

České vysoké učení technické v Praze

Fakulta elektrotechnická

Katedra radioelektroniky

Embedded systém měřicí stolice v konceptu IoT
Embedded System for Measuring Stool in IoT Concept

Studijní program: Komunikace, Multimédia a Elektronika

Studijní obor: Multimediální technika

Vedoucí práce: Ing. Bc. Lukáš Vojtěch, Ph.D., katedra telekomunikační techniky

Ondřej Vítek

Praha 2018

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Vítek** Jméno: **Ondřej** Osobní číslo: **457168**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra radioelektroniky**
Studijní program: **Komunikace, multimédia a elektronika**
Studijní obor: **Multimediální technika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Embedded systém měřicí stolice v konceptu IoT

Název bakalářské práce anglicky:

Embedded System for Measuring Stool in IoT Concept

Pokyny pro vypracování:

Navrhněte a realizujte embedded měřicí systém v konceptu IoT pro měřicí stoličky rovinnosti kovových tyčí. Pro potřeby řešení snímejte signály z matice indukčních snímačů dle související dokumentace. Tyto signály zpracujte, vyhodnoťte a výsledek měření vhodným způsobem prezentujte obsluze s využitím programu pro vyhodnocení dat získaných z indukčních snímačů. Uložená data zobrazujte a archivujte na zabezpečeném webovém serveru, instalovaném na embedded platformě. Otestujte funkčnost a stabilitu navrženého systému.

Seznam doporučené literatury:

[1] DOMES, Martin. Tvorba WWW stránek pro úplné začátečníky. Brno: Computer Press, 2008. ISBN 978-80-251-2160-3.
[2] OpenSSL Cookbook [online]. 2. London: Feisty Duck, 2016 [cit. 2018-02-12]. ISBN 978-1907117053. Dostupné z: <https://www.feistyduck.com/library/openssl-cookbook/online/index.html>
[3] L. Vojtěch, Zařízení pro kontrolu přesnosti tyčového materiálu, patent CZ-306949 [cit. 2018-02-12]. Dostupné z: <http://isdv.upv.cz/doc/FullFiles/Patents/FullDocuments/306/306949.pdf>

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Lukáš Vojtěch, Ph.D., katedra telekomunikační techniky FEL

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

Datum zadání bakalářské práce: **19.02.2018** Termín odevzdání bakalářské práce: **25.05.2018**

Platnost zadání bakalářské práce: **30.09.2019**

Ing. Lukáš Vojtěch, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.
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

Anotace:

Tato práce se zabývá návrhem embedded systému pro měřicí stolic. Soustředí se na konfiguraci systému, na napsání vyhodnocovacího programu a na prezentaci výsledků na webový server. Dále prozkoumává další možnosti vylepšení procesu měření a zahrnuje analýzu celkového zabezpečení a stability.

Klíčová slova:

Embedded zařízení, Internet věcí, indukční sensor, Raspberry Pi 3, programovací jazyk, Python, zabezpečení, OpenSSL, webový server, OpenWRT/LEDE

Abstract:

This work deals with designing an embedded system for measuring stool. It focuses on system configuration, on writing an evaluation program and on presenting the results on a web server. The work also explores other possibilities for improving the whole process and includes analysis of security and stability of the whole system.

Keywords:

Embedded device, Internet of Things, Inductive sensor, Raspberry Pi 3, programming language, Python, security, OpenSSL, web server, OpenWRT/LEDE

Čestné 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 25.5.2018

.....

podpis bakalanta

Poděkování:

Rád bych poděkoval vedoucímu práce panu Ing. Bc. Lukáši Vojtěchovi, Ph.D. za to, že mi nikdy neváhal vyjít vstříc s konzultacemi, nikdy se mnou neztratil trpělivost a jeho vedení mi vždy pomohlo postoupit s prací. Dále děkuji panu Ing. Marku Nerudovi, Ph.D., panu Ing. Zbyňku Kocurovi, Ph.D. a panu Ing. Ondřeji Votavovi za jejich cenné rady. Velké poděkování patří i mé přítelkyni, rodině a přátelům za nekončící podporu.

Obsah

1. Embedded zařízení v průmyslu	7
2. Analýza zadání	8
3. Platforma	9
3.1 Úvod	9
3.2 Charakteristika.....	9
3.3 Hardware	10
3.4 Vstupní rozhraní	11
3.5 Operační systém	12
3.6 Testování fyzického rozhraní.....	13
4. Architektura řešení hardwaru	15
5. Bezpečnost	17
6. Indukční snímač.....	19
7. Program pro načtení hodnot.....	21
8. Přejechod na operační systém OpenWRT	23
8.1 Úvod k OpenWRT	23
8.2 Sestavení systému	24
8.2.1 Instalace a konfigurace Oracle VirtualBox.....	24
8.2.2 Kompilace	26
8.2.3 Nahrání systému.....	28
8.2.4 Konfigurace na Raspberry Pi 3.....	29
9. Program na načítání dat v OpenWRT	32
9.1 Prerekvizity	32
9.2 Program	33
10. Výstup vyhodnocených dat	34
10.1 Rozšíření LuCI	34
10.2 Zobrazení dat.....	35
11. Cenové zhodnocení systému	36
12. Budoucnost systému	38
13. Závěr	39
Seznam použité literatury	40
Seznam zkratk.....	45
Seznam použitých obrázků.....	47
Seznam tabulek	47
Přílohy.....	48

1. Embedded zařízení v průmyslu

V době všeobecného urychlení je setkání s embedded, tedy vestavěnými zařízeními, běžné, ať už v podobě bankomatů, řídicích jednotek v automobilech a letadlech, či jako součást zdravotnických přístrojů nebo ve vybavení domácnosti. Současnost přímo vyžaduje téměř okamžitou činnost různých systémů a k tomu jistě přispívají embedded zařízení. Tyto přístroje fungují jednoúčelově, obsahují řídicí počítač zcela zabudovaný do zařízení, které ovládá. Používají se převážně na jednu předem definovanou činnost a liší se tak od univerzálních počítačů, jako je například PC.

Za jeden z prvních vestavěných systémů je považován navigační počítač v projektu Apollo, který navrhl Charles Stark Draper z MIT v roce 1965 [1]. Přestože se tým projektu obával, že bude prvkem nejvíc ohrožujícím misi kvůli svým monolitickým integrovaným obvodům, byl nainstalován jak v samotné raketě, tak lunárním modulu.

Veřejnost se dočkala vestavěných systémů v kalkulačkách v 60. letech minulého století. Tyto systémy jsou v dnešní době využívány naprosto běžně, a to díky příchodu internetu a bezdrátového připojení. Embedded technologie úzce souvisí s automatizací výroby. Výrobní linka se skládá z několika dílčích robotických jednotek, které jsou ovládány právě jednoúčelovými přístroji. Průkopníkem v této oblasti byla společnost Ford [2]. Postupem času si sice jednotlivé výrobní komponenty zachovaly svou jedinečnou činnost, ale výrobní linka se propojovala a synchronizovala do jednoho celku a v dnešní době se do celého velkého systému mohou připojit i další části. Počínaje bezpečnostními okruhy a kamerami přes požární hlásiče až po chytré elektrické zásuvky, které vypadají podobně jako klasické, ale mohou se vypnout nebo zapnout přes aplikaci a v ní mohou měřit spotřebu energie a zobrazit statistiky [3]. Vše je možné díky Internetu věcí, tzv. IoT – Internet of Things, kde všechny přístroje a stroje jsou připojené na jednu síť. S tím souvisejí další oblasti jako servery a datová centra nebo cloudová řešení, přičemž ve všech uvedených sektorech najdeme embedded zařízení.

I běžný člověk se už může setkat s propojenými vestavěnými přístroji. Vyjma izolovanou elektroniku, tedy jednotky, které nekomunikují s jiným subjektem než s uživatelem, se novým trendem stávají tzv. smart domovy. Tímto se myslí domovy, které obsahují několik zařízení propojených IoT a uživatel je může ovládat například z mobilu nebo tabletu.

2. Analýza zadání

Tato práce je zaměřena na systém pro měřící stolicí rovinnosti kovových tyčí. Měřicí stolice byla zkonstruována na základě užitého vzoru [4] a jedná se tedy o unikátní přístroj. Díky embedded komponentě v tomto přístroji by byl přehlednější jak měřící cyklus, tak prezentace výsledků. Dále by se přístroj mohl propojit do většího systému díky IoT, i přesto, že získá článek navíc s odlišnou stabilitou a životností oproti zbytku systému a místo náchylné k možnému narušení bezpečnosti. Tyto aspekty jsou rozebrány v závěru práce.

Řešení by se mělo skládat z následujících kroků. Libovolná tyč je položena na nakloněnou rovinu se zabudovaným hnízdem 16 indukčních snímačů. Tyč následně přejezdí hnízdo snímačů, a ty pošlou data buď do embedded zařízení (při nízkém počtu snímačů), nebo nejprve do mikrokontroleru a až potom do hlavní jednotky. Pokud všechny snímače sepnou, tyč se bude považovat za rovnou a vyhodnocovací program ukáže, že je v pořádku. Pokud jeden či více snímačů nesepe, tyč se bude považovat za nerovnou a program vyhodnotí, že v pořádku není. V obou případech bude program exportovat data pro výstup ve webovém rozhraní. To by mělo mít zabezpečené připojení a jednoduché přihlášení. Alternativy a případné návrhy na budoucí postup jsou popsány v průběhu a na konci práce.

Samotné hardwarové řešení i embedded systém popsány v této práci jsou jedinečné. V prvních kapitolách je popsán přístup k jednotlivým částím procesu zvlášť a až dále jsou spojeny v jeden celek.

3. Platforma

V dnešní době je na trhu velký výběr embedded zařízení. Některá jsou zaměřena na výkon (Dragonboard 820c [5]), jiná na nízkou cenu (Raspberry Pi Zero [6]), některá obsahují více komponent (Orange Pi [7]), jiná méně. Raspberry Pi je jedno z nejznámějších a bylo zvoleno nejen z důvodu popularity a velké komunity uživatelů, ale také díky výhodnému poměru cena/výkon. Další předností Raspberry Pi je, že platforma již podstoupila určitý vývoj a lze od ní očekávat stabilitu.

3.1 Úvod

Raspberry Pi 3 je třetí generací jednodeskových počítačů s deskou plošných spojů od britské společnosti Raspberry Pi Foundation. Nachází se na obrázku 1. Je velký přibližně jako platební karta a byl vyvinut s cílem podpořit výuku informatiky zejména na školách. Lze ho používat k ovládání různých zařízení (osvětlení, spotřební elektronika). Cenově se pohybuje od 150 Kč do 1200 Kč (nejnovější verze). Primárním operačním systémem je Raspbian [8].



Obrázek 1.: Raspberry Pi 3 Model B [9]

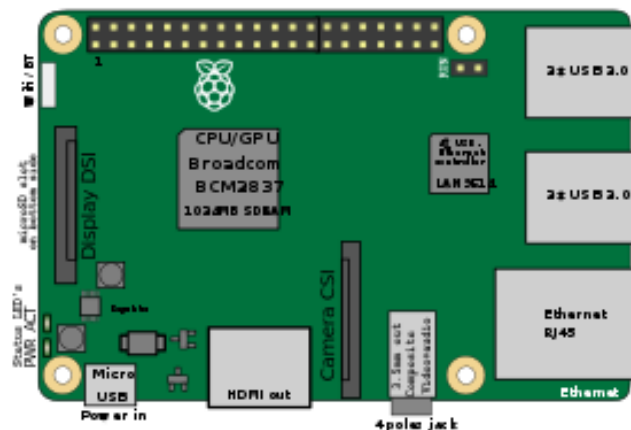
3.2 Charakteristika

Raspberry Pi je jednočipové zařízení, které je srovnatelné s méně výkonným stolním počítačem. Obsahuje rozhraní pro monitor (HDMI), přes USB je možné připojit klávesnici a myš. Vyvinuto bylo již několik generací tohoto počítače, které se liší výkonem a zamýšleným použitím. Zabudovaný mikroprocesor je z řady ARM. Podobné mikroprocesory se používají i ve smartphonech. Kromě OS Raspbian je na počítači Raspberry Pi možné provozovat i některé jiné distribuce Linuxu, RISC OS, nebo

Microsoft Windows 10 IoT Core. Výhodou je, že Raspberry Pi lze použít nejen jako samostatný počítač, ale i jako jednoúčelový embedded systém. Lze ho též využít jako multimediální přehrávač videa a hudby nebo pouze pro přístup k Internetu (dodávané linuxové distribuce obsahují webový prohlížeč a další potřebné aplikace). [8]

3.3 Hardware

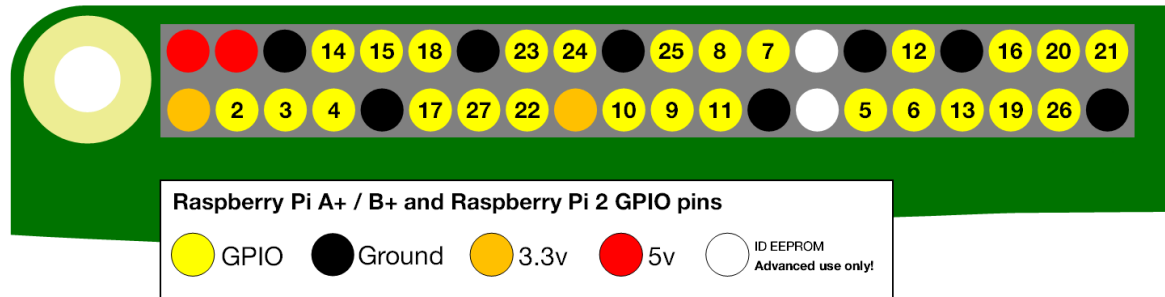
Raspberry Pi 3 je poprvé vybaven 64bitovým CPU o taktu 1,2 GHz, předchozí modely jej měly pouze 32bitové. CPU je čtyřjádro ARM Cortex-A53. Spolu s ním je také zabudována operační paměť SDRAM o velikosti 1 GB, sdílená s GPU. GPU podporuje OpenGL ES 2.0. Dále má zabudované čtyři USB 2.0 porty prostřednictvím on-board USB hubu s 5 porty, video vstup (CSI rozhraní MIPI, konektor, který se používá s kamerou Raspberry Pi nebo s Raspberry Pi NoIR), video výstup, např. HDMI, i DSI rozhraní MIPI, audio výstup (analogový přes 3,5milimetrový jack; digitální přes HDMI), úložný prostor (MicroSDHC slot), ethernetový port a GPIO rozhraní [3.4]. Další novinkou jsou integrované Bluetooth a Wi-Fi adaptéry [8]. Obrázek 2 znázorňuje rozložení některých komponent.



Obrázek 2.: Schéma hardwarových součástí [8]

3.4 Vstupní rozhraní

GPIO (General Purpose Input Output) piny se nacházejí ve dvou řadách na druhé straně od video výstupu HDMI. Tyto řady zahrnují 26 GPIO pinů, 8 pinů GND, 2 piny o 3,3 V, 2 piny o 5 V a 2 piny EEPROM [10]. Na obrázku 3 nalezneme číslování pinů, obrázek 4 zachycuje fyzický obrázek.



Obrázek 3.: Číslování GPIO a legenda [11]



Obrázek 4.: GPIO [12]

GPIO piny mají určité napěťové omezení. Na žádný z nich se nesmí přivést napětí větší než 3,3 V. Pokud se tak stane, velmi pravděpodobně se pin zničí [11].

3.5 Operační systém

Výrobce Raspberry Pi nabízí k počítači několik operačních systémů, které jsou uvedeny níže, a také ohlásil vývoj systému Rasdroid pro Raspberry Pi založený na systému Android 4.0. Protože je v Raspberry Pi použit mikroprocesor z rodiny ARM, je srovnatelný s běžným smartphonem. Dne 28. února 2014 uvolnila firma Broadcom ovladač grafického procesoru použitého v Raspberry Pi, pod Open Source BSD licencí [13]. Raspberry Pi neobsahuje žádné rozhraní pro pevný disk nebo SSD, pro zavedení systému a trvalé uchování dat je určen slot na MicroSD kartu.

Pro tuto práci byl vybrán operační systém Raspbian. Tento OS, odvozený od Debianu z unixové rodiny, je zdarma a open source. Alternativní operační systémy, které jsou podporované k instalaci jsou Arch Linux (pro Raspberry Pi), Rasdroid, dále Debian Jesse, NetBSD a W10 IoT Core a další [8]. OS W10 IoT Core nemá uživatelské rozhraní a dá se ovládat příkazovým řádkem pomocí Windows PowerShell.

Díky tomu, že je OS pro Raspberry Pi instalován na MicroSD kartu, dají se objednat už karty s předinstalovaným OS [14]. V této práci je použita naformátovaná, a tedy prázdná MicroSD karta bez předchozích instalací. K instalaci je potřeba PC nebo laptop s čtečkou karet.

Nejjednodušší způsob, jak nainstalovat daný OS pro Raspberry Pi, je pomocí programu/installeru NOOBS (New Out Of the Box Software), kde je i možnost výběru alternativních OS.

Nejprve je nutno stáhnout daný software z oficiálních stránek a rozbalit soubory [15]. Extrahované soubory poté přetáhneme na MicroSD kartu, vytáhneme ji z počítače a zasuneme do Raspberry Pi. Následně připojíme klávesnici, myš a video kabely a nakonec napájení. Tím se spustí první boot a díky NOOBS si můžeme vybrat, jaký OS chceme nainstalovat. Zde vybraný Raspbian je doporučený vývojářskou společností. Po instalaci se defaultně přihlašujeme pomocí přihlašovacího jména *pi* a hesla *raspberry* [16], [17].

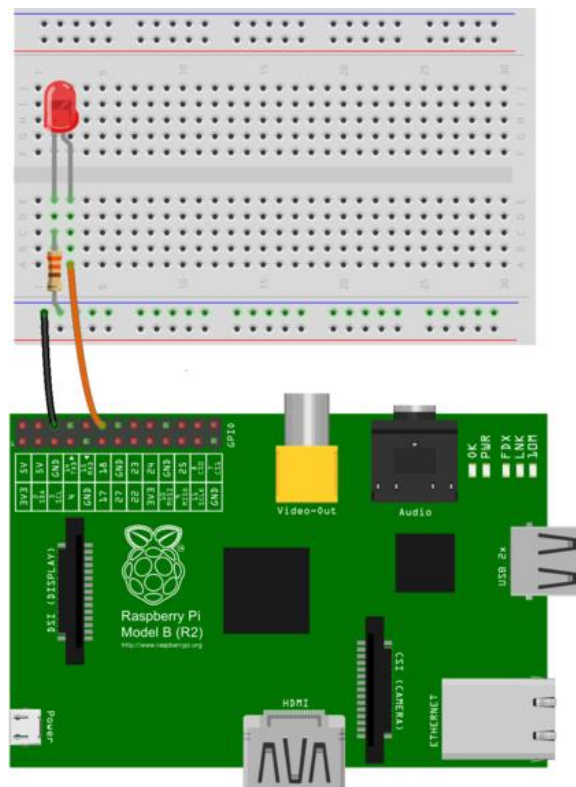
Jednotlivé OS se samozřejmě dají nainstalovat bez softwaru NOOBS. Výhodou NOOBS je přehlednější a jednodušší rozhraní a výběr OS.

Pokud je potřeba připojit Raspberry Pi k síti Wi-Fi, je možné použít Wi-Fi přijímač, který funguje ihned po zasunutí do portu USB.

3.6 Testování fyzického rozhraní

LED jsou jedním ze základních elektronických prvků a přístroj použitý v této práci by je mohl použít k signalizaci. Jedna dioda by mohla určovat, zda je zařízení zapnuté nebo vypnuté podle toho, zda by svítila nebo ne. Další by mohla být použita k signalizaci stavu: „přístroj je připraven“, „na přístroji probíhá měření“ nebo „přístroj nemůže měřit“. Poslední možností, jak využít diody, je vyhodnocení kovové tyče: „tyč je rovná“ nebo „tyč není rovná“. Diody mohou být napájeny externě nebo přímo z GPIO na zařízení Raspberry Pi.

Pro testování rozhraní GPIO byla připojena zkušební červená LED. Dále byly použity jumper wires, breadboard a Raspberry Pi 3. Zapojení bylo upraveno podle obrázku 5, který zachycuje starší model Raspberry Pi, konkrétně model 2B.



Obrázek 5.: Schéma zapojení s LED diodou [18]

Po zapojení byl v Raspberry Pi vytvořen nový soubor s názvem LED.py, napsán krátký program v programovacím jazyce Python a použit modul *gpiozero* [19] k ovládání pinů a modul *time* pro manipulaci s časovým úsekem.

```
from gpiozero import LED
from time import sleep #knihovny

led = LED(4) #dioda je na pinu
led.on()     #rozsvícení
sleep(3)     #na tři vteřiny
led.off()    #zhasnutí
```

Program se spouští příkazem `sudo python LED.py`. LED se rozsvítí a zhasne [18].

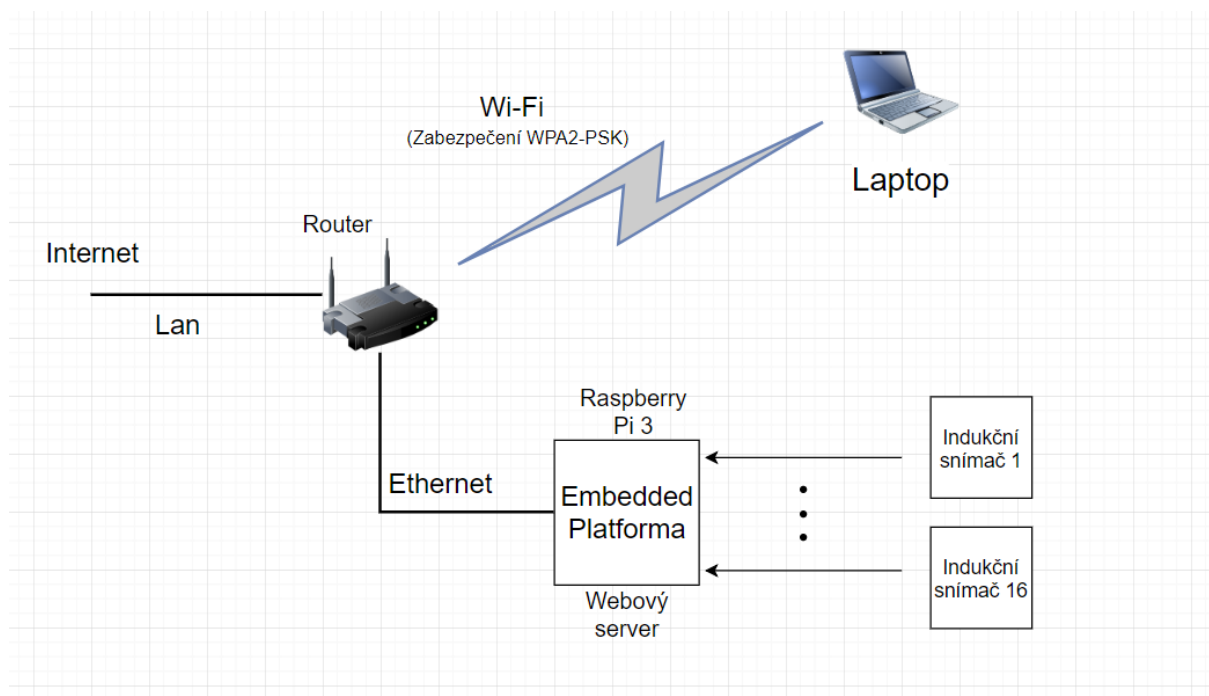
Programovací jazyk Python byl vytvořen v roce 1991 a je široce využíván. Jeho filozofie klade důraz hlavně na čitelnost a jednoduchost kódu, což jsou hlavní důvody, proč byl vybrán. Je to vyšší programovací jazyk a je open-source [20]. V této práci je použit také proto, že jsou v něm napsány moduly k ovládání GPIO pinů, jako například již použitý *gpiozero*, z jazyka C přepsaný *wiringPi* nebo níže použitý *RPi.GPIO*. Moduly jsou jednou z předností jazyka, existuje jich velmi mnoho a jsou velice rozmanité a užitečné. Další program bude napsán s pomocí jiného modulu, modulu jménem *RPi.GPIO* [21]. I když tvůrci *gpiozero* jsou součástí nadace Raspberry Pi Foundation, *RPi.GPIO* byl napsán již dříve a je tedy nejpopulárnější. Velká část příkladů na komunitních fórech je napsána právě s jeho pomocí [22].

4. Architektura řešení hardwaru

Celý systém je zapojen v jedné lokální síti. Internetové připojení poskytuje router. Uživatel s PC nebo laptopem je připojen k síti Wi-Fi, která mu umožňuje připojit se na webserver, který je spuštěn na embedded zařízení. Díky Wi-Fi adaptéru má i Raspberry Pi možnost připojit se k bezdrátové síti. Celkové propojení všech přístrojů můžeme vidět na obrázku 6.

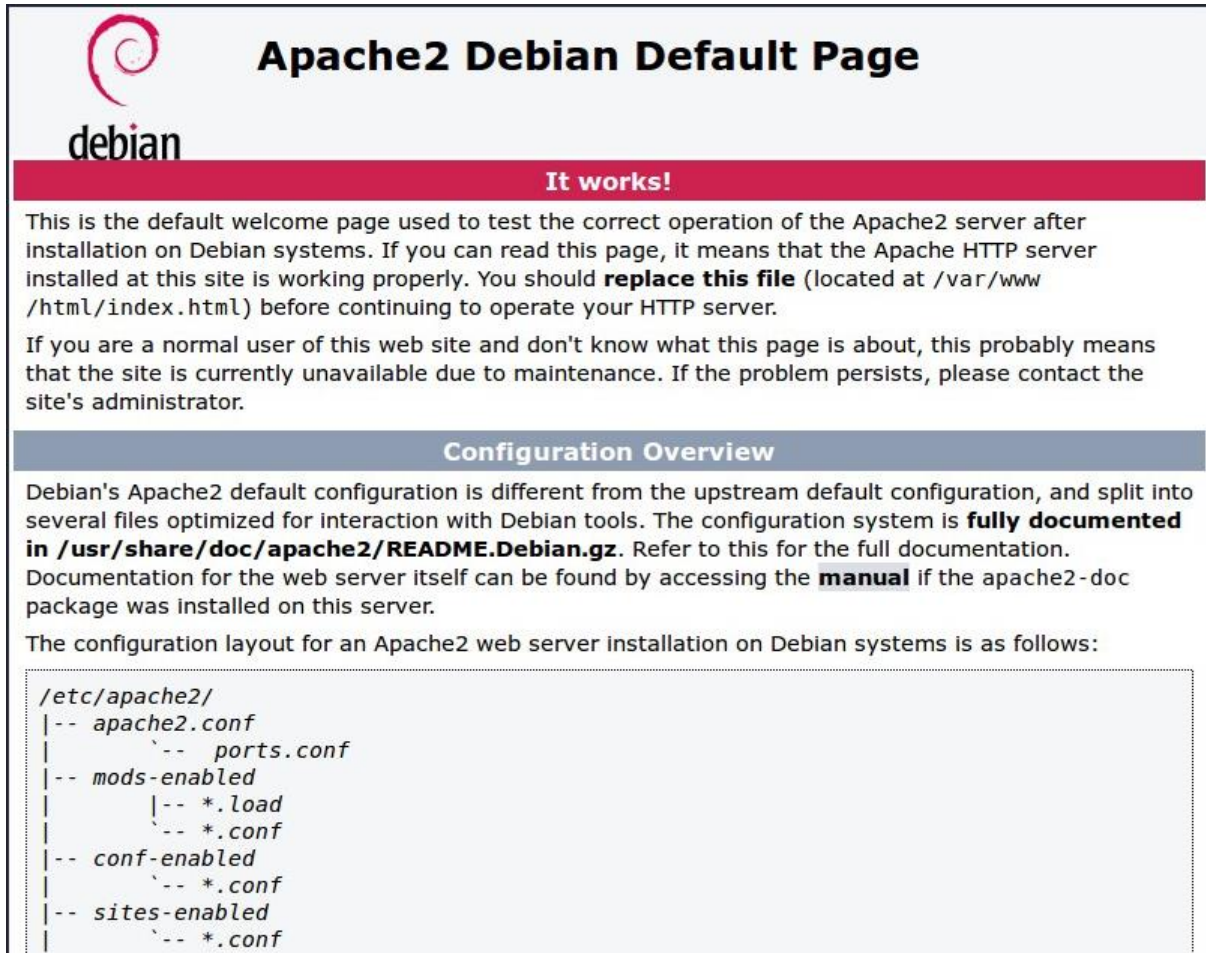
Signály, které jsou posílány z indukčních snímačů, jsou vyhodnoceny a vyhodnocení bude zobrazeno na webové stránce. Z toho důvodu byl na Raspberry Pi nainstalován webový server Apache HTTP.

Webový server v tomto použití je počítač (tedy Raspberry Pi 3), který je odpovědný za vyřizování požadavků HTTP od klientů (nejčastěji webových prohlížečů). Vyřizováním požadavků se rozumí odeslání cíle specifikovaného URL (typicky webová stránka, ale též statický text, obrázek, či jiný soubor). Webové stránky jsou obvykle dokumenty v jazyku HTML [23].



Obrázek 6.: Blokové schéma systému

Instalace webového serveru Apache se spouští příkazem `sudo apt-get install apache2 -y`, který nainstaluje vše potřebné. Až se instalace dokončí, můžeme již otestovat server ve webovém prohlížeči na adrese `http://localhost` nebo `http://` (adresa zařízení). Adresu zařízení zjistíme příkazem `hostname -i` v terminálu. Na obrázku 6 je vidět výchozí obsah.



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

Obrázek 7.: Defaultní obsah stránky

Defaultní obsah je možné měnit v souboru `index.html`, který se nachází ve `/var/www/html`, a zde by mohl být kód pro výslednou stránku. Stránka bude napsána v jazyce HTML [24].

5. Bezpečnost

Na samotné stránce je jednoduché zabezpečení formou přihlašovacích údajů, jména a hesla. Pokud se jméno a heslo shodují s nastavenou hodnotou, po kliknutí na tlačítko Login se uživatel dostane dál.

Modul nainstalovaný přímo v Apache webserveru, nazvaný OpenSSL, se stará o zabezpečené připojení, tzv. Secured Connection.

OpenSSL modul se skládá z kryptografické knihovny a toolkitu TLS/SSL (Transport Layer Security/Secure Sockets Layer). Projekt je open source a spravují ho dobrovolníci z celého světa [25].

Nejnovější verzi stáhneme pomocí příkazu `wget http://www.openssl.org/source/openssl-1.0.1p.tar.gz`.

Zabezpečení OpenSSL vyžaduje vygenerování unikátního klíče a certifikátu. Dále se buď certifikát pošle na příslušnou autoritu, aby ho schválila, nebo si uživatel certifikát podepíše sám pro testovací účely.

V práci byl použit testovací certifikát Self-Signed Certificate [26].

Unikátní klíče mají různou velikost, generují se pomocí různých algoritmů a uživatel si může nastavit heslo ke klíči.

OpenSSL podporuje RSA, DSA a ECDSA algoritmy, z nichž každý se hodí do jiných situací. RSA je nejrozšířenější a velmi podporovaný webovými servery.

Klíče pro RSA mohou být velké od 512 bitů, což však v současnosti již není bezpečné, až po 2048 bitů, které jsou nyní za bezpečné považovány.

Heslo ke klíči není povinné, ale je doporučováno. Zabezpečí se tak přístup ke klíči. Pro zobrazení některých jeho údajů nebo samotného klíče, je nutné zadat heslo.

V této práci byl vybrán klíč s algoritmem RSA o velikosti 2048 bitů, heslo bylo nastaveno.

Pro tuto konfiguraci byl použit příkaz `openssl genrsa -aes128 -out [jméno_klíče].key 2048`.

Klíče jsou ukládány ve formátu PEM, což je vlastně holý text [26].

Příklad několika řádků klíče:

```
vERmFJzsLeAEDqWdXX4rNwogJp+y95uTnw+bOjWRw1+O1qgGqxQXPtH3LWDUz1Ym
mkpxmIwlSidVSUuUrrUzIL+V21EJ1W9iQ71SJoPOyzX7dYX5GCAwQm9Tsb40FhV/
[21 lines removed...]
4phGTprEnEwrffRnYrt7khQwrJhNsw6TTtthMhx/UCJdpQdaLW/Tuy1aJMWL1JRW
i321s5me5ej6Pr4fGccNOe7lZK+563d7v5znAx+Wo1C+F7YgF+g8LOQ8emC+6AVV
```

Příkazem `openssl req -new -key key_name.key -out cert_name.csr` uživatel vytvoří soubor `.csr`, tedy Certificate Signing Request, který potom posílá buď na příslušnou autoritu (CA – Certificate Authority) nebo si ho jiným příkazem podepíše sám.

Při vytváření certifikátu nabídne program uživateli možnost vyplnit základní informace, což ovšem není povinné [26].

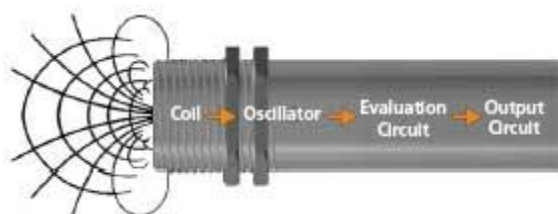
```
$ openssl req -new -key key_name.key -out cert_name.csr
Enter pass phrase for key_name.key: *****
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:CZ
State or Province Name (full name) [Some-State]:.
Locality Name (eg, city) []:Prague
Organization Name (eg, company) [Internet Widgits Pty Ltd]:.
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:localhost
Email Address []:[emailová adresa uživatele]
```

Pro testovací účely byla využita možnost podepsání certifikátu uživatelem. K tomu byl použit příkaz `openssl x509 -req -days 365 -in cert_name.csr -signkey key_name.key -out cert_name.crt`, který určuje, že certifikát s klíčem bude platný 365 dní [26].

6. Indukční snímač

Indukční snímače dokáží bezkontaktně nebo kontaktně detekovat přiblížení kovových předmětů. Jejich výhodou je možnost bezchybně pracovat v prašném nebo průmyslovém prostředí.

Snímač obsahuje cívku, za ní feritové jádro, oscilátor, vyhodnocovací okruhy a výstup. Tyto bloky na sebe navazují, jak je vidět na obrázku 8. Oscilátor pracuje na změně činitele jakosti jádra Q při přiblížení kovového materiálu. Tato změna se projeví útlumem a oscilátor přestane kmitat [27].



Obrázek 8.: Součásti snímače [27]

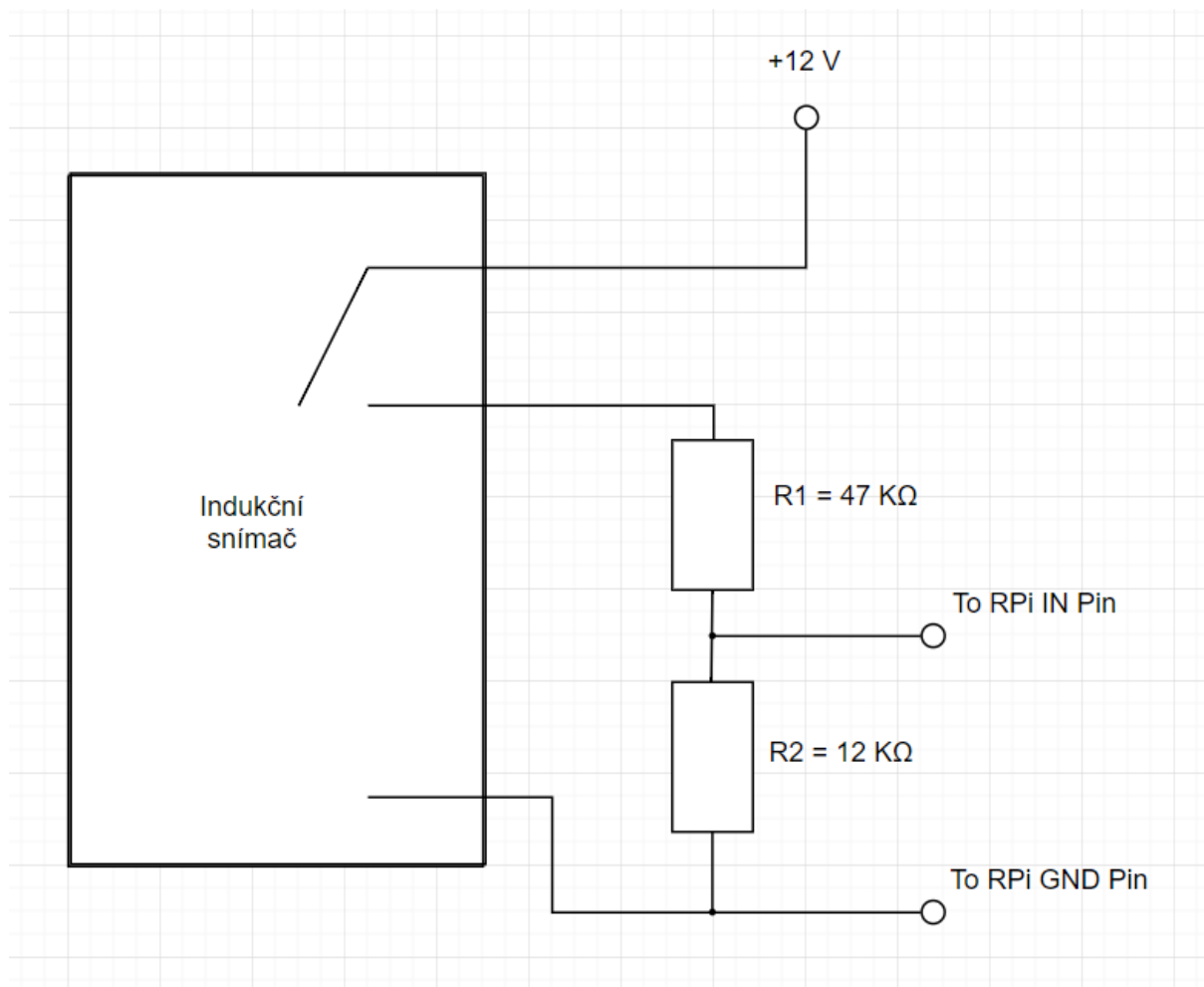
Pro tuto práci byl použit indukční snímač od společnosti Carlo Gavazzi s výrobním kódem EI0501PPOSS. Informace o něm jsou k nalezení na obrázku 9. Příloha 1 obsahuje celý datasheet.

Specifications

Rated operational volt. (U_B)	10 to 30 VDC (ripple included)	Usable operating dist. (S_u)	$0.85 \times S_r \leq S_u \leq 1.15 \times S_r$
Ripple	$\leq 10\%$	Ambient temperature	
Rated operational current (I_a)		Operating	-25° to +70°C (-13° to +158°F)
Continuous	$\leq 150 \text{ mA @ } +25^\circ\text{C (+75°F)}$	Storage	-30° to +80°C (-22° to +176°F)
No-load supply current (I_o)	$\leq 2.1 \text{ mA}$ OFF: 1.5 mA ON: 10 mA	Degree of protection	Plug versions: IP 65 Cable versions: IP 67 (Nema 1, 3, 4, 6, 13)
Voltage drop (U_d)	$< 1 \text{ V}$	Connection	
Protection	Reverse polarity, short-circuit, overload	Cable	2 m, 3 x 0.14 mm ² grey PUR, oil proof M8, CONH5A-xxx Serie
Transient voltage	$\leq 1 \text{ kV/0.5 J (prepared)}$	Plug	
Power ON delay	Ca. 500 μs	Housing material	
Frequency of operating cycles (f)	1000 Hz	Body	Stainless steel (1.4301)
Indication for output ON	LED, yellow	$\varnothing 4$	Aluminium
Assured operating dist. (S_a)	0 to 0.52 mm	5 x 5 mm	Plastic
Repeat accuracy (R)	$\pm 0.01 \text{ mm}$	Cap	
Hysteresis (H)		Weight	Cable versions: 25 g Plug versions: 4 g
(Differential travel)	3 to 15% of sensing distance	Tightening torque	1 Nm
Effective operating dist. (S_r)	$0.9 \times S_n \leq S_r \leq 1.1 \times S_n$	CE-marking	Yes

Obrázek 9.: Specifikace snímače

V datasheetu je uvedeno, že indukční snímač potřebuje napájení o alespoň 10 V, tím pádem nemůže být napájen přímo z Raspberry Pi. Jako napájecí zdroj byl tedy použit klasický adaptér o 12 V, připojený do standardní elektrické sítě. Od této chvíle je však potřeba zakomponovat odpory do zapojení, kvůli napětovému omezení na GPIO pinech. V zapojení byl použit napěťový dělič [obrázek 10].



Obrázek 10.: Schéma zapojení

7. Program pro načtení hodnot

Program je napsaný v programovacím jazyce Python a slouží k načítání dat z indukčního snímače, respektive z GPIO pinu, a k uložení těchto dat do CSV souboru. Díky nainstalovanému modulu *RPI.GPIO* mohou být ovládány piny, konkrétně pin 4.

```
import RPi.GPIO as GPIO
import time
import csv #knihovny

start =time.time() #do proměnné aktuální čas
PERIOD_OF_TIME = 120 #2 mins

GPIO.setmode(GPIO.BCM) #číslování pinů
GPIO.setup(4, GPIO.IN) #do pinu č. 4

x = [0, 0, 0] #pole s výsledky
y = 0 #pro cyklus a indexování pole

while y < 5:
    if time.time() > start+PERIOD_OF_TIME : break #končí program
    if y == 0:
        print('READY')
        y += 1
    elif y == 4:
        print('OK')
        y += 1
    else:
        #GPIO.add_event_callback(4, my_callback)
        print(str(y) + ", " + str(x[y-1]))
        print("Waiting for input..")
        while x[y-1] == 0: #cyklus čeká dokud nebude input 1 nebo neskončí program
            if time.time() > start+PERIOD_OF_TIME : break končí program
            mode = GPIO.getmode()
            if mode == None: #kvůli cleanupu, číslování znovu
                GPIO.setmode(GPIO.BCM)
                GPIO.setup(4, GPIO.IN)
            x[y-1] = GPIO.input(4)
            GPIO.cleanup()
        print(str(y) + ". position in array is " + str(x[y-1]))
        time.sleep(1) #kvůli rychlosti programu, aby se všechny kroky neudělali zároveň
        y += 1

with open('export.csv ', 'w') as f:
    writer = csv.writer(f)

    writer.writerow([x[0], x[1], x[2], 'OK']) #export do CSV souboru

input("Press Enter to continue..") #pouze na ukončení
```

Obsah CSV souboru po úspěšném vstupu:

1,1,1,OK

Při testování programu se na obrazovku vypíše dialog oznamující, že je program připraven. Vždy před tím, než program ohlásí, že čeká na vstup, se vypíše pozice v poli a aktuální obsah pole: 1, 0. To znamená první index pole s nulovou hodnotou. Po detekování vstupu se již pozice a její obsah napíšíou ve větě. Proběhnou tři opakování pro tři indexy pole. Když je vše správně, program potvrdí výpisem OK [obrázek 11].



```
Shell
>>> %Run TestExport.py
READY
1, 0
Waiting for input..
1. position in array is 1
2, 0
Waiting for input..
2. position in array is 1
3, 0
Waiting for input..
3. position in array is 1
OK
Press Enter to continue..
```

Obrázek 11.: Testování programu

8. Přejechod na operační systém OpenWRT

Kvůli zjednodušení celého vyhodnocovacího procesu a zkompletování dílčích sekcí (program, webserver, vyhodnocení, atd.) bylo, po konzultaci s vedoucím práce, rozhodnuto o dalším pokračování práce v OpenWRT/LEDE systému [28]. Tento systém je jinými slovy nakonfigurovatelný operační systém se zabudovaným webovým prostředím LuCI, ve kterém může uživatel po přihlášení jak upravovat nastavení a spravovat systém, tak zobrazit různé diagnostiky a ideálně i vyhodnocení programu. K tomu byly využity všechny nabyté zkušenosti z předchozích kapitol.

Tímto bylo navázáno na práci pana Bc. Ivana Eroshkina, který používal stejný operační systém, avšak na jiné platformě [29].

8.1 Úvod k OpenWRT

OpenWRT je operační systém založený na Linuxu a kompatibilní s různými embedded zařízeními. Primárně byl založen na firmwaru směrovače Linksys WRT54G, který použil kód licencovaný GNU licencí, a proto musel zveřejnit zdrojový kód. Ze začátku byl OpenWRT určen pouze pro směrovače série WRT54G (odtud OpenWRT), postupně se však projekt rozšířil i na další platformy [30].

V současnosti se projekt spojil s druhým projektem s názvem LEDE (Linux Embedded Development Environment) [31], který s OpenWRT sdílí spoustu cílů a funguje prakticky na stejné bázi. Poslední stabilní verze OpenWRT/LEDE je nazvána LEDE 17.01.4 na počest spojení dvou projektů. Je podporováno více než 1000 zařízení od většiny výrobců routerů, jako například TP-Link, Western Digital, ZyXEL a Linksys [32].

V této práci byla použita verze snapshot ze dne 12. 4., což je datum instalace. Snapshotové verze jsou aktualizovány většinou každé dva dny, někdy i každý den nebo dokonce několikrát denně [33].

Systém má spoustu užitečných vlastností a je sjednocen pro všechna zařízení, uživatel má tedy svobodu a není omezen firmwary od různých výrobců.

Mezi vlastnosti patří:

- nastavitelné metody manipulování síťových paketů, například Firewall [34], přesměrování, traffic shaping [35] nebo statistiky a diagnostiky v reálném čase,
- systém lze konfigurovat pomocí BusyBox ash, což je forma bashe [36], nebo pomocí webového rozhraní LuCI (viz níže),
- zapisovatelný souborový systém,
- správce balíčků *opkg*, který umožňuje instalovat nebo odebírat balíčky [37],

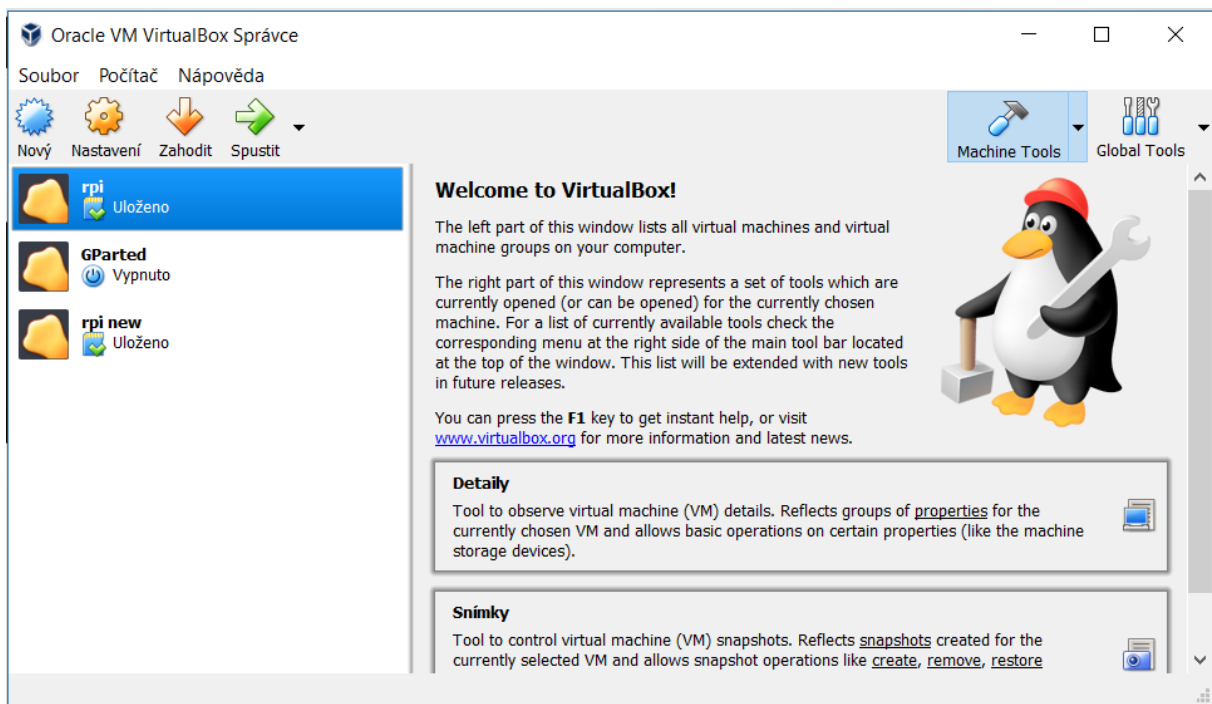
- balíčkový repozitář s více než 3000 balíčky,
- statické zapůjčení DHCP [38],
- pravidelné aktualizace i pro zařízení již nepodporované výrobcem,
- webové uživatelské rozhraní LuCI, kde je možné operovat s většinou nastavení, je chráněné heslem a napsané v programovacím jazyku Lua, v němž je snadnější například přidávat vlastní položky [39].

8.2 Sestavení systému

Celý proces se skládá ze stažení potřebných souborů, kompilace, nahrání na MicroSD kartu a následné konfigurace. Tyto dílčí kroky jsou popsány níže.

8.2.1 Instalace a konfigurace Oracle VirtualBox

Po stažení a instalaci Oracle VirtualBox je nutno také stáhnout image OS Raspbian, soubor .iso. Nové virtuální prostředí bylo v této práci pojmenováno Rpi, typ byl vybrán Linux a verze Other Linux, 32bitová. Velikost paměti závisí na tom, kolik chce uživatel paměti procesoru uvolnit, v této práci bylo uvolněno 1024 MB. Je důležité, aby velikost virtuálního hard disku byla větší než 12-16 GB, aby kompilace proběhla bez komplikací. Proces zabere nejméně 7-8 GB (na webové stránce je uvedeno, že stačí 3-4 GB, testování ukázalo, že je potřeba nejméně dvojnásobek), část disku zabírá OS, takže je výhodné již na začátku nastavit větší velikost [40]. Pokud už nastavení hard disku proběhlo, Oracle VirtualBox neumožňuje upravení v uživatelském rozhraní a pro zvětšení hard disku již nastaveného virtuálního stroje se musí použít jiné virtuální prostředí, kde se připojí image disku Gparted, což je manažer oddílů pevného disku a může se použít i pro upravení virtuálních disků [41]. Pro práci bylo nastaveno 48 GB. Na obrázku 12 se nachází uživatelské rozhraní Oracle VirtualBox s několika nainstalovanými virtuálními stroji.

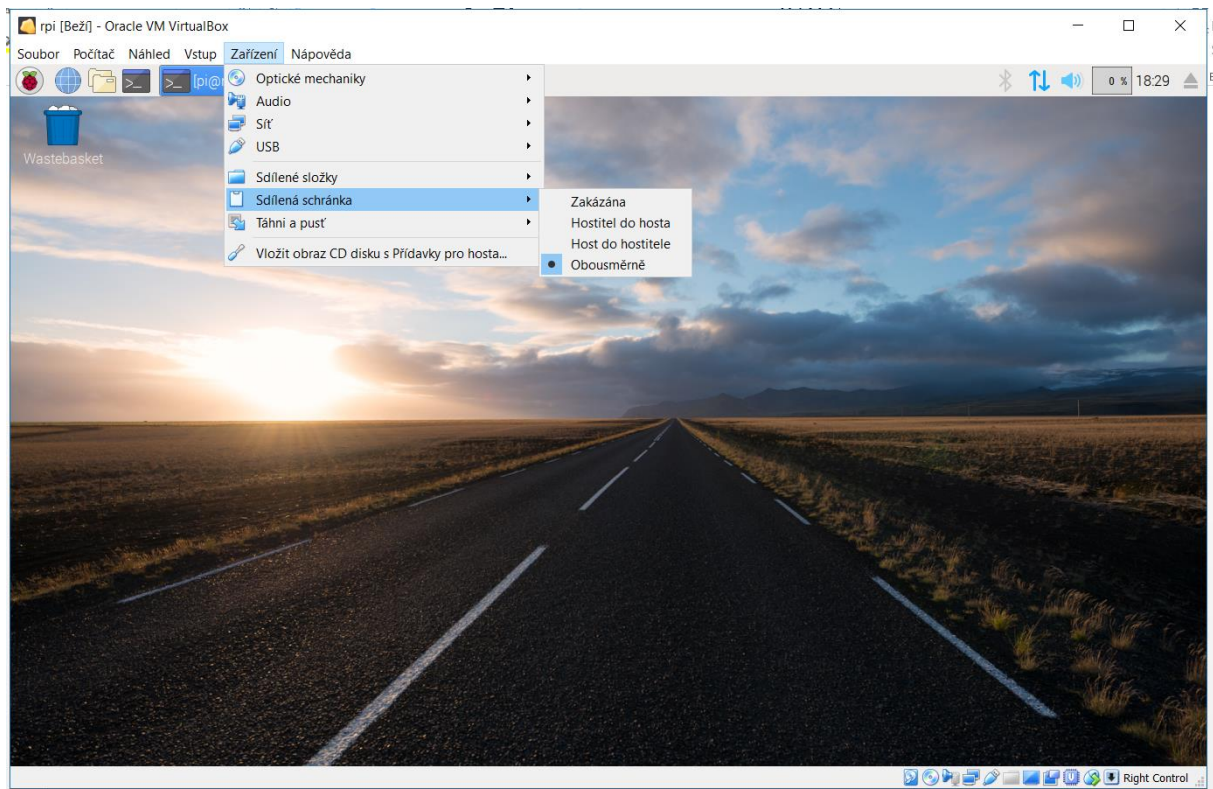


Obrázek 12.: VirtualBox se třemi virtuálními stroji, mezi nimi i GParted

Následuje první zapnutí virtuálního stroje, ke kterému musí být připojen stažený image OS Raspbian. Nejprve je třeba nainstalovat OS, samotnou instalaci je uživatel proveden i v grafickém rozhraní. Pro nastavení byly většinou voleny výchozí hodnoty. Když systém dokončí instalaci, objeví se chybové hlášení a virtuální stroj se vypne. V nastavení musí být povoleno rozšíření adresního prostoru (Enable PAE/NX). Při spuštění by už OS neměl vykazovat žádné chyby [40].

Pro zlepšení práce mezi reálným a virtuálním strojem se musí nainstalovat tzv. VirtualBox Guest Extensions. Toto umožní kopírování mezi oběma stroji. Po instalaci se schránka pro kopírování oběma směry musí povolit v nastavení Oracle VirtualBox [40]. Toto nastavení je zachyceno na obrázku 13.

Program Oracle VirtualBox má dvě velmi užitečné vlastnosti. První z nich je ta, že virtuální systémy vytvořené ve VirtualBoxu dědí internetové připojení mateřského PC. Tato funkce funguje ihned po spuštění a nemusí se nikde nastavovat. Druhou vlastností je uložení stavu virtuálního počítače při vypnutí. Tato funkce zmrazí všechny procesy a nahraje je při dalším zapnutí, přesně v tu chvíli, kdy uživatel ukončil činnost.



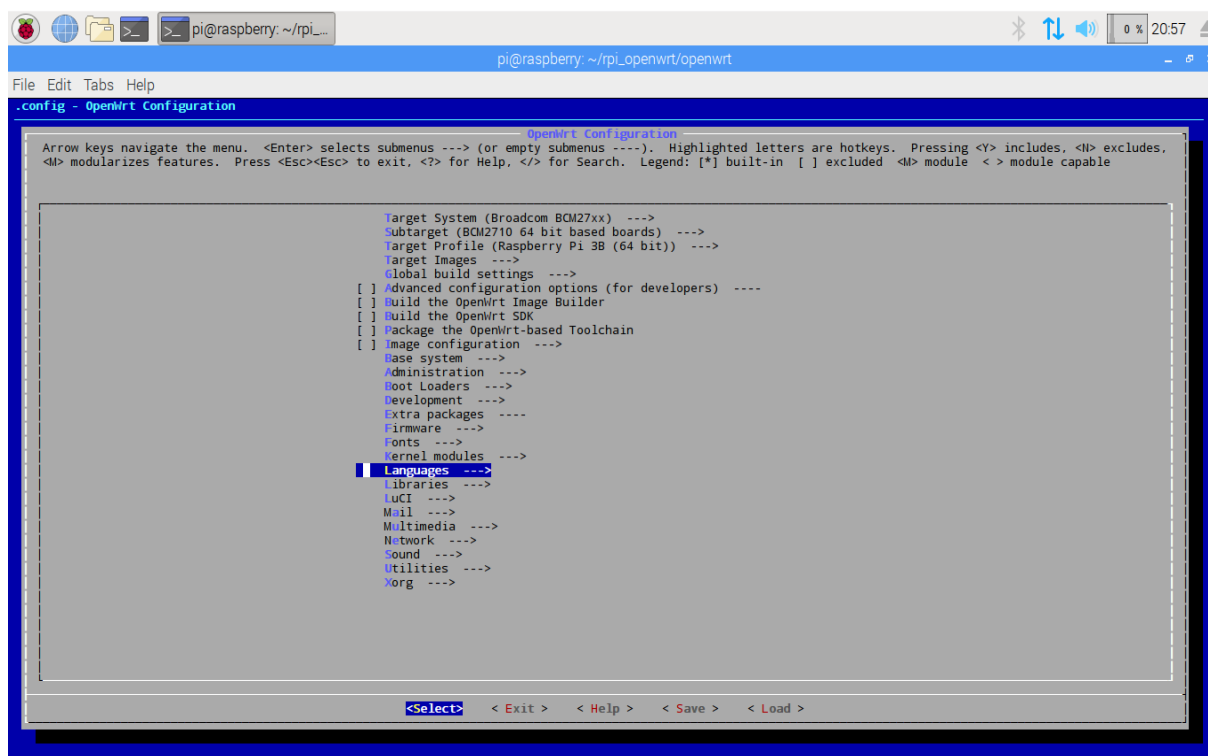
Obrázek 13.: Virtuální prostředí OS Raspbian s nastavením sdílené schránky

Virtuální prostředí bylo vybráno kvůli jednoduššímu postupu kompilace systému OpenWRT na OS Raspbian. Díky Oracle VirtualBox bylo i nastavení relativně snadné a přehledné. Samotný VirtualBox je jedním z nejpopulárnějších programů k vytváření virtuálních strojů, jeho hlavní předností je, že je zdarma.

8.2.2 Kompilace

Typicky se před každou instalací nejprve stáhne poslední update nainstalovaných balíčků pomocí příkazu `sudo apt-get update`. Dále musí být nainstalován `git-core` k snadnému stažení softwaru OpenWRT a další pomocné nástroje k celému procesu: `sudo apt-get install git-core build-essential libssl-dev libncurses5-dev unzip gawk zlib1g-dev subversion mercurial` [42].

Příkazem `git clone https://github.com/openwrt/openwrt.git` se vytvoří nová složka `openwrt` a do ní se stáhne trunk verze (nejnovější stabilní). Poté je nutno přejít do složky `openwrt`, stáhnout a nainstalovat feeds, tedy skupiny balíčků ze společného odkazu. Samotné balíčky nejsou instalovány ani kompilovány, pouze je jim umožněno být k dispozici [43]. Dalším příkazem `make menuconfig` lze otevřít menu konfigurace [obrázek 14], kde si uživatel může navolit vlastní možnosti instalace.



Obrázek 14.: Konfigurace OpenWRT před kompilací

Po vybrání platformy, architektury a modelu, existuje varianta globálního nastavení v položce Global build settings, která zaktivuje všechny moduly nezbytné pro daný model platformy a uživatel si může vybrat další volitelné moduly bez toho, aniž by se vyskytla chybová hlášení, pokud by některé povinné moduly chyběly. Pro tuto práci byl navíc zvolen balíček programovacího jazyka Python, který byl využit k testování v dřívějších kapitolách. Největší podporu v systému OpenWRT má Python ve verzi 2.7 [44].

Nyní může uživatel dodat vlastní balíčky nebo knihovny, které nejsou v modulech v menu konfigurace. OpenWRT je sice postavený na Unixu, ale neumí některé operace jako například Raspbian, který byl využit k testování v prvních fázích práce. V OpenWRT není možno stahovat z GitHubu, kde je umístěna většina užitečných knihoven, které nejsou součástí hlavních balíčků. Není povolen ani příkaz *make*, ten je možno však zaktivovat v menu konfigurace a po kompilaci by měl být dostupný.

Až uživatel dokončí výběr balíčků a uloží konfiguraci, může nastat samotná kompilace. Ta se spouští již uvedeným příkazem *make*. Pokud je potřeba zobrazit všechny procesy a případná chybová hlášení, kdyby se kompilace nedokončila, použije se parametr *V=s* za příkaz *make*. Doba kompilace závisí na volné paměti procesoru a počtu jader [44].

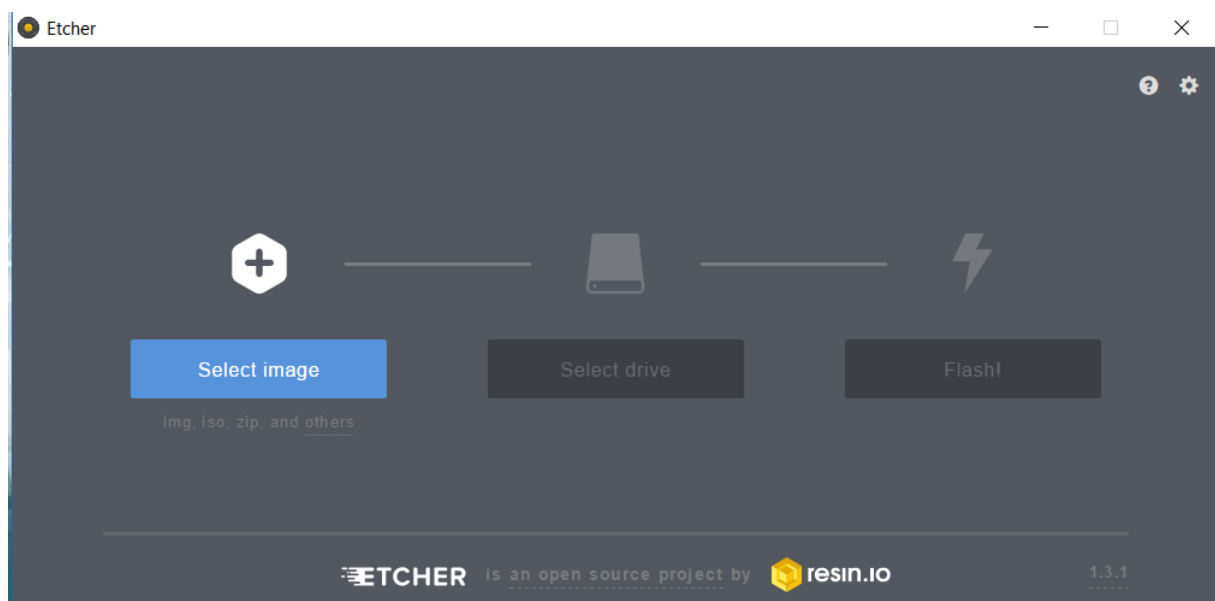
8.2.3 Nahrání systému

Úspěšné dokončení kompilace vytvoří image OS OpenWRT ve složce `/openwrt/bin/targets/brcm2708/bcm2710`. Pokud by se kompilace týkala jiné platformy, a tedy i jiného „terče“, poslední složky by se jmenovaly podle jeho názvu.

V této práci byl pro nahrání image OS na microSD kartu použit program Etcher.io nainstalovaný na mateřském PC. Aby mohl být soubor image přesunut na reálný stroj, byla zvolena metoda sdílených složek. Uživatel nejprve musí přidat cílovou složku v mateřském PC (v daném případě *sharePC*) do sdílených složek v nastavení programu VirtualBox. Dalším krokem je vytvoření sdílené složky ve virtuálním stroji, popřípadě použití již existující. V této práci byla vytvořena nová s názvem *shareVB*. Příkazem `sudo mount -t vboxsf sharePC ~/shareVB/` se složky propojí a cokoli je uchováno ve složce *shareVB* ve virtuálním stroji, je k nalezení a manipulování i ve složce *sharePC* v mateřském OS [45].

Oracle VirtualBox také nabízí, po dalším nastavení, funkci drag & drop, kdy uživatel může jednoduše přetáhnout soubor z virtuálního stroje do mateřského PC.

Když je image systému OpenWRT dostupný na reálném PC, může se přejít k nahrání OS OpenWRT na MicroSD kartu. Program Etcher.io [obrázek 15] je velmi snadno ovladatelný a dostupný zdarma [46]. Proces probíhá pouze ve třech krocích, prvním je vybrání image souboru, druhým stanovení média, kam se image nahraje, v tomto případě naformátovaná MicroSD karta o velikosti 16 GB. Detekování média může proběhnout automaticky. Posledním krokem je samotný proces vypálení, neboli nahrání OS na MicroSD kartu, který je následně spustitelný po zasunutí karty do příslušného portu.

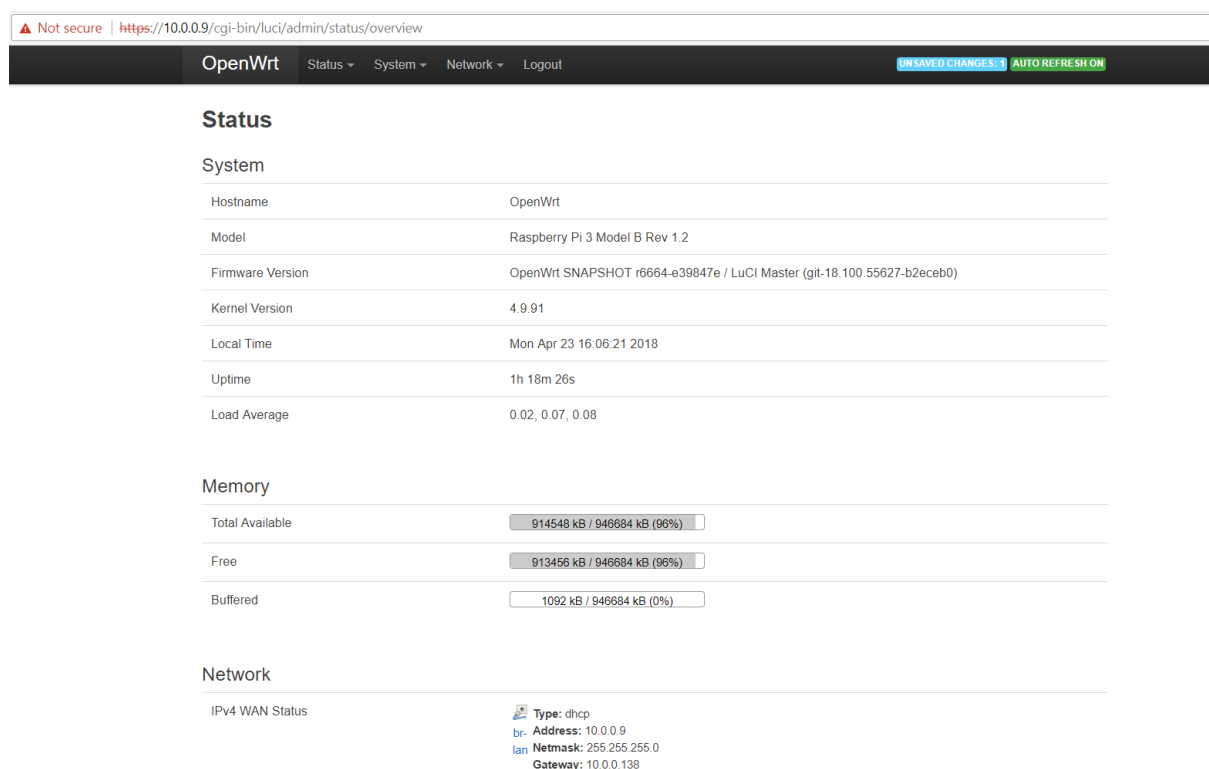


Obrázek 15.: Program Etcher.io

8.2.4 Konfigurace na Raspberry Pi 3

Nejprve musí být MicroSD karta zasunuta s image OS OpenWRT do Raspberry Pi 3, a poté musí být připojeno napájení, což spustí bootovací sekvenci. Systém OpenWRT si poradí i s USB klávesnicí, pokud byly porty USB vybrány při kompilaci (toto nastavení je zahrnuto v Global build settings). Systém uživateli navrhuje nastavení zabezpečení, tedy hesla. Pokud by proběhl reboot před nastavením hesla, veškerá konfigurace, která již proběhla, by byla smazána a spustilo by se výchozí nastavení. I přesto byl v této práci nejprve nastaven dynamický protokol v síťovém nastavení příkazem `uci set network.lan.proto=dhcp` a znovu načtena konfigurace příkazem `reload_config`. Příkazem `ip -4 a s br-lan | grep inet` byla poté zobrazena lokální IP adresa zařízení [47].

IP adresa může být zadána do webového prohlížeče na PC a po varování o nezabezpečeném připojení a neplatném certifikátu, se načte uživatelské prostředí LuCI [obrázek 16]. Autorizace certifikátu je uvedena níže. Stránka opět hlásí nutnost zabezpečit přístup heslem, zatím se tedy uživatel přihlásí bez hesla s přihlašovacím jménem `root`. Konfigurace hesla se nastaví kliknutím na odkaz ve varovném hlášení a vyplněním příslušných polí.



The screenshot shows the OpenWRT LuCI Status page. The browser address bar displays `https://10.0.0.9/cgi-bin/luci/admin/status/overview`. The page header includes the OpenWRT logo, navigation menus for Status, System, Network, and Logout, and buttons for 'UNSAVED CHANGES!' and 'AUTO REFRESH ON'. The main content is divided into three sections: Status, Memory, and Network.

Status

System	
Hostname	OpenWrt
Model	Raspberry Pi 3 Model B Rev 1.2
Firmware Version	OpenWrt SNAPSHOT r6664-e39847e / LuCI Master (git-18.100.55627-b2eceb0)
Kernel Version	4.9.91
Local Time	Mon Apr 23 16:06:21 2018
Uptime	1h 18m 26s
Load Average	0.02, 0.07, 0.08

Memory

Total Available	914548 kB / 946684 kB (96%)
Free	913456 kB / 946684 kB (96%)
Buffered	1092 kB / 946684 kB (0%)

Network

IPv4 WAN Status	Type: dhcp br- Address: 10.0.0.9 lan Netmask: 255.255.255.0 Gateway: 10.0.0.138
-----------------	--

Obrázek 16.: Úvodní stránka LuCI

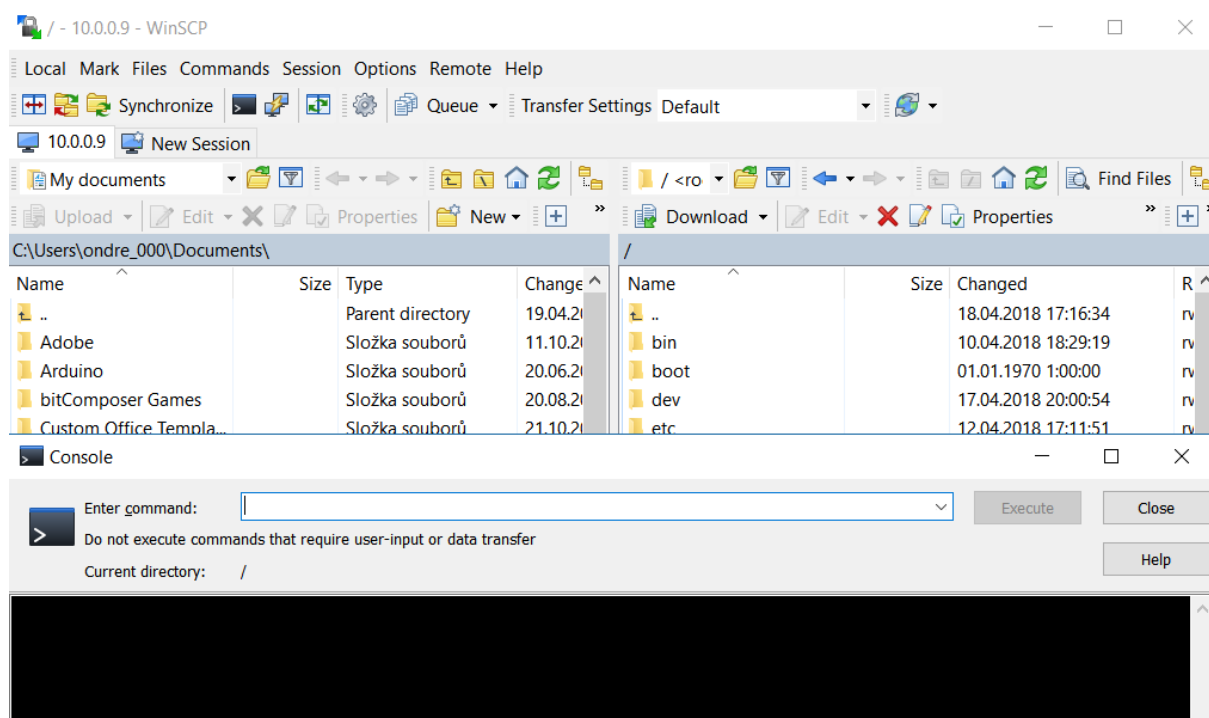
Na jednotlivých stránkách je opět k vidění velmi mnoho síťových možností, např. zapnutí a konfigurace nové Wi-Fi sítě a diagnostických nástrojů, různé grafy provozu a zatížení Wi-Fi sítě.

Uživatel má možnost instalace dalších balíčků v sekci System > Software, ale pouze těch, které jsou v hlavní databázi v OpenWRT repozitáři na webu GitHub.

Jako další funkci, kterou musíme povolit zvlášť, je SSH (Secured Shell), tedy zabezpečený vzdálený terminál. Jinými slovy si uživatel bude moci spustit příkazový řádek systému OpenWRT na jiném stroji, například na laptopu nebo PC. Funkci SSH v systému OpenWRT je možné povolit ve webovém rozhraní LuCI v sekci System > Administration, kde pod nastavením hesla uživatel nalezne i nastavení SSH. Když je konfigurace uložena, může být otevřen terminál. V této práci byl použit program WinSCP [48].

Před spuštěním programu WinSCP je třeba v systému OpenWRT doinstalovat balíček s názvem *openssh-sftp-server*, který je nezbytný k tomu, aby program WinSCP fungoval, jelikož nefunguje pouze jako terminál nebo příkazový řádek, ale umožňuje uživateli procházet OpenWRT způsobem podobným Windows OS [48].

Ihned po spuštění programu WinSCP se otevře možnost připojení se do sítě právě díky funkci SSH. Poté, co uživatel vyplní údaje, jako adresu, uživatelské jméno a heslo, se terminál vzdáleně připojí do systému OpenWRT. Na obrázku 17 vidíme dva sloupce složek, v pravém je vzdálený přístroj, v daném případě OS OpenWRT, v levém sloupci je hostující PC.



Obrázek 17.: Program WinSCP se spuštěným terminálem

Po těchto krocích je možné přejít k nastavení zabezpečeného připojení mezi PC a zařízením se systémem OpenWRT. OpenWRT automaticky poskytuje zabezpečení stahování balíčků nebo SSH, i přihlášení do LuCI je zabezpečené nastaveným heslem, nicméně prohlížeč potřebuje další informace

o zabezpečení ve formě certifikátu, který může uživatel buď podepsat sám, a nebo poslat příslušné autoritě k podepsání. V této práci certifikát slouží k testování a je tedy podepsán obdobně jako v kapitole 4.

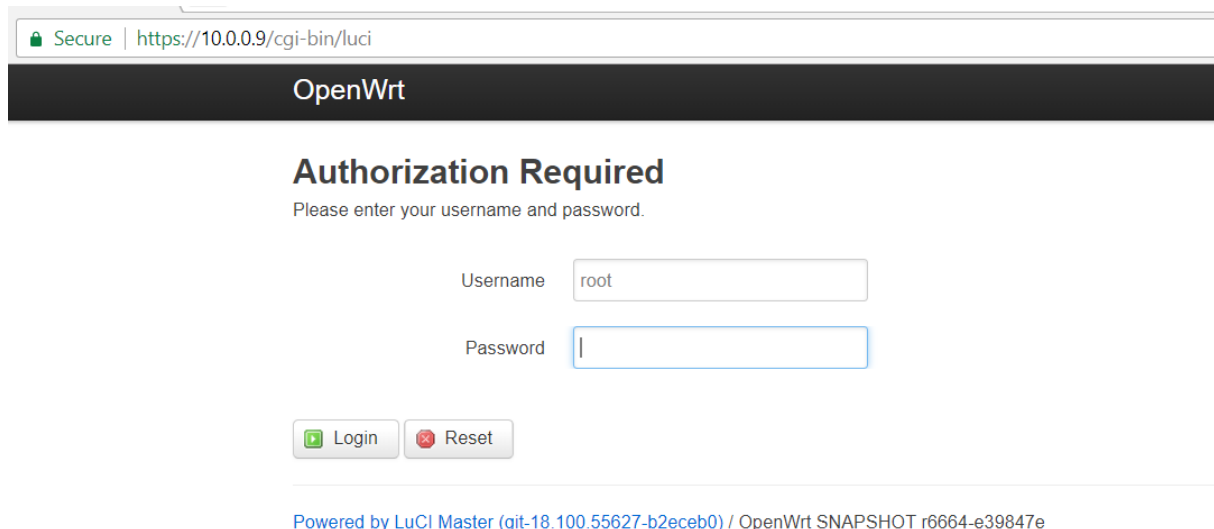
Je potřeba nainstalovat další balíčky, *libopenssl*, *openssl-util* a *luci-app-uhttpd*. Ve složce */etc/ssl/* byl vytvořen soubor *configssl.conf* s tímto obsahem:

```
[req]
distinguished_name = req_distinguished_name
x509_extensions    = v3_req
prompt             = no
[req_distinguished_name]
C                  = CZ
ST                 = SK
L                  = Prague
O                  = Home
OU                 = Router
CN                 = 10.0.0.9
[v3_req]
keyUsage           = keyEncipherment, dataEncipherment
extendedKeyUsage   = serverAuth
subjectAltName     = @alt_names
[alt_names]
DNS.1              = 10.0.0.9
IP.1               = 10.0.0.9
```

Začátek textu slouží k rozeznání jmen, prohlížeč Google Chrome použitý k této práci od verze 58 zkoumá jak Common name (tedy běžné jméno), tak i alternativní názvy. Další informace slouží k upřesnění polohy a při podepisování certifikátu bez toho, aby byl poslán do příslušné autority, jsou volitelné. Po uložení je třeba spustit příkaz *openssl req -x509 -nodes -days 730 -newkey rsa:2048 -keyout mycert.key -out mycert.crt -config configssl.conf*, který podle výše vytvořeného souboru *configssl.conf* vytvoří dva soubory, šifrovací klíč a podepsaný certifikát. Tyto soubory, respektive jejich cestu, musí uživatel uložit ve webovém rozhraní LuCi, v sekci Services (kterou nově vytvořil balíček *luci-app-uhttpd*) > uHTTPd. Změny se projeví po restartu uHTTPd příkazem */etc/init.d/uhttpd restart* [49].

Aby prohlížeč uznal tento certifikát, je nutné jej importovat do Důvěryhodných certifikačních autorit – Trusted Root Certification Authorities. Vygenerovaný certifikát si můžeme exportovat z OS OpenWRT a uložit ho v prohlížeči Chrome do sekce Nastavení > Pokročilé > Spravovat certifikáty. Pokud je

konfigurace provedena v prohlížeči Chrome, ostatní prohlížeče by měly certifikát akceptovat také [49]. Po restartu varování zmizí a uživatel uvidí, že připojení je zabezpečené [obrázek 18].



Obrázek 18.: Rozhraní LuCI se zabezpečeným připojením

9. Program na načítání dat v OpenWRT

Balíček programovacího jazyka Python, vybraný v kapitole 2, byl nainstalován při kompilaci a bude použit i v systému OpenWRT. V předchozích kapitolách je použit modul *RPi.GPIO*, ten ovšem nebude využit v systému OpenWRT, neboť není v hlavní databázi balíčků. Jeho instalace do systému OpenWRT se nepodařila z důvodu chybného konfiguračního souboru. S GPIO se tedy bude pracovat jako se soubory. OS Raspbian má přístup k GPIO umožněn již v prvotním nastavení, v systému OpenWRT je potřeba přístup udělit.

9.1 Prerekvizity

Hardware konfigurace je možná v souboru */boot/config.txt*, kde se případné nastavení promítne ihned při zapnutí nebo restartování systému. Uživatel dle potřeby upravuje tzv. Device Trees, datové struktury, popisující hardwarové komponenty určitého zařízení, v tomto případě Raspberry Pi 3. Tyto struktury slouží k tomu, aby s nimi kernel, tedy jádro OS, mohl pracovat [50]. V této práci byl použit parametr *dtoverlay=lirc-pi*, který umožňuje práci s GPIO. [51] Overlay, v doslovném překladu překrytí, je určité globální nastavení datové struktury pro GPIO. K systému v této práci je zapotřebí otevřít 18 GPIO pinů. Modelově lze pin 2 otevřít parametrem *gpio_in_pin=2*, který vložíme pod overlay [52].

Dále je nutné požádat kernel, aby exportoval informace o GPIO do souborů. To lze provést vložením čísla pinu do souboru *export* příkazem `echo "2" > /sys/class/gpio/export`. Obdobná procedura musí být

provedena i pro ostatní piny. U všech GPIO je také třeba nastavit směr, a to sice příkazem `echo "in" > /sys/class/gpio/gpio2/direction`. Tyto příkazy pro každý pin byly napsány do souboru `/etc/rc.local`, jehož příkazy se vykonají při spuštění systému [53].

9.2 Program

Samotný program by měl být pro uživatele přehledný a co nejjednodušší na ovládání. Měl by také poskytnout uživateli instrukce, jak s ním pracovat. V programu musí být zahrnuto zobrazení výsledku jak na obrazovce, tak na webovém serveru. Program by neměl skončit po jednom měření, aby ho uživatel nemusel ručně zapínat znovu, ale měl by nabídnout opakované měření.

Program v této práci nejdříve importuje nezbytné knihovny, poté přivítá uživatele a na obrazovku vypíše instrukce. Na pozadí se připraví text na export do HTML, do kterého se po měření přidá výsledek, cesty ke všem souborům, odkud importujeme hodnotu na pinech a cestu do složky pro uložení HTML souboru, a také metoda s názvem `GPIOread` pro čtení hodnot ze souborů. Dále se inicializuje pole o 16 buňkách. Do jednotlivých buněk se uloží hodnota 1, sepne-li příslušný snímač.

Ovládání funguje následovně. Kovová tyč je položena na začátek měřicí stolice, tím se sepne první snímač na GPIO2 a program je připraven načítat hodnoty. Připravenost signalizuje vypsáním hesla `READY` (připraveno) a zapnutím časovače, pro případ, že by tyč nedojela na konec měřicí stolice. Časovač je nastaven na 60 sekund. Předpokládá se, že tyč do té doby přejede vyhodnocovací hnízdo 16 snímačů a dorazí na konec, kde sepne poslední snímač na GPIO5. Program signalizuje ukončení cyklu vypsáním hesla `DONE` (dokončeno). Hned poté se na obrazovce objeví pole s načtenými hodnotami, podle kterých program zobrazí zprávu „Iron rod is flat.“ („Kovová tyč je rovná.“) nebo „Iron rod is not flat.“ („Kovová tyč není rovná.“). Tato informace se uloží i do jiné proměnné. Dále program spojí všechny příslušné proměnné typu `string`, aby dohromady vytvořily kód v jazyce HTML, který se následně uloží do složky `/usr/lib/lua/luci/view/luci-indsensor/`, jejíž vytvoření je popsáno v kapitole 11.1. Úspěšné uložení se objeví na obrazovce ve formě zprávy „The program has updated the webpage.“ („Program aktualizoval webovou stránku.“).

Program se poté zeptá uživatele, zda chce provést další měření. Po stisku „y“ jako `yes` cyklus pokračuje od začátku, po stisku „n“ se program vypne.

Na začátku programu bylo provedeno importování dvou modulů. První z nich se jmenuje *os* a přidává příkazy k ovládání funkcí operačního systému, například práci s adresářem složek [54]. Druhý s názvem *time* umožňuje ovládání funkcí souvisejících s časem, například získání aktuálního času.

K napsání tohoto programu bylo čerpáno z následujících materiálů: Kniha Python 3 [55], výukový kurz na stránce wikibooks [56] a online videokurz na stránce StackSkills.com [57].

Testování programu se nachází na obrázku 19. Příloha 2 obsahuje celý zdrojový kód programu.

```
root@OpenWrt:~# python main.py
Welcome to the program - start the measurement by putting the iron rod on the stool.
READY

DONE.
[1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]

Iron rod is not flat.

The program has updated the webpage.
Another measurement? (Confirm with enter) y/n: n
root@OpenWrt:~# |
```

Obrázek 19.: Příklad jednoho cyklu měření

10. Výstup vyhodnocených dat

Webové rozhraní LuCI je zabezpečené heslem a přehledné, proto je to vhodné místo, kde prezentovat výsledek vyhodnocení dat. Na horní liště je možnost přidání další sekce neboli stránky, což pro uživatele bude snadnější, než aby hledal v jiných podsekcích. Většina volitelných modulů si také vytváří své vlastní okno, zde je výsledek zobrazen v *Data > Results*. Je-li uložen nový soubor s výsledky, stránka se automaticky neaktualizuje, a proto bude přidán skript na obnovení v určitém intervalu.

10.1 Rozšíření LuCI

K rozšíření webového rozhraní LuCI jsou zapotřebí dva kroky. Prvním z nich je vytvoření tlačítka na horní liště, které bude odkazovat na novou stránku. Druhým krokem pak bude vytvoření samotné stránky, kde bude výstup dat. Poté je nutné přesunout se do složky */usr/lib/lua/luci/controller* a vytvořit novou složku *myapp*. Zde se uloží nový soubor *new_tab.lua*, který do horní lišty LuCI přidá sekci *Data* a zároveň tlačítko *Results*, které se objeví, když uživatel přejede sekci *Data* kurzorem myši [58].

Obsah souboru:

```
module("luci.controller.myapp.new_tab", package.seeall)
    --new_tab odkazuje na soubor new_tab.lua

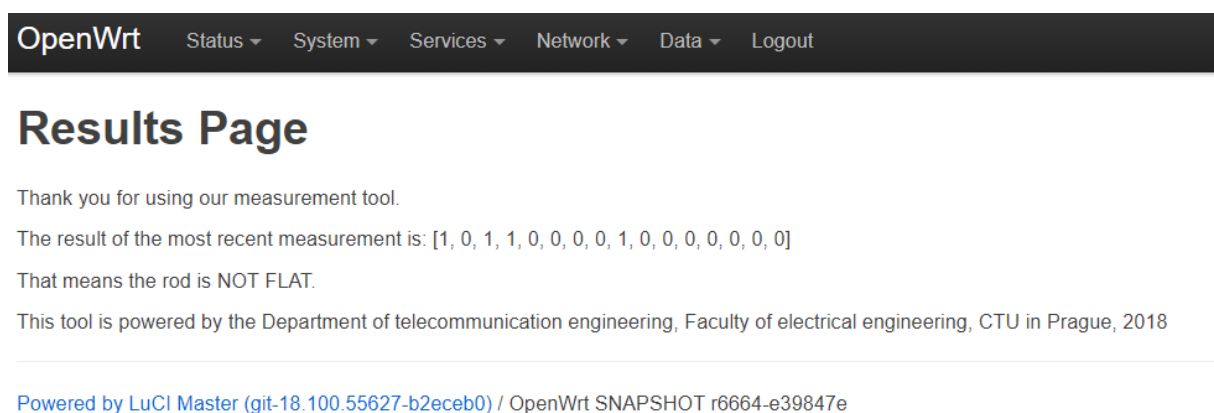
function index()
    entry({"admin", "new_tab"}, firstchild(), "Data", 60).dependent=false
    --tento řádek přidá sekci do horní lišty a zároveň se odkáže na stránku Results po kliknutí
    entry({"admin", "new_tab", "tab_from_view"}, template("luci-
        indsensord/index"), "Results", 1)
    --toto přidá tlačítko Results a po kliknutí odkazuje na soubor html, který se nachází ve
        složce luci-indsensord

end
```

Poslední příkaz odkazuje na složku a soubor, který zatím není vytvořen. Ve složce `/usr/lib/luas/luci/view` vytvoříme složku `luci-indsensord`, v níž se bude nacházet soubor `index.htm`, který se bude generovat programem pro vyhodnocení vždy po proběhnutí měřicího cyklu.

10.2 Zobrazení dat

Prezentace výsledků musí být přehledná a také automatická. V této práci byly uvedené informace omezeny pouze na výsledky měření. Do HTML kódu byl přidán příkaz `setTimeout(location.reload.bind(location), 6000);` z jazyka JavaScript, který znovu načte stránku každých 6 vteřin. Verzi JavaScriptu, která obsahuje tento příkaz podporují všechny moderní prohlížeče [59]. Výstup dat lze nalézt na obrázku 20.



Obrázek 20.: Stránka s výsledky

11. Zhodnocení navrženého systému

Slabé místo stability zapojení systému v této práci je v úseku indukční snímač - Raspberry Pi 3. Piny, které načítají data, by při vstupu měly mít napětí alespoň 2,3 V [62], první experimenty v této práci probíhaly pod napětím 2,4 V. Při kolísání napětí by pin mohl zůstat v nedefinovaném stavu.

Se stabilitou úzce souvisí životnost jednotlivých komponent. Indukční snímače se stálým zdrojem elektrického proudu, v místnosti beze změn teploty, mohou vydržet velmi dlouho [60]. Systém se ale pravděpodobně bude nacházet v průmyslovějším prostředí, a tak bude životnost nižší, kolem 10 let. Raspberry Pi 3 může fungovat i přes 10 let, zde životnost souvisí na teplotě a údržbě [61]. Velmi důležitý prvek Raspberry Pi 3 je MicroSD karta, na kterou se nahrává OS, ta má životnost nižší, mezi 5 a 10 lety. Karty jsou však náchylnější k poruchám a může se stát, že selže dříve [62].

Zabezpečení přihlášení uživatele je zakomponováno formou uživatelského jména a hesla. Připojení chrání šifrovací klíč a certifikát. OpenSSL, modul se kterým jsme při vytváření tohoto systému pracovali, podporuje nanejvýš šifrování RSA 2048bitové, což je v roce 2018 stále považováno za bezpečné [63]. Prognóza je taková, že ještě dvě dekády může vydržet, a potom bude nutné přejít na složitější šifrování [63]. Bezpečnost můžeme zvýšit tím, že platnost certifikátu zkrátíme, tím se ale zvýší nároky na údržbu, kvůli častějšímu obnovování. S bezpečností samozřejmě souvisí i zabezpečení internetového připojení hlavní sítě. Pokud by byl systém připojen k rozsáhlejší síti nebo cloudu, potenciální útočník by se případně mohl dostat až citlivým datům. Spojením projektů OpenWRT a LEDE, získal OS několik bezpečnostních prvků už při výchozím nastavení. Kernel i uživatelská prostředí jsou chráněna proti tzv. „stack smashing“, což je typ útoku, kdy program sbírá data z nedůvěryhodného zdroje a potenciální útočník může vložit spustitelný kód do už běžícího programu [64]. Tím může útočník získat přístup do systému. Dále systém obsahuje RELRO – Relocation Read Only, techniku, která chrání spustitelné soubory nebo linkovatelné objekty, tedy soubory ve formátu ELF [64]. Systém obsahuje několik dalších ochran [64].

Ekonomické zhodnocení je k nalezení v následující tabulce:

Název součástky	Počet kusů	Cena
Raspberry Pi 3 Model B+	1x	979,- Kč
Napájecí zdroje	2x	498,- Kč
MicroSD 16GB	1x	239,- Kč
HDMI kabel, zlatený, 2m	1x	87,- Kč
Síťový kabel, 1m	1x	59,- Kč
Indukční senzor, Carlo Gavazzi	1x	900,- Kč (přepočten z \$)
Celkem:		2.762,- Kč

Tabulka 1.: Ekonomické zhodnocení

V případě započítání všech 18 indukčních snímačů by cena vystoupala na přibližně 18.000,- Kč bez napájecích zdrojů.

Pro tuto práci byly součástky zapůjčeny katedrou telekomunikační techniky. Ceny byly převzaty z webových stránek [65], [66].

12. Budoucnost systému

Data mohou být ukládána buď na interní paměť nebo na cloud. Ukládání na interní paměť je lepší možností v případě, že systém není součástí větší sítě. Pokud její součástí je a síť už cloudové řešení obsahuje, bude výhodnější ukládat data na cloud, všechna data se pak budou nacházet na jednom místě. Na webové stránce by potom měla být další sekce, například s názvem Archiv, kam by se ukládala dřívější měření. Záznam v této sekci by měl obsahovat datum měření, identifikační číslo tyče a výsledek měření. Pokud se zařízením pracuje více uživatelů, měl by záznam obsahovat i údaje o jednotlivých uživateliích.

Přihlášení uživatele i identifikace tyčí může probíhat formou scanování jedinečných kódů RFID čtečkou. Čtečka by byla připojena USB portem a urychlila by přihlašovací proces. I když by se cena navýšila, čtečka by byla velmi užitečná.

O podrobnější snímání tyčí se může postarat například soustava založená na infračervených laserech. Ty by mohli vytvořit dokonce i 3D model snímaného objektu, avšak cena by se vyšplhala o řád nebo o dva výše. Zpracování naměřených dat by bylo také složitější a pomalejší, jak co se týče nastavení a naprogramování, tak vyhodnocení. Zařízení, které by sloužilo pro průmyslové účely, by mělo být rychlé. V laserovém provedení by se rychlost dala upravit s pomocí výkonějších přístrojů na zpracování dat, cena by ovšem opět stoupla.

Výstup vyhodnocení dat je další oblastí, kde je možný určitý pokrok. Mohly by například na jednom zařízení běžet dva servery, kde by výchozí port odkázal uživatele čistě na stránku přihlášení a zobrazení výsledků a jiný port by otevřel webové rozhraní LuCI pro další konfiguraci nebo diagnostiky. Stránka s výsledky může být doplněna o grafickou část, například o barevnou matici pro každou tyč, kde by zelená barva značila sepnutý indukční senzor a červená nesepnutý.

Pokud by vyvstal požadavek na zařazení celého systému do automatické linky, přístroj by po dodatečné úpravě obstál. Může nastat celková automatizace, při níž by musel být přidán třídící úsek pro tyče vhodné k použití a tyče poškozené, nebo částečná automatizace, při níž by třídění prováděl pracovník a automaticky by probíhalo pouze dodání skupiny tyčí a odvoz dvou roztříděných skupin. Zde by se musel přidat automatický dávkovač tyčí a také možnost pro pracovníka přivolat další skupinu a odeslat roztříděné tyče, například ve formě tlačítka. Více podrobností o zařazení do automatické linky nalezneme v užitém vzoru [4].

13. Závěr

V této práci byl navržen vyhodnocovací systém pro třídění kovových tyčí. V kapitole 2 byly analyzovány jednotlivé kroky k řešení zadání. Jako řídicí jednotka byl vybrán jednodeskový počítač Raspberry Pi 3, z důvodu zejména dobrého poměru ceny a výkonu. V kapitole 3 je platforma popsána a je otestováno rozhraní GPIO formou krátkého programu v jazyce Python. Všechny programy v této práci byly napsány pomocí tohoto jazyka. Důvody, proč byl vybrán, jsou čitelnost kódu a existence modulů k ovládání GPIO pinů. Moduly, které se objevují v této práci, jsou *gpiozero* a *RPi.GPIO*, z nichž byl vybrán druhý modul. V kapitole 4 je popsáno celkové propojení přístrojů v síti a vyobrazeno pomocí blokového schématu. Pokud by se webserver instaloval zvlášť, volbou by byl server Apache 2. Práce zahrnuje jeho instalaci na OS Raspbian. Následně bylo podrobněji prozkoumáno zabezpečení serveru pomocí OpenSSL, což bylo aplikováno v systému OpenWRT v kapitole 8.2.4. K detekování tyčí byl použit indukční snímač, jehož princip byl stručně popsán a bylo určeno jeho zapojení.

Vyhodnocovací proces a zobrazení výsledků by bylo možné sestavit na OS Raspbian z jednotlivých sekcí, avšak výhodnější byl přechod na OS OpenWRT, kde jsou některé části, jako například webserver nebo možnosti zabezpečení, už zakomponovány. Tato práce není konečná a celý systém lze dále rozvíjet, proto byla kompilace OS OpenWRT a následná konfigurace popsána detailně. V průběhu konfigurace vyvstaly různorodé problémy, například nedostatek místa na virtuálním disku v Oracle VirtualBox. V průběhu práce je na problémy poukázáno a je poskytnuto případné řešení. Modul *RPi.GPIO*, použitý v kapitole 7, nemohl být použit ve finální verzi programu z důvodu špatného instalačního souboru a v práci je tedy popsáno, jak s GPIO pracovat jiným způsobem.

V kapitole 9 je stručně popsán vyhodnocovací program a použité moduly. Okomentovaný zdrojový kód se nachází v příloze. Program neobsahuje možnost uložení dat. V další kapitole je popsána prezentace výsledků na webovou stránku, která je součástí uživatelského rozhraní LuCI. Výsledky jsou zobrazovány v textové formě a stránka se automaticky obnovuje v určitém časovém intervalu. Možnost ukládání dat a grafická prezentace výsledků jsou předmětem dalšího vývoje systému, který je popsán v předposlední kapitole.

Práce dále zkoumá některé bezpečnostní prvky OS OpenWRT, stabilitu a životnost komponent a celkové náklady. Navržený systém je považován za nízkonákladový.

Seznam použité literatury

- [1] TYLKO, John. MIT and navigating the path to the moon. *AeroAstro* [online]. Massachusetts Institute of Technology, 2009, **2008-2009**(6), 84 [cit. 2018-05-14]. Dostupné z: http://issuu.com/mitaeroastro/docs/aeroastro_no6?mode=window&pageNumber=1
- [2] Automation. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2018-05-11]. Dostupné z: <https://en.wikipedia.org/wiki/Automation>
- [3] Smart Meter Plug. *Revogi* [online]. Los Altos, California, USA: Revogi, LLC, 2017 [cit. 2018-05-09]. Dostupné z: <http://www.revogi.com/smart-power/smart-meter-plug/#section0>
- [4] *Zařízení pro kontrolu přesnosti tyčového materiálu*. 2017. Česká Republika. CZ 306949 B6 Užitiný vzor. Uděleno 23.08.2017. Zapsáno 04.10.2017. Dostupné také z: <http://isdv.upv.cz/doc/FullFiles/Patents/FullDocuments/306/306949.pdf>
- [5] DragonBoard 820c Development Board. *Qualcomm* [online]. San Diego, CA, USA: Qualcomm Technologies, Inc., 2018 [cit. 2018-05-16]. Dostupné z: <https://developer.qualcomm.com/hardware/dragonboard-820c>
- [6] RASPBERRY PI ZERO. *Raspberry Pi* [online]. Cambridge, UK: Raspberry Pi Foundation, 2018 [cit. 2018-05-16]. Dostupné z: <https://www.raspberrypi.org/products/raspberry-pi-zero/>
- [7] *Orange Pi* [online]. Shenzhen City, China: Xunlong Software CO., 2016 [cit. 2018-05-16]. Dostupné z: <http://www.orangepi.org/index.html>
- [8] Raspberry Pi. *Wikipedia* [online]. San Francisco: Wikimedia Foundation, 2018 [cit. 2018-02-12]. Dostupné z: https://cs.wikipedia.org/wiki/Raspberry_Pi
- [9] Raspberry Pi 3 - Model B. In: *ModMyPi* [online]. Tunbridge Wells, UK: ModMyPi LTD., 2012 [cit. 2018-05-14]. Dostupné z: <https://www.modmypi.com/pis-and-peripherals-1139/raspberry-pi-3-model-b>
- [10] GPIO: MODELS A+, B+, RASPBERRY PI 2 B AND RASPBERRY PI 3 B. *Raspberry Pi* [online]. Cambridge, UK: Raspberry Pi Foundation, 2018 [cit. 2018-02-12]. Dostupné z: <https://www.raspberrypi.org/documentation/usage/gpio-plus-and-raspi2/>
- [11] GPIO. *Raspberry Pi* [online]. Cambridge, UK: Raspberry Pi Foundation, 2018 [cit. 2018-02-12]. Dostupné z: <https://www.raspberrypi.org/documentation/hardware/raspberrypi/gpio/README.md>
- [12] Raspberry Pi vs Arduino: GPIO Interface. In: *SmoIex* [online]. Dongguan, China: SCNDAR Electronic Co.,Ltd., 2017 [cit. 2018-05-14]. Dostupné z: <https://www.ismolex.com/raspberry-pi-vs-arduino/>
- [13] Raspberry Pi marks 2nd birthday with plan for open source graphics driver. *Ars Technica* [online]. Web: Condé Nast Inc., 2001 [cit. 2018-05-09]. Dostupné z:

<https://arstechnica.com/information-technology/2014/02/raspberry-pi-marks-2nd-birthday-with-plan-for-open-source-graphics-driver/>

- [14] Raspberry Pi 8GB Preloaded (NOOBS) SD Card. *Amazon* [online]. Seattle, WA: Amazon.com Inc., 2018 [cit. 2018-02-12]. Dostupné z: <https://www.amazon.com/Raspberry-8GB-Preloaded-NOOBS-Card/dp/B00GFA48N4>
- [15] DOWNLOADS. *Raspberry Pi* [online]. Cambridge, UK: Raspberry Pi Foundation, 2018 [cit. 2018-02-12]. Dostupné z: <https://www.raspberrypi.org/downloads/>
- [16] NOOBS SETUP. *Raspberry Pi* [online]. Cambridge, UK: Raspberry Pi Foundation, 2018 [cit. 2018-02-12]. Dostupné z: <https://www.raspberrypi.org/help/noobs-setup/2/>
- [17] NOOBS. *Raspberry Pi* [online]. Cambridge, UK: Raspberry Pi Foundation, 2018 [cit. 2018-02-12]. Dostupné z: <https://www.raspberrypi.org/documentation/installation/noobs.md>
- [18] Turning on an LED with your Raspberry Pi's GPIO Pins. *The Pi Hut* [online]. London, UK: Mann Enterprises LTD, 2018 [cit. 2018-02-12]. Dostupné z: <https://thepihut.com/blogs/raspberry-pi-tutorials/27968772-turning-on-an-led-with-your-raspberry-pis-gpio-pins>
- [19] NUTTALL, Ben a Dave JONES. Gpiozero. *Gpiozero* [online]. Web: Raspberry Pi Foundation, 2015 [cit. 2018-05-07]. Dostupné z: <https://gpiozero.readthedocs.io/en/stable/>
- [20] Python (programming language). In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2018-02-12]. Dostupné z: [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))
- [21] RPi.GPIO module basics. *Source Forge: raspberry-gpio-python* [online]. La Jolla, CA: Slashdot Media, 2018 [cit. 2018-02-12]. Dostupné z: <https://sourceforge.net/p/raspberry-gpio-python/wiki/BasicUsage/>
- [22] Compare and contrast Python GPIO APIs. *StackExchange: Serverfault* [online]. London: Stack Exchange, 2008 [cit. 2018-05-07]. Dostupné z: <https://raspberrypi.stackexchange.com/questions/58820/compare-and-contrast-python-gpio-apis>
- [23] Webový server. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2018-02-12]. Dostupné z: https://cs.wikipedia.org/wiki/Webov%C3%BD_server
- [24] SETTING UP AN APACHE WEB SERVER ON A RASPBERRY PI. *Raspberry Pi* [online]. Cambridge, UK: Raspberry Pi Foundation, 2018 [cit. 2018-02-12]. Dostupné z: <https://www.raspberrypi.org/documentation/remote-access/web-server/apache.md>
- [25] *OpenSSL* [online]. online: OpenSSL Software Foundation, 1991 [cit. 2018-02-12]. Dostupné z: <https://www.openssl.org/>
- [26] *OpenSSL Cookbook* [online]. 2. London: Feisty Duck, 2016 [cit. 2018-02-12]. ISBN 978-1907117053. Dostupné z: <https://www.feistyduck.com/library/openssl-cookbook/online/index.html>

- [27] Bezkontaktní indukční snímače přiblížení. *Automatizace.hw* [online]. Praha: HW server, 1997-2014 [cit. 2018-02-12]. Dostupné z: <https://automatizace.hw.cz/komponenty-mereni-a-regulace/indukcni-snimace-priblizeni-obecny-popis.html>
- [28] OpenWRT Project: About. *OpenWRT* [online]. Web: OpenWRT, 2018 [cit. 2018-05-02]. Dostupné z: <https://openwrt.org/about>
- [29] EROSHKIN, Ivan. *Řízení technologických procesů v koncepci IoT*. Praha, 2017. Bakalářská práce. České vysoké učení technické. Vedoucí práce Lukáš Vojtěch.
- [30] OpenWRT. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2018-05-02]. Dostupné z: <https://cs.wikipedia.org/wiki/OpenWrt>
- [31] LEDE. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2018-05-02]. Dostupné z: <https://en.wikipedia.org/wiki/LEDE>
- [32] OpenWRT Project: Table of Hardware. *OpenWRT* [online]. Web: OpenWRT, 2018 [cit. 2018-05-02]. Dostupné z: <https://openwrt.org/toh/start>
- [33] Installing LEDE Development Snapshots. *OpenWRT* [online]. Web: OpenWRT, 2018 [cit. 2018-05-09]. Dostupné z: <https://openwrt.org/docs/guide-quick-start/developmentinstallation>
- [34] Firewall. *OpenWRT* [online]. Web: OpenWRT, 2018 [cit. 2018-05-02]. Dostupné z: https://openwrt.org/docs/guide-user/firewall/firewall_configuration
- [35] Traffic Shaping. *OpenWRT* [online]. Web: OpenWRT, 2018 [cit. 2018-05-02]. Dostupné z: <https://openwrt.org/docs/guide-user/network/traffic-shaping/start>
- [36] ANDERSEN, Erik. BusyBox | About. *BusyBox* [online]. Web: BusyBox, 2018 [cit. 2018-05-02]. Dostupné z: <https://busybox.net/about.html>
- [37] OPKG Package Manager. *OpenWRT* [online]. Web: OpenWRT, 2018 [cit. 2018-05-02]. Dostupné z: <https://wiki.openwrt.org/doc/techref/opkg>
- [38] DNS and DHCP Configuration. *OpenWRT* [online]. Web: OpenWRT, 2018 [cit. 2018-05-02]. Dostupné z: <https://wiki.openwrt.org/doc/uci/dhcp>
- [39] LuCI - Technical Reference. *OpenWRT* [online]. Web: OpenWRT, 2018 [cit. 2018-05-02]. Dostupné z: <https://wiki.openwrt.org/doc/techref/luci>
- [40] OAKLEY, Andrew. Raspbian x86 on VirtualBox on a Windows PC. *Aoakley.com* [online]. Gloucestershire UK: Andrew Oakley, 2017 [cit. 2018-05-02]. Dostupné z: <http://www.aoakley.com/articles/2017-07-04-raspbian-x86-virtualbox.php>
- [41] MOLLOY, Derek. How to resize a VirtualBox guest Linux VDI Disk under Windows Host. *Derekmolloy.ie* [online]. Dublin: Derek Molloy, 2018 [cit. 2018-05-02]. Dostupné z: <http://derekmolloy.ie/resize-a-virtualbox-disk/>
- [42] OpenWRT build system - Installation. *OpenWRT* [online]. Web: OpenWRT, 2018 [cit. 2018-05-02]. Dostupné z: <https://wiki.openwrt.org/doc/howto/buildroot.exigence>

- [43] OpenWRT Feeds. *OpenWRT* [online]. Web: OpenWRT, 2018 [cit. 2018-05-02]. Dostupné z: <https://wiki.openwrt.org/doc/devel/feeds>
- [44] OpenWRT build system - Usage. *OpenWRT* [online]. Web: OpenWRT, 2018 [cit. 2018-05-02]. Dostupné z: <https://wiki.openwrt.org/doc/howto/build>
- [45] How to mount a VirtualBox shared folder. *StackExchange: Serverfault* [online]. London, UK: Stack Exchange Inc., 2008 [cit. 2018-05-02]. Dostupné z: <https://serverfault.com/questions/674974/how-to-mount-a-virtualbox-shared-folder>
- [46] *Etcher.io* [online]. Seattle, USA: Resin.io, 2013 [cit. 2018-05-07]. Dostupné z: <https://etcher.io/>
- [47] Network configuration. *Openwrt* [online]. Web: OpenWRT, 2018 [cit. 2018-05-08]. Dostupné z: <https://wiki.openwrt.org/doc/uci/network>
- [48] SSH Access. *OpenWRT* [online]. Web: OpenWRT, 2018 [cit. 2018-05-08]. Dostupné z: <https://openwrt.org/docs/guide-quick-start/sshadministration>
- [49] LuCI Certificate. *OpenWRT* [online]. Web: OpenWRT, 2018 [cit. 2018-05-08]. Dostupné z: <https://openwrt.org/docs/guide-user/luci/getting-rid-of-luci-https-certificate-warnings>
- [50] Device tree. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2018-05-11]. Dostupné z: https://en.wikipedia.org/wiki/Device_tree
- [51] Definitions for lirc-rpi module. *GitHub* [online]. San Francisco: GitHub, Inc., 2007 [cit. 2018-05-08]. Dostupné z: <https://github.com/krzk/tizen-tv-rpi-linux/blob/master/arch/arm/boot/dts/overlays/lirc-rpi-overlay.dts>
- [52] Device Tree: Readme. *GitHub* [online]. San Francisco: GitHub, Inc., 2007 [cit. 2018-05-08]. Dostupné z: <https://github.com/raspberrypi/linux/blob/rpi-4.1.y/arch/arm/boot/dts/overlays/README>
- [53] GPIO Sysfs Interface for Userspace. *Linux Kernel Archives* [online]. San Francisco: Linux Kernel Organization, Inc., 2017 [cit. 2018-05-08]. Dostupné z: <https://www.kernel.org/doc/Documentation/gpio/sysfs.txt>
- [54] Miscellaneous operating system interfaces. *Python 2.7.15 documentation* [online]. Wilmington, Delaware, USA: Python Software Foundation, 2010 [cit. 2018-05-08]. Dostupné z: <https://docs.python.org/2/library/os.html>
- [55] SUMMERFIELD, Mark. *Python 3: výukový kurz*. Vyd. 1. Brno: Computer Press, 2010. ISBN 978-80-251-2737-7.
- [56] A Beginner's Python Tutorial. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2018-05-09]. Dostupné z: https://en.wikibooks.org/wiki/A_Beginner%27s_Python_Tutorial
- [57] The Complete Python Course: Beginner to Advanced!. *StackSkills* [online]. Venice, CA: StackCommerce, 2016 [cit. 2018-02-15]. Dostupné z: <https://stackskills.com/p/python-complete>

- [58] Adding new elements to LuCI. *OpenWRT* [online]. Web: OpenWRT, 2018 [cit. 2018-05-08]. Dostupné z: <https://openwrt.org/docs/guide-developer/luci>
- [59] Window setTimeout() Method. *W3Schools* [online]. Sandnes, Norsko: Refsnes Data, 1999 [cit. 2018-05-08]. Dostupné z: https://www.w3schools.com/jsref/met_win_settimeout.asp
- [60] RUBINKSKI, David. Life expectancy of an Inductive sensor. *Pepperl+Fuchs* [online]. Mannheim, Germany: Pepperl+Fuchs, 2018 [cit. 2018-05-11]. Dostupné z: <https://blog.pepperl-fuchs.us/blog/bid/352679/Life-Expectancy-of-an-Inductive-Sensor>
- [61] Raspberry Pi lifespan. *Raspberry Pi* [online]. Cambridge, UK: Raspberry Pi Foundation, 2018 [cit. 2018-05-11]. Dostupné z: <https://www.raspberrypi.org/forums/viewtopic.php?t=156347>
- [62] What's the life expectancy of an SD card?. *StackExchange: Superuser* [online]. London, UK: Stack Exchange, 2008 [cit. 2018-05-11]. Dostupné z: <https://superuser.com/questions/17350/whats-the-life-expectancy-of-an-sd-card>
- [63] How big an RSA key is considered secure today?. *StackExchange: Cryptography* [online]. London, UK: Stack Exchange, 2008 [cit. 2018-05-11]. Dostupné z: <https://crypto.stackexchange.com/questions/1978/how-big-an-rsa-key-is-considered-secure-today>
- [64] LEDE security features. *OpenWRT* [online]. Web: OpenWRT, 2018 [cit. 2018-05-11]. Dostupné z: <https://openwrt.org/docs/guide-user/security/security-features>
- [65] *RPishop* [online]. Česká Republika: RPishop.cz, 2018 [cit. 2018-05-11]. Dostupné z: <http://rpishop.cz/>
- [66] Carlo Gavazzi Type EI Proximity Sensors. *Instrumart* [online]. Burlington, VT, USA: Total Temperature Instrumentation, Inc., 2018 [cit. 2018-05-11]. Dostupné z: <https://www.instrumart.com/products/18012/carlo-gavazzi-type-ei-proximity-sensors>
- [67] *Draw.io* [online]. Northampton, UK: JGraph Ltd., c2016-2018 [cit. 2018-05-18]. Dostupné z: www.draw.io
- [68] GPIO Electrical Specifications. *Mosaic Documentation Web* [online]. Newark, Ca, USA: Mosaic Industries, Inc., c2005-2011 [cit. 2018-05-11]. Dostupné z: <http://www.mosaic-industries.com/embedded-systems/microcontroller-projects/raspberry-pi/gpio-pin-electrical-specifications>

Seznam zkratek

Zkratka	Celý výraz
IoT	Internet of Things
PC	Personal Computer
MIT	Massachusetts Institute of Technology
OS	Operating System
RISC	Reduced Instruction Set Computing
CPU	Central Processing Unit
SDRAM	Synchronous Dynamic Random Access Memory
GPU	Graphic Processing Unit
USB	Universal Serial Bus
CSI	Camera Serial Interface
HDMI	High-Definition Multimedia Interface
DSI	Display Serial Interface
I ² S	Inter-IC Sound
SD	Secure Digital
GPIO	General Purpose Input Output
GND	Ground
EEPROM	Electrically Erasable Programmable Read-Only Memory
BSD	Berkeley Software Distribution
SSD	Solid-State Drive
NOOBS	New Out Of the Box Software
LED	Light-Emitting Diode
HTTP	Hypertext Transfer Protocol
URL	Uniform Resource Locator
HTML	HyperText Markup Language
TLS/SSL	Transport Layer Security / Secure Sockets Layer
RSA	Rivest, Shamir, Adleman (Iniciály autorů)
DSA	Digital Signature Algorithm
ECDSA	Elliptic Curve Digital Signature Algorithm
PEM	Privacy-enhanced Electronic Mail

CA	Certificate Authority
CSV	Comma Separated Values
LEDE	Linux Embedded Development Environment
DHCP	Dynamic Host Configuration Protocol
LuCI	Lua Configuration Interface
PAE/NX	Physical Address Extension/Non eXecute
IP	Internet Protocol
SSH	Secured Shell
RELRO	Relocation Read Only
ELF	Executable and Linkable Format
RFID	Radio-frequency identification

Seznam použitých obrázků

Obrázek 1.: Raspberry Pi 3 Model B [9].....	9
Obrázek 2.: Schéma hardwarových součástí [8]	10
Obrázek 3.: Číslování GPIO a legenda [11]	11
Obrázek 4.: GPIO [12]	11
Obrázek 5.: Schéma zapojení s LED diodou [19]	13
Obrázek 6.: Rozložení v síti	15
Obrázek 7.: Defaultní obsah stránky	16
Obrázek 8.: Součásti snímače [27].....	19
Obrázek 9.: Specifikace snímače [více viz Příloha]	19
Obrázek 10.: Schéma zapojení	20
Obrázek 11.: Testování programu	22
Obrázek 12.: VirtualBox se třemi virtuálními stroji, mezi nimi i GParted	25
Obrázek 13.: Virtuální prostředí OS Raspbian s nastavením sdílené schránky	26
Obrázek 14.: Konfigurace OpenWRT před kompilací	27
Obrázek 15.: Program Etcher.io	28
Obrázek 16.: Úvodní stránka LuCI	29
Obrázek 17.: Program WinSCP se spuštěným terminálem	30
Obrázek 18.: Rozhraní LuCI se zabezpečeným připojením.....	32
Obrázek 19.: Příklad jednoho cyklu měření	34
Obrázek 20.: Stránka s výsledky	35

Všechny použité obrázky jsou převážně vytvořeny autorem přímo pro tuto práci, některé byly poskytnuty nadací Raspberry Pi Foundation a ostatní jsou volně přístupné na internetu. Schémata byla vytvořena pomocí webové stránky draw.io [67].

Seznam tabulek

Tabulka 1.: Ekonomické zhodnocení	37
---	----

Přílohy

Příloha 1 Datasheet k indukčnímu senzoru

Proximity Sensors Inductive Stainless Steel Housing Types EI, Ø 4, 5 x 5 mm



- Miniature stainless steel housing Ø 4, aluminium 5 x 5 mm
- Sensing distance: 1 mm
- Power supply: 10 to 30 VDC
- Output: Transistor NPN/PNP, make or break switching
- Protection: Short-circuit, reverse polarity and overload
- LED-indication for output ON
- 2 m PUR cable or M8 plug

Product Description

Miniature inductive proximity sensor available as Ø 4 and 5 x 5 mm types. Output configuration for NPN/PNP with both NO and NC types as standard. Housing materials

are stainless steel for Ø 4 and aluminium for the 5 x 5 mm type. Cabled models utilise 2 meters of PUR cable and plug versions features a bright ring LED.

Ordering Key

EI 0401 PPO SS-5

Type _____
 Housing diameter (mm) _____
 Rated operating dist. (mm) _____
 Output type _____
 Housing material _____
 Housing type _____
 Connection type _____

Type Selection

Housing diameter	Rated operating dist. (S _r)	Connection type	Ordering no. Transistor NPN Make switching	Ordering no. Transistor NPN Break switching	Ordering no. Transistor PNP Make switching	Ordering no. Transistor PNP Break switching
Ø 4 mm	0.8 mm ¹⁾	Cable	EI 0401 NPOSS	EI 0401 NPCSS	EI 0401 PPOSS	EI 0401 PPCSS
Ø 4 mm	0.8 mm ¹⁾	Plug	EI 0401 NPOSS-5	EI 0401 NPCSS-5	EI 0401 PPOSS-5	EI 0401 PPCSS-5
5 x 5 mm	0.8 mm ¹⁾	Cable	EI X501 NPOSS	EI X501 NPCSS	EI X501 PPOSS	EI X501 PPCSS
5 x 5 mm	0.8 mm ¹⁾	Plug	EI X501 NPOSS-5	EI X501 NPCSS-5	EI X501 PPOSS-5	EI X501 PPCSS-5

¹⁾ For flush mounting in metal

Specifications

Rated operational volt. (U_B)	10 to 30 VDC (ripple included)	Usable operating dist. (S_r)	0.85 x S _r ≤ S _r ≤ 1.15 x S _r
Ripple	≤ 10%	Ambient temperature	
Rated operational current (I_B)		Operating	-25° to +70°C (-13° to +158°F)
Continuous	≤ 150 mA @ +25°C (+75°F)	Storage	-30° to +80°C (-22° to +176°F)
No-load supply current (I₀)	≤ 2.1 mA OFF: 1.5 mA ON: 10 mA	Degree of protection	Plug versions: IP 65 Cable versions: IP 67 (Nema 1, 3, 4, 6, 13)
Voltage drop (U_d)	< 1 V	Connection	
Protection	Reverse polarity, short-circuit, overload	Cable	2 m, 3 x 0.14 mm ² grey PUR, oil proof M8, CONH5A-xxx Serie
Transient voltage	≤ 1 kV/0.5 J (prepared)	Plug	
Power ON delay	Ca. 500 μs	Housing material	
Frequency of operating cycles (f)	1000 Hz	Body	Stainless steel (1.4301)
Indication for output ON	LED, yellow	Ø 4	Aluminium
Assured operating dist. (S_a)	0 to 0.52 mm	5 x 5 mm	Plastic
Repeat accuracy (R)	± 0.01 mm	Cap	
Hysteresis (H)		Weight	
(Differential travel)	3 to 15% of sensing distance	Cable versions: 25 g	
Effective operating dist. (S_e)	0.9 x S _r ≤ S _e ≤ 1.1 x S _r	Plug versions: 4 g	
		Tightening torque	1 Nm
		CE-marking	Yes

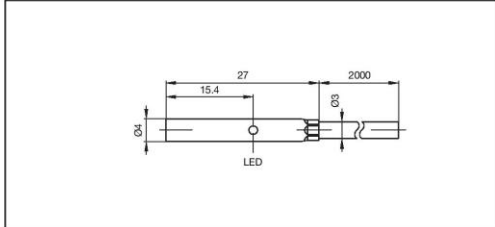
Specifications are subject to change without notice (20.08.01)

1

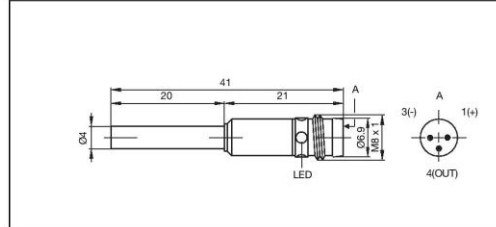
EI, Ø 4, 5 x 5 mm



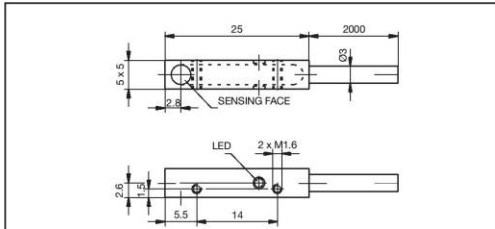
Dimensions



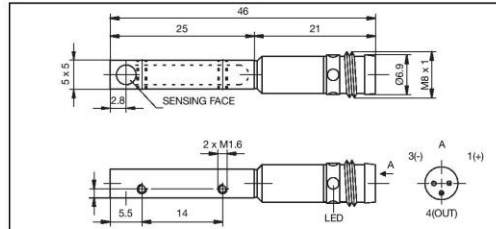
EI 0401 xPxSS



EI 0401 xPxSS-5

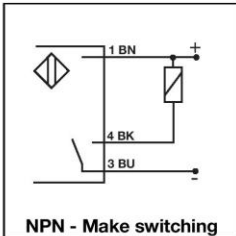


EI X501 xPxSS

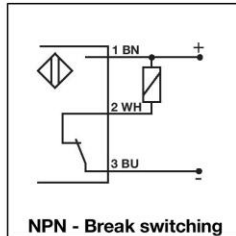


EI X501 xPxSS-5

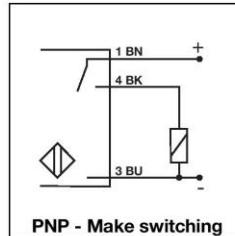
Wiring Diagrams



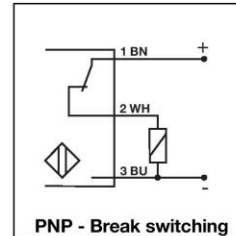
NPN - Make switching



NPN - Break switching

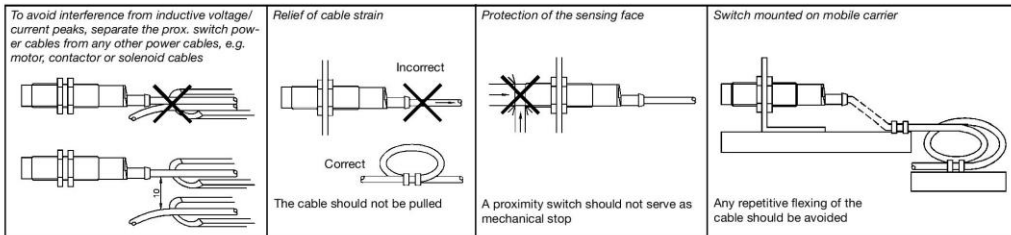


PNP - Make switching



PNP - Break switching

Installation Hints



Power Supplies

Power supplies VDC: > SS 130/140.
 Power supplies with amplifier relays: > SV 190.

Příloha 2 Hlavní program na vyhodnocení dat:

(Některá odřádkování byla přidána pro lepší čitelnost)

```
import os
import time          #importování modulů
print("Welcome to the program - start the measurment by putting the iron
rod on the stool.") #přivítání uživatele
check = 1
html_string_start = ""<%+header%>
<h1><%:Results Page%></h1>
<p>Thank you for using our measurement tool.</p>
<p> </p>
<p>The result of the most recent measurement is: ""
html_string_line_end = ""</p>
""
html_string_middle = ""
<p> </p>
<p>That means the rod is ""
html_string_end = ""
<p> </p>
<p> </p>
<p>This tool is powered by the Department of telecommunication engineering,
Faculty of electrical engineering, CTU in Prague, 2018 </p>
<script>setTimeout(location.reload.bind(location), 6000);</script>
<%+footer%>
""
#inicializace proměnných typu string ve formě html
os.chdir('.')          #podobný příkaz jako „cd ..“
htmlFPath = os.path.join('usr', 'lib', 'lua', 'luci', 'view', 'luci-
indsensor', "index.htm")
fGPIO2 = os.path.join('sys', 'devices', 'platform', 'soc', '3f200000.gpio',
'gpiochip0', 'gpio', 'gpio2', "value")
fGPIO3 = os.path.join('sys', 'devices', 'platform', 'soc', '3f200000.gpio',
'gpiochip0', 'gpio', 'gpio3', "value")
fGPIO4 = os.path.join('sys', 'devices', 'platform', 'soc', '3f200000.gpio',
'gpiochip0', 'gpio', 'gpio4', "value")
fGPIO5 = os.path.join('sys', 'devices', 'platform', 'soc', '3f200000.gpio',
'gpiochip0', 'gpio', 'gpio5', "value")
fGPIO7 = os.path.join('sys', 'devices', 'platform', 'soc', '3f200000.gpio',
'gpiochip0', 'gpio', 'gpio7', "value")
fGPIO8 = os.path.join('sys', 'devices', 'platform', 'soc', '3f200000.gpio',
'gpiochip0', 'gpio', 'gpio8', "value")
```

```

fGPIO9 = os.path.join('sys', 'devices', 'platform', 'soc', '3f200000.gpio',
'gpiochip0', 'gpio', 'gpio9', "value")
fGPIO10 = os.path.join('sys', 'devices', 'platform', 'soc',
'3f200000.gpio', 'gpiochip0', 'gpio', 'gpio10', "value")
fGPIO11 = os.path.join('sys', 'devices', 'platform', 'soc',
'3f200000.gpio', 'gpiochip0', 'gpio', 'gpio11', "value")
fGPIO14 = os.path.join('sys', 'devices', 'platform', 'soc',
'3f200000.gpio', 'gpiochip0', 'gpio', 'gpio14', "value")
fGPIO15 = os.path.join('sys', 'devices', 'platform', 'soc',
'3f200000.gpio', 'gpiochip0', 'gpio', 'gpio15', "value")
fGPIO17 = os.path.join('sys', 'devices', 'platform', 'soc',
'3f200000.gpio', 'gpiochip0', 'gpio', 'gpio17', "value")
fGPIO18 = os.path.join('sys', 'devices', 'platform', 'soc',
'3f200000.gpio', 'gpiochip0', 'gpio', 'gpio18', "value")
fGPIO22 = os.path.join('sys', 'devices', 'platform', 'soc',
'3f200000.gpio', 'gpiochip0', 'gpio', 'gpio22', "value")
fGPIO23 = os.path.join('sys', 'devices', 'platform', 'soc',
'3f200000.gpio', 'gpiochip0', 'gpio', 'gpio23', "value")
fGPIO24 = os.path.join('sys', 'devices', 'platform', 'soc',
'3f200000.gpio', 'gpiochip0', 'gpio', 'gpio24', "value")
fGPIO25 = os.path.join('sys', 'devices', 'platform', 'soc',
'3f200000.gpio', 'gpiochip0', 'gpio', 'gpio25', "value")
fGPIO27 = os.path.join('sys', 'devices', 'platform', 'soc',
'3f200000.gpio', 'gpiochip0', 'gpio', 'gpio27', "value")
        # cesty k jednotlivým hodnotám GPIO pinů
def GPIOread(fpath):          #metoda na čtení ze souboru
    f = open(fpath, "r")
    v = f.read(1) #přečíst pouze první char
    f.close()
    return v
x = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] #pole pro výsledky
while check == 1:
    if GPIOread(fGPIO2) == "1":          #kontrola pozice tyče, začátek měření
        rov = 1
        html_string_result = " "
        start = time.time()
        print("READY")
        while time.time() < start + 60:
            if GPIOread(fGPIO3) == "1":
                x[0] = 1
            if GPIOread(fGPIO4) == "1":

```

```

        x[1] = 1
    if GPIOread(fGPIO7) == "1":
        x[2] = 1
    if GPIOread(fGPIO8) == "1":           #pokud snímač sepne
        x[3] = 1           #do příslušné pozice v poli se napíše 1
    if GPIOread(fGPIO9) == "1":
        x[4] = 1
    if GPIOread(fGPIO10) == "1":
        x[5] = 1
    if GPIOread(fGPIO11) == "1":
        x[6] = 1
    if GPIOread(fGPIO14) == "1":
        x[7] = 1
    if GPIOread(fGPIO15) == "1":
        x[8] = 1
    if GPIOread(fGPIO17) == "1":
        x[9] = 1
    if GPIOread(fGPIO18) == "1":
        x[10] = 1
    if GPIOread(fGPIO22) == "1":
        x[11] = 1
    if GPIOread(fGPIO23) == "1":
        x[12] = 1
    if GPIOread(fGPIO24) == "1":
        x[13] = 1
    if GPIOread(fGPIO25) == "1":
        x[14] = 1
    if GPIOread(fGPIO27) == "1":
        x[15] = 1

    if GPIOread(fGPIO5) == "1": break           #GPIO5 - konec měření
print(" ")
print("DONE.")           #2 kontroly konce - sepnutí GPIO5 a časová
print(x)
print(" ")
for i in range(0, 15):
    if x[i] == 0:
        rov = 0           #kontrola buněk v poli
if rov == 1 :
    print("Iron rod is flat.")           #vyhodnocení, zda je tyč rovná
    html_string_result = "FLAT."
if rov == 0 :

```

```

        print("Iron rod is not flat.") #nebo není rovná
        html_string_result = "NOT FLAT."
        html_string_final = html_string_start + str(x) +
html_string_line_end + html_string_middle + html_string_result +
html_string_line_end + html_string_end #kompletování formy HTML
        htmlFile = open(htmlFPath,"w+")
        htmlFile.write(html_string_final) #zapsání do souboru
        htmlFile.close()
        print(" ")
        print("The program has updated the webpage.")

pop = raw_input("Another measurment? (Confirm with enter) y/n: ")
if pop == "y" : check = 1          #nabídka dalšího měření
if pop == "n" : check = 0

```