

Diplomová práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra měření

Elektronické kukátko

Bc. Josef Kiefmann

Vedoucí diplomové práce: prof. Ing. Miroslav Husák CSc.
Studijní program: Inteligentní budovy
Leden 2019

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Kiefmann** Jméno: **Josef** Osobní číslo: **406372**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra měření**
Studijní program: **Inteligentní budovy**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Elektronické kukátko

Název diplomové práce anglicky:

Electronic Viewer

Pokyny pro vypracování:

Navrhněte hardware a software řešení elektronického kukátko na platformě Raspberry PI s využitím dalších podpůrných obvodů a modulů. V práci se zaměřte na návrh softwarového vybavení pro ukládání obrazových dat a možnost jejich vzdálené správy, (webová aplikace pro vzdálenou správu zařízení a získání dat z přístroje a lokální aplikace na zařízení pro jeho ovládní), dále využití knihovny pro detekci pohybu. Realizujte praktickou implementaci akustické detekce zvuku zvonku pomocí číslicového zpracování signálu, od vzorce, přes ukázky v simulačním software, až po výsledný kód v některém z programovacích jazyků a reálnou aplikaci na zařízení.

Seznam doporučené literatury:

- [1] Jan Vobecký, Vít Záhlava: Elektronika - Součástky a obvody, principy a příklady. Grada Publishing, 2006. ISBN 80-247-1241-5
- [2] Deborah S. Ray, Eric J. Ray: Unix - podrobný průvodce. Grada Publishing, 2009. ISBN 978-80-247-2125-5
- [3] Jan Uhlíř, Pavel Sovka: Číslicové Zpracování Signálů. Vydavatelství ČVUT, 2002. ISBN 80-01-02613-2

Jméno a pracoviště vedoucí(ho) diplomové práce:

prof. Ing. Miroslav Husák, CSc., katedra mikroelektroniky FEL

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

Datum zadání diplomové práce: **14.02.2018**

Termín odevzdání diplomové práce: **08.01.2019**

Platnost zadání diplomové práce:

do konce zimního semestru 2019/2020

prof. Ing. Miroslav Husák, CSc.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

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

Datum převzetí zadání

Podpis studenta

Poděkování

Rád bych poděkoval vedoucímu své diplomové práce, panu prof. Ing. Miroslavu Husákovi CSc., za zajímavé téma diplomové práce, věcné připomínky k vypracování a vstřícnost. Poděkování patří také mým rodičům, neboť mne během celého studia podporovali.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 5. ledna 2019

Abstrakt

Cílem této diplomové práce je navrhnout laboratorní funkční vzorek zařízení nazývaného elektronické kukátko z hlediska hardware a software na platformě Raspberry PI s využitím dalších podpůrných obvodů a modulů.

Klíčová slova: Raspberry PI; PHP; webový framework; Nette; Linux; Bash; databáze; FFT; programovací jazyk C

Vedoucí diplomové práce: prof. Ing. Miroslav Husák CSc.

Abstract

The goal of this diploma thesis is design functional laboratory sample of electronic viewer device composed of hardware and software based on Raspberry PI platform with using others additional modules and circuits.

Keywords: Raspberry PI; PHP; web framework; Nette; Linux; Bash; database; FFT; programming language C

Title translation: Electronic viewer

Obsah

Seznam použitých zkratk a symbolů	1	
1 Úvod	5	
2 Rozbor současného stavu	7	
2.1 Zařízení dostupná na trhu	7	
2.2 Požadavky na zařízení a případy užití	9	
3 Návrh a realizace	13	
3.1 Návrh hardware	13	
3.1.1 Kamera	14	
3.1.2 Displej	15	
3.1.3 Zvuková karta	16	
3.1.4 Wi-Fi adaptér	17	
3.1.5 Modul vlastní konstrukce	18	
3.2 Číslicové zpracování zvuku	28	
3.2.1 Analýza v časové oblasti	29	
3.2.2 Laděné filtry	30	
3.2.3 Korelace	31	
3.2.4 Fourierova transformace	31	
3.2.5 Okénkové funkce	42	
3.2.6 Zvukový formát WAV	43	
3.3 Číslicové zpracování obrazu	45	
3.3.1 Obrazový formát BMP	46	
3.4 Návrh software	49	
3.4.1 Operační systém	49	
3.4.2 Software pro správu pořízených dat	50	
3.4.3 Software Motion	52	
3.4.4 Software pro detekci podobnosti zvuků a detekci zvonkového napětí	53	
3.5 Instalace	56	
3.5.1 Instalace systému Raspbian	57	
3.5.2 Povolení SSH	57	
3.5.3 Nastavení prováděné v aplikaci Raspi-config	58	
3.5.4 Motion	58	
3.5.5 Webový server	59	
3.5.6 Skriptovací jazyk PHP	60	
3.5.7 Databázový server	61	
3.5.8 phpMyAdmin	61	
3.5.9 I2CTools	61	
3.5.10 Nastavení spouštění aplikací po startu systému	62	
3.5.11 Spouštění lokální aplikace	62	
3.5.12 Přístupový bod Wi-Fi	63	
3.5.13 Knihovny nutné pro překlad zdrojových kódů	64	
3.5.14 Watchdog	65	
4 Dosažené výsledky	67	
5 Závěr	69	
Literatura	71	
A Obsah přiloženého DVD	73	
B Schéma	75	
C Zdrojový kód funkcí v jazyce C pro výpočet FFT	79	

Obrázky

2.1 Diagram znázorňující možné případy užití systému	12
3.1 Schéma vstupního obvodu pro připojení zvonkového napětí	19
3.2 Způsob číslování pinů GPIO na počítači Raspberry PI	20
3.3 Schéma předzesilovače elektretového mikrofону	23
3.4 Blokové zapojení	28
3.5 FFT DIT nad 8 prvky vstupní posloupnosti	33
3.6 FFT DIF nad 8 prvky vstupní posloupnosti	33
3.7 Testovací signál v časové oblasti	36
3.8 Amplitudové spektrum testovaného signálu	36
3.9 Amplitudové spektrum po aplikaci funkce mag2db	37
3.10 Amplitudové spektrum se vztažnou úrovní maxima spektra	38
3.11 Detail prvních 20 prvků amplitudového spektra	39
3.12 Detail posledních 20 prvků amplitudového spektra	40
3.13 Fázové spektrum	40
3.14 Amplitudové spektrum získané z osciloskopu Rigol MSO1104Z-S	41
3.15 Amplitudové spektrum s aplikovanou okénkovou funkcí Blackman	43
3.16 Diagram doménového modelu	51
3.17 Vývojový diagram programu pro vytváření událostí na základě akustické detekce zvonku a zvonkového tlačítka	54
3.18 Vývojový diagram vlákna pro detekci akustické podobnosti analyzovaného signálu vůči signálu vzorovému	55
3.19 Diagram nasazení znázorňující hardware i software	56
B.1 Zapojení modulu na konektor GPIO a připojení časovače DS1307	75
B.2 Tvorba galvanicky odděleného napájení pro vstupní obvod pro připojení zvonkového napětí a tvorba invertovaného napájení pro předzesilovač	76
B.3 Vstupní obvod pro připojení zvonkového napětí	77
B.4 Předzesilovač pro elektretový mikrofón	78

Tabulky

2.1	Obecné požadavky	11
2.2	Funkční požadavky	11
3.1	Zapojení signálů CSI sběrnice ..	14
3.2	Zapojení signálů DSI sběrnice ..	16
3.3	Parametry zvukových karet firmy Axagon dle zdroje [17]	16
3.4	Kmitočty kanálů Wi-Fi a jejich použití dle standardu IEEE 802.11	17
3.5	Zapojení pinů GPIO konektoru .	21
3.6	Matematický popis vybraných okénkových funkcí	42
3.7	Parametry vybraných okénkových funkcí	43
3.8	Popis hlavičky zvukového souboru WAV	44
3.9	Popis hlavičky obrazového souboru BMP	47



Seznam použitých zkratk a symbolů

Zkratka	Význam
IoT	internet věcí (internet of things)
PLC	programovatelný logický automat (programmable logic controller)
3D	třídídimenzionální (three dimensional)
EMC	elektromagnetická kompatibilita (electromagnetic compatibility)
EMI	elektromagnetické rušení (electromagnetic interference)
EMS	elektromagnetická odolnost (electromagnetic susceptibility)
ESD	elektrostatický výboj (electrostatic discharge)
UML	unifikovaný modelovací jazyk (unified modeling language)
TFT	tenkovrstvý tranzistor (thin-film transistor)
LCD	displej s tekutými krystaly (liquid crystal display)
PIR	pasivní infračervený detektor (passive infrared detector)
FIR	konečná impulzní odezva (finite impulse response)
IIR	nekonečná impulzní odezva (infinite impulse response)
BMP	bitmapa (bitmap)
DIB	bitmapa nezávislá na zařízení (device independent bitmap)
RGB	barvonosné složky červená, zelená, modrá (red, green, blue)
PCM	pulzně kódová modulace (pulse-code modulation)
ENBW	ekvivalentní šířka pásma šumu (equivalent noise bandwidth)
DDS	přímá číslicová syntéza (direct digital synthesis)
ADC	analogově číslicový převodník (analog digital converter)
CAS	počítačový algebraický systém (computer algebra system)
NTP	protokol synchronizace času v síti internet (network time protocol)
FET	tranzistor řízený polem (field effect transistor)
GSM	standard pro mobilní telefony (global system for mobile communications)

Zkratka	Význam
JPEG	Joint Photographic Experts Group
AVI	formát multimediálního videa (audio video interleave)
VAW	zvukový formát (Waveform audio file format)
RIFF	souborový formát pro ukládání multimediálních dat (resource interchange file format)
LSB	nejméně významný bit (least significant bit)
MSB	nejvíce významný bit (most significant bit)
MPx	megapixel (megapixel)
CMOS	complementary metal oxide semiconductor
RISC	počítač s redukovanou instrukční sadou (reduced instruction set computer)
LED	světlo vyzařující dioda (light emitting diode)
ARM	firma Advanced RISC Machines
GPL	všeobecná veřejná licence (general public licence)
SBC	jednodeskový počítač (single board computer)
RTC	obvod reálného času (real time clock circuit)
GPIO	vstupně-výstupní rozhraní pro obecné účely (general purpose input/output)
USB	univerzální sériová sběrnice (universal serial bus)
I ² C	meziobvodová sériová sběrnice (inter-integrated circuit)
SDA	seriová data (serial data)
SCA	seriové hodiny (serial clock)
SPI	sériové periferní rozhraní (serial peripheral interface)
CSI	kamerové sériové rozhraní (camera serial interface)
DSI	displejové sériové rozhraní (display serial interface)
DPI	displejové paralelní rozhraní (display parallel interface)
INT	přerušeni (interrupt)
DFT	diskrétní Fourierova transformace (discrete Fourier transform)
FFT	rychlá Fourierova transformace (fast Fourier transform)
DIT	decimace v čase (decimation in time)
DIF	decimace ve kmitočtu (decimation in frequency)
BCD	binárně kódované dekadické číslo (binary coded decimal)
AVČR	Akademie Věd České Republiky (Czech Academy of Science)
RMS	efektivní hodnota (root mean square)
ESR	ekvivalentní sériová rezistivita (equivalent series resistance)
DIN	Německý ústav pro průmyslovou normalizaci (Deutsches Institut für Normung e. V.)
DVD	optický disk pro záznam dat (digital versatile disk)
PDF	přenositelný formát dokumentu (portable document format)
CAD	počítačově podporovaný návrh (computer aided design)
EOF	konec souboru (end of file)

Zkratka	Význam
API	aplikačně-programové rozhraní (application programming interface)
RLE	run-length encoding
DPI	počet bodů na palec (dots per inch)
ORM	objektově-relační mapování (object-relation mapping)
MVC	architektura model-pohled-řadič (model-view-controller)
JSP	Java Server Pages
JPA	Java Persistence Api
SD	Secure Digital
SDHC	Secure Digital High Capacity)
FAT32	souborová alokační tabulka (file allocation table)
PHP	Hypertextový preprocesor (Hypertext preprocessor)
SQL	strukturovaný dotazovací jazyk (structured query language)
KDE	K Desktop Environment
XFCE	XForms Common Environment
LXDE	Lightweight X11 Desktop Environment
DHCP	dynamicky konfigurovaný protokol (dynamic host configuration protocol)
DNS	systém doménových jmen (domain name service)
PC	osobní počítač (personal computer)
IP	internetový protokol (internet protocol)
RAM	paměť s náhodným přístupem (random access memory)
FPGA	hradlové pole (field programmable gate array)
DSP	digitální signálový procesor (digital signal processor)

Symbol	Význam
G [dB]	decibel se vztažnou jednotkou výkonu
P [W]	výkon
L [dB]	decibel se vztažnou jednotkou napětí
Z [Ω]	impedance
U [V]	napětí
u(t)	napětí proměnné v čase

Kapitola 1

Úvod

Elektronická zařízení a zabezpečovací technika zastupuje v dnešní době již značnou pozici v budovách, a to nejen inteligentních. Čím dál častěji se setkáváme s domácí automatizací, zejména pak se vzdáleným ovládním, monitorováním a automatickým řízením některých technologických celků (vytápění, větrání, klimatizace, osvětlení, zavlažování zahrady, filtrace a desinfekce bazénu, apod.). Fenomén posledních let - IoT, tzv. internet věcí, integraci elektroniky a přeměnu budov a domácností na tzv. chytré domácnosti, ještě urychluje.

Velmi často se pro tyto účely používá hardware a technologií, jejichž aplikace byla ještě před pár lety omezena pouze na sektor nadšenců do techniky. Prakticky se tak do domácí automatizace, mimo tradičních PLC (programovatelných logických automatů), dostala například platforma Arduino a její podobné klony. Je však nutné podotknout, že zařízení založená na těchto, vesměs tzv. prototypovacích, platformách nemusejí splňovat některé požadavky, které jsou běžně kladeny na hotová zařízení. Mezi tyto požadavky patří například požadavek na samozhášivost skříňky přístroje, v případě požáru, neboť skříňky těchto zařízení jsou vyrobeny zejména metodou 3D tisku, dále neméně důležitý požadavek na EMC (elektromagnetickou kompatibilitu), který se dělí na dvě části - EMI (elektromagnetické rušení), tedy jaké rušení dané zařízení produkuje, a EMS (elektromagnetická odolnost), tedy vůči jakému rušení je dané zařízení odolné. Zařízení konstruované na prototypovacích platformách také vesměs nemá ochranné prvky vůči ESD (elektrostatickým výbojům) nebo tranzientnímu přepětí na vstupně-výstupních portech (transily, supresorové diody, jiskřiště, filtry, apod.), což je dáno jednak koncipováním prototypovacích platforem pro použití v návrhových kancelářích nebo laboratořích, ve kterých se zásady ESD, při práci s těmito vývojovými prostředky, zpravidla dodržují, dále zde hraje roli velikost desky plošného spoje a v neposlední řadě také výrobní, respektive konečná, cena těchto výrobků. Prakticky jsou tak vstupně-výstupní brány elektronických obvodů vyvedeny přímo na konektory.

Elektronická kukátka lze zařadit částečně mezi prvky zabezpečovací techniky, neboť jimi lze monitorovat prostor za dveřmi objektu, vybaveného tímto zařízením. V případě dražších modelů, tato zařízení nabízejí prakticky kompletní zabezpečovací systém, nabízí se však otázka, jak moc lze takové

zařízení považovat za robustní zabezpečovací systém, který by spolehlivě obstál, například vůči pečlivě připravenému vloupání nebo sabotáži. Zcela jistě však lze elektronická kukátka zařadit mezi prvky sběru audiovizuálních dat.

Praktickým výstupem práce by měl být návrh a realizace zařízení poskytujícího základní funkcionality elektronického kukátka na platformě Raspberry PI s použitím přídatných komponent, jednak oficiálně dostupných, tak i vlastní konstrukce. Zařízení by mělo umožňovat živé monitorování prostoru za dveřmi, ukládání obrazových dat, jednak na požadavek uživatele, tak i dle dané situace (zazvonění dveřního zvonku, pohyb osoby za dveřmi, apod.) a mělo by umožňovat provádět základní správu těchto dat.

Důvodem návrhu vlastního zařízení typu elektronické kukátko, je praktická absence zařízení na trhu, které by bylo vybavené software s otevřeným zdrojovým kódem (například pro možnosti budoucího rozšíření funkcionalit nebo možnosti aktualizace na nejnovější komunikační standardy a technologie), umožňovalo by připojení již existující topologie zapojení dveřního zvonku pro spouštění pořízení obrazového záznamu a v neposlední řadě také akustické detekce zvuku zvonku, založené na číslicovém zpracování signálů.

Kapitola 2

Rozbor současného stavu

Cílem této kapitoly je nalézt zařízení běžně dostupná na trhu, shrnout jejich funkcionality a představit i technologické prostředky použité pro realizaci těchto zařízení a jejich funkcionalit. Dále popsat návrh našeho zařízení z analytického pohledu, zejména z hlediska požadavků na software a hardware. Na závěr této kapitoly zformulujeme požadavky pomocí diagramu do případů užití, který nám objasní, jaké funkcionality bude výsledné zařízení nabízet.

2.1 Zařízení dostupná na trhu

Prakticky všechna zařízení lze pořídit do částky 10000 Kč, avšak žádné ze zařízení není založeno na software s otevřeným zdrojovým kódem (opensource), a není tedy možné v budoucnu firmware zařízení aktualizovat na nejnovější komunikační standardy a protokoly (například šifrovací) a hardwarové zapojení těchto zařízení je rovněž proprietární a nelze tedy tato zařízení v budoucnu doplnit novými funkcionalitami nebo dalšími kusy hardware. Žádné z nabízených kukátek nedisponuje možností připojení externího zvonkového tlačítka ani akustickou detekcí zvuku zvonku. Dražší modely kukátek poskytují pouze vlastní tlačítko zvonku.

Digitální dveřní kukátko R03

Přístroj distribuuje firma GM Electronic spol. s r. o. za cenu 960 Kč. V současnosti je tento výrobek veden již jako doprodej, ale jedná se patrně o nejlevnější zařízení na trhu, které je vybaveno nedotýkovým barevným displejem TFT o uhlopříčce 2,8". Kukátko je ovládáno pomocí čtyř tlačítek. Napájení je realizováno třemi tužkovými bateriemi (typ AA), doporučenými jsou baterie alkalického typu. Vzhledem k pracovnímu rozmezí napájecího napětí, které činí 3,5 až 4,5 V, lze očekávat, že zařízení bude pracovat i s dobíjecími akumulátory (typicky Ni-Cd nebo Ni-MH, jejichž napětí je 1,2 V na článek v plně nabitém stavu), avšak při minimálním napětí akumulátorů (cca 1,0 V na článek) již nebude zařízení fungovat.

Zařízení umožňuje pohled do prostoru za dveřmi a pořizování snímků na základě požadavku obsluhy (majitele objektu). Snímky o rozlišení 0,3 MPx

jsou ukládány ve formátu JPEG na externí paměťovou kartu typu Micro SD o maximální kapacitě 32 GB.

■ Digitální kukátko Richter Czech

Jedná se o další výrobek z řady levných elektronických kukátek, s pořizovací cenou 1044 Kč, který distribuuje internetový obchodní dům MALL.CZ. Jedinou poskytovanou funkcionalitou je zobrazování prostoru za dveřmi na dobu 10 s, po stisknutí tlačítka. Rozlišení obrazu je 0,3 MPx, uhlopříčka displeje LCD je 2,6", napájení je realizováno dvěma bateriemi typu AAA. Pozorovací úhel kamery je 100°.

■ Brinno Peephole Viewer PHV132512

Další z modelů levnějších digitálních dveřních kukátek, který se prodává za cenu 1990 Kč u distributora Alfa.cz (T.S.Bohemia). Jedinou funkcionalitou je rovněž zobrazování prostoru za dveřmi na displeji TFT o uhlopříčce 2,5". Hlavním účelem tohoto zařízení je zprostředkování funkcionality běžného kukátka osobám menšího vzrůstu (děti, senioři) nebo také handicapovaným osobám na invalidních vozících.

Zařízení je napájeno dvěma bateriemi typu AA, které, dle katalogových údajů, umožní zhruba 1700 náhledů prostoru za dveřmi, každý o délce 10 s. Jako obrazový sensor je použit snímač typu CMOS, který poskytuje snímky o rozlišení 1,3 MPx. Přístroj umožňuje také zvětšení obrazu (někdy označováno termínem zoom), není však uvedeno, jedná-li se o zvětšení optické nebo digitální, lze ale očekávat, že vzhledem k ceně zařízení, běžným cenám optiky, běžně dostupným kamerovým modulům, použitému snímači, ale i rozsahu potřebného zvětšení (odhadem do 5x zvětšení), bude transfokace s největší pravděpodobností implementována softwarově. Zařízení lze bez problému namontovat na dveře o tloušťce 35 - 57 mm.

■ Digitální dveřní kukátko VT2090

VT2090 je přístroj disponující již konektivitou, nočním viděním a IR pohybovým senzorem. Zařízení umožňuje kontaktovat majitele objektu přes síť Wi-Fi nebo GSM. V zařízení je použitý procesor architektury ARM v7 o dvou jádrech na taktovacím kmitočtu 1,2 GHz. Do kukátka lze instalovat paměťovou kartu, na kterou probíhá záznam fotografií o rozlišení 2 MPx (pravděpodobně ve formátu JPEG) nebo videa ve formátu AVI o rozlišení 1024x768 pixelů. Přístroj disponuje dotykovým displejem TFT o uhlopříčce 7". Zařízení je postaveno na operačním systému Android ve verzi 4.4.2. Vzhledem k architektuře hardware a vzhledu zařízení se patrně jedná o stavbu zařízení na základech tabletu. Pozorovací úhel kamery je 130°. Cena přístroje je 5990 Kč a distribuuje jej firma Valtech-team.

Digitální dveřní kukátko VT2095

Pravděpodobně nejdražším kukátkem, distribuovaným firmou Valtech-team, je typ VT2095 s cenou 7390 Kč. Tento model je ve své podstatě modelem VT2090, avšak doplněným o další příslušenství, díky kterým nabízí v podstatě kompletní malý zabezpečovací systém. Součástí standardní dodávky je nejen samotné dveřní kukátko, které je obdobně, jako model VT2090 od firmy Valtech-team řešeno na operačním systému Android, ale také detektory úniku plynu a kouře, pohybové čidlo PIR, magnetický dveřní sensor, sensor úniku vody a také siréna a dálkový ovladač. Veškerá tato příslušenství, vyjma kamerového modulu, který je k LCD panelu připojen vodiči, jsou řešena jako bezdrátová. Pozorovací úhel kamery je rovněž 130°.

2.2 Požadavky na zařízení a případy užití

Protože budeme navrhovat zařízení, které má jisté očekávané funkcionality, zformulujeme si požadavky na výsledné zařízení. Zpravidla se dělí požadavky na požadavky funkční a obecné. Funkčními jsou takové požadavky, které úzce souvisejí s funkcí zařízení a bez formulace těchto požadavků by výsledný produkt neměl přílišného významu. Obecné požadavky (někdy také označované jako nefunkční) nejsou již tolik spjaty s funkcí zařízení, avšak není možné je zanedbávat. Velmi často se totiž mezi obecné požadavky zařazují například požadavky na použité technologie, verze software, výslednou licenci, apod.

Z hlediska softwarového inženýrství platí, že změnou obecných požadavků by se neměla prakticky změnit funkčnost software. U návrhů systémů, kdy je součástí návrhu hardware i software, nemusí tato podmínka být vždy splněná, neboť se můžeme dostat změnou některé ze softwarových technologií, programového vybavení, ale i použitím algoritmu s příliš vysokou časovou nebo paměťovou složitostí na výkonový limit hardware. Časová náročnost algoritmu bývá označována jako asymptotická složitost tzv. notací $O(f(x))$. Velmi často jdou požadavky časové i paměťové složitosti algoritmů proti sobě, to jest menší časová složitost vyústí ve větší paměťovou složitost a naopak. Také se vyskytují případy, kdy zvyšující se časová složitost algoritmu zároveň zvyšuje i složitost paměťovou a naopak. Je nutné vzít v úvahu, že časová složitost algoritmu spotřebovává zpravidla jen tzv. strojový čas, zatímco složitost paměťová spotřebovává nejen paměťový prostor, ale i strojový čas, neboť se k daným datům musíme být schopní dostat, to jest umět je zpravidla přečíst a v mnoha případech i zapsat (u většiny pamětí bývá navíc operace zápisu pomalejší, než operace čtení). Je tedy důležité, aby korespondovaly funkční a obecné požadavky i s výsledným hardwarem, zvláště, jedná-li se o tzv. vestavný hardware.

Zadání práce nám určuje hlavní obecný požadavek, využití platformy Raspberry PI s možností doplnění o další podpůrné moduly a obvody s možností konstrukce vlastních modulů, ale i využití již modulů hotových. Protože je software počítače Raspberry PI zpravidla založen na principech otevřeného

zdrojového kódu a operačním systému Unixového typu, který tyto principy ctí a komunita vyvíjející tento systém zpravidla vyžaduje po autorech software, pro tento systém určený, dodržovat přístup otevřeného zdrojového kódu, je vhodné tento princip zahrnout do obecných požadavků.

V našem případě mezi funkční požadavky zcela jistě zahrneme možnost zobrazení prostoru za dveřmi na displej kukátka, neboť touto funkcionalitou disponují i nejlevnější přístroje, a navíc lze očekávat, že montáží elektronického kukátka dojde k nahrazení již existujícího ryze optického kukátka, a obsluha (majitel bytu) by byl nucen, v případě absence této funkce, vždy používat chytrého telefonu, tabletu či počítače.

Většina elektronických kukátek rovněž umožňuje uložit zobrazený snímek na paměťové médium (nejčastěji paměťová karta), proto tento požadavek zahrneme i do našeho návrhu. Levnější modely umožňují ukládat snímky pouze na požadavek majitele bytu, dražší modely disponují vlastním tlačítkem zvonku, protože však lze očekávat již existující topologii zapojení zvonku a zvonkového tlačítka, kterou majitel bytu nebude chtít zpravidla zcela nahradit zvonkem v kukátku a externím tlačítkem poskytovaným zařízením, lze považovat za vhodný funkční požadavek rovněž možnost zapojení elektronického kukátka do již existující topologie. Ze zadání práce nám vyplývá rovněž požadavek ukládání snímku v závislosti na detekci pohybu za dveřmi a akustickou detekci zvonku.

Měli bychom mít také možnost získaná a uložená data prohlížet a spravovat (archivace, ukládání do PC, odstranění). Levnější kukátka mají tuto funkcionalitu zajištěnou pomocí vyjímatelné paměťové karty. Dražší kukátka mají tuto funkcionalitu zajištěnou již lokální správou pomocí uživatelského rozhraní zobrazovaného přímo na zařízení. Protože ze zadání vyplývá obecný požadavek na využití platformy Raspberry PI, která poskytuje dostatek prostředků pro realizaci lokální správy nebo, již v dnešní době u podobných zařízení velmi rozšířené a ze zadání vyplývající, vzdálené správy získaných dat bez nutnosti vyjímání paměťového média ze zařízení, lze zahrnout požadavek na správu obrazových dat na zařízení také mezi funkční požadavky.

Získaná data je vhodné přiřazovat k údajům o datu a času, kdy byla pořízena, proto je nutné mít možnost také nastavit datum a čas v zařízení. Obdobně by měla být možnost nastavit referenční zvukový signál pro akustickou analýzu. Je vhodné, aby zařízení umožňovalo povolení či zakázání detekce pohybu za dveřmi a detekce zvonku. Zakázání detekce pohybu za dveřmi je vhodné zejména v případě provádění řemeslných prací ve snímaném prostoru, ale i v prostorech, kde se očekává velký pohyb lidí, neboť by docházelo k častému ukládání obrazových dat. Zakázání detekce zvonku je užitečné například v případě očekávání velkého počtu hostů do domácnosti nebo pro otestování funkčnosti zvonku.

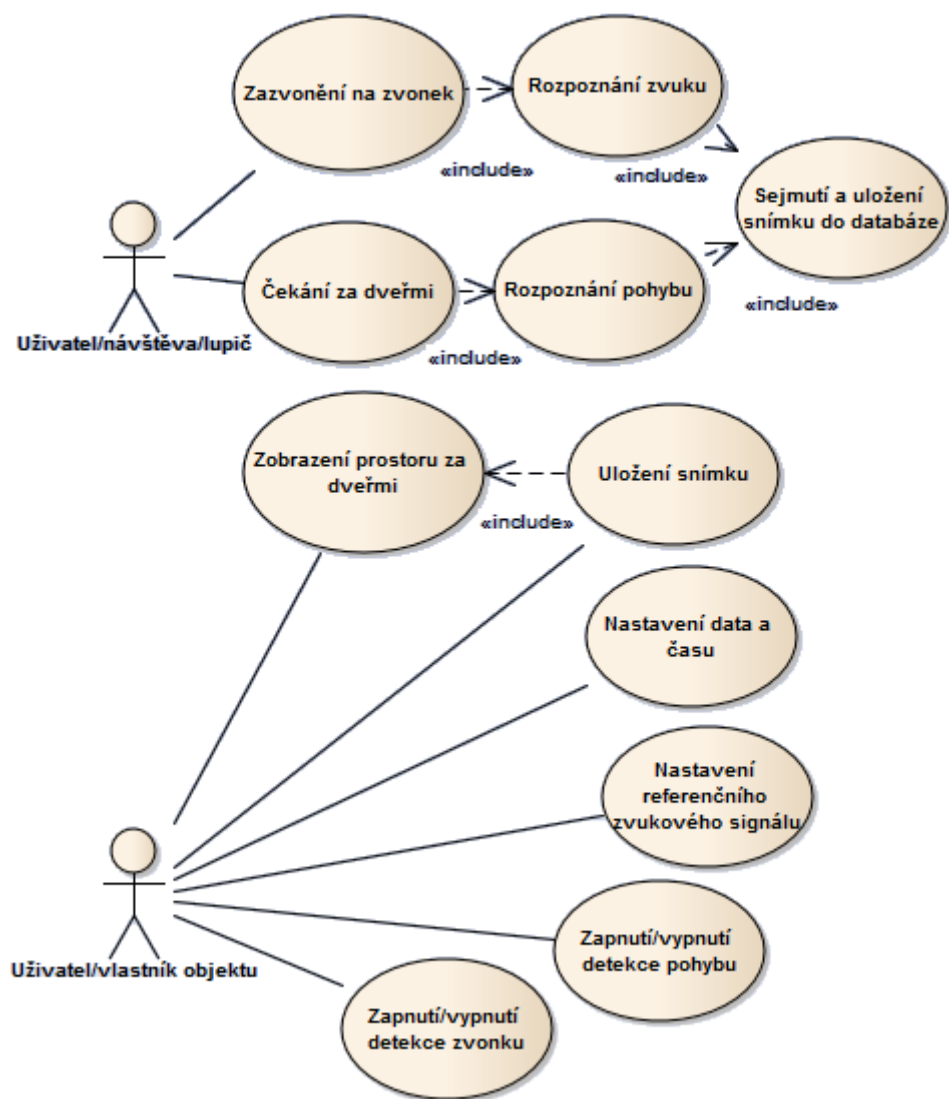
Tabulky 2.1 a 2.2 shrnují jednotlivé požadavky na výsledný hardware a software a obrázek 2.1 znázorňuje možné případy užití systému (dle UML tzv. usecase diagram).

Tabulka 2.1: Obecné požadavky

Požadavek	Popis požadavku
Použití mikropočítače Raspberry PI	Zařízení by mělo být postavené na mikropočítači Raspberry PI s využitím dalších podpůrných modulů a obvodů. Je možné využít již hotových modulů nebo navrhnout a vyrobit modul vlastní.
Použití volně dostupné vývojové technologie	Celé zařízení by mělo využívat výhradně software s bezplatnou licencí, s otevřeným zdrojovým kódem, licencí MIT, licencí GPL, apod.

Tabulka 2.2: Funkční požadavky

Požadavek	Popis požadavku
Zobrazení prostoru za dveřmi	Zařízení musí umožnit zobrazení prostoru za dveřmi nezávisle na jiných akcích.
Uložení snímku	Zařízení musí umožnit uložení snímku, a to zejména na tyto události: <ul style="list-style-type: none"> ■ uživatelský vstup ■ pohyb za dveřmi ■ zazvonění zvonku (vstup elektrického signálu) ■ zazvonění zvonku (vstup akustického signálu)
Správa uložených obrazových dat	Zařízení musí umožnit správu uložených obrazových dat. Obrazová data by mělo být umožněno zobrazovat, ukládat (stahovat) a mazat.
Vzdálený přístup	Zařízení by mělo umožňovat vzdálený přístup s využitím některé ze síťových technologií s možností správy uložených obrazových dat.
Nastavení data a času	Zařízení musí umožnit nastavit datum a čas dle požadavku uživatele.
Nastavení referenčního zvukového signálu	Zařízení by mělo umožnit nastavit referenční zvukový signál pro akustickou detekci zvuku, zejména zvonku.
Nastavení detekce pohybu	Zařízení musí umožnit zapnutí nebo vypnutí detekce pohybu v prostoru za dveřmi na přání uživatele. Pro detekci pohybu bude použito obrazových dat a je možno užít již hotové knihovny.
Nastavení detekce zvonku	Zařízení musí umožnit zapnutí nebo vypnutí detekce zvonku, a to jak detekce elektrického signálu, tak i signálu akustického.



Obrázek 2.1: Diagram znázorňující možné případy užití systému

Kapitola 3

Návrh a realizace

V kapitole je popisován návrh a realizace hardware a software elektronického kukátka v souladu s obecnými a funkčními požadavky a případy užití uvedených v podkapitole 2.2. Protože základem elektronického kukátka je hardware, bez kterého není možné software realizující požadované funkcionality provozovat, a vyplývá nám částečně ze zadání (platforma Raspberry PI), budeme se nejprve věnovat popisu této platformy včetně všech univerzálních přídatných modulů, které je možné v rámci hardware elektronického kukátka vhodně použít a návrhu vlastního obvodového řešení pro tuto specifickou oblast použití.

Ve druhé části je popsán software provozovaný na elektronickém kukátku, a to jak možné operační systémy dostupné pro platformu Raspberry PI, tak i software pro správu pořízených dat, software detekující zvonek a software detekující pohyb v obrazových datech.

3.1 Návrh hardware

Počítač Raspberry PI je tzv. jednodeskový počítač založený na mikroprocesorech s redukovanou instrukční sadou (RISC), architektury ARM a konkrétně se můžeme setkat s obvody řady BCM2835, BCM2836 a BCM2837, nejčastěji od firmy Broadcom, v případě BCM2835 i od firmy Samsung.

Původně byl tento počítač navržen britskou nadací Raspberry PI Foundation s cílem použití v rámci výuky programování a elektroniky ve školách. Postupem času si však tento počítač oblíbili kutilové, konstruktéři a stavitelé prototypů, neboť Raspberry PI nabízí relativně vysoký taktovací kmitočet (v současné verzi již 1,4 GHz na čtyřjádrovém procesoru BCM2837) a kapacitu operační paměti (u současné verze 1 GB) a stává se tak solidním základem pro realizaci složitějších aplikací.

Zařízení disponuje několika rozhraními. Zejména se jedná o, v našem případě nejpodstatnější, rozhraní pro připojení kamery - CSI (camera serial interface), rozhraní pro připojení displeje - DSI (display serial interface), porty USB, které jsou k dispozici celkem 4, a vstupně-výstupní brány s obecným použitím GPIO (general purpose input output), které jsou vyvedeny na 40 pinový konektor (u dřívějších verzí počítače 20 pinový) o roztečích 2,54 mm, kde jsou také dostupná napájecí napětí +5 V, +3,3 V a přístrojová zem.

3.1.1 Kamera

Mezi volitelné oficiální příslušenství počítače Raspberry PI patří kamera komunikující po rozhraní CSI. Propojení mezi deskou počítače a modulem kamery je realizováno ohebným plochým kabelem o 15 vodičích, po kterých probíhá nejen datová komunikace, ale i napájení modulu kamery.

V současné době je možné se setkat zpravidla se čtyřmi variantami kamerových modulů. Kamerový modul se obecně nabízí se dvěma různými značkami obrazovými sensory poskytujícími rozdílné rozlišení a každý z nich je navíc vyráběn ve dvou dalších verzích podle toho, zda-li se jedná o standardní verzi nebo verzi NoIR, která není oproti standardní verzi osazena infračerveným filtrem a lze tedy takový modul, v případě použití LED diody emitující záření v infračervené oblasti, použít i pro snímání obrazu ve tmě. Značnou nevýhodou NoIR kamerového modulu je poskytování barevně zkraslených dat při použití ke snímání obrazu při viditelném světle. Obrazovým senzorem pro 5 MPx verzi kamerového modulu sensor OmniVision OV5647 a pro 8 MPx verzi se jedná o sensor Sony IMX219.

Ovladač s názvem V4L2 pro tyto kamerové moduly je již součástí standardních distribucí operačního systému Raspbian a je možné, v současné době, provozovat na tento ovladač, jak 5 MPx verzi, tak verzi 8 MPx, obě samozřejmě jak ve standardním provedení, tak v provedení NoIR. Použití některého z těchto kamerových modulů je tedy vhodné. Protože je však rozhraní CSI, pro připojení kamerových modulů k počítači Raspberry PI, proprietární, není možné nalézt v dokumentaci očekávané úrovně napětí, logických úrovní a přenosových rychlostí na jednotlivých vodičích této sběrnice, ale pouze zapojení konektoru, které je uvedeno v následující tabulce 3.1.

Tabulka 3.1: Zapojení signálů CSI sběrnice

Pozice	Význam
1	Napájení - přístrojová zem
2	CAM1_DN0
3	CAM1_DP0
4	Napájení - přístrojová zem
5	CAM1_DN1
6	CAM1_DP1
7	Napájení - přístrojová zem
8	CAM1_CN
9	CAM1_CP
10	Napájení - přístrojová zem
11	CAM_GPI00
12	CAM_GPI01
13	SCL0
14	SDA0
15	Napájení + 3,3 V

Uvážíme-li jistou podobnost se zapojením konektoru rozhraní DSI, popsaného dále v sekci 3.1.2, pak CAM1_DNx bude znamenat datový vodič s negativními napěťovými úrovněmi, CAM1_DP_x bude znamenat datový vodič s kladnými napěťovými úrovněmi, CAM1_CN vodič s negativními synchronizačními pulzy a CAM1_CP vodič s kladnými synchronizačními pulzy, kde x značí číslo datového páru vodičů. Na konektor jsou vyvedeny rovněž piny GPIO a sběrnice I²C.

■ 3.1.2 Displej

Ačkoliv existuje mnoho displejů připojitelných k počítači Raspberry PI, například na sběrnici I²C, SPI, nebo sběrnici DPI (display parallel interface), jejichž použití vyžaduje další konfigurace (konfigurace pinů do alternativních funkcí), a v případě použití displeje na sběrnici DPI bychom navíc využili prakticky všechny vstupně-výstupní brány dostupné na GPIO konektoru, nejví se jejich použití jako příliš vhodné. Pouze jediný dostupný displej komunikuje po sběrnici DSI, která je na desce počítače dostupná na zvláštním konektoru. Oficiální dotykový displej o uhlopříčce 7", s podporou více doteků současně, patří mezi další příslušenství vyvinuté nadací Raspberry. Obraz se přenáší po sběrnici DSI prakticky shodným 15 žilovým plochým kabelem, jakým je připojen kamerový modul. Data z dotykové vrstvy jsou přenášena za pomoci sběrnice I²C, buďto přes konektor DSI nebo přes zvláštní vodiče (SDA a SCL), které jsou připojeny na vstupně-výstupní brány s obecným použitím (GPIO). Propojení napájení displeje s napájením počítače Raspberry PI je realizováno rovněž zvláštními vodiči na napájecí piny (+5V a přístrojová zem) konektoru GPIO. Displej poskytuje navíc další pin s názvem INT, který není v současné době připojen k počítači Raspberry PI, a při měření za pomoci osciloskopu nevykazuje žádné změny logických úrovní. Je však možné, že v budoucích revizích hardware a software displeje bude mít, vzhledem ke svému pojmenování (INT - angl. interrupt, tedy přerušeni), funkci vyvolání přerušeni v cílovém systému, například na základě doteku na displej.

V tabulce 3.2 jsou uvedeny pozice a význam jednotlivých vodičů sběrnice DSI. V rozhraní DSI je využito dvojice diferenčních párů, po kterých probíhá datová komunikace. Data jsou synchronizována jedním hodinovým signálem, který je rovněž přenášen pomocí jednoho diferenčního páru vodičů. Logické úrovně se pohybují v rozmezí 0 až ± 200 mV dle polarity příslušného vodiče. Na sběrnici DSI je rovněž dostupné napájení, které napájí integrovaný obvod řídicí displej. Napájení je dostupné na více pinech, a to jednak z důvodu umožnění větší proudové zatížitelnosti daného napájení, dále snížení úbytku napájecího napětí na přenosové cestě a v neposlední řadě pro redukci možných přeslechů mezi páry vodičů přenášejících data a synchronizační signál, neboť i napájecí napětí pocházející z tvrdého zdroje prokládané na plochém kabelu mezi datovými vodiči lze použít k redukci přeslechů mezi těmito datovými vodiči.

Nejvhodnější zvuková karta pro náš účel, vzhledem k dostupnosti a parametřům je Axagon ADA-17 MINI HQ, neboť má největší odstup signálu od šumu a nejmenší harmonické zkreslení. Stejně, jako karta ADA-15 MINI HQ, ani zbylé dvě karty nepotřebují ke své funkci dodatečnou uživatelskou instalaci ovladačů do operačního systému.

3.1.4 Wi-Fi adaptér

Rovněž Wi-Fi adaptéry je možné připojit k mikropočítači Raspberry PI s využitím sběrnice USB. Stejně, jako v případě zvukové karty, i zde mohou nastat potíže s kompatibilitou. Mikropočítač Raspberry PI 3 Model B+ však již má adaptér kombinující Wi-Fi a Bluetooth osazený na základové desce a navíc umožňuje provozovat současně síť Wi-Fi v režimu hotspot (tj. adaptér síť Wi-Fi poskytuje) i v režimu klienta (tj. adaptér se do již existující sítě Wi-Fi připojuje). Většina USB Wi-Fi adaptérů současný provoz v obou režimech neumožňuje. V současném návrhu elektronického kukátka se sice oba režimy provozu nevyužijí, neboť přístroj bude pouze síť Wi-Fi poskytovat (režim hotspot), avšak nebude do budoucna limitováno možné rozšíření funkcionalit.

Vestavěný Wi-Fi adaptér rovněž umožňuje, vzhledem k celosvětovému určení jediné hardwarové verzi počítače, nastavit prakticky jakýkoliv kanál Wi-Fi sítě dle standardu IEEE 802.11b,g,n, pokud se takový kanál v dané zemi používá (země, kde se počítač provozuje se nastavuje ručně). USB Wi-Fi adaptéry mají zpravidla nastavena omezení na nastavení kanálu sítě Wi-Fi již od výrobce. Tabulka 3.4 shrnuje kmitočty jednotlivých kanálů a jejich souhrnné použití v různých částech světa, některé kanály však nemusejí být povoleny v závislosti na cílové zemi.

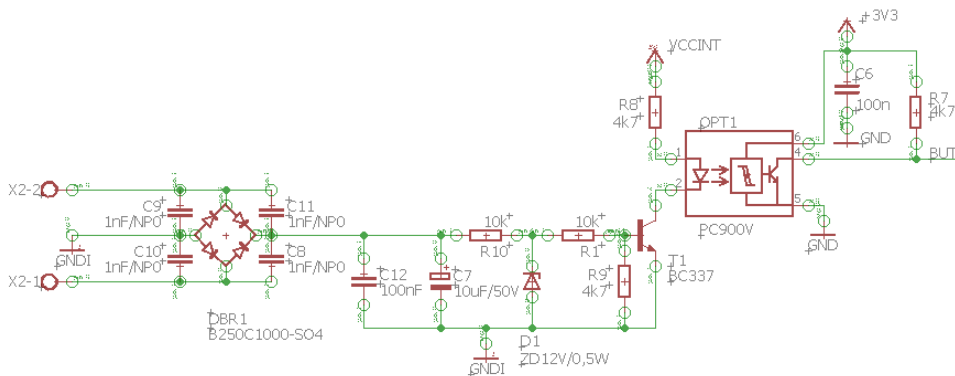
Tabulka 3.4: Kmitočty kanálů Wi-Fi a jejich použití dle standardu IEEE 802.11

Č. kanálu	Kmitočet [MHz]			Použití		
	nejnižší	střední	nejvyšší	EU	USA	Japonsko
1	2401	2412	2423	ANO	ANO	ANO
2	2406	2417	2428	ANO	ANO	ANO
3	2411	2422	2433	ANO	ANO	ANO
4	2416	2427	2438	ANO	ANO	ANO
5	2421	2432	2443	ANO	ANO	ANO
6	2426	2437	2448	ANO	ANO	ANO
7	2431	2442	2453	ANO	ANO	ANO
8	2436	2447	2458	ANO	ANO	ANO
9	2441	2452	2463	ANO	ANO	ANO
10	2446	2457	2468	ANO	ANO	ANO
11	2451	2462	2473	ANO	ANO	ANO
12	2456	2467	2478	ANO	NE	ANO
13	2461	2472	2483	ANO	NE	ANO
14	2473	2484	2495	NE	NE	Pouze 802.11b

davých, ve které mohou nastat nejrůznější tranzientní přepětí a jiné podobné poruchy, je velmi účinnou ochranou přenášený signál galvanicky oddělit, aby v případě poruchy přepětí nezničilo vstup počítače, ale případně pouze vstupní obvody před galvanickým oddělením a galvanický oddělovač. Rovněž je nutné uvážit teoretickou možnost připojení napětí mimo rozsah, kdy nám galvanické oddělení rovněž ochrání vstup procesoru. Nejčastěji se k účelu galvanického oddělení logického signálu používá optočlen, neboť vykazuje při této aplikaci nejlepší parametry (izolační schopnost, přenos stejnosměrné složky) při zachování nízké ceny. V praxi jsou však možné ještě další dva způsoby galvanického oddělení signálů, kapacitní bariérou nebo pomocí, již zmíněného, transformátoru. Oba tyto způsoby galvanického oddělení standardně nepřenášejí stejnosměrnou složku signálu, avšak vyrábějí se monolitické integrované obvody kapacitních galvanických oddělovačů (Burr-Brown ISO122 a ISO124) a hybridní integrované obvody transformátorových galvanických oddělovačů (Analog Devices AD202 a AD204), které jsou schopné přenášet i ryze stejnosměrné signály díky modulaci signálu na vysílací straně a demodulaci na straně přijímací. Protože však pořizovací cena takovýchto galvanických oddělovačů je vysoká a pro naší aplikaci plně postačuje funkce jednoduchého optočlenu se Schmittovým klopným obvodem, využijeme v našem návrhu jednoduchý optočlen PC900V od firmy Sharp. Použití Schmittovo klopného obvodu je vhodné, neboť zaručuje krátké setrvání výstupního napětí v tzv. zakázaném pásu, v případě příliš dlouhého setrvání výstupního napětí v zakázaném pásu můžeme vstupní bránu mikropočítače poškodit.

$$U_{RMS} = \frac{U_{MAX}}{\sqrt{2}} \quad (3.1)$$

$$U_{MAX} = U_{RMS} \cdot \sqrt{2} \quad (3.2)$$



Obrázek 3.1: Schéma vstupního obvodu pro připojení zvonkového napětí

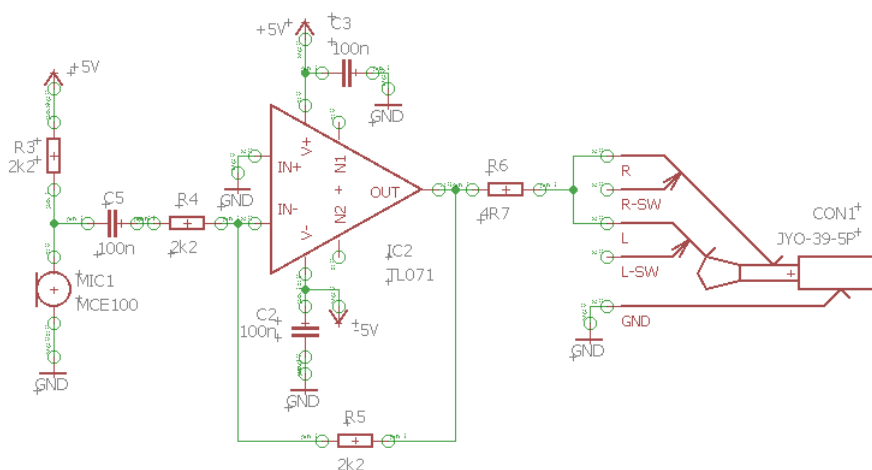
Obrázek 3.1 znázorňuje vstupní obvod pro paralelní připojení ke zvonku. Obvod je dimenzován na připojení zvonkového napětí 3 V až 24 V střídavých, respektive s využitím úpravy vzorce (3.1) pro výpočet efektivní hodnoty střídavého napětí s průběhem sinusovým z hodnoty maximálního napětí sinusového průběhu na tvar (3.2) je vstupní obvod schopen pracovat také

Tabulka 3.5: Zapojení pinů GPIO konektoru

Pozice	Význam
1	Napájení + 3,3 V
2	Napájení + 5 V
3	I ² C sběrnice - vodič SDA (datový vodič sběrnice)
4	Napájení + 5 V
5	I ² C sběrnice - vodič SCL (hodinový vodič sběrnice)
6	Napájení - přístrojová zem
7	GPIO - pin procesoru č. 4
8	UART - Tx - vysílací datový vodič
9	Napájení - přístrojová zem
10	UART - Rx - přijímací datový vodič
11	GPIO - pin procesoru č. 17
12	GPIO - pin procesoru č. 18, alternativně PWM0
13	GPIO - pin procesoru č. 27 (dříve 21)
14	Napájení - přístrojová zem
15	GPIO - pin procesoru č. 22
16	GPIO - pin procesoru č. 23
17	Napájení + 3,3 V
18	GPIO - pin procesoru č. 24
19	SPI0 - MOSI
20	Napájení - přístrojová zem
21	SPI0 - MISO
22	GPIO - pin procesoru č. 25
23	SPI0 - SCLK
24	SPI0 - chip select 0
25	Napájení - přístrojová zem
26	SPI0 - chip select 1
27	GPIO - pin procesoru č. 0
28	GPIO - pin procesoru č. 1
29	GPIO - pin procesoru č. 5
30	Napájení - přístrojová zem
31	GPIO - pin procesoru č. 6
32	GPIO - pin procesoru č. 12, alternativně PWM0
33	GPIO - pin procesoru č. 13, alternativně PWM1
34	Napájení - přístrojová zem
35	SPI1 - MISO
36	GPIO - pin procesoru č. 16
37	GPIO - pin procesoru č. 26
38	SPI1 - MOSI
39	Napájení - přístrojová zem
40	SPI1 - SCLK

Oproti měřicí technice, kde se zaměřujeme zejména na parametry operačního zesilovače, jako jsou již zmíněné vstupní nesymetrie napěťové a proudové a vstupní klidový proud, nás ve zvukové technice zajímá zejména rychlost přeběhu. V obou případech aplikace nás zajímá také šum samotného operačního zesilovače. Pro naši aplikaci je vhodný například operační zesilovač TL071 od firmy Texas Instruments se vstupní částí (diferenčním zesilovacím stupněm) realizovaným pomocí tranzistorů J-FET, který vykazuje nízký vstupní proud. Operační zesilovač TL071 rovněž vykazuje vysokou rychlost přeběhu $13 \text{ V} \cdot \mu\text{s}$, která značně převyšuje požadavek v naší aplikaci, ikdybychom totiž zvukový signál omezovali pouze napájecími napětími $+5 \text{ V}$ a -5 V (což v případě operačního zesilovače TL071 nenastane, neboť není typu rail-to-rail), dostali bychom se s využitím zjednodušeného vztahu pro výpočet rychlosti přeběhu (3.3) pro případ sinusového periodického zvukového signálu o kmitočtu 20 kHz na rychlost přeběhu $0,4 \text{ V}/\mu\text{s}$. Navíc je operační zesilovač TL071 velmi často využívá právě ve zvukové technice. Operační zesilovač vyžaduje bipolární napájení, které však počítač Raspberry PI neposkytuje, proto z napájecího napětí $+5 \text{ V}$ vytvoříme napájecí napětí -5 V pomocí obvodu ICL7660 standardně zapojeného dle katalogového listu. Obvod ICL7660 je nábojová pumpa pracující na spínacím kmitočtu 10 kHz , lze tedy očekávat, že tento šum se částečně přenesou do analyzovaného zvukového signálu, avšak protože se nejedná o Hi-Fi aplikaci a je možné předpokládat, že se spínací kmitočet v průběhu provozu nebude výrazně měnit a takto vzniklá harmonická složka bude mít podstatně nižší úroveň, než harmonické složky užitečného signálu, můžeme si tento obvodový návrh dovolit. Harmonická složka 10 kHz vzniklá použitím nábojové pumpy tedy bude vystupovat teoreticky shodně jak ve vzorovém, tak i analyzovaném signálu a tedy by neměla analýzu ovlivňovat.

$$SR = \frac{U_{MAX} - U_{MIN}}{\frac{1}{\frac{f_{MAX}}{2}}} \quad (3.3)$$



Obrázek 3.3: Schéma předzesilovače elektretového mikrofónu

možné pro konstrukci oscilátoru využít i krystal o kmitočtu 4,194304 MHz, neboť je možné takovéto kmitočty snadno vydělit metodou opakovaných čtverců, a to 2^{15} , respektive 2^{22} . Nejjednodušší dělička kmitočtu 2 je sestavena z vhodně zapojeného klopného obvodu typu D a tato dělička je za sebou zapojena 15 krát pro kmitočet 32,768 kHz, respektive 22 krát pro kmitočet 4,194304 MHz.

Výstup 1 Hz kmitočtu je pak veden do soustavy BCD čítačů, které jsou vhodně zapojeny tak, že při svém přetečení inkrementují čítač jiný, apod.

Prakticky je však možné obvod reálného času zakoupit jako integrovaný obvod, který výše zmíněnou logiku obsahuje a velmi často přidává další funkcionality, jako například paměť typu RAM, nejrůznější programovatelné alarmy, apod.

Mezi nejvýznamnější výrobce těchto obvodů patří firma Maxim Integrated a Dallas Semiconductor, méně známými výrobci RTC obvodů jsou například firma Microchip a Epson. Vyjmenujme alespoň některé zástupce integrovaných obvodů reálného času, které jsou běžně dostupné na trhu.

- **DS1302** je časovač na sériovou sběrnici 3-wire. Umožňuje napájecí napětí v rozmezí 2 až 5,5 V a obvod má vyvedený oddělený napájecí pin pro napájení záložní baterií respektive akumulátorem poskytujícím napětí rovněž v rozmezí 2 až 5,5 V, který je možné obvodem i programovatelně nabíjet skrze soustavu diod a rezistorů (tj. kontaktním proudem s limitací napětí). Obvod pro svoji korektní funkci vyžaduje krystal o základním kmitočtu $f_0 = 32,768$ kHz.
- **DS1306** je časovač na sériovou sběrnici, podporující přímo sběrnici SPI (serial peripheral interface) v módu 1 a 3 a nebo sběrnici 3-wire, jako obvod DS1302. Stejně jako obvod DS1302, i tento pracuje ve stejném rozmezí napájecího a zálohovacího napětí a je rovněž nutné obvod doplnit o krystal 32,768 kHz. Narozdíl od obvodu DS1302, tento neumožňuje nabíjení akumulátoru použitého pro zálohu času.
- **DS1307** je časovač na sériovou sběrnici I²C. Tento obvod je možné napájet napětím pouze v rozmezí od 4,5 do 5,5 V, což jej činí nevhodným pro systémy pracující pouze na napájení + 3,3 V, avšak je možné jeho rozhraní používat na 3,3 V CMOS logice, neboť jeho vstupně-výstupní brány SCL a SDA jsou typu otevřený kolektor a vyžadují externí pull-up rezistory, tak jak to bývá u I²C sběrnice obvyklé. Raspberry PI napájecí napětí + 5 V poskytuje, rovněž je k dispozici hardwarová podpora I²C sběrnice, na kterou je již připojena komunikace dotekové vrstvy displeje, mohlo by připojení časovače na tuto sběrnici zvětšit odezvu mikropočítače na ovládání pomocí dotekového displeje. Časovač vyžaduje krystal 32,768 kHz.
- **MCP7940N** je časovač na sběrnici I²C, ale na rozdíl od předchozích typů, je jeho výrobcem firma Microchip. Výhodou tohoto obvodu je jednak možnost napájení širokým rozmezím napájecích napětí od 1,8 do 5,5 V, také možnost nízkého zálohovacího napětí, které musí být

Zařízení watchdog timer je obvykle koncipováno jako čítač s předem určeným celkovým časovým intervalem, během kterého se čítač inkrementuje, a po uplynutí nastavené doby dojde k přetečení čítače a zařízení vyše resetovací puls do příslušného systému. Pokud však během inkrementace čítače je zresetován watchdog timer (zpravidla kontrolovaným systémem), běží tento čítač znovu od počáteční - většinou nulové hodnoty.

Watchdog timer mají některé procesory přímo zahrnutý ve své architektuře jako svoji periférii, ty procesory, které interní watchdog timer nemají je nutné v případě požadavku na watchdog timer doplnit vnějším obvodem (například Intersil X5045, Analog Devices ADM705, apod.). Vnější obvody watchdog timer jsou většinou koncipovány jako obvody na sériové sběrnice, u kterých je možno nastavit maximální dobu čítání čítače a zápisem do určité oblasti dojde k jejich vynulování. Dalším typem jsou obvody s pevně nastavenou dobou a s resetovacím vstupem, který reaguje většinou na sestupnou hranu signálu (z důvodu častého připojení na výstup s otevřeným kolektorem a pull-up rezistorem, kdy sestupné hrany jsou podstatně strmější, než hrany náběžné, kde jsou patrné integrace způsobené vlivem nabíjení parazitních kapacit spojů a vstupů). Další možností velmi jednoduchého watchdog timeru je možnost implementace této komponenty do zvláštního mikroprocesoru, je však více, než vhodné koncipovat daný program co nejjednodušší, do jediné programové smyčky, bez jakéhokoliv přerušení a v neposlední řadě též v co nejnižším programovacím jazyce (například jazyk symbolických adres dané architektury).

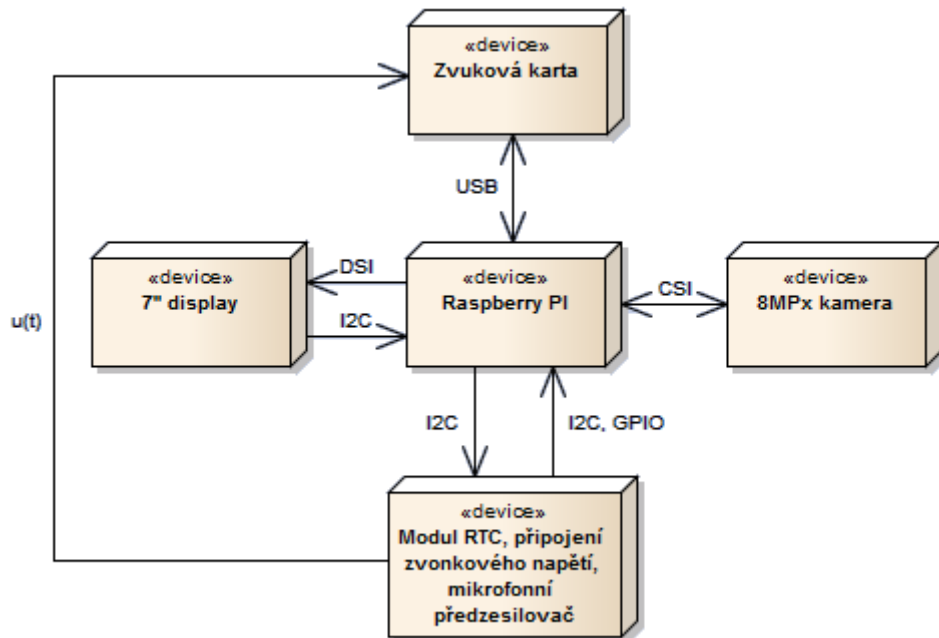
Dalším typem obvodu, někdy nazývaného také jako watchdog, je obvod monitorující napájecí napětí, který v případě nutnosti vyše resetovací puls. Důvodem k resetu mohou být kromě přílišného poklesu napájecího napětí také krátkodobá podpětí nebo krátkodobé (tranzientní) napěťové špičky či přepětí, neboť při všech těchto situacích může dojít k narušení konzistence dat v pamětech typu RAM.

Mikropočítač Raspberry PI je standardně vybaven obvodem monitorujícím napájecí napětí, avšak nikoliv obvodem watchdog timeru. Použití externího watchdog timeru je sice možné, neboť Raspberry PI může skrze vstupně-výstupní brány komunikovat s watchdog timerem, rovněž je možné, aby watchdog timer počítač Raspberry PI zrestartoval, pomocí zkratování dvou pinů s označením RUN umístěného u DSI portu, avšak toto resetování se ve výsledku provádí pouze softwarově, tedy ekvivalent příkazu: `sudo reboot -h now`, používaného v operačních systémech typu Unix nebo Linux.

Provádění tzv. tvrdého restartu (například odpojením napájecího napětí) není výrobcem počítače doporučeno, neboť při takovýchto opakovaných restartech může dojít k narušení konzistence dat operačního systému na paměťové kartě a celá situace může v krajním případě způsobit nespustitelnost standardně distribuovaných operačních systémů pro tento počítač určených. Návrh vlastního operačního systému pro elektronické kukátko, který by bylo možné bez obav za běhu úloh vypínat, je nad rámec této diplomové práce.

Funkcionalitu interního watchdogu zastupuje rovněž softwarový balíček s názvem `watchdog`, který je možné bez obav použít, neboť se nejedná o

bezpečnostně-kritickou aplikaci. Watchdog tedy nebudeme hardwarově realizovat na modulu vlastní konstrukce, neboť jeho realizace by oproti softwarovému balíčku nepřinesla zásadní změny a odpojování napájecího napětí počítače není vzhledem k běhu operačního systému doporučeno.



Obrázek 3.4: Blokové zapojení

3.2 Číslicové zpracování zvuku

Protože navržené elektronické kukátko má umožnit detekci pohybu na základě obrazových dat a akustickou detekci zvuku zvonku, je vhodné si popsat možné způsoby zpracování takovýchto signálů. Obraz i zvuk jsou z pohledu matematického aparátu, užitého v číslicovém zpracování signálů, diskrétními signály. Nejprve si popíšeme zvukový signál, neboť ten je na popis jednodušší.

Zvukový signál musí být zpravidla signál proměnný v čase, neboť vnímání zvuku v běžném prostředí (vzduch o běžném atmosférickém tlaku) je založeno na změnách akustického tlaku. Slyšitelný zvuk je pro lidské ucho takový zvuk jehož harmonické kmitočty jsou zhruba v rozmezí od 20 Hz do 20 kHz.

Zvukový signál je obecně v čase spojitý, avšak pro účely číslicového zpracování signálu je nutné jej tzv. navzorkovat, čímž se z prostoru se spojitým časem přesuneme do prostoru s časem diskrétním. Matematicky lze zapsat tento převod jako zobrazení (3.4).

$$x(t), t \in \mathbb{R} \rightarrow y(n), n \in \mathbb{Z} \quad (3.4)$$

$x(t)$ reprezentuje zvukový signál spojitý v čase, tj. přirozeně vznikající nebo analogově zpracovaný, zatímco $y(n)$ reprezentuje zvukový signál diskrétní

v čase, tj. zaznamenaný například zvukovou nebo měřicí kartou.

Diskrétní zvukový záznam je tedy minimálně jednodimenzionálním signálem, kdy vzorky zvukového signálu jsou funkcí diskrétního času. Je však nutné zmínit, že i zvuk může být vícedimenzionálním signálem, tj. v čase může být přítomno více zvukových signálů, například v případě stereofonie nebo kvadrafonie, pak tedy vzorky zvukového signálu jsou nejen funkcí diskrétního času, ale mohou být i funkcí tzv. zvukového kanálu.

Volba vzorkovacího kmitočtu se řídí dle Shannon-Nyquistova vzorkovacího teorému, který říká, že chceme-li, být schopní rekonstruovat zaznamenaný signál, musí být tento signál zaznamenan vzorkovací frekvencí, která je větší, než dvojnásobek maximálního kmitočtu signálu, který požadujeme rekonstruovat. Pro věrnou rekonstrukci se však doporučuje zaznamenat signál vzorkovacím kmitočtem zhruba pětinasobným, než je maximální kmitočet zpracovávaného signálu. Za maximální slyšitelný kmitočet je považována, již zmíněná, hodnota 20 kHz, je tedy nutné, aby zvuková karta snímala zvuk minimálně 40 kHz vzorkovacím kmitočtem. Typickými vzorkovacími kmitočty zvukových karet jsou 44,1 kHz, 48 kHz a 96 kHz. I minimální vzorkovací frekvence zvukové karty tedy splní Shannon-Nyquistův teorém.

Obecně se detekcí podobnosti dvou objektů zabývá věda zvaná umělá inteligence, disciplína zvaná datamining. V Českém jazyce se pro disciplínu zvanou datamining vžil pojem „vytěžování dat“. V našem případě se budeme zabývat detekcí podobnosti dvou zvuků, konkrétně zvuků zvonku. V praxi se detekce podobnosti zvuku používá v mnoha aplikacích.

Jednou z těchto aplikací může být akustický detektor rozbití skleněné výplně, neboť tříštivý zvuk skla má jasně identifikovatelný zvuk. Detektory rozbití skla jsou však zpravidla koncipovány ryze analogovými filtry (kombinace integračního a derivačního članku s operačním zesilovačem) a to zejména z důvodu požadavku na nízkou cenu výsledného zařízení. Z tohoto důvodu také není možné těmito detektory detekovat rozbití skla vybaveného fólií, neboť zvuk tříštění takovéto výplně není již tak jednoznačně identifikovatelný a detektor by rozbití výplně nemusel zdetekovat.

Další aplikací detekce podobnosti zvuků může být například elektronická ladička hudebních nástrojů. I zde je však pravděpodobné užití pásmových propustí z důvodu nízké ceny a malého příkonu výsledného zařízení.

■ 3.2.1 Analýza v časové oblasti

Pro detekci podobnosti dvou signálů je možné užít několik způsobů. Nejjednodušší je užití aparátu dostupného v časové oblasti, kdy se analyzují průběhy zvuků (signálů) v čase (tzv. obálka signálu), můžeme teoreticky uvažovat, že se stejné signály vícekrát navzorkované v ideálním případě nebudou lišit. V praxi však bude zpravidla docházet k situacím, kdy se začne vzorkovat vždy na jiné části průběhu signálu (tzv. posun v čase). Negativní vliv posunu v čase na analýzu můžeme řešit například detekcí změny znaménka amplitudy (obecně průchod měnícího se signálu klidovou úrovní) a následně se pokusit spárovat stejně dlouhé intervaly period. Dalším negativním vlivem je měnící se amplituda analyzovaného signálu vůči signálu referenčnímu (změna hlasitosti)

3.2.3 Korelace

Korelace umožňuje vzájemné porovnávání navzorkovaných signálů v časové oblasti. Vstupem algoritmu jsou tedy dvě posloupnosti vzorků signálů a výstupem korelace je opět posloupnost vzorků, která vyjadřuje nejen podobnost porovnávaných signálů, ale je možné z výstupní posloupnosti zjistit i například zpoždění dvou podobných signálů vůči sobě nebo nalézt vzor hledaného signálu v signálu zarušeném. Algoritmus korelace vyjadřuje vzorec (3.5).

$$\hat{R}_{x_1, x_2}[k] = \frac{1}{N} \sum_{n=0}^{N-k-1} x_1[n] \cdot x_2[n+k] \quad (3.5)$$

Ze vzorce je patrná funkce algoritmu korelace, která je založena na vzájemném vynásobení dvou vzorků ze vstupních signálů a jejich sečtení, přičemž první prvek výstupní posloupnosti je výsledkem součtu součinů z celé vstupní posloupnosti o délce N . Prvek na další pozici výstupní posloupnosti je však již výsledkem součtu součinů ze vstupní posloupnosti o délce $N-1$, kdy je navíc jeden ze vstupních signálů vůči prvnímu posunut o jeden vzorek (například u zvukového záznamu se bude jednat o posun v čase) a bude se provádět již o jeden méně součtů. Poslední prvek korelace bude tvořen pouze součinem prvního prvku vstupní posloupnosti x_1 a posledního prvku vstupní posloupnosti x_2 .

Z již zmíněných vlastností korelace nám vyplývá, že je nutné mít oba vstupní signály převedeny do diskrétní podoby shodnou vzorkovací frekvencí, bez této podmínky by použití algoritmu nemělo smysl, neboť by docházelo k falešným respektive opomenutým detekcím. Oproti tomu je možné, aby obě vstupní posloupnosti měly rozdílný počet vzorků, neboť nám nic nebrání doplnit chybějící vzorky signálu vzorky o nulové hodnotě. V takovém případě opomenuté detekce nehrozí, neboť vždy bude možné nalézt určitý úsek signálu, který je si se vzorovým signálem podobný, se zkracujícím se vstupním signálem (jak vzorovým, tak neznámým) se však zvyšuje pravděpodobnost falešné detekce, neboť může vzniknout velmi podobný úsek signálu i v jinak odlišném signálu.

Algoritmus korelace se však v praxi používá zpravidla v oblastech, kde je určitý předpokládaný průběh signálu, který je například silně zarušen, například radarová technika.

3.2.4 Fourierova transformace

Mnohem komplexnější pohled na problematiku podobnosti dvou signálů poskytuje kmitočtová oblast, neboť lze předpokládat, že každý zvukový signál má více či méně charakteristických harmonických kmitočtů. Abychom získali obraz navzorkovaného signálu ve frekvenční oblasti z oblasti časové, musíme provést Fourierovu transformaci. Obecný vzorec Fourierovy transformace je pro oblast se spojitým časem vyjádřen vzorcem (3.6). Transformovat tak lze pouze analyticky vyjádřitelné integrandy (signály), které jsou na intervalu $(-\infty; \infty)$ integrovatelné, to jest je na tomto intervalu funkce alespoň po

částech spojitá (bodů nespojitosti funkce může být konečný počet nebo jich může být spočetně mnoho, ale nesmí jich být nekonečně mnoho, zcela obecně musí mít množina bodů nespojitosti této funkce nulovou Lebesgueovu míru).

$$F(\omega) = \int_{-\infty}^{\infty} f(t) \cdot e^{-j\omega t} dt \quad (3.6)$$

Protože se však pohybujeme v oblasti s diskretním časem, musíme použít diskretní verzi Fourierovy transformace. Nejjednodušší forma Fourierovy transformace v diskretní formě je popsána vzorcem (3.7).

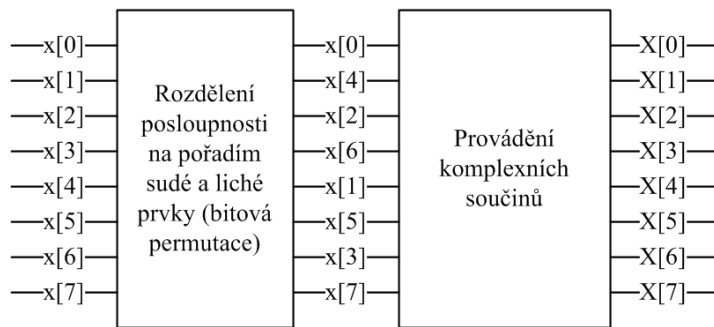
$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-j \cdot \frac{2\pi kn}{N}} \quad (3.7)$$

Člen $e^{-j \cdot \frac{2\pi kn}{N}}$ se někdy také nazývá tzv. twiddle faktorem. Využijeme-li převodu komplexního čísla z exponenciálního tvaru na tvar goniometrický, získáme vzorec (3.8).

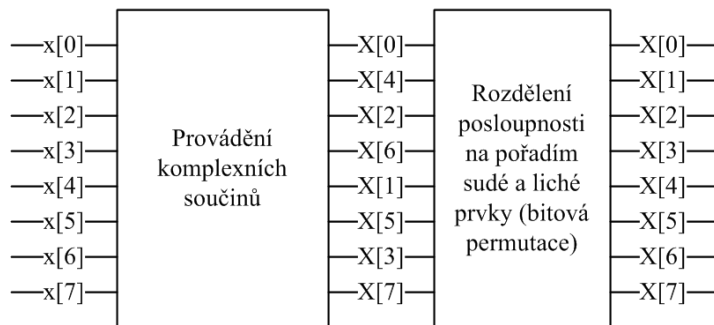
$$X_k = \sum_{n=0}^{N-1} x_n \cdot \left[\cos\left(\frac{2\pi kn}{N}\right) - i \cdot \sin\left(\frac{2\pi kn}{N}\right) \right] \quad (3.8)$$

Za povšimnutí stojí fakt, že se prvek ze vstupní posloupnosti x_n považuje při převodu tvaru komplexního čísla za jeho absolutní hodnotu, tedy amplituda vzorku vstupního signálu je automaticky absolutní hodnotou komplexního čísla prvku vstupní posloupnosti pro diskretní fourierovu transformaci. Výhodou goniometrického tvaru vzorce je možnost implementace v některém z konvenčních programovacích jazyků, neboť můžeme od sebe oddělit reálnou a imaginární část komplexního čísla, což v případě exponenciálního tvaru komplexního čísla není možné. Diskretní fourierova transformace je tedy založena na komplexních součinech, kterých je obecně celkem N^2 , algoritmus tak má asymptotickou složitost $O = (n^2)$. Pro sudý počet prvků vstupní posloupnosti je však možné počítat pouze první polovinu spektra z první poloviny vstupní posloupnosti, neboť obě poloviny spektra se budou lišit pouze znaménkem imaginární složky. V takovém případě je asymptotická složitost $O = ((\frac{n}{2})^2)$, což však pro velmi velká vstupní data znamená řádově totéž, jako $O = (n^2)$, neboť mocninná funkce roste rychleji, než multiplikační konstanta (0,5). V konečném důsledku je však ale možné, pokud má vstupní posloupnost počet prvků mocniny dvou ($N = 2^Z$), dělit na poloviny diskretní fourierovu transformaci až do tvaru, kdy nám zbydou dva komplexní součiny. Této výhodné vlastnosti využívá efektivní algoritmus rychlé fourierovy transformace FFT (fast Fourier transform), který však navíc ještě odstraňuje redundantní součiny komplexních čísel, které jsou v algoritmu DFT obsaženy. Nejrozšířenější varianta algoritmu FFT je známá jako DIT (decimation in time - decimace v čase) a byla popularizována v roce 1965 americkými matematiky Jamesem Cooleyem a Johnem Tukeyem. Tento algoritmus je založen na rozdělení vstupní posloupnosti na pořadím liché a sudé prvky a následné provedení komplexních součinů s twiddle faktorem (někdy je taková operace nazývána „motýlkem“). Twiddle faktor pro každý vstupní

prvek se liší úhlem na jednotkové kružnici. Výhodou tohoto algoritmu je výstup uspořádané posloupnosti. Existuje však také varianta DIF (decimation in frequency - decimace ve kmitočtu), která rozděluje uspořádanou vstupní posloupnost na prosté poloviny, avšak nevýhoda této varianty je výstup neuspořádané posloupnosti prvků. Pro rozdělení vstupní posloupnosti na sudé a liché prvky je možno užít, zejména v případě iterativních algoritmů, bitové permutace (tj. nahrazení vyšších významových bitů za bity méně významné a naopak symetricky dle středu datového slova), kterou je rovněž možné použít pro uspořádání výstupní posloupnosti, neboť dvojnásobné provedení bitové permutace vrací původní hodnotu. V případě rekurzivních algoritmů, které jsou zpravidla založeny na decimaci v čase, je častější využití rozdělení vstupní posloupnosti na pořadí sudé a liché prvky, neboť opakovaným tříděním prvků vstupní posloupnosti je docíleno bitové permutace indexu vyjadřujícího pořadí prvku. Obecně lze také tvrdit, že rozdíl mezi decimací v čase a decimací ve kmitočtu je pouze v tom, kde je prováděna bitová permutace, zda-li již v časové oblasti nebo až v oblasti kmitočtové, jak znázorňují obrázky 3.5 a 3.6, které demonstrují provádění dvou variant algoritmu rychlé fourierovy transformace nad vstupní posloupností 8 vzorků dat.



Obrázek 3.5: FFT DIT nad 8 prvky vstupní posloupnosti



Obrázek 3.6: FFT DIF nad 8 prvky vstupní posloupnosti

nických kmitočtech. Oproti tomu fázové spektrum nám může poskytnout informace o posunech jednotlivých tónů vůči sobě.

Nyní si ukážeme, jak je možné provádět rychlou fourierovu transformaci v jazyce Matlab, který je typickým zástupcem CAS (computer algebra system) systému. Nejprve si vytvoříme vektor obsahující nezávislou proměnnou (pro představu: v reálné aplikaci by tato proměnná byla s největší pravděpodobností čas).

```
t = 0:1:255;
```

Vektor obsahuje 256 položek s čísly od 0 do 255 včetně, s krokem 1. Protože Matlab má již implementovanou funkci pro výpočet rychlé fourierovy transformace, budeme se držet délky vstupní posloupnosti, která je celočíselnou mocninou čísla 2.

Nyní si utvoříme testovací signál složený ze tří sinusovek o stejných amplitudách, z nichž jedna sinusovka má na intervalu pouze jednu periodu, druhá periody dvě a třetí sinusovka čtyři periody. Očekáváme tedy, že signál bude mít mimo prvního harmonického kmitočtu i jeho druhý a čtvrtý násobek. Matlab umí automaticky pracovat s vektory, proměnná x bude tedy rovněž vektor o 256 prvcích.

```
x = 100*sin(t*((360/256)*(pi/180)))+  
100*sin(t*((720/256)*(pi/180)))+  
100*sin(t*((1440/256)*(pi/180)));
```

Pomocí příkazu `plot(t,x)` můžeme zobrazit, jak vypadá v časové oblasti 256 vzorků vstupního signálu. Osa nezávislé proměnné zde vyjadřuje nejen pořadí vzorku, ale zároveň i čas, který je zde vyjádřen pro vzorkovací kmitočet 1000 Hz. Obecně vypočteme časový rozdíl (Δt v s) mezi dvěma vzorky signálu ze vzorce (3.12), kde N je celkový počet vzorků a f_S v Hz je vzorkovací kmitočet. Osa závislé proměnné zde vyjadřuje, pro účely demonstrace, bezrozměrné číslo úrovní signálu, v praxi by například osa y mohla obsahovat ADC úroveň (číslíkový výstup z analogově-číslíkového převodníku). ADC úrovně však jsou zpravidla jen kladné, pro účely analýzy by bylo nutné rozdělit rozsah na kladnou a zápornou část a hodnoty úrovní přepočítat, neboť pro korektní funkci vyžaduje fourierova transformace nulovou hodnotu pro nulovou (klidovou) úroveň signálu a jakýkoliv offset signálu je analyzován jako stejnosměrná složka a ovlivňuje v kmitočtové oblasti nultý prvek posloupnosti (kmitočet 0 Hz - stejnosměrnou složku).

$$\Delta t = \frac{N}{f_S} \quad (3.12)$$

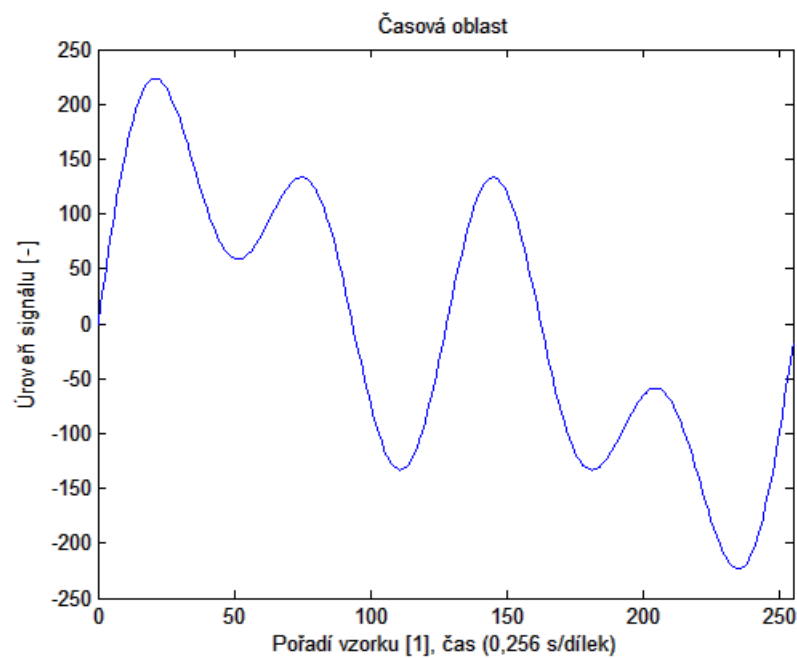
S využitím znalosti vzorce (3.12), znalosti vzorkovacího kmitočtu a znalosti přes kolik vzorků v analyzovaném úseku probíhala jedna perioda určitého kmitočtu jsme schopni vypočítat periody respektive kmitočty. V našem případě se jedná o 256, 128 a 64 vzorků, výsledkem tedy budou periody 0,256 s, 0,128 s a 0,064 s, respektive kmitočty 3,90625 Hz, 7,8125 Hz a 15,625 Hz.

Nyní provedeme nad tímto úsekem signálu rychlou fourierovu transformaci.

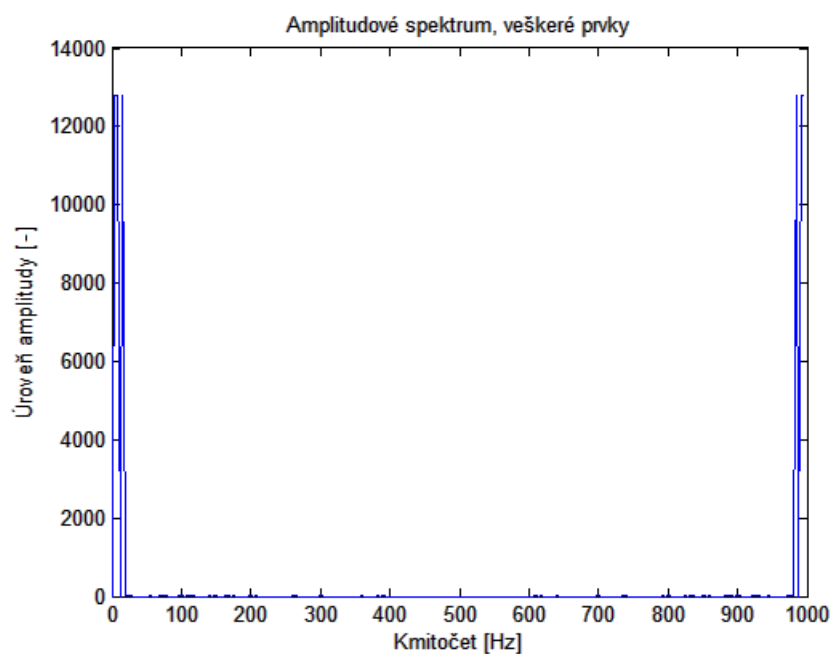
```
y = fft(x)
```

Abychom získali grafickou závislost amplitudy na kmitočtu, musíme dále získat absolutní hodnoty komplexních čísel výstupní posloupnosti:

$$m = \text{abs}(y)$$



Obrázek 3.7: Testovací signál v časové oblasti



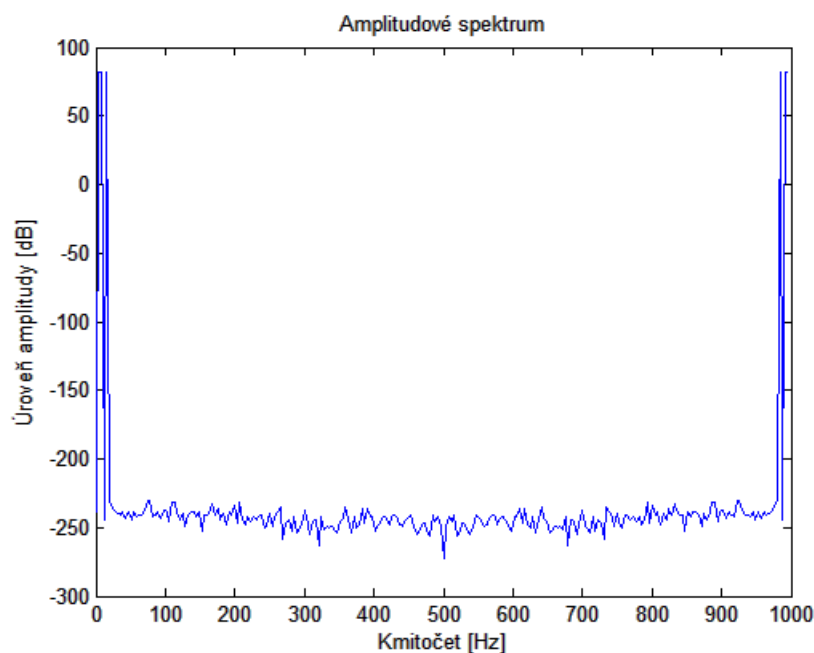
Obrázek 3.8: Amplitudové spektrum testovaného signálu

Dále vytvoříme vektor obsahující nezávislou proměnnou kmitočtů, zde se demonstruje vzorkovací kmitočet 1000 Hz:

```
f = (0:length(y)-1)*1000/length(y);
```

Pro zobrazení amplitudového spektra použijeme příkaz `plot(f,m)`, jehož výstup je na obrázku 3.8.

Na ose nezávislé proměnné, jak již bylo zmíněno, bývají vynášeny kmitočty v Hz (případně v kHz, MHz, apod.) a i v našem příkladu jsme označili osu x dodrželi a za vzorkovací kmitočet jsme si zvolili 1000 Hz a analyzovali jsme právě jednu periodu signálu, jedná se tedy o tzv. neprůměrované spektrum. Osa y v našem případě má pouze informativní charakter, neboť je v bezrozměrných jednotkách, a ukazuje pouze přímý matematický výstup. V praxi bývá většinou v jednotkách dB a může se jednat jak o decibely vztažené k jednotce napětí (se vztažnou hodnotou napětí $U_0 = 0,775 V$ typickou pro telekomunikační techniku, ale i se vztažnou hodnotou napětí $U_0 = 1 V$, která je typická pro komerční zařízení, např. osciloskopy), tak v případě analýzy výkonnostního spektra vztažené k jednotce výkonu (vztažná hodnota výkonu $P_0 = 0,001 W$).



Obrázek 3.9: Amplitudové spektrum po aplikaci funkce `mag2db`

Na některých osciloskopech (např. značky Rigol) je rovněž možné zobrazit spektrum v absolutních jednotkách V (efektivní hodnota napětí). Protože se však absolutní jednotky ve spektrogramech užívají zřídka, nebudeme se touto možností v textu dále zabývat, a ukážeme si možný výstup amplitudového spektrogramu s osou závislé proměnné již v decibelech. Máme možnost použít příkazu `mdB = mag2db(m)`, který je možné v jazyce Matlab použít jak pro výpočet v decibelech vztažených k hodnotě napětí 1 V, tak i pro decibely

vztažené k hodnotě výkonu 1 mW. Výstup tohoto příkazu je znázorněn na obrázku 3.9. Pro snazší určení odstupů obecně n-tých harmonických kmitočtů od prvního harmonického kmitočtu však můžeme užít vztažné úrovně maxima amplitudového spektra, pak je však nutné provádět výpočet decibelů manuálně.

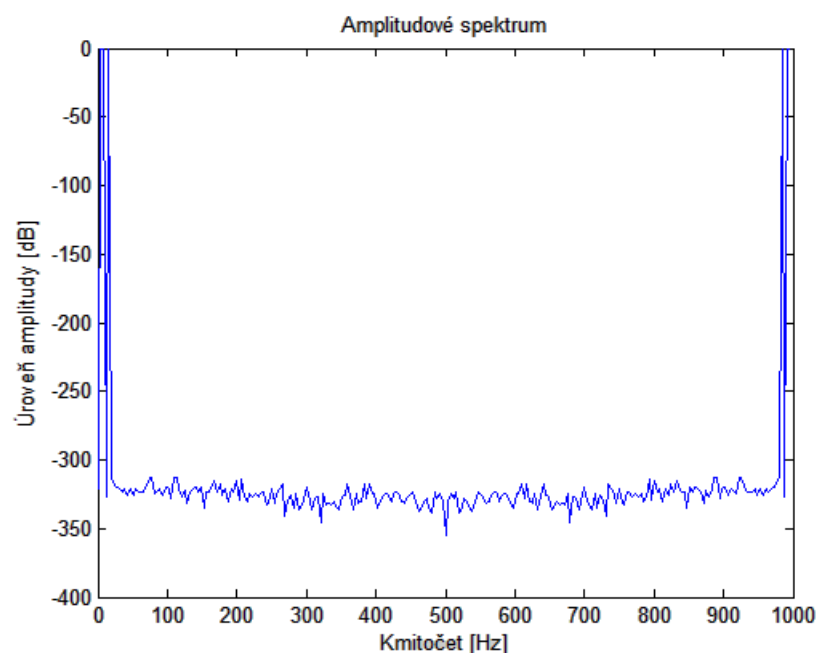
Připomeňme si nyní vzorec pro výpočet poměru v decibelech, které se liší pro případ vztažné jednotky napětí od jednotky výkonu. Vzorec (3.13) představuje základní obecně definovaný vztah pro výpočet decibelů vztažených na jednotku výkonu, ze kterého se vztah pro výpočet decibelů vztažených k jednotce napětí (3.14) odvozuje.

$$G = 10 \cdot \log \frac{P}{P_0} \quad (3.13)$$

$$L = 20 \cdot \log \frac{U}{U_0} \quad (3.14)$$

Příkaz pro výpočet spektra v decibelech se vztažnou jednotkou napětí a vztažnou úrovní maxima amplitudového spektra pak vypadá takto:

mdB = 20*log10(m/max(m)) a jeho výstup je na obrázku 3.10.



Obrázek 3.10: Amplitudové spektrum se vztažnou úrovní maxima spektra

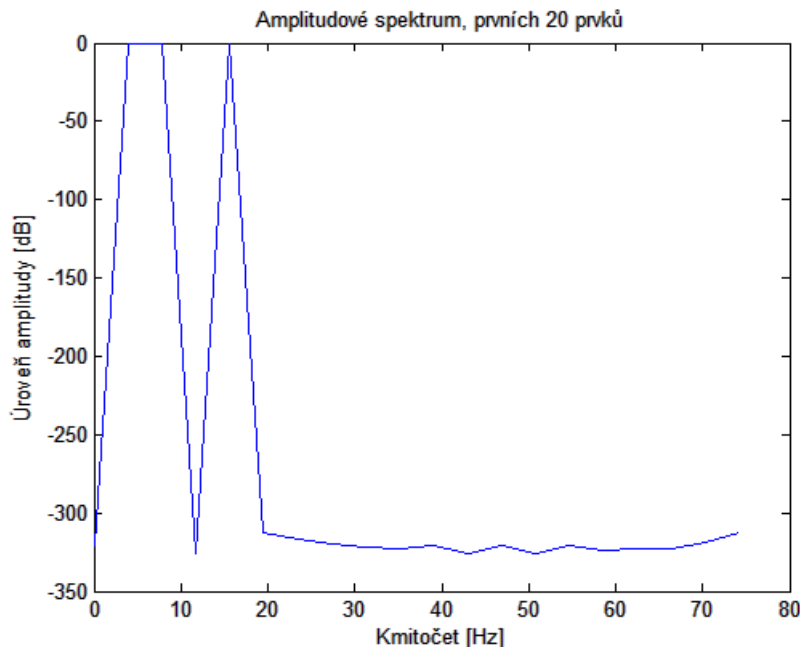
Z výstupu je patrná symetrie amplitudového spektra, výstupu nyní využijeme a zobrazíme diagramy pro 20 prvních, respektive posledních, kmitočtů.

V detailnějším pohledu na začátek a konec amplitudového spektra (obrázky 3.11 a 3.12) je již zřetelné, že spektrum je kolem svého středu symetrické. V praxi se proto využívá výpočtu spektra jen do první poloviny kmitočtů, v našem případě navíc kmitočty druhé poloviny spektra již nejsou relevantní,

neboť signály o takovýchto kmitočtech nejsou vzorkovány v souladu se vzorkovacím teoremem.

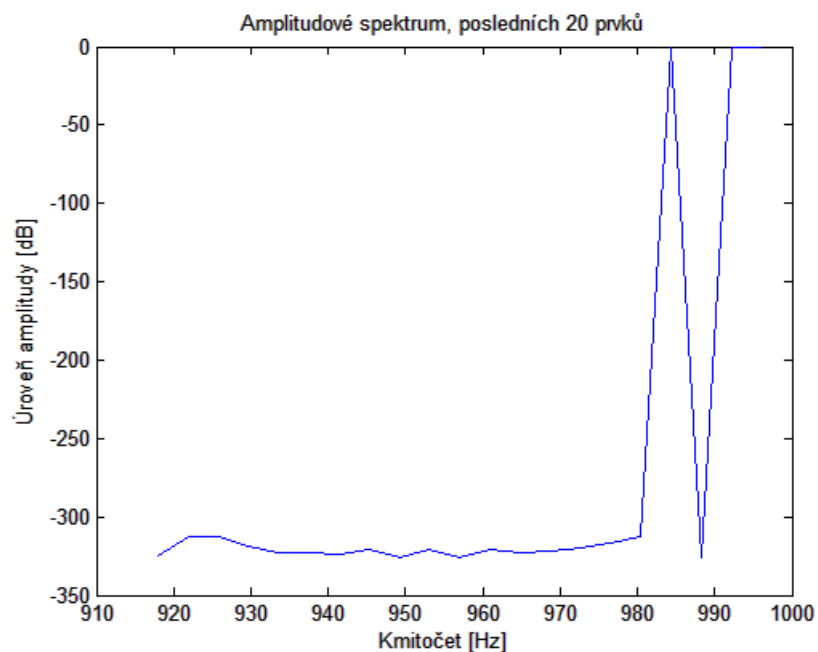
Na výstupu je jasně zřetelná shodná úroveň amplitudy prvního, druhého a čtvrtého harmonického kmitočtu, což koresponduje s matematickým modelem signálu v časové oblasti, kde všechny tři sinusovky byly násobeny konstantou 100.

S využitím znalosti vzorce (3.11), znalosti vzorkovacího kmitočtu, pořadí prvku a celkový počet prvků amplitudového spektra můžeme zjistit kmitočty jednotlivých harmonických složek. Abychom však našli harmonické kmitočty, musíme nalézt hodnoty amplitud podstatně větší, než jsou úroveň šumu, z amplitudového spektra a určit jejich pořadí. První prvek takovéto posloupnosti se nazývá 1. harmonická složka. Velmi často mívá tato složka maximální hodnotu amplitudy ze spektra, avšak toto nemusí zcela obecně platit, neboť existují signály mající například 2. harmonickou složku výraznější. Z časové oblasti však víme, že máme signál složený z 3 harmonických složek, konkrétně z 1., 2. a 4. harmonické složky a víme, že je v analyzovaném úseku právě jedna perioda 1. harmonické složky, dvě periody 2. harmonické složky a čtyři periody 4. harmonické složky. 1. harmonická složka má tak v tomto případě pořadí $k = 1$, neboť se jedná o nejnižší kmitočet, který lze v tomto případě analyzovat, dále už je jen stejnosměrná složka $k = 0$, která v modelovém případě nebyla užita. 2. harmonická složka bude mít v tomto případě tedy pořadí $k = 2$ a 4. harmonická složka $k = 4$. Provedeme-li výpočet, zjistíme kmitočty 3,90625 Hz, 7,8125 Hz a 15,625 Hz, což koresponduje s kmitočty vypočítanými v časové oblasti.

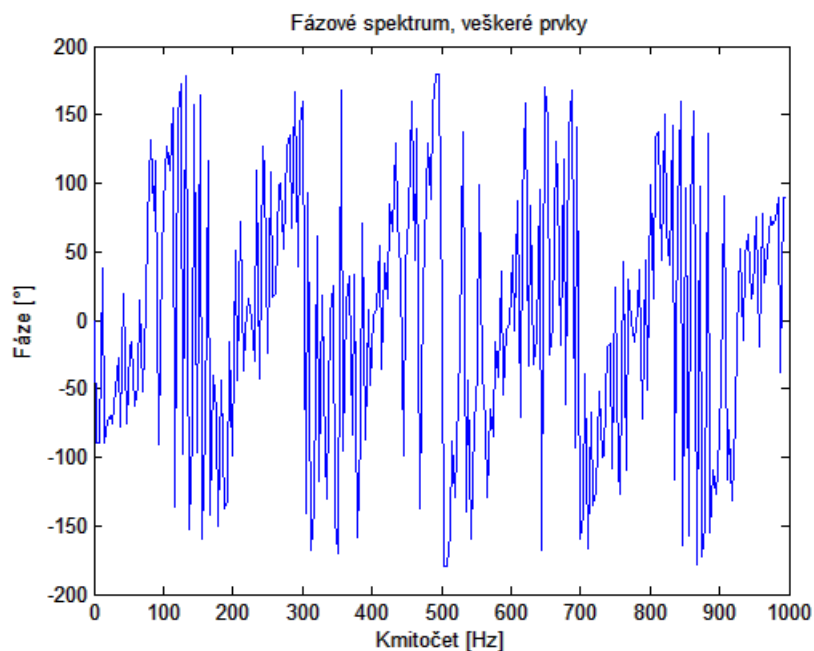


Obrázek 3.11: Detail prvních 20 prvků amplitudového spektra

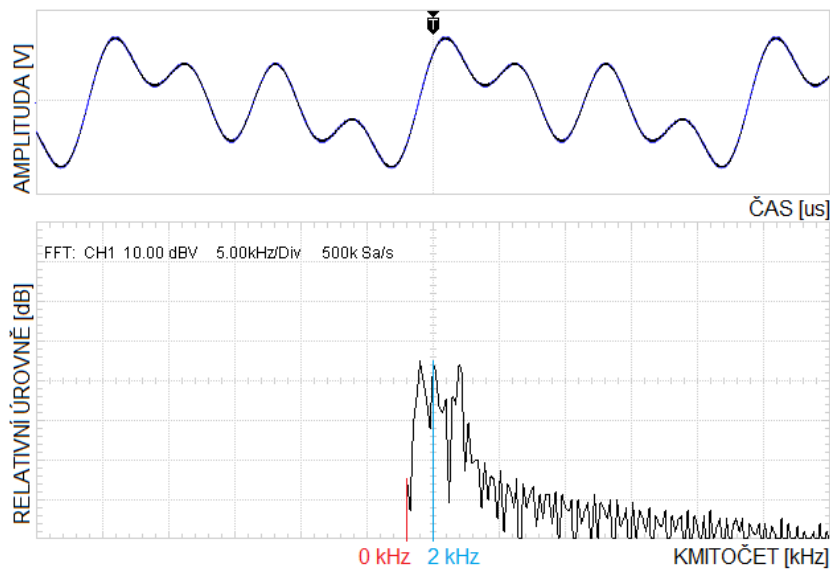
Pro úplnost si ještě ukážeme grafické znázornění fázového spektra na obrázku 3.13. Matlab nativně poskytuje výstup v úhlové míře (radiánech), pro vyjádření ve stupních je nutné prvky výstupní posloupnosti vynásobit konstantou $\frac{180}{\pi}$.



Obrázek 3.12: Detail posledních 20 prvků amplitudového spektra



Obrázek 3.13: Fázové spektrum



Obrázek 3.14: Amplitudové spektrum získané z osciloskopu Rigol MSO1104Z-S

Podobný signál jsme nechali generovat signálním generátorem Rigol DG1022Z, který dokáže generovat sinusový signál složený ze základního kmitočtu a jeho harmonických kmitočtů. Za základní kmitočet byl tomto případě zvolen kmitočet 10 kHz a opět je v signálu přítomna druhá a čtvrtá harmonická frekvence. Amplituda signálu byla zvolena 10 V. Analýza probíhala na číslicovém paměťovém osciloskopu pro smíšený signál Rigol MSO1104Z-S.

Výstup amplitudového spektra z osciloskopu na obrázku 3.14 znázorňuje harmonické kmitočty užitečného vstupního signálu prakticky shodně, stejně jako spektrogram z jazyka Matlab na obrázku 3.11. Je zde však viditelný rozdíl v odstupu signálu vůči šumu, který bude vždy větší u spektrogramu z jazyka Matlab, neboť se jedná o matematickou simulaci, zatímco kmitočtová analýza reálným přístrojem - číslicovým osciloskopem analyzuje signál pocházející z generátoru, který ačkoliv generuje signál 14 bitovou přímou digitální syntézou (DDS - direct digital synthesis), má rovněž parazitní vlastnosti, které signál ovlivní.

Je nutné zmínit, že na číslicových paměťových osciloskopech se u periodických průběhů typicky jedná o tzv. průměrované spektrum, neboť je získáno z více navzorkovaných period měřeného signálu. Tato vlastnost je dána zejména možností analyzovat paměť vzorků nebo obrazovku osciloskopu a rovněž širokým vstupním pásmem osciloskopu, čímž se téměř vždy analyzuje více jak jedna perioda vstupního signálu.

Kmitočtová analýza obecně předpokládá, že se vstupní posloupnost chová „rozumně“, jedním z předpokladů je tzv. koherentní vzorkování, které je splněno tehdy, je-li poměr vzorkovacího kmitočtu ke kmitočtům vstupního signálu soudělný. Abychom však byli schopni zpracovávat obecně stochastické (náhodné) signály, byly objeveny tzv. okénkové funkce, které provádí korekci vstupní posloupnosti.

3.2.5 Okénkové funkce

V praxi je velmi běžné, že vstup obecně diskrétní fourierovy transformace nebude periodický signál, ale rovněž, obzvláště vyžaduje-li algoritmus rychlé fourierovy transformace nejen sudý počet vzorků vstupní posloupnosti, ale i počet vzorků rovnající se mocnině dvou, se dostaneme nezřídka do situace, kdy nebudeme schopni splnit koherentní vzorkování (kdy nebude celkový průběh signálu složený z analyzovaných úseků po částech spojitý). V případě, kdy budou ve vstupním průběhu strmé přechody způsobené nekoherentním vzorkováním, bude spektrum bohaté, obdobně jako u obdélníkového signálu, avšak můžeme přijít o složky spektra užitečného signálu, neboť tyto nemusejí být rozlišitelné od složek falešných způsobených právě nekoherentním vzorkováním. Výstup transformace v případě nekoherentního vzorkování však navíc nemusí být u stejného signálu shodný, bude-li přítomen posun signálu v čase, neboť posun v čase nám ovlivní, kde bude ke strmým přechodům u vstupního signálu docházet. Při nekoherentním vzorkování dochází k tzv. rozmazávání kmitočtového spektra. Tyto obtíže řeší tzv. okénkové funkce, které realizují výběr vstupní posloupnosti signálu a navíc upravují jeho amplitudu. Provádí nám tzv. korekci. Lze tedy říci, že okénkové funkce nám do jisté míry přemění neperiodický průběh na průběh periodický. Okénkové funkce však mají nevýhodu, že obdobně jako jiné funkce, mají i tyto své vlastní spektrum, které se, v případě aplikace okénkové funkce na signál (operací násobením), přičte ke spektru analyzovaného signálu (výsledné spektrum je konvolucí samostatných spekter signálu a okénkové funkce). Oproti přístupu prosté analýzy nekoherentně navzorkovaného signálu sice získáme navíc určité harmonické kmitočty (spektrum) vlivem aplikace okénkové funkce, avšak toto spektrum je deterministické (stálé) a získáme detailnější pohled na složky spektra užitečného signálu a navíc v případě vyskytujícího se posunu v čase shodného signálu i celkové shodné spektrum. V tabulce 3.6 si nyní uvedeme nejčastěji používané okénkové funkce a jejich matematické popisy a v tabulce 3.7 jejich parametry.

Tabulka 3.6: Matematický popis vybraných okénkových funkcí

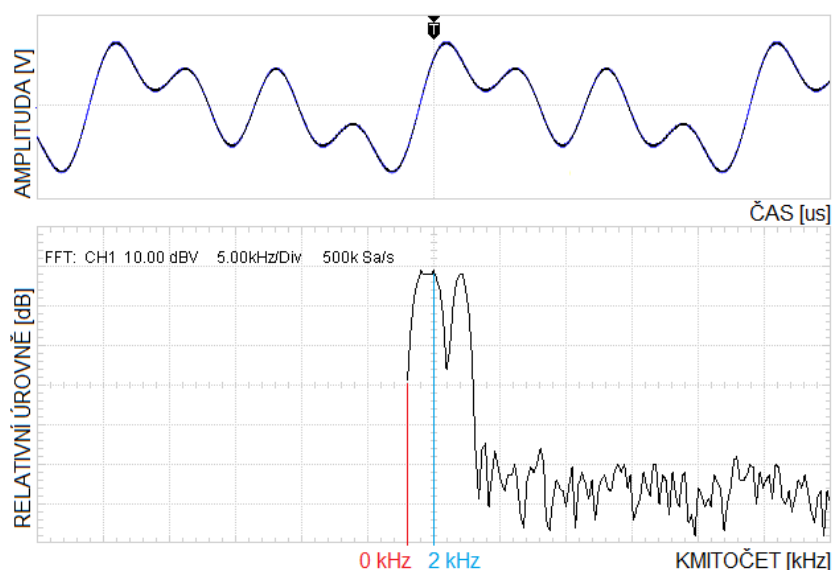
Typ okénka	Matematický zápis
Hann	$w(k) = 0,5 - 0,5 \cdot \cos\left(2 \cdot \pi \cdot \frac{k}{N}\right)$
Hamming	$w(k) = 0,54 - 0,46 \cdot \cos\left(2 \cdot \pi \cdot \frac{k}{N}\right)$
Blackman-Harris	$w(k) = 0,422323 - 0,49755 \cdot \cos\left(2 \cdot \pi \cdot \frac{k}{N}\right) + 0,07922 \cdot \cos\left(4 \cdot \pi \cdot \frac{k}{N}\right)$
Flat-top	$w(k) = 1 - 1,93 \cdot \cos\left(2 \cdot \pi \cdot \frac{k}{N}\right) + 1,29 \cdot \cos\left(4 \cdot \pi \cdot \frac{k}{N}\right) - 0,388 \cdot \cos\left(6 \cdot \pi \cdot \frac{k}{N}\right) + 0,028 \cdot \cos\left(8 \cdot \pi \cdot \frac{k}{N}\right)$

Tabulka 3.7: Parametry vybraných okénkových funkcí

Typ okénka	Potlačení hlavního laloku [dB]	ENBW
Hann	32	1,5
Hamming	43	1,37
Blackman-Harris	58	2,01
Flat-top	96	3,77

Okénkových funkcí je mnoho typů, obecně jsou nejčastěji založeny na goniometrických funkcích (nejčastěji cosinus, méně často sinus), jsou však také typy okénkových funkcí založených na polynomech a aproximacích křivek (například B-spline). Kritériem pro výběr okénkové funkce je zejména parametr potlačení hlavního laloku, kde potlačení hlavního laloku signálu by mělo být minimálně o 10 dB až 20 dB menší, než je předpokládaný šum soustavy (digitalizátoru nebo zvukové karty). Při výběru okénkové funkce je nutné brát v úvahu také náročnost okénkové funkce na výpočet. Standardně užívanou okénkovou funkcí je Blackmann-Harris.

Číslicové osciloskopy zpravidla poskytují možnost zapnout některou z okénkových funkcí při kmitočtové analýze. Obrázek 3.15 ukazuje konkrétně zapnutou okénkovou funkci typu Blackman nad stejným analyzovaným průběhem jako na obrázku 3.14. Již na první pohled je zřejmé zvětšení odstupu užitečného signálu od šumu. To je způsobeno vlastností okénkové funkce, konkrétně parametrem potlačení hlavního laloku.



Obrázek 3.15: Amplitudové spektrum s aplikovanou okénkovou funkcí Blackman

3.2.6 Zvukový formát WAV

Aby bylo možné zvukové signály analyzovat a zpracovávat, je nutné uchovávat zvukový záznam v některém z datových formátů. Patrně nejznámějším a nejjednodušším formátem je formát WAV (RIFF). Soubor WAV je uložen

úrovních (absolutní hodnotou -32768 až 32767). Záznam zvuku musí obsahovat konzistentní rámce dat, tedy není například možné, aby poslední vzorek zvukových dat obsahoval pouze jeden zaznamenaný zvukový kanál, bude-li jinak záznam vícekanálový. Jelikož čtení binárních souborů probíhá vždy po bytech a záznam zvuku v souboru je proveden v tzv. rámcích dat (obsahujících vzorky pro oba zvukové kanály, nejprve pro levý a poté pro pravý zvukový kanál), je třeba číst korektní počet byte, aby byly korektně čteny oba kanály.

3.3 Číslíkové zpracování obrazu

Číslíkového zpracování obrazu se v případě elektronického kukátka může s výhodou použít k detekci pohybu. Obrazové signály je možné dělit na konstantní v čase (snímky) a proměnné v čase (videoklipy). Obecně jsou obrazové signály tzv. vícedimenzionálními signály.

Minimálně je obrazová informace ukryta v trojdimenzionálním signálu, budeme-li uvažovat černobílý statický snímek, kde dvě dimenze představují souřadnice obrazových bodů a třetí pak barvonosnou složku signálu, v tomto případě monochromatickou, tj. jednokanálovou.

Maximálně je obrazová informace zaznamenána do sedmidimenzionálního signálu, budeme-li uvažovat videoklip s oddělenými jednotlivými barevnými složkami, jasovým kanálem (tzv. kanál alfa). První dimenze představuje čas, další dvě dimenze představují rozměry obrázku a zbylými dimenzemi jsou jednotlivé barevné složky (RGB - červená, zelená, modrá) a jasová složka.

Obecně je možné detekci pohybu, pomocí obrazových dat, jednoduše realizovat prostým porovnáváním dvou nasnímaných obrázků mezi sebou (pixel po pixelu), významným problémem však při tomto zpracování obrazových dat je šum. Je nutné zmínit, že úroveň šumu je dána nejen výrobní technologií obrazového sensoru, ale také velikostí jeho jednotlivých buněk a v neposlední řadě svojí roli hraje také následné zpracování obrazových dat do formátu komprimovaného souboru, kde další šum přidá komprese. Z tohoto důvodu má profesionální obrazová technika nejen kvalitní optiku, ale rovněž rozměrnější obrazové sensory (zpravidla i každý na jednu barevnou složku) a ukládání probíhá nekomprimovaně do souboru formátu RAW (surová data z obrazového sensoru).

Šum je však možné částečně redukovat Gaussovým filtrem, který je možné aplikovat konvolucí, avšak v číslíkovém zpracování signálu (prostor s diskretním rozdělením) se zpravidla aplikuje násobením. Gaussův filtr pracuje na principu dvourozměrné varianty Gaussova normálního rozdělení a je definován vztahem (3.15).

$$G(x, y) = \frac{1}{2 \cdot \pi \cdot \sigma^2} \cdot e^{-\frac{x^2+y^2}{2 \cdot \sigma^2}} \quad (3.15)$$

Filtr má jeden nastavitelný parametr σ , což je standardní odchylka rozdělení, koeficienty x a y jsou souřadnice obrazových bodů. Parametr σ se nastavuje v závislosti na úrovni šumu, je tedy nutné filtr před použitím

bitů) nemají tedy v naší aplikaci přílišný význam a proto se na ně nebudeme v popisu zaměřovat. Nutno poznamenat, že formát BMP je ukládán rovněž ve formátu little-endian, jako datový formát WAV, tj. i zde je jako první v posloupnosti bytů uložen byte s nejnižším významem (LSB). Vyjímkou tvoří opět řetězce, které jsou kódovány ve standardním 1-byte ASCII kódu a jsou uloženy v čitelné podobě. Zajímavostí formátu je, že ukládání probíhá od levého dolního obrazového bodu po pravý horní obrazový bod. Obraz je tedy ukládán zleva doprava a odspodu nahoru, přestože například obrazový standard PAL vykresluje obrazové body od levého horního bodu po pravý dolní. Tato vlastnost formátu BMP pramení z historie grafických software, kdy například tento přístup dodnes implementuje program Autodesk AutoCAD nebo i dříve Cadsoft Eagle.

Tabulka 3.9: Popis hlavičky obrazového souboru BMP

Offset	Délka	Příklad HEX	Hodnota	Popis
0	2 byte	42 4D	BM	Formát souboru BMP.
2	4 byte	00 04 B0 00	307200 byte	Velikost souboru. ⁽¹⁾
6	2 byte	00 00	0	Vyhrazeno. ⁽²⁾
8	2 byte	00 00	0	Vyhrazeno. ⁽²⁾
10	4 byte	36 00 00 00	54 byte	Offset dat. ⁽³⁾
14	4 byte	28 00 00 00	40 byte	⁽⁴⁾
18	4 byte	20 03 00 00	800 pixelů	Šířka obrazu.
22	4 byte	58 02 00 00	600 pixelů	Výška obrazu.
26	2 byte	01 00	1	Počet bit. rovin. ⁽⁵⁾
28	2 byte	18 00	24 bitů	Počet bitů na pixel.
30	4 byte	00 00 00 00	0	Komprimace. ⁽⁶⁾
34	4 byte	00 00 00 00	0	Velikost obrazu. ⁽¹⁾
38	4 byte	00 00 00 00	0	⁽⁷⁾
42	4 byte	00 00 00 00	0	⁽⁷⁾
46	4 byte	00 00 00 00	0	Použité barvy. ⁽⁸⁾
50	4 byte	00 00 00 00	0	Důležité barvy. ⁽⁹⁾
54		–	–	Obrazová data.

⁽¹⁾ Některé aplikace ukládají soubory s touto položkou nabývací hodnoty 0. Celková velikost souboru se pak musí zjistit buďto z informací poskytovaných souborovým systémem nebo je nutné soubor projít po bytech do tzv. příznaku EOF (end of file) a počet bytů počítat. Velikost obrazu je rovněž nepovinnou položkou a poté je velikost obrazu možno vypočítat buďto součinem počtu řádkových pixelů s počtem sloupcových pixelů a navíc bitovou hloubkou, nebo je možné z celkové velikosti souboru odečíst délku hlavičky souboru, která je v případě 24 bitových bitmap vždy 54 bytů.

⁽²⁾ Aktuální verze souboru BMP tyto položky nepoužívají. Byly navrženy, při tvorbě formátu, jako rezervní pro budoucí použití v dalších verzích souborů. Pro korektní čtení a zápis běžně dostupnými programy musejí tyto položky nabývat hodnot 0.

- (3) Offset obrazových dat od začátku souboru, v případě 24 bitových bitmap vždy 54 byte.
- (4) Počet bytů do začátku obrazových dat od pozice této položky, v případě 24 bitových bitmap vždy 40 byte. Tato položka a položka na offsetu 10 byte poskytuje do jisté míry datovou redundaci, která je zde přítomna patrně z důvodu možnosti použití dvou odlišných datových struktur v rozhraní WinAPI.
- (5) Počet bitových rovin je položka přítomná z historických důvodů, která měla sloužit pro nastavení bitových rovin na výstupním zařízení. Prakticky vždy nabývá hodnoty 1.
- (6) Metoda komprimace je položka, která může nabývat 3 rozdílných hodnot, neboť i formát BMP může mít komprimaci obrazu, která se však používá zřídka. 0 označuje obrazová data bez komprese, 1 označuje komprimaci metodou RLE8, a hodnota 2 je příznakem komprimace metodou RLE4.
- (7) Položka na pozici 38 udává horizontální počet pixelů na metr a položka na pozici 42 udává vertikální počet pixelů na metr. Obě položky jsou obdobou DPI (dots per inch - počet bodů na palec). Protože je však formát BMP nezávislý na zařízení, tak se tyto položky většinou nepoužívají.
- (8) Celkový počet použitých barev v obrazu. Jedná se převážně o položku používanou při optimalizacích, neboť je většinou nastavena na hodnotu 0 a počet použitých barev se v takovém případě zjišťuje z bitové hloubky obrazu.
- (9) Počet barev, které jsou důležité pro vykreslení bitmapy. V případě nastavení na hodnotu 0 platí, že všechny barvy jsou důležité. Položka se používá při vykreslování na zařízeních s omezeným počtem barev (starší grafické karty a plottery).

3.4 Návrh software

Hardware popisovaný v předchozí podkapitole je převážně založen na číslicových obvodech, navíc se jedná o velmi rozsáhlý a komplexní systém, a aby takovéto komplexní zařízení poskytovalo požadovanou funkcionalitu, je nutné jej vybavit softwarovým vybavením. Jelikož se jedná o malý jednodeskový počítač, pro který existují vyvíjené operační systémy, jejichž použití nám může usnadnit práci s některými hardwarovými komponentami (kamera, displej, zvuková karta, wi-fi adaptér, vstupně-výstupní piny GPIO, apod.), ale může nám zjednodušit i softwarovou realizaci, neboť součástí celkového řešení elektronického kukačka je i návrh softwarového vybavení pro ukládání obrazových dat a možnost jejich vzdálené správy webovou aplikací, popíšeme si v této kapitole nejprve operační systémy dostupné pro počítač Raspberry PI. Dále si představíme možnost řešení webové aplikace pro správu dat včetně doménového modelu, neboť pro správu a ukládání dat je nejvhodnější některý z databázových systémů. Představíme si i možnou podobu nasazení výsledného softwarového řešení na zařízení.

3.4.1 Operační systém

Na počítačích Raspberry PI lze provozovat několik operačních systémů, které jsou z drtivé většiny modifikovanými verzemi standardních operačních systémů typu Linux (Unix), případně je možné se setkat s operačními systémy, které jsou pouze přeloženy pro architekturu ARM.

Patrně nejznámějšími zástupci modifikovaných operačních systémů jsou distribuce Raspbian, což je modifikace distribuce Debian a Pidora, což je modifikace distribuce Fedora. V současné době již není oficiální nadací Raspberry nabízena distribuce Pidora ke stažení, ale pouze distribuce Raspbian.

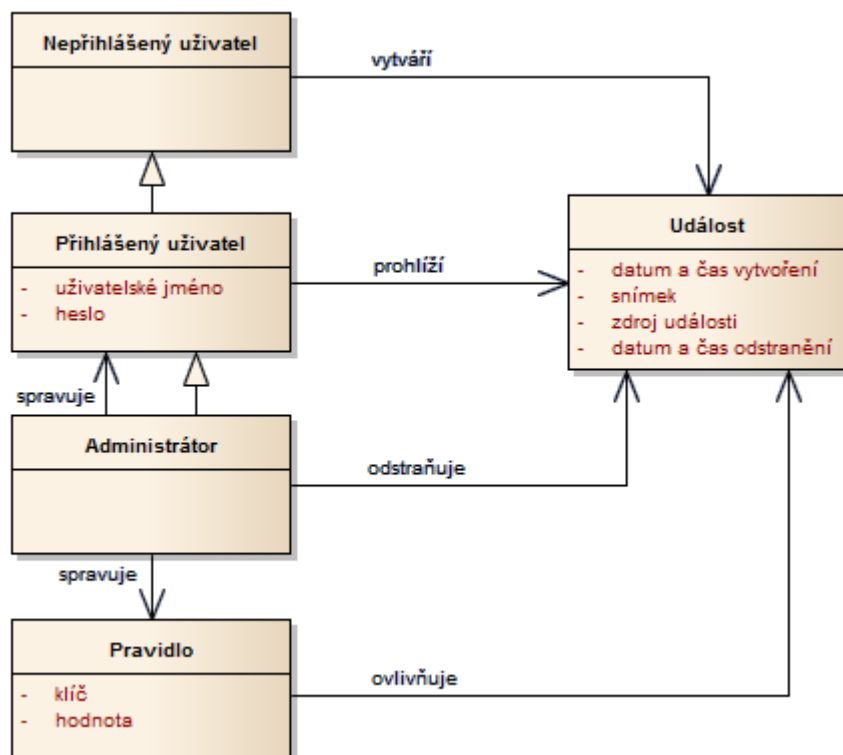
Typickým operačním systémem, který je pouze přeložen pro architekturu ARM je také distribuce Debian, neboť komunita vyvíjející Debian tento systém překládá pro různé architektury, mezi nimiž nechybí nejen typické architektury i386, amd64, ale právě i armhf, pod kterou spadá i počítač Raspberry PI.

Pro návrh se jeví jako nejvýhodnější použít operační systém Raspbian, neboť je stále vyvíjen, velmi dobře zdokumentován, v současnosti má podporu ovládání vstupně-výstupních bran GPIO a kamery již v základním systému, a v neposlední řadě na něm lze provozovat bez problémů širokou škálu standardních aplikací (například webový server Apache s podporou skriptovacího jazyka PHP nebo databázový server MySQL). Tyto aplikace je však nutné instalovat ve verzi armhf, nikoliv i386 nebo amd64, jak bývá zvykem na počítačích řady PC. Pro pohodlnou instalaci je Raspbian standardně vybaven balíčkovacím systémem Advanced Packaging Tool, pomocí kterého je možné instalovat požadovaný software včetně závislých balíčků, čímž odpadá nutnost studování závislostí mezi balíčky, je však nutné pro instalaci připojit počítač Raspberry PI k síti Internet.

Pro jazyk PHP je k dispozici rovněž několik ucelených frameworků, například Laravel, Nette, Symfony nebo Zend. Pro samostatnou práci s databází existuje framework Doctrine.

Hlavními rozdíly mezi webovými aplikacemi vytvořenými v jazyce Java a PHP jsou kromě použitého jazyka a komponent také nároky na hardware a software. Aplikace vyvíjené v jazyce Java jsou sice bezpečnější, například i kvůli tomu, že jazyk Java je oproti jazyku PHP silně typovaným jazykem, ale mají i větší nároky na paměť a vyžadují specifický server (například Apache Tomcat). Virtuální servery v síti internet poskytované k bezplatnému umístění webových aplikací navíc mají standardně k dispozici jen webový server Apache 2 a podobné, tedy podporu poskytují pouze jazyku PHP. Z těchto důvodů je jazyk PHP více rozšířen.

V případě elektronického kukátka můžeme software pro správu pořízených dat rovněž koncipovat jako webovou aplikaci. S výhodou můžeme použít jazyk PHP a frameworku Nette, který je vyvíjen českou komunitou vedenou Davidem Grudlem a má tedy dokumentaci v českém jazyce. Protože je v operačním systému Raspbian dostupné rovněž grafické prostředí a webový prohlížeč, můžeme software pro správu pořízených dat zobrazovat rovněž na displeji zařízení, avšak vzhledem k absenci klávesnice jen v omezené podobě.

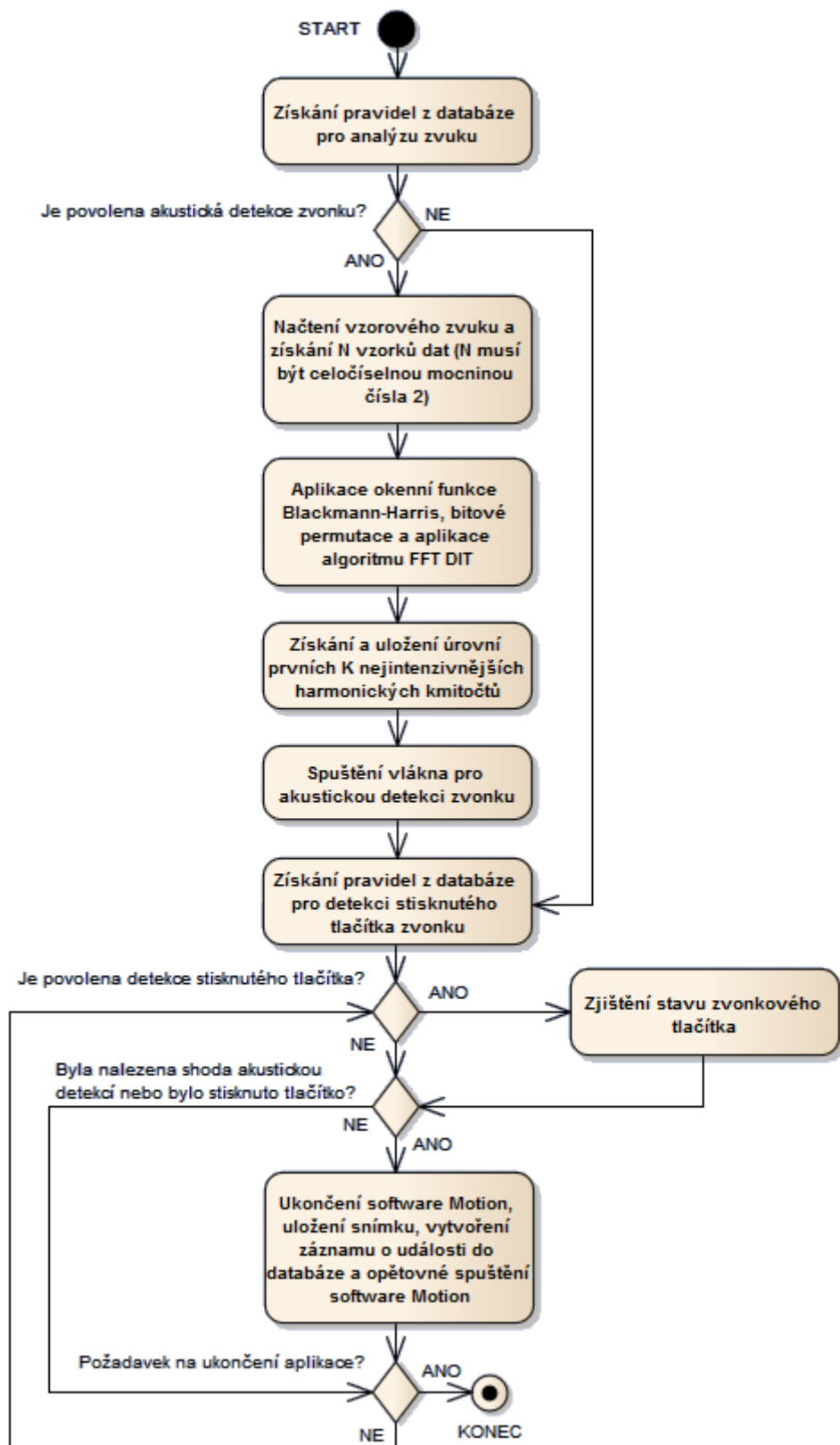


Obrázek 3.16: Diagram doménového modelu

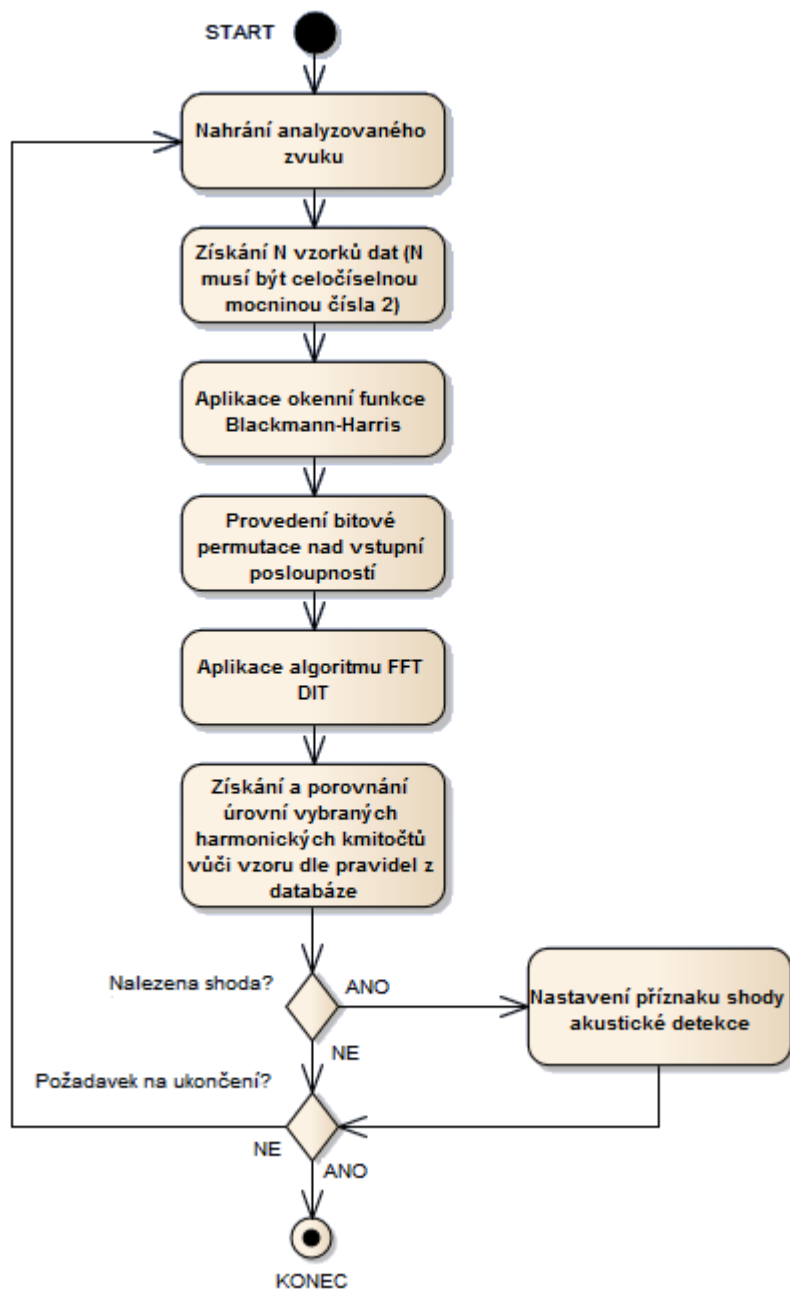
Protože aplikace bude pracovat s databází, je vhodné nadefinovat doménový model. V doménovém modelu, na obrázku 3.16, jsou znázorněny možné databázové entity (tabulky) a jejich atributy. Entita Nepřihlášený uživatel

■ 3.4.4 Software pro detekci podobnosti zvuků a detekci zvonkového napětí

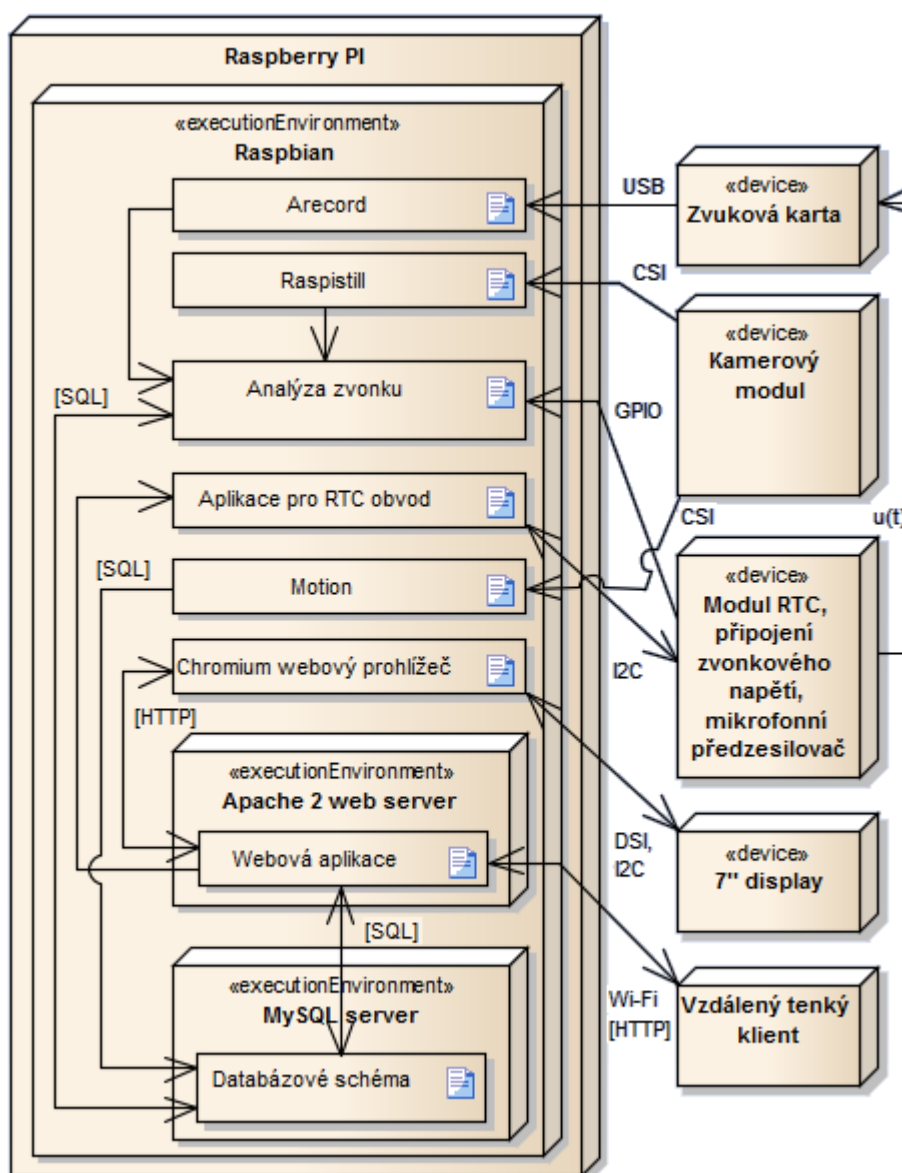
Abychom mohli detekovat podobnost dvou zvuků zvonku - vzorového a analyzovaného a detekovat zvonkové napětí, je nutné napsat vlastní software. Z důvodu vyšších nároků na rychlost daného řešení je vhodné aplikaci realizovat v kompilovaném jazyce C a pro analýzu podobnosti zvuků je vhodné rezervovat navíc jedno vlákno. V ideálním případě dojde k vyhrazení jednoho jádra procesoru na jedno vlákno aplikace. V případě vlákna nenáročného na výkon nebo nepříznivé situaci (například více vláken, než jader procesoru nebo více spuštěných programů najednou) bude docházet ke střídání jádra procesoru nad vlákny. Aplikace je popsán vývojovým diagramem 3.17, který znázorňuje algoritmicky hlavní vlákno programu (v jazyce funkce `main()`) a vývojovým diagramem 3.18, který znázorňuje algoritmus vlákna vyhodnocující akustickou podobnost signálů. Vlákno detekující akustickou podobu narozdíl od začátku hlavního programu neprochází celé kmitočtové spektrum, ale prochází pouze významné kmitočty určené ze vzorového signálu. Tento postup je zvolen s ohledem na menší časovou náročnost. Na začátku hlavního programu (mimo programovou smyčku) oproti tomu procházíme celé spektrum a nalezneme K nejintenzivnějších kmitočtů. Mimo hlavní programovou smyčku a smyčku vlákna si můžeme dovést procházet celé spektrum, neboť tato část je po startu programu prováděna pouze jednou a nezáleží příliš na časové náročnosti tohoto algoritmu. Vzorový signál (zvukový soubor) se vytvoří prostřednictvím webové aplikace. Před vytvářením vzorového signálu však dojde k dočasnému vypnutí programu, neboť obdobně jako v případě kamery, ani v případě zvukové karty není možné k danému zařízení přistupovat více procesy najednou.



Obrázek 3.17: Vývojový diagram programu pro vytváření událostí na základě akustické detekce zvonku a zvonkového tlačítka



Obrázek 3.18: Vývojový diagram vlákna pro detekci akustické podobnosti analyzovaného signálu vůči signálu vzorovému



Obrázek 3.19: Diagram nasazení znázorňující hardware i software

3.5 Instalace

V této kapitole si popíšeme postup pro instalaci celého systému krok za krokem nad čistou instalací operačního systému Raspbian ve verzi Stretch s datem sestavení 9.10.2018 nad počítačem Raspberry PI 3 Model B+ V1.2. Popsaný postup instalace se může od reálného postupu odlišovat, například v závislosti na verzi systému Raspbian, na již nainstalovaných balíčcích, použité desce počítače Raspberry PI, apod. Popis instalace také neuvažuje možné konflikty při instalaci balíčků, vzniklé zejména dřívější instalací některých komponent, které je třeba vyřešit individuálně. Je nutné také podotknout, že proces

instalace uvažuje provádění jednotlivých kroků s právy správce systému (root), do jehož uživatelského účtu se přepneme příkazem: `sudo su` a že po dobu instalace balíčků by počítač měl být připojen do sítě internet.

Před instalací jednotlivých částí systému elektronického kukátka je vhodné aktualizovat závislosti balíčků v systému Raspbian pomocí příkazu: `apt-get update`.

■ 3.5.1 Instalace systému Raspbian

Nejjednodušší možností instalace systému Raspbian je pořízení prázdné paměťové karty typu microSDHC o kapacitě alespoň 8 GB. Je vhodné přezkontrolovat její správné naformátování na jediný logický oddíl o souborovém systému FAT32. Případné přeformátování na tento souborový systém je možné provést utilitou SD Card Formatter od SD Association, která je volně dostupná na adrese: https://www.sdcard.org/downloads/formatter_4/.

Na prázdnou paměťovou kartu je nutné zapsat soubor s obrazem disku systému Raspbian. Aktuální obraz v komprimovaném souboru ZIP je možné získat z adresy: https://downloads.raspberrypi.org/raspbian_latest.

Obsahem archivu ZIP je soubor obrazu disku formátu (*.img), který je nutné vyextrahovat a zapsat na kartu microSDHC. K zápisu souboru je možné použít několik utilit a příkazů. Pod operačním systémem Windows je možné použít volně dostupný program Win32 Disk Imager, jehož domovská stránka je na adrese: <https://sourceforge.net/projects/win32diskimager/>.

Vstupem programu je soubor s obrazem disku a písmeno fyzického disku se kterými chceme pracovat. Následně program obsahuje několik tlačítek, mezi důležitá tlačítka patří tlačítko: **Read** pro načtení dat z fyzického disku do souboru obrazu a tlačítko: **Write** pro zápis dat z obrazu disku na fyzický disk.

■ 3.5.2 Povolení SSH

Nejvhodnější možností instalace balíčků a přístupu na mikropočítač Raspberry PI je vzdálený přístup přes protokol SSH (secure shell). Jedná se o protokol zabezpečující autentizaci klienta (v tomto případě PC) vůči serveru (v tomto případě Raspberry PI) a jejich následnou komunikaci s využitím asymetrické kryptografie, tedy s využitím privátních a veřejných klíčů.

Od verze Stretch již není standardně povolen vzdálený přístup k Raspberry PI přes protokol SSH, proto je nutné jej po instalaci systému Raspbian dodatečně povolit. Toto je možné provést dvěma možnými způsoby:

1. Vytvořením prázdného souboru s názvem `ssh` na logickém oddílu boot systémové SD karty. Toto je možné provést i na operačním systému Windows, neboť logický oddíl boot je souborový systém typu FAT32.
2. Připojením standardních vstupních a výstupních periférií (monitor s rozhraním HDMI, klávesnice a myši s rozhraním USB) k Raspberry PI a spuštěním jako samotného počítače a zadáním příkazu `sudo raspi-config` v příkazové řádce, kde v sekci **Interfacing Options** a podsekcí **SSH**

Je nutné zmínit, že pro použití software Motion, je vyžadován ovladač pro kamerový modul. O přítomnosti ovladače kamerového modulu se můžeme přesvědčit zadáním příkazu: `v4l2-ctl --info`.

Pokud výstupem příkazu bude: `Failed to open /dev/video0 : No such file or directory`, ovladač není přítomen a je nutné jej zavést příkazem: `sudo modprobe bcm2835-v4l2`.

Opětovným zadáním příkazu: `v4l2-ctl --info` by měla být již zobrazena informace o ovladači kamerového modulu.

Nutným krokem je korektní nastavení software Motion, které se provádí v souboru: `/etc/motion/motion.conf`. Zejména je nutné vyhledat atributy:

- `target_dir`, kde je nutné nastavit hodnotu na:
`/var/www/Kukatko/Nette-2.3.5/sandbox/www/motion`.
- `stream_port`, který musí být nastavený na: 8081.
- `stream_localhost`, kde je nutné změnit hodnotu: `on` na hodnotu: `off` a tím zrušit omezení poskytování živého videa pouze na počítači Raspberry PI, ale v rámci celé sítě, do které je Raspberry PI připojeno.
- `sql_log_picture` má být nastaven na hodnotu: `on`.
- `sql_log_snapshot` má být nastaven na hodnotu: `off`.
- `sql_query` je nutné doplnit o příkaz jazyka SQL:
`INSERT INTO Event(created_at, path_to_image, action_from_user, action_from_ring_button, action_from_motion, action_from_ring_sound)`
`values('%Y-%m-%d %T', '%f', '0', '0', '1', '0')`, který vytvoří záznam do databázové tabulky Event (Událost) po pořízení snímku.
- `database_type` a nastavit jej na hodnotu `mysql`.
- `database_dbname` a přiřadit hodnotu `viewer`.
- `database_host` a nastavit hodnotu `localhost`.
- `database_user`, kde v našem případě nastavíme hodnotu `root`.
- `database_password`, který má být nastaven na hodnotu `pass`.
- `database_port`, kde je nutné nastavit číslo portu 3306.
- `database_busy_timeout`, kde je vhodné vložit hodnotu 500.

■ 3.5.5 Webový server

Software elektronického kukátka je tvořen zčásti webovou aplikací, je tedy nutné na počítač Raspberry PI nainstalovat webový server. Vhodným webovým serverem je server Apache 2, který lze nainstalovat příkazem: `apt-get install apache2`.

ze skriptovacího jazyka PHP: `apt-get install libapache2-mod-php`.

Dalším vyžadovaným modulem pro skriptovací jazyk PHP je modul pro MySQL, který je nutné doinstalovat příkazem: `apt-get install php-mysql`.

■ 3.5.7 Databázový server

Elektronické kukátko ukládá záznamy o datu a času a fotografie do databáze, je nutné nainstalovat také databázový server. Výchozím databázovým serverem pro Nette Framework je server MySQL, který se nainstaluje příkazem: `apt-get install mysql-server`.

■ 3.5.8 phpMyAdmin

Ačkoliv Nette Framework poskytuje rozhraní pro správu databázového serveru, tzv. Adminer, standardně dostupný z webového prohlížeče nad adresou: `localhost/adminer`, kdy na adrese `localhost` je hlavní stránka webové aplikace, toto rozhraní je dostupné pouze při lokálním přístupu k aplikaci v rámci serveru, nikoliv v případě přístupu vzdáleného. Pro budoucí možné změny tabulek v databázi v uživatelsky přívětivém prostředí velmi podobném rozhraní Adminer, je však možné doinstalovat balíček `phpMyAdmin` příkazem: `apt-get install phpmyadmin`. Při instalaci probíhá rovněž konfigurace, kterou je možné kdykoliv poté změnit příkazem: `dpkg-reconfigure phpmyadmin` z příkazové řádky systému Raspbian. V našem případě je však konfigurace následující: Připojovací metoda pro MySQL databázi nad `PhpMyAdmin` je `TCP/IP`, aby bylo možné se připojovat i vzdáleně. Hostname databázového serveru MySQL nad `PhpMyAdmin` je `localhost`. Číslo portu služeb MySQL databázového serveru je `3306`. Název databáze MySQL pro `PhpMyAdmin` je `root`. Uživatelské jméno MySQL pro `PhpMyAdmin` je `root@localhost`. Aplikační heslo MySQL pro `PhpMyAdmin` je `pass` a v následujícím kroku jej zadáme pro potvrzení znovu. Jméno administrátora databáze je `root`. Automaticky budeme chtít provádět rekonfiguraci webového serveru `apache2`.

Aby nebylo nutné ručně vkládat počáteční hodnoty do databáze, byl napsán inicializační skript databáze `skript.sql`, který je na doprovodném médiu. Tento skript zkopírujeme již zmíněným programem WinSCP do složky `/home/pi`. Inicializační skript spustíme příkazem:

```
mysql < /home/pi/skript.sql -u root -p a na vyzvání zadáme heslo: pass.
```

■ 3.5.9 I2CTools

Pro přístup k časovači DS1307 na sběrnici I²C je třeba nainstalovat balíček s názvem `I2CTools` příkazem: `apt-get install i2c-tools`.

Pro korektní funkci sběrnice je také třeba upravit soubor: `/etc/modprobe.d/raspi-blacklist.conf`, kde je nutné vložím znaku `#` na začátek řádky vytvořit komentář z řádky s informací `blacklist i2c-bcm2708`. Případně je také možné tuto řádku vymazat. Rovněž je také možné, v závislosti na verzi

V prostředí LXDE-pi, nad implicitním uživatelem pi, má automatizované spuštění aplikací na starosti skript: `/home/pi/.config/lxsession/LXDE-pi/autostart`.

Příkaz: `sudo nano /home/pi/.config/lxsession/LXDE-pi/autostart` otevře soubor se skriptem pro úpravy. Na konec tohoto souboru se přidávají řádky s názvy aplikací, které mají být po spuštění prostředí spuštěny a parametry, které se aplikacím předávají. V našem případě se bude jednat o aplikaci webového prohlížeče. Za webový prohlížeč jsme zvolili Chromium, z důvodu standardní dostupnosti v grafickém prostředí LXDE-pi. Do souboru `autostart` tedy přidáme řádek: `@chromium-browser --kiosk localhost`. Tento přidaný řádek nám říká, že se má spustit program `chromium-browser`, což je spustitelný program prohlížeče Chromium s parametrem `--kiosk`, který určuje, že se prohlížeč spustí na celou obrazovku podobně, jako u informačních panelů, kterým se také někdy říká informační kiosky, odtud název parametru, a posledním parametrem je parametr `localhost`, který aplikaci předá informaci, jaká webová stránka se má po spuštění aplikace zobrazit, v našem případě lokální webová aplikace. Po restartování zařízení by se na displeji již měla zobrazit webová aplikace kukátka.

■ 3.5.12 Přístupový bod Wi-Fi

Abychom mohli přistupovat k elektronickému kukátku bezdrátově, je vhodné nakonfigurovat vestavěný Wi-Fi adaptér jako přístupový bod.

K tomuto účelu jsou zapotřebí dva programy `hostapd` a `dnsmasq`. Program `hostapd` má na starosti vytvoření bezdrátové sítě a `dnsmasq` kombinuje DHCP (dynamic host configuration protocol) a DNS (domain name service) server. DHCP server je nezbytný, neboť ten umožňuje dynamicky přiřazovat připojovaným zařízením IP adresy, bez této možnosti by připojení ke kukátku vyžadovalo od uživatele znalost číslování sítí a konfigurace IP adres.

Programy `hostapd` a `dnsmasq` nainstalujeme příkazy: `apt-get install hostapd` a `apt-get install dnsmasq`.

Následně je nutné upravit soubor `dhcpcd.conf` nacházející se v adresáři `/etc` příkazem: `sudo nano /etc/dhcpcd.conf` a přidat na konec souboru následující záznam:

```
interface wlan0
static ip_address=192.168.1.1/24
static routers=192.168.1.1
static domain_name_servers=192.168.1.1
```

Rovněž je nutné upravit soubor `dnsmasq.conf` nacházející se v adresáři `/etc` příkazem: `sudo nano /etc/dnsmasq.conf` a jeho obsah změnit na následující záznam:

```
interface=wlan0
domain-needed
bogus-priv
dhcp-range=192.168.1.10,192.168.1.100,255.255.255.0,12h
```

Abychom byli schopní se k síti Wi-Fi připojit například pomocí přenosného počítače, je nutné vytvořit nový soubor `hostapd.conf` do adresáře

/etc/hostapd příkazem: `sudo nano /etc/hostapd/hostapd.conf` a do něj zapsat následující údaje:

```
interface=wlan0
driver=nl80211
ssid=SSID_SITE
hw_mode=g
channel=7
wmm_enabled=0
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=HESLO_SITE
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

O tomto souboru je nutné programu `hostapd` předat cestu, toto učiníme editací souboru `/etc/default/hostapd` pomocí již zmíněného editoru `nano` a úpravou parametru `DAEMON_CONF` na tvar:

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

Abychom se mohli připojovat ke kukátku také již zmíněným názvem namísto pouze IP adresy, upravíme soubor `/etc/hosts`, kde přidáme na poslední řádek záznam `192.168.1.1 kukatko.viewer`. Název i heslo sítě je možné prostřednictvím aplikace změnit.

Po restartu zařízení bychom měli objevit na přenosném počítači v seznamu bezdrátových sítí síť s názvem `SSID_SITE` která by měla po zaslání požadavku pro připojení vrátit dialogové okno s požadavkem zadání hesla `HESLO_SITE`. Po zadání hesla Wi-Fi sítě by mělo být možné zobrazit webovou aplikaci kukatka ve webovém prohlížeči po zadání IP adresy `192.168.1.1`, tak i adresy `http://kukatko.viewer`. Prosté zadání adresy `kukatko.viewer` může způsobit, že se mohou některé prohlížeče pokusit začít zadaný text vyhledávat, proto je nutné doplnit i použitý protokol `HTTP`.

3.5.13 Knihovny nutné pro překlad zdrojových kódů

Pro případ úpravy zdrojových kódů aplikace provádějící analýzu zvuku zvonku a detekci zvonkového tlačítka jsou nutné knihovny `wiringPi` (knihovna nutná pro přístup k pinům GPIO) a `libmariadbclient-dev` (knihovna pro přístup k databázi z uživatelských aplikací).

Knihovnu `libmariadbclient-dev` nainstalujeme pomocí příkazu: `sudo apt-get install libmariadbclient-dev`, je však více, než pravděpodobné, že systém bude předtím vyžadovat provedení příkazu: `sudo apt-get update`, neboť dříve zajišťoval přístup k databázi jiný balíček.

Knihovnu `wiringPi` však budeme instalovat ze zdrojového kódu, jak je to doporučeno na oficiální stránce knihovny [22]. Tak, že pomocí programu `WinSCP` zkopírujeme archiv `tar.gz` například do složky `/home/pi`, kde jej následně rozbalíme pomocí příkazu `tar xzf wiringPi-8d188fa.tar.gz`. Poté

je nutné se do dané složky přepnout pomocí příkazu `cd wiringPi-8d188fa`, kde spustíme překlad a instalaci knihovny příkazem `./build`.

Zdrojové kódy aplikace provádějící analýzu zvuku zvonku a detekci zvonkového tlačítka poté provádíme příkazem:

```
gcc bellDetection.c -o bellDetection -lwiringPi -lpthread -lm  
'mysql_config -cflags -libs'
```

■ 3.5.14 Watchdog

V sekci 3.1.5 byl zmíněn balíček `watchdog`, který zabraňuje zablokování a ne-reagování procesů spuštěných pod systémem Raspbian. Balíček nainstalujeme příkazem: `apt-get install watchdog`.

Kapitola 4

Dosažené výsledky

Výstupem předloženého návrhu a realizace elektronického kukátka je funkční laboratorní vzorek zařízení, které má za cíl demonstrovat schopnost navržené koncepce. Z návrhu bylo vypuštěno několik, u finálních výrobků, běžně se vyskytujících a potřebných součástí.

Pro běžné použití elektronického kukátka by bylo zejména potřebné doplnit dané řešení o bateriové zálohování provozu přístroje a dobíjení tohoto zdroje energie při přítomnosti napájení z elektrické sítě a sofistikované řešení vypínání a zapínání přístroje. Důvodem pro nerealizaci této části byla nízká přidaná hodnota této podčásti do výsledného návrhu.

Opomenout nelze ani vhodnost vybavení finálního přístroje panoramatickou optikou, neboť v současné realizaci je pozorovací úhel na aplikaci kukátka malý ($62,2^\circ$ v jedné ose a $48,8^\circ$ v druhé ose; zdroj [23]). Důvodem pro vypuštění optiky z návrhu funkčního laboratorního vzorku je zpravidla nutnost zakázkové výroby čoček a jednotlivých elementů optiky a s tím spojená vysoká cena.

Další vhodnou částí by byla optimalizace uživatelského rozhraní webové aplikace tak, aby byla použitelná i na mobilních zařízeních s malou uhlopříčkou displeje a rovněž přidat možnost kompletního nastavení přístroje z jeho lokální obrazovky. Současné řešení je spíše koncipováno pro tablety a počítače typu PC. I tato část nebyla realizována z důvodu nízké přidané hodnoty tohoto řešení do výsledného návrhu. Kompletní nastavení přístroje z obrazovky na přístroji nebylo možné realizovat z důvodu absence vyhovující alfanumerické klávesnice na displeji přístroje.

Živé zobrazení videa ve webové aplikaci by bylo vhodné upravit tak, aby bylo v případě získání snímku detekční aplikací automaticky obnoveno a nebylo nutné tento úkon provádět ručně obnovou stránky v prohlížeči. Tato část nebyla realizována z důvodu nenalezení spolehlivě fungujícího řešení.

Část týkající se analýzy podobnosti zvuků zvonku by bylo vhodné upravit na nepřetržitou analýzu, neboť současné řešení je spíše vhodné k analýze složených tónů nebo velmi krátkých melodií. Realizovaný přístup byl zvolen s ohledem na skutečnost, že většinou zvuk zvonku bývá buďto složený tón (například zvonivý zvuk klasického paličkového zvonku) nebo krátká jednoduchá melodie (například gong). Dále pak také z důvodu, že se jedná o funkční vzorek, u kterého bylo třeba zjistit alespoň částečnou funkčnost daného návrhu a její náročnost vzhledem k více řešeným úlohám současně na

platformě Raspberry PI (detekce pohybu z obrazových dat, DHCP a DNS server, běh webového a databázového serveru, detekce stisknutého tlačítka a detekce zvuku zvonku). Pro nepřetržitou analýzu by bylo vhodné naprogramovat vlastní software ovladače pro zvukovou kartu a používat tzv. real-time operační systém, případně analýzu řešit na zvláštním obvodu uzpůsobeném pro rychlé operace zpracování signálů jako je například digitální signálový procesor DSP, ale i hradlové pole FPGA.

V poslední řadě by také bylo žádoucí, pro reálné nasazení, doplnit návrh o možnost připojení k přístroji vzdáleně mimo objekt a dosah sítě Wi-Fi poskytované zařízením. Pro tento účel jsou možné dva druhy řešení, buďto přiřazení statické IP adresy majiteli a přístup k zařízení přes tuto adresu, případně realizace tzv. proxy serveru (prostředníka) se statickou IP adresou v síti internet, ke kterému by se elektronické kukátko připojilo a majitel objektu by se mohl vzdáleně připojovat k zařízení přes tento server. Důvodem pro nerealizaci návrhu se statickou IP adresou je obecný předpoklad nízkého počtu uživatelů internetu vlastnících statickou IP adresu a disponujících znalostmi konfigurace sítí. Návrh proxy serveru pak nebyl realizován z důvodu následného přenosu obrazových dat mimo lokální síť, což vyžaduje zejména ke stále se zpřísňujícím předpisům v oblasti ochrany osobních údajů a dat, kterými obrazová data z elektronického kukátka jsou, pečlivé zabezpečení datové komunikace.

Kapitola 5

Závěr

Předložený text diplomové práce obsahuje stručnou analýzu trhu s dostupnými zařízeními elektronických kukátek, analýzu požadavků na hardware a software a možné případy užití. V textu je návrh doménového modelu a architektury aplikace s ohledem na ukládání obrazových dat do systému a jejich sběr. Dále je v textu uvedený ucelený popis hardwarových modulů, včetně obvodového návrhu modulu vlastní konstrukce.

Obsahem textu je i popis algoritmizace Fourierovy transformace od obecného vzorce v oblasti spojitého času přes diskrétní variantu transformace včetně objasnění nutných operací zejména pro získání harmonických kmitočtů z amplitudového spektra na demonstračním příkladu a ilustraci amplitudového spektra získávaného z reálného přístroje-osciloskopu až po výpis kódu algoritmu FFT DIT v jazyce C uvedeného v příloze.

Text rovněž obsahuje kompletní instalační manuál nezbytný pro možnost instalace v případě vytvoření dalšího exempláře přístroje nebo poškození původní instalace v zařízení.

Praktickým výstupem diplomové práce je realizace hardware a software funkčního vzorku elektronického kukátka s využitím převážně již hotových modulů a komponent s doplněním modulu vlastní výroby, který je nezbytný pro možnost připojení zvonkového napětí, na které není žádný standardní vstup hotových modulů konstruován. Zároveň vlastní modul řeší i nezbytnou zálohu času pomocí obvodu reálného času a rovněž obsahuje předzesilovač pro elektretový mikrofon sloužící jako senzor pro analýzu zvuku. Software zařízení je složené ze standardně dostupného softwarového vybavení (operační systém, webový a databázový server, program pro detekci pohybu-změn v obrazu, apod.) doplněného o software vytvořený.

Vytvořený software je složen z několika částí, které spolu interagují. Jednou z částí je webová aplikace pro správu obrazových dat uložených v databázi, doplněnou o možnost jejich tvorby na základě uživatelského požadavku a možnou úpravu některých parametrů a nastavení. Celá aplikace je dostupná vzdáleně přes síť Wi-Fi poskytovanou zařízením a v omezené míře je poskytována přímo na zařízení.

Druhá část software je vícevláknový program složený z vláknů pro detekci zvuku zvonku založeného na porovnávání harmonických kmitočtů nad získanými úseky zvuku ze zvukové karty vůči úsekům vzorového zvuku a vláknů pro detekci přítomnosti zvonkového napětí.

Třetí část software je jednoduchý program sloužící ke komunikaci s obvodem reálného času po sběrnici I²C.



Literatura

- [1] FOWLER, Martin. *Destilované UML*. 1. vydání, Nakladatelství Grada, Praha, 2009.
- [2] VEDRAL, Josef; SVATOŠ, Jakub. *Zpracování a digitalizace analogových signálů v měřicí technice*. 1. vydání, Česká technika - nakladatelství ČVUT, Praha, 2018.
- [3] NEUMANN, Přemek; UHLÍŘ, Jan. *Elektronické obvody a funkční bloky 1*. 2. přepracované vydání, České vysoké učení technické v Praze, Praha, 1999.
- [4] NEUMANN, Přemek; UHLÍŘ, Jan. *Elektronické obvody a funkční bloky 2*. 1. vydání, České vysoké učení technické v Praze, Praha, 2001.
- [5] VOBECKÝ, Josef; ZÁHLAVA, Vít. *Elektronika - Součástky a obvody, principy a příklady*. 3. rozšířené vydání, Grada Publishing, Praha, 2006.
- [6] RAY, S., Deborah; RAY, J., Eric. *Unix - podrobný průvodce*. 1. vydání, Grada Publishing, Praha, 2009.
- [7] STEPHENS, Ryan; PLEW, Ron; JONES, D., Arie. *Naučte se SQL za 28 dní*. Dotisk 1. vydání, Nakladatelství Computer Press, Brno, 2012.
- [8] UHLÍŘ, Jan; SOVKA, Pavel. *Číslicové zpracování signálů*. 2. přepracované vydání, Vydavatelství ČVUT, Praha, 2002.
- [9] CORMEN, H., Thomas; LEISERSON, E., Charles; RIVEST, L., Ronald; STEIN, Clifford; *Introduction to algorithms*. 3. vydání, The MIT Press, Cambridge, 2009.
- [10] *Window function* [online].
Dostupné na: <www.en.wikipedia.org/wiki/Window_function>.
- [11] *Rychlá Fourierova transformace (FFT) pro AVR* [online].
Dostupné na: <<http://elektronika.kvalitne.cz/ATMEL/necoteorie/transformation/AVRFFT/AVRFFT.html>>.

- [12] *Discrete Fourier Transform - MATLAB & Simulink Documentation* [online].
Dostupné na: <www.mathworks.com/help/signal/ug/discrete-fourier-transform.html>.
- [13] *Raspberry Pi GPIO Pinout* [online].
Dostupné na: <<https://pinout.xyz>>.
- [14] *Raspberry Pi LCD DSI Display Connector* [online].
Dostupné na: <www.petervis.com/Raspberry_PI/Raspberry_Pi_LCD/Raspberry_Pi_LCD_DSI_Display_Connector.html>.
- [15] *SPI - Raspberry Pi Documentation* [online].
Dostupné na: <<https://www.raspberrypi.org/documentation/hardware/raspberrypi/spi/README.md>>.
- [16] *Raspberry Pi FAQs - Frequently Asked Questions* [online].
Dostupné na: <<https://www.raspberrypi.org/help/faqs/#powerReqs>>.
- [17] *ADA-15 USB - HQ MINI audio* [online].
Dostupné na: <www.axagon.eu/produkty/ada-15>.
- [18] *Grafický formát BMP - používaný a přitom neoblíbený* [online].
Dostupné na: <<http://www.root.cz/clanky/graficky-format-bmp-pouzivany-a-pritom-neoblibeny>>.
- [19] *Wav (RIFF) File Format Tutorial* [online].
Dostupné na: <<http://www.topherlee.com/software/pcm-tut-format.html>>.
- [20] *Download PuTTY - a free SSH and telnet client for Windows* [online].
Dostupné na: <<https://www.putty.org>>.
- [21] *WinSCP - free SFTP, SCP and FTP client for Windows* [online].
Dostupné na: <<https://winscp.net/eng/download.php>>.
- [22] *Wiring Pi - GPIO Interface library for the Raspberry PI* [online].
Dostupné na: <wiringpi.com/download-and-install/>.
- [23] *Rpi Camera Module- eLinux.org* [online].
Dostupné na: <https://elinux.org/Rpi_Camera_Module#Technical_Parameters_.28v.2_board.29>.
- [24] *Cannyho hranový detektor* [online].
Dostupné na: <https://cs.wikipedia.org/wiki/Cannyho_hranový_detektor>.
- [25] *Algoritmus korelace v digitálním zpracování signálů* [online].
Dostupné na: <<https://automatizace.hw.cz/clanek/2006031701>>.

Příloha A

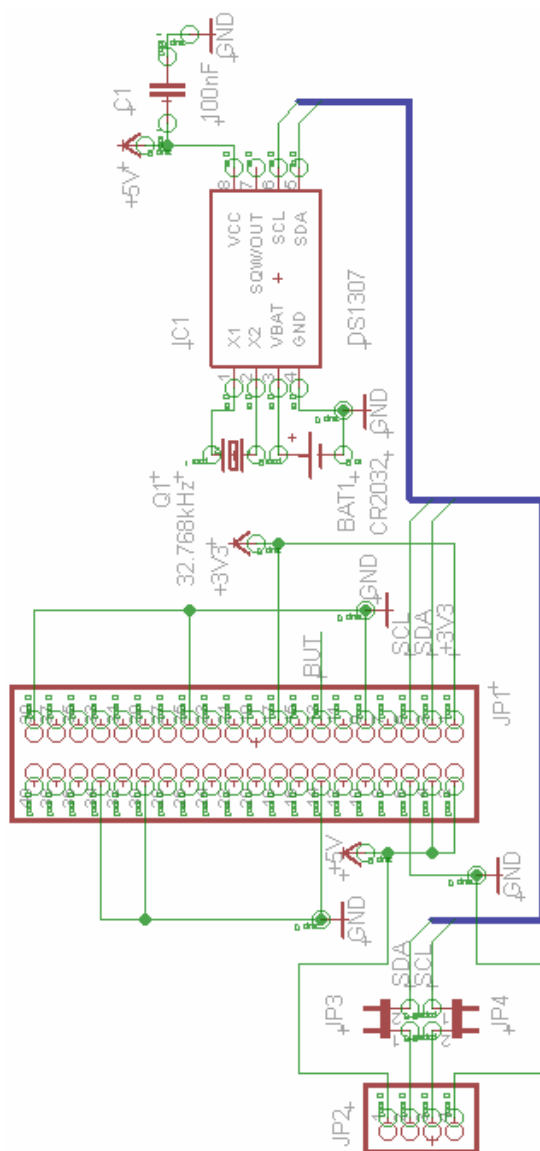
Obsah přiloženého DVD

DVD, které je přiloženo k této diplomové práci obsahuje následující přílohy. V závorkách je uvedena absolutní cesta na disku DVD.

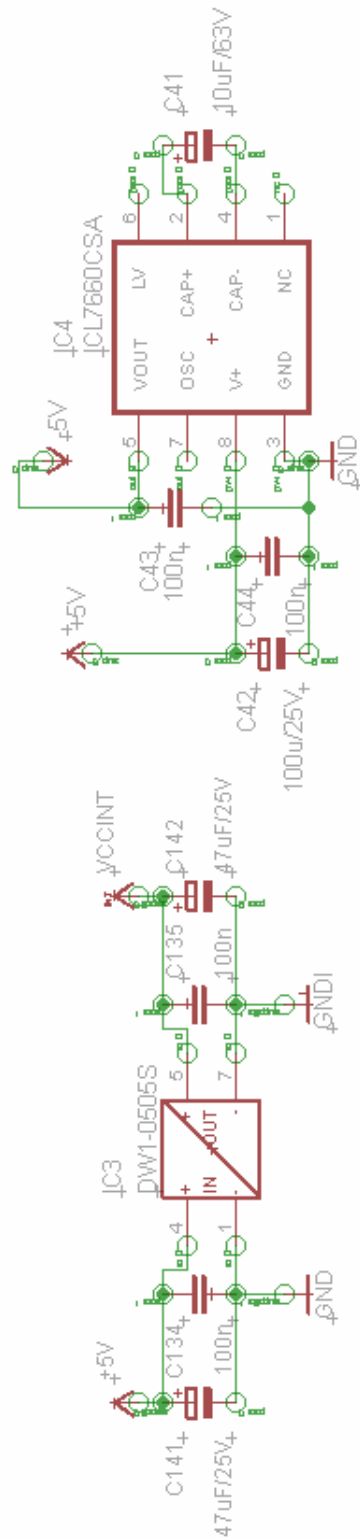
- Zadání diplomové práce (/DP_Zadani.pdf)
- Text diplomové práce ve formátu PDF (/DP-Josef_Kiefmann.pdf)
- Archív s aplikací Kukátko /Kukatko.zip
- Adresář s katalogovými listy obvodů a součástek použitých pro konstrukci vlastního modulu (/Katalogove_listy)
- Soubory se schématickým podkladem a plošným spojem (/DPS.zip)
- Soubory podkladů pro výrobu desky plošných spojů (/DPS_vyroba.zip)
- Inicializační skript SQL pro tvorbu databáze, databázových tabulek a inicializaci administrátorského účtu aplikace v databázi (/skript.sql).
- Knihovna wiringPi pro obsluhu GPIO pinů ve verzi použité pro realizaci aplikace detekující zvonkové napětí (/wiringPi-8d188fa.tar.gz).

Příloha B

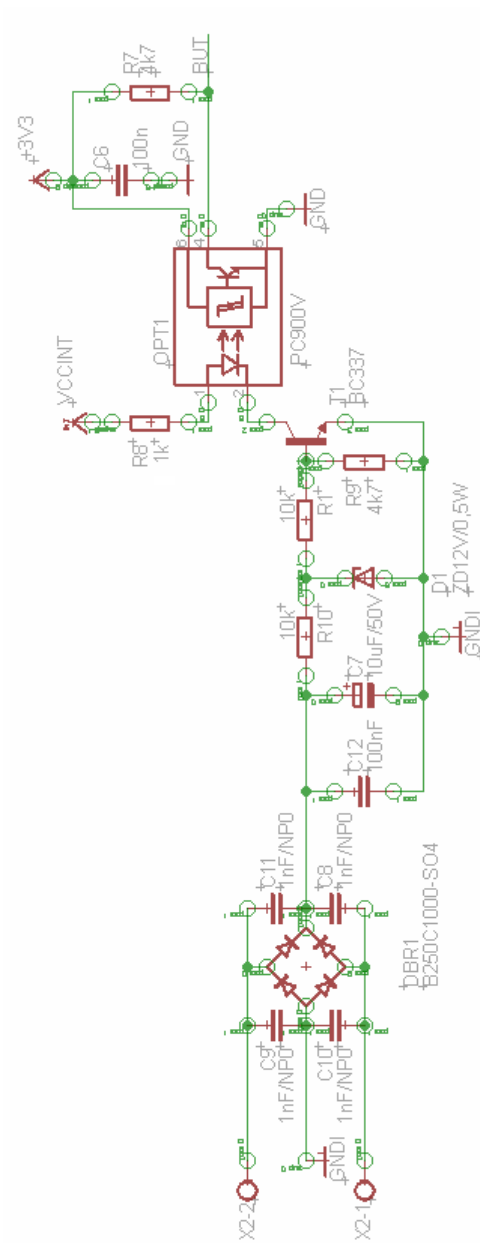
Schéma



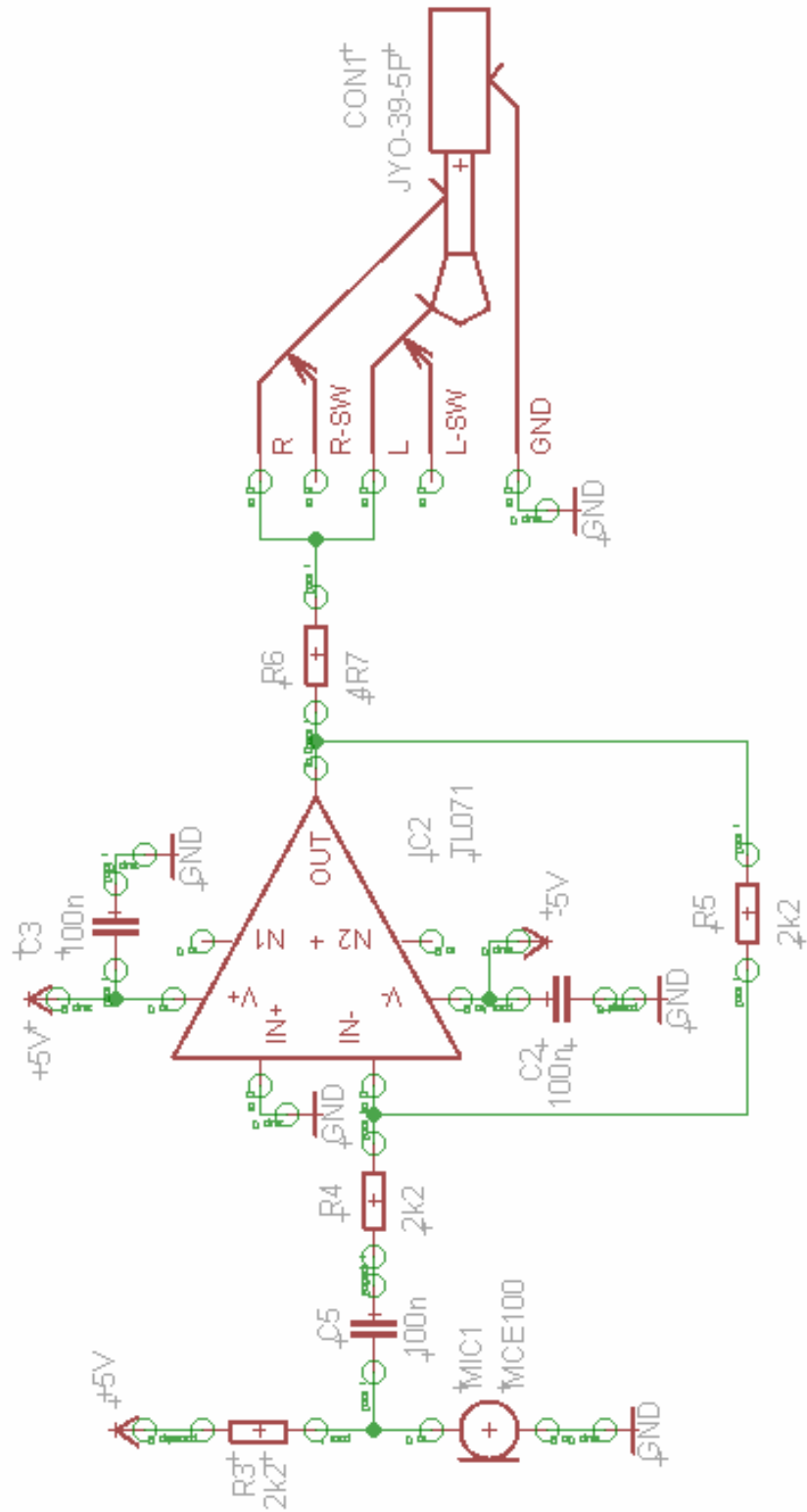
Obrázek B.1: Zapojení modulu na konektor GPIO a připojení časovače DS1307



Obrázek B.2: Tvorba galvanicky odděleného napájení pro vstupní obvod pro připojení zvonkového napětí a tvorba invertovaného napájení pro předzesilovač



Obrázek B.3: Vstupní obvod pro připojení zvonkového napětí



Obrázek B.4: Předzesilovač pro elektretový mikrofon

Příloha C

Zdrojový kód funkcí v jazyce C pro výpočet FFT

```
typedef struct{
    double *samples;
    unsigned long length;
    unsigned long samplingFrequency;
} signal_structure;

typedef struct{
    double *realValues;
    double *imagValues;
    unsigned long length;
    unsigned long samplingFrequency;
} complex_sequence;

typedef struct{
    double *amplitudes;
    double *frequencies;
    unsigned long length;
} amplitude_spectrum;

typedef struct{
    double *amplitudes;
    double *frequencies;
    unsigned long length;
} harmonic_frequencies;

void applyBlackmannHarrisWindow(signal_structure *signal){
    unsigned long k;
    double w;
    unsigned long N = (signal->length);
    for(k = 0; k < N; k++){
        w = (0.422323 - 0.49755 * cos(2*M_PI*((double)k/N)));
        w = w+0.07922*cos(4*M_PI*((double)k/N));
        (signal->samples[k]) = (signal->samples[k])*w;
    }
}
```

```

    }
}

void applyBitPermutation(signal_structure *signal){
    double lg2, temp;
    unsigned int lg2d, i, j, odd, i2;
    lg2 = (log10((signal->length))/log10(2));
    lg2d = lg2;
    for(i = 0; i < (signal->length); i++){
        if((int) lg2d % 2 == 0){
            odd = 0;
            i2 = 0;
        } else{
            odd = 1;
            i2 = 0;
            i2 = i2 | (i & (1 << ((lg2d)/2)));
        }
        for(j = 0; j < ((lg2d-odd)/2); j++){
            i2 = i2 | (((i >> (lg2d-1-j)) & 1) << j);
            i2 = i2 | (((i & (1 << j)) >> j) << (lg2d-1-j));
        }
        temp = (signal->samples[i2]);
        (signal->samples[i2]) = (signal->samples[i]);
        (signal->samples[i]) = temp;
    }
}

void computeFFT(complex_sequence seq, signal_structure sig){
    unsigned long i, j, k, m;
    double tReal, tImag, uReal, uImag, angle, real, imag;
    (seq->length) = (sig->length);
    (seq->samplingFrequency) = (sig->samplingFrequency);
    unsigned long N = (seq->length);
    for(i = 0; i < N; i++){
        (seq->realValues[i]) = sig.samples[i];
        (seq->imagValues[i]) = 0;
    }
    for(i = 0; i <= (unsigned long) ((double)log(N)/log(2)); i++){
        m = pow((double)2,(double)i);
        for(k = 0; k < N; k=k+m){
            for(j = 0; j < (unsigned int) ((double)m/2); j++){
                angle = (2*M_PI*j)/m;
                real = (seq->realValues[k+j+(long)((double)m/2)]);
                imag = (seq->imagValues[k+j+(long)((double)m/2)]);
                tReal = (cos(angle)*real)-(sin(angle)*imag);
                tImag = (cos(angle)*imag)+(sin(angle)*real);
            }
        }
    }
}

```

```

    uReal = (seq->realValues[k+j]);
    uImag = (seq->imagValues[k+j]);
    (seq->realValues[k+j]) = uReal + tReal;
    (seq->imagValues[k+j]) = uImag + tImag;
    (seq->realValues[k+j+(long)((double)m/2)]) = uReal - tReal;
    (seq->imagValues[k+j+(long)((double)m/2)]) = uImag - tImag;
}
}
}
}

void getASpectrum(amplitude_spectrum *sp, complex_sequence seq){
    unsigned long i;
    double real, imag;
    unsigned long N = (seq.length);
    for(i = 0; i < N; i++){
        (sp->frequencies[i]) = i*((seq.samplingFrequency)/N);
        real = seq.realValues[i];
        imag = seq.imagValues[i];
        (sp->amplitude[i]) = sqrt(pow(real,2)+pow(imag,2));
    }
}

```