahlédnutí tři QR kódy, které po načtení přesměrují uživatele na stejnou webovou stránku. První zleva je QR kód ve verzi 1, který pojme nejvýše 25 alfanumerických znaků. Uprostřed je kód vygenerovaný umělou inteligencí DALL·E 3, kde tmavé části nahrazují stromy nebo mraky. Díky korekci chyb je čitelný většinou dostupných 2D čteček. Posledním kódem je QR kód verze 40, který má v sobě uloženou stejnou URL adresu jako dva předchozí příklady.



Obrázek 3 - Zleva: QR kód verze 1; obrázek generovaný AI (umělá inteligence) obsahující funkční QR kód; QR kód verze 40

Další, v průmyslu a obchodu rozšířenou metodou, je elektronické načítání zboží za využití RFID (identifikace na rádiové frekvenci) technologie. Tato technologie z roku 1973 využívá rádiových vln k pasivní identifikaci označeného objektu. Přesněji RFID systém se skládá z malého rádiového transpondéru – štítku a rádiového přijímače, který je současně také vysílačem. Řešení je k dispozici ve dvou variantách:

- Pasivní štítek a aktivní čtečka – RFID štítek je bez vlastního zdroje napětí. Je aktivován elektromagnetickým impulzem, který ve smyčce vytváří vysílač. Na anténě přijímače je indukováno napětí, pomocí kterého jsou ze štítku vyslána digitální data, obvykle jeho identifikační číslo.
- Aktivní štítek a aktivní čtečka – RFID tag má svoji baterii nebo superkondenzátor, který jej napájí. Díky tomu je dosah komunikace řádově až stovky metrů. Tohoto se využívá zejména při výrobním procesu v továrnách.

Na rozdíl od čárových a QR kódů nemusí být RFID čtečka ve vizuálním kontaktu se štítkem, proto štítky někteří výrobci i prodejci zabudovávají do svých produktů [7]. Příkladem může



být francouzská firma Decathlon, která do oblečení zašívá RFID štítky. U pokladny pak jen stačí vysypat obsah košíku do boxu, který funguje jako lokální vysílač – po jeho obvodu je namotaná cívka a vše uvnitř je zaregistrováno. V paměti štítků se následně aktualizuje údaj o tom, že zboží bylo zapláceno. Další podobný vysílač se nachází u vchodu do prodejny. Pokud se zákazník pokusí vynést nezaplácené zboží, bezpečnostní rám, který obsahuje další cívku, upozorní ostrahu, které rovnou sdělí, jaký produkt zákazník nezaplátil.

RFID tagy umí data nejen vysílat, ale také ukládat ve své interní paměti. Záleží ovšem na typu čipu a kompatibilitě se čtečkou. Například data uložená na platebních kartách jsou šifrovaná a chráněna proti snadnému replikování. RFID tagy mají řadu výhod, ale také zranitelností. Na trhu existují zařízení, jako Flipper Zero, která stojí okolo 200 € a dokáží bez problému emulovat většinu RFID tagů. A to včetně platebních karet fyzických i virtuálních z chytrých telefonů, kdy se telefon pro platební terminál jeví jako platební karta za využití technologie NFC (blízkopolní komunikace). [8]

Pokud by domácnost chtěla využít některou z těchto technologií, musela by pravděpodobně sáhnout po komerčním produktu, který výše zmíněné identifikátory umí využívat. Příkladem komerčního řešení inventarizačního zařízení může být český pokladní systém POHODA od společnosti Stormware. Program podporuje správu skladové agendy tedy inventarizace ve verzích Jazz a vyšších. Cena za licenci pro jeden počítač je 350 € za verzi Jazz. Software také podporuje čtečky čárových kódů, které je nutné zakoupit zvlášť [9]. Kompatibilní je například bezdrátový skener Honeywell Voyager XP 1472g, který se na trhu běžně prodává za cenu 200 € [10]. Nevýhodou tohoto řešení je jeho omezení na operační systémy Windows. Skener navíc neobsahuje displej pro zobrazení základní informací o skenovaném artiklu. Pro správné fungování navíc musí být v blízkém kontaktu s počítačem. Instalace tohoto řešení je náročná a může vyžadovat odborného technika. Proto například společnost NextSoft spol. s r. o. nabízí kompletní řešení, včetně zprovoznění, ovšem za cenu 14 600 € [11].

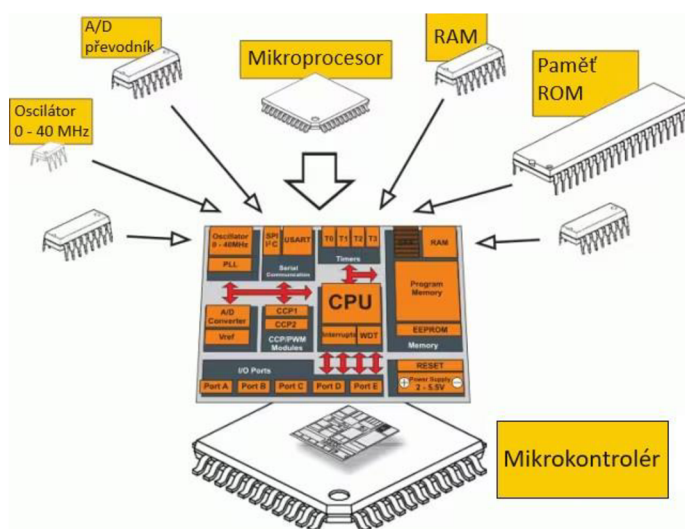
Zařízení nabízená na současném trhu jsou pro rodiny často drahá. Kvůli velké pořizovací ceně inventarizačních zařízení, tak domácnosti běžně využívají již zmíněné nepraktické záznamy o majetku. Levnější řešení inventarizačního zařízení, které by mohla domácnost uvítat, je předmětem této práce. Inventarizační zařízení je možné realizovat za pomoci

snadno a levně dostupných součástek. Nejpodstatnější částí, tvořící většinu elektrických zařízení se stává mikrokontrolér. Mikrokontrolér zajišťuje komunikaci mezi hardwarovou a softwarovou částí přístroje. Proto je nezbytnou částí také pro inventarizační zařízení.

### 3.2 Mikrokontrolér

Mikrokontrolér je malý počítač, který je postaven na jednom čipu integrovaného obvodu, obrázek 4. Obsahuje jedno nebo více procesorových jader, paměť a programovatelné vstupní či výstupní periférie. Periférie jsou určeny k integraci v jednoúčelových (vestavěných) systémech. Mezi takové jednoúčelové systémy se řadí jednoduchá zařízení jako je například mikrovlnná trouba nebo dálkový ovladač. Stejně tak lze mezi jednoúčelové systémy zařadit i složitější řídicí jednotky, které jsou základní částí komplikovanějších strojů. Takovými stroji jsou kupříkladu řídicí jednotky automobilů, herní konzole anebo síťové prvky. [12]

V provozu vestavěných systémů hrají mikrokontroléry klíčovou roli. Poskytují inteligenci a řízení potřebné k provádění úloh, jako je zobrazování informací, příjem vzdálených signálů a ovládání periferních zařízení. Jejich hlavními výhodami oproti klasickým počítačům je nižší cena, nízká energetická náročnost, malá velikost a jednoduchá programovatelnost. Díky tomu jsou rozšířeny i v nositelné elektronice, kde je kladen důraz právě na energetickou náročnost a malé rozměry. [13]



Obrázek 4 - Části mikrokontroléru [14]

Tvorba programové výbavy mikrokontroléru obvykle zahrnuje psaní kódu v jazyce vysoké úrovně, jako je C nebo C++. S využitím těchto programovacích jazyků je vývojář schopen napsat velice efektivní kód, jelikož má k dispozici řadu nástrojů pro přímou správu většiny hardwarových prvků na zařízení. Mikrokontroléry je možné programovat také interpretovanými jazyky, jako je Python či MicroPython [15]. Ale vzhledem k tomu, že se kód překládá až za běhu programu, dochází ke zbytečně vyššímu využití paměti a vyčerpání výkonu zařízení. Ladění takového programu navíc bývá složitější.

K programování mikrokontrolerů je nejběžněji využit jazyk Wiring, který je frameworkem jazyka C++. Jeho největším přínosem je možnost programovat jednotlivé hardwarové části, i bez jejich hlubších znalostí. Autorem Wiringu je Hernando Barragán, který programovací jazyk definoval jako součást své diplomové práce [16]. Pokud je při vývoji potřeba vytvářet komplexní kód a složité algoritmy, vývojář může psát přímo do jazyka Wiring jazykem C++. Jedním z nativních vývojových prostředí pro Wiring je Arduino IDE. To využívá pro kompilaci překladač jazyka C++, přesněji AVR verzi GCC<sup>1</sup>. Minimální kostra programu psaném v jazyce Wiring obsahuje dvě základní funkce, které jsou nezbytné pro úspěšnou kompilaci a běh programu. Tyto funkce jsou *setup()* a *loop()*.

```
// funkce setup() se spustí jednou při spuštění programu a definují se v ní
nastavení prostředí běhu programu
void setup() {
    // inicializace zabudovaného digitálního pinu LED_BUILTIN jako výstup
    pinMode(LED_BUILTIN, OUTPUT);
}
// funkce loop() běží v nekonečné smyčce shora dolů, řádek po řádku, dokud je
deska napájena nebo není restartována
void loop() {
    digitalWrite(LED_BUILTIN, HIGH); // zapnutí LED (HIGH je možné nahradit
logickou 1)
    delay(1000);                     // čekání 1000ms = 1s
    digitalWrite(LED_BUILTIN, LOW);  // vypnutí LED (LOW je možné nahradit logickou
0)
    delay(1000);                     // čekání 1000ms = 1s
}
```

*Úryvek kódu 1 – Základní kód v jazyce Wiring. Jeho nahrání do mikrokontroléru Arduino způsobí blikání zabudované LED diody v intervalu jedné sekundy*

---

<sup>1</sup> GNU Compiler Collection je open-source sada překladačů pro jazyky C a C++

Úryvek kódu 1 je po zkompilování nahrán jako zdrojový kód do FLASH paměti mikrokontroléru. Nahrávání běžně probíhá s využitím externího programátoru, který se k počítači vývojáře připojí pomocí USB a k mikrokontroléru na SPI (sériové periferní rozhraní) piny. Binární kód je následně přehrán do paměti zařízení a je připraven ke spuštění. Některé vývojové sety, kupříkladu zařízení z rodiny Arduino nebo ESP mají programátory přímo integrované na desce. Vývojář jen propojí počítač s mikrokontrolérem a může zkompilovaný program nahrávat do paměti zařízení bez nutnosti využití externího programátoru.

Mikrokontroléry disponují na svých čipech několika druhy pamětí, nejpoužívanější z nich jsou:

- **FLASH** paměť je nevolatilní druh úložiště, tedy úložiště, které je schopno si udržet data i při výpadku napájení. Běžně se využívá k ukládání programového kódu a jiných dlouhodobých dat. Jako dlouhodobá data lze označit konstanty a definice.
- **EEPROM** (Electrically Erasable Programmable Read-Only Memory) je dalším typem nevolatilní paměti, kterou lze elektronicky vymazat a přeprogramovat. Výhodou oproti FLASH paměti je její možnost vyššího počtu přepisů dat, a to až o několik řádů. U čipu ATmega16 se uvádí dvě stě tisíc zápisů. Nevýhodou možnosti vyššího počtu přepisů je snížená rychlost čtení a zápisu dat. Ukázkou dat uložených v EEPROM je hodnota hlasitosti u televizoru nebo zůstatek na platební kartě. [17]
- **SRAM** (statická paměť) je volatilní paměti, tedy při odpojení od napájení dochází ke ztrátě dat. Jde o velice rychlý typ paměti, avšak výrobní náklady na 1 Kb jsou vyšší než u DRAM (asynchronní počítačová paměť). Proto je volatilní paměť SRAM použita v zařízeních běžně nízká. Na rozdíl od DRAM, data uložená v SRAM paměti nemusí být průběžně obnovovaná. Kvůli tomu je statická RAM výkonnější a energeticky méně náročná. SRAM je realizována pomocí několika tranzistorů, zatímco DRAM ukládá data jako elektrický náboj v kondenzátorech [18].

Mezi nejznámější mikrokontrolery lze zařadit ATmega328 od firmy Atmel, který pohání vývojové desky od firmy Arduino, jako je Arduino UNO a NANO. Dále jsou velice populární čipy ESP8266 a ESP32 vyvíjené firmou Espressif Systems, které mají integrovanou podporu WiFi a v případě ESP32 i Bluetooth.

### 3.2.1 Arduino

Arduino je open-source platformou vyvíjenou italskou společností Ivrea Interaction Design Institute. Již od roku 2005 firma navrhuje a konstruuje jednoduché PCB desky, zaměřené na rychlé a snadné vytváření prototypů jak pro pokročilé vývojáře, tak i začátečníky, kteří se s elektronikou teprve seznamují [19]. Jejich nespornou výhodou při prototypování jsou nízké náklady. Během procesu prototypování jsou proto zařízení jako Arduino nenahraditelná. Při prototypování totiž často dochází k nevratnému poškození, ať již mechanickému (přepětí, zkrat, polití nebo pád), tak softwarovému (nenávratně poškozený zavaděč). Vývojových desek je pak vysoká spotřeba. Cenově dostupné vývojové desky jako je Arduino jsou proto populární a celý vývoj zásadně zlevňují a urychlují. Pod pojmem Arduino je široký pojem, pod kterým se skrývá nejen bohatá produktová řada mikrokontrolerů, ale třeba i softwarový nástroj Arduino IDE (integrované vývojové prostředí). To je velice jednoduché na používání a bohaté na funkce a kompatibilitu. [20]. Nejvíce prodávaným Arduinem je Arduino UNO, vyfocené na obrázku 5.



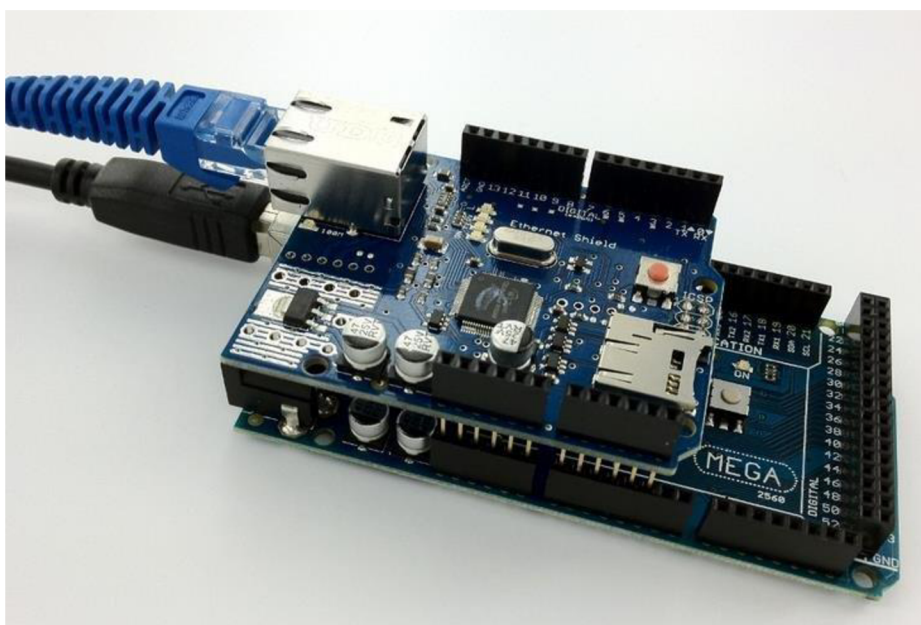
Obrázek 5 - Deska Arduino UNO [21]

Většina z desek Arduino je založena na některém ze série čipů AVR, které jsou vyráběny firmou Atmel [22]. Veškerou funkcionalitu a vlastnosti, kterou Arduina nabízejí, udávají právě zmíněné čipy od Atmelu. Arduino desky až jako celek dovolují využít celkový potenciál čipů co nejefektivněji. Ať už se jedná o snadné nahrávání firmwaru, jednoduché připojování externích komponent nebo široké možnosti napájení. Vše je přehledně zdokumentováno a popsáno širokou komunitou, která okolo této platformy vznikla. Právě i proto jsou Arduina oblíbená také u pokročilejších uživatelů elektroniky. Ti ocení

nejenom rychlost a snadnost používání, ale zejména všechny výhody open-source, jako třeba nepřehledné množství knihoven, které komunita vytvořila a zdokumentovala [23].

Mezi nejznámější Arduina můžeme zařadit například Arduino UNO, MEGA 2560, NANO a MINI. Všechna pohání čipy od firmy Atmel, ovšem liší se ve velikost flash paměti (32 kB až 256 kB), velikosti (nejmenší MINI je veliké asi jako poštovní známka, MEGA zase jako cestovní pas), nebo třeba počtem vstupních a výstupních pinů (20 až 70, z toho několik z nich je PWM (pulzně šířková modulace) kompatibilní a jiné jsou analogové). Všechny, až na poslední MINI, obsahují integrovaný převodník pro snadné nahrávání zdrojového kódu. U MINI je nutné využít převodník externí [24]. Další výhodou je, že Arduina mají zabudované regulátory napětí. Díky tomu je možné Arduino napájet zdrojem napětí v rozmezí 5 V až 12 V, nárazově však převodníky zvládnou až 20 V. Výstupním napětím pak je 3,3 V (převodník LP2985) a 5 V (převodník SPX1117) [25].

Nejprodávanejší desky, jako je UNO a MEGA je možné rozšířit o doplňkové moduly, takzvané shiedly, viz obrázek 6. Jsou to fyzické zásuvné moduly – rozšíření, která se nasadí na jednu z desek a piny ze shiedlu se zapojí do I/O (vstup/výstup) portů na Arduinu. Na trhu jsou stovky rozdílných shiedlů, které Arduinům přidávají nové funkcionality. Například tolik potřebnou bezdrátovou konektivitu, jako je Bluetooth a WiFi nebo displej s joystickem. Tyto standarty bohužel ani UNO, ani MEGA ve svých základních verzích nenabízejí. Dalšími, velmi oblíbenými shiedly jsou rozšíření pro kontrolu servo motorů. [26]



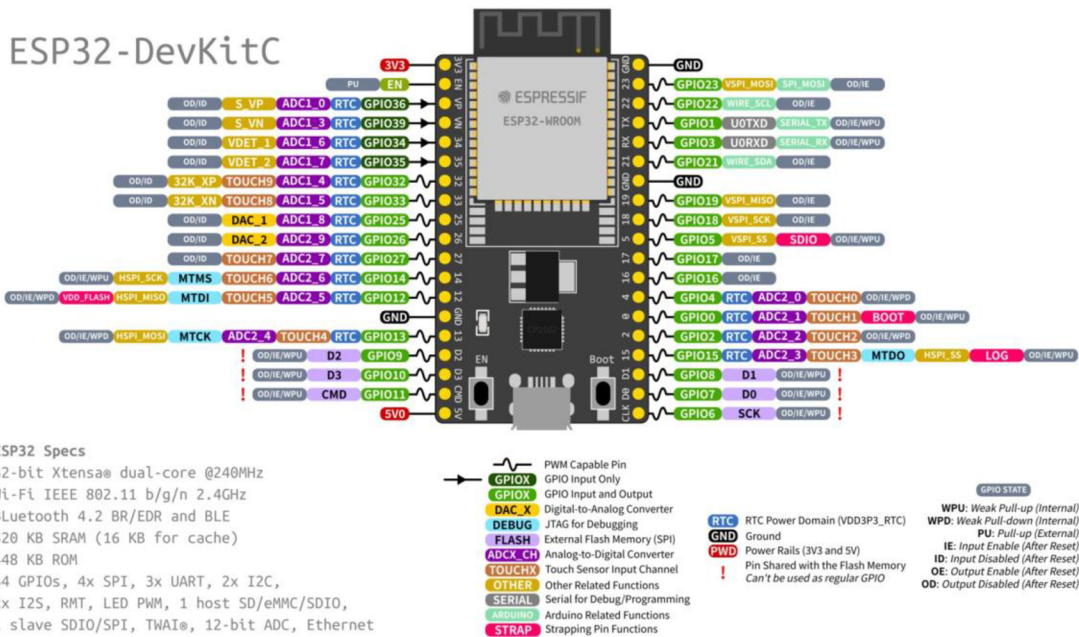
Obrázek 6 - Arduino MEGA s Ethernet shiedlem [27]

### 3.2.2 ESP32

Čínská firma Espressif Systems v roce 2014 uvedla na trh svůj první čip ESP8266. Ten je pro svoji velice nízkou cenu (maximálně do 2 €) natolik populární, že ho na desky integrovalo například i Arduino, a to do verzí UNO a MEGA WiFi. Mikrokontroler má dostatek RAM i flash paměti, že na něm uživatelé běžně provozují lokální webové servery. Novější SoC (systém na čipu) z produktové řady od Espressif Systems je ESP32. Jedná se také o nízkoenergetický čip s nízkou pořizovací cenou. ESP32 je postaveno na jednočipovém mikrokontroléru s integrovanou WiFi 802.11 b/g/n, duálním Bluetooth 4.2, nízkoenergetickým BLE (nízkoenergetické bluetooth) a řadou dalších vlastností. Cena za vývojovou desku se podle jejího určení a přidaných rozšíření pohybuje mezi 2 € až 30 €.

ESP32 čipy je možné pořídit v několika variantách. Jako samostatný čip, který se dá využít při návrhu vlastních PCB nebo ho integrovat do komplexnějších systémů. Dále je dostupný ve formě modulů, které jsou rozšířeny o anténu a nezbytné certifikace pro bezpečný provoz bezdrátových sítí [28]. Nejčastěji se s ESP32 setkáme ve formě vývojových desek, které stačí pouze doplnit o softwarovou výbavu a deska může fungovat. Mají již obsaženou anténu, napájecí obvody a podle cílového zaměření jsou desky osazeny různou kombinací periférií, pamětí a mají rozdílné velikosti [29].

Čip obsahuje dvě 32bitová jádra Xtensa LX6 vyrobená 40 nm technologií, taktovaná až na 240 MHz, a to podle cílového určení a verze. Každé jádro může být individuálně kontrolováno. K dispozici je 520 kB SRAM přímo v čipu pro data a instrukce. Počet vstupně výstupních GPIO (univerzální vstupní/výstupní pin) pinů může být mezi 17 až 36, z toho 16 je jich PWM a externí flash paměť je běžně až 4 MB, v závislosti na druhu desky. SoC dále obsahuje integrovanou hardwarovou kryptografickou akceleraci AES (standard pokročilého šifrování), SHA-2 (podpisový algoritmus verze dvě), RSA (šifra s veřejným klíčem) a RNG (generátor náhodných čísel) [30]. Řešení je natolik výkonné, že je schopné v reálném čase zpracovávat vstupní proud dat, jako je video. Video může na základě předem natrénovaného modelu dále analyzovat a poskytovat údaje, jako třeba počet osob v místnosti, nebo jejich pohlaví. [31]. Na obrázku 7 je zachyceno rozložení pinů ESP32.



Obrázek 7 – Oblíbené ESP32-DevKitC a jeho rozložení pinů [32]

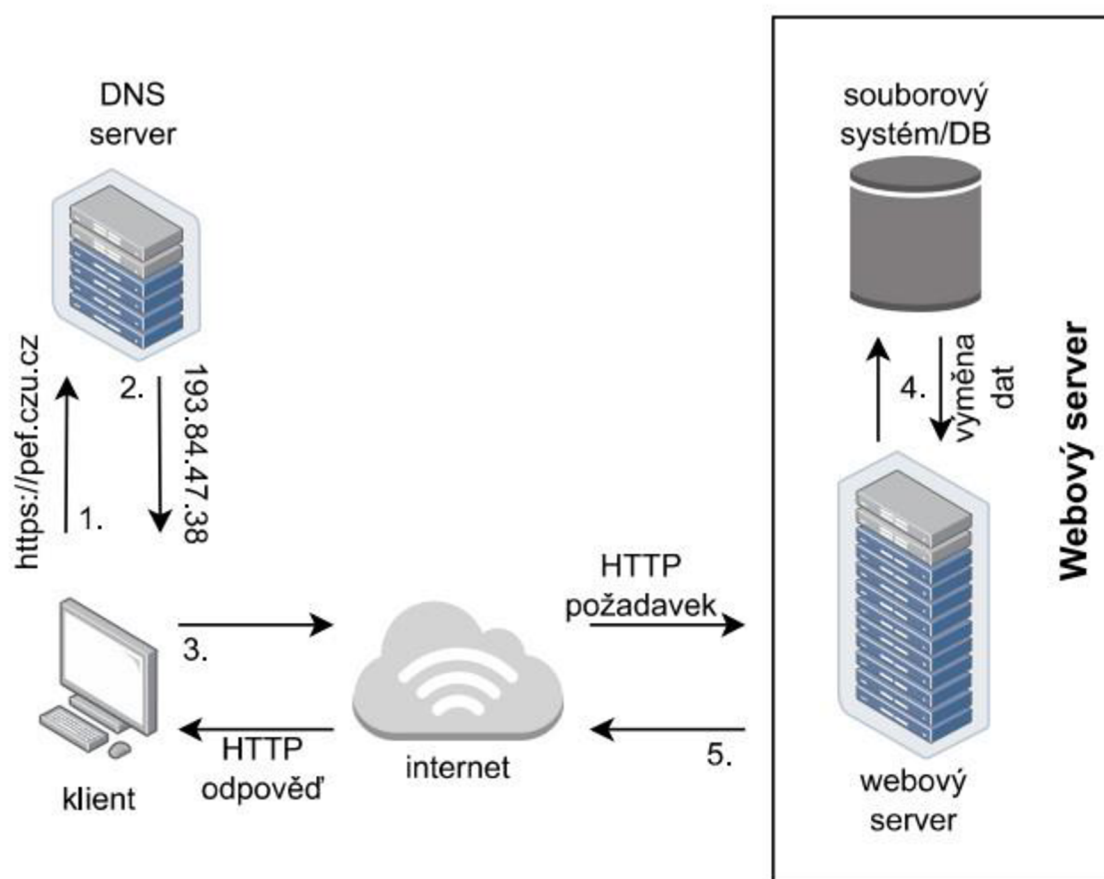
Stejně jako Arduino desky můžeme pro ESP32 vytvářet softwarové vybavení v jazyce Wiring a C++. K vývoji lze využít Arduino IDE, případně robustnější nativní Espressif IoT Development Framework. Aktuální verze Arduino IDE v základu podporuje pouze desky Arduino, ale z repozitáře *Board Manager* můžeme přidat podporu nepřeberného množství dalších čipů, včetně ESP. Oproti jednojádrovým čipům od Atmel je ESP32, jak již bylo zmíněno, dvoujádrový. Proto tedy kód, který běží v základní funkci *loop()* (více v *Úryvek kódu 1*) běží na jádře číslo 1, tedy jádře druhém, jelikož indexace jader začíná na nule. Arduino IDE je vhodné pro jednoduché a rychlé prototypování. U složitějších projektů však dochází ke ztrátě přehlednosti a udržitelnosti kódu.

Pokud tedy chceme vytvářet vysoce optimalizované zabudované systémy, postavené na ESP32 mikrokontroléru, je doporučováno využívat k vývoji *Espressif IoT Development Framework*, který dokáže naplno využít potenciál čipu. Opět se jedná o bezplatný open-source projekt dostupný z platformy GitHub [33]. Tento framework obsahuje základní konfigurační soubory pro ESP, zdrojový kód a k dispozici je také utilita pro aktualizaci firmwaru ESP32. Narozdíl od Arduino IDE, které je naprogramováno v Javě (i to může mít za následek jeho občasnou nižší stabilitu), je nativní aplikace pro vývoj na ESP32 napsána v jazyce C. Framework je vyvíjen pouze pro Unixové operační systémy, ale pomocí emulace můžeme program spustit i na jiných operačních systémech. Z toho je zřejmé, že pokud je cílem vytvořit jednodušší projekt, je výhodnější k vývoji zvolit Arduino IDE [34].



### 3.3 Webový server

Webový server je počítač, který pomocí na něm běžícího specializovaného softwaru reaguje na uživatelské požadavky – requests. Zjednodušeně řečeno se jedná o místo, kde jsou uloženy a zpracovávány webové stránky. Typicky se serverem komunikuje klient, pod kterým si můžeme představit webový prohlížeč. Klientem ale může být i nějaký automatizovaný proces, jako jsou různé crawly<sup>2</sup>. Vzájemná komunikace probíhá pomocí HTTP (hypertextový výměnný protokol) protokolu, respektive HTTPS (šifrované HTTP), kde je komunikace navíc šifrovaná [35]. Na obrázku 8 je popsán proces komunikace s webovým serverem.



Obrázek 8 - Ukázka komunikace mezi klientem a serverem

V momentě, kdy se uživatel pokouší dostat na určitou webovou stránku, nejčastěji zadá URL (jednotný popis umístění zdroje) adresu do adresní řádky webového prohlížeče, ten adresu zpracuje a odešle požadavek. Pokud o dané adrese nemá žádný záznam, nejprve

<sup>2</sup> Crawler je internetový robot, který neustále prochází internet, například provádí indexaci pro internetové prohlížeče.

kontaktuje DNS (server doménových jmen) server. Ten po obdržení URL vrátí klientovi příslušnou IP (internetový protokol) adresu serveru. Následně klient zašle HTTP dotaz na adresu webového serveru, v případě HTTP na port 80, při HTTPS je port 443, na kterých server naslouchá. Webový server zpracuje přijatý požadavek a vygeneruje odpověď, kterou klientovi pošle zpět, například jako HTML (hypertextový značkovací jazyk) kód.

HTTP definuje sadu metod, které určují druh služby, kterou klient očekává že server vykoná. Metody se vždy uvádějí velkými písmeny. Nejpoužívanější metodou je metoda GET, která se posílá na server pro získání dat ze specifikovaného zdroje. Dalšími často používanými metodami je POST, PUT, CONNECT nebo HEAD [36]. Tyto požadavky mají návratové kódy, které klientovi říkají, s jakým výsledkem byl jeho požadavek zpracován. Dle zdroje [37] jsou návratovými kódy:

- **100–199** informují o tom, že požadavek byl přijat a je zpracováván,
- **200-299** klienta vyrozumí o úspěšné operaci (*200 OK; 201 Created; 202 Accepted; ...*),
- **300-399** vyžadují další interakci od klienta pro dokončení požadavku,
- **400-499** říkají, že klient udělal chybu a požadavek nemůže být uspokojen. Nejznámější chybovou hláškou je „*404 Not Found*“,
- **500-599** ukazují chybu na straně serveru.

### 3.3.1 Provozní náklady webserveru

K provozu webového serveru je zapotřebí počítač. O jeho výkonu rozhoduje několik faktorů, mezi kterými je celkový počet dotazů za časovou jednotku, nebo počet a složitost vykonávaných operací. U řešení typu *Smart Home*, kde server zpracovává menší množství převážně lokálních dat z několika IoT senzorů není třeba vysokého výpočetního výkonu a s přehledem postačí mikroprocesor Raspberry Pi (RPi). Na RPi poběží Linuxová distribuce s nainstalovaným webovým serverem, například populárním Apache [38]. Takové řešení je energeticky nenáročné, pořizovací náklady jsou minimální a provoz není náročný na údržbu. Předpokládaná cena za pořízení je okolo 200 € a náklady na provoz jsou maximálně 10 € ročně. Avšak v momentě, kdy by měl server obsluhovat vysoké počty dotazů najednou a vykonávat složité výpočty, bylo by potřeba vybrat řešení s vyšším výpočetním výkonem.

Provozovat vlastní server s sebou nese rizika, jako jsou výpadky proudu, poruchy hardwaru, přehřívání v letních měsících, ale i nepříjemnosti jako je hluk, složitá údržba a zastarávání hardwaru i softwaru. Už i pro malé projekty se tedy často vyplatí zakoupit si místo na serveru u nějaké z firem, které se pronájmem výpočetního výkonu zabývají. Nejlevnější a nejjednodušší variantou současně jsou webhostingy, kde poskytovatel už připravil všechny potřebné technologie a uživatel je bez složitého nastavování může rovnou využívat. Zde ale spočívá i jisté omezení, a to takové, že uživatel nemá možnost si technologie vybrat sám, má jen minimální možnosti konfigurace serveru a jeho data jsou uložena v prostoru spolu s daty jiných uživatelů. Pořízení webhostingu je tedy ideální pro pohodlnější, nebo méně zkušené uživatele, kteří se nechtějí starat o vlastní hardware, ani software, ale zároveň chtějí spolehlivou službu za přijatelné peníze. Ceny se pohybují v závislosti na výpočetním výkonu mezi 2 € až 10 € měsíčně, bez dalších nutných investic.

Dále má uživatel možnost si u firmy pronajmout dedikovaný nebo VPS server (virtuální privátní server). U virtuálního serveru si uživatel platí za určitý výpočetní výkon, ale server je pouze virtuální a běží na sdíleném hardwaru s jinými uživateli. U dedikovaného serveru se platí přímo za daný kus hardwaru, který využívá jen ten uživatel, který si jej platí. Smyslem takového řešení je poskytnout uživateli plnou kontrolu nad serverem [39]. Uživatel se sám stará o všechny součásti, od operačního systému až po aktualizace jednotlivých

aplikací. Na rozdíl od vlastního serveru jsou tato řešení provozována v objektech specializovaných firem a zákazník k nim přistupuje pouze vzdáleně. Pro uživatele tím odpadá starost o zabezpečení dodávek elektrické energie, údržbu komponent a o zabezpečení před fyzickým zneužitím. Firma garantuje téměř stoprocentní dostupnost přes vysokorychlostní internetové připojení, zabezpečení sítě proti DDoS (odepření služby)<sup>3</sup>, ale i proti fyzickému vniknutí narušitelů do objektu [40]. V následující tabulce je srovnání těchto dvou možností, spolu s provozováním vlastního serveru. U všech tří variant je nutná vyšší technická znalost pro obsluhu.

Tabulka 1 - Srovnání tří variant provozu webového serveru [40], [41], [42]

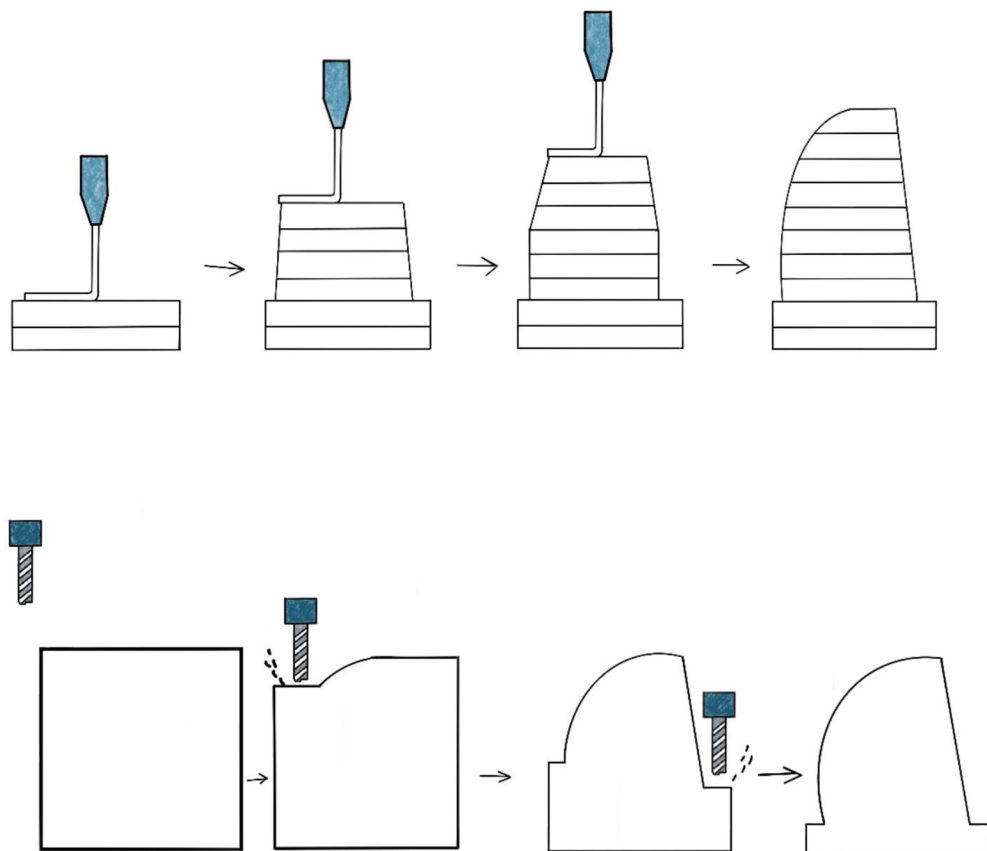
	Dedikovaný server	Vlastní server	VPS
Pořizovací náklady (€)	0	500 a více (násobně)	0
Měsíční náklady (€)	40–1300 (v závislosti na konfiguraci serveru)	4–200 (odvíjí se od spotřeby a ceny za 1 kWh)	37–200 (v závislosti na konfiguraci)
Údržba HW	Žádná	Nutnost pravidelných kontrol, testů a oprav	Žádná
Údržba SW	Náročná	Náročná; nutnost aktualizovat ovladače pro HW	Náročná
Hlavní výhody	Spolehlivost; bezpečnost; dostupnost;	Modifikovatelnost; opravdové „vlastnění dat“	Spolehlivost; dostupnost; cena
Hlavní nevýhody	Cena	Údržba; vysoké pořizovací náklady; riziko výpadku a nedostupnosti	Sdílený HW

<sup>3</sup> DDoS (Denial of service) je druh útoku, kdy je cílový počítač napadán vysokým množstvím dotazů, dokud nedojde k jeho zahlcení a nucenému odpojení.

### 3.4 Aditivní výroba

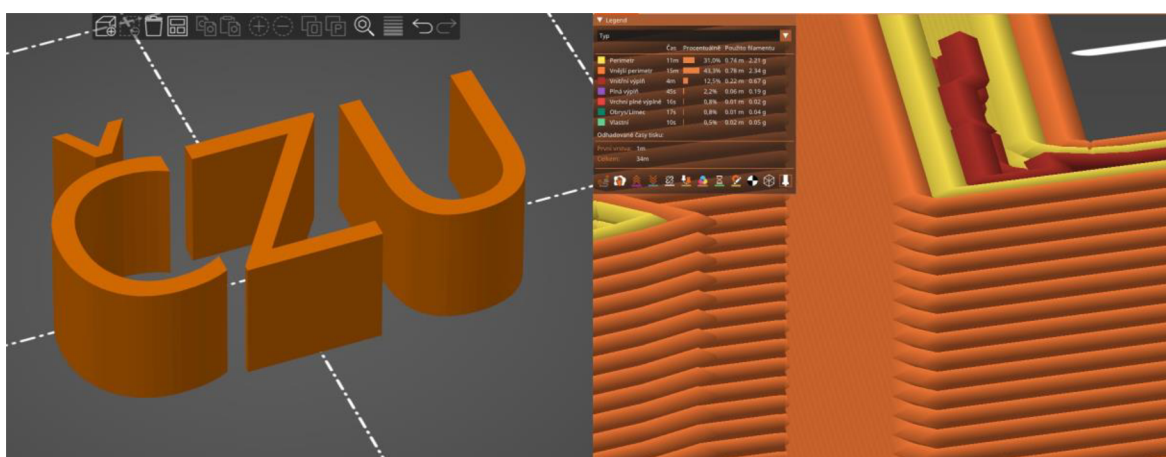
Aditivní výroba (AM) je rychle se rozvíjející technologie, která umožňuje vytváření komplexních objektů vrstvu po vrstvě. Největší zásluhy za rozmach AM si připisuje výrobní průmysl, který AM využívá především díky rychlému a cenově příznivému prototypování. V posledních dvou dekadách se AM velice rozšířila i mezi širokou veřejností. AM tak přináší zjednodušení práce například kutilům nebo umělcům. Materiálů, ze kterých lze tisknout je nepřehledné množství, nejčastěji jsou používány různé druhy polymerů, jako například ABS vyráběné z ropy nebo PLA ze škrobu. Lze tisknout i kovy, biomateriály nebo třeba beton. [43], [44]

Ještě před masivním rozšířením aditivní výroby, potažmo 3D tisku, se například průmyslové prototypy vytvářely subtraktivní výrobou (SM). To je takový způsob opracovávání surového materiálu, kdy výsledný objekt vzniká jeho postupným odebíráním, například sochař vytvářející sochu, nebo CNC (číslicové řízení pomocí počítače) stroj obrábějící kov [45]. Obrázek 9 zjednodušeně zobrazuje proces AM a SM.



Obrázek 9 - Srovnání aditivní (nahore) a subtraktivní (dole) metody

Oba výrobní postupy mají ale jedno společné, a to že jako předlohu využívají počítačový grafický model CAD. Při tisku je CAD model „rozřezán“ pomocí softwarového nástroje slicer. Program grafický 3D model převede ze souboru STL (počítačový formát pro stereolitografické CAD softwary), do GCOdu – celý model je rozkouskován na jednotlivé dvourozměrné vrstvy, včetně pokynů pro tiskovou hlavu, odkud kam se má při vytlačování materiálu pohybovat. Právě při slicování modelu se dá specifikovat úroveň detailu výsledného produktu, například pomocí výšky vrstvy, počtu perimetrů nebo rychlosti přejezdů tiskové hlavy. Na obrázku 10 je vidět model textu. Vlevo je celý STL model nápisu ČZU o výšce 1,5 cm. Napravo je vyobrazena jeho slicovaná verze, s nastavením tiskové vrstvy 0,2 mm. Jednotlivé vrstvy lze rozlišit a jejich počet je celkem 75. Počet perimetrů udává, kolik má stěna tisknutého objektu vrstev. Pokud je objekt větší a tloušťka stěny menší, je třeba dutý prostor objektu vyplnit, aby při tisku nedošlo k jeho zhroucení dovnitř. [46]



Obrázek 10 - Náhled STL souboru (vlevo) a GCOdu (vpravo) připraveného k tisku

### 3.4.1 Tvorba modelu

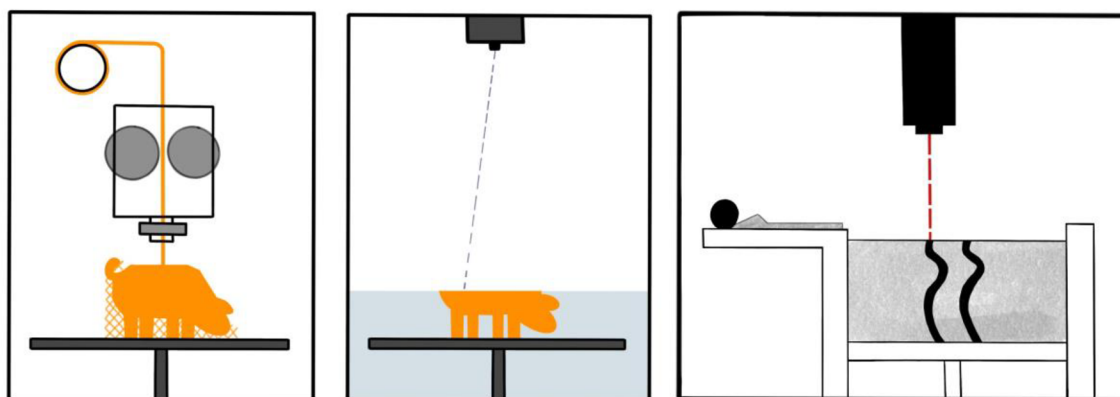
Uživatel si dnes může vybírat z mnoha nástrojů, jak model pro 3D tisk vytvořit. Nejběžnější postup je s využitím jednoho z dostupných softwarových řešení pro tvorbu CAD modelů, jako je například AutoCAD od firmy Autodesk, Fusion 360 nebo jednoduchý Tinkercad, opět od firmy Autodesk. Všechny ale mají jedno společné, a to grafické rozhraní, ve kterém je uživatel schopen pomocí myši, stylusu nebo textových příkazů vytvářet digitální vícerozměrné modely hmotných objektů [47]. Výstupem takové práce může být již zmíněný model STL.

### 3.4.2 3D tisk

Historie 3D tisku začíná už ve druhé polovině 20. století. V roce 1981 si japonský vědec Hideo Kodama nechal patentovat přístroj, který pomocí paprsku UV (ultrafialového) světla vytvářel na podložce ponořené v termocitlivém polymeru pevné fyzické vrstvy a jejich vrstvením vznikal výsledný výtisk. Výzkum byl ale značně podfinancován a celý projekt tak zanikl. V roce 1988 přišla na trh první 3D SLA (stereolitografický aparát) tiskárna, kterou na základě svého patentu z roku 1986 vyráběl Charles Hull. Jeho cílem byla možnost jednoduše vytvářet vlastní malé díly pro jeho nábytkářský business [48]. Velice úspěšná ve světě 3D tisku je firma Prusa Research, kterou založil Josef Průša. Ten je spoluautorem projektu RepRap, což je open-source hardwarový komunitní návrh tiskárny. Návrh spočívá v tom, že jednotlivé díly modelů jsou volně dostupné a každý kdo má k dispozici 3D tiskárnu si je může replikovat zdarma. Jeho firma, stejně jako konkurence, vyrábí převážně tiskárny typu FDM (výroba taveného vlákna). V následující tabulce je srovnání komerčně nejrozšířenějších technologií ve 3D tiskárnách. V tabulce 2 a na obrázku 11 je srovnání častých druhů AM tisků.

Tabulka 2 - Základní srovnání nejběžnějších technologií 3D tiskáren [49]

	FDM	SLA	SLS
Rychlost	Rychlá	Pomalá	Pomalá
Přesnost	Střední	Vysoká	Vysoká
Náročnost	Nízká	Střední	Vysoká
Cena (€)	100–2 000	1 000 (a více)	20 000 (a více)
Odolnost	Střední	Nízká	Vysoká
Popis	Tiskový materiál je filament navinutý na špulce. Extruderem je vtlačován do zahřáté trysky, která jej v kapalném stavu vrství na podložku.	UV paprsek z laseru ozařuje podložku, na které pryskyřice vytvrdne. Povytažení podložky dochází k vrstvení vytvrzené pryskyřice.	Laser rozpéká trnkou vrstvu prášku. Po vychladnutí je nanasena a vyrovnána přesně vysoká další vrstva a proces se opakuje.
Materiál	Polymery	UV citlivá pryskyřice	Plasty, kovy, sklo,...
Nevýhody	Vyžaduje tiskové podpěry a leptání materiálu	Výtisk se musí chemicky omývat, aby se zbavil pryskyřice. Vyžaduje podpěry.	Výsledný produkt se musí leštit kvůli vysoké zrnitosti povrchu.



Obrázek 11 - Technologie 3D tisku, zleva: FDM, SLA, SLS

### 3.4.3 Tiskové filamenty

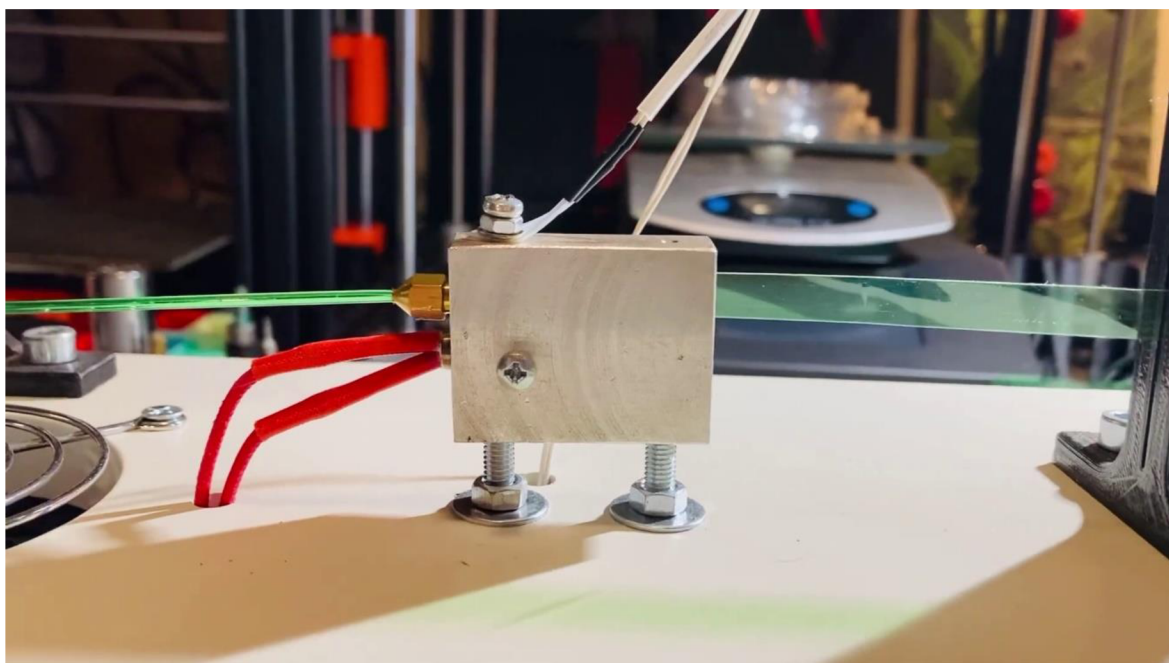
Mezi rekreačními tiskaři je nejvíce rozšířený a populární FDM tisk. Nejen díky modulární konstrukci tiskáren, levnému a rychlému tisku, ale především i kvůli širokému výběru tiskových materiálů, které se označují jako filamenty. Tiskový filament je termoplastická surovina pro 3D tiskárny, dodávaná nejčastěji jako tenká šňůra namotaná na špulce, nejčastěji o hmotnosti jednoho kilogramu. Průměr filamentu je nejběžněji 1,75 mm [50]. S rostoucí oblibou 3D tisku roste i nabídka filamentů u běžných prodejců, ti nabízejí stovky druhů filamentů, s cenou přibližně okolo 30 €.

Filamenty se od sebe rozlišují barvou, konzistencí, průměrem, ale především materiálem, ze kterého je filament vyroben. Nejpopulárnější je bezesporu PLA. Jedná se o organický polymer – kyselinu polymléčnou. Je tak jediným běžně prodávaným filamentem, který je rozložitelný [51]. Což je současně i nevýhodou materiálu. Na slunci rychle degraduje, ztrácí barvu a při vyšších letních teplotách se může deformovat. Jinak je tisk s PLA nejsnazší ze všech materiálů, a proto se hodí na rychlé prototypování, nebo díly určené k vnitřnímu používání.

Tisk s filamentem PETG (polymer Polyethylene Terephthalate modifikovaný Glykolem) je také poměrně jednoduchý, jen oproti PLA vyžaduje vyšší tiskové teploty. Je vhodnější do namáhaných dílů a nevdá mu tolik expozice povětrnostním podmínkám. PETG je podobný polymer, jako PET (Polyethylentereftalát), ze kterého se vyrábějí obaly



na nápoje. Je tedy celkem běžné, že si kutilové stavějí vlastní výrobní linky na PETG filament, které jsou konstrukčně velice jednoduché. Stačí rovnoměrně rozřezat PET lahve na dlouhé špagáty, které jsou vsouvány do rozehřáté bronzové trysky – hotendu, o průměru 1,75 mm. Za tryskou je pak cívka, navíjející vzniklý filament na prázdnou špulku od filamentu. Výroba je zachycena na obrázku 12. Veškeré komponenty na takovéto řešení stojí do 100 € [52]. Výsledná tloušťka filamentu je ovlivněna rychlostí navíjení na špulku. Úhlová rychlost se musí tedy regulovat, protože se obvodová rychlost s přibývajícím materiálem zvyšuje. Alternativním řešením je měřit aktuální tloušťku na výstupu z trysky laserovým měřidlem a v reálném čase měnit rychlost otáčení.



*Obrázek 12 - Ukázka procesu výroby vlastního filamentu recyklací PET lahví [52]*

Mezi další značně využívané filaments můžeme zařadit například ABS (Acrylonitrile Butadiene Styrene). Ten je velice pevný a teplotně odolný, proto se hodí i na extrémně fyzicky namáhané díly. Například stavebnice LEGO je vyráběna z tohoto materiálu, ale výroba probíhá injektáží do forem, nikoli FDM tiskem. Jeho výhodou je, že je rozpustný v acetonu, čehož se využívá k vyhlazení povrchu po tisku. Třeba PLA lze vyhladit pouze chloroformem, který není snadno dostupný. Výtisky, které budou celoročně vystavené slunečním paprskům, například meteostanice, je vhodné tisknout materiálem ASA (Acrylic Styrene Acrylonitrile). Tu je ale nutné tisknout v uzavřeném boxu a dobře větrané místnosti, protože silně zapáchá [53]. Pro ohebné díly, jako jsou razítka, pružné panty nebo kolečka

u modelů se využívají materiály obecně nazývané FLEXI. V závislosti na jejich pružnosti a výrobci jsou FLEXI filameny vyráběny z různých materiálů, které jsou často obchodním tajemstvím [54]. Většina filamentů je hygroskopických a snadno tedy pohlcují vzdušnou vlhkost. Je tedy žádoucí je skladovat na suchém místě, nebo je před tiskem vysušit v troubě, teplotou v rozmezí 45 °C až 95 °C, podle sušeného materiálu. Vlhkost ve filamentu způsobuje, že při průchodu hotendem je prudce odpařena a vznikají bublinky, které způsobují nekonzistentní tisk a vizuální vady. V následující tabulce 3 jsou zachyceny základní fyzikální vlastnosti nejběžněji používaných filamentů.

*Tabulka 3 - Komparace základních vlastností běžně dostupných filamentů. U sloupců označených „\*“ je rozsah 1=minimální až 10=maximální. [55]*

Materiál	PLA	PETG	ABS	ASA	FLEXI
Tiskový box	×	×	✓	✓	×
Teplota trysky (°C)	190–235	215–270	230–255	220–275	220–260
Teplota podložky (°C)	50–60	70–90	95–110	90–110	40–85
Teplotní odolnost *	3	5	4	5	3
Odolnost v tahu *	8	8	6	6	1
Pružnost *	2	5	4	5	10
Cena	30	30	20	30	40

## 4 Vlastní práce

V této části práce je popsán kompletní proces návrhu mobilního zařízení určeného pro usnadnění inventarizace. Dále je navržena a vytvořena webová aplikace, se kterou inventarizační zařízení komunikuje. Ta zároveň uživateli nabízí pohodlnou správu a přehled nad svými daty.

### 4.1 Návrh zařízení

Samotnému návrhu předcházela soupis požadavků na funkce a vlastnosti inventarizačního zařízení. Současně byl od samého začátku kladen důraz na jednoduchost, intuitivnost a rychlost používání. Dále byly vybrány vhodné komponenty tak, aby odpovídaly požadavkům. Finální grafický návrh následoval až po propojení jednotlivých hardwarových částí řešení. Nakonec bylo s využitím FDM 3D tiskárny vytisknuto ochranné pouzdro.

#### 4.1.1 Funkční požadavky

Z uživatelského hlediska bylo žádoucí, aby zařízení bylo snadno ovladatelné, kompaktní a skladné. Při modelování ochranné krabičky byla tedy snaha vytvořit zařízení co nejlehčí a nejmenší, avšak za podmínky využití všech zvolených komponent.

Seznam základních inventarizačních funkcí, které byly požadovány v inventarizačním zařízení a jejich výčet si stanovil autor práce sám na základě svých potřeb je následující:

- **Zalistování nového záznamu do databáze** – po načtení zatím nevidovaného výrobku je uživateli nabídnuta možnost tento produkt zařadit do zvolené tabulky.
- **Vyhledání požadovaného záznamu** – po zadání nezbytných identifikačních údajů je uživateli vrácen popis produktu spolu se všemi údaji, které o něm v evidenci má.
  - Fulltextové vyhledávání ve webové aplikaci, kde má uživatel možnost vyhledávat a filtrovat na základě několika parametrů.
  - Dohledání záznamu na základě načtení 1D nebo 2D kódu.
  - Možnost vyhledat produkt po načtení jeho RFID tagu.
- **Aktualizace existujícího záznamu** – možnost upravit doplňující údaje vedené o vybraném produktu, včetně změny kusů skladem (naskladnit/vyskladnit).
- **Export všech záznamů** – funkce, která uživateli dovolí jednoduše přenášet svá data.

#### 4.1.2 Volba komponent

Na základě požadavků byly vybrány vhodné komponenty. Při jejich výběru byl brán zřetel na nabízené funkce, existenci dokumentace od výrobce, ale také na pořizovací náklady. Zařízení budou využívat domácnosti, proto se hledal ideální poměr mezi kvalitou a cenou u dostupných produktů.

Jako řídicí jednotka bylo zvoleno ESP32, a to ve své verzi ESP32 WROOM-32 modul s 38 piny. Po důkladné rešerši bylo zjištěno, že ESP32 je nejvhodnějším mikrokontrolérem pro tento projekt. Výhodou oproti jiným mikrokontrolerům je, že ESP32 již od výrobce podporuje bezdrátové aktualizace (OTA). V návrhu se totiž nepočítá s výstupem integrovaného USB, což by kvůli obtížnému přístupu k portu na EPS32 velmi ztížilo budoucí aktualizace firmwaru zařízení. S využitím OTA může být aktuální verze softwaru distribuovaná na dálku, bez nutnosti zapojení uživatele [56]. Další parametry tohoto modulu jsou [57]:

- Pořizovací cena se pohybuje podle kvality výroby mezi 2 až 8 € za kus.
- Rozměry jsou 55x28 mm (s anténou a osmnácti piny na každé straně.)
- SPI Flash paměť o velikosti 4 MB a SRAM 520 kB.
- WiFi komunikující v pásmu 2,4 GHz, Bluetooth 4.2 s podporou BLE a anténou o impedanci 50 Ω.
- Dvoujádrový procesor Tensilica LX6 s frekvencí nastavitelnou v rozsahu 80 MHz až 240 MHz. V režimu spánku je spotřeba pouze 5 μA.
- Z celkových 38 pinů je jich 26 digitálně vstupně výstupních, 16 podporuje PWM a dalších 18 pinů je analogových.
- Integrovaný řadič CP2102, což je převodník z USB, které je na desce integrované ve verzi Micro USB na UART (univerzální asynchronní přijímač a vysílač). Ten převádí data z USB na sériovou linku bez nutnosti použití převodníku externího [58].
- Doporučené provozní podmínky jsou -40 až 85 °C, ovšem výrobce zařízení certifikoval až do teplot 105 °C, ve kterých je schopné fungovat více než 24 hodin.
- Integrovaný regulátor napětí AMS1117 3.3 stabilizuje výstupní napětí na 3.3 V. Desku lze napájet prostřednictvím USB nebo externím zdrojem napájení přes konektor VCC/5V na obrázku 13 vpravo dole. Samotný regulátor je na tomto obrázku umístěn nad pinem SD2 u tlačítka RST.

O napájení se stará dvojice lithium-iontových akumulátorů typu 18650 zapojených v sérii. Nominální napětí jednoho článku činí 3,7 V a jmenovitá kapacita je 2600 mAh. V obchodech je řada falsifikátů, které udávají vyšší kapacity, než reálně mají. Na trhu je jen hrstka výrobců, kteří mají dostatečné know-how aby byli schopni vyrobit články s kapacitou vyšší než 5000 mAh. Do tohoto projektu byly zvoleny kvalitní články od korejského výrobce Samsung. Cena jednoho akumulátoru činila 8 €. Výhodou Li-Ion článků je jejich vysoká životnost, která se pohybuje mezi 600 až 1200 nabíjecími cykly a minimální samovybití, pokud je akumulátor nepoužíván. Tento typ akumulátorů, vyobrazených na obrázku 13, vyrábí a používá automobilka Tesla, k ukládání energie ve svých vozech.

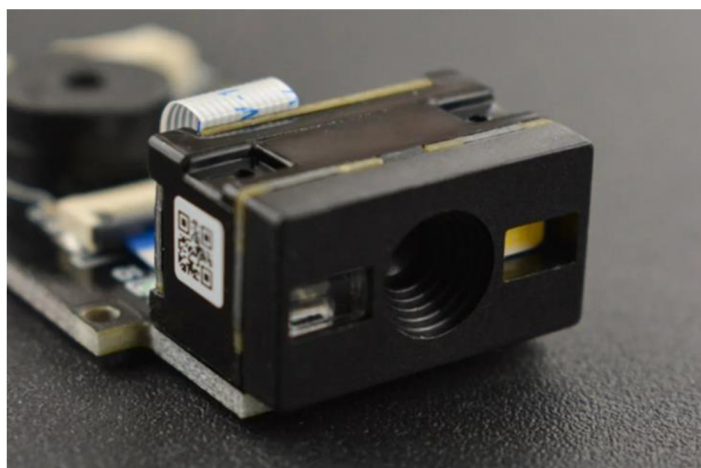


Obrázek 13 - Akumulátor Samsung 18650 s kapacitou 2600 mAh [59]

Rizikem je u Li-Ion jejich podbití, tedy stav, kdy napětí na článku klesne pod 2,5 V [60]. Z toho důvodu je v obvodu spolu s bateriemi zařazen i balancér s ochranným obvodem. Balancér se stará o rovnoměrné nabíjení obou článků. Ochranný obvod zase chrání baterie před přebíjením, podbíjením, zkratu a přetížením. Do řešení byl zvolen model HX-2S-01 a jeho cena je přibližně 1 €. Nabíjení je vyřešeno připojením stejnosměrného jack konektoru, o průměru 2,1 mm a hloubce 5,1 mm na konektory balancéru. Zvažován byl ještě konektor typu USB C, ale ten by vyžadoval přidání step-up modulu, případně PD (rychlý dobíjecí standard) modulu, který by dokázal vytvořit požadovaných 9 V. Napájecí síťový adaptér má výstupní napětí 9 V a proud 1 A. K bateriím byl navíc zařazen ukazatel stavu napětí. Jedná se o jednoduchý voltmetr, který při naměření 8 V rozsvítí všechny čtyři led diody. S klesajícím napětím postupně zhasíná diody. Spodní hranice měření je 5 V, kdy indikuje úplné vybití soustavy a jeho cena je 2 €.

Aby bylo možné elektronicky číst čárové a QR kódy, bylo potřeba vybrat vhodný skener. Ideální poměr mezi cenou a požadovanými vlastnostmi je skener CM65. Ten je schopný číst většinu běžně dostupných 1D kódů. Při čtení 2D kódů podporuje QR kódy, PDF417 a Data

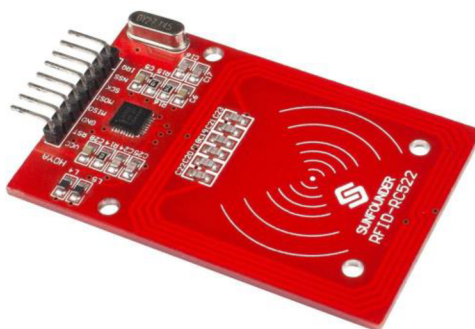
Matrix kódy. Komunikovat s tímto modulem lze přes USB nebo UART. V této práci byla využita sériová linka. Čtečka má vlastní LED přísvit, takže je možné skenovat kódy i po tmě. Ideální vzdálenost mezi čtečkou a skenovaným kódem je mezi 4 až 25 cm a náklon může být  $\pm 65^\circ$ . Modul nabízí široké možnosti nastavení, které se provádí načtení QR kódů z dokumentace od výrobce. Volit lze mezi USB nebo UART, nastavovat přenosovou rychlost v baudech, nebo také režim skenování. Čtečka má tři volby režimu: nepřetržitě skenování (to je energeticky více náročné, proudový odběr je až 120 mA), softwarově aktivované skenování (skener je aktivován po přijetí startovacího příkazu, například `scanner.start()` u sériové linky) nebo mechanická aktivace skeneru. Mechanická aktivace se provádí stisknutím membránového vypínače nacházejícího se na těle modulu [61]. Na modulu je integrovaný piezo bzučák, který uživateli poskytuje zvukovou odezvu o stavu skenování. Nastavení úrovně hlasitosti, tónu a melodie lze opět provést využitím QR kódů dodaných výrobcem. Cena této čtečky se pohybuje mezi 40 až 80 €. Výrazně ušetřit lze zakoupením ručního skener již jako hotového produktu, včetně rukojeti a spínače. Jeho demontáží můžeme získat modul CM65 za necelých 20 €. Na obrázku 14 je vyobrazený skener CM65 z čelní strany, kde jsou umístěny přísvětlovací LED diody a snímač.



Obrázek 14 - Čtečka 1D a 2D kódů CM65 [61]

Pro bezdrátové načítání karet byl zvolen modul RFID-RC522 (obrázek 15) s již zabudovanou anténou. Jedná se o komunikační zařízení, které bezkontaktně čte a zapisuje data na karty a přívěšky typu MIFARE. Komunikace s řídicí jednotkou probíhá skrze SPI sběrnici. K napájení je zapotřebí zdroj s 3,3 V. Čtečka se automaticky uvádí do úsporného režimu, při kterém má proudový odběr maximálně 13 mA.

Při její aktivaci a čtení karty je odběr 26 mA. Modul je schopný detekovat chyby čtení a podporuje šifrovací algoritmy [62]. Pořizovací náklady jsou přibližně 4 €.



Obrázek 15 - RFID čtečka RC522 [63]

Pro zobrazení údajů byl zvolen 1,3 palcový OLED displej s rozlišením 128 px na 64 px (obrázek 16). Jedná se o malý, 2,4 cm vysoký a 3,3 cm široký monochromatický display. Počet zobrazovacích bodů je naprosto dostačující pro zobrazení komplexnějších znaků a grafických návrhů. Komunikace probíhá přes I<sup>2</sup>C sběrnici (multi-masterová počítačová sériová sběrnice). Napájení může být v rozmezí 2,8 V až 5,5 V. Technologie OLED (organická světlo emitující dioda) je energeticky velice úsporná, jelikož jsou napájeny pouze ty pixely, které v danou chvíli zobrazují. Kromě energetických benefitů jsou navíc výborně čitelné, a to i na přímém slunečním svitu. Dostupný je za cenu 6 €.



Obrázek 16 - 1,3 palcový OLED display SH1106 [64]

Výstupní napětí z ESP32 WROOM-32 je tedy pouze 3,3 V a výstupní napětí akumulátorů se pohybuje mezi 5–8 V. Součástí řešení jsou ale spotřebiče, které potřebují přesně 5V zdroj napětí, například čtečka čárových kódů nebo RFID čtečka. Proto je v obvodu, paralelně k baterii, připojen regulátor napětí AMS1117 5.0 (obrázek 17). Je téměř identický, jako

ten zabudovaný na ESP32, ale stabilizuje napětí na vyšší hladině. Jeho pořizovací náklady jsou v jednotkách centů.



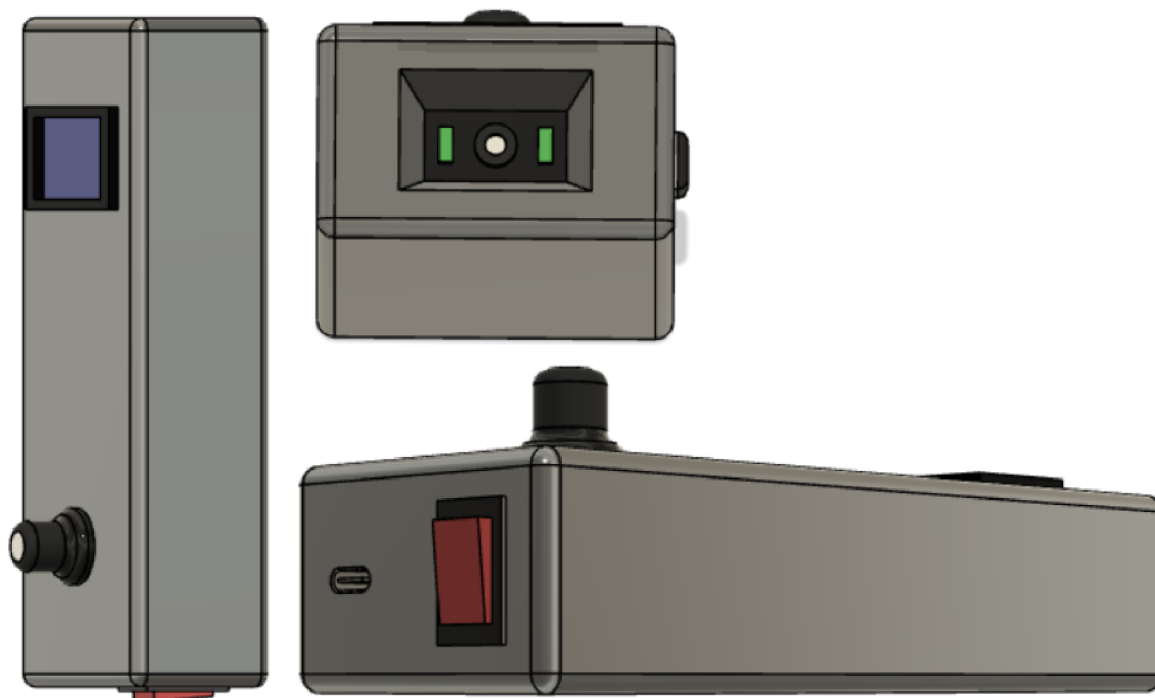
*Obrázek 17 - Stabilizátor AMS1117 5.0 [65]*

Celkové náklady na všechny součásti, bez započtení nákladů na 3D tisk a spojovacích materiálů činily 55 €. Nejnákladnější položkou byla čtečka čárových kódů GM65 a dva akumulátory typu 18650.



### 4.1.3 Design modelu

V momentě, kdy byl seznam všech komponent, včetně jejich datových listů kompletní, byl navržen výsledný design inventarizačního zařízení, jelikož byly přibližně známy jeho parametry. Pro ucelení představy bylo třeba zrealizovat návrh výsledného designu. K návrhu designu 3D modelu byla vybrána aplikace Fusion 360 od firmy Autodesk. Design 3D modelu byl navržen tak, aby zajistil ochranu hardwarové části před mechanickým poškozením. Navrhované pouzdro má na přední straně OLED display spolu s ovladačem, který umožňuje řízení uživatelského rozhraní. Levý bok navrženého pouzdra je opatřen spouštěcím tlačítkem. Spodní strana pouzdra obsahuje otvor pro připojení napájecího konektoru USB C a vypínače. Podstatnou část tvoří čelo navržené krabičky, kde se nachází zkosený otvor pro čtečku. Aby byl návrh co nejvíce autentický byly zvoleny i jednotlivé materiály. Pro display byla tedy zvolena imitace OLED displeje. Ovladač se skládá z kovové a plastové části. Zbylé komponenty jsou realizovány pomocí plastového materiálu. Navržená krabička 3D modelu v aplikaci Fusion 360 se nachází na obrázku 18.



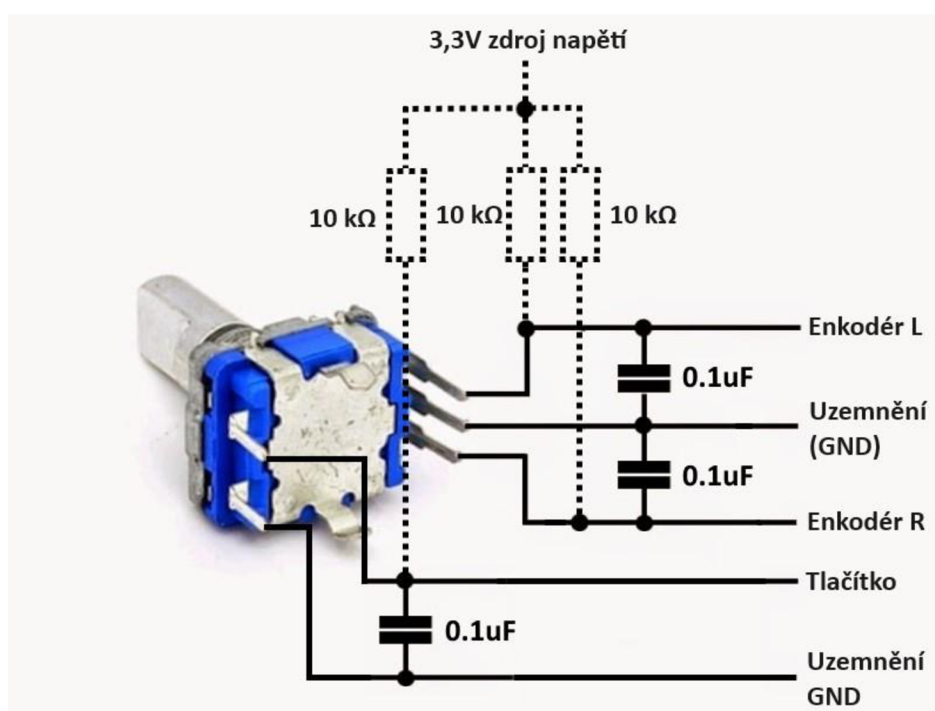
Obrázek 18 – Předpokládaný model výsledného ochranného boxu pro inventarizační zařízení

## 4.2 Konstrukce zařízení a doprovodné webové aplikace

### 4.2.1 Sestavení prototypu

Vytváření programového vybavení pro ESP32 předcházelo otestování a propojení všech vybraných komponent. K tomu posloužilo nepájivé pole, takzvaný breadboard. Výhodou zapojení přes nepájivé pole je, že všechny připojené díly lze snadno nahradit jinými. Toho bylo využito i v této práci, jelikož při testování bylo zjištěno, že zakoupený OLED display je vadný a musel být nahrazen novým. Aby bylo možné při vývoji využít všechny piny na ESP32, bylo umístěné mezi dvě nepájivá pole

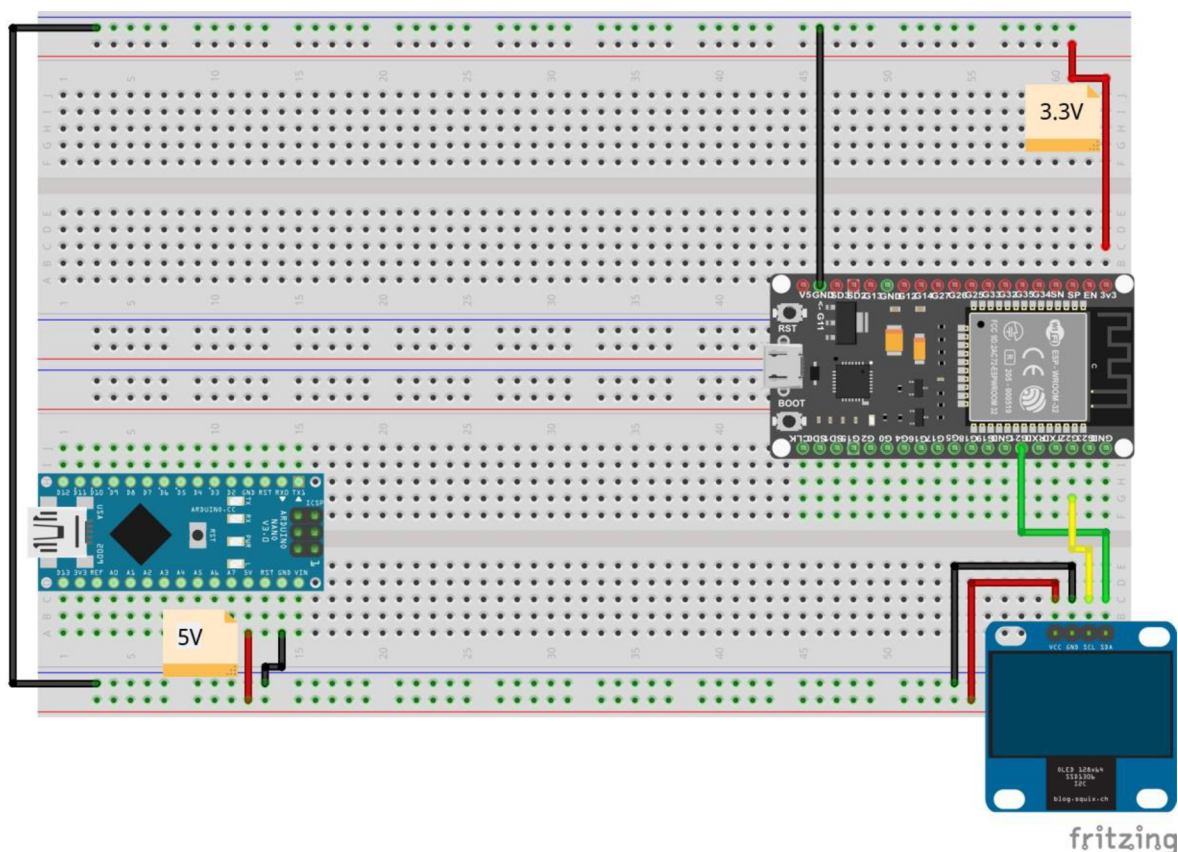
K ovládání celého řešení byl vybrán rotační enkodér s tlačítkem. Ten v sobě obsahuje disk s rovnoměrně rozloženými vodivými a nevodivými zónami. Při jeho otáčení tak vznikají impulzy, které lze zachytávat a určit, o kolik a jakým směrem se otočil. Vybraný enkodér také obsahuje membránový spínač, který je v projektu využit jako potvrzovací tlačítko. Aby ESP32 dokázalo s enkodérem spolupracovat, bylo nutné přidat na jeho výstupy pull-up rezistory, které zajistí, že digitální piny na vstupu ESP32 budou schopny zaznamenat změnu stavu. Zároveň jsou přidány na všechny výstupy pull-down kondenzátory, které se starají o to, že při zmáčknutí tlačítka, nebo otočení enkodéru, zůstane ještě na výstupu logická hodnota HIGH, dokud se kondenzátor nevybije [66]. Schéma zapojení je na obrázku 19.



Obrázek 19 - Zleva: Schématické zapojení rotačního enkodéru [66]

Celý prototyp byl sestaven bez zapojení akumulátorů, a to tak, aby se eliminovalo riziko poškození elektroniky vlivem zkratu nebo vyššího než přípustného napětí. K napájení čtečky čárových kódů, RFID čtečky a displeje bylo nutné přivést 5 V z Arduina Nano, které sloužilo jako napájecí zdroj. ESP32 i Arduino samotné bylo během testování a vývoje napájené přes USB. Veškerý vývoj probíhal v programu Arduino IDE, které přes integrovaný repositář nabízí vývojáři velké množství dostupných knihoven pro usnadnění vývoje.

Použitý display je napájený 5 V z Arduina Nano. SCL pin (sériové hodiny) je přiveden na pin D22 u ESP32. Pin SDA (sériová data) je přiveden na ESP32 pin D21. Pro komunikaci s ESP32, které je v roli master, tedy díky I<sup>2</sup>C sběrnici stačí pouze dva vodiče. GND (uzemnění), je zapojeno do jednoho ze čtyř GND pinů na ESP32, které jsou současně spojeny s piny na Arduinu Nano.



Obrázek 20 - Zapojení 1.3 palcového I<sup>2</sup>C displeje k ESP 32 s využitím nepájivého pole

Pro usnadnění vývoje byly využity dvě knihovny od společnosti Adafruit. První z nich je Adafruit GFX Library [67], která je velice populární při práci se všemi druhy displejů.

Knihovna poskytuje sadu grafických základních prvků, jako jsou body, linky, kruhy a podobně. Adafruit je americká společnost vyvíjející a prodávající hardware. Všechny její produkty jsou open-source a jejich koupí se podporuje další vývoj. Druhou použitou knihovnou je Adafruit\_SH1106 [68]. Knihovna byla přepracována uživatelem GitHubu *wonho-maker* speciálně pro displeje SH1106, úpravou původní knihovny SSD1306 od Adafruit [69]. Jedná se o rozšiřující grafickou knihovnu pro předchozí knihovnu GFX. Zároveň dodává podporu I<sup>2</sup>C a SPI pro komunikaci s mikrokontroléry. Před využitím v projektu ale musela být celá knihovna ještě upravena, jelikož zakoupené displeje měly nepatrně odlišnou indexaci a atypicky byla adresa prvního pixelu na místě 1x1, namísto 0x0. Úprava a dodání funkcí vznikla kombinací s další knihovnou od Adafruit SH110x [70], která poskytuje univerzální podporu pro displeje z řady SH110. Následující ukázka kódu vypíše na displej text „Zdravím PEF!!“

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SH110X.h>

#define SCREEN_WIDTH 128 // OLED displej sirka v pixelech
#define SCREEN_HEIGHT 64 // OLED displej vyska v pixelech
#define SCREEN_ADDRESS 0x3C // typ disp.:0x3D pro 128x64,0x3C pro 128x32
#define OLED_RESET -1 // resetovací pin, nutný pro fungování knihovny
#define OLEDC 22 // oled clock pin
#define OLEDD 21 // oled data pin
Adafruit_SH1106G display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET); //inicializace objektu display

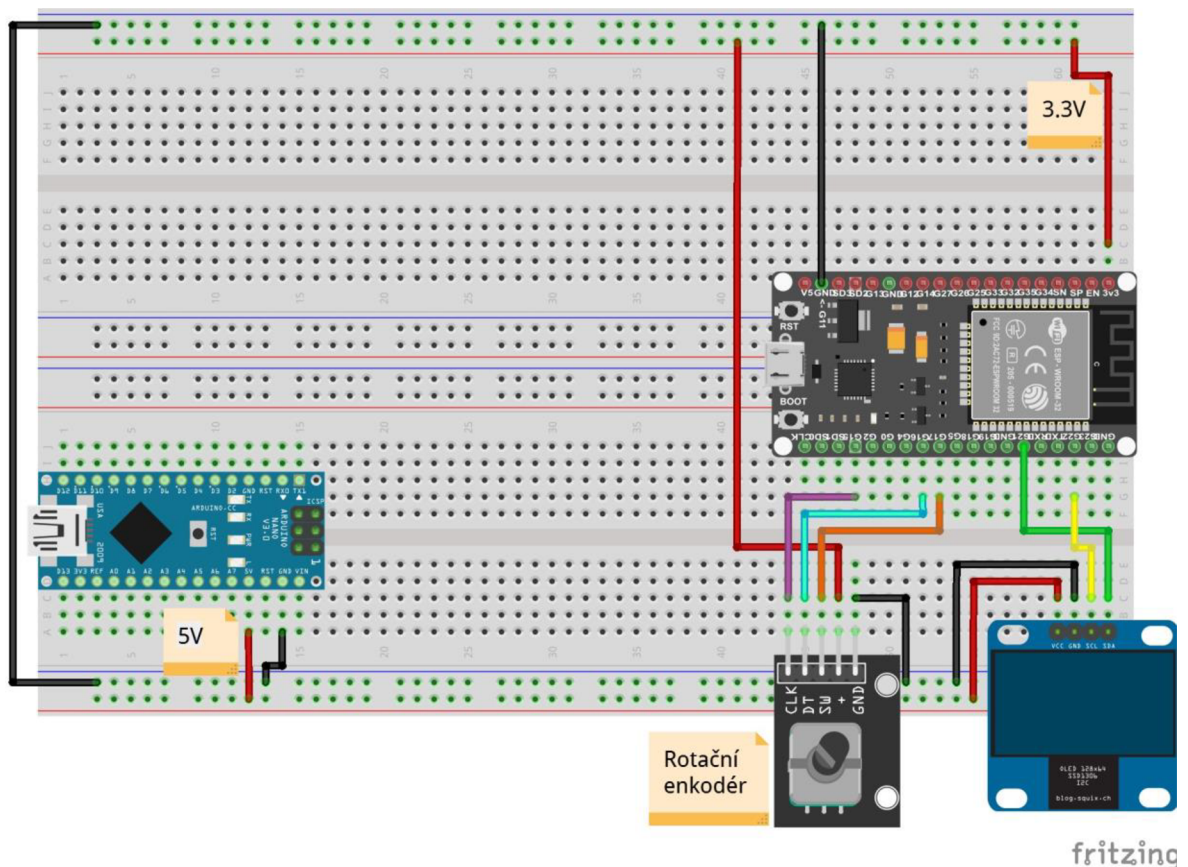
void sayhello(vodi) {
  display.clearDisplay(); // vycisteni cele plochy displeje
  display.setTextSize(1); // nastaveni velikosti fontu
  display.setTextColor(WHITE); // vyber barvy fontu
  display.setCursor(0, 0); // umisteni kurzoru
  display.println(F("Zdravim PEF!!")); // vytisknuti textu
  display.println(); // vytisknuti nove radky
}

void setup() {
  display.begin(SCREEN_ADDRESS, true)
}

void loop() {
  sayhello(); // volani funkce pro tisk na displej
}
```

Úryvek kódu 2 - Kód pro zobrazení textu na displeji

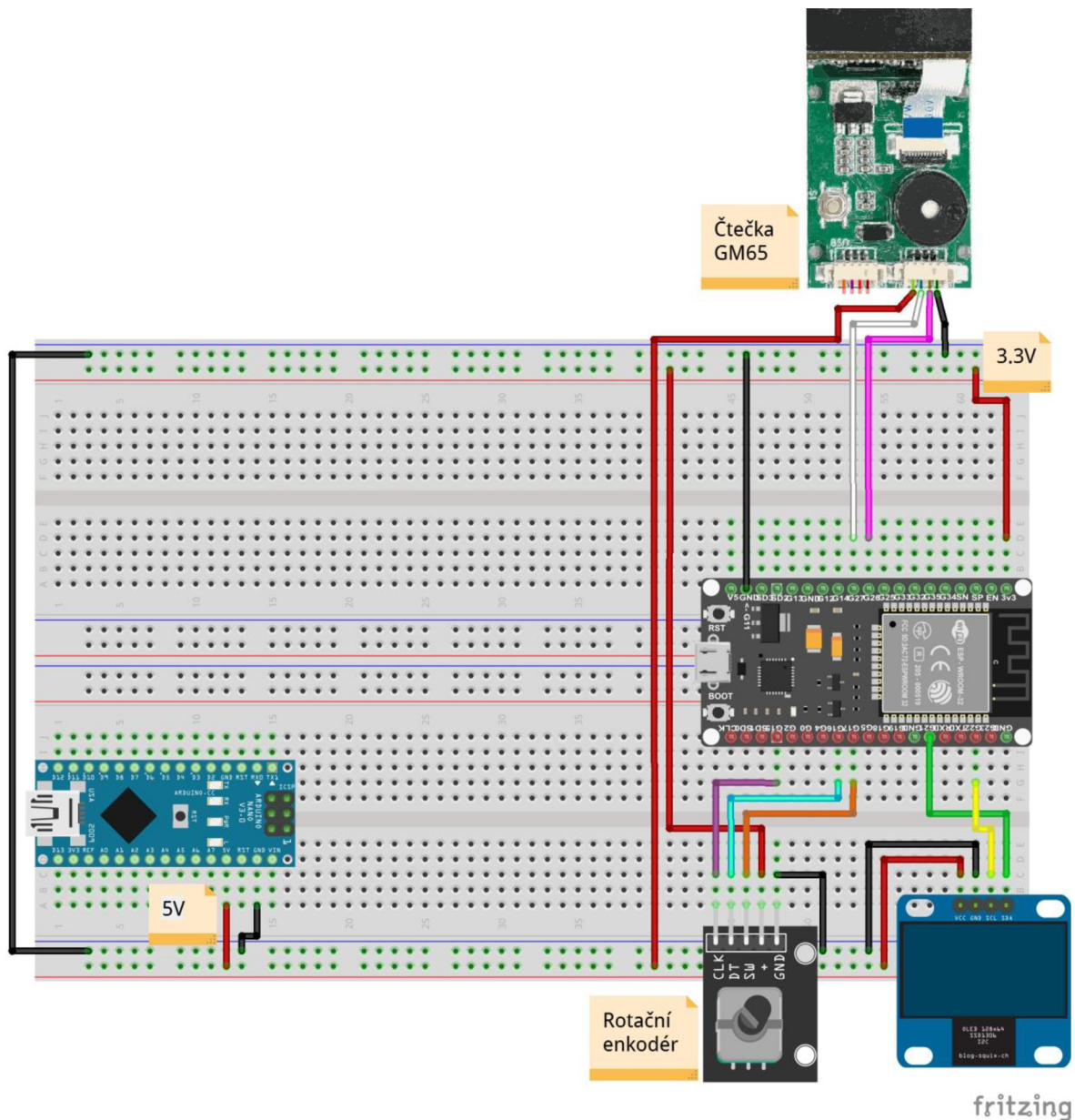
Další přidanou komponentou byl rotační enkodér. Ještě před samotným zapojením k němu byly připájeny všechny tři odpory a kapacity, přesně podle obrázku 19. Pájení zde bylo nutné, jelikož při čtení se neustále ztrácely hodnoty, nejspíše kvůli slabým kontaktům. Napájení bylo připojeno na 3,3V výstup u ESP32, GND na jeden ze zemnicích pinů. Výstupy L a R byly připojeny na piny D16 a D17, výstup tlačítka na pin D15. Výpočet aktuální pozice enkodéru zajišťovala knihovna od Igora Antoliče, který ji napsal přímo pro ESP32 a ESP8266. Knihovna kromě výpočtu pozice nabízí i další užitečné vlastnosti, jako je třeba akcelerace, díky které může uživatel snadněji navolit vyšší číslo, aniž by musel tolikrát otočit pozice. Knihovna je často používaná u domácích rádií, kdy je enkodérem nastavovaná vysílací frekvence.



Obrázek 21 - Přidání rotačního enkodéru do nepájitvého pole

V tuto chvíli již bylo zapojené ovládání a displej pro zobrazování údajů. Zatím ale nebylo co zobrazovat, proto další připojenou součástí byla čtečka čárových kódů GM65. Dva datové piny RX (příjem) a TX (volání) pro sériovou komunikaci byly zapojeny do ESP32 pinů D26 a D27. 5V napájení zajišťuje Arduino NANO a pin GND je opět

přiveden na jeden z GND pinů na ESP32. Samotné nastavení čtečky probíhá čtením QR kódů dodaných výrobcem. V tomto případě byla čtečka nastavena na komunikaci přes sériovou linku, tedy přes sběrnici UART.



Obrázek 22 - Schéma zapojení čtečky GM65

Veškerá komunikace ze čtečky GM65 přichází po sériové lince. Při zapojení do přípravku Arduino by bylo třeba využívat knihovnu SoftwareSerial.h [71] a replikovat funkce hardwarových sériových portů. Arduino totiž ve verzích UNO a NANO neposkytuje žádné další volné hardwarové porty pro využití v uživatelských projektech. Naštěstí ESP32 má integrované hardwarové tři sériové porty, stejně jako dražší a větší Arduino MEGA. Porty se nazývají U0UXD, který je vyhrazen pro programování ESP32, včetně úpravy

zavaděče. Dále U1UXD a U2UXD, které jsou volné a mohou být využity při tvorbě vlastních projektů [72]. Na desce jsou označeny jako RX0, TX0 a RX1, TX1. Pro čtení dat ze čtečky tedy stačí jen pár řádků kódu, protože o veškerou logiku spojenou s rozšifrováním kódů se stará výpočetní jednotka na modulu GM65 a po sériové lince již sdělují jednotlivé kódy. Čtečka GM65 byla nastavena do manuálního režimu. Bylo tedy nutné pro její spuštění stisknout tlačítko integrované na její základní desce. To ale bylo odpájeno a nahrazeno externím, vyvedeným ven.

```
#define TX 26      // pin TX pro QR
#define RX 27      // pin RX pro QR
#define QR Serial1 // přiřazení seriového portu

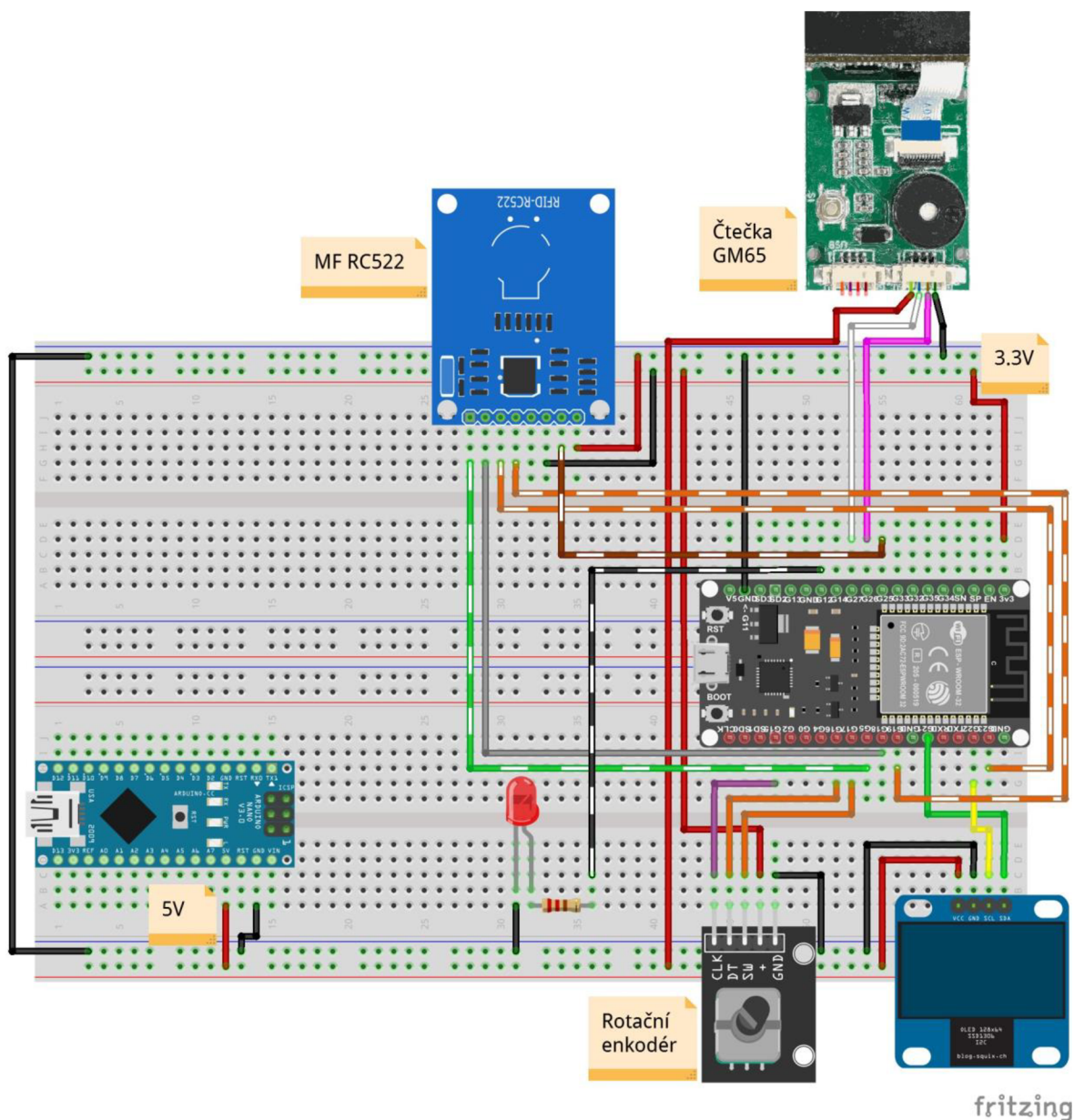
void setup() {
  Serial.begin(9600);
  QR.begin(9600, SERIAL_8N1, TX, RX); // Definování seriového portu pro QR
  čtečku GM65
}
void qrRead() {
  // funkce uloží do proměnné x hodnotu z výstupu sériové linky 2,
  // kterou následně vypíše na sériovou linku 1
  byte x = QR.read();
  Serial.println(x);
}
void loop() {
  if (QR.available()) qrRead(); // je spojení se navázáno? -> volá seqeRead()
}
```

Úryvek kódu 3 - Nejjednodušší kód pro čtení čárových kódů s využitím čtečky GM65

Poslední komponentou je RFID čtečka karet MFRC522. Napájení je připojeno na 3,3V port u Arduino NANO a GND na ESP32. Čtečka má celkem osm pinů a komunikuje přes SPI sběrnici. Pro správné fungování stačí zapojit pinů sedm, a v projektu byly zapojeny následovně:

Tabulka 4 - Přiřazení výstupních pinů z RC522 k pinům na ESP32

RFID RC522 piny	ESP32 piny
RST	D25
MISO	D19
MOSI	D23
SCK	D18
SDA	D5



Obrázek 23 - Schéma zapojení všech součástek do nepájivého pole, včetně čtečky karet RC522 a notifikační LED diody

Využita byla knihovna vytvořena Arduino komunitou *MFRC522.h* [73]. Díky této knihovně je čtečka kompatibilní s většinou mikrokontrolerů z rodiny Arduino, ale i dalších, mezi kterými je také ESP32. Všechny piny jsou přiřazeny tak, jak požaduje knihovna, jen RST a SDA jsou volitelné, přesněji viz *Tabulka 4*. Díky této knihovně jdou RFID karty nejen číst, ale dají se přepisovat i jejich hodnoty, včetně unikátních identifikátorů. Nejzajímavější funkcí je klonování karet, které umožňuje duplikovat i šifrované karty. V rámci testování byl vytvořen funkční klon školní studentské karty. Kromě RFID modulu byla ještě dodána LED (světlo emitující dioda) dioda. Ta se rozsvítí po každém úspěšném



přečtení karty nebo čárového kódu. Vizuálně tak informuje uživatele o úspěšném přijetí vstupu.

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

#define SS_PIN 5           // ESP32 pin GIOP5
#define RST_PIN 25        // ESP32 pin GIOP25
#define LED 12            // pin pro LED

MFRC522 rfid(SS_PIN, RST_PIN); // instance třídy rfid

void fRFID() {
  if (rfid.PICC_IsNewCardPresent()) { // nová karta je přiložena?
    if (rfid.PICC_ReadCardSerial()) { // byla karta přečtena?
      // tisk ID karty na sériový monitor v hex formátu
      for (int i = 0; i < rfid.uid.size; i++) {
        sRFID += String(rfid.uid.uidByte[i], DEC);
      }
      Serial.println(sRFID);
      sQR.trim();           // odstraní se whitespacy
      blink();             // led blikne
      rfid.PICC_HaltA();    // pozastavení čtení a zabrání opakování
      rfid.PCD_StopCrypto1(); // zastaví se decryptování
    }
  }
}

void blink() { // funkce pro bliknutí přidané LED diody.
  int wait = 200;
  digitalWrite(LED, HIGH); // rozsvítí se LED
  delay(wait);             // čeká se po dobu "wait" v ms
  digitalWrite(LED, LOW);  // vypne se LED
  delay(wait);
}

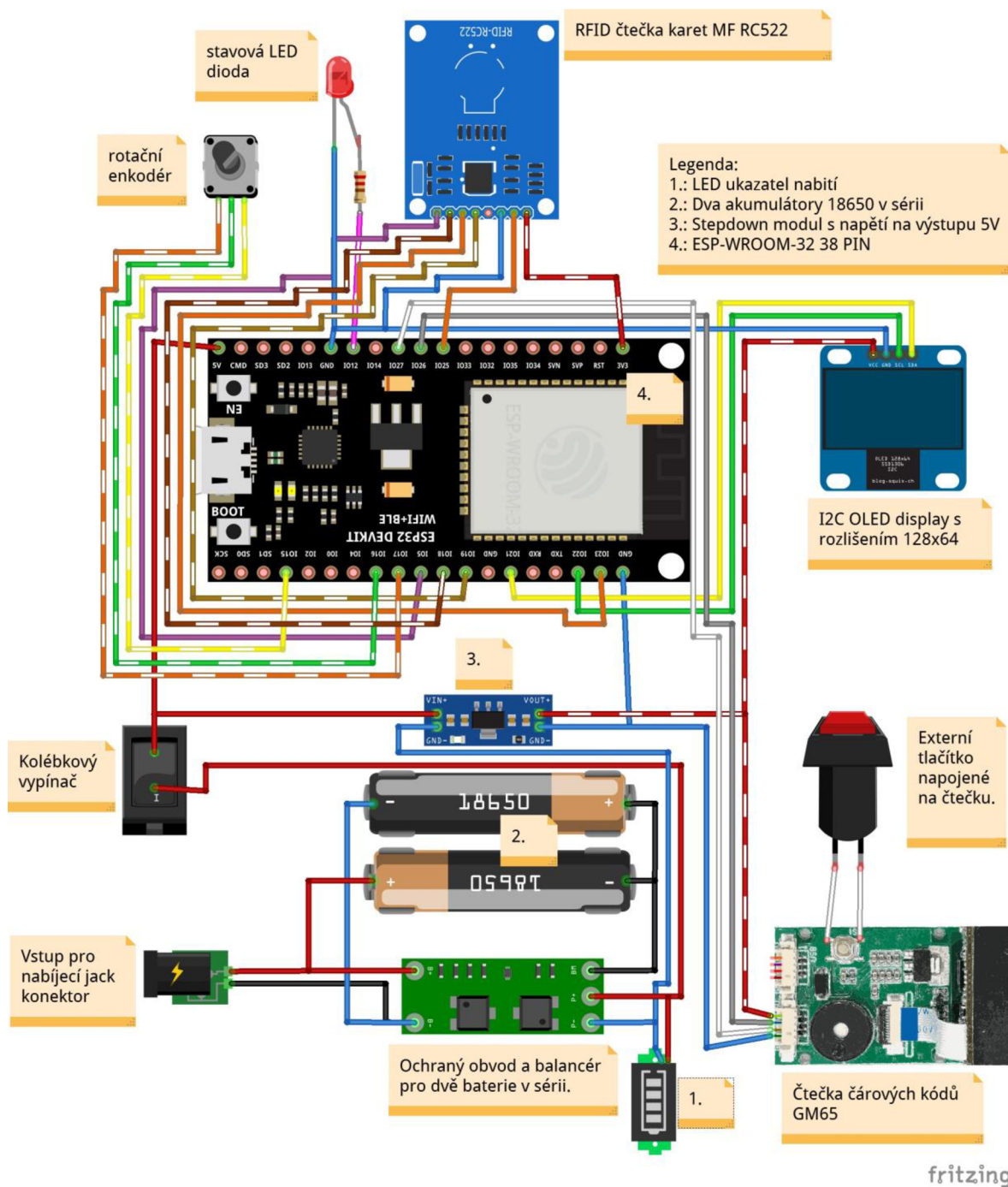
void setup() {
  Serial.begin(9600);
  SPI.begin();             // inicializace SPI sběrnice
  rfid.PCD_Init();         // inicializace MFRC522 čtečky karet
  pinMode(LED, OUTPUT);
}

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

Úryvek kódu 4 - Ukázka přečtení ID RFID karty a jeho vypsání v šestnáctkové soustavě na sériovou linku včetně funkce pro blikání LED diody blink()

## 4.2.2 Kompletace řešení

Po úspěšném otestování a propojení všech dílů byly všechny komponenty znovu rozpojeny, aby bylo možné nahradit nepájivé pole napevno připájenými vodiči. Pevné spojení přináší vyšší spolehlivost přenosu dat a minimalizuje riziko rozpojení nebo zkratů. Jako poslední byly připojeny akumulátory. Schéma výsledného zapojení je na následujícím obrázku 24:



Obrázek 24 - Finální schéma zapojení všech součástek, včetně napájecího obvodu a externího spínače pro čtečku čárových kódů

Náročnou částí pájení bylo vytvořit akumulátor. Li-Ion články jsou náchylné na vysoké teploty, které je poškozují a klesá jejich celková kapacita. Při teplotách nad 150 °C navíc hrozí jejich exploze s následným intenzivním zahořením [74]. Pájení ale vyžaduje řádné prohřátí pájeného materiálu, aby výsledný spoj byl pevný a neobsahoval mikropraskliny, které mohou zvýšit odpor a vést k přehřívání až zkratu. V průmyslové výrobě se využívají bodové svářečky, které dokážou vytvořit v jednom bodě teplotu přesahující 25000 °C, ovšem jen na několik milisekund. V případě baterií je teplota nižší. [75] Na fotografii 25 je zachycen proces výroby akumulátoru.

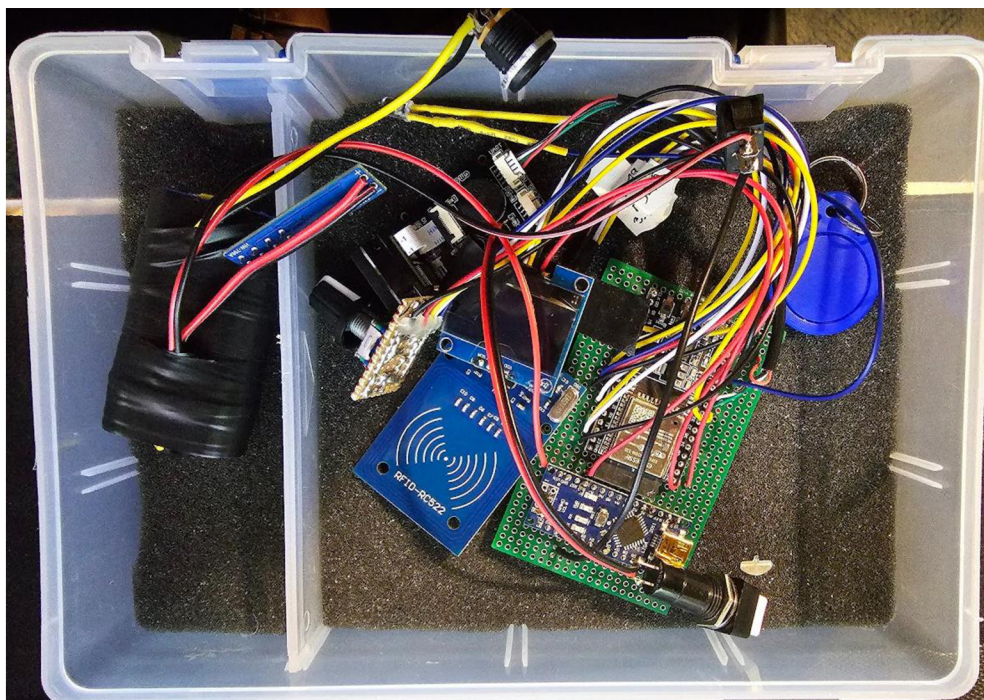


Obrázek 25 - Spojené akumulátory s přepětovou ochranou a balancérem umístěné v pájecím držáku vyrobeném s využitím FDM tiskárny

Bodové svářečky jsou ale finančně nákladné a v domácích podmínkách pro ně prakticky neexistuje využití, proto byla k zapojení baterií do série a instalaci ochranných prvků využita pájecí stanice. Oba póly byly obroušeny abrazivním brusným papírem a odmaštěny acetonem pro lepší přilnavost pájky. Současně na ně byla aplikována tenká vrstva kalafuny rozpuštěné v ethanolu, také ke zvýšení přilnavosti a omezení oxidace při pájení. Kladný pól Li-Ion akumulátoru není v přímém kontaktu s vnitřním vinutím, dělí ho vzduchová kapsa, která funguje jako tepelný izolant. Proto je možné ho prohřívát po delší dobu a snadněji připájet vodič. Teplota pájecího hrotu byla nastavena na 300 °C a prohřívání trvalo necelých

osm sekund, než byl do pájky – směsi cínu a olova, vložen vodič a vyčkalo se na pozvolné vychladnutí. Záporný pól akumulátorů byl prohříván pouze čtyři sekundy, teplotou 400 °C, měděný drát byl již ponořen v pájce, když byla zahřívána. Chladnutí bylo do vytvrdnutí pájky pozvolné za pokojové teploty. Poté byly akumulátory z bezpečnostních důvodů umístěny do chladného prostředí s průměrnou teplotou -5 °C, jelikož byly na dotek vařící. Po úplném vychladnutí a opětovném ohřátí na pokojovou teplotu byla zjištěna skutečná kapacita soustavy, která se nijak nezměnila oproti stavu před pájením.

Pájením se spojil minusový pól jednoho článku s plusovým druhého. Současně byl přidán na kladný pól další vodič, který vedl do BM výstupu na balancéru a přepětové ochraně HX-2S-01, aby modul mohl správně vypočítat napětí na každém článku zvlášť. Na konektory B+ a B- byly připájeny vodiče z druhého konce baterií, jak jde vidět na obrázku 25. Na stejné konektory byl připojen i nabíjecí jack konektor. Konektory P+ a P- na ochranném modulu byly propojeny s ESP32 podle schématu na obrázku 24. Mezi konektorem P+ a pinem 5V na ESP32 byl vložen kolébkový vypínač, kterým se dá zařízení zapnout nebo vypnout. Za kolébkový vypínač byl ještě přidán step-down modul AMS1117 pro vytvoření 5V výstupu z baterie. Na obrázku 26 je kompletní zařízení, bez ochranného boxu.

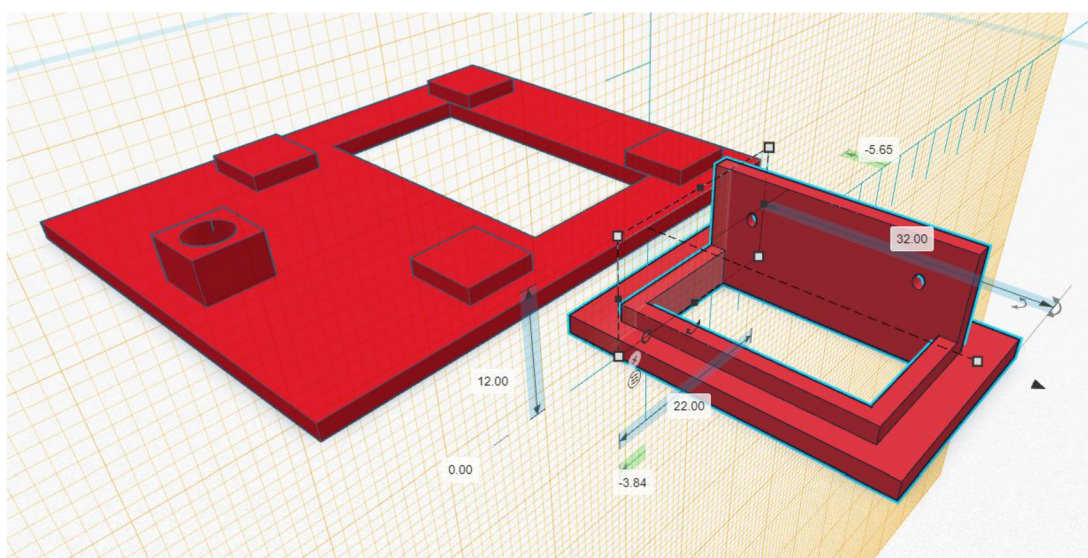


Obrázek 26 - Finální pevné zapojení všech komponent. Akumulátor vlevo, i všechny plochy, které mohou způsobit zkrat, je obalen ve flexibilní černé pásce

### 4.2.3 Tvorba a tisk 3D modelu ochranného boxu

Návrh 3D modelu byl pro svou jednoduchost realizován v aplikaci AUTODESK Tinkercad. Jedná se o bezplatný online 3D modelovací software, využívaný k tvorbě jednoduchých, modelů. Jeho předností je jednoduchost a rychlost navrhování. Při tvorbě modelu se vycházelo zejména z grafického prototypu, popsaném v kapitole 4.1.3. Rozměry všech součástí, včetně pájivého pole, na kterém se nacházelo ESP32, regulátor napětí AMS1117 byly změřeny digitálním posuvným měřidlem s přesností 0,2 mm. Poté byly postupně navrhovány průřezy a úchyty pro jednotlivé komponenty, kterými se na závěr udělaly průřezy nebo došlo ke sjednocení s navrženým boxem.

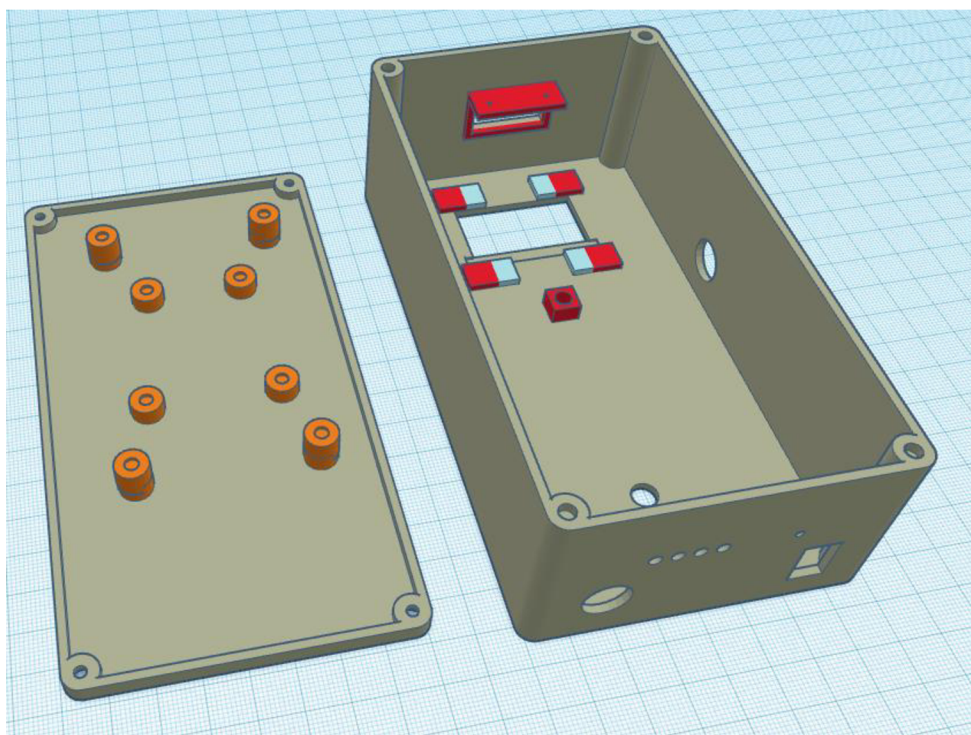
Nejprve byl vytvořen průřez pro 1,3 palcový display spolu výstupky, do který se pomocí upraveného pájecího hrotu zataví mosazné závity. Tyto závity slouží pro zafixování displeje pomocí plastových šroubů. Pod průřezem na displej je umístěn držák na LED diodu, který má na spodní straně tloušťku pouhých 40 mm. To odpovídá dvěma vrstvám 3D tisku a světlo z LED diody díky tomu ambientně prosvítá pouze v tomto místě. Světlo informuje uživatele o provedené akci. Současně byl navržen průřez na čtečku čárových kódů GM65. Horní výčnělek má dva průřezy pro uchycení modulu šrouby a pomáhá držet celý skener pevně na místě.



Obrázek 27 - Modely průřezů pro displej a čtečku čárových kódů GM65 vytvořené v aplikaci AUTOCAD Tinkercad

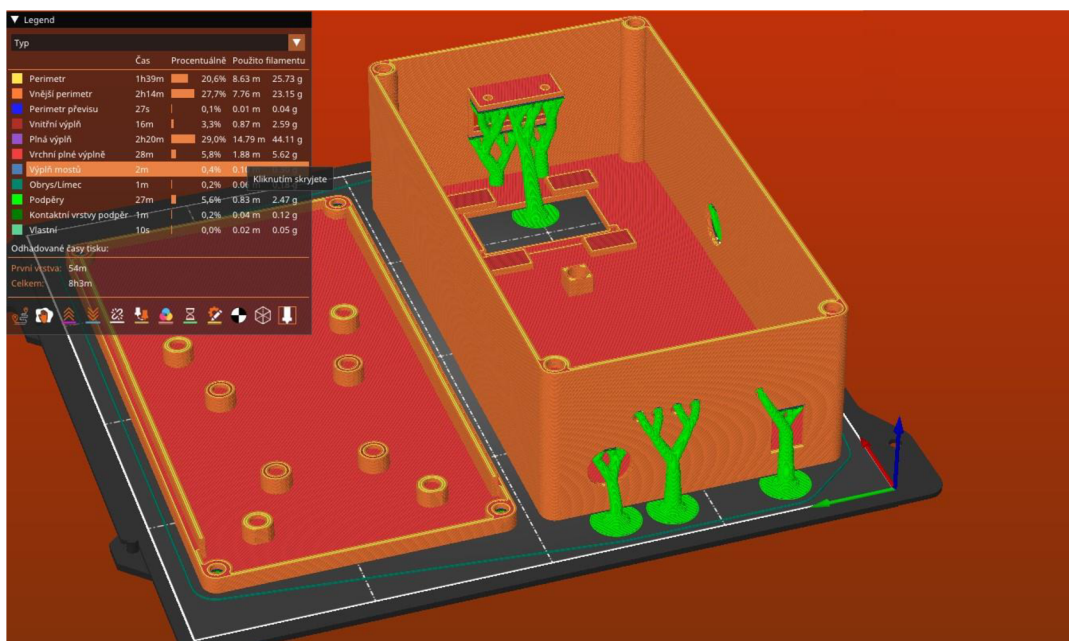
Stejným postupem byly vytvořeny další průřezy pro napájecí jack konektor, který nahradil v původním modelu zamýšlený USB port, ukazatel stavu akumulátoru, otvor pro kolébkový

vypínač, rotační enkodér, tlačítko čtečky čárových kódů a výstupky pro zatavení mosazných závitů. Samotný model se skládal z krabičky a jejího víka, které je umístěné ve spodní části. Na víku jsou připraveny distanční výstupky pro zatavení mosazných závitů, do kterých se zašroubují všechny komponenty. Čtečka čárových kódů se umístí do spodní části tak, aby se dnem krabičky mohly načítat RFID karty. Ve druhé vrstvě je pak umístěno ESP32. Ve víku je dále vyhrazený prostor pro akumulátor. Zbývající prostor krabičky je převážně vyplněn kabely vedoucími z komponent. Velikost a jednotlivé rozmístění bylo několikrát upraveno v závislosti na ergonomii ovládání a volného prostoru uvnitř boxu. Box má rozměry 15 cm na výšku, 8 cm šířku a 5 cm hloubku a jeho model je zachycen na obrázku 28.



*Obrázek 28 - Finální verze 6.0 ochranného boxu a víka pro inventarizační zařízení*

Model byl vytvořen ve formátu STL, který bylo nutné nařezat ve sliceru. Byla zvolena aplikace PrusaSlicer ve verzi 2.6.0, která přinesla revoluční organické podpěry, které byly při tisku využívány [76]. Při nastavení výšky vrstvy 0,2 mm byl odhadovaný čas tisku osm hodin, se spotřebou filamentu 104 gramů, tedy přibližně 35 metrů. Při tisku byl použit materiál PLA z důvodu příznivé ceny (2,6 €) a kvality finálního výtisku. Slicovaný model je na obrázku 29.



Obrázek 29 - Rozlicovaný 3D model s organickými podpěrami a detailními údaji o průběhu tisku.

Byly testovány také materiály PETG a ABS, ale ty při chladnutí nepatrně měnily své rozměry a otvory na display nebo čtečku pak byly příliš těsné. Celkem bylo vytištěno pět zkušebních a testovacích modelů, kdy některé nebyly dotištěny kvůli chybnému nastavení tiskárny. Dále byl vytištěn nespočet pomocných průřezů, které sloužily k co nejpřesnějšímu vyměření potřebných rozměrů. Výtisky jsou zachyceny na obrázku 30. Testovací filamenty byly od firmy Creality a finální výtisk byl tvořen filamentem Průšament z holešovické továrny Prusa Research. Creality filamenty jsou až o 10 € na kilogram levnější než filamenty od Průši, ovšem kvalitou zaostávají.

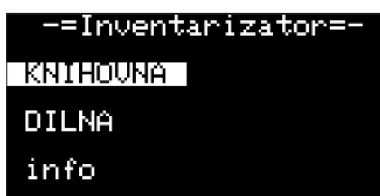


Obrázek 30 - Nezdařilé a testovací výtisky (žluté) s viditelnými verzovacími čísly.

#### 4.2.4 Tvorba menu

Tato podkapitola popisuje vývoj programového vybavení zaměřeného především na displej a jednotlivé položky menu. Úryvky kódu popisující fungování jednotlivých komponent, jako jsou čtečka GM65 nebo RC522 byly vysvětleny v kapitole 4.2.1. Níže je popsán kód pro pohyb v menu zařízení.

První obrazovka, kterou uživatel po zapnutí napájení zařízení uvidí je úvodní menu, viz obrázek 31. Tomu dominuje název zařízení „*Inventarizator*“ v první řádce. Dalšími třemi řádky jsou „*KNIHOVNA*“, která uživatele nasměruje do správy položek knihovny. Další položkou je „*DILNA*“, která je obdobná jako knihovna. Poslední je položka „*info*“, která uživatele nasměruje na obrazovku s podrobnostmi o aktuální bezdrátové síti, na kterou je připojen. Právě označená položka je bíle podbarvená s černým textem, ostatní nevybrané jsou psány bílým textem.



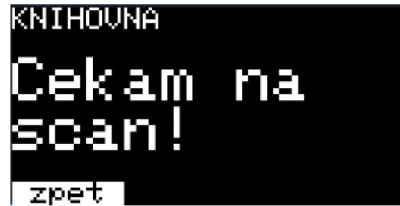
Obrázek 31 - Úvodní obrazovka inventarizačního zařízení s kurzorem na položce KNIHOVNA

```
int menuItems = 3; // specifikování počtu položek v menu
const char *options[menuItems] = { // statické pole obsahující položky menu
    " KNIHOVNA ", " DILNA ", " info "};
if (entered == -1) {
// pokud se pozice v zařízení nezměnila/bylo zapnuto, hodnota proměnné je -1
display.clearDisplay(); // vyčištění displeje
display.setTextSize(1); // volba velikosti fontu
display.setTextColor(WHITE); // volba barvy fontu
display.setCursor(0, 0); // kurzor na pozici 0x0
display.println(F(" -=Inventarizator=-")); // tisk názvu zařízení
display.println(); // vložení prázdné řádky
for (int i = 0; i < menuItems; i++) { // vypsání položek menu
    if (i == selected) { // vybrání aktivní položky
        display.setTextColor(BLACK, WHITE); // bílé podbarvení, černý font
        display.println(options[i]); // tisk aktuální položky menu
        display.println(); // nová řádka
    } else if (i != selected) { // položky kde není kurzor
        display.setTextColor(WHITE); // mají bílý font
        display.println(options[i]); // a jsou vytištěny
        display.println();
    }
} } }
```

Úryvek kódu 5 - Příkazy pro vytisknutí položek úvodního menu



Pokud je v úvodním menu pomocí rotačního enkodéru zvolena položka *KNIHOVNA* nebo *DILNA*, zobrazí se uživateli obrazovka z obrázku 32. V ten moment zařízení čeká na data z jedné ze čteček kódů voláním funkcí „*getQR(sQR)*“ a „*getRFID(sRFID)*“. Pokud kód obdrží, dotáže se na něj v databázi, jestli o něm eviduje záznam. Pokud ne, vytvoří v databázi nový záznam a informuje o tom uživatele. V případě že záznam existuje, je otevřeno menu s možností editace záznamu, jako na obrázku 33.



Obrázek 32 - Obrazovka, na které je zařízení ve čtecím módu a čte kódy z RFID i GM65 čtečky

```
sQR = ""; // vyprázdnění řetězce sQR
sRFID = ""; // vyprázdnění řetězce sRFID

display.clearDisplay(); // vyčištění displeje
display.setTextSize(1); // volba velikosti fontu
display.setTextColor(WHITE); // volba barvy fontu
display.setCursor(0, 0); // kurzor na pozici 0x0
display.println(F("KNIHOVNA\n")); // tisk názvu obrazovky
display.setTextColor(WHITE); // volba barvy fontu
display.setTextSize(2); // větší font
display.println("Cekam na\nscan!"); // textová výzva uživatele
display.setTextSize(1); // původní font
display.setTextColor(BLACK, WHITE); // podbarvené tlačítko
display.println(); // nová řádka
display.println(" zpet "); // tlačítko zpět

// getQR() a getRFID() jsou volány jen pokud je nějaký vstup v zásobníku
if (sQR != "") scanRes = getQR(sQR); // získání čárového kódu
if (sRFID != "") scanRes = getRFID(sRFID); // získání RFID kódu
```

Úryvek kódu 6 - Výzva uživatele k načtení kódu a volání funkcí pro jeho čtení

Na obrázku 33 jsou vidět všechny zbylé obrazovky, které ještě nebyly popsány. Horní dva snímky ukazují akci, kterou vyvolá naskenování neznámého identifikátoru, ať již RFID tagu, nebo čárového kódu. Spodní levý snímek ukazuje situaci, kdy byla kniha nebo nářadí v databázi podle naskenovaného kódu nalazena. Z databáze je načten název knihy a počet kusů. Submenu nabízí možnost záznam editovat, a to přidáním dalšího kusu nebo jeho vyskladněním. Na posledním snímku je obrazovka zobrazující technické informace

o aktuální síti, ke které je zařízení připojeno. Zdrojový kód je pro jednotlivé obrazovky tvořen analogicky, jako úryvku kódu 5 a 6.



Obrázek 33 – Obrazovky na zařízení Inventarizátor

#### 4.2.5 Tvorba databáze

Pro účely projektu byla vybrána jako nejvhodnější varianta webhosting od české firmy Wedos, ve variantě NoLimit. Ten již v základu nabízí HTTPS certifikát a poskytuje 2 GB prostor pro uživatelské databáze. Wedos poskytuje open-source databázi MariaDB, která vychází z MySQL a vznikla jako její fork, tedy nástupnická vývojová větev pod licencí GNU GPL. Cena za hosting je přibližně 40 € ročně. Pro základní potřeby inventarizačního zařízení byly vytvořeny v databázi „DP“ dvě tabulky – „books“ pro inventarizaci knihovny a „workshop“ ukládající data z dílny. Postup jejich návrhu je analogický, proto je popsána pouze tabulka pro knihovnu. Ve struktuře tabulky se zaznamenává celkem osm sloupců, kterými jsou:

- **id** – unikátní číslo přidělené každému záznamu. Jedná se o celočíselné číslo typu *int* a každý nový záznam dostane o jednotku vyšší číslo, díky automatické inkrementaci. V tabulce slouží jako primární klíč a je unikátní.
- **code** – jedná se o kód čárového nebo QR kódu ze skeneru. U knih je obvykle shodný s kódem ISBN (standartní číslo knihy). Sloupec je však evidovaný zvlášť, jelikož jsou publikace bez čárových kódů. Datový typ je *varchar*, což je string – textový řetězec proměnlivé délky. Každý záznam je unikátní.
- **rfid** – unikátní identifikační kód RFID tagů a karet, zjištěné čtečkou karet. Datový typ je *varchar*. Každý záznam je unikátní.
- **title** – sloupec ukládající názvy knih. Datový typ je *varchar*.

- **author** – sloupec ukládající jméno autora nebo autorů. Datový typ je `varchar`.
- **isbn** – unikátní identifikační číslo knihy ISBN. Není však povinné ho v tabulce mít, jelikož existují publikace bez ISBN. Datový typ je `varchar`. Každý záznam je unikátní.
- **published\_date** – sloupec ukládající rok vydání jednotlivých knih. Datový typ je `year(4)`, tedy čtyřciferný údaj o roce vydání.
- **created\_at** a **updated\_at** – sloupce ukládající přesný datum a čas vytvoření záznamu v databázi a datum jeho poslední aktualizace. Datový typ je `DATETIME`, který spravuje přesnou časovou značku ve formátu „YYYY-MM-DD hh:mm:ss“. U sloupce „`created_at`“ je přidána funkce „`current_timestamp()`“, která při vytvoření záznamu automaticky přidá přesný aktuální čas.
- **on\_stock** – sloupec evidující celkový počet kusů knihy s daným id. Datový typ je `unsigned int`, tedy pouze kladné celé číslo včetně nuly.
- **text** – pole s doplňujícím komentářem k záznamu. Datový typ je `text`, který může pojmut až 2 GB dat. V některých návrzích databází jsou pomocí tohoto datového typu ukládány multimediální soubory, jako například obrázky přebalů knih.

```
CREATE table books (
  `id` int (10) unsigned auto_increment,
  `code`          VARCHAR (255) NULL,
  `rfid`          VARCHAR (255) NULL,
  `title`        VARCHAR (255) NULL,
  `author`       VARCHAR (255) NULL,
  `isbn`         VARCHAR (20) NULL,
  `published_date` year(4) NULL,
  `created_at`   DATETIME DEFAULT current_timestamp(),
  `updated_at`   DATETIME NULL,
  `on_stock`     int(10) unsigned NOT NULL DEFAULT 1,
  `notes`        text NULL,
  PRIMARY KEY (id),
  UNIQUE (code, rfid, isbn)
);
```

Úryvek kódu 7 - SQL příkaz k vytvoření tabulky books

## 4.2.6 Vývoj webové aplikace pro správu dat v databázi

[Domů](#)

[Přidat záznam](#) [Export](#)  [Hledat](#)

ID	Kód	RFID	Název	Autor	ISBN	Skladem kusů	Rok vydání	Záznam vytvořen	Záznam aktualizován	
1	9788072372805	3415410325	SAS Příručka jak přežít	John Wiseman	9788072372805	4	2010	25.02.2023 11:08:50	18.11.2023 17:46:40	<a href="#">[upravit]</a> <a href="#">[smazat]</a>
2	9788027702459	4492034242109133	Znamení mrtvoila	Agustina Bazterrica	9788027702459	1	2010	25.02.2023 11:51:22	18.11.2023 17:47:59	<a href="#">[upravit]</a> <a href="#">[smazat]</a>
3	9788075550972	70212508	Atomové návyky	James Clear	9788075550972	3	2010	25.02.2023 11:52:13	18.11.2023 17:49:09	<a href="#">[upravit]</a> <a href="#">[smazat]</a>
4	9788074885532	1141039745	ÚZ 1513 Občanský zákoník	prof. Karel Eliáš a Mgr. Marek...	9788074885532	2	2010	25.02.2023 11:54:04	18.11.2023 17:52:25	<a href="#">[upravit]</a> <a href="#">[smazat]</a>
5	9788027136896	67216713	Osudné svědectví	Robert Bryndza	9788027136896	1	2010	25.02.2023 12:03:06	18.11.2023 17:53:07	<a href="#">[upravit]</a> <a href="#">[smazat]</a>
6	9788027704958	0	Dívka za ostatním drátem	Imogen Matthews	9788027704958	1	2023	25.02.2023 12:03:45		<a href="#">[upravit]</a> <a href="#">[smazat]</a>
7	9788027704682	0	Zvrhlý král	Rina Kent	9788027704682	1	2022	25.02.2023 12:04:35		<a href="#">[upravit]</a> <a href="#">[smazat]</a>
8	9788075550569	0	Faktomluva	Hans Rosling, Anna Roslingová	9788075550569	1	2018	25.02.2023 12:15:58	25.02.2023 12:16:30	<a href="#">[upravit]</a> <a href="#">[smazat]</a>

8 záznamů celkem

Obrázek 34 - Ukázka aplikace fungující nad tabulkou knihovna

Pro správu dat uložených v databázi bylo potřeba vytvořit webovou aplikaci. Pro přehledné zobrazení dat a práci s nimi byl vybrán open-source PHP (Hypertext Preprocessor) grid Lazy Mofo (LM) [77]. Jedná se o jedno třídovou PHP aplikaci s podporou CRUD (Create, Read, Update, Delete). CRUD je sad základních pokynů pro práci s databází. Výhodou integrace této funkce byla výrazná časová úspora při vytváření projektu. LM poskytuje přehledný a lehký design, všechny požadované funkce, a to včetně možnosti exportu uživatelských dat do CSV (hodnoty oddělené čárkou) souboru. Součástí je také soubor s možností volby jazyka. Český jazyk zde chyběl, proto byl přidán autorem práce. Nasazení proběhlo nahráním kódu aplikaci na webový server, nastavením propojení s databází a definováním funkcí pro práci s databází.

Upravit záznam	
ID:	1
Kód:	<input type="text" value="9788072372805"/>
RFID:	<input type="text" value="3415410325"/>
Název:	<input type="text" value="SAS Příručka jak přežít"/>
Autor:	<input type="text" value="John Wiseman"/>
ISBN:	<input type="text" value="9788072372805"/>
Rok vydání:	<input type="text" value="2010"/>
Skladem kusů:	<input type="text" value="4"/>
Poznámky:	<input type="text" value="Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin pede metus, vulputate nec, fermentum fringilla, vehicula vitae, justo. Aliquam erat volutpat. Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur? Cras elementum. In rutrum. Duis ante nisi molestiae vitae vehicula venenatis. tincidunt ac neda"/>

Obrázek 35 - Formulář pro úpravu nebo přidání nového záznamu do databáze

V následující ukázce kódu 10 je předvedeno, jakým způsobem byla aplikace připojena k databázi (v ukázce je napojení na lokální testovací databázi). Dále dochází k načtení sloupců a jejich přejmenování, k zobrazení v mřížce. Načtením aplikace LM se sama postará o designové prvky, jako je šířka sloupců a vybere vhodné pole pro editaci, v závislosti na datovém typu sloupce. Celá aplikace je napsána v PHP, s využitím CSS (kaskádové styly) na stylizaci obsahu a HTML pro editaci textu a prvků. Úvodní rozcestník práce je dostupný na adrese <https://dp.sebesta.fun>.

```

<?php
// přírupové údaje k lokální DB pro testování
$db_host = "localhost:3306";
$db_name = "dp";
$db_username = "root";
$db_password = "";
// připojení před PDO framework k databázi
$dbh = new PDO("mysql:host=$db_host;dbname=$db_name;", $db_username,
$db_password);
// vytvoření objektu LM, předání PDO připojení
$l = new lazy_mofo($dbh, 'cz-cs');
// query pro definování grid zobrazení
$l->grid_sql = "
SELECT `id`, `code`, `rfid`, `title`, `author`, `isbn`, `on_stock`,
`published_date`, `created_at`, `updated_at`, `id`
FROM books
WHERE COALESCE(`title`, '') LIKE :_search
or COALESCE(`author`, '') LIKE :_search
or COALESCE(`isbn`, '') LIKE :_search
ORDER BY `id` ASC
";
// přejmenování sloupců pro zobrazení v přehledu
$l->rename['id'] = 'ID';
$l->rename['code'] = 'Kód';
$l->rename['rfid'] = 'RFID';
$l->rename['title'] = 'Název';
$l->rename['author'] = 'Autor';
$l->rename['isbn'] = 'ISBN';
$l->rename['on_stock'] = 'Skladem kusů';
$l->rename['published_date'] = 'Rok vydání';
$l->rename['created_at'] = 'Záznam vytvořen';
$l->rename['updated_at'] = 'Záznam aktualizován';
$l->rename['notes'] = 'Poznámky';
?>

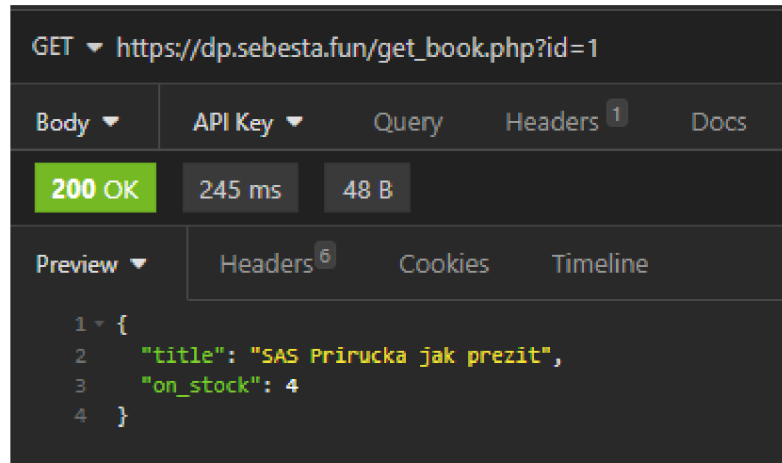
```

Úryvek kódu 8 - Načtení dat z DB do aplikace LM, s využitím frameworku PDO

#### 4.2.7 Komunikace ESP32 s databází

Přes framework PDO se k databázi připojují i další PHP skripty, které byly napsány pro práci s daty prostřednictvím HTTPS GET a POST protokolů. Jedním z nich je skript *get\_book.php*, kterému se předá parametr přes URL adresu ([https://dp.sebesta.fun/get\\_book.php?id=1](https://dp.sebesta.fun/get_book.php?id=1)). Nejprve je zadána adresa, kde se skript nachází. Za adresu se přidá otazník, za kterou se přidávají parametry. V tomto případě se dotazujeme na knihu s ID 1. V tomto případě je nám serverem vrácena HTTP odpověď, a to návratový kód 200 OK, více o návratových kódech bylo napsáno v kapitole 3.2. Skript *get\_book.php*

navíc vrací kromě návratového kódu i JSON (javaskriptová objektová notace) objekt, ve kterém je název knihy a počet kusů skladem. Na obrázku 36 je zobrazena odpověď GET dotazu.



Obrázek 36 - Ukázka zaslání GET požadavku na server přes API aplikaci Insomnia

```
<?php
$db_host = "localhost:3306";
$db_name = "dp";
$db_username = "root";
$db_password = "";

// připojení k DB je stejné jako v ukázce předtím
$dbh = new PDO("mysql:host=$db_host;dbname=$db_name;", $db_username,
$db_password, array(PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8"));

// proměnná id je předána pomocí GET parametru
$id = $_GET["id"];

// vytvoření JSON objektu z výsledku dotazu za pomoci SQL dotazu
echo json_encode($dbh->query("SELECT title, on_stock FROM books WHERE id=
$id")->fetchObject(), JSON_UNESCAPED_UNICODE);
?>
```

Úryvek kódu 9 - Skript na získání JSON stringu z databáze po předání ID knihy

Nejprve bylo potřeba ESP32 připojit k internetu, aby mohlo komunikovat se serverem. Toho bylo dosaženo s využitím knihovny *WiFi.h*, kterou napsala a dále vyvíjí sama společnost Arduino [78]. Kód pro ESP32 byl rozdělen s cílem zvýšit přehlednost do více souborů. Na začátku hlavičkového souboru *server.h* jsou uloženy přístupové údaje k WiFi, na kterou se má zařízení připojit. Knihovně *WiFi.h* se přístupové údaje předají funkcí na čtvrtém řádku, která musí být umístěna do těla funkce *setup()*.

```

#include <WiFi.h>
const char WIFI_SSID[] = "WifiJmeno";           // připojovací údaje k WiFi
const char WIFI_PASSWORD[] = "SilneHezkeHeslo"; // heslo k WiFi
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);          // připojení k WiFi

```

Úryvek kódu 10 - Správa údajů a připojení k WiFi síti

Po připojení může zařízení začít komunikovat přes internet. Pro zaslání GET požadavku skriptu *get\_book.php* byla napsána následující funkce *readFromDB(..)* v ukázce kódu 11. Ta očekává tři parametry, prvním je řetězec *PATH\_NAME*, což může být například „*get\_book.php*“, záleží, kterou funkcí je „*readFromDB()*“ volána. Druhý parametr je řetězec „*queryString*“. Při dotazování se na stav knihy, viz úryvek kódu 11, obsahuje tento string „*?id=X*“, kde proměnná *X* je nahrazena ID číslem požadované knihy. Posledním parametrem je ukazatel na adresu v paměti textového řetězce „*result*“, předávaná jako „*&payload*“. Právě do této proměnné, vytvořené v těle volající funkce je uložen návratový JSON řetězec, vrácený volanou funkcí „*get\_book.php*“.

```

int readFromDB(String PATH_NAME, String queryString, String &payload) {
    // funkce, která se připojí na server a získá odpověď
    HTTPClient http; // vytvoření objektu HTTPClient
    http.begin(HOST_NAME + PATH_NAME + queryString); // poskládání URL
    int httpCode = http.GET(); // proměnná pro návratový kód
    if (httpCode > 0) { // pokud se podaří navázat spojení
        if (httpCode == HTTP_CODE_OK) { // pokud je návratový kód 200
            payload = http.getString(); // uloží se odpověď do proměnné payload
        } else return httpCode; // vrátíme návratový kód v případě chyby
        // pokud se nepodaří navázat spojení, vypíšeme chybu na sériovou linku
    } else Serial.printf("error: %s\n", http.errorToString(httpCode).c_str());
    http.end(); // ukončení spojení
}

```

Úryvek kódu 11 - Odeslání dotazu do DB a přijetí odpovědi ve formátu JSON

Funkci *readFromDB(..)* volá například funkce *getID()*. Ta má pouze jeden vstupní parametr, a to ID záznamu v databázi. Právě ta se stará o předání všech parametrů předchozí funkci a vytvoření proměnné *result*, jejíž adresa se předává. Pokud volaná funkce *readFromDB(..)* skončí úspěchem, tedy návratovým kódem 200, funkce *getID()* přijatý JSON řetězec zpracuje. Nejprve je string *result* převeden na ukazatel na znak [79]. Následně je JSON deserializován na proměnnou *name* a *stock*, tedy jsou ze stringu získány údaje o názvu



a počtu kusů knihy skladem s využitím funkce „*deserializeJson()*“ z knihovny „*ArduinoJson.h*“ [80].

```
void getID(String value) {
  // funkce, která volá předchozí readFromDB() a zpracovává JSON
  // jako vstupní parametr přijímá ID knihy, předané volající funkcí
  // výsledek zpracování je uložen do globálních proměnných name a stock
  String a = "get_book.php"; // cesta k PHP skriptu
  String b = "?id=" + value; // parametr pro PHP skript
  String result = ""; // proměnná pro uložení odpovědi
  readFromDB(a, b, result); // volání funkce readFromDB()

  StaticJsonDocument<200> doc; // vytvoření objektu pro zpracování JSON
  const char* json = result.c_str(); // převedení odpovědi na char*
  DeserializationError error = deserializeJson(doc, json); // zpracování JSON
  if (error) { // pokud dojde k chybě, vypíšeme ji na sériovou linku
    Serial.print(F("deserializeJson() failed: "));
    Serial.println(error.f_str());
    return;
  }
  String title = doc["title"]; // získání názvu knihy z JSON
  int on_stock = doc["on_stock"]; // získání počtu kusů na skladě z JSON
  name = title; // uložení názvu knihy do globální proměnné
  stock = on_stock; // uložení počtu kusů na skladě do globální proměnné
  result = ""; // vynulování proměnné pro uložení odpovědi
}
```

Úryvek kódu 12 - Ukázka funkce *getID()*, která získá JSON s informacemi o knize a vytěží z něj jednotlivé proměnné

Funkci „*getID()*“ volají funkce „*getQR()*“ a „*getRFID()*“, které jsou téměř identické, jako samotná funkce „*getID()*“, ale jejich vstupem je údaj od skeneru RFID nebo QR. Po získání jednoho z těchto údajů se pomocí PHP skriptu „*qr\_look\_for\_book.php*“ dotáží v databázi na ID, které patří danému kódu. Pokud je ID nalezeno, předá se funkci „*getID()*“, v opačném případě je volána další funkce pro vytvoření nového záznamu. Celkem tedy pro práci s databází na ESP32 funguje osm následujících funkcí:

- **readFromDB** – funkce získá z databáze informace o názvu knihy a jejím počtu na skladě
- **getID** – funkce popsaná v předchozím odstavci, pomocí ID daného záznamu získává metodou GET ze serveru informace o určité knize.
- **insertToDB** – používá HTTP metodu POST a na server odesílá JSON objekt. V tomto případě přidává nový záznam do databáze, a to ID kód RFID tagu nebo čárového kódu.

- **updateDB** – funkce opět využívá metody POST a upravuje počet záznamů daného objektu v databázi.
- **addQR** a **addRFID** – změní počet kusů skladem v databázi o daný počet kusů voláním funkce *updateDB()*.
- **getQR** a **getRFID** – funkce v databázi přiřadí kódu získaného skenerem příslušné ID řádku, nebo volají funkci „*insertToDB()*“ pro vytvoření nového.

Na serveru je celkově pět PHP skriptů starajících se o HTTP dotazy, které jsou volány funkcemi z ESP32. Jejich seznam je následující:

- **updatebook.php** – skript má na starosti aktualizovat údaje již v databázi uložené.
- **rfid\_look\_for\_book.php** a **qr\_look\_for\_book.php** – skript dostane ID číslo RFID nebo čárového kódu a vrátí ID záznamu v databázi.
- **insert\_book.php** – skript vytváří v databázi nový záznam.
- **get\_book.php** – získává z databáze údaje o knize.

### 4.3 Testování

Po naprogramování bylo víko ochranného boxu napevno přišroubováno k horní části čtyřmi šrouby. Poté bylo zařízení používáno ve zkušebním provozu, paralelně spolu s papírovým deníkem z obrázku 1 k evidenci knih v rodinné knihovně. Během používání byly otestovány všechny požadavky na vlastnosti, a to přidávání a odebrání knih, aktualizace údajů a vyhledávání v záznamech. Testování probíhalo od března roku 2023 do začátku listopadu a bylo nezávislým testerem ohodnoceno jako použitelné. Jediná stížnost byla na ovládání. Na vině je rotační enkodér, který má tendence vynechávat skoky a potvrzení.

## 5 Výsledky a diskuse

Stavbě řešení předcházela teoretická příprava, kdy byly nastudovány parametry jednotlivých komponent. Inventarizační zařízení se podařilo sestavit a naprogramovat dle původního zadání. Výsledná podoba zařízení je zobrazena na obrázku 37. Zařízení je kompaktní a lehké, díky čemuž je manipulace s ním snadná. Zároveň kvalita konstrukce dovoluje i méně šetrné zacházení, ale pouze ve vnitřních prostorech. Zařízení není nijak chráněno proti vodě, která by mohla způsobit zkrat nebo oxidaci komponent. Také zvolený filament PLA je citlivý na sluneční paprsky a vyšší teploty. Proto je důležité zařízení takovýmto podmínkám nevystavovat po delší dobu. Náklady na pořízení komponent, spojovací a tiskový materiál byly celkově 60 €. Roční náklady na webhosting jsou 40 €. Jak již bylo zmíněno v kapitole 4.1.2 mezi cenově nejpříznivější technologií pro inventarizaci patří zařízení od společnosti Stormware. Zde se cena za licenci programu pro jeden počítač rovná 350 € a k němu kompatibilní skener se na trhu vyskytuje za ceny od 200 € [10]. Nejnižší možná cena komerčního inventarizačního řešení je v porovnání s výsledným inventarizačním zařízením uvedeným v této diplomové práci pětkrát vyšší.

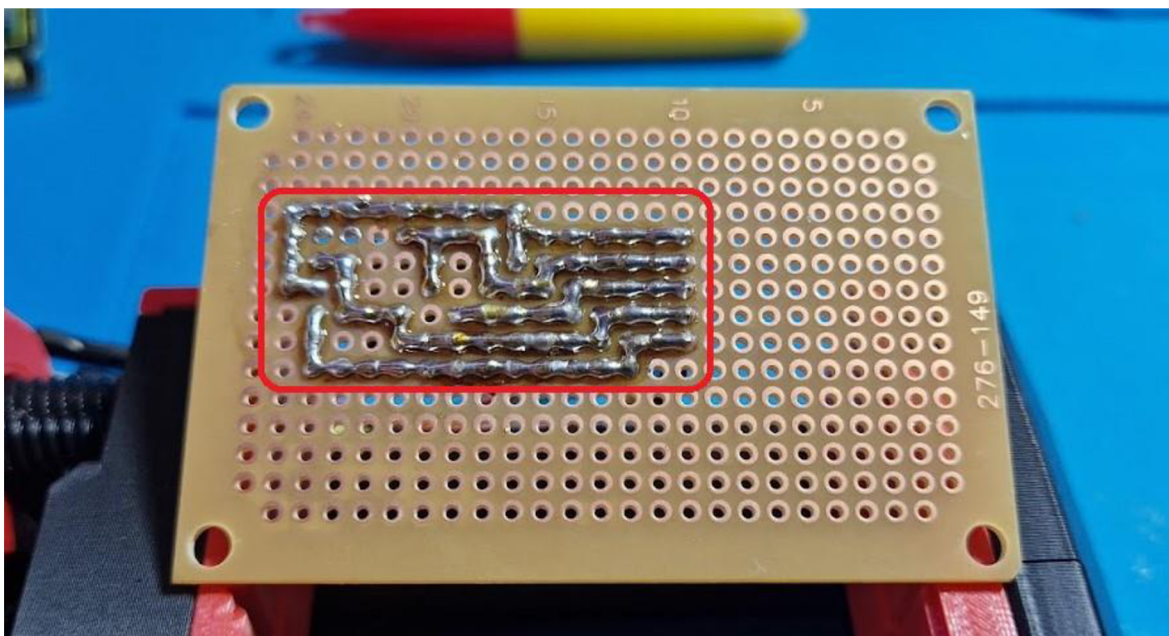


Obrázek 37 - Finální vzhled inventarizačního zařízení určeného k domácí inventarizaci. Pohled shora a čelní strany, kde se nachází skener čárových kódů GM65

Pro projekt je stěžejní hlavní řídicí a výpočetní jednotka ESP32, která díky svým 38 pinům poskytla dostatek prostoru pro připojení veškerých periférií. Volných pinů na ESP32 zbylo třináct a je zde tedy prostor na budoucí rozšíření o další funkce, například přidání více ovládacích prvků nebo klávesnice a většího displeje.

Zabudovaný akumulátor, skládající se ze dvou Li-Ion článků 18650 má celkovou kapacitu 2600 mAh a jmenovité napětí 7,4 V. To se ukázalo jako dostatečné, jelikož akumulátor zařízení byl dobý na plnou kapacitu koncem března roku 2023. Do začátku listopadu kleslo procento nabití z úrovně 100 %, tedy čtyř svítících indikačních LED diod na úroveň mezi 75 % až 99 %, tedy tři svítící stavové LED diody. Kapacita se ukázala jako naprosto dostatečná pro úroveň využití, které zařízení během testování mělo.

Během testování se projevil problém s ovládním – rotačním enkodérem (obrázek 38). Ve firmwaru zařízení tedy byla zvýšena citlivost a zároveň bylo implementováno hardwarové přerušení. To ovšem problém nevyřešilo. Nejspíše se tedy jedná o vadný díl. Možné řešení by bylo nahradit domácí modul enkodéru modulem sériově vyráběným, nebo se pokusit vyrobit nový s jinými parametry kondenzátorů a rezistorů. Oprava stávajícího není možná, jelikož je vyrobená součástka velice malá a rezistory spolu s kapacitami jsou umístěny pod rotačním enkodérem.

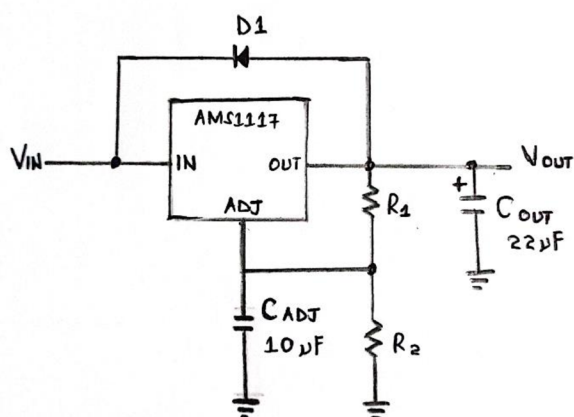


Obrázek 38 - Spodní strana pájivého pole vlastnoručně vyrobené součástky rotačního enkodéru. Červený obrys ukazuje řez, kterým se odstranil přebytečný materiál před instalací do zařízení

Webová aplikace funguje díky použitému datagridu Lazy Mofo velice spolehlivě a kombinaci s poskytovatelem webhostingu Wedos zatím nebyly objeveny žádné nedostatky. Občasné výpadky má webová aplikace na správu databází phpMyAdmin ve verzi 3.5.8.2, kterou má Wedos nasazenou, ale to se nijak neodráží na používání zařízení a ovlivňuje to pouze vývoj.

Problematickou částí práce byl 3D tisk. Samotné prototypování probíhalo v pořádku, avšak při tisku poslední verze se na tiskárně Průša MINI+ poškodily kabely topného bloku. Po výměně výhřevu hotendu však tiskárna hlásila další chybu, a to vadný termistor. Finální výtisk tedy proběhl na tiskárně Průša MK3S s menším zpožděním. Za vadu nejspíše mohla únava materiálu.

Sestavování prototypu proběhlo bez problému, ale k napájení 5V součástek se muselo použít Arduino. Bohužel při analýze vlastností ESP32 bylo přehlédnuto, že má pouze jeden step-down modul, a to jen 3,3V. To je oproti deskám z rodiny Arduino jediný nedostatek. Do verze finální, kde se díly k sobě pájely, nepřipadalo využití Arduina NANO jako pouhého měniče napětí v úvahu, zabíralo totiž zbytečně moc prostoru. Byl tedy odpájen stabilizátor AMS1117 5.0 ze staršího zkratovaného Arduina UNO, kde poškozený byl pouze 3,3V stabilizátor. 5V stabilizátor fungoval v pořádku a byl vytvořen modul podle schématu na obrázku 39. Hodnoty odporů a kondenzátorů byly určeny na základě dokumentace výrobce [81] a příspěvku na blogu WELLPCB [82].



Obrázek 39 - Schéma obvodu lineárního regulátoru AMS1117 [82]

Po zapojení akumulátoru bylo celé zařízení připraveno k instalaci do ochranného boxu, zbývalo jen preventivně omotat konektory izolační páskou. Celé zařízení bylo položeno na latexové nevodivé podložce, která je vhodná k pájení, jelikož nevede teplo. Bohužel

se na ní nacházely odštířky z nožiček rezistorů a LED diody, které zkratovaly mikrokontrolér ESP32. Do řešení bylo nutné použít verzi s 38 piny, kvůli specifickému regulátoru napětí.

## 6 Závěr

Cílem této diplomové práce bylo vytvoření mobilního inventarizačního zařízení určeného pro domácnosti. Výstupem práce je přenosná čtečka RFID Mifare štítků, čárových 1D a 2D QR kódů. Zařízení komunikuje se serverem, na kterém jsou v databázi uloženy všechny záznamy. Inventarizační zařízení je schopné záznamy do databáze přidávat a aktualizovat jejich údaje. Využívá k tomu připojení přes bezdrátovou WiFi síť. K obsluze záznamů lze využít také doprovodnou webovou aplikaci dostupnou na webu <https://dp.sebesta.fun>. Z té lze spravovat všechny záznamy v databázi uložené.

Ke splnění cílů byly využity teoretické předpoklady zjištěné v první části práce. Byl podrobně popsán mikrokontrolér, včetně ukázek několika komerčních řešení, která jsou běžně dostupná a populární mezi vývojáři hardwaru. Dále bylo vysvětleno, co je to webový server a zjištěno, jaké jsou výhody a náklady provozování vlastního serveru, oproti využití možnosti pronájmu serveru nebo místa na serveru. Další kapitolou v teoretické části práce byla inventarizace, kde byly definovány pojmy a vysvětlen základní princip tohoto procesu. Také byly uvedeny příklady nástrojů, které se v evidenci majetku využívají, a to čárové a QR kódy a RFID technologie. Poslední kapitolou teoretické části bylo představení aditivní technologie výroby a její srovnání se subtraktivní metodou. Byly popsány nejběžnější typy 3D tisku, včetně uvedení detailů tiskových materiálů.

V praktické části práce byl nejprve vytvořen návrh zařízení pro inventarizaci a byly definovány funkce, které má zařízení splňovat. Po výběru vhodných komponent byl vytvořen první grafický návrh vytvořený v modelovacím softwaru Fusion 360, který sloužil jako předpoklad pro následný vývoj 3D modelu určeného k tisku na FDM tiskárně.

V následujících kapitolách byly podrobně popsány všechny kroky vedoucí k sestavení zařízení. Nejdříve byl vytvořen model s využitím nepájivého pole, kde byly všechny součástky otestovány a byla ověřena jejich vzájemná kompatibilita. Poté byly jednotlivé komponenty pevně propojeny za využití pájky a pájecí stanice.

V momentě, kdy bylo celé řešení zkompletované, byly jeho celkové rozměry přeměřeny a v bezplatném softwaru AUTODESK Tinkercad byl vytvořen přesný 3D model, určený

k tisku. Model se musel specializovaným softwarem zvaným slicer připravit pro 3D tiskárnu do sady příkazů pro tiskovou hlavu ve formátu GCODE. Po výtisku několika prototypů a úpravě modelu byl vytištěn finální výtisk, do kterého bylo spájené řešení nainstalováno.

Poté byla vytvořena programová vybava pro mikrokontrolér ESP32, obsluhující veškeré komponenty. Dále byla napsána databáze, na kterou se napojila webová aplikace nad ní pracující. Aplikace byla napsána s využitím hotové datagridové aplikace Lazy Mofo podporující CRUD. V PHP s pomocí frameworku PDO byla napsána sada skriptů, které se starají o úpravu dat v databázi. Tyto skripty jsou volány pomocí HTTP metod mikrokontrolérem ESP32. Inventarizační zařízení tak získalo možnost pracovat se záznamy v online databázi. Celý projekt byl následně více než půl roku úspěšně používán pro správu domácí knihovny. Tímto byly splněny všechny cíle diplomové práce.



## 7 Seznam použitých zdrojů

- [1] MARTÍNKOVÁ, Michaela. Inventarizace – na co bychom neměli zapomenout? *Na co bychom neměli zapomenout? - Portál POHODA* [online]. 2016 [cit. 2023-11-13]. Dostupné z: <https://portal.pohoda.cz/dane-ucetnictvi-mzdy/ucetnictvi/inventarizace-%E2%80%93-na-co-bychom-nemeli-zapomenout/>
- [2] SILVER, Edward A, David F PYKE a Douglas J THOMAS. *Inventory and production management in supply chains*. 4. London, England: CRC Press, 2021. ISBN 9781032179322.
- [3] FISHMAN, Charles. American Way Mag. *American Way Mag* [online]. 2001 [cit. 2023-11-13]. Dostupné z: <https://www.americanwaymag.com/the-killer-app-bar-none/>
- [4] LAURER, George J. U.P.C. Resume. *Nenalezený vydavatel* [online]. 2001 [cit. 2023-11-13]. Dostupné z: [https://web.archive.org/web/20080925105745/http://bellsouthpwp.net/l/a/laurergj/UPC/upc\\_work.html](https://web.archive.org/web/20080925105745/http://bellsouthpwp.net/l/a/laurergj/UPC/upc_work.html)
- [5] BROOK, Benedict. It's the barcode's 43rd birthday. *News* [online]. [cit. 2023-11-13]. Dostupné z: <https://www.news.com.au/technology/innovation/inventions/its-the-barcodes-43rd-birthday/news-story/1ef1f692ab48a7bb3cdbcb66dda9e84d>
- [6] Information capacity and versions of the QR Code. *DENSO WAVE* [online]. [cit. 2023-11-13]. Dostupné z: <https://www.qrcode.com/en/about/version.html>
- [7] HAYES, Adam. Radio Frequency Identification (RFID): What It Is, How It Works. *Investopedia* [online]. [cit. 2023-11-13]. Dostupné z: <https://www.investopedia.com/terms/r/radio-frequency-identification-rfid.asp>
- [8] KINGSLEY-HUGHES, Adrian. 7 cool and useful things to do with your Flipper Zero. *ZDNET* [online]. [cit. 2023-11-14]. Dostupné z: <https://www.zdnet.com/article/7-cool-and-useful-things-to-do-with-your-flipper-zero/>
- [9] STORMWARE. E-shop STORMWARE. *Nenalezený vydavatel* [online]. 2023 [cit. 2023-11-23]. Dostupné z: <https://www.stormware.cz/zc/eshop.aspx>
- [10] A.S., Alza. Honeywell Voyager XP 1472g, všesměrový, 2D, černý, USB, BT. *Alza.cz* [online]. 2023 [cit. 2023-11-24]. Dostupné z: <https://www.alza.cz/honeywell-voyager-xp-1472g-vsesmerovy-2d-cerny-usb-bt-d5786782.htm>
- [11] NEXTSOFT SPOL. S R. O. RFID Inventarizace. *RFID Inventarizace* [online]. 2022 [cit. 2023-11-24]. Dostupné z: <https://www.nextsoft.cz/inventory.php?lang=cs>
- [12] BARR, Michael a Anthony J. MASSA. *Programming embedded systems: with C and GNU development tools*. 2nd ed. Sebastopol: O'Reilly, 2006. ISBN 978-0-596-00983-0.
- [13] DÜKING, Peter, Andreas HOTH, Hans-Christer HOLMBERG, Franz Konstantin FUSS a Billy SPERLICH. Comparison of Non-Invasive Individual Monitoring of the Training and Health of Athletes with Commercially Available Wearable Technologies. *Frontiers in Physiology* [online]. 2016, 2016-03-09, 7 [cit. 2023-11-14]. ISSN 1664-042X. Dostupné z: [doi:10.3389/fphys.2016.00071](https://doi.org/10.3389/fphys.2016.00071)

- [14] AYYUB, Ibrar. How Microcontrollers Work. *Projects Tutorials Code Library for Atmels Atmega32 AVR* [online]. 2013 [cit. 2023-11-11]. Dostupné z: <https://atmega32-avr.com/how-microcontrollers-work/>
- [15] MicroPython. *Python for microcontrollers* [online]. 2023 [cit. 2023-11-14]. Dostupné z: <http://micropython.org/>
- [16] BARRAGÁN, Hernando. *Wiring: Prototyping Physical Interaction Design* [online]. 2003 [cit. 2023-11-16]. Dostupné z: [https://people.interactionivrea.org/h.barragan/thesis/thesis\\_low\\_res.pdf](https://people.interactionivrea.org/h.barragan/thesis/thesis_low_res.pdf)
- [17] SKOROBOGATOV, Sergei. Copy Protection in Modern Microcontrollers. *University of Cambridge - Department of Computer Science and Technology* [online]. 2017 [cit. 2023-10-30]. Dostupné z: [https://www.cl.cam.ac.uk/~sps32/mcu\\_lock.html](https://www.cl.cam.ac.uk/~sps32/mcu_lock.html)
- [18] SKOROBOGATOV, Sergei. *Low temperature data remanence in static RAM*. 2002. Dostupné z: doi:10.48456/TR-536
- [19] What is Arduino? *Arduino* [online]. 2018 [cit. 2023-11-11]. Dostupné z: <https://www.arduino.cc/en/Guide/Introduction>
- [20] SIMON. *Programming arduino: Getting started with sketches, second edition: Getting started with sketches, second edition. 2*. Columbus, OH: McGraw-Hill Education, 2016. ISBN 9781259641633.
- [21] Arduino Uno Rev3. *Arduino Online Shop* [online]. 2022 [cit. 2023-11-11]. Dostupné z: <https://store-usa.arduino.cc/products/arduino-uno-rev3>
- [22] SHIFFMAN, Daniel. Interview with Casey Reas and Ben Fry. *Rhizome* [online]. 2009 [cit. 2023-11-11]. Dostupné z: <https://rhizome.org/editorial/2009/sep/23/interview-with-casey-reas-and-ben-fry/>
- [23] VODA, Zbyšek. *Průvodce světem Arduina*. Vydání druhé. Bučovice: Martin Stříž, 2017. ISBN 978-80-87106-93-8.
- [24] Arduino Hardware. *Arduino* [online]. 2022 [cit. 2023-11-11]. Dostupné z: <https://www.arduino.cc/en/hardware>
- [25] KASHIF, Masiv. Does Arduino Uno Have a Voltage Regulator. *Nenalezený vydavatel* [online]. 2022 [cit. 2023-11-11]. Dostupné z: <https://linuxhint.com/arduino-uno-voltage-regulator/>
- [26] MONK, Simon. *TAB Book of Arduino Projects: 36 Things to Make with Shields and Proto Shields*. New York: McGraw-Hill Education Tab, 2014. ISBN 978-0071790673.
- [27] ERBIR, Muhammed Ali a Halil ÜNVER. Control of A Car, Via Internet. *Uluslararası Muhendislik Arastirma ve Gelistirme Dergisi*. 2018/06/29, 252-258. Dostupné z: doi:10.29137/umagd.446803
- [28] *Espressif Systems* [online]. [cit. 2023-11-12].
- [29] HÜBSCHMANN, Ida. ESP32 for IoT: A Complete Guide. *Nabto* [online]. [cit. 2023-11-12]. Dostupné z: <https://www.nabto.com/guide-to-iot-esp-32/>
- [30] BABIUCH, Marek, Petr FOLTYNEK a Pavel SMUTNY. Using the ESP32 Microcontroller for Data Processing. In: *2019 20th International Carpathian Control Conference (ICCC)* [online]. IEEE, 2019, s. 1-6 [cit. 2023-11-12]. ISBN 978-1-7281-0702-8. Dostupné z: doi:10.1109/CarpathianCC.2019.8765944

- [31] KURNIAWAN, Agus. *Internet of Things Projects with ESP32: Build exciting and powerful IoT projects using the all-new Espressif ESP32: Build exciting and powerful IoT projects using the all-new Espressif ESP32*. Birmingham, England: Packt Publishing, 2019. ISBN 9781789956870.
- [32] ESP32-DevKitC V4 Getting Started Guide. *ESP-IDF Programming Guide latest documentation* [online]. 2020 [cit. 2023-11-12]. Dostupné z: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/hw-reference/esp32/get-started-devkitc.html>
- [33] ESPRESSIF. Search code, repositories, users, issues, pull requests.. *GitHub* [online]. 2016 [cit. 2023-11-12]. Dostupné z: <https://github.com/espressif/esp-idf>
- [34] MAIER, Alexander, Andrew SHARP a Yuriy VAGAPOV. Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things. In: *2017 Internet Technologies and Applications (ITA)* [online]. IEEE, 2017, s. 143-148 [cit. 2023-11-12]. ISBN 978-1-5090-4815-1. Dostupné z: doi:10.1109/ITECHA.2017.8101926
- [35] NELSON, William, Arvind SRINIVASAN a Murthy CHINTALAPATI. *Sun web server: The essential guide: The essential guide*. Philadelphia, PA: Prentice Hall, 2009. ISBN 9780137128921.
- [36] FIELDING, Roy, Mark NOTTINGHAM a Julian RESCHKE. *HTTP Semantics*. Berlin, 2022. Dostupné z: doi:10.17487/rfc9110
- [37] HTTP response status codes. *MDN Web Docs* [online]. [cit. 2023-11-14]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>
- [38] HAMMAD, Zahid. How to Set up the Apache Web Server on the Raspberry Pi. *Nenalezený vydavatel* [online]. 2021 [cit. 2023-11-14]. Dostupné z: <https://linuxhint.com/set-up-apache-web-server-raspberry-pi/>
- [39] ADAMANT ISP. What are the differences between renting and buying a server? *Adamant Company* [online]. 2023 [cit. 2023-11-14]. Dostupné z: <https://adamant.ua/en/blog/what-are-the-differences-between-renting-and-buying-a-server>
- [40] CRABB, Kristin. Shared Hosting vs. Dedicated Hosting. *Domain.com | Blog* [online]. 2023 [cit. 2023-11-14]. Dostupné z: <https://www.domain.com/blog/shared-hosting-vs-dedicated-hosting/>
- [41] S.R.O., Cybersoft. Servedy. *Czech-Server* [online]. 2023 [cit. 2023-11-14]. Dostupné z: [https://www.czech-server.cz/servedy\\_c8764596.html](https://www.czech-server.cz/servedy_c8764596.html)
- [42] HETZNER, Online GmbH. MANAGED SERVERS. *Worry-free focus on your projects* [online]. 2023 [cit. 2023-11-14]. Dostupné z: <https://www.hetzner.com/managed-server>
- [43] GIBSON, Ian, David ROSEN a Brent STUCKER. *Additive manufacturing technologies: 3D printing, rapid prototyping, and direct digital manufacturing*. Second edition. New York: Springer, [2015]. ISBN 978-1493921126.
- [44] NOORANI, Rafiq. *3D printing: Technology, applications, and selection: Technology, applications, and selection*. London, England: CRC Press, 2021. ISBN 9781498783750. Dostupné z: doi:10.1201/9781315155494
- [45] DOE, Jamie. 3D Printing vs CNC Machining: Which is best for prototyping? MARTEL, Alexandre. *3Dnatives* [online]. 2018 [cit. 2023-11-01]. Dostupné z: <https://www.3dnatives.com/en/3d-printing-vs-cnc-160320184/>

- [46] ZUZA, Mikolas. Vrstvy a perimetry. *Prusa Knowledge Base* [online]. 2022 [cit. 2023-11-11]. Dostupné z: [https://help.prusa3d.com/article/vrstvy-a-perimetry\\_1748](https://help.prusa3d.com/article/vrstvy-a-perimetry_1748)
- [47] NARAYAN, K. Lalit, K. Mallikarjuna RAO a M.M.M. SARCAR. *Computer Aided Design and Manufacturing*. PHI, 2008. ISBN 978-8120333420.
- [48] TURNEY, Drew. History of 3D Printing: It's Older Than You Think. *Autodesk - Design & Make* [online]. 2021 [cit. 2023-11-01]. Dostupné z: <https://www.autodesk.com/design-make/articles/history-of-3d-printing>
- [49] CAREW, Rachael M. a David ERRICKSON. An Overview of 3D Printing in Forensic Science: The Tangible Third-Dimension. *Journal of Forensic Sciences* [online]. 2020, 65(5), 1752-1760 [cit. 2023-11-01]. ISSN 0022-1198. Dostupné z: doi:10.1111/1556-4029.14442
- [50] What is 3D Printer Filament? *Raise3D: Reliable, Industrial Grade 3D Printer* [online]. 2022 [cit. 2023-11-14]. Dostupné z: <https://www.raise3d.com/academy/what-is-3d-printer-filament/>
- [51] KITCHEN, CNC. *I tried composting "biodegradable" 3D prints!*. 2022. Dostupné také z: <https://www.youtube.com/watch?v=tavrkWrazWI>
- [52] CONRADIE, Danie. HACKADAY. PETBOT: TURN PET BOTTLES INTO FILAMENT. *Hackaday* [online]. 2021 [cit. 2023-11-14]. Dostupné z: <https://hackaday.com/2021/06/29/petbot-turn-pet-bottles-into-filament/>
- [53] PRŮŠA, Josef. Průvodce materiály. *Prusa Knowledge Base* [online]. [cit. 2023-11-14]. Dostupné z: [https://help.prusa3d.com/category/pruvodce-materialy\\_220](https://help.prusa3d.com/category/pruvodce-materialy_220)
- [54] PRUSA RESEARCH. Flexibilní materiály. *Prusa Knowledge Base* [online]. 2023 [cit. 2023-11-14]. Dostupné z: [https://help.prusa3d.com/article/flexibilni-materialy\\_2057](https://help.prusa3d.com/article/flexibilni-materialy_2057)
- [55] Prusa Tabulka Materiálů. *Prusa Knowledge Base* [online]. [cit. 2023-11-14]. Dostupné z: [https://help.prusa3d.com/materials?\\_gl=1\\*1x11szh\\*\\_ga\\*MTEyNzE0MzcwOC4xNjk4Njk4MTYw\\*\\_ga\\_3HK7B7RT5V\\*MTY5OTk2Njk5Ny41LjEuMTY5OTk2NzA1Mi4wLjAuMA..](https://help.prusa3d.com/materials?_gl=1*1x11szh*_ga*MTEyNzE0MzcwOC4xNjk4Njk4MTYw*_ga_3HK7B7RT5V*MTY5OTk2Njk5Ny41LjEuMTY5OTk2NzA1Mi4wLjAuMA..)
- [56] ESPRESSIF. Search code, repositories, users, issues, pull requests.. *GitHub* [online]. 2023 [cit. 2023-11-15]. Dostupné z: <https://github.com/espressif/esp-idf/blob/c8243465e45489835d645bf217a6929fd0c01b7f/docs/en/api-reference/system/ota.rst>
- [57] ESPRESSIF SYSTEMS. ESP32 WROOM 32 Datasheet. *Nenalezený vydavatel* [online]. 2023 [cit. 2023-11-15]. Dostupné z: [https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf)
- [58] MICHÁLEK, Luboš. Převodník / programátor USB-UART CP2102. *Návody Drátek* [online]. 2021 [cit. 2023-11-15]. Dostupné z: <https://navody.dratek.cz/navody-k-produktum/prevodnik-programator-usb-uart-cp2102.html>
- [59] LASKAKIT. Samsung ICR18650-26J 2600mAh 18650 Li-Ion Baterie. *Laskakit.cz* [online]. 2023 [cit. 2023-11-15]. Dostupné z: <https://www.laskakit.cz/2600mah-samsung-icr18650-26j-18650-li-ion-baterie/>
- [60] LUCIFER. Technika. *Špičkové li-ion baterie* [online]. 2021 [cit. 2023-11-15]. Dostupné z: <https://luciferlights.net/cs/spickove-li-ion>

- [61] LITTLE BIRD ELECTRONICS. CM65 QR Barcode Scanner Module [DF-DFR0660]. *Little Bird* [online]. 2023 [cit. 2023-11-15]. Dostupné z: <https://littlebirdelectronics.com.au/products/cm65-qr-barcode-scanner-module>
- [62] MFRC522 data sheet. *NXP Semiconductors* [online]. [cit. 2023-11-15]. Dostupné z: <https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf>
- [63] LITTLE BIRD ELECTRONICS. Mifare RC522 Card Read Antenna RF Module RFID Reader IC Card Proximity Module [SU-RC522CARDREAD]. *Little Bird* [online]. [cit. 2023-11-15]. Dostupné z: <https://littlebirdelectronics.com.au/products/mifare-rc522-card-read-antenna-rf-module-rfid-reader-ic-card-proximity-module>
- [64] AUDIOPHONICS. White OLED 1.3" Screen 128X64 SH1106 interface I2C. *Audiophonics* [online]. [cit. 2023-11-16]. Dostupné z: <https://www.audiophonics.fr/en/screens-vu-meters/white-oled-13-screen-128x64-sh1106-interface-i2c--p-11573.html>
- [65] LASKAKIT. Stabilizátor AMS1117 5.0 0.8A, SOT-223. *Laskakit.cz* [online]. [cit. 2023-11-15]. Dostupné z: [https://www.laskakit.cz/stabilizator-ams1117-5-0-0-8a--sot-223/?gad\\_source=1&gclid=CjwKCAiA9dGqBhAqEiwAmRpTC5-vqB6a8SEFGVL2cBKXStdN\\_LjW5WJvnXhtz9lZr1ErUcJjHu3vLBoCzxwQAvD\\_BwE](https://www.laskakit.cz/stabilizator-ams1117-5-0-0-8a--sot-223/?gad_source=1&gclid=CjwKCAiA9dGqBhAqEiwAmRpTC5-vqB6a8SEFGVL2cBKXStdN_LjW5WJvnXhtz9lZr1ErUcJjHu3vLBoCzxwQAvD_BwE)
- [66] ENJOYNEERING. Search code, repositories, users, issues, pull requests.. *Enjoyneering/RotaryEncoder: This is small and fast Arduino library for Rotary Encoder with interrupts* [online]. 2018 [cit. 2023-11-16]. Dostupné z: <https://github.com/enjoyneering/RotaryEncoder>
- [67] ADAFRUIT. Search code, repositories, users, issues, pull requests.. *GitHub* [online]. [cit. 2023-11-16]. Dostupné z: <https://github.com/adafruit/Adafruit-GFX-Library>
- [68] WONHO-MAKER. Search code, repositories, users, issues, pull requests.. *GitHub* [online]. [cit. 2023-11-16]. Dostupné z: [https://github.com/wonho-maker/Adafruit\\_SH1106](https://github.com/wonho-maker/Adafruit_SH1106)
- [69] ADAFRUIT. Search code, repositories, users, issues, pull requests.. *GitHub* [online]. [cit. 2023-11-16]. Dostupné z: [https://github.com/adafruit/Adafruit\\_SSD1306](https://github.com/adafruit/Adafruit_SSD1306)
- [70] ADAFRUIT. Search code, repositories, users, issues, pull requests.. *GitHub* [online]. [cit. 2023-11-16]. Dostupné z: [https://github.com/adafruit/Adafruit\\_SH110x](https://github.com/adafruit/Adafruit_SH110x)
- [71] SoftwareSerial Library. *Arduino Documentation* [online]. [cit. 2023-11-17]. Dostupné z: <https://docs.arduino.cc/learn/built-in-libraries/software-serial>
- [72] G6EJD. Search code, repositories, users, issues, pull requests.. *GitHub* [online]. [cit. 2023-11-17]. Dostupné z: [https://github.com/G6EJD/ESP32-Using-Hardware-Serial-Ports/blob/master/ESP32\\_Using\\_Serial2.ino](https://github.com/G6EJD/ESP32-Using-Hardware-Serial-Ports/blob/master/ESP32_Using_Serial2.ino)
- [73] MIGUELBALBOA. MFRC522. *GitHub* [online]. 2016 [cit. 2023-11-17]. Dostupné z: <https://github.com/miguelbalboa/rfid>
- [74] MA, Shuai, Modi JIANG, Peng TAO, Chengyi SONG, Jianbo WU, Jun WANG, Tao DENG a Wen SHANG. Temperature effect and thermal impact in lithium-ion batteries: A review: A review. *Prog. Nat. Sci.* 2018, 28(6), 653-666. ISSN 1002-0071. Dostupné z: doi:10.1016/j.pnsc.2018.11.002

- [75] PODRZAJ, Primoz a Samo SIMONCIC. Temperature measurement during resistance spot welding using a digital camera. In: *2013 IEEE International Conference on Mechatronics and Automation* [online]. IEEE, 2013, s. 1305-1310 [cit. 2023-11-17]. ISBN 978-1-4673-5560-5. Dostupné z: doi:10.1109/ICMA.2013.6618102
- [76] PRANAV, G. Prusa Slicer 2.6 - Organic Supports, Embossing and More!. *Obico Knowledge Base* [online]. 2023 [cit. 2023-11-17]. Dostupné z: <https://www.obico.io/blog/prusa-slicer-2.6/>
- [77] LAZYMOFO. Search code, repositories, users, issues, pull requests.. *Lazymofodatagrid: PHP MySQL CRUD Datagrid* [online]. 2016 [cit. 2023-11-18]. Dostupné z: <https://github.com/lazymofodatagrid>
- [78] ARDUINO-LIBRARIES. Search code, repositories, users, issues, pull requests.. *GitHub* [online]. 2023 [cit. 2023-11-18]. Dostupné z: <https://github.com/arduino-libraries/WiFi>
- [79] SAVE, Alok. When to use const char \* and when to use const char []. *Stack Overflow* [online]. [cit. 2023-11-18]. Dostupné z: <https://stackoverflow.com/questions/7903551/when-to-use-const-char-and-when-to-use-const-char>
- [80] BBLANCHON. Search code, repositories, users, issues, pull requests.. *GitHub* [online]. 2023 [cit. 2023-11-18]. Dostupné z: <https://github.com/bblanchon/ArduinoJson>
- [81] ADVANCED MONOLITHIC SYSTEMS. 1A Adjustable/Fixed Low Dropout Linear Regulator. *Nenalezený vydavateľ* [online]. 2019, [http://www.ams-semitech.com/attachments/File/AMS1117\\_20120314.pdf](http://www.ams-semitech.com/attachments/File/AMS1117_20120314.pdf) [cit. 2023-11-19]. Dostupné z: [http://www.ams-semitech.com/attachments/File/AMS1117\\_20120314.pdf](http://www.ams-semitech.com/attachments/File/AMS1117_20120314.pdf)
- [82] ZHAO, Hommer. Ams1117 Pinout: The Best Guide for Beginners. *Circuit Board Fabrication and PCB Assembly Turnkey Services - WellPCB* [online]. [cit. 2023-11-19]. Dostupné z: <https://www.wellpcb.com/ams1117-Pinout.html>

## 8 Seznam obrázků, tabulek, grafů a zkratk

### 8.1 Seznam obrázků

Obrázek 1 – Ručně vedená inventarizační kniha o stavu rodinné domácí knihovny .....	14
Obrázek 2 - U.P.C. čárový kód .....	15
Obrázek 3 - Zleva: QR kód verze 1; obrázek generovaný AI (umělá inteligence) obsahující funkční QR kód; QR kód verze 40 .....	16
Obrázek 4 - Části mikrokontroléru [14] .....	18
Obrázek 5 - Deska Arduino UNO [21] .....	21
Obrázek 6 - Arduino MEGA s Ethernet shieldem [27] .....	22
Obrázek 7 – Oblíbené ESP32-DevKitC a jeho rozložení pinů [32] .....	24
Obrázek 8 - Ukázka komunikace mezi klientem a serverem.....	25
Obrázek 9 - Srovnání aditivní (nahore) a subtraktivní (dole) metody .....	29
Obrázek 10 - Náhled STL souboru (velvo) a GCODEu (vpravo) připraveného k tisku.....	30
Obrázek 11 - Technologie 3D tisku, zleva: FDM, SLA, SLS .....	32
Obrázek 12 - Ukázka procesu výroby vlastního filamentu recyklací PET lahví [52] .....	33
Obrázek 13 - Akumulátor Samsung 18650 s kapacitou 2600 mAh [59].....	37
Obrázek 14 - Čtečka 1D a 2D kódů GM65 [61] .....	38
Obrázek 15 - RFID čtečka RC522 [63] .....	39
Obrázek 16 - 1,3 palcový OLED display SH1106 [64].....	39
Obrázek 17 - Stabilizátor AMS1117 5.0 [65].....	40
Obrázek 18 – Předpokládaný model výsledného ochranného boxu pro inventarizační zařízení .....	41
Obrázek 19 - Zleva: Schématické zapojení rotačního enkodéru [66].....	42
Obrázek 20 - Zapojení 1.3 palcového I <sup>2</sup> C displeje k ESP 32 s využitím nepájivého pole ..	43
Obrázek 21 - Přidání rotačního enkodéru do nepájivého pole.....	45
Obrázek 22 - Schéma zapojení čtečky GM65 .....	46
Obrázek 23 - Schéma zapojení všech součástek do nepájivého pole, včetně čtečky karet RC522 a notifikační LED diody .....	48
Obrázek 24 - Finální schéma zapojení všech součástek, včetně napájecího obvodu a externího spínače pro čtečku čárových kódů .....	50
Obrázek 25 - Spojené akumulátory s přepětovou ochranou a balancérem umístěné v pájecím držáku vyrobeném s využitím FDM tiskárny.....	51
Obrázek 26 - Finální pevné zapojení všech komponent. Akumulátor vlevo, i všechny plochy, které mohou způsobit zkrat, je obalen ve felxibilní černé pásce .....	52
Obrázek 27 - Modely průřezů pro displej a čtečku čárových kódů GM65 vytvořené v aplikaci AUTOCAD Tinkercad .....	53
Obrázek 28 - Finální verze 6.0 ochranného boxu a víka pro inventarizační zařízení.....	54
Obrázek 29 - Rozlicovaný 3D model s organickými podpěrami a detailními údaji o průběhu tisku.....	55
Obrázek 30 - Nezdařilé a testovací výtisky (žluté) s viditelnými verzovacími čísly. ....	55
Obrázek 31 - Úvodní obrazovka inventarizačního zařízení s kurzorem na položce KNIHOVNA .....	56
Obrázek 32 - Obrazovka, na které je zařízení ve čtecím módu a čte kódy z RFID i GM65 čtečky .....	57
Obrázek 33 – Obrazovky na zařízení Inventarizátor .....	58
Obrázek 34 - Ukázka aplikace fungující nad tabulkou knihovna.....	60

Obrázek 35 - Formulář pro úpravu nebo přidání nového záznamu do databáze .....	61
Obrázek 36 - Ukázka zaslání GET požadavku na server přes API aplikaci Insomnia .....	63
Obrázek 37 - Finální vzhled inventarizačního zařízení určeného k domácí inventarizaci. Pohled shora a čelní strany, kde se nachází skener čárových kódů GM65.....	67
Obrázek 38 - Spodní strana pájivého pole vlastnoručně vyrobené součástky rotačního enkodéru. Červený obrys ukazuje řez, kterým se odstranil přebytečný materiál před instalací do zařízení.....	68
Obrázek 39 - Schéma obvodu lineárního regulátoru AMS1117 [82] .....	69

## 8.2 Seznam tabulek

Tabulka 1 - Srovnání tří variant provozu webového serveru [40], [41], [42].....	28
Tabulka 2 - Základní srovnání nejběžnějších technologií 3D tiskáren [49] .....	31
Tabulka 3 - Komparace základních vlastností běžně dostupných filamentů. U sloupců označených „*“ je rozsah 1=minimální až 10=maximální. [55].....	34
Tabulka 4 - Přiřazení výstupních pinů z RC522 k pinům na ESP32 .....	47

## 8.3 Seznam úryvků kódu

Úryvek kódu 1 – Základní kód v jazyce Wiring. Jeho nahrání do mikrokontroléru Arduino způsobí blikání zabudované LED diody v intervalu jedné sekundy .....	19
Úryvek kódu 2 - Kód pro zobrazení textu na displeji .....	44
Úryvek kódu 3 - Nejjednodušší kód pro čtení čárových kódů s využitím čtečky GM65 ....	47
Úryvek kódu 4 - Ukázka přečtení ID RFID karty a jeho vypsání v šestnáctkové soustavě na sériovou linku včetně funkce pro blikání LED diody blink() .....	49
Úryvek kódu 5 - Příkazy pro vytisknutí položek úvodního menu .....	56
Úryvek kódu 6 - Výzva uživatele k načtení kódu a volání funkcí pro jeho čtení .....	57
Úryvek kódu 7 - SQL příkaz k vytvoření tabulky books.....	59
Úryvek kódu 8 - Načtení dat z DB do aplikace LM, s využitím frameworku PDO .....	62
Úryvek kódu 9 - Skript na získání JSON stringu z databáze po předání ID knihy.....	63
Úryvek kódu 10 - Správa údajů a připojení k WiFi síti .....	64
Úryvek kódu 11 - Odeslání dotazu do DB a přijmutí odpovědi ve formátu JSON .....	64
Úryvek kódu 12 - Ukázka funkce getID(), která získá JSON s informacemi o knize a vytěží z něj jednotlivé proměnné.....	65



## 8.4 Seznam použitých zkratek

- AES – Advanced Encryption Standard (standard pokročilého šifrování)
- AI – Artificial Intelligence (umělá inteligence)
- AM – Additive Manufacturing (aditivní výroba)
- BLE – Bluetooth Low Energy (nízkoenergetické bluetooth)
- CAD – Computer Aided Design (počítačem podporované projektování)
- CNC – Computer Numerical Control (číslicové řízení pomocí počítače)
- CRUD – Create, Read, Update, Delete
- CSS – Cascading Style Sheets (kaskádové styly)
- CSV – Ultraviolet Comma Separated Values (hodnoty oddělené čárkou)
- DDoS – Denial of service (odepření služby)
- DNS – Domain Name Server (server doménových jmen)
- EAN – European Article Number (evropský artiklový kód)
- EEPROM – Electrically Erasable Programmable Read-Only Memory
- FDM – Fused Deposition Modeling (tavené depoziční modelování)
- GND – Ground (uzemnění)
- GPIO – General-Purpose Input/Output (univerzální vstupní/výstupní pin)
- HTML – Hypertext Markup Language (hypertextový značkovací jazyk)
- HTTP – Hypertext Transfer Protocol (hypertextový výměnný protokol)
- HTTPS – Hypertext Transfer Protocol Secure (šifrované HTTP)
- I/O – input/output (vstup/výstup)
- I<sup>2</sup>C – Inter-Integrated Circuit
- IDE – Integrated Development Environment (integrované vývojové prostředí)
- IP – Internet Protocol (internetový protokol)
- ISBN – International Standard Book Number (standartní číslo knihy)
- IoT – Internet of Things (internet věcí)
- JSON – JavaScript Object Notation (javaskriptová objektová notace)
- LED – Light Emitting Diode (světlo emitující dioda)
- LM – Lazy Mofo
- NFC – Near Field Communication (blízkopolní komunikace)
- OLED – Organic Light-Emitting Diode (organická LED dioda)
- OTA – Over the air (bezdrátové aktualizace)

PCB – Printed Circuit Board (deska plošných spojů)  
PD – Power Delivery (rychlý dobíjecí standard)  
PHP – Hypertext Preprocessor  
PLA – Polylactic Acid (kyselina polyléčná)  
PWM – Pulse Width Modulation (pulzně šířková modulace)  
QR – Quick Response (rychlá odpověď)  
RAM – Random Access Memory (možnost náhodného přístupu)  
RFID – Radio Frequency Identification (identifikace na rádiové frekvenci)  
RNG – Random Number Generator (generátor náhodných čísel)  
RSA – (iniciály autorů Rivest, Shamir, Adleman)  
RX – Receive (příjem)  
SDA – Serial Data (sériová data)  
SHA-2 – Secure Hash Algorithm 2 (podpisový algoritmus verze dvě)  
SLA – Stereolithography apparatus (stereolitografický aparát)  
SLC – Serial CLock (sériové hodiny)  
SLS – Selective Laser Sintering  
SM – Subtractive Manufacturing (subtraktivní výroba)  
SoC – System on Chip (systém na čipu)  
SPI – Serial Peripheral Interface (sériové periferní rozhraní)  
SRAM – Static Random Access Memory  
STL – Stereolithography (stereolitografie)  
TX – Transmission (volání)  
U.P.C. – Universal Product Code (univerzální produktový kód)  
UART – Universal Asynchronous Receiver Transmitter (asynchronní přijímač a vysílač)  
URL – Uniform Resource Locators (jednotný popis umístění zdroje)  
USB – Universal Serial Bus (univerzální sériová sběrnice)  
UV – Ultraviolet (ultrafialové)  
VPS – Virtual Private Server (virtuální privátní server)

## **Přílohy**

Na přiloženém CD se nachází zdrojové kódy webové aplikace, a to ve složce „*webaap*“. SQL skripty pro vytvoření databáze jsou umístěny ve složce *SQL*. Ve složce „*display*“ jsou umístěny zdrojové kódy pro naprogramování přípravku ESP32.