1 Project documentation

The project is a multimedia station that can play Spotify songs or radio over frequency modulation (FM). It is implemented with a Raspberry Pi 3 Model B V1.2 (RaPi 3B) running Ubuntu Server 24.04 LTS, and an ESP-32D (ESP32D) running micro-Robot Operating System (ROS). Due to hardware (i2c-related) issues of the radio receiver, and restrictions on the Spotify Application Programming Interface (API), the implementation of the music players (radio and Spotify) could not be finished.

The RaPi 3B contains multiple ROS nodes for reading the sensor input for Infrared (IR) signals, running the FM receiver (that does not work properly due to an i2c issue), and displaying information on a small Liquid-Crystal Display (LCD). The RaPi 3B also runs a web server ensuring control without relying on the IR receiver.

The ESP32D runs micro-ROS with FreeRTOS, and it hosts multiple ROS nodes for displaying the frequency and volume on a 7-segment Light Emitting Diode (LED) display and a LED strip, respectively. Furthermore, it also contains two nodes for reading and publishing the potentiometer values for the volume and frequency.

Also, two shell scripts were implemented for launching all Robot Operating System 2 (ROS2) nodes running directly on the RaPi 3B, and for starting the micro-ROS agent running in a Docker instance on the RaPi 3B.

2 Used materials

2.1 Computers & microcontrollers

- **RaPi 3B**: Single Board Computer (SBC) used as the main component of the multimedia station.
- **ESP32D**: Additional microcontroller used for sensor inputs and displaying information.

2.2 Sensors

- **IR-based remote control**: Allows user interactions such as skipping songs or other commands.
- **IR receiver (CHQ1838)**: Receives the IR signals from the IR-based remote control.
- **3-pin linear potentiometer (B10k** (10000Ω)**)**: Used for changing the volume.
- **3-pin linear potentiometer (B100k** (100000Ω)**)**: Used for adjusting radio frequency.
- **FM receiver (Si470x)**: Used in combination with an antenna to receive FM signals at a selected frequency. Does not work due to an i2c issue.

2.3 Interfaces

- **7-segment LED display (3461BS-1 440)**: Used for displaying the frequency as a number.
- **LED strip (2510SR-1 440)**: Used for visualizing the current volume level.
- **LCD (ST7735S)**: Used for displaying a header and a usual text.
- **Audio interface**: The FM receiver plays the radio sound over a connected Auxiliary (AUX) speaker, and the RaPi 3B also has a speaker connected via AUX.

2.4 Others

- **Two Shift registers (SN74HC595N)**: Used for easier control of the digits board and LED strip.

3 ROS diagram

This section contains the figure 1 showing the diagram with the ROS nodes and ROS topics. The project will only use one overall ROS package for Python and one for micro-ROS code. The music controller node manages the music player. The music player is the parent class of radio player and Spotify player. These two classes are the actual players. Due to hardware and API restrictions, the implementation of the actual players could not be finished.

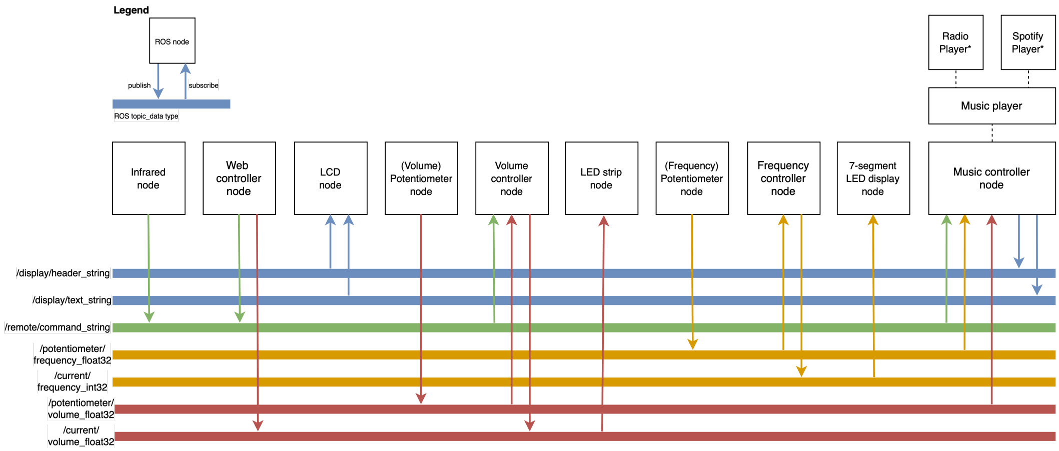

Figure 1: Diagram showing the ROS nodes and ROS topics.