**Amplified FM Tuner Evaluation Board**

From: Aldo Ndreu, Ryan Antolin, & Erick Cantos  
Discipline: Computer Engineering Technology  
Date: 03/12/2018

# Declaration of Joint Authorship

We, Aldo Ndreu, Ryan Antolin, and Erick Cantos confirm that this work submitted for assessment is the joint work of ourselves and is expressed in our own words. Any uses made within of other works of any other author, in any form (ideas, equations, figures, previous technologies, tables, programs, texts) are properly acknowledged at the point of use. A list of the references used is included. Aldo Ndreu has handled the software and mobile application, while Ryan Antolin has handled the Database, and Erick Cantos has handled the hardware aspects of this project.

# Approved Proposal

## Executive Summary

As students in the Computer Engineering Technology program, we will be integrating the knowledge and skills we have learned from our program into this Internet of Things themed capstone project. This proposal requests the approval to build the hardware portion that will connect to a database as well as to a mobile device application. The internet connected hardware will include a custom PCB with the following sensors and actuators such as the Speaker Bonnet & FM Tuner Evaluation Board - Si4703. The database used in this project will be Firebase and will store FM radio stations, and possibly favorited stations.

The mobile device functionality will include setting or selecting different FM radio stations, favoriting different FM radio stations, displaying song or station currently playing, and will be further detailed in the mobile application proposal. We will be collaborating with the following department which is the Prototype Lab (For Extra Help). In the winter semester we planned to form the group of Aldo Ndreu and following students, who are also building similar hardware this term Ryan Antolin and Erick Cantos. The hardware will be completed in CENG 317 Hardware Production Techniques independently and the application will be completed in CENG 319 Software Project but will need some changes and adjustments in order to meet specific requirements. These will be integrated together in the subsequent term in CENG 355 Computer Systems Project as a member of a 3-student group.

## Background

The problem solved by this project which we will be creating will be the capability of being able to connect to an amplified Speaker Bonnet via FM Radio. How this will work is by taking a mobile device and connecting to the database in order for the FM Radio stations to play through the Speaker Bonnet. A bit of background about these topics will include both devices being used. One of the device being used will be the FM evaluation board tuner chip. This device does more then tuning into FM stations, it can also detect both data service and radio broadcast data service. It can also be used to display station id and song to the user as well as have great filtering and carrying detection. This board will be able to pick up multiple radio stations and makes a great tool in order for it to be implemented with a Raspberry Pi. The other device used is the amplified speaker bonnet. By using the speaker bonnet, this will act as the output for FM Tuner sensor and will be the primary source in which the sound will be coming from. It will amplify the audio by boosting the signal in certain areas that you usually cannot hear with your current mobile speakers.

We have searched for prior art via Humber’s IEEE subscription selecting “My Subscribed Content” and have found and read which provides insight into similar efforts.

## Concluding remarks

This proposal presents a plan for providing a solution for FM Tuner sensor to be connected with the amplified speaker bonnet for high quality sound. This is an opportunity to integrate the knowledge and skills developed in our program to create a collaborative capstone project demonstrating my ability to learn how to support projects. We request approval of this project.

# Abstract

Amplified sound is important when wanting to hear certain audio at a greater and increased state. By connecting an FM tuner sensor to a speaker bonnet, the outputted sound should be at a level where it is enjoyable and meets user satisfaction when tuning to different frequencies. This system will allow users to tune into certain frequencies while also producing amplified sound. The FM tuner sensor will obtain data through the mobile application from the database and allow different frequencies to be sent to the Raspberry Pi in order to allow users to listen to certain stations. The database will grab most of the sensors data, and make it available to an Android application. The application will display different user selections as well as a favourites tab to satisfy user preference. This system has the potential to be unique given all the necessary implementation and should provide users great satisfaction when used assuming they have internet connectivity.

Table of Contents

[Declaration of Joint Authorship iii](#_Toc508618339)

[Approved Proposal v](#_Toc508618340)

[Executive Summary v](#_Toc508618341)

[Background vi](#_Toc508618342)

[Concluding remarks vi](#_Toc508618343)

[Abstract vii](#_Toc508618344)

[Illustration List xi](#_Toc508618345)

[1. Introduction 1](#_Toc508618346)

[2. Project Description 2](#_Toc508618347)

[2.1 Problem 2](#_Toc508618348)

[2.2 Rationale Behind Project 2](#_Toc508618349)

[2.3 Project Scope 2](#_Toc508618350)

[2.4 Requirement Specifications 3](#_Toc508618351)

[2.4.1 Database 3](#_Toc508618352)

[2.4.2 Mobile Application 3](#_Toc508618353)

[2.4.3 Software 4](#_Toc508618354)

[2.4.4 Hardware 4](#_Toc508618355)

[2.5 Project Overview 5](#_Toc508618356)

[2.5.1 Bill of Materials 5](#_Toc508618357)

[2.5.2 Time Commitment 5](#_Toc508618358)

[2.5.3 Mechanical Assembly 6](#_Toc508618359)

[2.5.4 PCB and Soldering 8](#_Toc508618360)

[2.5.5 Power Up 9](#_Toc508618361)

[2.5.6 Unit Testing 9](#_Toc508618362)

[2.5.7 Production Testing 10](#_Toc508618363)

[2.6 Problems Encountered 11](#_Toc508618364)

[2.7 Approaches 12](#_Toc508618365)

[2.8 Walkthrough of System 13](#_Toc508618366)

[2.8.1 Server 13](#_Toc508618367)

[2.8.2 Phone Application 13](#_Toc508618368)

[2.8.3 Website 13](#_Toc508618369)

[3. Progress Reports 14](#_Toc508618370)

[4. Conclusions 15](#_Toc508618371)

[5. Recommendations 16](#_Toc508618372)

[6. Technical References 17](#_Toc508618373)

# Illustration List

2.3a UML Diagram. p.2

2.5.3a PCB Schematic p.7

2.5.3b PCB Layout p.8

# 1. Introduction

The task of being able to deliver amplified quality sound while being able to connect a device or sensor to this type of device requires specific hardware setup. To this end, we have developed an integrated solution by creating a speaker bonnet which acts as the output to the FM tuner sensor by picking up the local frequencies and producing them in an amplified state.

Our system will have a database that will have local radio station frequencies which will be accessible via an android application and Raspberry Pi.

Despite possible redundancy of our system due to other products capable of playing FM radio stations, and other amplified speaker created and used, we nonetheless feel that our system will be unique in the sense that we are combining both sensors in a way that has never been done before.

Due to both sensors needing to be connected to specific pins, our system needed both sensors to be able to interact together without any conflicting issues, which we solved by creating a custom PCB board as well as acquiring a header in order to have the appropriate connection.

We will make all source code for our Android Application using Android Studio and in addition we will be using Python in order to communicate to the Raspberry Pi available.

# 2. Project Description

## 2.1 Problem

## 2.2 Rationale Behind Project

## 2.3 Project Scope

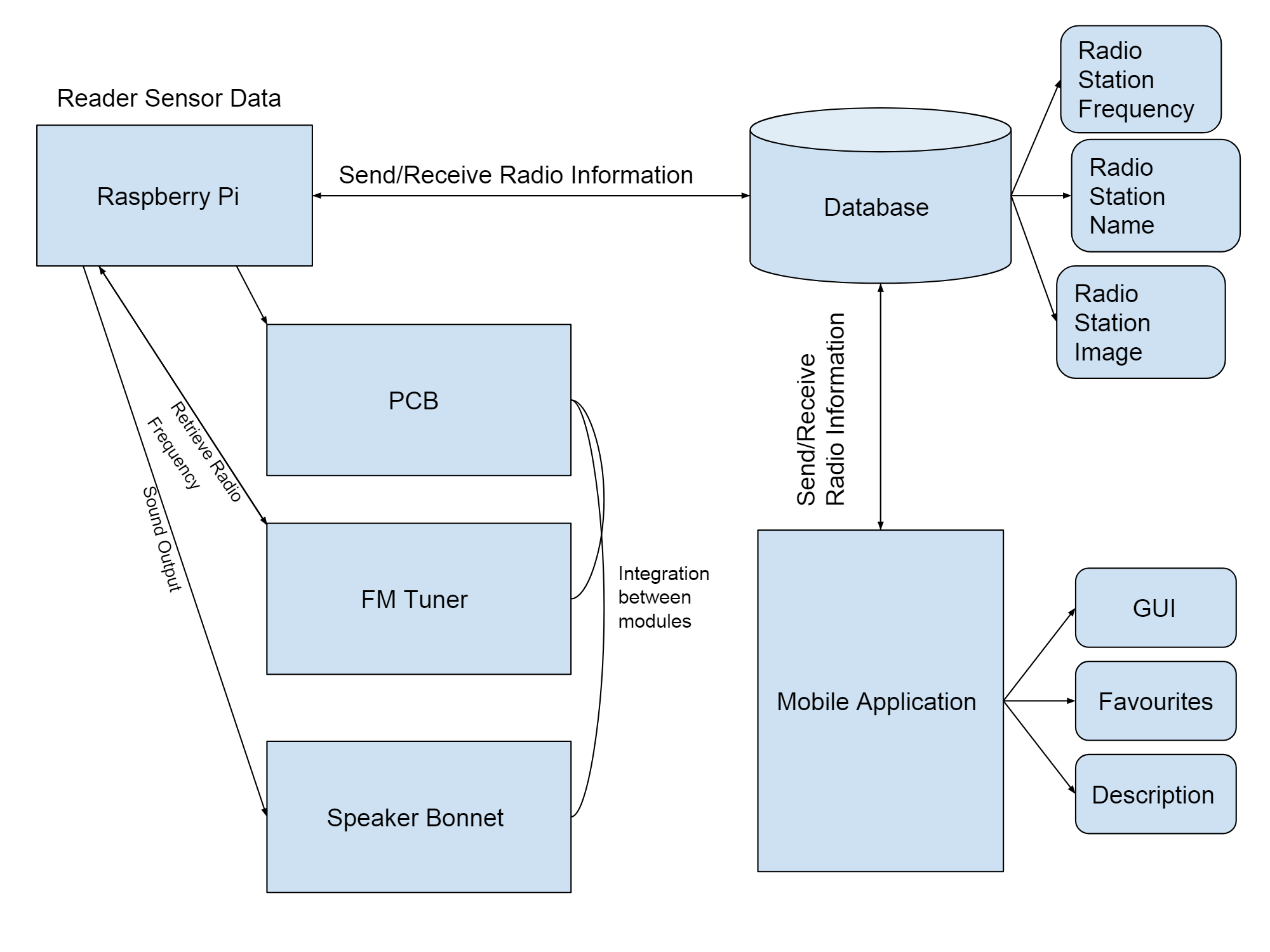


Image 2.3a UML Diagram

## 2.4 Requirement Specifications

### 2.4.1 Database

There has been a Firebase database created in order for this project to store different values for multiple radio frequencies. This database holds a variety of frequency information regarding each station in relation to the FM Tuner evaluation board. The database contains a single table for each radio frequency (or station), and the fields in the table include details such as the name, frequency number, and a description or logo of each FM radio station. Dummy data has been stored so far as the connection between the database, app, and Raspberry Pi has yet to be established. (Developed by Ryan Antolin)

### 2.4.2 Mobile Application

There has been a mobile application (currently only available on Android platforms) created which will take the data from the database and correspond it to the appropriate FM radio station to display all necessary information when selected. This will trigger the FM tuner to find the station that is selected to play and a corresponding image will be taken from an online source (with copyrights) according to each station. The application created is setup with a selection of different radio stations that currently play through hard coded radio streams. The favorites portion of the app has been created that allow users to favourite certain radio stations by using shared preference. The main focus now is retrieving frequencies from the database in order for them to be played through the Raspberry Pi. This will remove the hard-coded links for each radio station and will be using the FM Tuner board as it’s source for playing radio stations. (Developed by Aldo Ndreu)

### 2.4.3 Software

The software aspect of this project will combine the code used on the FM Tuner with the drivers installed specifically for the speaker bonnet. This part of the project requires the python code to be taken from the FM tuner sensor and be combined with the speaker bonnet. The speaker bonnet uses an installation process along with a specific script in order to run and play sound which is then integrated with the python code used for the FM tuner sensor. Once the hardware has been correctly setup and the correct code has been applied, they should perform the appropriate task which is getting the frequencies from the FM tuner and audio being outputted through the speaker bonnet. (Developed by Aldo Ndreu)

### 2.4.4 Hardware

The hardware implemented for this project has been specifically designed and created to connect both sensors being used onto a single platform. Both sensors require specific pin connection on the Raspberry Pi which is why a fritzing diagram and other layouts have been created in order to brainstorm the connection procedure. A custom PCB board has also been created in order for the connections to align according to requirements and two header pins are used in order to connect both sensors into the appropriate slots. Once all the appropriate procedures have been followed, there should be a successful connection by both sensors. (Developed by Erick Cantos)

# 2.5 Project Overview

### 2.5.1 Bill of Materials

1x Spark Fun FM Tuner Evaluation Board Si4703 – $28.99

1x Straight Break Away Headers - $2.99

1x Adafruit Speaker Bonnet for Raspberry Pi – $12.95

1x Raspberry Pi starter kit - $99.99

1x 8 Ohm 3-Watt speaker - $1.95

2x 22 AWG gauge electrical wire (3 cm) – $14.50 (10m / 32ft)

2x Alligator clips - $7.73 (Pack of 10)

1x Female connection header 8 positions .1” TIN - $0.87

1x Female connection header 40 positions .1” TIN - $3.49

1x Stacking Raspberry Pi header 40 positions - $3.94

Product prices are subjected to change and will vary over time due to currency changes and/or

supplier price change.

### 2.5.2 Time Commitment

After the acquisition of the required materials, it is expected that at least five hours of time

must be allocated in order to reproduce the project if the code included is used. This estimate

on completion takes into consideration that the person building the project has adept

knowledge on soldering and programming. However, if the code is developed yourself more

time may be expended into completing the project. The importance of correctly ordering the

parts and thoroughly following the build instructions minimizes the opportunity to create

any project errors or possible delays.

### 2.5.3 Mechanical Assembly

FM Radio Tuner:

1. Prepare the Spark Fun FM Tuner Evaluation Board Si4703 by soldering the straight

break away headers onto it. These pins will be connected to the Si4703 GPIO terminals

1. Once the pins have been soldered onto the Si4703 board, plug in a set of

headphones or Speakers with a 3.5mm jack into the Si4703 board.

Speaker:

1. First, start with the preparation of the Adafruit Speaker Bonnet by soldering the twenty

position female header onto the GPIO terminals located on the border. It is considered

easier to solder the rest of the pins if you begin by soldering the first two pinouts and

the last two to keep the header in place while you solder the rest of the pins.

1. When the soldering is complete, take the two pieces of electrical wire and screw one

wire into the positive terminal on the right speaker terminal and the other wire into the

negative terminal on the right speaker connector. If you wanted to use an additional

speaker, it could be wired to the left speaker terminal using the same method.

1. Lastly, connect one end of one alligator clip onto the electrical wire on the positive

terminal and the other end will connect to the positive terminal on the speaker. The

other alligator clip will hook onto the electrical wire on the negative terminal which will

then connect to the negative terminal on the speaker.

Printed Circuit Board:

1. The schematic for the PCB is provided within the Project Files Folder. When

designing/replicating the diagram consider that the PCB must fit the raspberry pi. The

recommended size for the PCB should reach 55mm in width and 57mm in height to sit all

components comfortably together. This is first adjustment that should be completed prior

making the design.

1. After the PCB size orientation has been completed, connections from raspberry pi to

header connections must be implemented. The schematic below should be a reference

on how the circuit is connected.

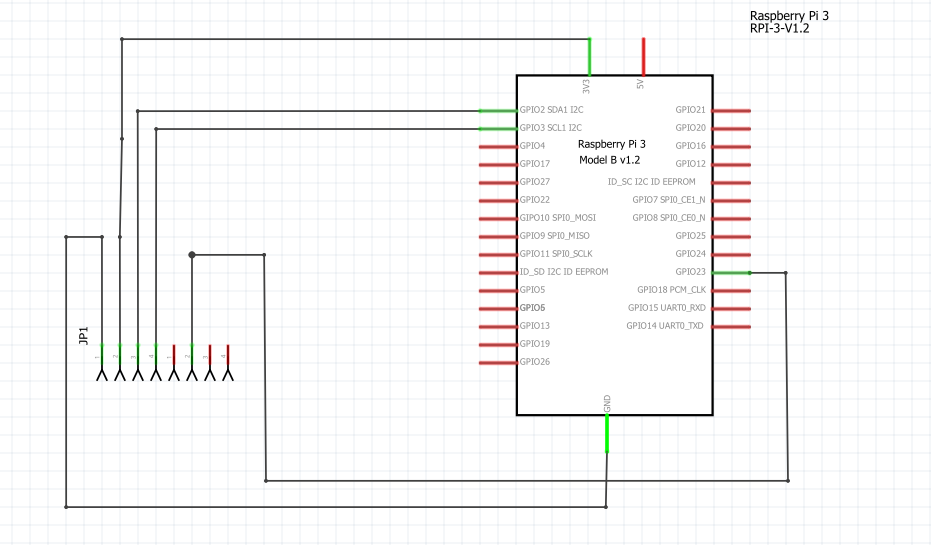


Image 2.5.3a PCB schematic

Five out of the eight pins are used on the JP1 header which are SCL, SDA, GND, 3v3, and

GPIO23. These pins will enable sensor reading and power up to the FM Tuner.

1. Following the schematic should come the PCB layout. Ensure there is enough space

between the raspberry pi connections and the header for the FM Tuner prior to submitting

Gerber files for manufacturing.

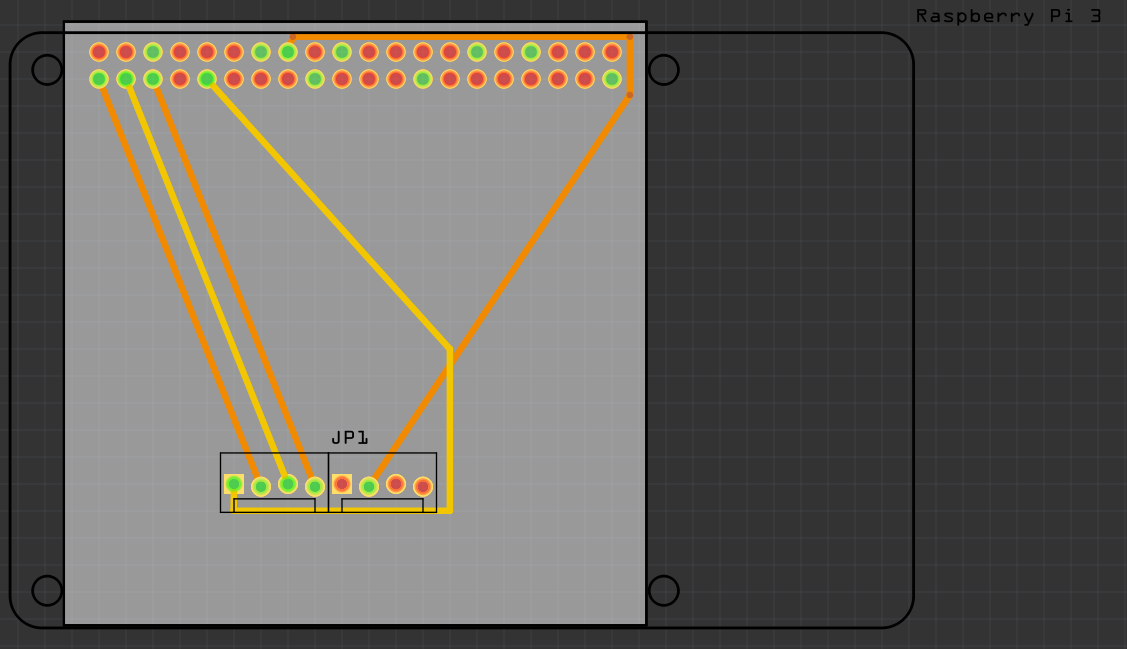


Image 2.5.3b PCB layout

1. The printed board will now be prepared to have the header connections soldered on.

Solder on the eight-pin header on top of the PCB following up with soldering the Raspberry

Pi stackable header underneath the Raspberry Pi GPIO.

1. Finally, complete the assembly by aligning the stackable header portion of the PCB to

sit comfortably on the Raspberry Pi. Seat the speaker bonnet directly on top of the

stackable header. Lastly, take the FM Tuner and seat it into the eight-pin header across

from the speaker bonnet. It is now ready for software testing.

### 2.5.4 PCB and Soldering

The PCB will provide a platform that will enable the integration between the speaker

bonnet and the FM tuner. Both these devices will be mounted onto a single board so that

communications can be established between them and progress their functionality together.

Depending on the type of software development tool that you use, some traces that are

connected on the top side of the board may have to connected manually to the to bottom

of the board and header pins. If this is the case then it is suggested that you acquire electrical

wire with about 18 mm in diameter and solder the wire onto a trace with enough wire to pass

through a pinout hole. Once the wire is through the pinout hole solder the header pin and wire

onto the board at the same time.

### 2.5.5 Power Up

First, if you are booting for the first time you will need to insert a micro SD card into the micro

SD reader on the Raspberry pi. The micro SD card must contain an image of the Raspberry pi

NOOBS software so we can install an operating system and can be downloaded from the link:

https://www.raspberrypi.org/downloads/noobs. Plug in the power, boot to the Raspberry

Pi, and select one of the Raspbian preinstalled operating systems. After the installation of the

operating system you must configure basic settings before anything else, these settings include

configuration of the internet, and VNC viewer. Lastly, users should also have I2C enabled in

order to complete this project. Follow these steps to enable I2C:

**Start> Raspberry Pi configuration > Interfaces and ensure I2C is enabled.**

### 2.5.6 Unit Testing

A reminder that the FM Tuner must have a 3.5 mm headphone connected to it for proper

testing. After the verification of headphones, we will need to ensure that the FM Tuner sensor

is operating correctly by opening a terminal on the raspberry pi and entering the command

“sudo i2cdectec -y 1”. The use of this command will display an output of many addresses and

you must ensure that the sensor is connected at address 10. Afterwards the correct python

code must be compiled to test functionality of the sensor which can be found in the Firmware

portion of the technical document. Save the python file as sudo yourfilename.py and compile

the program within the terminal using the command “python filename.py”. This command

will compile and run the program if there are no errors and will be executed.

Moving onto the Speaker bonnet, you are required to install the drivers and acquire the scripts

through the Adafruit website. There is an alternative which is using the advanced version but it

is not recommended for a novice user. To run the script, use the command “curl –

sS https://raw.githubusercontent.com/adafruit/Raspberry-Pi-Installer-

Scripts/master/i2amp.sh | bash”. You will be prompted to reboot your Raspberry Pi device

after the installation. You will be able to use the speakers once the Raspberry Pi is rebooted.

Finally, you must open a terminal and download mpg123 which will be the media player for the

device. You can do so by running the command “sudo get-apt install -y mpg123”. Personal

music can now be playable via mpg123. On a side note ensure files do not have spaces and

they are replaced with dashes. Spaces on UNIX based operating systems are not comparable to

spaces on the Windows Operating Systems.

# 2.5.7 Production Testing

If the program for the FM Tuner runs successfully without any errors then the output will

display multiple registry values and prompt the use to enter a radio station. As an example,

if the user would like to listen to the radio station 99.9 Virgin radio, simply enter 999. This is

programmed to run in a loop so that every time the user is prompted to enter a radio station

without having to re-run the program. If the FM Tuner is not providing any registry values,

ensure that the sensor is seated correctly onto the PCB. Ensure I2C is enabled at all times.

# 2.6 Problems Encountered

# 2.7 Approaches

# 2.8 Walkthrough of System

### 2.8.1 Server

### 2.8.2 Phone Application

### 2.8.3 Website

# 3. Progress Reports

# 4. Conclusions

# 5. Recommendations

# 6. Technical References