

České vysoké učení technické v Praze

Fakulta elektrotechnická

Katedra radioelektroniky



Bakalářská práce

Úschovné boxy se vzdáleným ovládáním a možností rezervace

Storage Boxes with Remote Control and Reservation Option

Autor: Ondřej Novotný

Vedoucí práce: doc. Ing. Stanislav Vítek PhD.

Studijní program: Elektronika a komunikace

Praha 2023



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Novotný** Jméno: **Ondřej** Osobní číslo: **476719**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra radioelektroniky**
Studijní program: **Elektronika a komunikace**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Úschovné boxy se vzdáleným ovládním a možností rezervace

Název bakalářské práce anglicky:

Storage Boxes with Remote Control and Reservation Option

Pokyny pro vypracování:

Cílem práce je návrh a implementace úschovného boxu s možností vzdálené konfigurace a rezervace registrovaným uživatelům. Úschovné boxy jsou vybaveny elektronickým zámekem ovládaným mikrokontrolérem (ESP32) s možností připojení do lokální sítě. Součástí řešení je uživatelské rozhraní, které umožňuje registraci a přihlášení uživatele. Registrovaný uživatel má možnost rezervace boxu v neobsazeném časovém intervalu. V práci bude diskutována bezpečnost řešení, možnosti zabezpečení a řešení výpadků konektivity.

Seznam doporučené literatury:

- [1] KURNIAWAN, Agus. Internet of Things Projects with ESP32: Build exciting and powerful IoT projects using the all-new Espressif ESP32. Packt Publishing Ltd, 2019.
- [2] GENG, Yeow Guo; FHONG, Soom Chin. Cloud-Controlled Parcel Storage Box with User Image Capturing. Evolution in Electrical and Electronic Engineering, 2022, 3.2: 59-68.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

doc. Ing. Stanislav Vítek, Ph.D. katedra radioelektroniky FEL

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

Datum zadání bakalářské práce: **15.02.2023** Termín odevzdání bakalářské práce: _____

Platnost zadání bakalářské práce: **22.09.2024**

doc. Ing. Stanislav Vítek, Ph.D.
podpis vedoucí(ho) práce

doc. Ing. Stanislav Vítek, Ph.D.
podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

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

Datum převzetí zadání

Podpis studenta

Prohlášení

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

V Praze dne

.....

Podpis autora práce

Poděkování

Rád bych poděkoval panu doc. Ing. Stanislavu Vítkovi, Ph.D. za trpělivost a důležité rady při tvorbě této práce. Děkuji také všem lidem v mém blízkém okruhu, kteří mě podporovali nejen při tvorbě této práce, ale i během celého studia.

Abstrakt

Práce se zabývá návrhem systému úschovných boxů a následným jeho vytvořením pro fungování v testovacím režimu. V rámci druhé kapitoly je představeno IoT a požadované fungování systému. Třetí kapitola se věnuje návrhu systému a rozebírá se problematika bezpečnosti jak mechanické, tak i elektronické. Výsledkem práce je vytvoření základního testovacího systému, který se následně nasadí. Testovací systém byl v rámci projektu naprogramován i byly vybrány a připraveny jeho fyzické prvky.

Klíčová slova: IoT, úschovný systém, úschovné boxy, ESP32

Abstract

The thesis is focused on the design of a system of storage boxes and creation of the subsystem for the test mode. The second chapter introduces the IoT and description of the system being developed. The third chapter explore the design of the system and discusses the security issues both mechanical and electronic. As a result of the thesis is creation of test system as proof of concept. The test system has been programmed as part of the project as well as its physical elements have been selected and prepared.

Keywords: IoT, storage system, storage box, ESP32

Seznam obrázků

Obrázek 1 - přehled architektury IoT[5]	11
Obrázek 2 - blokové schéma systému	15
Obrázek 3 - obrázek devkitu STM32 NUCLEO[27]	18
Obrázek 4 - devkit ESP32[29]	19
Obrázek 5 - reálné blokové schéma zapojení systému	24
Obrázek 6 - EPS32 použité v systému[35]	25
Obrázek 7 - použitý zdroj[36]	25
Obrázek 8 - zámek a relé modul[37][38]	26
Obrázek 9 - D-Link DWR-116[39]	26
Obrázek 10 - Raspberry Pi 4[41]	27
Obrázek 11 - rozložení webového rozhraní	28
Obrázek 12 - flowchart odemření skříňky	30

Obsah

1. Úvod.....	9
2. Obecný princip navrhovaného systému.....	10
2.1. Internet věcí – IoT.....	10
2.2. Používaná zařízení a technologie v IoT	11
2.2.1. Koncová zařízení.....	11
2.2.2. Koncentrátor/Gateway	12
2.2.3. Komunikace	12
2.2.4. Serverové komerční služby	13
2.3. Popis fungování systému úschovných boxů.....	14
3. Návrh systému úschovných boxů	15
3.1. Schéma systému úschovných boxů.....	15
3.2. Komponenty v systému úschovných boxů.....	15
3.2.1. Lokální server.....	15
3.2.2. Mikrokontroler	17
3.3. Bezpečnostní nároky na systém	19
3.3.1. Mechanická ochrana.....	19
3.3.2. Elektronická ochrana.....	20
3.3.3. Řešení mimořádné události výpadku zdroje energie.....	22
4. Realizace systému úschovných boxů.....	24
4.1. Reálné zapojení systému.....	24
4.2. Použitá zařízení.....	24
4.2.1. Mikrokontroler	24
4.2.2. Zdroj napájení	25
4.2.3. Zámek, jeho ovládání	25
4.2.4. Síťový prvek.....	26
4.2.5. Server	26
4.3. Vytvořený software.....	27
4.3.1. Webové prostředí	27
4.3.2. Server	28
4.3.3. Mikrokontroler	29
5. Závěr.....	31

6. Citace 32

1. Úvod

Technologie IoT se již stala přítomnou součástí normálního fungování člověka. Možnosti aplikací téměř neznají mezí i vzhledem k stále lepší dostupnosti nástrojů i zařízení s lepší výpočetní schopností a pokrocích v komunikačních možnostech.

Tato práce se zabývá projektem, který má za úkol zlepšit přístupnost novým úschovným kapacitám pro studenty na FELu. Cílem práce je vytvořit tzv. proof of concept pro systém úschovy předmětů za pomoci principu IoT.

Tento projekt lze přiblížit k již podobně vzniklým projektům. A nejen projektům, ale i k reálnému použití ve skutečném světě. To je velmi hojně nasazováno v logistice, kde se často využívá jako místo dodání, odkud si zákazníci můžou vyzvednout své zásilky. Zajímavým příkladem je projekt z Malajsijské univerzity „Cloud-Controlled Parcel Storage Box with User Image Capturing.“ V rámci projektu vytvořili úschovný box, který je ovládán z cloudu. Zároveň je v rámci boxu zabudovaná kamera, která má za úkol identifikovat člověka, jenž si objekt z boxu odebírá, a obrázek je poté nahrán zpátky do cloudu[1]. Příkladem nasazených logistických komerčních řešení je například Z-box[2] (od společnosti Zásilkovna) nebo PPL Parcelbox[3]. Tato řešení jsou velmi podobná ve svém základním principu jako bude tento projekt. Společnost alokuje box pro zásilku, kde si pak následně osoba vyzvedne svoji zásilku pokynem zpravidla přes aplikaci v telefonu.

Na fakultě se vyskytují dva způsoby, kde si je možné uschovat osobní věci. Jednou z možností je využití odkládacích skříněk v suterénu, kde si je nutné půjčit klíč od ostrahy na vrátnici. Druhou možností je v zimních obdobích si uschovat bundu v šatně, kde lze po domluvě uschovat i případně jiné vhodné osobní věci.

Práce je strukturována od částí rozboru konceptu řešení celého záměru, tedy čistě teoretické poznámky a koncept jako takový, až po praktické provedení.

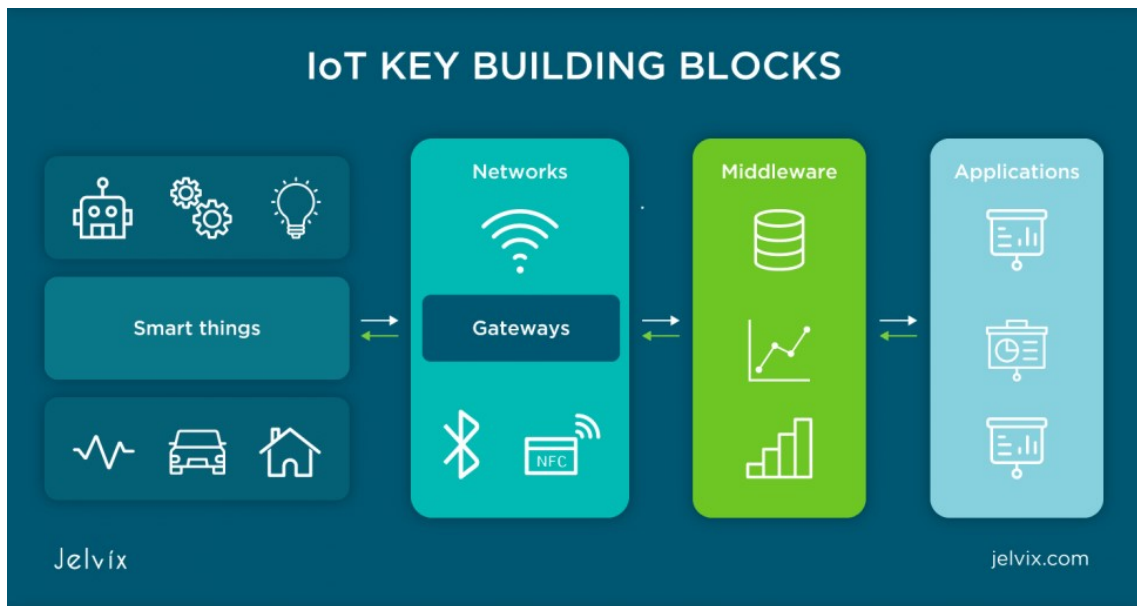
2. Obecný princip navrhovaného systému

Princip fungování systému pro úschovu je soustava navzájem propojených zařízení nebo prvků, které spolu musí fungovat. Zařízení jsou spolu spojena různými druhy standardizovaných komunikačních protokolů. Pro tento projekt je nezbytná komunikace přes internet, a proto navrhovaný systém je typovou variací Internetu věcí (Inthernet of Things – IoT).

2.1. Internet věcí – IoT

Internet věcí je termín, který popisuje navzájem spojené zařízení nebo systémy, jež navzájem komunikují přes drátové nebo bezdrátové sítě. Tato zařízení či tyto systémy od sebe mohou být vzdáleny i několik tisíc kilometrů, a proto souhrnně používají internet jako médium pro svoje vzájemné propojení.[4] Odtud název Internet věcí.

Obecným úkolem systémů IoT je získávat, zpracovávat a následně vytěžovat získaná data. Proto lze schematicky rozdělit IoT do 3 až 4 částí, které jsou propojeny. První vrstva je soubor zařízení, které sbírají data. Zpravidla to jsou vestavěné (embedded) systémy, které v sobě mají senzory získávající dané hodnoty či informace. Tato zařízení nutně nemusí být připojena k internetu. Pokud zařízení nejsou spojena, je potřeba vsunutá druhá vrstva. Ta funguje jako přestupní bod pro přenášení data. Zpravidla se jedná o zařízení, které přijímá data přes bezdrátovou komunikaci. Přijatá data zpracuje a následně pošle přes internet dál ke sběrně dat. Tento typ zařízení se ale nemusí implementovat, pokud jsou zařízení pro sběr data připojena přímo k internetu. Následuje místo sběru dat, což je 3. vrstva. Zde se přijmou data a následně jsou uložena. Poslední (4.) částí je analytická vrstva. Ta má za úkol přijatá data zpracovat a analyzovat. V závislosti na požadavcích mohou být výsledné analýzy čistě informativní nebo mohou být připojeny na nějaký akční člen, který by ihned reagoval na výsledky analýzy.[5]



Obrázek 1 - přehled architektury IoT[5]

V aktuální době je IoT běžně používán v průmyslu. A v dohledné době se stane i neodmyslitelnou součástí života velké části osob v jejich osobním životě. Internet věcí se dotýká téměř každého odvětví. Využívá se při správě budov, kdy pomáhá s automatizací budov. V energetice dohlíží např. na stav sítě v distribuci, při generování elektrické energie anebo v zařízeních na úschovu energie. A další odvětví by mohly následovat, např. logistika, ICT a zdravotnictví.[6] Data v průmyslu pomáhají efektivnější práci se společnostmi jako celkem, to snižuje náklady, zpravidla zrychluje služby a v případě nasazení automatizace robotů i rychlejší manipulaci.

2.2. Používaná zařízení a technologie v IoT

V rámci řetězce Internetu věcí se vyskytují různé druhy zařízení, které vykonávají z podstaty jiné úkony

2.2.1. Koncová zařízení

Tato zařízení mají za úkol sběr dat. Skládají se zpravidla ze vstupů dat a jednotky, která data schraňuje a odesílá je jinam.

Sledované informace zpravidla získáváme skrz senzory. Ty obecně můžeme rozdělit na dva druhy: na analogové a digitální[7]. Analogové senzory jsou základní, a tedy přenáší získané hodnoty analogově. U těch musí následně mikrokontroler přichozí data převést do digitální formy. U digitálních senzorů se už původní analogový signál převedl na digitální v rámci senzoru a data posílá kontroléru digitální formou přes vybraný komunikační protokol.[5]

Mikrokontroler základní data ze senzoru zpracuje a připraví je s přidruženými informacemi k odeslání. Mikrokontroler lze přirovnat k počítači, který má ale všechny důležité komponenty integrované v jednom pouzdře. Mikrokontrolery se rozlišují jak výpočetními možnostmi, tak integrovanými technologiemi komunikace nebo možným počtem analogových/digitálních vstupů, tak i energetickou náročností, cenou a používanou architekturou. Preferovaná technologie přenosu dat je pomocí bezdrátové komunikace, což je smyslem celého konceptu Internetu věcí. Není to ale nutnost, na místech, kde by bezdrátová komunikace byla limitována okolními vlivy, je možné používat i drátovou komunikaci[5].

2.2.2. Koncentrátor/Gateway

Koncentrátor funguje jako místo, které sbírá data z několika zařízení. Tato data pak odesílá jako soubor dat z několika zařízení do cloudu k zpracování. Je to vložené zařízení, které v rámci komunikace mezi zařízeními a cloudem dělá prostředníka. Koncentrátor je pro data vstupním bodem do internetu, kudy pak putují na zpracování k serverům.[8] Do koncentrátoru jsou data posílána komunikačními protokoly, které lze použít pouze na krátké vzdálenosti.

Na vstupu do každého serveru je Gateway. To je vstupním místem k serverovému zpracování. Je to bod, kam se směřují všechna data putující internetem. Příchozí data předává přímo serveru, která zpravidla hned putují do databáze k uložení.[8]

2.2.3. Komunikace

Komunikační protokoly jsou zásadní pro přenos dat, přesněji jejich výběr. Největší důraz se klade na vybrání správného protokolu pro přenos mezi zařízeními a vstupem do internetu.

2.2.3.1. Drátová komunikace

Drátová komunikace je v případě Internetu věcí méně používaná, ale má svoje opodstatnění. Prvním důvodem pro možnou preferenci drátové komunikace je silné elektromagnetické rušení v průmyslovém prostředí, jelikož například v hutnictví a velkých tavných pecích dochází ke vzniku silných elektromagnetických polí. Druhým důvodem je možná interference s jinými bezdrátovými komunikačními protokoly, kdy by docházelo k silné vytíženosti části spektra až do takové míry, že by nebylo možné kanály v rámci toho spektra používat. Zároveň dokážou poskytovat přenos většího objemu dat.[9]

Mohou se používat průmyslové standardy, např.:

- RS-232: Asynchronní sériový protokol. Může operovat v halfduplex i fullduplex módu. Má velmi nízkou základní přenosovou rychlost (cca 20 kbit/s). Není vhodné

pro zapojení v režimu více zařízení. Maximální délka spojení je přibližně 15 metrů. Výhodou je jednoduchost.[10][9]

- RS-485 – Asynchronní sériový protokol. Maximální délka spojení je 60 metrů. Lze dosáhnout i větších vzdáleností při snížení přenosové rychlosti. Vyšší přenosová rychlost (cca 10 Mbit/s). Vhodné pro zapojení více zařízení do jedné smyčky.[10]
- I²C – Synchronní komunikace. Vhodné pro zapojení master/slave, kde je možných až 1024 zařízení. Pouze na vzdálenosti do 10 metrů. Nízká přenosová rychlost (cca 100 kbit/s). Výhodou je automatická konfigurace v rámci sítě. Nízká energetická náročnost.[9]

2.2.3.2. Bezdrátová komunikace

Bezdrátová komunikace přináší systémům Internetu věcí relativně značné množství možností. Jednou z výhod je jednoduché umístění na různá místa bez nutnosti pokládání vodičů pro komunikaci[9]. Spolu s vylepšením efektivity využití energie v rámci zařízení, je možné zařízení mít bez jakéhokoliv přivedení kabelů napájená, protože zařízení je schopno fungovat z baterie dostatečně dlouhé období. U bezdrátové komunikace vzniká nový požadavek na zabezpečení komunikace, protože bez ní by bylo možné jednoduše útočit na dané sítě.[11]

Pro Internet věcí se mohou používat tyto protokoly, např.:

- Wi-Fi – Běžně používaný protokol. Je velmi komplexní. Nabízí dobré šifrování komunikace a dostatečnou rychlost pro přenos dat (v závislosti na verzi od 10 Mbit/s např. až do 100 Mbit/s). Pokrytí sítě do rádiu až 100 metrů. Nevýhodou je relativně velká energetická náročnost. V oblasti Internetu věcí se používá pásmo 2,4 GHz.[9]
- BLE (Bluetooth Low Energy) – Speciální verze Bluetooth. Přenosová rychlost do 2 Mbit/s. Rádus pokrytí sítě od 10 do 30 metrů. Nízká energetická náročnost. AES šifrování s 128-bit klíčem. Od verze 5.0 je možné tvořit Bluetooth Mesh sítě.[9][12]
- ZigBee – Pro sítě s nízkým datovým zatížením. Rychlost přenosu je cca 250 kbit/s. Nízká energetická náročnost. Vysoká spolehlivost a zabezpečení komunikace. Možnost velké sítě až s 65 tisíci zařízeními. V síti se zařízení sama organizují. Nepodporuje internetové protokoly. Podporuje různé topologie včetně mesh.[9][13]
- LoRa – Protokol pro komunikace na dlouhé vzdálenosti (až 5 kilometrů). Energeticky nenáročný, možnost využití napájení z baterie. Nízký datový přenos. Dobré šifrování. [9]

2.2.4. Serverové komerční služby

Místem, kde se data začínají vytěžovat, je server. Toto místo zastřešuje uschovávání dat, jejich analýzu. Ačkoliv je možné v rámci projektu vytvořit celé vlastní řešení, v dnešní době je výhodné serverové služby zřídit jako službu.

Poskytovatelé zpravidla nabízejí značné množství nástrojů pro správu, analýzy a přehledy získaných dat. Používáním těchto služeb lze pak používat pokročilé nástroje, jako např. použití strojového učení na predikci údržby strojů ve výrobě nebo jednoduchost škálování daného systému. Většinou tyto služby nabízejí i kvalitní zabezpečení před napadením systému. Mezi hlavní poskytovatele služeb patří např.: AWS IoT (Amazon Web Services), Azure IoT (Microsoft), IoT Core (Google Cloud)[14] nebo ThingsBoard (open-source)[15].

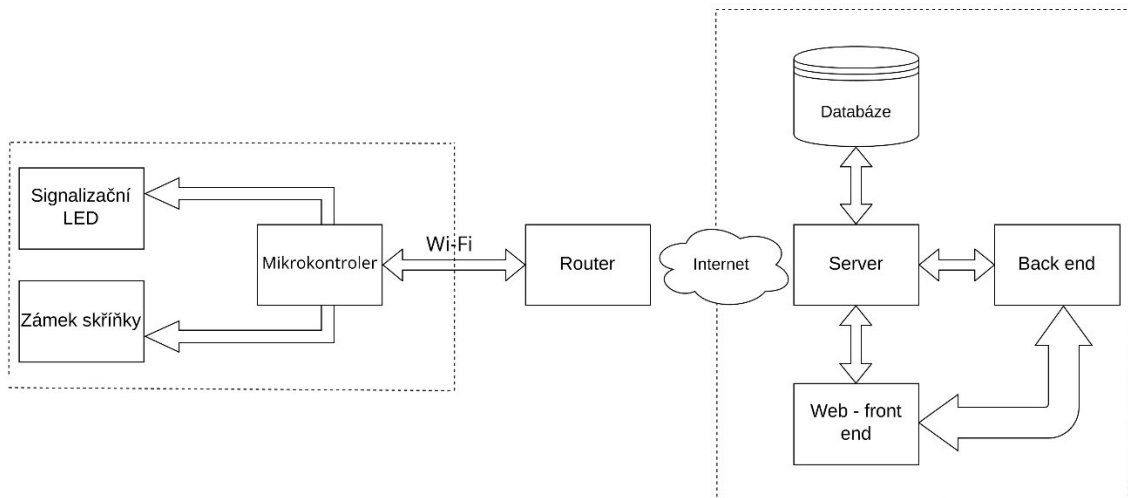
2.3. Popis fungování systému úschovných boxů

Osoba má potřebu si následující den na pár hodin uschovat svoje zavazadlo, proto přes webový prohlížeč otevře stránky webu s rezervací. Pomocí Google účtu (každý student ho má vytvořen spolu se školním účtem) se přihlásí. Po přihlášení vytvoří rezervaci na požadované období. Vytvořenou rezervaci je možné případně zrušit. Následující den, přibližně v čase počátku rezervace je osobě zpřístupněno ovládání přidělené skříňky. Ve webovém rozhraní je ovládací tlačítko na otevření dvířek. Po vložení zavazadla a zavření je zavazadlo uschováno. Obsazenost skříňky označuje červená kontrolní LED na dvířkách. Osoba si pak může po dobu rezervace otevírat skříňku, jak potřebuje. Po vyzvednutí zavazadla osoba ve webovém prostředí ukončí rezervaci a skříňka se v systému uvolní k další úschově.

V rámci tohoto projektu, který je proof of concept, budeme implementovat tento systém jako testovací. Data, která během zkušebního období nasbíráme, využijeme k analýze a případné reakci, zda bude toto řešení využíváno nebo jestli bude potřebné kapacity úschovných boxů navýšit.

3. Návrh systému úschovných boxů

3.1. Schéma systému úschovných boxů



Obrázek 2 - blokové schéma systému

Ve smyslu technologie Internetu věci lze popsat i systému úschovných boxů. První vrstvou je mikrokontroler, který ovládá notifikační LED a otevírá zámek skříňky. Jako druhá vrstva funguje Access Point routeru. Ten zajišťuje komunikaci mezi mikrokontrolerem a serverem. Koncentrátor komunikuje se zařízením pomocí Wi-Fi a informace předává do internetu serveru. Poslední dvě vrstvy jsou v rámci serveru, tedy část sběru dat a provádění analýzy. Na serveru bude i zároveň web server, který bude zodpovídat za příjem a správu rezervací a ovládání zámku na mikrokontroleru.

Mikrokontroler řídí zámek přes logické stavy pomocí relé modulu. Signalizační LED řídí přímo. Mikrokontroler komunikuje s nadřazenými vrstvami pomocí Wi-Fi. Koncentrátor informace přeloží na ethernetové rámce a pošle je do internetu. Server je propojen k internetu přes ethernet. Komunikace probíhá přes TCP/IP protokoly.

3.2. Komponenty v systému úschovných boxů

3.2.1. Lokální server

Lokální server je řídicím členem celého systému. Je základem pro poskytování webových služeb, zpracovává požadavky na rezervaci a vykonává řídicí činnost.

Pro tento projekt je jako web server ideální Apache Server (celý název projektu: Apache HTTP Server Project). Apache je open-source, volně dostupná a vyvíjená implementace HTTP severu.

Je multiplatformní a je vyvíjená komunitou.[16] Minimálním požadavkem na bezproblémové fungování serveru je OS Windows 2000 a novější nebo na vybraných distribucích Linuxu, procesor 64-bitový s architekturou x86, 512 MB RAM (stačí pro přibližně 80 tis. statických stránek), 300 MB (závisí na obsahu stránek a zároveň na velikosti databáze).

Pro běh serveru se použije platforma Docker. Ta dovoluje vývojářům vytvářet, nasazovat, aktualizovat a spravovat kontejnery. Kontejner je standardizovaná komponenta, která vytváří virtualizaci v rámci OS, na které běží vybraná aplikace. To je možné díky možnosti izolace a virtualizace v rámci kernelu Linuxu. Standardizované prostředí kontejnerů zabezpečuje, že na různých typech serverů je zaručen bezproblémový chod kontejnerů v rámci Dockeru na tomto serveru.[17]

3.2.1.1. Uživatelské rozhraní

Webové rozhraní, přes které interaguje člověk s webovou aplikací. Poskytuje vizuální stránku systému. Je tvořen zpravidla značkovacím jazykem HTML. Ten definuje, jak se daná stránka zobrazuje v uživatelské prohlížeči[18]. Jazyk HTML je často obohacen o CSS[19][20], což je jazyk pro popis zobrazovaných elementů na stránce a posouvá designové možnosti zobrazení. Samo uživatelské rozhraní je čistě statické[20], proto se ještě doplňuje skriptovacími jazyky jako PHP nebo JavaScript[21], aby mohly být stránky dynamické, tedy aby mohly zobrazovat požadované informace a pružně reagovat na uživatelské akce[22].

Pro tento projekt je potřeba vytvořit minimálně několik stránek, do kterých bude uživatel zadávat důležité informace. Tedy vznikne stránka se základními informacemi. Dále stránka, kde se bude vyplňovat informace k rezervaci, stránka přihlášení pro uživatele a stránka pro přehled vlastních registrací.

3.2.1.2. Server

Aby web mohl plnohodnotně fungovat, musí se provádět i úkony na pozadí. Proto s ním uživatel neinteraguje přímo a ani nevidí, co ve skutečnosti dělá[22]. V zásadě s tím nepřijde do styku, pouze přes požadavky skrz uživatelské rozhraní. Obecně se na pozadí serveru uchovávají a zpracovávají data, která uživatel potřebuje nebo jsou potřeba k bezchybnému chodu kódu. Jsou v nich tvořeny API pro komunikaci s jinými aplikacemi či pro propojení s jinými službami. Možnost výběru jazyka pro programy na serveru je široká, je možné vybírat mezi např.: PHP[21], Java a Node.js. V případě uchování dat se v rámci serveru vytváří databáze. Ta, pokud je relační, je zpravidla vybranou verzí SQL databáze.

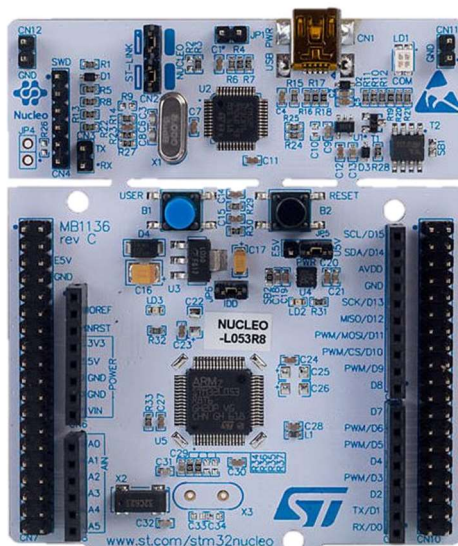
Návrhem vznikla potřeba řešit několik různých procesů. Důležitým procesem je registrace uživatele, případně jeho přihlášení[23]. To je možné nahradit využitím přihlášení přes účet

společnosti Google, která přes API předá autentifikaci serveru, že uživatel existuje[24]. Další procesy jsou vytvoření spojení s databází, aby bylo možné zapisovat a číst z databáze, vytvoření rezervace z dat přijatých z formuláře z uživatelského rozhraní. Pak server pro ovládání mikrokontroleru použije Rest API[25] a bude na základě požadavků ze stránky s uživatelskými vytvořenými rezervacemi otevírat zámek skříňky. Pro stálost systému je nutné použít relační databázi, neboť se budou uschovávat jednotliví uživatelé a jejich identifikační informace, budoucí rezervované sloty a již proběhlé rezervace.

3.2.2. Mikrokontroler

Mikrokontroler má v tomto systému na starost ovládání zámku, signalizaci pomocí LED a komunikaci se serverem. Požadavky na mikrokontroler jsou relativně mírné. Mikrokontroler musí mít dostatek vstupně-výstupních portů, Wi-Fi modul, základní podporu web serveru a běžné požadavky na výpočetní možnosti. Jiné projekty, které se podobným systémem zabývají, jsou zpravidla velmi robustní systémy, které jsou myšleny pro nasazení s velkým množstvím úschovných boxů. Dalo by se to označit za „enterprise“ řešení. Tedy dalším kritériem bude cena, kdy sice budeme zachovávat předem daná kritéria, ale budeme zároveň hledat nízkou cenu.

Na výběr máme 3 možné zástupce mikrokontrolerů: ESP(32/8266), STM a Arduino. Arduino může vyloučit hned. Žádný z jeho SoC (System on Chip) přímo nenabízí modul bezdrátové technologie. Pokud ano, je to zpravidla ve spojení například s ESP8266, který nese funkci čipu dedikovaného pro komunikaci. STM vyrábí několik sérií, která nachází využití v jiných místech. Obecně lze říci, že v oblasti, ve které hledáme mikrokontroler pro tento projekt, má STM lepší výpočetní možnosti, robustnost celého ekosystému a mnohem větší počet vstupně/výstupních portů. Na druhou stranu, výrobky STM mají relativně vyšší pořizovací cenu, jejich možnosti jsou občas přítěží, jsou mnohem náročnější na nastavení a přípravu pro projekt, a hlavní nevýhodou je, že nenabízejí SoC s Wi-Fi komunikačním protokolem, což je pro systém úschovných boxů podstatná nevýhoda, musel by se tedy spojit s Wi-Fi modulem.[26]



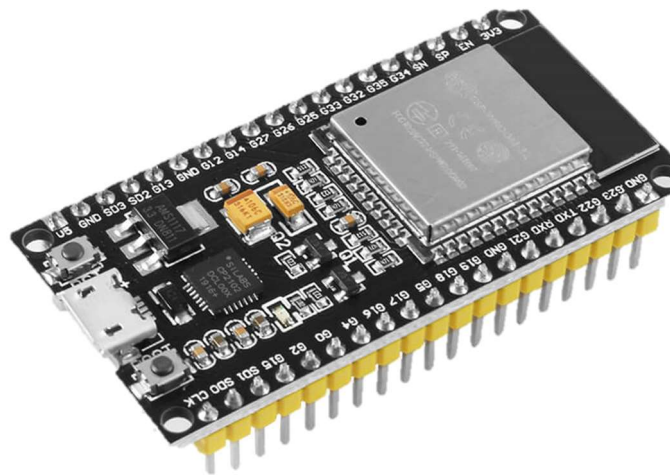
Obrázek 3 - obrázek devkitu STM32 NUCLEO[27]

ESP je významným hráčem v segmentu levných, malých zařízení s podporou komunikace přes Wi-Fi. ESP nabízí dvě verze ESP8 a ESP32. Podrobné porovnání je níže v tabulce.

	ESP8266	ESP32
MCU	Xtensa L106	Xtensa LX6
WiFi	802.11 b/g/n 20 MHz	802.11 b/g/n 40 MHz
Bluetooth	NE	BL i BLE
SRAM	NE	ANO
Flash	NE	ANO
GPIO	17	34
ADC	10-bit	12-bit
CAN	NE	ANO
Hallův senzor	NE	ANO
Ethernet MAC rozhraní	NE	ANO

Tabulka 1 - porovnání STM8266 a ESP32[28]

Shrnutím můžeme říct, že ESP32 je výkonnější, přináší podporu nových vestavěných funkcí, má více vstupně-výstupních portů a má lepší modul pro bezdrátovou komunikaci, a proto bude výběr probíhat z vývojových desek s ESP32. Ty nabízí velké množství výrobců. U těch pak jen zpravidla závisí na počtu vyvedených portů a na velikost paměti FLASH.[28]



Obrázek 4 - devkit ESP32[29]

3.3. Bezpečnostní nároky na systém

System úschovných boxů musí zaručovat určitou úroveň zabezpečení, aby mohl poskytovat úschovné služby. Z tohoto pohledu jsou dva možné způsoby se neoprávněně dostat k uloženým věcem, a to buď mechanicky, kdy útočník použije fyzickou sílu k otevření boxu, nebo pokud se podaří útočníkovi využít řízení boxu. Je potřeba definovat účel úschovných boxů, aby tomu následně odpovídala i úroveň zabezpečení. Účelem není úschova cenných věcí, drahé elektroniky či peněz, ale možnost si odložit méně cenné věci. Za odložené věci nikdo nepřebírá zodpovědnost.

3.3.1. Mechanická ochrana

Mechanická ochrana je obecně jedním typem možnosti ochrany objektů a věcí. O ochranu před proniknutím se starají mechanické prvky, mezi něž patří například zámky, bezpečnostní kování, mříže, bezpečnostní fólie, ale i trezory nebo přenosné pokladny. Každý mechanický prvek je přemožitelný, a to buď po dostatečném čase, nebo po požití správného nářadí. Mechanické prvky se dělí do různých bezpečnostních tříd, dle náročnosti na přemožení ochrany a osazení jiných doplňujících mechanických prvků. Tímto tématem se zabývá několik norem. Pro trezory platí např. tyto normy: ČSN EN 1143-1[30], ČSN 91 6012[31] a pro trezory s vysokou bezpečností ČSN EN 1300[32]. Ty definují parametry různých druhů trezorů a postupy zkoušek pro rozdělení do jednotlivých tříd. To rozšiřuje ještě možná certifikace Národním bezpečnostním úřadem (NBÚ), která normu rozšiřuje o všechny typy úschovných objektů. V případě rozlišení tříd pro překonání plášťové ochrany (dveře, okna, mříže a okenice) jsou platné normy ČSN EN 1627[33] až 1630.

Bezpečnostní třída	Popis, vůči kterému typu útoku odolá
RC1	Útok jednoduchým nářadím a fyzickým násilím (kopáním atd.). Útočník nemá žádné znalosti MZS. Snaží se nezpůsobit hluk. Příležitostný zloděj.
RC2	Stejně jako RC1, ale útočník má elementární znalosti MZS.
RC3	Použití páčidel (do 71 cm), šroubováku či jiných ručních nářadí. Zloděj má znalosti o MZS a dokáže je využít.
RC4	Zkušený zloděj používá akumulátorovou vrtačku a dláta. Již neřeší hluk.
RC5	Velmi zkušený zloděj. Používá jednoruční elektrické nářadí (úhlová bruska, pila).
RC6	Stejně jako RC5 může používat jakékoliv elektrické nářadí.

Tabulka 2 - popis bezpečnostních tříd RC[33]

Hlavním mechanickým prvkem boxu je jeho opláštění, jež je z pevné překližky, která je široká 1,9 cm a elektromagnetickým zámekem.

V případě boxů v našem systému se nedá mluvit o plnění norem, protože úkolem systému není úschova cenin či jiných podobných věcí, jak jsme již definovali. Ale pokud bychom museli nějakou definici použít, jelikož žádná norma pro úschovné objekty tohoto typu neexistuje, zkombinujeme do sebe několik dokumentů. Z pohledu normy ČSN EN 1627[33] až 1630, která definuje stupnici bezpečnostní třídy RC (Resistance class), by bylo možné zařadit box do třídy RC1. Je možné, že pokud by se provedla zkouška dle příslušné normy, by byl box ve třídě RC2, ale to je spekulace. Z pohledu certifikace NBÚ je box úschovný objekt typu 0, tedy uzamykatelný kancelářský nábytek. Do značné míry to odpovídá úrovni mechanické ochrany úschovných skříní v prvním podzemním podlaží budovy FEL Technická.

3.3.2. Elektronická ochrana

Elektronický útok je v dnešní době velmi diskutované téma, protože se stupňuje počet útoků na infrastruktury ICT všech typů subjektů. Elektronickou ochranou v tomto případě je schopnost odolat proti různým druhům útoků na software i hardware[11]. Odborným zaměřením zabývajícím se touto problematikou je Cybersecurity (kybernetická bezpečnost).

Postupně si představíme, jakým útokům mohou čelit systémy IoT. Typy hrozeb rozlišujeme podle typu a účelu útok. Z pohledu kybernetické bezpečnosti se vytváří vlastní architektura, která lépe definuje jednotlivé vrstvy a typy útoků. Přehled je v tabulce číslo 3.

Vrstva	Popis	hlavní nedokonalosti
Sběru dat	Snímání dat a objektů. Směrování útoku: důvěryhodnost dat	síla bezdrátového signálu, dynamická topologie IoT, možnost vystavení senzorových uzlů, HW omezení zařízení
Síťová	Správa sítě a posílání dat. Zaměření útoku: důvěryhodnost dat, ochrana soukromí a kompatibilita	kompatibilita komunikace a předání dat, důvěryhodnost dat
Middleware	Předání dat do aplikační vrstvy. Zaměření útoku: autentifikace zařízení, integrita a důvěryhodnost	zabezpečení komunikace, spojení se službami, stálost SW řízení vrstvy
Aplikační	Poskytování požadovaných služeb. Zaměření útoku: ochrana osobních dat a autentifikace identit	udělování povolení přístupu k datům, ochrana dat a jejich obnova, schopnost pracovat s velkými objemy dat, softwarové nedostatky

Tabulka 3 - přehled vrstev útoku na systém IoT[11]

1. *Vrstva sběru dat:* Vrstva je složena ze senzorů a sítě a má za úkol sběr dat, jejich základní zpracování a vysílání do celé sítě. Pro napadení této vrstvy se používá: **Replay útok** – zachycení autorizovaného toku dat, který je znovu přehrán, k získání nelegálního přístupu. K tomu se používají metody úpravy zpráv nebo opakované odesílání informace s ukradenou identitou zařízení v síti (dále spoofing); **Časový útok** – útočník ukradne časově limitovaná šifrovaná data; **Node capture attack** – útok s úkolem převzetí kontroly nad koncovým zařízením a tím pak infiltrovat síť; **Side channel attack** – získání metadat o zařízení a jejich následné zneužití. Příklady útoku: František potřebuje dokázat svoji identitu Julii pro udělení přístupu k webovému účtu. Julie požádala o heslo, jako prokázání Františkovy identity. Tuto komunikaci odposlouchává Radek a uloží odposlechnuté heslo. Později se Radek identifikuje heslem Františka a získá tak přístup k jeho účtu.[11]
2. *Síťová vrstva:* vrstva spravující toky dat a přesměrovávání dat různých IoT hubů a jiných zařízení skrz internet a mobilní síť. Používají se tyto útoky: **Spoofing** – útoky na toky dat, generování falešných zpráv a vytváření směrovacích smyček; **Sybil útok** – útoky, kdy se v síti objevuje jedna nebo více identit na různých místech zároveň,

útočí tím na snížení efektivity sítě a její důvěryhodnosti. Příklad útoku: Útočník dokáže infikovat celou síť posláním falešných směrovacích dat. Například po vyplnění dotazníku jste vyzván se přihlásit na sociální síť pro „ověření“, že vyplňující není robot. Přihlášení je směrováno na falešnou adresu. Po vyplnění přihlašovacích údajů je uloží, ale zobrazí error, a přesměruje dotyčného na pravé přihlašovací stránky.[11]

3. *Middleware vrstva*: SW mezi-vrstva, která je prostředníkem mezi síťovou a aplikační vrstvou. Na této vrstvě jsou správa autentifikace komunikace, ověřování integrity důvěryhodnosti dat. Tato vrstva je náchylná na tyto útoky: **Útok zevnitř sítě** – interní záměrná úprava a získání dat nebo informací ze sítě; **Útok komponenty třetí strany** – komponenty, které se používají na vrstvě PaaS (Platform-as-a-Service), mohou za určitých okolností představovat hrozbu pro síť; **Virtualizační útok** – případ, kdy VM (Virtual Machine) může být poškozena nebo, kdy ovlivňuje jiné VM. Příklad útoku: Osoba nelegálně získá přístup do systému. Zmapuje ho a zjistí jeho slabá místa. Poté může chtěná data krást nebo celý systém zničit.[11]
4. *Aplikační vrstva*: poskytuje všechny funkcionality pro koncového uživatele. Možné typy útoků: **Phishing** – získání užitečných informací zneužitím cizích přístupových dat, útočník následně nahraje do systému škodlivý kód (malware), nebo spyware (sw, který pouze pozoruje a získává informace); **Útok na přístupy** – útočník zamezí v přístupu připojených služeb v rámci systému. Příklad útoku: Útočník získal přístup do ovládací vrstvy, kde jsou propojení na různé služby třetích stran. Záměrnou změnou API klíče lze aspoň částečně paralyzovat celý systém.[11]

3.3.3. Řešení mimořádné události výpadku zdroje energie

Situace, kdy se systém setká s nějakou formou výpadku primárního zdroje elektrické energie, není nic, co by se nemohlo stát. Proto je potřeba brát v úvahu i tento problém. V situaci, kdy vypadne zdroj elektrické energie, nastane problém, kdy nebude možné si vyzvednout uschované osobní věci.

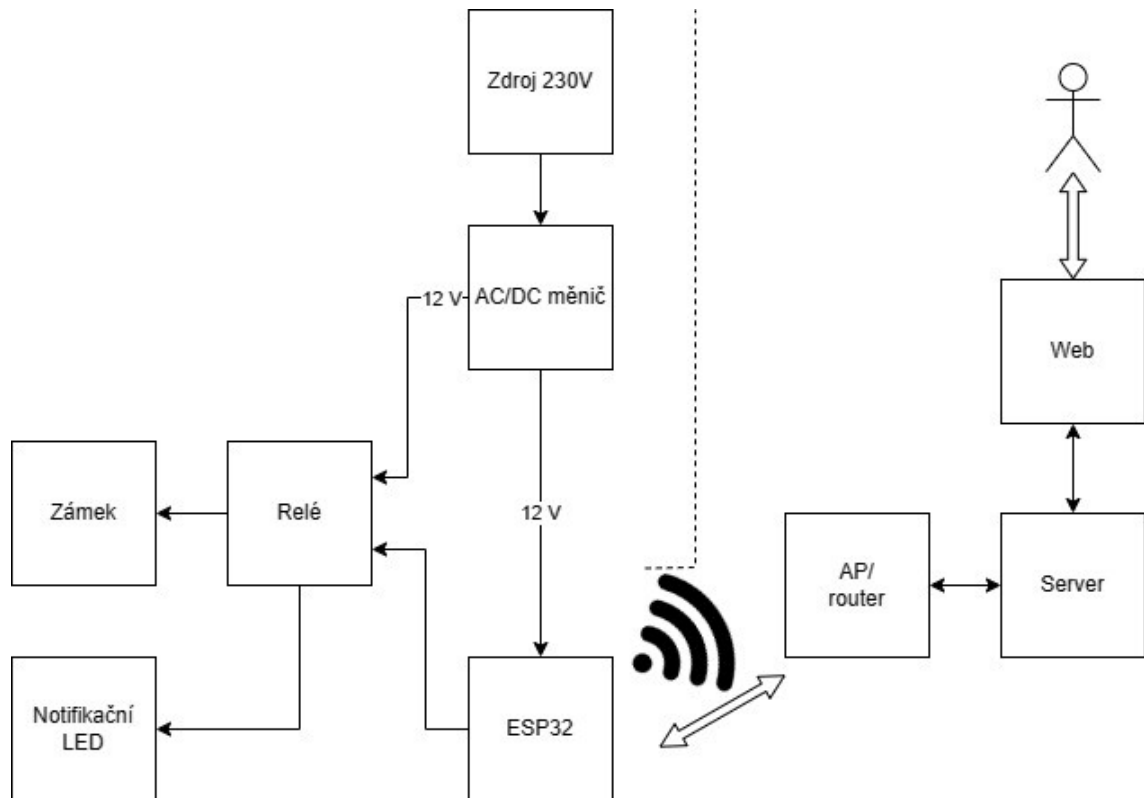
Nejdříve definujeme případy, na jaké části systému může mít výpadek energie vliv. První možná situace je, že dojde k lokálnímu výpadku energie v místě boxu. To může ovlivnit mikrokontroler, síťový prvek (AP + router) nebo obojí zároveň, což zapříčiní, že server nedokáže komunikovat s mikrokontrolerem, tedy dochází k tomu, že ho nelze ovládat. V té chvíli nastává problém s nedostupností uschovaných věcí. V případě výpadku energie pouze u síťového prvku, kdy mikrokontroler je stále napájen, je situace s nouzovým otevřením skříňky jednodušší. Pokud v mikrokontroleru připravíme nějaký nouzový přístup je tato situace lépe řešitelná, než kdyby ani mikrokontroler neměl napájení. Další možností je, že výpadek postihne místo, kde se nachází

server. To zapříčiní v zásadě stejnou situaci, jako kdyby nefungoval síťový prvek v prvním případě. Tedy mikrokontroler nebude řízen.

Je důležité definovat, čeho chceme dosáhnout při situaci, kdy se některá z částí systému dostane do situace, že nemá napájení. Nejideálnější řešení by bylo mít vytvořený celý sekundární systém, který by byl redundantní nad primárním. Řešilo by to veškeré možné výpadky. V okamžiku výpadku by se primární systém vypnul, a do vyřešení situace by celou funkčnost zajišťoval sekundární systém. To je vzhledem k účelu systému zbytečné. Nejenom, že by bylo nutné vytvořit klon primárního systému (tedy pořídit co nejpodobnější hardware), ale zajistit do obou míst redundantní zdroje energie, což v závislosti na umístění není vždy možné. Nejjednodušším vyřešením situace by bylo, že věci by byly uschovány až do doby, kdy celý systém bude funkční. To je ale z pohledu poskytování služby nechtěné. Pro nouzový přístup k věcem uschovaným v boxu je potřeba, aby mikrokontroler byl stále napájen, a to ideálně i po nastání výpadku napájení. To vyřešíme záložním napájením z baterie. V zásadě to bude systém UPS (Uninterruptible Power Supply/Source = zdroj nepřerušovaného napětí)[34]. Tento zdroj zaručuje, že i po výpadku energie ze sítě bude mikrokontroler napájen z energie uložené v bateriích. Následně vytvoříme tlačítko na nouzové otevření boxu. Z pochopitelných důvodů bude tlačítko skryto, aby ho nebylo možné zneužít. Zdroj UPS bude sloužit i jako napájení zámku, aby po zadání příkazu mohl být otevřen i zámek.

4. Realizace systému úschovných boxů

4.1. Reálné zapojení systému



Obrázek 5 - reálné blokové schéma zapojení systému

4.2. Použitá zařízení

V této kapitole základně představíme vybraná zařízení, která jsou součástí systému. V testovací verzi je server lokálně, ale vzhledem k tomu, že k tvoření rezervací bude možné využít internet, router bude mít i přístup do internetu.

4.2.1. Mikrokontroler

Při výběru podle stanovených podmínek bylo velké množství kandidátů. Nakonec byla vybrána deska s integrovaným modulem ESP32 WeMos D1 R32 UNO ESP32[35]. Deska splňuje všechny HW požadavky. Má dostatečný počet vstupně-výstupních portů a komunikační modul. Výčet parametrů:

- Integrovaný Wi-Fi vysílač – podporované standardy 802.11 b/g/n
- Podpora šifrování Wi-Fi WPA2 PSK
- Řadič CH340 pro komunikaci

- UART skrz USB Mikro konektor
- 34 vstupně-výstupních portů [35]



Obrázek 6 - EPS32 použité v systému[35]

Po výběru se mikrokontroler osvědčil dobrou nabídkou funkcí a vybaveností. Jediné, s čím nastal problém, byla komunikace přes Wi-Fi. Po delším vyšetřování se našla chyba. Daný výrobek měl vadnou planární integrovanou anténu Wi-Fi modulu. To mělo za příčinu, že nebyla možná komunikace mikrokontroleru a AP. To bylo vyřešeno úpravou antény, kdy byl připájen konektor SMA, na který byla připojena externí anténa.

4.2.2. Zdroj napájení

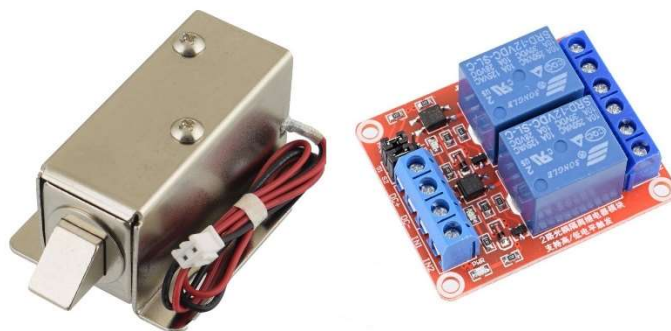
Pro účely napájení byl vybrán průmyslový zdroj napájení MEAN WELL RS-75-12. Zdroj je napájen ze sítě 230 V/60 Hz. Výstupní napětí je 12 V a maximální výkon dodávaný zdrojem je 72 W. Dané výstupní napětí se vybíralo, protože tímto napětím se ovládá elektromagnetický zámek. Tedy nebylo potřeba použití měničů na převedení na vyšší napětí potřebné pro ovládání. Má nízký výstupní šum a toleranci odchylky výstupního napětí $\pm 1\%$. [36]



Obrázek 7 - použitý zdroj[36]

4.2.3. Zámek, jeho ovládání

Zámek (GM201) je jednoduchý elektromagnetický. Funguje na principu přitažení západky, čímž dovolí otevření boxu dveří. Mechanismus je spínán stejnosměrným 12V napětím.[37]



Obrázek 8 - zámek a relé modul[37][38]

Pro ovládání zámku se používá relé modul. Ten je napájen napětím 12 V DC. Samotné spínání relé je řízeno signálem z mikrokontroleru. Relé osazená na modulu jsou běžná JOC-3FF-S-Z. Relé má možnost využívat jak metodu spínání NC/NO, tak i možnost logického invertování příchozího signálu z řídicího mikrokontroleru.[38]

4.2.4. Síťový prvek

Pro nasazení v režimu proof of concept bylo potřeba vybrat zařízení, které bude zastupovat více zařízení v jednom, tedy v rámci jednoho pouzdra Wi-Fi AP, router a switch, a proto byl prozatím vybran D-Link DWR-116. Podporuje protokoly 802.11 n/g/b, což je nezbytné pro bezdrátovou komunikaci s mikrokontrolerem.[39]



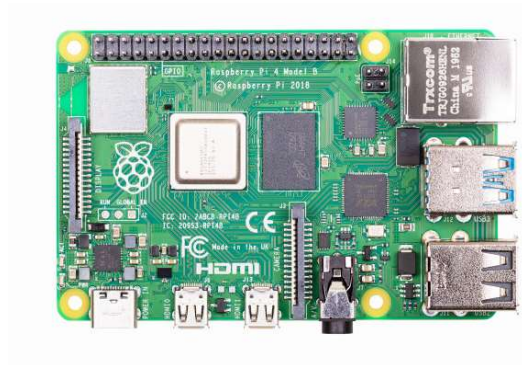
Obrázek 9 - D-Link DWR-116[39]

4.2.5. Server

Požadavky na server nejsou nijak náročné. Jediný požadavek, tedy fungování web serveru, je relativně jednoduše splnitelný. Pro tuto instalaci byla vybrána možnost použít jako server Raspberry Pi. Tento jednodeskový počítač bezpečně splňuje požadavky. Má konektor RJ-45 pro

připojení do sítě. Na tomto serveru nebudou vytvářeny VM jako takové, ale bude používán Docker[17], aby bylo možné v budoucnu třeba web server přesunout do velkého serveru, kde bude fungovat i s jinými aplikacemi. Vybraný server je serverový soubor XAMPP[40], což je distribucí projektu Apache[16].

Vybraná konfigurace je osazena čtyř jádrovým procesorem ARM. Pro bezproblémový běh web serveru je přístupných 4 GB RAM. Zařízení je napájeno přes USB-C konektor 5 V DC.[41][42]



Obrázek 10 - Raspberry Pi 4[41]

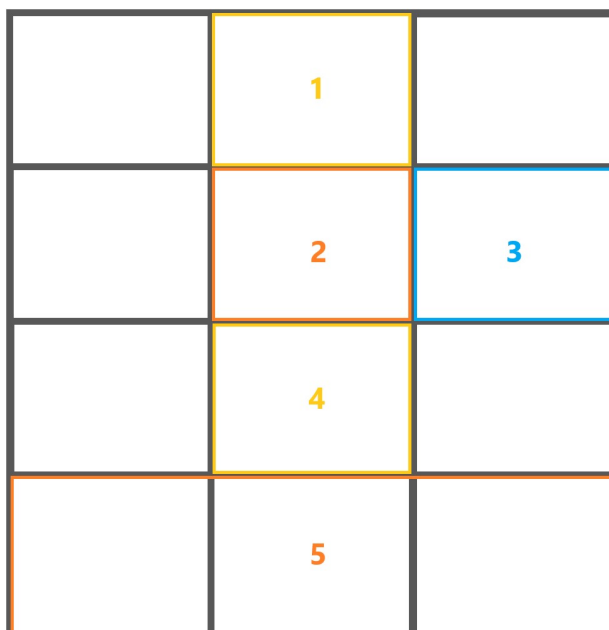
4.3. Vytvořený software

Níže bude popsán software, který definuje funkčnost systému. Bude rozebráno obecné fungování.

4.3.1. Webové prostředí

Webové prostředí bylo vytvořené pomocí značkovacího jazyka HTML[18]. Tím byly vytvořeny hlavní vizuální prvky prostředí, jež je doplněné jazykem kaskádových stylů CSS[20], který přidává možnosti zobrazení. Nebylo použito žádných předem vytvořených šablon. Pro dynamické prvky stránky byl použit skriptovací jazyk PHP[21].

Obecně se každá stránka skládá ze šablony, respektive je rozsegmentována na několik oblastí, která má každá svůj vlastní účel. Toto rozdělení bylo vytvořeno pomocí modulu CSS Flex Grid[43]. Tento modul zjednodušuje segmentaci stránek, protože dříve se musela segmentace zpravidla provádět v rámci HTML[19].



Obrázek 11 - rozložení webového rozhraní

Horní segment (1) je pro úvodní obrázek, kde je aktuálně vloženo jednoduché logo systému. Prostřední segment (2) je hlavním navigačním panelem, v kterém se nachází odkazy na další stránky v rámci webu. V rámci pravého segmentu (3) je tlačítko pro přihlášení, případně odhlášení uživatele. Prostřední segment (4) je místem, kde se zobrazují hlavní prvky dané stránky. Tedy v případě podmínek použití tam je text, v případě registrace tam jsou pole na vyplnění potřebných informací. Posledním segmentem (5) je zápatí. To je zatím bez využití.

Byly vytvořeny tyto stránky:

- Úvodní – přivítání uživatele
- Podmínky použití – zjednodušené podmínky použití úschovného systému boxů
- Rezervace – vytváření rezervace na určité období
- Přehled rezervací – místo, kde je přehled rezervací uživatele

4.3.2. Server

Celý část, která je na pozadí webového serveru, která řídí systém, je napsán v jazyku PHP. Tato část serveru spravuje rezervace a jejich uživatele. Zároveň řídí mikrokontroler, kterému předává pokyny na otevírání boxů. V rámci této části je i zřízena MySQL databáze, která ukládá všechna důležitá data nutná k fungování systému.

V databázi se ukládají data o uživateli a informace o rezervacích, jak budoucích, tak i minulých. Uložená data uživatelů jsou pouze jejich ID, aby je bylo možné identifikovat.

V případě informací o rezervaci se ukládá čas a datum počátku využití a čas a datum ukončení. Veškerý přístup do databáze je skrz PHP kód.

V rámci serveru jsou dále používány běžné funkce PHP, které definují jednotlivé funkčnosti systému. Výčet úkolů:

- Komunikace s MySQL databází[44]
- Komunikace s API Google pro autentifikaci uživatelů[24]
- Zabezpečení, že osoba bez přihlášení nemůže vytvořit rezervaci
- Zpracovávání dat přijatých skrz HTML formuláře[19]
- Správa rezervací
- Řešení různých chyb v rámci celého systému
- Komunikace s mikrokontrolerem – REST API[25]

Komunikace s mikrokontrolerem je vedena přes protokol HTTP(S), pomocí REST API[25]. Příkazy jsou posílány skrz metody GET/POST, kdy jsou přenášena informativní data mikrokontroleru.

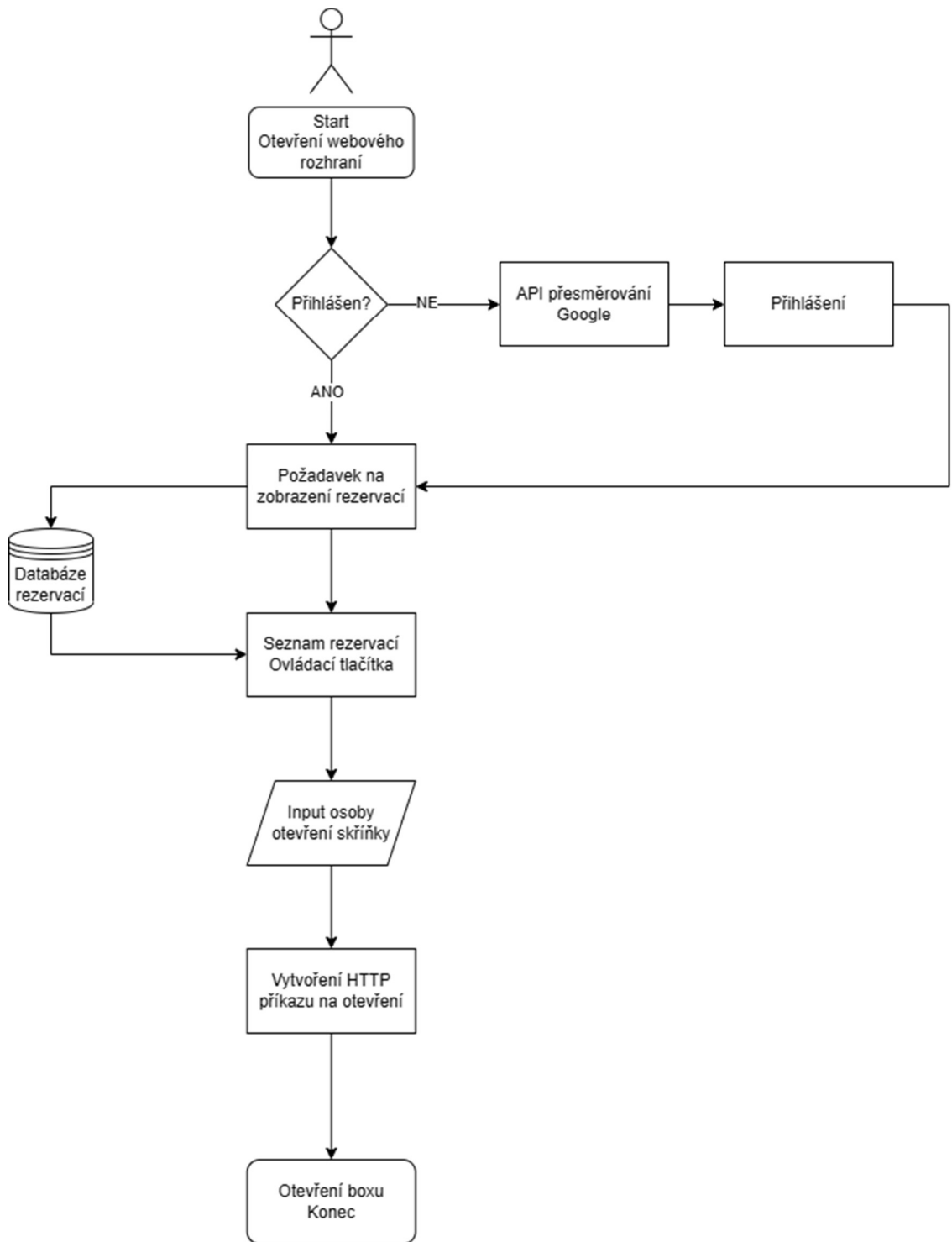
4.3.2.1. Google API - přihlášení

Původně bylo zamýšleno vytvoření vlastní registrace a přihlašování uživatelů. To se po úvaze změnilo, protože každý student univerzity má při studiu vytvořen účet Google, je tedy k přihlašování do systému použito ověření účtu Google. Je využívána API brána Google Cloud, která předá serveru potvrzení o autentifikaci uživatele tedy, že existuje, a jsou předány základní informace o uživateli[24]. Tyto údaje se uloží do databáze. O autorizační službu si může zažádat téměř každý uživatel služeb Google. Pro implementaci připojení k API bylo přiděleno několik údajů, pod kterými Google přiděluje přístupová práva. Je přidělen Client ID (identifikace uživatele Google Cloud účtu) a Client Seceret (bezpečnostní klíč pro komunikaci). Pro možnost jednoduchého programování, je potřeba stáhnout knihovnu poskytovanou přímo společností Google.

4.3.3. Mikrokontroler

Na mikrokontroleru se nachází rutinní program. Ten je vytvořen tak, aby čekal pouze na pokyny ze serveru. Program je napsán v jazyku C, za použití platformy Platformio[45].

Cyklus mikrokontroleru se skládá z prvotního připojení k Wi-Fi a následného udržování připojení. Následně cyklus jen čeká na příchozí HTTP metody, aby splnil požadavek, a to zpravidla otevření boxu.



Obrázek 12 - flowchart odemčení skříňky

5. Závěr

Tato práce je popisem jednoduchého systému pro úschovu osobních předmětů. V rámci práce je teoreticky popsáno, o jaký obecný typ systému se jedná. Následuje návrh systému a jeho funkce. Rozebralo se zabezpečení systému a možná místa, kde může dojít k jeho napadení. Poslední kapitola je ukázkou implementace navrženého systému, podle daných požadavků.

Vytvořený systém měl být odlehčenou konkurencí pro velké systémy poskytované pro nasazování ve velkém měřítku, a to pouze pro logistické firmy. Tyto původně mohutné systémy se v malém měřítku implementují velmi špatně, a to především kvůli robustnosti systému, která klade nároky na okolní systémy a ITC vybavení. Cílem tedy bylo navrhnout cenově rozumný systém, který by splňoval základní funkce podobných systémů. Tedy zarezervovat místo, uschovat objekt a následně na pokyn vydat obsah.

Navržený systém je plnohodnotným systémem úschovných boxů, který splňuje stanovené požadavky. Pro ověření navrhovaného systému vznikla verze, která je v testovacím režimu. V rámci testovacího režimu jsou zprovozněny pouze základní funkce systému. Momentálně je systém ve stavu, že rezervuje volnou kapacitu boxu a následně zpřístupňuje ovládání dvířek osobě, která má daný časový slot zarezervovaný. Z běhu systému se budou získávat údaje pro další vyhodnocení a rozhodnutí o chodu systému. Testovací verze systému aktuálně trpí menšími softwarovými problémy. Jsou například jen ošetřeny základní chyby, které během běžného fungování mohou nastat.

Další vývoj systému závisí na jeho osvědčení v praxi, jestli je navržen správně, jestli bude využíván. Následující verze systému, která by měla být již verzí téměř produkční, by měl být nasazen v celém navrhovaném zapojení a s vylepšenou verzí software, tedy například s náhradním zdroje energie a případně přidáním dalších funkcí v rámci webového rozhraní.

6. Citace

- [1] GUO GENG, Yeow a Soon CHIN FHONG. Cloud-Controlled Parcel Storage Box with User Image Capturing. *Evolution in Electrical and Electronic Engineering* [online]. 2022, **3**(2), 59-68 [cit. 2023-05-17]. ISSN 2756-8458. Dostupné z: doi:10.30880/eeee.2022.03.02.008
- [2] Z-Box. *Zásilkovna* [online]. [cit. 2023-05-18]. Dostupné z: <https://www.zasilkovna.cz/zbox>
- [3] PPL spouští do provozu samoobslužné výdejní i podací boxy – PPL Parcelboxy. *PPL* [online]. [cit. 2023-05-01]. Dostupné z: <https://www.ppl.cz/cs/w/spusteni-parcelboxu>
- [4] What is the Internet of Things?. *MCKinsey & Company* [online]. McKinsey & Company, 2023, 17 August 2022 [cit. 2023-04-18]. Dostupné z: <https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-the-internet-of-things>
- [5] ITSEKSON, Alex. What Are the 7 Layers of IoT Architecture?. In: *Jelvix* [online]. [cit. 2023-04-21]. Dostupné z: <https://jelvix.com/blog/iot-architecture-layers>
- [6] World of IoT. *IoT Now* [online]. [cit. 2023-05-17]. Dostupné z: <https://www.iot-now.com/world-of-iot/>
- [7] Sensors in Internet of Things(IoT). *GeeksforGeeks* [online]. [cit. 2023-05-18]. Dostupné z: <https://www.geeksforgeeks.org/sensors-in-internet-of-thingsiot/>
- [8] POSEY, Brien, LAVEREY, Tréa, ed. IoT gateway. *TechTarget* [online]. [cit. 2023-05-18]. Dostupné z: <https://www.techtarget.com/iotagenda/definition/IoT-gateway>
- [9] GLORIA, Andre, Francisco CERCAS a Nuno SOUTO. Comparison of communication protocols for low cost Internet of Things devices. In: *2017 South Eastern European Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)* [online]. IEEE, 2017, 2017, s. 1-6 [cit. 2023-05-17]. ISBN 978-618-83314-0-2. Dostupné z: doi:10.23919/SEEDA-CECNSM.2017.8088226
- [10] Základní rozdíly mezi RS-232, RS-422 a RS-485. *Ipc2U* [online]. [cit. 2023-05-18]. Dostupné z: <https://ipc2u.cz/blogs/news/zakladni-rozdily-mezi-rs-232-rs-422-a-rs-485>
- [11] LU, Yang a Li Da XU. Internet of Things (IoT) Cybersecurity Research: A Review of Current Research Topics. *IEEE Internet of Things Journal* [online]. 2019, **6**(2), 2103-2115 [cit. 2023-05-18]. ISSN 2327-4662. Dostupné z: doi:10.1109/JIOT.2018.2869847
- [12] AL-SARAWI, Shadi, Mohammed ANBAR, Kamal ALIEYAN a Mahmood ALZUBAIDI. Internet of Things (IoT) communication protocols: Review. In: *2017 8th*

International Conference on Information Technology (ICIT) [online]. IEEE, 2017, 2017, s. 685-690 [cit. 2023-04-22]. ISBN 978-1-5090-6332-1. Dostupné z: doi:10.1109/ICITECH.2017.8079928

[13] QUINN, Jones. IoT Architecture: Topology and Edge Compute Considerations. *Digi* [online]. Digi International, 22.7.2020 [cit. 2023-04-10]. Dostupné z: <https://www.digi.com/blog/post/iot-architecture-topology-and-edge-compute>

[14] RANA, Diskha. Top 11 Cloud Platforms for Internet of Things (IoT). *DZone* [online]. [cit. 2023-05-01]. Dostupné z: <https://dzone.com/articles/10-cloud-platforms-for-internet-of-things-iot>

[15] *ThingsBoard* [online]. [cit. 2023-05-10]. Dostupné z: <https://thingsboard.io/>

[16] About Apache HTTP Server Project. *Apache HTTP Server Project* [online]. [cit. 2023-05-18]. Dostupné z: https://httpd.apache.org/ABOUT_APACHE.html

[17] What is Docker?. *IBM* [online]. [cit. 2023-05-18]. Dostupné z: <https://www.ibm.com/topics/docker>

[18] KOLADE, Chris. What is HTML: Definition and Meaning of Hypertext Markup Language. *FreeCodeCamp* [online]. 24.8.2021 [cit. 2023-05-18]. Dostupné z: <https://www.freecodecamp.org/news/what-is-html-definition-and-meaning/>

[19] HTML basics. *Mdn* [online]. May 2023 [cit. 2023-05-18]. Dostupné z: https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics

[20] CSS: Cascading Style Sheets. *Mdn* [online]. 15.4.2023 [cit. 2023-05-18]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/CSS>

[21] What is PHP?. *Php* [online]. [cit. 2023-05-17]. Dostupné z: <https://www.php.net/manual/en/intro-what-is.php>

[22] What can PHP do?. *Php* [online]. [cit. 2023-05-17]. Dostupné z: <https://www.php.net/manual/en/intro-whatcando.php>

[23] , nehaahlawat. How to create admin login page using PHP?. *GeeksforGeeks* [online]. [cit. 2023-05-17]. Dostupné z: <https://www.geeksforgeeks.org/how-to-create-admin-login-page-using-php/>

[24] *Using OAuth 2.0 to Access Google APIs* [online]. 6.3.2023 [cit. 2023-05-17]. Dostupné z: <https://developers.google.com/identity/protocols/oauth2>

[25] What is a REST API?. *IBM* [online]. [cit. 2023-05-18]. Dostupné z: <https://www.ibm.com/topics/rest-apis>

[26] STM32 Wireless MCUs. *STMicroelectronics* [online]. [cit. 2023-05-18]. Dostupné z: <https://www.st.com/en/microcontrollers-microprocessors/stm32-wireless-mcus.html>

- [27] ARM STM32 NUCLEO-L053R8 Nucleo-64. In: *LaskaKit* [online]. [cit. 2023-05-20]. Dostupné z: https://www.laskakit.cz/arm-stm32-nucleo-l053r8-nucleo-64-vyvojova-deska/?gclid=EAIaIQobChMIYXDq8qD_wIV2vd3Ch1eVAQmEAQYAYABEgIIIID_BwE
- [28] SANTOS, Sara. ESP32 vs ESP8266 – Pros and Cons. *Maker Advisor* [online]. 6.9.2021 [cit. 2023-05-18]. Dostupné z: <https://makeradvisor.com/esp32-vs-esp8266/>
- [29] ESP32 Development Board. In: *GRobotronics* [online]. [cit. 2023-05-20]. Dostupné z: <https://grobotronics.com/esp32-development-board-devkit-v1.html?sl=en>
- [30] ČSN EN 1143-1. *Bezpečnostní úschovné objekty — Požadavky, klasifikace a metody zkoušení odolnosti proti vloupání: Část 1: Skříňové trezory, ATM trezory, trezorové dveře a komorové trezory*. Praha: Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2022.
- [31] ČSN 91 6012. *Bezpečnostní úschovné objekty — Požadavky, klasifikace a metody zkoušení odolnosti proti vloupání: Trezory se základní bezpečností*. Praha: Český normalizační institut, 2001.
- [32] ČSN EN 1300. *Bezpečnostní úschovné objekty — Klasifikace zámků s vysokou bezpečností vzhledem k jejich odolnosti proti nepovolenému otevření*. Praha: Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2020.
- [33] ČSN EN 1627. *Dveře, okna, lehké obvodové pláště, mříže a okenice – Odolnost proti vloupání: Požadavky a klasifikace*. Praha: Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2022.
- [34] MCFARLANE, Robert. Uninterruptible power supply (UPS). *TechTarget* [online]. May 2022 [cit. 2023-05-18]. Dostupné z: <https://www.techtarget.com/searchdatacenter/definition/uninterruptible-power-supply>
- [35] WeMos D1 R32 UNO ESP32. *LaskaKit* [online]. [cit. 2023-05-18]. Dostupné z: <https://www.laskakit.cz/wemos-d1-r32-uno-esp32/>
- [36] MEAN WELL RS-75-12: spínaný zdroj pro vestavbu krytý. *GME* [online]. [cit. 2023-05-18]. Dostupné z: <https://www.gme.cz/v/1507474/mean-well-rs-75-12-spinany-zdroj-pro-vestavbu-kryty>
- [37] GM201: 12V elektromagnetický zámek. *GME* [online]. [cit. 2023-05-18]. Dostupné z: <https://www.gme.cz/v/1509871/gm201-up-12v-elektromagneticky-zamek>
- [38] Relé modul 12V. *GME* [online]. [cit. 2023-05-18]. Dostupné z: <https://www.gme.cz/v/1507722/rele-modul-12v-2x-10a-250v-high-low-trigger>
- [39] Bezdrátový N300 Multi-WAN router: DWR-116. *D-Link* [online]. [cit. 2023-05-22]. Dostupné z: <https://eu.dlink.com/cz/cs/products/dwr-116-wireless-n300-multi-wan-router>

[40] *XAMPP Apache + MariaDB + PHP + Perl* [online]. Apache Friends [cit. 2023-05-18]. Dostupné z: <https://www.apachefriends.org/index.html>

[41] Raspberry Pi 4 Model B. *RPishop* [online]. [cit. 2023-05-20]. Dostupné z: <https://rpishop.cz/raspberry-pi-4/2611-raspberry-pi-4-model-b-8gb-ram-0765756931199.html>

[42] Raspberry Pi 4. *Raspberry Pi* [online]. [cit. 2023-05-18]. Dostupné z: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>

[43] HOUSE, Chris. A Complete Guide to CSS Grid. *CSS-TRICKS* [online]. 12.5.2021 [cit. 2023-05-17]. Dostupné z: <https://css-tricks.com/snippets/css/complete-guide-grid/>

[44] *MySQL* [online]. [cit. 2023-05-18]. Dostupné z: <https://www.mysql.com/>

[45] *PlatformIO* [online]. [cit. 2023-05-18]. Dostupné z: <https://platformio.org/>