

Hardware

1

Jedním z úkolů práce bylo ukázat, že je možné pomocí veřejně dostupných technologií vytvořit efektivní zařízení. Pro tento a další účely byl v práci testován a použit následující hardware:

1. Raspberry Pi 4 model B
2. ReSpeaker 4-Mic Array pole čtyř mikrofónů
3. Krokový motor NEMA17 (47 mm) model 42HD6021-03
4. Světelné senzory TSL2591 v počtu 4 kusů
5. Solární panel

1.1 Raspberry Pi 4 model B

Raspberry Pi 4B¹ je nízkonákladový a malý jednodeskový počítač s dostatečným výkonem pro multifunkční úlohy. Pro naši úlohu byl zvolen operační systém Debian Bullseye² (Desktop, 64-bit). V testovací verzi a v práci byl použit model s novějším typem operační paměti LPDDR4 a 2 GB RAM. Zařízení je vybaveno čtyřjádrovým procesorem ARM Cortex-A72 s taktem 1,5 GHz. Grafika VideoCore (verze VI). Což dohromady zajišťuje dobrý výkon a vysokou rychlost počítače.

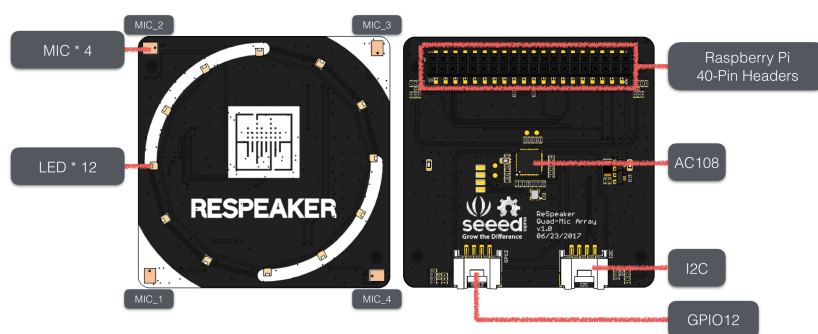
1.2 ReSpeaker 4-Mic Array

Tato součástka od výrobce Seeed Studio³ je vybavena čtyřmi analogovými mikrofony a audio kodekem AC108 pro snímání a zpracování hlasu ve vysokém rozlišení.

¹<https://rpishop.cz/raspberry-pi-4/1599-raspberry-pi-4-model-b-2gb-ram-765756931175.html>

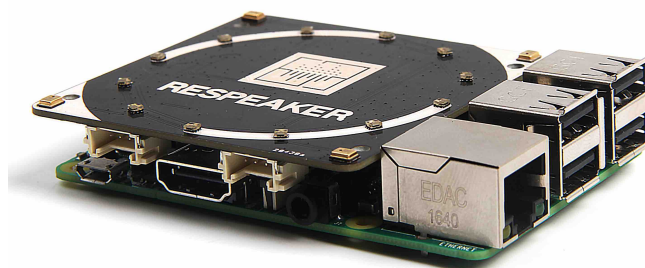
²<https://wiki.debian.org/DebianBullseye>

³https://wiki.seeedstudio.com/ReSpeaker_4_Mic_Array_for_Raspberry_Pi/



Obrázek 1.1: Přehled mikrofonu ReSpeaker 4-Mic Array použitého v práci [3]

Zařízení je určeno k použití jako hlasové uživatelské rozhraní primárně v kombinaci s Raspberry Pi. Tato verze mikrofonu je navíc vybavena LED kroužkem s 12 programovatelnými LED diodami APA102. Díky těmto čtyřem mikrofونům a LED kroužku má ReSpeaker 4-Mic Array schopnost detekovat a indikovat hlasovou aktivitu (VAD - Voice Activity Detection), odhadovat směr zvuku (DOA - Direction Of Arrival) a provádět detekci klíčových slov (KWS - Keyword Spotting) široce používaných v aplikacích hlasové interakce.

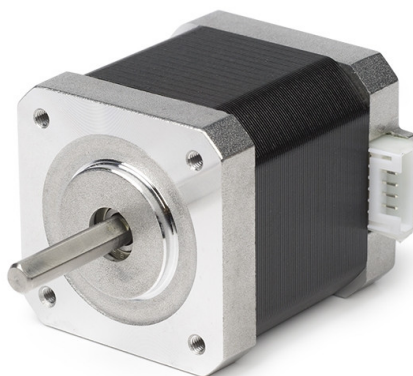


Obrázek 1.2: ReSpeaker 4-Mic Array mikrofon propojený s Raspberry Pi 4B [3]

- *Voice Activity Detection (VAD)* – detekce přítomnosti nebo nepřítomnosti lidské řeči ve vstupním akustickém signálu.
- *Direction Of Arrival (DOA)* – směr, ze kterého se obvykle šíří zvukové vlny.
- *Keyword Spotting (KWS)* – proces rozpoznávání předem definovaných slov z řečového signálu.

1.3 Krokový motor

Pro zajištění rotačního pohybu robota byl vybrán krokový motor od výrobce NEMA17⁴ (konkrétně model 42HD6021-03). Tento malý dvoufázový krokový motor má na svůj rozměr (47 mm) dostatečně velký statický moment (500 mN.m) a vysokou přesnost - 200 kroků na otáčku, což je 1,8 stupně. Motor se jednoduše instaluje a demontuje bez použití speciálních nářadí. Často se používá pro 3D tiskárny. V bakalářské práci byl použit k rotaci horní části cílového zařízení.



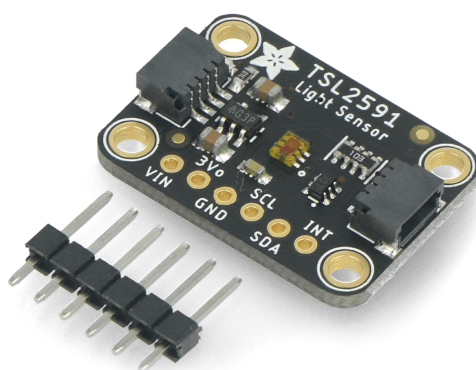
Obrázek 1.3: Krokový motor NEMA17 [4]

1.4 Světelný senzor

Pro určení místa s nejvyšším dopadem slunečního světla v práci byly použity čtyři světelné senzory TSL2591⁵ od výrobce Adafruit. Světelný senzor TSL2591 má vysoký dynamický rozsah (600M:1) a velkou citlivost s účinným maximem do 88.000 Lx. Obsahuje infračervené a celospektrální diody, díky čemu umožňuje měřit odděleně různé typy světla. Sensor lze snadno spojit s počítačem Raspberry Pi přes I2C rozhraní.

⁴<https://novo3d.in/product/nema17-stepper/>

⁵<https://rpishop.cz/adafruit/269-svetelny-senzor-tsl2591.html>



Obrázek 1.4: Světelný senzor TSL2591 [5]

1.5 Solární panel

Polykrystalický solární panel ve velikosti 110 x 60 x 2,5 mm⁶. S výstupním napětím 6 V a s výstupním proudem do 200 mA. Solární panel byl použit k zajištění dodatečného nabíjení cílového zařízení.

⁶<https://dratek.cz/arduino/1589-solarni-panel-6v-1w-az-200ma.html>

V této části budou podrobně popsány technologie a způsoby jejich aplikace, použité v bakalářské práci. Technologie byly zvoleny na základě poměru jejich snadné integrace a efektivity jejich použití. Některé z technologií byly otestovány a dále využity prostřednictvím realizovaných demo verzí, některé z nich posloužily jako podklad pro vypracování této práce. Základní technologie a nástroje:

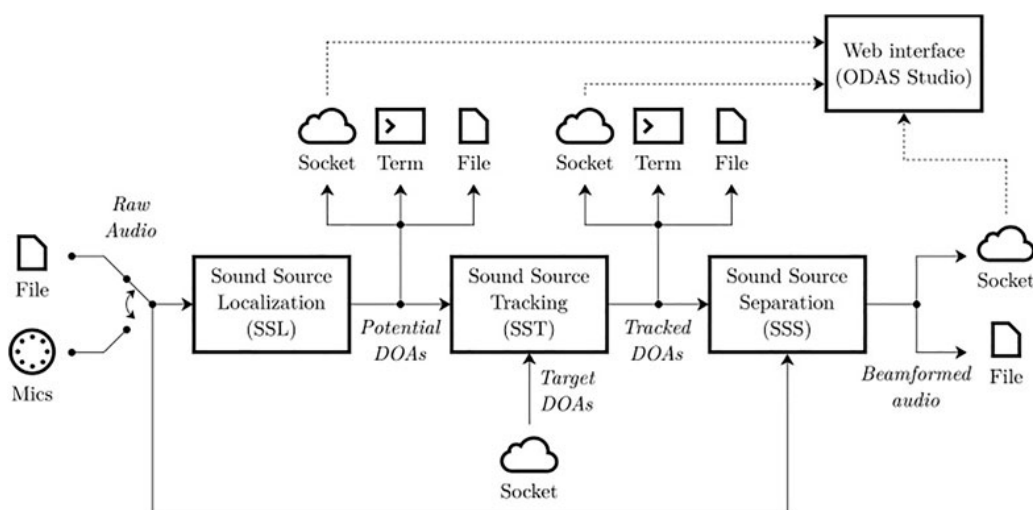
1. ODAS (Open embeddeD Audition System)
2. Platforma Picovoice
3. Porcupine
4. ROS (Robot Operating System) - používá se ke spojení všech technologií a jejich komunikace v cílovém zařízení
5. Tkinter - drobný popis...

Vše bylo implementováno v programovacím jazyce Python, byl použit operační systém Debian. Pro testování a modelování byl použit také operační systém Windows.

2.1 Framework ODAS

Open embeddeD Audition System (ODAS) je open source knihovna určená k lokalizaci, sledování, separaci a následnému filtrování zdrojů zvuku. ODAS je framework vyvinutý pro použití na nízkonákladovém a nízkoúčinném vestavěném hardwaru. Je napsaný výhradně v programovacím jazyce C, díky čemuž má lepší optimalizaci a je lépe přenositelný na různá vývojová prostředí [1]. Maximální přenositelnost frameworku je dosažena tím, že má jenom jednu externí závislost na známé knihovně třetí strany FFTW3, která se používá k efektivnímu výpočtu rychlých Fourierových transformací [6]. V rámci práce ODAS byl použit k lokalizaci zdroje zvuku a jeho postfiltraci.

Na obr. 2.1 je znázorněn zvukový kanál pro zpracování zvuku a vstupně-výstupní rozhraní, z nichž každé běží v samostatném vlákne, aby bylo možné plně využít možností vícejádrových procesorů. Zdrojový zvuk může být reprezentován předem nahraným vícekanálovým audio souborem RAW nebo může být získán přímo z připojeného mikrofону pro zpracování v reálném čase. Modul lokalizace zdroje zvuku (SSL - Sound Source Localization) generuje pevný počet potenciálních zdrojů zvuku, které se předávají modulu sledování zdroje zvuku (SST - Sound Source Tracking). Modul SST identifikuje sledované zdroje a přenáší tyto DOA^{1,2} do modulu provádějícího separaci zdrojů zvuku (SSS - Sound Source Separation) k provedení formování paprsku pro každý cílový zdroj zvuku. Uživatel může rovnou definovat cílové DOA, pokud je směr zdroje zvuku předem známý a není vyžadována lokalizace ani sledování. Vygenerované segmenty lze zapsat do audio souborů RAW nebo také odeslat prostřednictvím socketu [6].



Obrázek 2.1: ODAS schéma zpracování příchozího zvuku [6]

- *Sound Source Localization (SSL)* – schopnost určit polohu nebo zdroj detekovaného zvuku podle směru a vzdálenosti.
- *Sound Source Tracking (SST)* – schopnost sledovat měnící se zdroj zvuku v reálném čase.
- *Sound Source Separation (SSS)* – schopnost oddělit aktivní řeč nebo určitý signál od šumu na pozadí nebo ticha.

2.1.1 Lokalizace zdroje zvuku

Lokalizace zvuku ve frameworku se opírá o metodu Generalized Cross-Correlation with Phase Transform (GCC-PHAT), která se vypočítá pro každý pár mikrofónů. ODAS používá Inverse Fast Fourier Transform (IFFT) algoritmus k efektivnímu výpočtu křížové korelace ze signálů ve frekvenční oblasti. Při práci s malými nebo krátkými poli signálů, ODAS může také interpolovat křížovou korelaci, aby se zlepšila přesnost lokalizace a eliminoval artefakt diskretizace TDOA, který vzniká při Fourierově transformaci. Hlavním principem výpočtu DOA^{1,2} je metoda Steered-Response Power with Phase Transform (SRP-PHAT). Pro každý DOA framework vypočítá SRP-PHAT součtem hodnoty křížové korelace spojené s odpovídajícím časovým rozdílem (TDOA), získaným pomocí GCC-PHAT pro každý pár mikrofónů a vrátí hodnotu DOA s nejvyšším výkonem.

2.2 Platforma Picovoice

Pro přesné zpracování zvukových dat a zároveň pro rychlé, ale efektivní trénování zvukových konceptů byla zvolena platforma Picovoice¹. Picovoice je multiplatformní vývojářský software určený k vytváření hlasových produktů. Ve srovnání s populárními produkty pro rozpoznávání a zpracování řeči od velkých společností (např. Google, Amazon), které k analýze a práci s hlasem využívají Cloud, Picovoice se zaměřuje výhradně na on-device procesy[7]. Tato metoda umožňuje dosáhnout přesnějších výsledků a pracovat bez připojení k internetu. Platforma Picovoice má webový systém Picovoice Console, pomocí kterého je možné rychle natrénovat a otestovat hlasové koncepty, pro jejich další využití v KWS^{1,2}, VAD^{1,2} a dalších systémech.

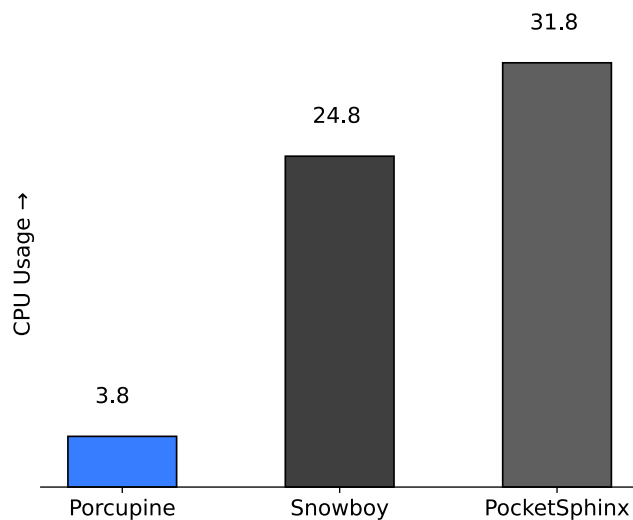
2.2.1 Porcupine

Porcupine² je vysoce přesný a procesově nenáročný nástroj pro detekci klíčových slov (Keywords^{2.2.1}), který rozpoznává jedinečné signály pro přechod softwaru z pasivního stavu na aktivní. Porcupine je jeden z nástrojů platformy Picovoice^{2,2}, který umožňuje vytvářet aplikace s možností neustálého poslechu hlasových signálů. Jedním z cílů bakalářské práce bylo zprovoznění potřebných technologií na low-cost zařízení, což Porcupine umožňuje díky své procesní nenáročnosti (viz Obrázek 2.2) a tomu, že je multiplatformní. Zakladní překážkou implementace mechanismů Wake-Word^{2.2.1} (Keyword) je jejich závislost na masivním sběru dat pro trénování každého nového modelu. Tento nástroj společně s Picovoice má webové rozhraní, kde je možné snadno a rychle natrénovat a otestovat vlastní klíčové slovo a z toho vytvořit

¹<https://picovoice.ai/docs/>

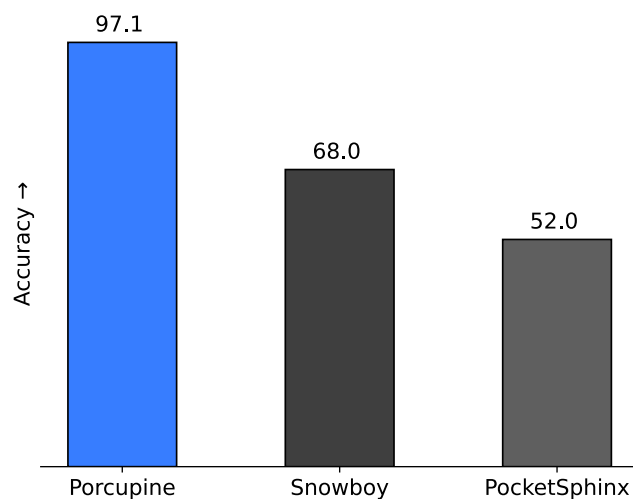
²<https://picovoice.ai/platform/porcupine/>

vlastní model pro použití v dalším rozpoznávání. Takovým způsobem Porcupine eliminuje potřebu sběru dat pro každý nový model.



Obrázek 2.2: Využití CPU (single-core) Porcupine ve srovnání s jinými populárními nástroji pro detekci Keyword [8]

Dalším důvodem pro výběr tohoto nástroje je jeho přesnost. Obrázek 2.3 zobrazuje přesnost nástrojů, když frekvence false alarm je jednou za 10 hodin s šumem (10 dB SNR) a řečí na pozadí[8]:



Obrázek 2.3: Porovnání přesnosti detekce Keyword Porcupine s jinými nástroji [8]

- *Keyword (Wake-Word, Hot-Word, Keyphrase atd.)* – specifické slovo nebo fráze, která po vyslovení může aktivovat neaktivní zařízení, například: "Hey Siri", "OK Google" a tak dále. Používá se v KWS^{1,2}.

2.3 Framework ROS

Robot Operating System (ROS) je open source framework spravovaný společností Open Robotics se sadou softwarových knihoven a nástrojů pro vytváření robotických aplikací. ROS poskytuje hardwarovou abstrakci, nízkoúrovňové ovládání zařízení, předávání zpráv mezi procesy, nabízí ovladače zařízení, knihovny, vizualizéry, správu balíčků a další funkce. Tento framework implementuje několik různých stylů komunikace, včetně RPC synchronní komunikace realizovatelnou přes služby (Services), asynchronního streamování dat přes témata (Topics) a ukládání dat na serveru (Parameter Server) používaného uzly (Nodes) za procesním běhu. Krátký popis základních ROS prvků:

- *Package (Balík)* – hlavní jednotka pro organizaci softwaru v ROS. Slouží jako strukturovaný kontejner pro organizaci a správu softwarových komponent nezbytných pro efektivní strukturování, sdílení kódu, dat a zdrojů v rámci ekosystému ROS.
- *Node (Uzel)* – nezávislý modul, který v softwarové architektuře zařízení vykonává určitou funkci nebo úkol. Tyto moduly mohou vzájemně komunikovat, aby dosáhly celkové funkcionality. ROS systém se obvykle skládá z velkého množství nodů.
- *Message (Zpráva)* – jednoduchá datová struktura sestávající z typizovaných polí obsahujících standardní primitivní typy (integer, float, boolean, atd.). Uzly komunikují mezi sebou pomocí zpráv.
- *Topic (Téma)* – speciální kanál, prostřednictvím čeho uzly mohou publikovat a přijímat zprávy. Je to prostředek pro výměnu dat mezi uzly, který jim umožňuje vzájemnou interakci a spolupráci v ROS systému.
- *Services (Služby)* – speciální typ komunikace využívající témata, který umožňuje uzlům provádět vzdálené operace a požadavky. Služby jsou založeny na modelu požadavek-odpověď. Jeden uzel odešle požadavek (request) na provedení určité operace a jiný uzel tento požadavek splní a odešle zpět odpověď (reply).