

Zobrazení hlášení z PZTS na obrazovce pomocí Raspberry Pi

Tomáš Hulín

Bakalářská práce
2024



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav bezpečnostního inženýrství

Akademický rok: 2023/2024

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Tomáš Hulín**
Osobní číslo: **A21305**
Studijní program: **B1032A020001 Bezpečnostní technologie, systémy a management**
Forma studia: **Kombinovaná**
Téma práce: **Zobrazení hlášení z PZTS na obrazovce pomocí Raspberry Pi**
Téma práce anglicky: **Displaying messages from I&HAS control panel on screen using Raspberry Pi**

Zásady pro vypracování

1. Provedte rešerši PZTS a doplňkových zařízení.
2. Zpracujte programování ústředny PZTS.
3. Popište obecně využití Raspberry Pi k zobrazování informací.
4. Navrhněte Raspberry Pi k přijímání a vyhodnocení signálů z ústředny PZTS.
5. Vytvořte funkční zapojení PZTS s Raspberry Pi.
6. Zhodnotte možnosti tohoto řešení.

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

Seznam doporučené literatury:

1. LUKÁŠ, Luděk. Bezpečnostní technologie, systémy a management. Zlín: Radim Bačuvčík – VeRBuM, 2012. ISBN 978-80-87500-19-4
2. LUKÁŠ, Luděk. Bezpečnostní technologie, systémy a management. Zlín: Radim Bačuvčík – VeRBuM, 2013. ISBN 978-80-87500-35-4
3. LOVEČEK, Tomáš. Bezpečnostné systémy: bezpečnosť informačných systémov. Žilina: Žilinská univerzita v Žiline, 2007. ISBN 978-80-8070-767-5.
4. VALOUCH, Jan. Projektování bezpečnostních systémů. Zlín: Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, 2019. ISBN 978-80-7454-858-
5. DRGA, Rudolf. Speciální technologie komerční bezpečnosti. Zlín: Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, 2012. ISBN 978-80-7454-146-9.

Vedoucí bakalářské práce: **Ing. Rudolf Drga, Ph.D.**
Ústav bezpečnostního inženýrství

Datum zadání bakalářské práce: **8. prosince 2023**

Termín odevzdání bakalářské práce: **28. května 2024**

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



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

Ve Zlíně dne 8. prosince 2023

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 26.5.2024

Tomáš Hulín v. r.

ABSTRAKT

Bakalářská práce se zabývá využitím počítače Raspberry Pi v bezpečnostním systému pro zobrazení hlášení. Vysvětluje, jak funguje ústředna PZTS, k ní připojená zařízení a jak lze přes programovatelné výstupy systém dále rozšiřovat. V praktické části pak ukáže aplikaci zapojení s Raspberry Pi a zhodnotí možnosti tohoto řešení.

Klíčová slova: RaspberryPi, PZTS, PGM, PGIO, Display

ABSTRACT

This bachelor's thesis examines the use of Raspberry Pi in security systems for displaying notifications. The theoretical part of the thesis describes the principle of operation of the I&HAS control unit, the devices connected to it, and the possibilities of expanding the system via programmable outputs. The practical part of the thesis demonstrates the application of connecting the Raspberry Pi to the security system and evaluates the possibilities of this solution.

Keywords: RaspberryPi, PZTS, PGM, PGIO, Display

Rád bych poděkoval vedoucímu mé bakalářské práce, doktoru Rudolfovi Drgovi, za podnětné rady a odbornou pomoc, kterou mi poskytoval při zpracovávání mé bakalářské práce a za čas, který mi věnoval.

Cílem života by měla být slova velkého myslitele: „*moudrý člověk se učí od všech*“.

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

OBSAH

ÚVOD.....	8
I TEORETICKÁ ČÁST.....	9
1 POPLACHOVÉ ZABEZPEČOVACÍ TÍŠŇOVÉ SYSTÉMY.....	10
1.1 ČÁSTI PZTS	11
1.1.1 Periferie	11
1.1.2 Ústředna	12
1.1.3 Ovládací jednotka.....	14
1.1.4 Napájení	14
1.1.5 Výstupní moduly	15
1.1.6 Moduly pro komunikaci.....	15
1.2 PROGRAMOVÁNÍ ÚSTŘEDNY	16
1.2.1 Rozsah	17
1.2.2 Sekce	17
1.2.3 Periferie	18
1.2.4 Uživatelé	18
1.2.5 PG výstupy	19
1.2.6 Reporty uživatelům	20
1.2.7 Parametry	20
1.2.8 Kalendář	21
1.2.9 Diagnostika	21
1.2.10 Komunikace	21
1.2.11 PCO	23
2 RASPBERRY PI.....	24
2.1 SPECIFIKACE	25
2.2 GPIO.....	26
2.3 SYSTÉM.....	27
2.3.1 Raspberry Pi OS	28
2.3.2 OSMC	28
2.3.3 Ubuntu.....	29
2.3.4 DietPi.....	29
II PRAKTICKÁ ČÁST	30
3 ZOBRAZENÍ INFORMACÍ POMOCÍ RASPBERRY PI.....	31
3.1 KOMERČNÍ APLIKACE	31
3.2 WEBOVÁ KOMUNIKACE.....	32
3.2.1 Backend.....	32
3.2.1.1 Ruby on Rails.....	32
3.2.1.2 Node.js	33
3.2.2 Frontend	34
3.2.2.1 React	34
3.2.2.2 Angular	34
4 FUNKČNÍ ZAPOJENÍ	36
4.1 KOMUNIKACE Z ÚSTŘEDNY DO RASPBERRY PI.....	36
4.1.1 Ústředna pošle signál	37
4.1.1.1 Periferie pošle ústředně.....	37

4.1.1.2	Ústředna vyhodnotí.....	37
4.1.2	Raspberry Pi přijme signál.....	39
4.1.2.1	Rozdělení pinů.....	39
4.1.2.2	Definice logické úrovně.....	40
4.1.2.3	Čtení na pinu.....	41
4.2	ZOBRAZENÍ INFORMACE NA OBRAZOVCE	41
4.2.1	Vysílající.....	41
4.2.2	Sledující.....	48
ZÁVĚR		51
SEZNAM POUŽITÉ LITERATURY.....		53
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....		57
SEZNAM OBRÁZKŮ		59
SEZNAM PŘÍLOH.....		60

ÚVOD

Bezpečnostní systémy se nachází všude kolem nás. Setkáváme se s nimi denně ať už na našem pracovišti nebo přímo v domácnosti. Jeden z nich zvaný PZTS, neboli poplachové zabezpečovací a tísňové systémy, mě zaujal ke zpracování. Jde o systém pro včasné varování na nežádoucí vniknutí nebo pokusu o vniknutí do střeženého prostoru nebo nežádoucí činnosti narušitele, chrání majetek a osoby a v případě nouze přivolá pomoc. Zařízení je také určeno k přenosu poplachových stavů do místa s trvalou ostrahou. Pomocí ústředny lze komunikovat s dohledovým centrem a ti už dále buď pomocí připojených detektorů nebo vyslané zásahové jednotky zjistí stav a zakročí. Při lokálním zjištění narušení lze pomocí tabla zjistit, o jaký sektor a detektor se jedná, který hlásí narušení. Tabla obvykle zobrazují zprávy pomocí LCD obrazovek, či dotykových panelů. Autorovou otázkou však je, co když jsem v objektu, když dojde k narušení klidu a nevím, odkud jde nebezpečí. Ústředna PZTS je vybavena moduly PGM neboli programovatelných výstupů. S nimi lze ústřednu PZTS naprogramovat k určitému zachování, podle vyhodnocení situace. Jednou z těchto možností je zobrazení na tablu, jak bylo již dříve psáno, kde došlo k narušení. Pomocí LCD obrazovky ukáže, o který detektor se jedná. Tato práce se zabývá rozšířením této možnosti. Za použití počítače schopného vyhodnocovat vstupní hodnoty a dle toho na výstupu promítat na tablo či obrazovku námi chtěné hodnoty. Jedním z těchto zařízení je minipočítač Raspberry Pi od společnosti Raspberry Pi Foundation z Velké Británie. Disponuje možností připojení vedlejších zařízení, které díky programování dokáže vyhodnotit a dále s nimi pracovat. Jedním z uvažovaných řešení je jeho další napojení na obrazovky v objektu. Jelikož se jedná o počítač, dokáže plně pracovat se svojí síťovou kartou a tím rozšiřovat svoji funkčnost při zapojení do sítě. Otázkou této práce je tedy, zdali jde zapojit do bezpečnostního systému, jmenovitě ústředny PZTS, pomocí PGM minipočítač k promítání varovných zpráv a hlášení na displaye a případně tabla přímo v objektu.

I. TEORETICKÁ ČÁST

1 POPLACHOVÉ ZABEZPEČOVACÍ TÍŠŇOVÉ SYSTÉMY

PZTS neboli poplachové, zabezpečovací a tísňové systémy se zabývají detekcí narušení či narušitele, vyhodnocení a následného vyhlášení poplachu. Jejím cílem není zadržet, či zabránit pachateli fyzicky v činnosti, ale slouží k zaznamenání události, vyhodnocení její skutečnosti, zdali se nejedná o falešný poplach a odeslání následné zprávy dál do systému. [1]

Mnohdy je tento systém znám pod svojí anglickou zkratkou I&HAS – Intrusion and Hold-up Alarm Systém(s). [1]

Poplachový zabezpečovací tísňový systém je kombinovaný systém určený k detekci poplachu vniknutí a tísňového poplachu. Poplachový systém může mít také funkci pouze jako poplachový zabezpečovací systém nebo poplachový tísňový systém. PZS neboli Poplachový zabezpečovací systém slouží k detekování a indikaci přítomnosti, vniknutí nebo pokusu o vniknutí vetřelce do střeženého prostoru. Anglicky je značen jako IAS – Intrusion Alarm System. Zkratkou PTS se rozumí poplachový tísňový systém. Ten poskytuje možnost uživateli úmyslně vyvolat poplachový stav. V angličtině se s ním setkáváme pod zkratkou HAS – Hold-up Alarm Systém. [2]

Jak rozsáhlý systém se do objektu hodí nám určuje bezpečnostní systém, do kterého ho zařadíme. Dělí se na 4 stupně:

- stupeň zabezpečení 1 pro nízké riziko,
- stupeň zabezpečení 2 pro nízké až střední riziko,
- stupeň zabezpečení 3 pro střední až vysoké riziko,
- stupeň zabezpečení 4 pro vysoké riziko.

Do stupně 1 a 2 by se daly přiřadit rodinné domy a byty, případně sídla firem. Do stupně 3 už by se řadily místa s větším rizikem hrozeb, jako jsou třeba směnárny nebo zlatnictví. Do stupně 4 patří strategicky důležitá místa – tiskárna cenin, jaderné elektrárny. [3]

Odlišným dělením se na stupeň 1 přihlíží, že vetřelec nebo lupič mají malou znalost bezpečnostního systému a mají k dispozici omezený sortiment snadno dostupných nástrojů. U stupně 2 se předpokládá, že vetřelec nebo lupič mají omezenou znalost bezpečnostního systému a používají běžné nářadí a přenosné přístroje. Stupeň 3 se dívá na situaci, že vetřelec nebo lupič jsou s místním bezpečnostním systémem obeznámeni a mají rozsáhlý sortiment nástrojů a přenosných elektronických zařízení. Stupeň 4 je využíván v situacích, kdy

zabezpečení má prioritu nad všemi ostatními hledisky. Předpokládá se, že vetřelec nebo lupič mají kompletní sortiment zařízení. [1]

1.1 Části PZTS

System PZTS se dá obecně popsat těmito prvky:

- periferie,
- ústředna,
- ovládací jednotka,
- napájení,
- výstupní moduly,
- moduly pro komunikaci.

1.1.1 Periferie

Jde o jednotlivé prvky v systému. Od detektorů po různá tísňová tlačítka. Cokoliv, s čím ústředna pracuje a dokáže podle toho vyhodnotit změnu stavu, vyvolat poplach. Jednotlivé periferie jsou umístěny do objektu dle potřeb klienta k zajištění bezpečí majetku a osob.

Její součástí může být i Tamper a Anti-masking.

- Tamper – slouží k vyhodnocení situace, zdali nedošlo k sabotáži narušením úschovné schránky (pokud má) či odtrhnutím od jeho místa.
- Anti-masking – pomocí vln nebo záření monitoruje, zda byl detektor zakryt či přestříkán za účelem zamezení detekování prostředí.

[4]

Detektory nám slouží k monitorování či odhalení určitých skutečností. Mezi nejčastěji používané patří:

- Infračervené bariéry – tato technologie pracuje s infračerveným zářením. Infračervený paprsek je vyzařován z vysílače ve tvaru kuželovitého paprsku s úhlem rozsvícení 2° a na straně přijímače je signál přijímán a vyhodnocován. Vyhodnocovací jednotka porovnává úroveň a dobu přerušování paprsku i jejich počet a na základě nastavených parametrů aktivuje poplachové relé.
- Mikrovlnné bariéry – tato technologie pracuje s proudem mikrovln vyzařovaného mezi vysílačem a přijímačem. Mikrovlnné vlnění je vyzařováno ve tvaru elipsoidu

(doutníku), jeho tvar a velikost je definován typem antény a vzdáleností mezi vysílačem a přijímačem.

- Detektory tříštění skla – pracuje akusticky. Detekuje dvě frekvence vzniklé při porušení skla. Nízkofrekvenční vlnu nárazu a vysokou frekvenci tříštění.
- Magnetické detektory – najdeme je třeba v rámu dveří a zárubni. Při otevření dojde k narušení magnetického pole v detektoru.
- Infračervené detektory – pracují s infračerveným zářením podobně jako bariéry. Pozorují rozdíl snímaného infračerveného záření v monitorovaném prostředí.
- Mikrovlnné detektory – podobně jako jejich bariéry pracují s mikrovlnami. Mikrovlny jsou vyzařovány do prostředí a je monitorován jejich odraz.

[5, 6]



Obrázek 1 – JA-110P PIR detektor [7]

1.1.2 Ústředna

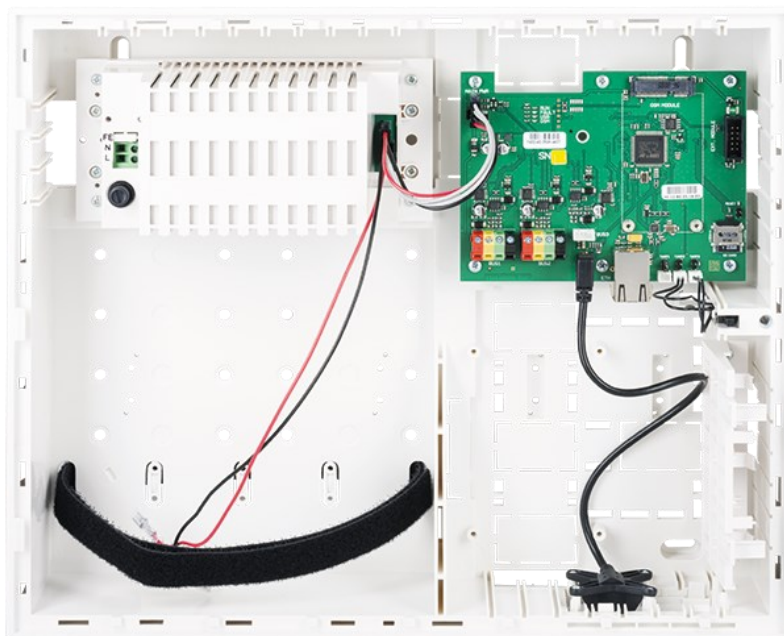
Jedná se o mozek celé operace. Jsou na ni zapojené detektory, pomocí kterých samotná ústředna vyhodnocuje aktuální stav objektu a jestli došlo k jeho narušení. Podle toho pak dále předává zprávy. V závislosti na stupni zabezpečení ústředny musí mít prostředky k přijetí a zpracování signálů nebo zpráv z detektorů. Při zajištění vyhodnocených zpráv je pak záznam prováděn přímo na ústřednu nebo PCO neboli Pult Centrální Ochrany. [8]

Detektory jsou na ústřednu napojeny buď drátově nebo bezdrátově.

- Drátově – pomocí smyček jsou detektory napojené na ústřednu. Rozlišujeme NC – Normaly Closed a NO – Normaly Open
 - NC – v klidovém stavu je smyčka uzavřená. Poplach je vyvolán otevřením smyčky a zpozorovaném nárůstu odporu na smyčce.
 - NO – v klidovém stavu se smyčka nachází v otevřeném stavu a poplach je vyvolán uzavřením smyčky. Tím dojde ke snížení odporu, který pozorujeme.
- Bezdrátově – detektory obsahují baterie a komunikace s ústřednou je pomocí vln. Máme zapojení typu hvězda nebo routovací.
 - Hvězda – každý detektor komunikuje přímo s ústřednou.
 - Routovací – detektory spolu komunikují v cestě. Postupně se buduje mapa, kdy každý detektor komunikuje s detektory v dosahu a pokud může, tak i s ústřednou, které předává informace o ostatních, se kterými komunikuje.

Detektory jsou sami o sobě napojené na napájení, ale také na smyčku. V detektoru jsou tyto kontakty oddělené – tzv. galvanicky oddělené, aby nedošlo ke vzájemnému rušení.

[4]



Obrázek 2 – JA-107K ústředna [9]

1.1.3 Ovládací jednotka

Pomocí klávesnice, dotykové obrazovky či konektoru zde dochází ke komunikaci přímo s ústřednou. Lze zde zadávat kódy, přihlašovat uživatele s oprávněními, programovat ústřednu. Slouží pro zastřežení nebo odstřežení objektu a další uživatelské funkce zabezpečovacího systému. [10]



Obrázek 3 – JA-155E-AN Klávesnice s displejem a RFID čtečkou [11]

1.1.4 Napájení

O napájení prvků PZTS se stará zdroj. I přes výpadek proudu se musí zaručit funkčnost bezpečnostního systému. Tím se zabývá záložní zdroj. Rozlišujeme zdroje:

- samostatný s nezávislými napájecími zdroji,
- integrovaný alespoň s jednou další součástí PZTS, s nezávislými napájecími výstupy připojenými k PZTS a s alespoň jedním napájecím výstupem určeným pro integrovanou část,
- integrovaný alespoň s jednou další součástí PZTS, ale bez nezávislých napájecích výstupů připojených k ostatním součástkám PZTS.

[12]

1.1.5 Výstupní moduly

Zde vstupují na řadu veškeré moduly, které máme po vyhodnocení zprávy. Tyto moduly umožňují různé formy reakcí a interakcí na základě výsledků vyhodnocení. Mezi tyto moduly patří například audio a vizuální signalizace nebo programovatelné výstupy, díky kterým dokážeme realizovat zaměření této bakalářské práce.

1.1.6 Moduly pro komunikaci

Ústředna vyhodnocuje zprávy zachycené detektory. Pokud vyhodnotí, že hrozí nebezpečí, tak odešle zprávu na PCO nebo přímo klientovi.

Jedná se třeba o GSM nebo LTE komunikátory. Každý využívající příslušnou technologii a komunikuje pomocí mobilní sítě s pultem centrální ochrany, umožňuje vzdálené ovládání pomocí webové a mobilní aplikace, předává poplachové SMS a hlasové zprávy. Zároveň umožňuje vzdálené nastavování systému. [13]

Dále se dá využít drátové komunikace, či pomocí tónové volby přes pevnou telefonní síť.



Obrázek 4 – JA-192Y Modul GSM komunikátoru [14]

1.2 Programování ústředny

Součástí zprovoznění systému je úkon jejího programování. Každý výrobce k tomu používá svoji aplikaci a postupy.

Obecně se jedná o „seznámení“ ústředny se systémem, do kterého je zapojena, a co bude řídit. Definuje se zde, na jaké části máme objekt rozdělen neboli na kolik sekcí. Přímo pod sekcemi se nachází jednotlivé periferie. Dále co za uživatele se bude v systému pohybovat, jaké zprávy se jim budou posílat a kam. Posledně případné programovatelné výstupy či nastavení k osobním potřebám klienta.

Ke zkonkretizování popíší programování ústředny od firmy Jablotron. Využívá programu F-Link, ve kterém lze nejen nastavit ústřednu, avšak ji i ovládat – ať už lokálně tak vzdáleně. Systém JABLOTRON 100+ se nastavuje výhradně počítačem, pomocí programu F-Link.

Po spuštění program nabídne programovat pomocí spojení:

- a) lokálního – počítač je s ústřednou propojen kabelem USB
- b) dálkového – nabídne výběr ústředny z uložených souborů a umožní navázat dálkové spojení. K navázání vzdálené komunikace s ústřednou je nutný přístup počítače k internetu a v ústředně instalován GSM komunikátor s osazenou SIM kartou v a musí mít aktivované datové přenosy dat GPRS. K bezproblémovému spojení musí být splněny další požadavky jako např. povolená vzdálená konfigurace v ústředně, správný registrační kód, znalost servisního kódu, ale také dostatečný GSM signál v místě ústředny nebo připojení k síti LAN
- c) offline nastavení – umožní přístup k informacím o nastavení ústředny. Lze se tak dostat např. k seznamu periférií nebo k poznámkám o poslední výměně baterií apod.

[15]

Jednotlivé úkony programování, jak byly již v předešlém odstavci popsány, program rozlišuje na jednotlivé záložky:

- rozsah,
- sekce,
- periferie,
- uživatelé,
- PG výstupy,
- reporty uživatelům,

- parametry,
- diagnostika,
- komunikace,
- PCO.

[15]

1.2.1 Rozsah

Zde nastavujeme základní hodnoty systému. Nastavené údaje lze kdykoliv změnit. Hodnoty rozsahu ovlivňují velikost databáze, a tím i čas pro načítání a ukládání dat. V programu volíme, kolik sekcí má ústředna spravovat, kolik periférií k ní je připojeno, kolik uživatelů v systému a kolik programovatelných výstupů bude. Dále se tu volí funkce *Kódy s prefixem*. Určuje způsob zadávání všech přístupových kódů při autorizaci uživatelů. Při zapnuté funkci systém vyžaduje před zadáním přístupového kódu ještě zadání jedno až třímístného pořadového čísla kódu (prefix) zakončeného hvězdičkou. Díky tomu si uživatelé mohou, při přihlášení s prefixem, z LCD klávesnice i svůj kód změnit. Při vypnuté funkci musí veškeré kódy přidělovat a měnit výhradně Správce systému, který se musí vyhnout situaci, kdy by dva uživatelé měli nastavený stejný kód. Volí se tu také délka kódu. Slouží nám ke zvýšení bezpečnosti systému. Můžeme zvolit délku kódu na 4, 6 až 8 míst. Pokud dojde ke změně délky po instalaci, smažou se všechny zapamatované. Jediné kódy, které zůstanou, jsou kódy servis a správce. Servis má při čtyřmístném kódu 1010, s každou dvojicí se přidává další 10 a správce má posloupnost začínající 1. takže správce při čtyřmístném má 1 až 4. [15]

1.2.2 Sekce

Zde nastavujeme vlastnosti jednotlivých střežených sekcí. Počet odpovídá číslu, které jsme zadali o krok před tímto. Dáváme jim jméno k lepšímu zorientování – to se pak ukazuje na display LCD klávesnic společně se stavem, ve kterém se nyní nachází. [15]

Umožňuje nastavení společných sekcí. Vhodné pro využití na chodbách, schodištích a společenských prostor. Ovládání společných sekcí probíhá naprosto automaticky. Společná sekce se automaticky zajišťuje ve chvíli, kdy dojde k zajištění všech přidružených sekcí v nižší úrovni a automaticky odjišťuje, když dojde k odjištění alespoň jedné sekce z této nižší úrovně. Vezmeme si za příklad tři sekce A, B a C. Sekce A a B lze zajišťovat nezávisle, sekce C je společná a hlídá pouze, jsou-li sekce A a B obě zajištěny. [15]

Dává možnost kontrolovat a monitorovat sekce. Pokud je sekce odjištěná a nedojde v ní k aktivaci žádného detektoru během nastavené doby, funkce *Hlásit nezajištění* odešle hlášení. Pokud k tomu dojde, lze umožnit automatické zajištění po určitém časovém intervalu od odeslání hlášení. Tato funkce lze spustit pouze pokud je zapnuté Hlášení nezajištění. Omezení přístupu funguje na principu umožnění odjištění sekce pomocí kalendáře určitým uživatelům. Částečné střežení zajistí sekci, ale bude ignorovat hlášení periferií s nastavením „vnitřní“.[15]

Nastavení systému HISTORIE						
Přihlášen: Správce Master						
Sekce		Uživatelé	PG výstupy	Diagnostika	Kalendář	
Pozice	Název sekce	Společná sekci	Časově omezený přístup	Stav	Poznámka	
1	Section 1	Ne	Ne	Offline		
2	Section 2	Ne	Ne	Offline		

Obrázek 5 – Záložka Sekce v programu F-Link [vlastní zpracování]

1.2.3 Periferie

Zde zapisujeme jednotlivé připojené periferie do systému a nastavujeme k nim jejich vlastnosti. Počet periferií zase odpovídá číslu zadanému v sekci *Rozsah*. Ústředna se automaticky přiřadí na pozici číslo 0, zbytek pokračuje číselnou posloupností. Každá má jméno, zase k lepší orientaci a ukazuje i stav. Je zařazena pod svoji sekci, kam hlásí události. Můžeme zde nastavit jednotlivé periferii její reakci při aktivaci. Od okamžité, po zpožděnou či úplně žádnou. Zde přiřazujeme i případnou aktivaci programovatelného výstupu, který využíváme k realizaci této práce. Vlastnost vnitřní lze přisoudit periferiím k detekci vloupání. Pokud bude sekce částečně zastřežena, sekce bude ignorovat hlášení takové periferie. [15]

1.2.4 Uživatelé

V této záložce se setkáváme s jednotlivými uživateli. Počet pozic databáze odpovídá číslu zadaném v *Rozsah*. Pojmenování slouží nejen k orientaci, ale i v textových hlášeních událostí. U uživatele může být i telefonní číslo, sloužící k identifikaci ale i ke kontaktování pomocí prozvonění či SMS při události nebo spuštění programovatelného výstupu. Čísla se musí zadávat v mezinárodním tvaru, takže i s předčíslem. Nachází se zde i kód uživatele, se kterým pracuje při událostech a slouží k jeho autorizaci. Pokud systém používá RFID čtečku, je zde zapsáno číslo karty nebo čipu, kterou uživatel používá. Připisujeme zde jednotlivé oprávnění – jestli spadá pod nějaké časové omezení, které se řídí pomocí záložky Kalendář

nebo jakou sekci dokáže zajistit a odjistit. Také se kterými programovatelnými výstupy dokáže osoba operovat. Velikost této databáze odpovídá číslu zadaném v *Rozsah*. [15]

Nastavení systému HISTORIE											
Přihlášen: Správce Master											
Sekce	Uživatelé	PG výstupy		Diagnostika	Kalendář						
Pozice	Jméno	Logika	Funkce	Čas	Aktivace	Blokování PG	Reporty	Záznam PG do paměti	Vypnutí	Aktuální stav	Poznámka
1	PG output 1	Spínací	Vypnuto		Aktivace	Žádné	Vstupit	<input type="checkbox"/>			
2	PG output 2	Spínací	Vypnuto		Aktivace	Žádné	Vstupit	<input type="checkbox"/>			
3	PG output 3	Spínací	Vypnuto		Aktivace	Žádné	Vstupit	<input type="checkbox"/>			

Obrázek 6 – Záložka Uživatelé v programu F-Link [vlastní zpracování]

1.2.5 PG výstupy

Každá ústředna má pevný počet programovatelných výstupů. Ty se dají sběrnice či bezdrátovým komunikátorem rozšířit. V *Rozsah* pak vypíšeme, kolik jich osobně potřebujeme.

Jednotlivé výstupy pojmenujeme, přisoudíme jim, zda se jedná o spínací či rozepínací logiku a jakou funkci výstup bude mít po aktivaci. Dále ohlašování při změně stavu a ukazuje aktuální stav. Lze zde nastavit blokování některého z výstupů událostí, stavem sekce nebo další programovatelným výstupem. Zapíná se zde i záznam aktivity programovatelného výstupu do paměti událostí. Dále se zde dá jednotlivý výstup otestovat, k zajištění správnosti funkčnosti. [15]

Součástí je pak i mapa aktivačních vazeb jednotlivých programovatelných výstupů. Ta nám říká, na jaké podněty výstup reaguje. Nastavuje, kteří uživatelé jsou oprávněni ovládat PG výstup, který má nastavený požadavek autorizace. Dovoluje určit až dvě klávesnice, které mohou výstup aktivovat pouhou autorizací – to funguje pouze pokud je výstup nastaven na impulz. Umožňuje oprávnit uživatele, kteří dokáží spustit programovatelný výstup prozvoněním svým telefonním číslem. Pojem “prozvonění” znamená, že volající po vytočení telefonního čísla systému vyčká nejméně jednoho vyzváněcího tónu (maximálně však do nastavení vyzvednutí hovoru ústřednou) a ukončí volání. PG reaguje na položení volání. Pokud dojde k vyzvednutí hovoru ústřednou, nedojde k aktivaci výstupu. Umožňuje aktivovat výstup vybraným interním stavem systému. nastavit textové povely pro zapnutí a vypnutí PG výstupu telefonem. Doručení příslušné SMS má podobný účinek jako stisk tlačítka Zapni či Vypni na ovládacím segmentu klávesnice. Pro ovládání výstupů použijte tvar SMS kód(mezera)příkaz. Kód není třeba před příkazem vypsát, pokud je v záložce *Komunikace* povoleno

ovládání pomocí SMS bez kódu a bude identifikováno telefonní číslo uživatele s oprávněním na ovládání příslušného PG výstupu. [15]

F	Jméno	Telefonní číslo	Kód	Karta	Oprávnění	Vzor	Povolit zm...	Časové o...	Sekce	PG	Report...	Prozvon...	Vypnutí	Poznámka
0	Service		****	0	Servis	Ne	<input checked="" type="checkbox"/>	Ne	1 až 15	1 až 128	<input checked="" type="checkbox"/>			
1	Master		****	0	Správce	Ne	<input checked="" type="checkbox"/>	Ne	1 až 15	1 až 128	<input checked="" type="checkbox"/>			
2	User 2			0		Ne			Ne	Ne	<input checked="" type="checkbox"/>			

Obrázek 7 – Záložka PG výstupy v programu F-Link [vlastní zpracování]

1.2.6 Reporty uživatelům

Přiřazujeme zde, komu budou chodit ohlašovací zprávy na změnu stavu zajištění/odjištění – hlášení o změně stavu zajištění se odesílá s pevně nastaveným zpožděním 60 sekund po zajištění. Dále říkáme, komu budou chodit zprávy o poruchách, popláchích. Pod skupinu *Poplach voláním* řadíme takové uživatele, kteří po odeslání SMS hlášení jsou ještě upozorněni hlasovou poplachovou zprávou. Ta zvoní 30 sekund. Pokud ji volaný nezvedne, hlasovou zprávu dostane další uživatel v pořadí. Zpráva se skládá z „Váš alarm hlásí – Typ poplachu – Sekce číslo“. Uživatel může potvrdit příjem volání stiskem klávesy # na telefonu a po hlasové výzvě musí zadat platný kód. Po zadání platného kódu se ukončí poplach a dalšímu uživateli se již nevolá. Zprávy mohou být vybrány přednastavené, ale je možnost si je i přemluvit dle svých potřeb. Jde zde vybrat, kterým uživatelům se bude posílat hlášení o spuštění či vypnutí programovatelného výstupu. [15]

1.2.7 Parametry

Nastavujeme tady parametry a volitelné funkce ústředny. Jaké je datum a čas, jak se bude čas seřizovat (jestli ručně, z mobilní sítě nebo ze serveru Jablotronu), jaký je časový posun, jestli je zimní nebo letní čas a upozornění, jestli je čas nastavený rozdílný na ústředně vůči počítači, se kterým programujeme ústřednu. Upravuje profily systému (podle výroby, dle normy EN50131-1 nebo INCERT), způsoby autorizace (jednoduchá, potvrzení karty kódem či dvojitá), případně různá blokování – jak režimů (údržba, servis, PCO), zajištění, tak i poplachem. Zapíná se tu funkce *Požadavek servisu*, která po 12 měsících od posledního ukončení režimu *Servis* způsobí v systému událost „Systém vyžaduje servisní prohlídku“, která se spolu s ikonou Informace zobrazí na klávesnicích s LCD displejem a zapíše se do historie událostí. Hlášení z displeje se vypne automaticky lokálním přístupem servisního technika do systému. Tím dojde k automatickému znovunastavení ročního počítadla. Řeší

se tu časovače a jak vůbec ústředna vyhodnocuje potvrzení poplachu. Lze tu nastavit potvrzení poplachu jiným detektorem zajištěné sekce, doba čekání na potvrzení požárního poplachu jiným detektorem a třeba doba čekání na opakování aktivace stejného detektoru. Nastavený čas musí být delší než Minimální zklidnění detektoru před opakováním. [15]

1.2.8 Kalendář

Jedná se o záložku zabývající se časovým programem akcí, které bude systém pravidelně a automaticky provádět. Říká, jaká akce se má provést nad sekcemi nebo PG výstupy (Odjisti, Zajisti, Zajisti částečně, ovládání PG, požadavek servisní prohlídky). [15]

Pořadí	Akce	Sekce/PG	Dny v týdnu	Dny v měsíci	Měsíce v roce	Časování	Blokování	Vypnutí	Poznámka
1	Ne	Ne	po, út, st, č...	1 až 31	1 až 12	Ne	Ne		
2	Ne	Ne	po, út, st, č...	1 až 31	1 až 12	Ne	Ne		
3	Ne	Ne	po, út, st, č...	1 až 31	1 až 12	Ne	Ne		

Obrázek 8 – Záložka Kalendář v programu F-Link [vlastní zpracování]

1.2.9 Diagnostika

Zde se dá kontrolovat a zjišťovat stav periférií a jejich vlastností. Obsahuje-li periferie baterii, zobrazuje se její stav. Pro ústřednu (pozice 0, stejně jako v záložce *Periferie*) se zobrazuje napětí zálohovacího akumulátoru. Na pozici ústředny se zobrazuje napětí na svorkách ústředny a proud, který je odebírán sběrníkovými perifériemi z ústředny (zobrazeno pro jednotlivé sběrnice). U sběrníkových periférií je zobrazen úbytek napětí na vedení vůči ústředně (úbytek nesmí být větší než 2V). U ústředny je i informace o kvalitě signálu GSM. [15]

1.2.10 Komunikace

Tady nastavíme výběr kanálu, kterým ústředna ohlašuje hlasové události, unikátní číslo ústředny, zda při ovládání z autorizovaného telefonu voláním nemusí uživatel zadávat svůj kód (autorizuje se tím, že volá ze svého telefonu). Výběr uživatele, kam budou přeposílány pro ústřednu nesrozumitelné SMS zprávy (informace od operátora o vyúčtování apod.). Funkce *SMS o neúspěšném zajištění* v případě neúspěšného zajištění odešle informační SMS. Pokud je zajištěno s autorizací, je zpráva odeslána danému uživateli, pokud je zajištěno bez autorizace, je SMS zaslána Správci (v uživateli se jedna o osobu na pozici 1). [15]

V systému lze zvolit i několik typů vzdálené komunikace či konfigurace:

- Žádná – chová se jako autonomní zařízení s vlastní SIM kartou. Zařízení komunikuje směrem ven (odesílá SMS a hlasové zprávy) i přijímá povelové SMS a má funkční hlasové menu. Nekomunikuje datově. Nelze uskutečnit vzdálené nastavování software F-Link.
- Omezená (GSM) – komunikuje jako předešlý typ, a navíc umožňuje vzdálené nastavení systému. Vzdálená konfigurace je možná z počítače s programem F-Link s internetovým připojením. Pro navázání spojení s ústřednou se F-Link spojuje se serverem výrobce a předává mu registrační kód a telefonní číslo SIM karty vložené v komunikátoru ústředny. V ústředně musí být funkční datová komunikace (LAN nebo GSM/GPRS).
- Trvalá (LAN) – ústředna udržuje trvalou datovou komunikaci (LAN) se serverem. Je možné připojení a správa ústředny SW F-Link.
- Komunikace Jablotron – zařízení komunikuje se serverem výrobce (aplikace MyJABLOTRON) a odesílá na něj průběžně aktuální stav zařízení. Při požadavku na vzdálené připojení F-Linkem je tak server připraven okamžitě navázat spojení. Tato komunikace dále umožňuje uživateli využít serverových služeb. Do mobilních zařízení se systémem Android, iOS (Apple) lze instalovat aplikace umožňující uživateli obsluhu systému. Při této volbě je nutné mít použitou Bezpečnostní SIM kartu Jablotron.

[15]

Také zde upravujeme parametry komunikací skrze jednotlivé komunikátory. U GSM můžeme nahlédnout a vidět sílu signálu, nastavení datové komunikace, limit rozsahu reálného volání na 5 až 250 minut za den, limit počtu odeslaných SMS z ústředny za den (vztahuje se na poplachové i nepoplachové události), limit poplachových SMS, možnost ovládat systém dálkově pomocí hlasového menu (je-li nastaveno na *Uživatelé*, lze do menu vstoupit pouze z telefonů nastavených uživatelů a i bez kódů, jak bylo dříve zmíněno v této části), počet vyzvánění do automatického vyzvednutí komunikátorem a třeba hlasitost generované DTMF (Dual-tone multi-frequency, komunikace na bázi osmi akustických frekvencí, kde jednotlivé znaky odpovídají kombinaci dvou). [15]

U LAN nastavení upravujeme sítíovou komunikaci – zdali si ústředna nastaví parametry automaticky ze serveru. Pokud to neumí, tak musíme adresu, masku podsítě, výchozí bránu vypsát sami. Při připojení LAN komunikátoru k internetu lze otestovat správnost nastavení

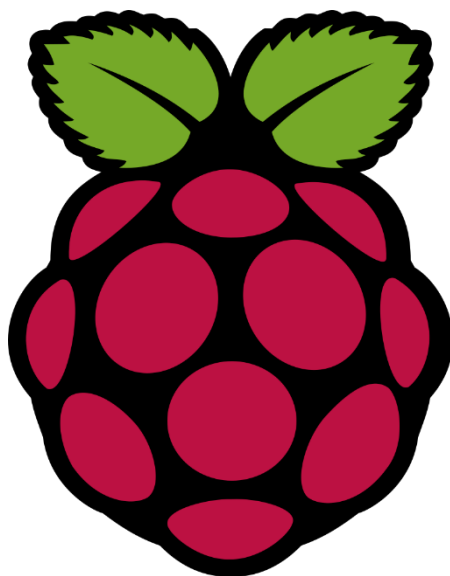
Tlačítkem DNS. To po pár vteřinách zasvítí zeleně, že se podařilo nebo červeně, že se nepodařilo navázat spojení. [15]

1.2.11 PCO

V poslední záložce definujeme vztah mezi ústřednou a pultem centrální ochrany. V případě volby *Komunikace Jablotron* (jak bylo zmíněno v kapitole *Komunikace*), kdy dochází k podstatnému zjednodušení nastavení komunikační části systému, je toto nastavení nedostupné. Volí se tu způsob přenosu dat na PCO. Možnosti voleb jsou: GSM, LAN a Automaticky, ale nabízí se jen ty, co jsou aktuálně dostupné. Volba Automaticky nastavuje kombinaci dříve zmíněných komunikátorů LAN a GSM. Primárně využívá LAN a až při jeho nedostupnosti se automaticky přechází na záložní komunikátor GSM. Vybírá se zde i typ přenášených zpráv, které se posílají na PCO. [15]

2 RASPBERRY PI

Co to je? Jedná se o počítač, který se vejde do lidské dlaně. Je přibližně velký, jako platební karta. Raspberry Pi Foundation byla založena v roce 2009 Ebenem Uptonem, Robem Mullinsem, Jackem Langem, Alanem Myclem a Petem Lomasem, kteří měli za cíl vyvinout cenově dostupný počítač, který by podpořil výuku informatiky ve školách. Její hlavní vizí je snaha rozšířit potenciál mládeže v digitálních technologiích a práci s počítačem. Díky nízkým nákladům a otevřenému přístupu k softwaru a hardwaru je Raspberry Pi přístupný pro širokou veřejnost, což podporuje inovaci a kreativitu ve všech věkových skupinách [16]



Obrázek 9 – Logo Raspberry Pi [17]

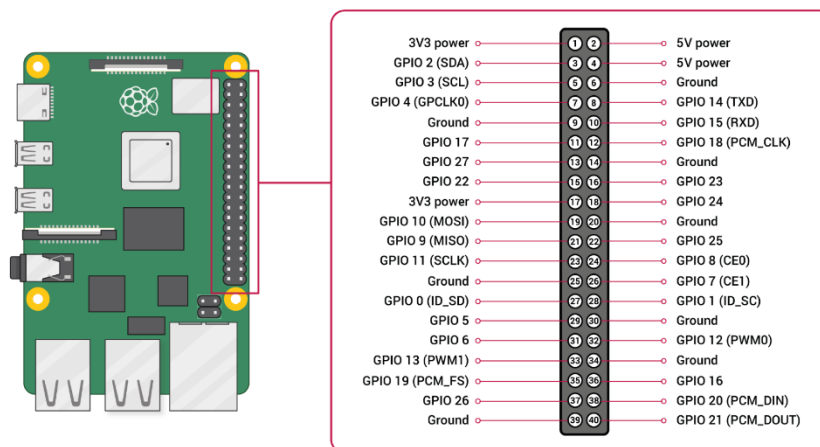
Myšlenka levného počítače pro mládež se začala rodit v roce 2006, když začali Eben Upton a jeho kolegové v počítačové laboratoři University of Cambridge pocítovat znepokojení nad stále klesajícím počtem a úrovní studentů hlásících se ke studiu počítačových věd. Zatímco v devadesátých letech se k tomuto studiu hlásili zkušení fandové programování, v roce 2000 už drtivou většinu tvořili ti, kteří byli sotva schopní vytvořit jednoduché stránky v HTML. Počítačové kurzy se také většinou omezovaly na ovládání Wordu, Excelu a špetky HTML. Mezi roky 2006 a 2008 proto začal vznikat několik prototypů. Po roce 2008 začaly být procesory vyvíjeny pro mobilní zařízení velmi výkonné a dostupné, což vedlo ke spuštění projektu. Série prvních deseti tisíc kusů byla vyrobena v Číně, stejně jako několik dalších výrobních dávek. V současné době je ale výroba přesunuta do Velké Británie, do továrny společnosti Sony ve Walesu. [18]



Obrázek 11 – Raspberry Pi Pico [21]

2.2 GPIO

Raspberry Pi rozvířilo vody na trhu pomocí svého GPIO – General Purpose Input/Output. GPIO je klíčovým prvem, který umožňuje Raspberry Pi komunikovat s okolním hardwarovým světem. Tento systém je tvořen celkem 40 piny, které slouží jako rozhraní pro různé externí zařízení. Zařzení, která využívají piny GPIO, jsou často charakterizována otevřeným přístupem. Tento přístup znamená, že uživatelé mají možnost modifikovat a modulovat zařízení podle svých potřeb. [22]



Obrázek 12 – Popis GPIO a jednotlivých pinů na Raspberry Pi [23]

Taková zařízení jsou navržena tak, aby byla co nejvíce přizpůsobitelná, což podporuje inovaci a kreativitu v oblasti hardwarových projektů. Většina elektrotechnických součástek, které se používají v obvodech, má piny. Piny jsou důležité proto, že umožňují snadnou integraci těchto součástek do elektronických obvodů. Součástky se pomocí pinů jednoduše zapojí do otvorů určených v obvodových deskách, což zjednodušuje proces montáže a zajišťuje stabilní elektrické spojení. GPIO na Raspberry Pi tedy nejenže poskytuje flexibilitu a rozšiřitelnost, ale také podporuje širokou škálu projektů, od jednoduchých experimentů až

po komplexní systémy. Díky tomu se Raspberry Pi stalo populárním nástrojem mezi nadšenci, studenty i profesionály v oblasti elektroniky a programování. [22]

V Raspberry Pi komunitě se často setkáváme s pojmem "Raspberry Pi Hat", známým také jako "klobouk". Tento název označuje malý hardwareový prvek, který se přímo nasazuje na GPIO (General Purpose Input/Output) pinový konektor Raspberry Pi a plní specifickou funkci. Hlavním účelem těchto klobouků je poskytnout jednoduché a přímé rozšíření funkcí Raspberry Pi. Klobouky jsou navrženy s ohledem na snadnou integraci a použití. Typicky mají stejný formát celých 40 pinů, což znamená, že je lze snadno připojit a odpojit. Některé z nejčastěji používaných klobouků zahrnují:

- ventilátorové klobouky – ty mají vestavěný ventilátor, který pomáhá chladit Raspberry Pi během intenzivních výpočtů nebo zátěže. Obvykle k nim existuje skript, který pomocí informačních pinů upravuje rychlost větráku podle teploty procesoru,
- rozšiřující klobouky – ty poskytují dodatečné GPIO piny nebo další rozhraní, jako jsou USB porty, sériové porty nebo konektory pro připojení periferií. To umožňuje snadné připojení a rozšíření funkcí Raspberry Pi bez nutnosti složitého drátování.
- bateriové klobouky – obsahují vestavěnou baterii nebo nabíjecí obvod, který umožňuje provoz Raspberry Pi v terénu nebo v místech, kde není dostupné stabilní napájení. Což je užitečné pro mobilní aplikace nebo projekty na venkovních místech,
- různé sensorové klobouky: existují klobouky vybavené různými senzory, jako jsou teplotní senzory, vlhkostní senzory, senzory pohybu nebo senzory světla. Tyto klobouky umožňují snadné přidání sensorických funkcí k Raspberry Pi projektům.

[24]

2.3 Systém

Raspberry Pi model A, B a Zero se od modelu Pico liší. Každý z nich má možnost fungovat jako počítač. Přímo na desce všech Raspberry Pi modelů (kromě modelu Compute, který je o velikosti RAM paměti a je potřeba ho zapojit na nějakou další desku) je místo pro micro SD kartu. Dřívější verze, jmenovitě modely první generace A a B s označením Raspberry Pi 1 Model A nebo B, měli slot pro SD kartu, ze kterého sešly. Lze přikoupit i jeden ze dříve zmíněných Hat modifikací, přidávající Sata sloty k připojení standardních disků se systémy. Pico se liší, jak bylo dříve zmíněno, že se řadí spíše mezi mikrokontrolery. Výjimkou je model Compute, který svým vzhledem spíše připomíná RAM paměť – ten musí být vždy zapojen do speciální desky, kde pak pracuje jako mozek celého projektu. [19]

To dává možnost vybrat si systém, který vyhovuje přímo pro projekt, do kterého Raspberry Pi použijeme. Samotné Raspberry Pi Foundation vydali Raspberry Pi OS. Je to sice oficiální distribuce, ale zdaleka ne jediná. V závislosti na modelu, který používáte, vašich cílech a osobních preferencích může být dobré prozkoumat i další možnosti. [25]

Zde je jich pár popsaných.

2.3.1 Raspberry Pi OS

Nejprve je dobré se seznámit s pojmem Debian. Debian je distribucí Linuxu. Jedná se o jednu z nejstarších vytvářenou distribucí, kterou nevyvíjí komerční subjekt. Jedná se o projekt, který je připravován velkým množstvím dobrovolníků z celého světa vývojářů. [26]

Raspberry Pi OS je bezplatný operační systém založený na Debianu a optimalizovaný pro hardware Raspberry Pi. Raspberry Pi OS podporuje více než 35 000 balíčků Debianu. Pro většinu případů použití Raspberry Pi je doporučováno Raspberry Pi OS. Protože je Raspberry Pi OS odvozen od Debianu, řídí se odstupňovanou verzí cyklu vydávání Debianu. K vydání dochází zhruba každé dva roky. Dříve byl znám pod názvem Raspbian, jakožto kombinace Raspberry Pi a Debian. [27, 28]

Existuje ve dvou verzích, podle toho, k čemu ho budeme využívat. Desktop představuje běžný operační systém, se kterým jsme se jistě někdy setkali – vypadá jako Windows. Pak ještě verze Lite, která je ochuzená o přehrávač. Tím ale nepřichází o své funkce. Jedná se o verzi, kde máme čistě příkazový řádek. Využívá se tehdy, kdy potřebujeme projekt mít co nejmenší, nebo pokud nám nevadí, že nevidíme na obrazovku a na naše Raspberry Pi se připojujeme vzdáleně, třeba pomocí SSH protokolu. [28]

2.3.2 OSMC

OSMC (Open Source Media Center) je multimediální přehrávač, který je open source (open-source znamená, že se jedná o software s volně dostupným zdrojovým kódem aplikace, díky čemuž je lépe modifikovatelný) a je založený na systému Debian. OSMC byl založen v roce 2014 a umožňuje přehrávat média z místní sítě, připojeného úložiště a internetu. OSMC je předním multimediálním centrem z hlediska souboru funkcí a komunity a vychází z projektu Kodi (jedná se o open source aplikaci k přehrávání vše různých medií). [29]

OSMC na Raspberry Pi je vynikajícím řešením pro každého, kdo hledá výkonné a flexibilní mediální centrum. Díky snadné instalaci, široké podpoře multimediálních formátů a

možnosti přizpůsobení je OSMC ideální volbou pro vytvoření domácího zábavního systému. [29]

2.3.3 Ubuntu

Ubuntu, jedna z nejznámějších distribucí Linuxu, je známá svou stabilitou, bezpečností a uživatelsky přívětivým prostředím. Tato kombinace otevírá dveře pro širokou škálu projektů a aplikací. Ubuntu na Raspberry Pi přináší spoustu možností a příležitostí pro vývojáře, vzdělávací instituce a hobbyisty. Díky spojení síly dvou populárních platforem – Raspberry Pi a Ubuntu – můžete vytvářet a experimentovat s různými projekty a aplikacemi. Ubuntu poskytuje systémy specifikované jak pro počítače, servery tak i cloudové služby. [30]

2.3.4 DietPi

DietPi je odlehčená verze Debianu (ten jsme si již popsali). Tento operační systém je speciálně navržen tak, aby snížil zátěž procesoru a paměti RAM počítače Raspberry Pi a uvolnil více výpočetního výkonu pro provádění potřebných úloh. Jeho programy používají nabídky Whtail (jedná se o shell skripty k zobrazení dialogových display oken pro informační účely), které jsou lehké a snadno se s nimi pracuje těm, kteří nechtějí trávit spoustu času zadáváním příkazů do konzole. Operační systém také umožňuje snadnou instalaci populárního softwaru, má jednoduché nastavení konfigurace, má vestavěný systém zálohování, umožňuje uživatelům přizpůsobit volby pro přihlašování, má kontrolní systém pro přidělování úrovní priority instalovaného softwaru, má vestavěný systém aktualizací a vestavěný systém automatizace instalace [31]

II. PRAKTICKÁ ČÁST

3 ZOBRAZENÍ INFORMACÍ POMOCÍ RASPBERRY PI

Raspberry Pi, jak bylo zmíněno, je sám o sobě počítačem. Tudíž jeho využití na zobrazení informací z pohledu hardwaru jsou stejně limitovaná tak, jak je normální počítač – propojením kabelem nebo vysíláním signálu. Svoji „otevřeností“ avšak vychází vstříc veškerým projektům a fantazii. Jak bylo zmíněno – celá myšlenka, proč projekt Raspberry Pi vznikl, bylo kvůli přiblížení programování a počítačové techniky široké veřejnosti. [22]

Existuje spousta možností, jak vyřešit zobrazování informací na obrazovce pomocí Raspberry Pi. Jde to přes různé operační systémy, které pustí video, pokud dojde ke změně pozorovatelných stavů, až po aplikace, které podobně jak systémy pustí médium.

Jedním z elegantních řešení, které se v této práci použije, je za pomoci internetové stránky. Předpokládaný plán je velmi jednoduchý. Máme celkem dva typy zařízení v našem systému:

- vysílací – kde sledujeme změnu. Tu následně zapíšeme na webovou stránku,
- sledovací – zde jsme připojeni na stránce a sledujeme změnu pomocí webové stránky.

3.1 Komerční aplikace

Na internetu je nespočet firem nabízejících tak zvaný *Digital Signage*. Jedná se obecně o systém digitálních zobrazovacích zařízení ve veřejně přístupných prostorách. Jejich účelem je obvykle zobrazení informačních, navigačních informací a především digitálně řízená reklama. Mezi takovéto řešení patří třeba PICKEL, yodeck, posterbooking, Easy Signage nebo piSignage. Všechny mají společné, že se jedná o software, který se nainstaluje na zařízení, kde se pak pomocí internetu hlásí do databáze. Obsah databáze, přístupné přes stránky příslušného poskytovatele této služby, spravuje pak sám klient. Obsah se sem nahraje a pak rozešle a přehrává na zařízeních, které jsou v databázi vybrány.

Zdá se být jednoduchým řešením, avšak za cenu nutného připojení k internetu a znemožnění funkce pouštění materiálu na obrazovku při výpadku služby.

Všechny fungují na podobném principu – schovává se tam webová stránka, na kterou se přehrávající zařízení připojí a z ní se pouští obsah. Což přímo nabízí řešení v podobě vytvoření si své vlastní stránky.

3.2 Webová komunikace

Jak již bylo zmíněno, předešlé řešení spoléhá na připojení se na stránku, ze které je pouštěn obsah. Trik je ve stránce, která není statická, ale dokáže v reálném čase upravovat svůj obsah. Toho se dosáhne pomocí navázání otevřeného spojení, díky kterému není třeba pro nový obsah znovu načíst stránku.

Webovou aplikaci můžeme dělit na dva typy:

- backend,
- frontend.

3.2.1 Backend

Backend je část aplikace, která se stará o zpracování dat, logiku, autentizaci, autorizaci a další serverové operace. Je to neviditelná část aplikace, která se nachází za frontendem a zajišťuje, že aplikace funguje správně.

Backend zpracovává požadavky z frontendu – tedy části aplikace, kterou uživatelé vidí a se kterou interagují – a poskytuje odpovědi zpět. Když uživatel provede akci na frontendu, například vyplní a odešle formulář, tato akce generuje požadavek, který je odeslán na server. Backend tento požadavek přijme, zpracuje jej podle dané logiky a vrátí odpověď zpět na frontend. Backendová část aplikace běží na serveru. To znamená, že kód backendu je spuštěn na serveru, který poskytuje služby a data frontendu na vyžádání. Backend může být napsán v různých programovacích jazycích, jako jsou Python, Java, Ruby, PHP, Node.js a další. Každý z těchto jazyků má své výhody a je vybrán podle specifických potřeb projektu. Zde je pár příkladů.

Backend tedy hraje klíčovou roli v každé aplikaci, protože zajišťuje, že data jsou správně zpracována, a že aplikace funguje podle očekávání. Bez silného a efektivního backendu by frontend nemohl poskytovat uživatelům požadovanou funkcionalitu a zážitek.

3.2.1.1 *Ruby on Rails*

Ruby on Rails, zkráceně Rails nebo RoR, je framework určený pro rychlý vývoj moderních webových aplikací či webů. Jedná se o open-source řešení, které s cílem zvýšení produktivity vývojářů v roce 2004 představil dánský programátor David Heinemeier Hansson. Ruby on Rails jsou napsané v programovacím jazyce Ruby a postavené na dvou základních myšlenkách: konvence nad konfigurací a DRY – Don't Repeat Yourself. To znamená, že by

vývojáři měli upřednostnit předpřipravené moduly před tvorbou svých vlastních. Čili naplno využívat hotových (standardních) řešení, které RoR nabízí, namísto vytváření nových tříd se stejnými vlastnostmi. K vlastní konfiguraci jednotlivých částí aplikace se tak typicky ubíráme pouze v případě, kdy se liší od běžného nastavení. Současně by měl být každý prvek v kódu definovaný pouze jednou, díky čemuž je možné vytvářet jednodušší, čistší a přehlednější programy. Obě pravidla navíc šetří vývojářův čas, který může strávit nad složitější funkcionalitou. [32]

3.2.1.2 Node.js

Node.js je asynchronní prostředí JavaScriptu řízené událostmi, které je určeno k vytváření škálovatelných síťových aplikací. Při každém spojení se spustí zpětné volání, ale pokud není třeba provést žádnou práci, Node.js se uspí. V Node.js je kladen důraz na vysokou škálovatelnost, tzn. schopnost obsloužit mnoho připojených klientů současně. Pro tuto vlastnost a vysokou výkonnost je dnes Node.js velmi oblíbený pro tvorbu tzv. API serverů pro klientské single page aplikace rovněž v JavaScriptu. [33]

Jedná se o single-thread řešení. To znamená, že pracuje pouze v jednom vlákne, což podmiňuje využívání principů asynchronního programování, kdy není řešení jednoho požadavku závislé na dokončení požadavku druhého. Jinými slovy se nezdržuje čekáním na vyřešení složitějšího příkazu. Tím se minimalizuje režie a maximalizuje potenciál výkonu. [34]

Jednotlivé klientské požadavky se ukládají do zásobníku, z něhož jsou vyjímány ve stejném pořadí, jako byly vloženy (princip FIFO). NodeJS tak funguje na událostmi řízeném (event-driven) modelu neblokujícím I/O. [34]

Jádrém celé technologie je tedy V8 engine, který má na starosti kompilaci do strojového kódu. Architektura Node.js přitom stojí na takzvané smyčce událostí – Event Loop. Smyčka utváří prostor pro přijímání uživatelských požadavků, jež ve formě událostí dále přiděluje jednotlivým zdrojům. Právě ze smyčky událostí a využívání jediného vlákna pak vychází potřeba asynchronní komunikace a princip fronty. [34]

V praxi to funguje tak, že vývojář prostřednictvím Node.js vytvoří webový server, jenž bude na portu čekat na uživatelské požadavky. Jakmile nějaký požadavek zachytí, pošle ho do smyčky, která jej zadá příslušnému controlleru. V mezičase ovšem zpracovává i další, mnohdy, méně náročné požadavky. Jakmile je původní požadavek zpracován, dojde k zavolání (callback) výsledku a následnému dokončení prvotního procesu. Složitější požadavky tak neblokují vlákno těm, které mohou být provedeny rychleji. [34]

3.2.2 Frontend

Frontend je část webové aplikace, kterou klient nebo uživatel využívá přímo v prohlížeči. Tato část zahrnuje rozložení stránky, design, interaktivní prvky a veškerý obsah, který uživatel vidí a se kterým může pracovat:

- rozložení stránky – to znamená základní struktura webové stránky, která určuje, kde se nachází různé prvky, jako jsou záhlaví, navigační menu, hlavní obsah, postranní panely a zápatí. Rozložení stránky je navrženo pomocí HTML a CSS, aby bylo responzivní a přizpůsobilo se různým zařízením a velikostem obrazovek,
- design – zahrnuje barevné schéma, obrázky, ikony a další vizuální prvky, které dohromady tvoří vzhled stránky,
- interaktivní prvky – toto jsou části webu, které umožňují uživateli interakci, například tlačítka, formuláře, rozbalovací menu, galerie obrázků a další dynamické komponenty. JavaScript, spolu s knihovnamy a frameworky jako React, Angular nebo Vue.js, se používá k přidání interaktivity a dynamiky.

[35, 36]

Frontendová část aplikace je zodpovědná za uživatelský zážitek, takže se zaměřuje na estetiku a funkčnost rozhraní, aby byla pro uživatele příjemná a intuitivní. Frontend aplikace běží v prostředí prohlížeče uživatele. Kód Frontendu je vykonán přímo v prohlížeči poté, co si ho stáhne ze serveru. Tento kód je obvykle napsán v jazycích jako HTML, CSS a JavaScript.

3.2.2.1 *React*

React je open-source javascriptová knihovna pro tvorbu uživatelského rozhraní (anglicky UI jako User Interface). Vyvinula ji společnost Meta (dříve Facebook). Uživatelská rozhraní v React typicky komunikují s nějakým webovým API serverem, ze kterého získávají data, případně mu data posílají. Díky tomu lze React aplikaci používat snadno na více zařízeních a máme stále aktuální data, protože jsou uložena na serveru. React tvoří právě a pouze pohledovou vrstvu aplikace, která prezentuje data uživateli. React se z tohoto důvodu s oblibou používá s dalšími knihovnamy a existuje kolem něj celý ekosystém. [37]

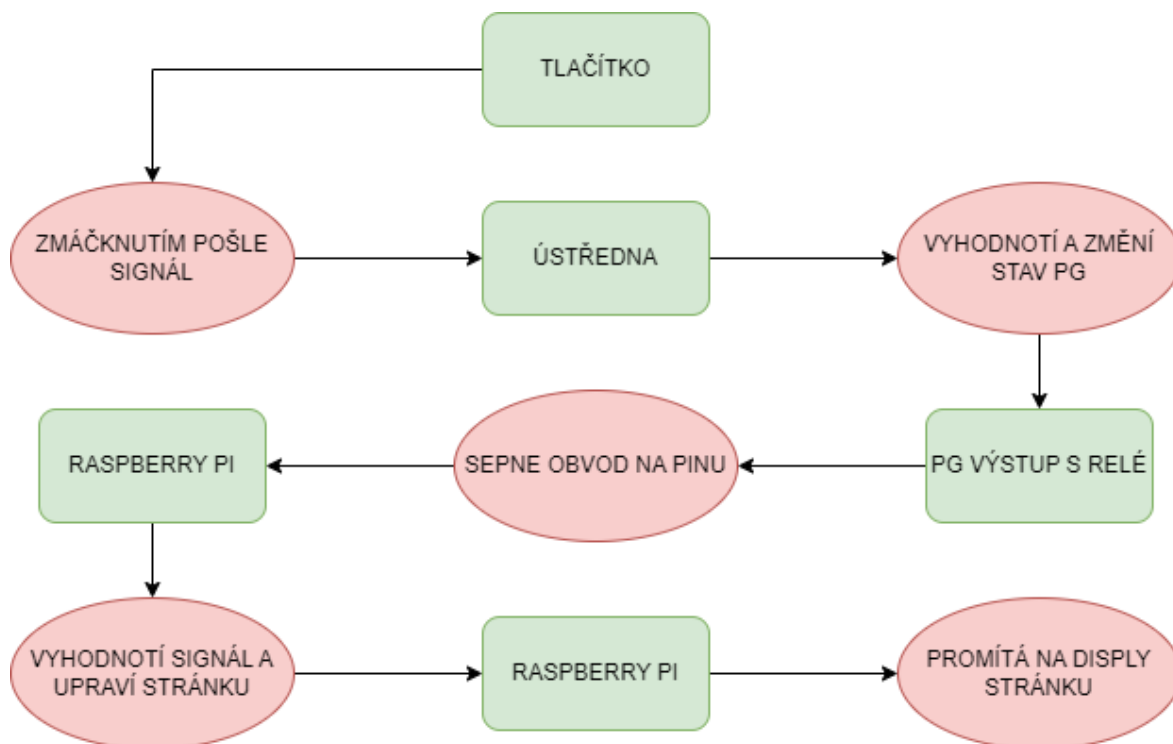
3.2.2.2 *Angular*

Na rozdíl od Reactu se jedná o celkovou aplikaci, nejen knihovnu. Angular je dobře známý pro svůj funkčně bohatý framework, který nabízí řadu funkcí usnadňujících vývoj aplikací.

Jednou z jeho klíčových vlastností je architektura MVC (Model-View-Controller). Tento návrhový vzor rozděluje aplikaci na tři vzájemně propojené komponenty – model, zobrazení a kontrolér – a umožňuje tak vývojářům efektivněji spravovat složité projekty. Další význačnou vlastností architektury Angular je obousměrná vazba dat. Tento systém automaticky aktualizuje zobrazení při změnách v modelu a naopak, což vývojářům šetří čas a snižuje složitost. Angular také nabízí špičkové animace a funkce pro zpřístupnění, které výrazně zvyšují uživatelský komfort. Podporuje také opakované použití komponent, které zvyšuje produktivitu vývojářů tím, že jim umožňuje opakovaně používat kód v různých částech aplikace. [38]

4 FUNKČNÍ ZAPOJENÍ

Zapojení, kterým se tato bakalářská práce zabývá, spočívá v poslání signálu od základní periferie přes ústřednu po vyhodnocení na počítači Raspberry Pi na obrazovku. Pracujeme s úvahou, že jsme v objektu, kdy dojde k narušení během běžného provozu a chceme upozornit na nebezpečí a kde se nachází. Jedním z příkladů může být škola, kdy během školního dne dojde k narušení střelcem. V zájmu snahy o vyvarování se studentů střetnutí se střelcem v objektu, nebo k rychlejšímu upřesnění lokace, kde se momentálně nachází, lze využít systému PZTS. Ten sám o sobě pracuje s periferiemi a dokáže pomocí nich zpozorovat, kde se útočník nachází. Dochází tam však ke komunikační prodlevě, než se informace dostane na potřebné místo. To se snaží tato práce vyřešit pomocí zobrazení varovných zpráv přímo v objektu na obrazovkách.



Obrázek 13 – Diagram zapojení [vlastní zpracování]

Zde je jednoduchý diagram popisující řešení našeho případu, následně popíši jednotlivé kroky.

4.1 Komunikace z ústředny do Raspberry Pi

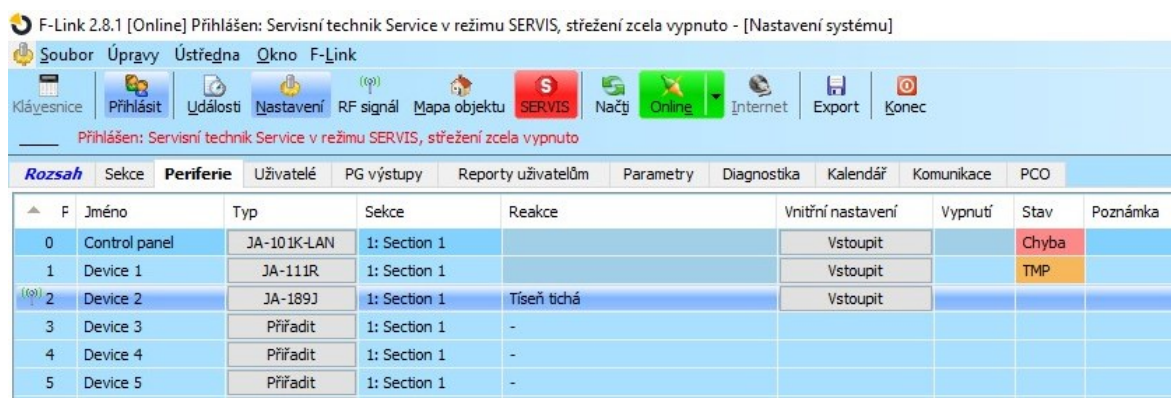
Než se dostaneme k obrazovkám, potřebujeme se podívat na komunikaci mezi ústřednou a počítačem Raspberry Pi. Je třeba nalézt, kde náš signál vznikne, abychom ho mohli později ukázat na obrazovce.

4.1.1 Ústředna pošle signál

Potřebujeme nejdříve najít popud, na který dokážeme navázat dál v této práci. Tento popud může být získán pomocí některé z periférií, které máme k dispozici v našem systému PZTS. Uvažujeme o využití některé konkrétní periferie, která by nám umožnila efektivně upozornit na násilníka přímo v objektu.

4.1.1.1 Periferie pošle ústředně

Řešení, které se hned nabízí, jsou tlačítka, která v systému ústředny pojmenujeme. V tu chvíli se pro nás stávají tísňovými tlačítky. Rozmístíme je v objektu podle potřeby, kde chceme a můžeme jednoznačně určit, že nebezpečí je v dané části. Pokud bychom přemýšleli třeba nad budovu o třech patrech a třech křídlech, umístíme jedno až dvě tlačítka do jednotlivých křídel a pater. Ty napojíme na ústřednu a v perifériích jim připišeme k názvu označení, abychom se v nich vyznali. Tlačítkům v aplikaci F-Link přiřadíme spuštění tiché tísně. Tím přijmeme signál do ústředny.



F	Jméno	Typ	Sekce	Reakce	Vnitřní nastavení	Vypnutí	Stav	Poznámka
0	Control panel	JA-101K-LAN	1: Section 1		Vstoupit		Chyba	
1	Device 1	JA-111R	1: Section 1		Vstoupit		TMP	
2	Device 2	JA-189J	1: Section 1	Tíseň tichá	Vstoupit			
3	Device 3	Přiřadit	1: Section 1	-				
4	Device 4	Přiřadit	1: Section 1	-				
5	Device 5	Přiřadit	1: Section 1	-				

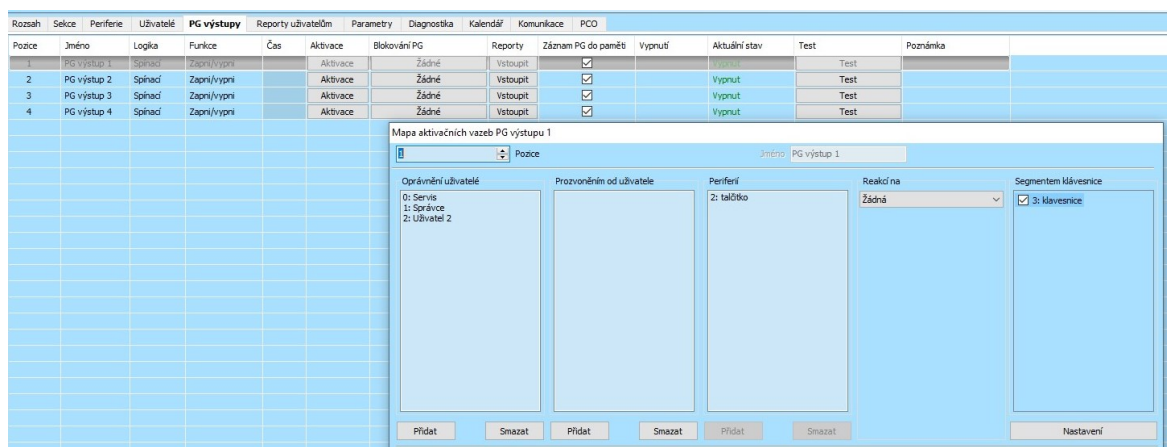
Obrázek 14 – Zaznamenání periferie do systému PZTS [vlastní zpracování]

Zde je příklad zapojení bezdrátového tlačítka JA-189J. pozici číslo 0 automaticky zabírá ústředna. Na pozici 1 se nachází rádiový komunikátor k přijímání signálů. Díky tomu lze naše tlačítko napojit. Při kliknutí na ikonu *Přiřadit*, se bezdrátová zařízení obvykle připojují učením neboli zmáčknutím na požadovanou dobu uvedenou výrobcem.

4.1.1.2 Ústředna vyhodnotí

V nastavování ústředny pak lze k určitým perifériím přiřadit jednotlivé programovatelné výstupy. Několik tlačítek lze napojit na jeden programovatelný výstup. Proto lze, kdyby se útočník pohyboval po objektu a byl by v nějaké chodbě, dát tlačítka na začátek a konec chodby a napojit je na stejný programovatelný výstup, který by hlásil změnu stavu. Tím se

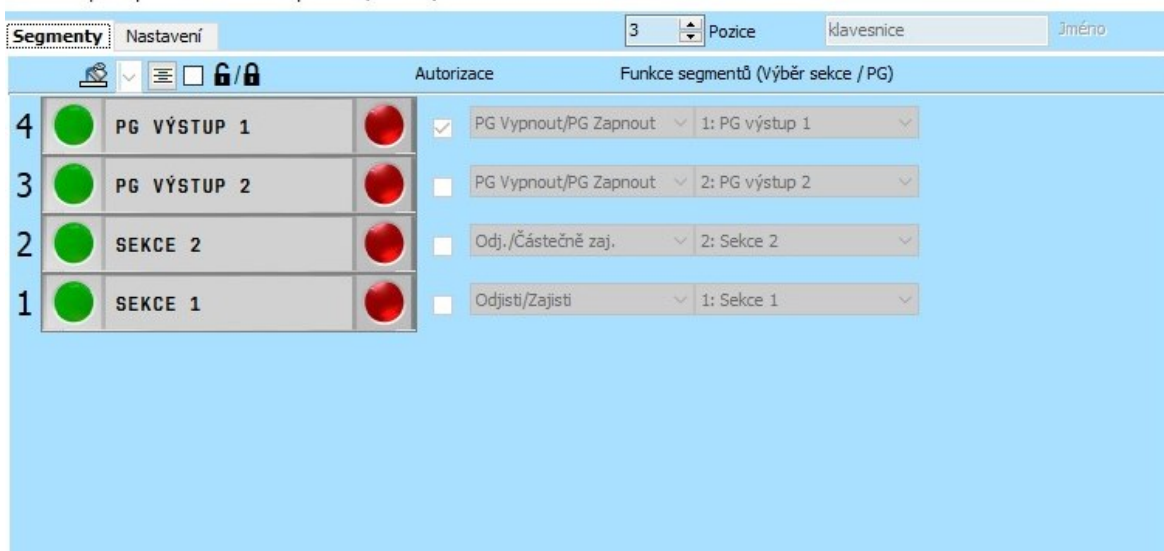
předejte možnosti, že by útočník stál u tlačítka a tím znemožnil přiblížení nebo ukázání jeho lokace.



Obrázek 15 – Přiřazení vlastností PG výstupu [vlastní zpracování]

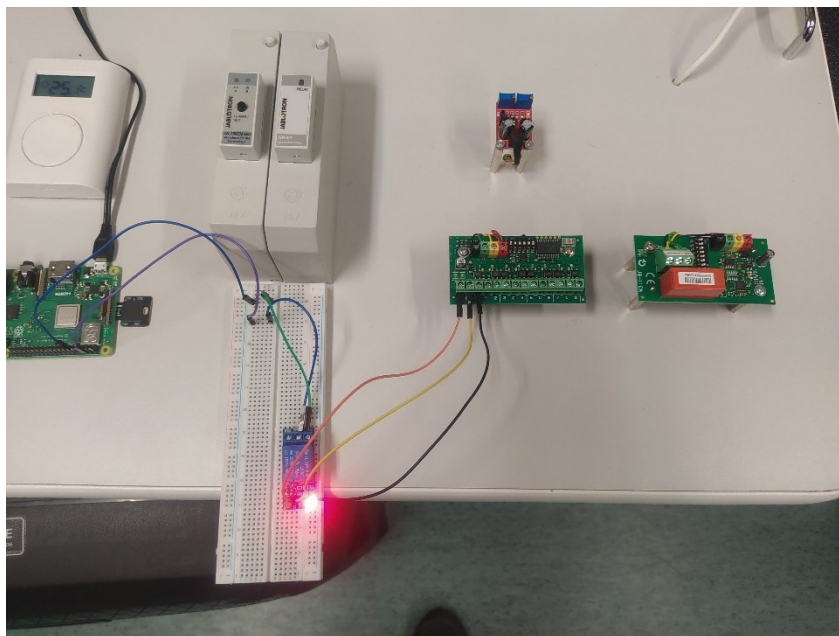
Ústředna se kupuje podle potřebného počtu programovatelných výstupu, avšak ty lze pomocí sběrnic rozšířit na požadovanou hodnotu. Pomocí tohoto systému dokážeme na podnět zmáčknutí tlačítka spustit tichou tiseň a dát signál k reakci na programovatelný výstup, který třeba pomoci elektromechanického relé sepne obvod.

Nastavení přístupového modulu na pozici 3 (JA-114E)



Obrázek 16 – Nastavení klávesnice [vlastní zpracování]

Pokud máme k ústředně připojenou klávesnici, lze zde nastavit zapnutí, vypnutí či nastavení obou dvou stavů programovatelného výstupu. V obrázku č 17. je příkladné zapojení přímo na sběrnici programovatelných výstupů.



Obrázek 17 – Zapojení Raspberry Pi na PG výstup s relé

4.1.2 Raspberry Pi přijme signál

Raspberry Pi, díky svojí stavbě, je vybaveno k přijímání a posílání signálů mnoho způsoby. Jedním ze zajímavých způsobů realizace přijímání signálů je využití GPIO – General Purpose Input Output. Podle schématu z kapitoly o Raspberry Pi můžeme vidět, že se nejedná jen o jedno typové piny. Celkem se jedná o 28 programovatelných pinů (zbytek se zabývá napájením či uzemněním). Pomocí různých programátorských knihoven lze tyto piny velmi jednoduše ovládat a kontrolovat.

4.1.2.1 Rozdělení pinů

V základní verzi je přímo na Raspberry Pi k dispozici knihovna RPi.GPIO, která umožňuje ovládání pinů pomocí jazyka Python. Tato knihovna poskytuje rozhraní pro manipulaci s GPIO piny, což je klíčové pro mnoho projektů a aplikací na Raspberry Pi.

Je však důležité si uvědomit, že při použití této knihovny je nutné dát pozor na to, na jaký systém určování pinů se odkazujeme. Existují dva hlavní systémy určování pinů:

- GPIO.Board – Díky této funkci určujeme polohu pinu na desce Raspberry Pi podle čísla. Na desce Raspberry Pi pracujeme s čísly pinů od 0 do 40, které označují jednotlivé piny na GPIO (General Purpose Input/Output) konektoru. Je důležité si

uvědomit, že v tomto kontextu nepracujeme s čísly, která označují napájecí piny (například 3V3, 5V) nebo uzemňovací piny (GND). Tyto piny mají speciální funkci a jsou určeny k napájení připojených zařízení nebo k zajištění elektrického spojení se zemí. Proto při práci s GPIO piny na desce Raspberry Pi pracujeme pouze s čísly 0 až 40, které identifikují volné I/O piny, které můžeme použít pro komunikaci s různými periferiemi a senzory,

- GPIO.BCM – V tomto systému pracujeme přímo s označením jednotlivých pinů, což jsou čísla od 0 do 27. Tato čísla představují jednotlivé piny na GPIO (General Purpose Input/Output) konektoru Raspberry Pi.

[22]

Při našem zapojení, signál z ústředny je vyslán sepnutím obvodu pomocí elektromechanického relé. To dokážeme monitorovat na Raspberry Pi čistě propojením jednoho z 26 programovatelných pinů a jednoho z ground, neboli uzemňovacích pinů. Vezmeme si pro náš případ pin GPIO 27. Pomocí GPIO.BCM určit jeho číslo pro programování je jednoduché – jedná se o pin s číslem 27, takže v kódu bude zapsán jako 27. Při použití GPIO.Board se musíme podívat na schéma, nebo zadat do konzole příkaz „*pinout*“, který nám ho taky ukáže. Tam se dozvíme, že pro GPIO 27 nám náleží na desce pin s číslem 13. proto pokud budeme programovat dle tohoto systému, musíme ho uvádět jako 13.

4.1.2.2 *Definice logické úrovně*

Při sepnutí obvodu nebo pouštění proudu do nějakého z obvodů, může dojít ke kolísání. Pokud chceme definovat logické úrovně v obvodě, když není připojen k žádnému jinému signálu, lze toho docílit pomocí „pull-up“ a „pull-down“ rezistorů:

- Pull-up rezistor – zde ho máme připojený mezi vstupní pin a napájecí napětí. Když vstupní pin není připojen k žádnému jinému signálu, „pull-up“ rezistor udržuje pin na logické "1" úrovni (vyšší napětí). Když je pin připojen k zemi přes jiný spínač nebo obvod, napětí na pinu klesne na logickou "0".
- Pull-down rezistor – v tomto případě je připojený mezi vstupní pin a zem. Když vstupní pin není připojen k žádnému jinému signálu, pull-down rezistor udržuje pin na logické "0" úrovni (nízké napětí). Když je pin připojen k napájecímu napětí přes jiný spínač nebo obvod, napětí na pinu stoupne na logickou "1"

Pro naše účely využijeme „pull-down“ rezistor.

4.1.2.3 Čtení na pinu

Pro příklad bereme dvě tlačítka, dva programovatelné výstupy, dva obvody, takže pracujeme se dvěma piny. Jsou napojeny na piny GPIO 27 a GPIO 22. Na desce jsou popsány jako piny 13 a 15. Tento kód nám při spuštění bude říkat, jaký je momentální stav na pinech. Při změně dojde k vypsání do konzole.

4.2 Zobrazení informace na obrazovce

Signály jsou dva, protože pracujeme se dvěma tlačítky. Ta byla zmáčknuta po zpozorování narušitele a díky tomu poslala signály do ústředny, kde byly následně vyhodnoceny. Ústředna je poslala na programovatelné výstupy PG1 a PG2, kde sepnuly elektromechanické relé a tím uzavřely obvod na pinech GPIO 27 a GPIO 22. (na desce jsou popsány jako piny 13 a 15). Nyní potřebujeme vyčíst změnu při uzavření obvodu a díky tomu upozornit na nebezpečí.

Dosáhneme tím tak, že budeme mít jedno Raspberry pi zapojené k vyhodnocení stavu a druhé, či více k zobrazení stavu. Pojmenujeme si je jako *Vysílající* a *Sledující* trefně podle toho, jakou roli zastávají v této komunikaci. Jelikož se jedná, jak již bylo v minulé kapitole zmíněno, o webovou komunikaci. Jedno Raspberry Pi bude připojeno na programovatelné výstupy. Poběží nám na něm také Node.js server, který se stará o backend a rozložení naší webové stránky. Pokud nastane změna pozorovaná na pinech, dojde k vyhodnocení a změna se promítne na stránce. Na ní budou připojeny veškerá Raspberry Pi v objektu. Ty budou u obrazovek a promítat stránku přímo na obrazovku.

Řešením vlastní stránky eliminujeme potřebu přístupu k internetu, čímž se zbavíme závislosti na něm. Pro naše účely nám postačí pouze lokální síť, ve které bude naše vysílající zařízení provozovat server, na který se připojí veškerá zařízení, pomocí kterých chceme promítat na displaye obsah webové stránky. Díky tomuto přístupu by se dalo říct, že se jedná o bezpečnější a spolehlivější systém než dříve zmíněné *Digital Signage*, které spoléhají na neustálé připojení. Bezpečnější je z hlediska izolace sítě, na které běží. V objektu lze pak vyhradit samostatný okruh k provozu sítě.

4.2.1 Vysílající

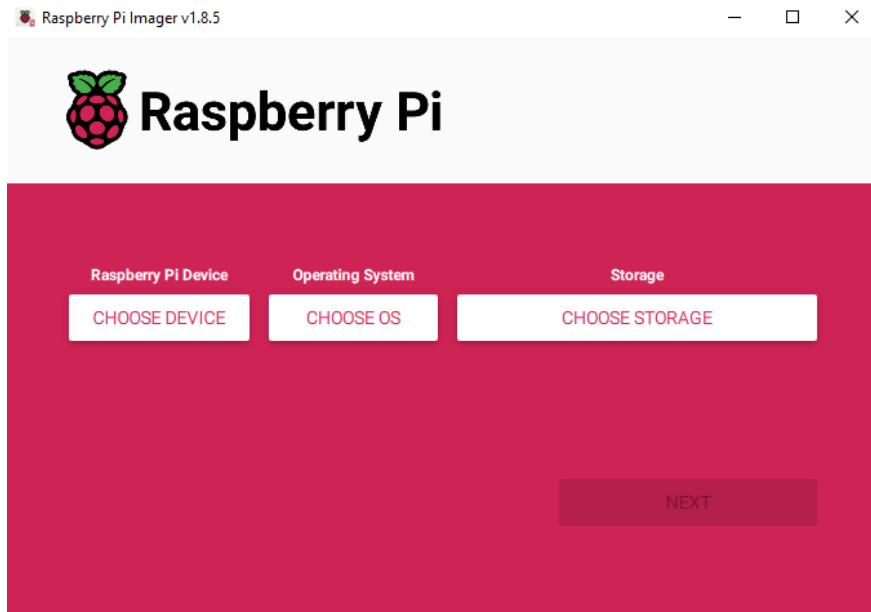
Vysílajícím Raspberry Pi si zvolíme model B, pro jeho výkon. Nainstalujeme na něm základní verzi Raspberry Pi OS. Stačí nám verze Lite. Raspberry Pi OS přichází ve dvou

hlavních verzích: Desktop a Lite. Hlavní rozdíl mezi těmito dvěma verzemi spočívá v obsahu aplikací a rozhraní, které jsou při instalaci zahrnuty, což ovlivňuje velikost a výkon systému.

Raspberry Pi OS Desktop obsahuje Grafické uživatelské rozhraní (GUI). To znamená, že má plně vybavené desktopové prostředí s grafickým uživatelským rozhraním, konkrétně LXDE. Má s sebou řadu předinstalovaných aplikací, jako je webový prohlížeč, kancelářské aplikace (např. LibreOffice), textové editory, nástroje pro správu souborů a další nástroje, které uživatelé běžně očekávají od desktopového operačního systému. Proto je ideální pro uživatele, kteří chtějí používat Raspberry Pi jako běžný stolní počítač, s přístupem k GUI a všemi základními aplikacemi. [28]

Raspberry Pi OS Lite je bez GUI. Takže tato verze neobsahuje žádné grafické uživatelské rozhraní. Je to čistě textově orientovaný operační systém. Hned po instalaci má pouze základní systémové soubory a několik klíčových nástrojů. Nejsou zde předinstalované žádné desktopové aplikace, což ho dělá ideálním pro serverové aplikace nebo situace, kde je potřeba minimalizovat využití systémových zdrojů. Mnozí uživatelé ji volí pro headless (bez hlavy, tj. bez monitoru) konfigurace, kde je systém spravován na dálku přes SSH nebo jiný terminálový přístup. [28]

Instalaci provedeme na nosič naší volby. Třeba se může jednat o micro SD kartu. Na oficiálních stránkách Raspberry Pi existuje aplikace „Raspberry Pi Imager“. Operační systém nelze jen přetáhnout. Aplikace nám slouží k vybrání, stáhnutí a také instalaci specifické verze operačního systému na námi vybraný nosič. Před instalací lze zvolit určité nastavení k zjednodušení procesu prvního spuštění. Lze tu zvolit jméno a heslo k účtu používaného v systému, pokud může tak zadat údaje od WiFi, kam se má připojit a protokoly, které od začátku bude mít zaplé. Důležité je zde vyplnit jméno a heslo účtu, které si někde zapíšeme nebo zapamatujeme a že SSH protokol bude od instalace povolen.

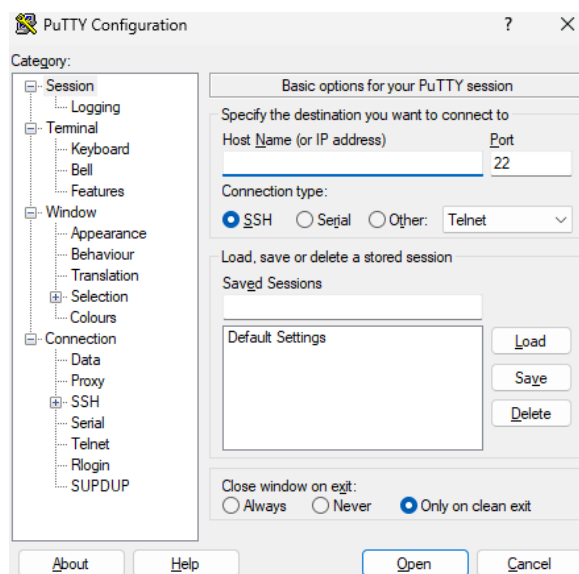


Obrázek 18 – Raspberry Pi Imager [vlastní zpracování]

SSH neboli Secure Shell, vznikl jako náhrada za telnet. Telnet je protokol, který umožňuje vzdálené připojení k jinému počítači (např. i routery, switche) a textovou komunikaci mezi nimi (ve formě vzdáleného terminálu). Nemusíte tedy fyzicky přijít k danému zařízení a připojit si klávesnici a monitor. Stačí být připojen k jedné síti. Obrovská nevýhoda telnetu je to, že komunikace probíhá jako plain-text – posílaná data nejsou nijak šifrována a kdokoliv v síti je tedy může odposlouchávat. SSH se skládá ze dvou částí, které operují nezávisle na sobě – klient a server. Funguje to tak, že pokud se chcete připojit na vzdálený počítač, stačí Vám pouze SSH klient. Podmínkou ale je, že na onom vzdáleném počítači musí být nainstalovaný právě SSH server, který Vám zprostředkuje potřebné služby. To znamená, že pokud i Vás počítač má přijímat SSH spojení, musí na něm být nainstalovaný SSH server. [39]

Po instalaci systému nás program upozorní, že dokončil. Kartu nyní můžeme připojit do Raspberry Pi, dále připojíme ethernetový kabel (pokud jsme nezadali lokální WiFi), který je zapojený do lokální sítě s internetem a zdroj v podobě micro USB, který přišel v balení.

Počítač by po připojení displaye ukazoval příkazový řádek a mohli bychom ho ovládat přímo připojenou klávesnicí, avšak my se k němu připojíme pomocí protokolu SSH. V *Terminal* na Linuxu nebo macOS lze přímo. U Windows se hodí stáhnout si aplikaci *PuTTY*. V *Terminal* se používá příkaz „ssh (jméno, co jsme při instalaci zvolili)@(ip kterou má)“ a v *PuTTY* jen se jen zadá IP adresa. Raspberry Pi dokážeme v síti i zavolat pod adresou „(jméno, co jsme při instalaci zvolili).local“, která funguje jako jeho IP adresa.



Obrázek 19 – Rozhraní PuTTY [vlastní zpracování]

Po navázání připojení se musíme přihlásit údaji, které jsme vyplnili při instalaci. U Desktop verze tyto kroky může vykonat při prvním spuštění.

Začneme získáním nejnovější verze těmito příkazy:

- `sudo apt update`
- `sudo apt upgrade`

. Musíme je použít s příkazem „sudo“, abychom měli oprávnění k práci se systémem. Mnohé operace jsou nám, jakožto uživatel, zablokovány a vyžadují vyšší oprávnění. O to se stará účet s názvem „root“, který jak název vypovídá je kořenem, tedy základní. Tím si zajistíme systém. Nyní potřebujeme aplikace, pomocí kterých spustíme náš web server.

- `curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.1/install.sh | bash`
`source ~/.bashrc`
- `nvm install node`

Pomocí příkazu „curl“, který stáhne soubor z internetové lokace, stáhneme „nvm“ aplikaci. Jedná se o *Node Version Manager*. Ten se stará o aplikaci *Node.JS* a její verze. Následně stáhneme a nainstalujeme Node.js na tento počítač, protože na něm pojede náš webový server. Příkazy, které mají parametr „-v“, zkontrolujeme, že jsou opravdu nainstalované. Do konzole nám vrátí, jaká verze je momentálně nainstalovaná.

```
piBPma@piBPma:~ $ nvm install node
Downloading and installing node v22.2.0...
Downloading https://nodejs.org/dist/v22.2.0/node-v22
#####
Computing checksum with sha256sum
Checksums matched!
Now using node v22.2.0 (npm v10.7.0)
Creating default alias: default -> node (-> v22.2.0)
piBPma@piBPma:~ $ node -v
v22.2.0
piBPma@piBPma:~ $ npm -v
10.7.0
piBPma@piBPma:~ $ █
```

Obrázek 20 – Kontrola nainstalované verze programů [vlastní zpracování]

Nyní se vrhneme na tvoření našeho projektu. Vytvoříme složku s názvem „projekt“ pomocí příkazu „mkdir“, dostaneme se do ní a dáme programu NPM (správce balíčku pro JavaScript) příkaz, aby se v této složce zinicilizoval a vytvořil všechny potřebné souboru pro projekt v Node.js.

- mkdir projekt
- cd projekt
- npm init

NPM (Node Package Manager) je správce balíčků pro JavaScript, který významně usnadňuje instalaci a správu knihoven a modulů. Původně byl NPM navržen pro práci se serverovým JavaScriptem v rámci Node.js, ale dnes se široce využívá i pro správu frontendových balíčků a různých knihoven. NPM umožňuje snadno instalovat a aktualizovat knihovny a moduly, které vývojáři používají ve svých projektech. Správa závislostí je automatizovaná, což šetří čas a eliminuje problémy spojené s ruční správou knihoven. Pomocí jednoduchých příkazů „npm install <název_balíčku>“ lze rychle přidat potřebné knihovny do projektu. Balíčky mohou být nainstalovány lokálně (pro konkrétní projekt) nebo globálně (dostupné ve

všech projektech). NPM byl vytvořen pro Node.js, což je prostředí pro serverový JavaScript. Jeho cílem bylo usnadnit správu modulů, které jsou potřebné pro vývoj serverových aplikací. Díky své flexibilitě a užitečnosti se NPM rozšířil i do světa frontendového vývoje. Dnes je standardním nástrojem pro správu nejen serverových, ale i klientských knihoven a nástrojů. [40]

NPM se používá k instalaci a správě různých frontendových nástrojů a knihoven, jako jsou React, Angular, Vue.js a mnoho dalších. Umožňuje vývojářům snadno spravovat různé verze knihoven a jejich závislosti, což je klíčové pro udržení konzistence a stability projektů. NPM má obrovský registr balíčků, který obsahuje miliony veřejně dostupných knihoven a modulů. Vývojáři mohou také publikovat své vlastní balíčky a sdílet je s komunitou. NPM je neocenitelný nástroj pro každého JavaScriptového vývojáře. Zajišťuje efektivní a bezproblémovou správu balíčků, což umožňuje soustředit se na vývoj aplikací místo na správu závislostí. Jeho původní zaměření na Node.js se postupem času rozšířilo i na frontendový vývoj, čímž se NPM stal nedílnou součástí moderního vývoje webových aplikací. [40]

Je třeba se nyní podívat, jaké knihovny nebo balíčky pro realizaci potřebujeme a stáhnout je. Ty stáhneme dle příkazu, který jsme uvedli v předchozím odstavci („npm install <název_balíčku>“).

- *Express.js*, nebo jednoduše *express*, je backendová část webové aplikace. Zjednodušuje proces vytváření webových serverů a aplikací v Node.js tím, že poskytuje jednoduché rozhraní pro přeposílání HTTP požadavků, správu stavu relace, zpracování požadavků a odpovědí, systém pro generování šablon s kódem HTML stránek a mnoho dalšího.
- *ws* je zkratka pro *WebSocket*, komunikační protokol navržený pro trvalé, plně duplexní, dvousměrné spojení mezi klientem a serverem prostřednictvím jednoho TCP spojení. Tento protokol umožňuje interaktivní komunikaci v reálném čase mezi webovým prohlížečem a serverem
- *array-gpio* nám slouží ke komunikaci s GPIO pro kontrolování pinů. Je to knihovna založená na JavaScriptu. Tím, že jsme sjednotili naše příkazy na JavaScript, tedy jeden jazyk, může být konfigurace serveru jeden soubor.
- *path* zajišťuje sadu funkcí k manipulaci s cestami k souborům a adresářům.

[41]

Příkazem „nano“ se nejen díváme, podobně jako s textovým editorem, ale také vytváříme soubory. Je třeba vytvořit soubor „server.js“, kde se bude nacházet konfigurace našeho webového serveru. Konfigurace serveru je v příloze. Jedná se o jednoduchý Node.js server s aktualizací v reálném čase.

Vytvoříme ve složce projekt složku „public“, kde bude schovaný soubor „index.html“ napsaný v HTML kódu, aby si server mohl načíst, jak má vypadat a jak se chovat. Ten zase vytvoříme pomocí příkazu „nano“. Kód stránky ukazuje na stav v reálném čase. Toho jsme



Obrázek 21 – Struktura složky s prací [vlastní zpracování]

docílili tím, že stav na pinech pozorujeme jako proměnnou. Stránka se tedy načte, naváže kontakt mezi klientem a serverem a server pak už jen upravuje hodnotu proměnné dle změny. Je tam napsaný příkladový školní rozvrh s nejdůležitější částí. Tou je, že když dojde k sepnutí okruhu na pinu 13 a pinu 15, vyskočí přes obrazovku a upozornění, které překryje rozvrh, zastíní ho a vypíše „Sepnut pin 13“ nebo „Sepnut pin 15“. To může být přepsáno dle potřeb. Kód HTML stránky je v příloze, stejně jako kód od serverové aplikace. Struktura naší práce by měla odpovídat obrázku č. 21.

Nyní nám stačí zapnout server pomocí příkazu „node server.js“. Tento příkaz spustí náš server, který jsme definovali v souboru server.js. Po spuštění serveru bude možné zobrazit webovou stránku, která je definovaná v souboru index.html. Když server běží, bude dostupný na IP adrese našeho zařízení, konkrétně na portu 3000. Obecně se jedná o adresu „http://ip_adresa_vysílajícího:3000“. Pro upřesnění – pokud naše zařízení má v síti IP adresu „192.168.1.100.“, stránka bude dostupná na adrese http://192.168.1.100:3000. Server bude naslouchat na této adrese a portu a obsluhovat HTTP požadavky, které na tuto adresu přijdou.

4.2.2 Sledující

Se serverem zapnutým nám nyní zbývá už jen nastavit zařízení u obrazovky ke sledování. Připojit by se dalo pomocí displaye umožňujícího připojení k internetu a s nainstalovaným prohlížečem. Tomu by jen stačilo zadat do prohlížeče, nebo udělat skript, s připojením na stránku s adresou popsanou v předešlém bodě („`http://ip_adresa_vysilajiciho:3000`“). Při využití nějakého počítače jde ale zaručit, že přehrávač nebude problém. Využijeme tedy i tady Raspberry Pi, také model B.

Začneme jak s *Vysilajícím*, a to instalací systému. Jelikož nyní nám záleží, aby zařízení bylo schopné grafického prostředí, tak nainstalujeme Desktop verzi Raspberry Pi OS. Dala by se nainstalovat i verze Lite, ale musel by se tam ručně stáhnout a nainstalovat přehrávač. Po instalaci připojíme k Raspberry Pi display a ethernet se stejnou sítí, jako je *Vysilající*.

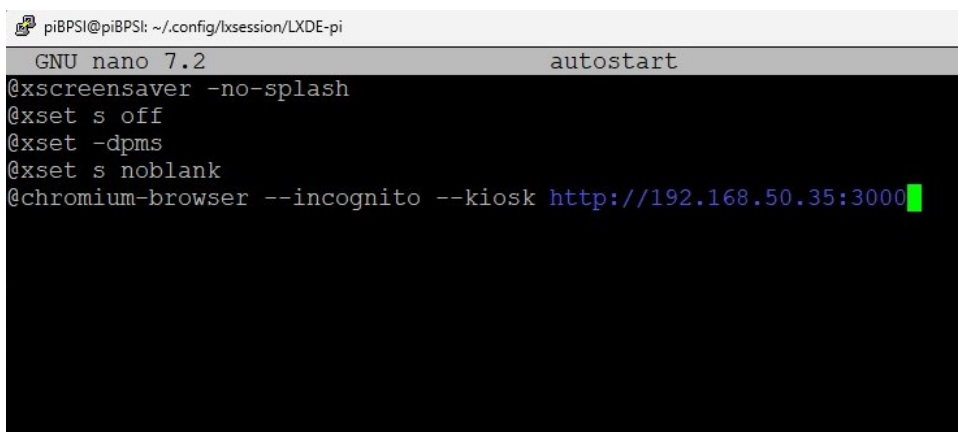
Jak u *Vysilajícího*, se přihlásíme. Buď můžeme pracovat na displayi, který jsme k zařízení připojili nebo může zvolit cestu SSH.

„`lxsession`“ je program, který spravuje a hlídá přehrávač X11. X11 pak spravuje rozhraní plochy. My ho potřebujeme změnit a dát mu jasné parametry. Potřebujeme tam upravit soubor s názvem „`autostart`“. Jak název napovídá, jedná se o soubor zajišťující automatické spuštění při každém zapnutí.

- `cp -r /etc/xdg/lxsession ~/.config/`

Překopírováním do složky našeho účtu zajistíme, že bude platit jen pro tento účet. Má to své výhody – když dojde k aktualizaci systému, nedojde k přepsání tohoto souboru. Následně pomocí „`nano`“ vytvoříme ve složce soubor „`autostart`“

- `nano /home/(jméno našeho účtu)/.config/lxsession/LXDE-pi/autostart`



```
piBPSI@piBPSI: ~/.config/lxsession/LXDE-pi
GNU nano 7.2                                autostart
@xscreensaver -no-splash
@xset s off
@xset -dpms
@xset s noblank
@chromium-browser --incognito --kiosk http://192.168.50.35:3000
```

Obrázek 22 – Obsah souboru „`autostart`“ [vlastní zpracování]

Příkazy začínající s „x“ nám upravují náš přehrávač X11. postupně mu vypneme spořič obrazovky a další šetřící programy, aby pouštění obrazu bylo nepřetržité. V posledním řádku mu říkáme, aby při zapnutí zapnul *Chromium-browser* (ten je nainstalovaný při operaci instalace operačního systému) , dal ho pomocí parametru „—kiosk“ do režimu přes celou obrazovku a zapnul stránku našeho serveru („http://:ip_adresa_vysílajícího:3000“)

- `sudo sed -i -- "s/#xserver-command=X/xserver-command=X -nocursor/" /etc/lightdm/lightdm.conf`

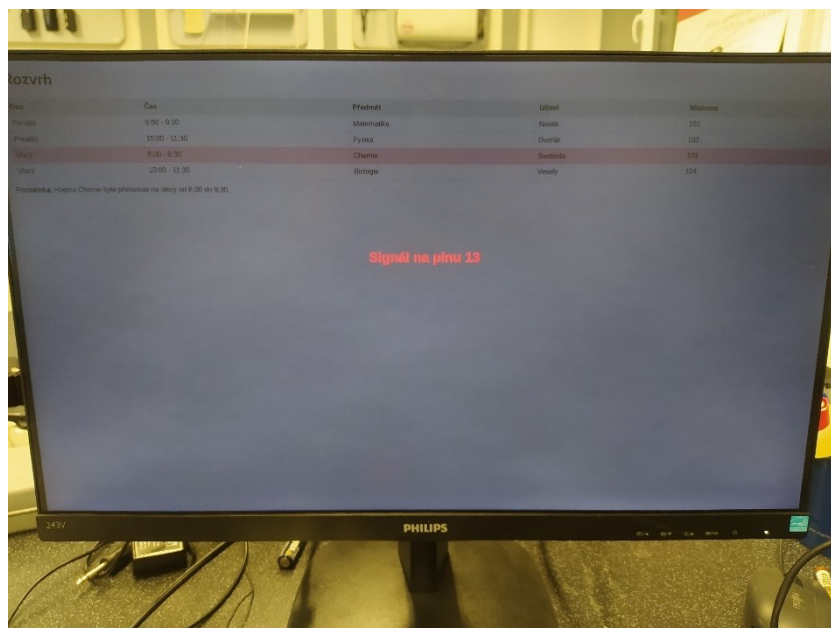
Poslední příkaz se znovu odkazuje na grafické prostředí v přehrávači X11 a vypne kurzor, aby nebránil ve výhledu.

Je nutné se ještě dostat do celkové konfigurace systému a přidat, aby při naběhnutí systému se rovnou dostal do plochy a nečekal na přihlášení uživatele. Díky tomu, kdyby nastala chyba, nám stačí zařízení pouze restartovat a nebude čekat na přihlašovací obrazovce, ale rovnou se spustí okno s *Chromium* prohlížečem a naší nastavenou stránkou. Konfigurace se dá změnit třeba příkazem „sudo raspi-config“.

Jelikož vše je nastavené v HTML kódu dané stránky a na serverové straně, není třeba zde stahovat nic navíc a jen sledovat stránku. Na stránce můžeme pozorovat základní rozvrh s funkcí sledování sepnutí obvodu na námi zvolených pinech 13 a 15, které odpovídají GPIO 27 a GPIO 22.

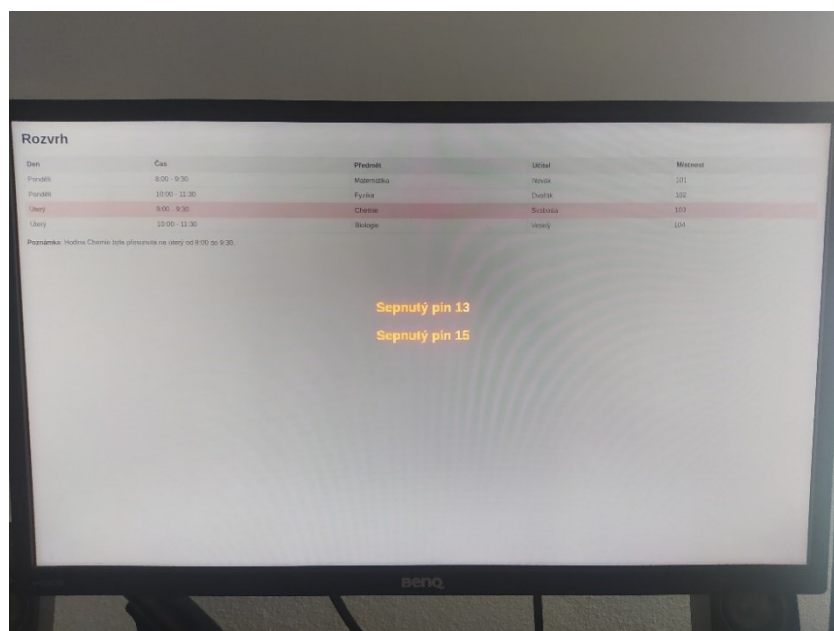
Při spuštění serveru bude stránka vysílána na IP adrese vysílajícího zařízení. Stránka při nesepnutí ukazuje na přehrávajícím zařízení pouze námi předepsaný rozvrh.

Při sepnutí jednoho z pinů, dojde k varovnému hlášení.



Obrázek 23 – Zobrazení varovného hlášení na displeji [vlastní zpracování]

Varovné hlášení je nastavené na monitorování, zdali je sepnutý obvod na pinu 13. Samotný text lze dle požadavků upravit v souboru „index.html“. Při sepnutí obou se varovná hlášení posunou a zobrazí se pod sebou. Při sepnutí obou bude stránka vypadat dle obrázku č. 24.



Obrázek 24 – Zobrazení obou varovných hlášení [vlastní zpracování]

ZÁVĚR

Cílem této práce bylo zjistit, zdali je možné zapojit malý počítač Raspberry Pi za ústřednu PZTS k vyhodnocení zpráv a zobrazit je na displayi. V teoretické části se věnuje seznámení čtenáře s jednotlivými bloky, kterými se tato práce zabývá.

V první kapitole popisuje, co to vůbec PZTS je, jaké jsou jeho části a jaké doplňky má. V druhé kapitole ukazuje, co je počítač Raspberry Pi, jaké doplňky má a s jakými systémy pracuje. Ve třetí části stručně říká, jakými způsoby lze využít Raspberry Pi k zobrazování informací. Ve čtvrté části popisuje funkční zapojení.

Tato práce vezme základní vstup PZTS. Ústředna následně vyhodnotí a sepne programovatelný výstup spojený s daným vstupem periferie, který sepne obvod pomocí elektromechanického relé. Díky PGIO na desce Raspberry Pi lze takovéto změny stavů monitorovat a dále s nimi pracovat. Dané změny jsou monitorovány a promítány na stránku, která je zobrazená na displayích.

Použití jednoduchého tlačítka beru jako dostatečné, jen je třeba dát pozor na falešné poplachy. Správným školením lze falešným poplachům zamezit. Jelikož stránky nejsou nijak programované ze strany *Sledujících* a jen se připojují na server neboli *Vysílajícího*, veškeré varovné zprávy se ukážou přímo na všech *Sledujících* v objektu. Tímto řešením se dá přímo v objektu varovat na případné hrozící nebezpečí.

V budoucnu vidím, že by se mohlo realizovat napojení této práce na kamerový systém. Za pomoci umělé inteligence, která se neustále vyvíjí, lze přepokládat, že implementace může nastat velmi brzy. Již dnes máme umělou inteligenci, která dokáže rozpoznat podle kamerového záznamu rozeznat, jaké emoce osoba cítí. [42]

Myslím si, že až bude umělá inteligence pokročilá na tolik, že z kamerového přenosu dokáže vyčíst, zdali pachatel je ozbrojen a případně čím, tak pomocí tohoto systému by dokázala efektivněji a přesněji upozornit na hrozbu. Dokonce by lépe upřesňovala veškeré jeho pohyby.

I kdyby se takovýto pokročilý stav podařilo zrealizovat, pořád si pevně stojím za možností fyzické spuštění, třeba pomocí avizovaných tlačítek. I při pokročilém stavu automatizace je lepší nechat možnost fyzického vstupu.

Celkově tato práce splnila svůj cíl, kterým bylo ověřit, zda je možné připojit malý počítač Raspberry Pi k ústředně PZTS (poplachové zabezpečovací a tísňové signalizace) za účelem

vyhodnocení zpráv a jejich zobrazení na displeji. V příloze se nachází video zobrazující funkční zapojení.

SEZNAM POUŽITÉ LITERATURY

- [1] ČSN EN 50131-1 ed. 2. Poplachové systémy – Poplachové zabezpečovací a tísňové systémy – Část 1: Systémové požadavky. Praha: Český normalizační institut, 2007. 40 s. Třídící znak 334591. . [cit. 2024-5-19]
- [2] NEČESAL, Luboš. Ústředny poplachového zabezpečovacího a tísňového systému. Online. 2012. Dostupné z: https://www.atpjournalsk/budovy/rubriky/prehladove-clanky/ustredny-poplachoveho-zabezpecovaciho-atisnoveho-systemu.html?page_id=14869. [cit. 2023-12-20]
- [3] Elektronická zabezpečovací signalizace (EZS). Online. Alarmtechnik Praha. Dostupné z: <https://www.alarmtechnik.cz/el-zabezpecovaci-signalizace>. [cit. 2023-12-20].
- [4] DRGA, Rudolf. Presentace: Poplachové zabezpečovací a tísňové systémy. Online. Dostupné z: <https://moodle.utb.cz/mod/folder/view.php?id=600585>. [cit. 2023-12-12]
- [5] PRODUKTY PERIMETRICKÉ OCHRANY. Online. Perimetrie. Dostupné z: <https://www.perimetrie.cz/produkty/>. [cit. 2023-12-20].
- [6] PZTS – technické zabezpečovací systémy pro ochranu majetku. Online. TOP security. Dostupné z: <https://www.topsecurity.cz/blog/clanek/pzts-technicke-zabezpecovaci-systemy-pro-ochranu-majetku>. [cit. 2023-12-20].
- [7] PIR detektor pohybu. Online. Jablotron. Dostupné z: <https://www.jablotron.com/cz/produkt/sbernicovy-pir-detektor-pohybu-203/>. [cit. 2024-5-19]
- [8] ČSN EN 50131-3. Poplachové systémy – Poplachové zabezpečovací a tísňové systémy – Část 3: Ústředny. Praha: Český normalizační institut, 2010. Třídící znak 334591. . [cit. 2024-5-19]
- [9] Ústředna JA-107K. online. Dostupné z: <https://www.jablotron.com/cz/produkt/gsm-ustredna-s-lan-komunikatorem-dodavano-s-ja-192y-ja-194y-nezahrnuty-v-cene-1237/>. [cit. 2024-5-19]
- [10] Poplachové zabezpečovací a tísňové systémy (PZTS). Online. DALTECH. Dostupné z: <http://www.daltech.cz/poplachove-zabezpecovaci-a-tisnove-systemy.html>. [cit. 2023-12-20].

- [11] Klávesnice s RFID čtečkou. Online. Jablotron. Dostupné z: <https://www.jablotron.com/cz/produkt/bezdratova-ctyrsegmentova-klavesnice-s-displejem-a-rfid-ctec-kou-antracitova-2210/>. [cit. 2024-5-19]
- [12] ČSN EN 50131-6 ed. 3. Poplachové systémy – Poplachové zabezpečovací a tísňové systémy – Část 6: Napájecí zdroje. Praha: Český normalizační institut, 2018. Třídící znak 334591. . [cit. 2024-5-19]
- [13] Komunikátory. Online. Jablotron. Dostupné z: <https://www.jablotron.com/cz/katalog-produktu/alarmy/jablotron-100/systemove-prvky/komunikatory/> . [cit. 2024-5-19]
- [14] GSM komunikátor. Online. Jablotron. Dostupné z: <https://www.jablotron.com/cz/produkt/modul-gsm-komunikatoru-1255/>. [cit. 2024-5-19]
- [15] Ústředny JA-107K a JA-103K zabezpečovacího systému JABLOTRON 100+. Online. Absolon. Dostupné z: https://www.absolon.cz/deploy/files/ja-107k-instal_cs_mmd20100.pdf . [cit. 2024-5-19]
- [16] About us. Online. Raspberry Pi Foundation. Dostupné z: <https://www.raspberrypi.org/about/> . [cit. 2024-5-19]
- [17] Dosya:Raspberry Pi Logo.svg. online. Wikipedia. Dostupné z: https://tr.wikipedia.org/wiki/Dosya:Raspberry_Pi_Logo.svg. [cit. 2024-5-19]
- [18] Horák, Milan. Raspberry Pi: Co to vlastně je. ABC Linuxu. Online. 2012. Dostupné z: <http://www.abclinuxu.cz/clanky/raspberry-pi-co-to-vlastne-je> [cit. 2024-05-20]
- [19] Lekce 7 - Velká rodina Raspberry Pi - Přehled modelů a jejich funkcí. Online. Itnetwork.cz. Dostupné z: <https://www.itnetwork.cz/hardware-pc/raspberry-pi/velka-rodina-raspberry-pi-prehled-modelu-a-jejich-funkci>. [cit. 2024-5-19]
- [20] RASPBERRYPI3-MODB-1GB. Online. Farnell Dostupné z: <https://cz.farnell.com/raspberry-pi/raspberrypi3-modb-1gb/sbc-raspberry-pi-3-mod-b-1gb-ram/dp/2525225>. [cit. 2024-5-19]
- [21] Raspberry Pi Pico. Online. Farnell. Dostupné z: <https://cz.farnell.com/raspberry-pi/raspberry-pi-pico/raspberry-pi-32bit-arm-cortex/dp/3643332>
- [22] Lekce 3 – Raspberry Pi – Piny a Protokoly. Online. Itnetworks.cz. Dostupné z: <https://www.itnetwork.cz/hardware-pc/raspberry-pi/raspberry-pi-piny-a-protokoly> [cit. 2024-5-19]

- [23] Použití GPIO pro binární vstup, Online. Microsoft. Dostupné z: <https://learn.microsoft.com/cs-cz/dotnet/iot/tutorials/gpio-input>. [cit. 2024-5-19]
- [24] Lekce 13 - Pestrý šatník Raspberry Pi - Přehled HAT modulů. Online. Itnetwork.cz. Dostupné z: <https://www.itnetwork.cz/hardware-pc/raspberry-pi/pestry-satnik-raspberry-pi-prehled-hat-modulu>. [cit. 2024-5-19]
- [25] 15 Best Operating Systems for Raspberry Pi. Online. RaspberryTips. 2019. Dostupné z: <https://raspberrytips.com/best-os-for-raspberry-pi/>. [cit. 2024-05-20].
- [26] Debian. Online. debian.cz. Dostupné z: <https://www.debian.cz/>. [cit. 2024-05-22].
- [27] Welcome to Raspbian. Online. Raspbian. Dostupné z: <https://www.raspbian.org/FrontPage>. [cit. 2024-05-22].
- [28] Operating system images. Online. Raspberry Pi. Dostupné z: <https://www.raspberrypi.com/software/operating-systems/>. [cit. 2024-05-22]
- [29] About OSMC. Online. OSMC. Dostupné z: <https://osmc.tv/about/>. [cit. 2024-05-23]
- [30] Ubuntu. Online. Ubuntu. Dostupné z: <https://ubuntu.com/>. [cit. 2024-05-23].
- [31] What Is DietPi, And Should You Install It On Your Raspberry Pi?. Online. SLASH FEAR. Dostupné z: <https://www.slashgear.com/1484447/what-is-dietpi-raspberry-pi-explained/>. [cit. 2024-05-23].
- [32] KOŘOUSKOVÁ, Barbora. Proč a kdy pro vývoj webových aplikací zvolit Ruby on Rails? Online. 2024. Dostupné z: <https://www.rascasone.com/cs/blog/vyvoj-webovych-aplikaci-ruby-on-rails>. [cit. 2024-05-23].
- [33] About Node.js ®. Online. node. Dostupné z: <https://nodejs.org/en/about>. [cit. 2024-05-23]
- [34] KOŘOUSKOVÁ, Barbora. Proč k vývoji webových aplikací použít technologii NodeJS? Online. 2024. Dostupné z: <https://www.rascasone.com/cs/blog/node-js-architektura-moduly-npm>. [cit. 2024-05-23].
- [35] Co je Front-End. Online. LER.STUDIO. Dostupné z: <https://lerstudio.cz/slovník/front-end>. [cit. 2024-05-23]
- [36] Co je frontend. Online. strafelda. Dostupné t: <https://www.strafelda.cz/frontend>. [cit. 2024-05-23]
- [37] Lekce 1 – Úvod do React, Online, Dostupné z: <https://www.it-network.cz/javascript/react/zaklady/uvod-do-react> [cit. 2024-5-19]

- [38] What is Angular in Web development? Online. SANITY. Dostupné z: <https://www.sanity.io/glossary/angular>. [cit. 2024-05-23].
- [39] SSH, Online, Itnetworks.cz Dostupné z: <https://www.itnetwork.cz/site/ssh> [cit. 2024-5-19]
- [40] MICHÁLEK, Martin. NPM: Průvodce začátky a základními příkazy. Online. Vzhůru dolů. 2018. Dostupné z: <https://www.vzhurudolu.cz/prirucka/npm#historie-konkurence>. [cit. 2024-05-23]
- [41] Array-gpio. Online. npmjs. Dostupné z: <https://www.npmjs.com/package/array-gpio>. [cit. 2024-05-23]
- [42] AI Emotion Recognition and Sentiment Analysis (2024). Online. viso.ai. Dostupné z: <https://viso.ai/deep-learning/visual-emotion-ai-recognition/>. [cit. 2024-05-23].

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

CSS	Cascading Style Sheets.
DNS	Domain Name System.
DRY	Don't Repeat Yourself.
DTMF	Dual-tone multi-frequency.
GPIO	General purpose Input/Output.
GPRS	General Packet Radio Service.
GSM	Groupe Spécial Mobile.
GUI	Graphic user interface.
HAS	Hold-up and Alarm System.
HDMI	High-Definition Multimedia Interface.
HTML	Hypertext Markup Language.
HTTP	Hypertext Transfer Protocol.
I&HAS	Intrusion and Hold-up Alarm System.
IAS	Intrusion Alarm System.
IoT	Internet of Things
JS	JavaScript.
LAN	Local area network.
LCD	Liquid crystal display.
LTE	Long-term evolution.
NC	Normally closed.
NO	Normály open.
NPM	Node Package Manager.
OS	Operační systém.
OSMC	Open source media center.
PCO	Pult centrální ochrany

PGM	Programovatelný výstup.
PTS	Poplachový tísňový systém.
PZS	Poplachový zabezpečovací systém.
PZTS	Poplašné, zabezpečovací a tísňové systémy.
RFID	Radio Frequency Identification.
RoR	Rail on rubber.
SMS	Short message service.
SSH	Secure Shell.
UI	User Interface.
USB	Universal service bus.

SEZNAM OBRÁZKŮ

Obrázek 1 – JA-110P PIR detektor [7].....	12
Obrázek 2 – JA-107K ústředna [9]	13
Obrázek 3 – JA-155E-AN Klávesnice s displejem a RFID čtečkou [11].....	14
Obrázek 4 – JA-192Y Modul GSM komunikátoru [14].....	15
Obrázek 5 – Záložka Sekce v programu F-Link [vlastní zpracování]	18
Obrázek 6 – Záložka Uživatelé v programu F-Link [vlastní zpracování]	19
Obrázek 7 – Záložka PG výstupy v programu F-Link [vlastní zpracování].....	20
Obrázek 8 – Záložka Kalendář v programu F-Link [vlastní zpracování].....	21
Obrázek 9 – Logo Raspberry Pi [17]	24
Obrázek 10 – Raspberry Pi 3 Model B [20]	25
Obrázek 11 – Raspberry Pi Pico [21]	26
Obrázek 12 – Popis GPIO a jednotlivých pinů na Raspberry Pi [23].....	26
Obrázek 13 – Diagram zapojení [vlastní zpracování]	36
Obrázek 14 – Zaznamenání periferie do systému PZTS [vlastní zpracování]	37
Obrázek 15 – Přiřazení vlastností PG výstupu [vlastní zpracování]	38
Obrázek 16 – Nastavení klávesnice [vlastní zpracování]	38
Obrázek 17 – Zapojení Raspberry Pi na PG výstup s relé.....	39
Obrázek 18 – Raspberry Pi Imager [vlastní zpracování]	43
Obrázek 19 – Rozhraní PuTTY [vlastní zpracování]	44
Obrázek 20 – Kontrola nainstalované verze programů [vlastní zpracování].....	45
Obrázek 21 – Struktura složky s prací [vlastní zpracování]	47
Obrázek 22 – Obsah souboru „autostart“ [vlastní zpracování].....	48
Obrázek 23 – Zobrazení varovného hlášení na display [vlastní zpracování]	50
Obrázek 24 – Zobrazení obou varovných hlášení [vlastní zpracování].....	50

SEZNAM PŘÍLOH

- CD

PŘÍLOHA P I: CD

Přiložené CD obsahuje:

- PDF soubor obsahující bakalářskou práci
- Video s ukázkou funkčního zapojení
- Kód pro realizaci serveru v JavaScriptu pomocí Node.JS
- Kód v HTML pro realizaci webové stránky