Raspberry Pi fm_transmitter_master
Hacked Version of PiFM

1 MHz to 200 MHz (original software range was around 70 MHz to 108 MHz)

The following is a guide to Installation, Setup, Uses, and Possibilities.

NOTES *– The fm_transmitter-master software is NOT my own work - I have cloned a version that wortked for my set-up, made few little tweaks, and redistributed it with PROPER instructions.*

This is NOT a practical method of Radio communication - this particular method transmits a square wave - in short it produces an unstable wave that will 'spread' unwanted RF across other frequencies. There are a LOT of Raspberry Pi boards being developed that have the proper RF filtering - and signal generator software that can produce triangle, sawtooth, and sine waves as-well as square waves - I have not tested extensively with an oscilloscope.....yet.

You can use this software to transmit .wav files in mono and stereo format. My testing has not required stereo use.

Due to the square wave emissions causing "spread" this may interfere with nearby electrical equipment.

A 20 cm wire on GPIO pin 4 will produce good results. That is ALL you need other than the software. See my antenna notes and designs for better designs and range.

CONTENTS

Ben Woodfield raser.13@hotmail.com fm_transmitter-master

What is it?

This project uses the general clock output to produce frequency modulated radio transmissions. In lamens terms you can send signals to your GPIO - to specified frequencies
The main code for this program is C. it allows a lot deeper access to hardware on this level, and can utilise the clock output very efficiently

This is basically another version of the pifm software, and was originally used by ICROBOTICS

http://icrobotics.co.uk/wiki/index.php/Turning_the_Raspberry_Pi_Into_an_FM_Transmitter

This version does not utilise a DMA controller in order to distribute samples to output (GPIO clock generator), so the overall sound quality will be lower than the original PiFm project and only mono transmition is available but this makes possible to run it on all kind of boards.

Very basic operation, a lot of code had to be removed or edited from the original software but it has been "hacked" to allow access across ALL Pi models. And also has unlocked frequency range. Opening up the possibility for Amateur Radio usage - with the correct modulation and filtering - in time I hope people will build upon these things. At the moment the software transmits without the basic ID info that used to be sent and displayed to the end-user / receiver.

I plan to use this software as a remote-beacon type of thing. Very similar to PiFox – where a Pi is pu into a fox-hunt style scenario for antenna design competitions – a Rpi is set-up somewhere, and left transmitting a signal at a specific frequency, directional or loop antennas are designed and constructed to find the source of the transmission. Some Amateur Radio clubs have competitions doing this.

I have included other software and written some simple porograms / documents to accompany this program.

- An antenna length calculator
- Convert audio to Wav instructions (mpg321)
- Improved Antenna Design PDF

Some Basic results

Heres what worked: (using JUST a 20 cm wire on GPIO 4 as an antenna)

COMMERCIAL STATION FREQUENCIES

From around 70 to 108 (You will have weaker results If you choose a frequency that's already being transmitted on - your little Pi will be competing against Commercial Broadcasting Transmitters and Amplifiers)

No ID message shows up on the receivers device with this version of the software - no digital information at face value....meaning it won't show any info to the average radio user...
However if you use this software stupidly there are encryption procedures built in for legal reasons.

If you test the frequencies with just a 20cm wire on GPIO 4 and you wont get more than 10 meters or so. So for basic range testing you wont be annoying too many people

FREQUENCIES TESTED AT TIME IF WRITING

CB (citizens Band) Around 27 MHz
Radio tested - Harrier CBX, RF Pre-amp, 13.8v 20 Amp PSU.
CB radio was in the same building, but my 26 ft antenna is way up high on a 3rd floor roof.
I can't confirm whether it is picking the rf through the coax or theough the antenna.

VHF (Very High Frequencies) and UHF (Ultra High Frequencies)
Radios tested - Baofeng UV5-RA and Baofeng BF-F8+.
Transmitting audio from the RPi within my handheld radio's entire frequency ranges -
130.000 MHz to 176.000. MHz – some frequencies better than others.
The UHF – 400 to 520 MHz range needs further testing - odd and unpredictable results.

Commercial Radio Frequencies
From around 70 MHz to 108 MHz FM.
Picked up on Both Baofeng Handheld Amateur Radio's, Nokia Mobile Phone Radio, Tecknik. docking station stereo.

Software Download

Clone the software from my github account. There is no apt-get install sorted yet, so you will have to go to:

https://github.com/raseribanez

Find the software repository for this project on my page
Click clone / download - you will be given a .zip package

It should :

- Run on all versions of Raspberry Pi
- Read both mono and stereo
- Read data from stdin

Software Installation

Unzip or extract the package into Downloads or your chosen folder - if you right click the zipped file and "extract" it will make a new folder in the same directory as .zip file
I like to "extract to" another location than downloads – but for simplicity I am explaining based on the .zip file being extracted into the Downloads folder

Either right click the zipped folder and select *Extract here*

Or unzip from within the terminal with:

```
$ unzip fm_transmitter-master.zip
```

you shouldn't need to provide a destination folder – unless you want to. The following steps assume the .zip folder was extracted to the Downloads folder. After the above command you would have an

unzipped version of the folder in Downloads.

Open a command terminal and enter the following commands:

```
$ cd Downloads              # Navigate to the folder of extracted fm_transmitter-master.zip
                            # - edit the aove command if your .zip isn't in downloads
$ cd fm_transmitter-master      # Change into the fm_transmitter-master directory
$ sudo apt-get install make gcc g++       # Compile the C source code
$ make                                    # Make the required installation changes
```

Installation should now be complete.

You have to run the software (transmit) from **within** the fm_transmitter-master folder. If you've just installed it you should be in the right directory

Basic Usage

When you have completed the installation you can test it works with the following:
sudo ./fm_transmitter [-f frequency] [-r filename.wav]

So for example....the package comes with a StarWars theme tune....star_wars.wav - to play this tune over radio frequencies do the following:
$ sudo ./fm_transmitter-master -f 92.0 -r star_wars.wav

This transmits the .wav file at 92.0 MHz FM. Very basic in MONO

If you want to change the audio file to another one:
You need to convert your audio to WAVE format to play

To play a mono .wav with some more options, use the following to read audio data from stdin:

```
$ sox star_wars.wav -r 22050 -c 1 -b 16 -t wav - | sudo ./fm_transmitter -f 155.5
```

I have noticed that if you move the decimal point over you get more accuracy – so for 155.500 MHz, use 15.50. This is not my doing, it may just be RF related, or there is a decimal point error somewhere in the coding. You will have to play about with what works for you on different frequencies....EXPERIMENT!

To use a USB sound-card type this:

```
$ arecord -D hw:1,0 -c1 -d 0 -r 22050 -f S16_LE | sudo ./fm_transmitter -f 100.6 -
```

If you get a warning saying `buffer overflow` and you find disturbances in the transmission – slowing and breaking up, then you can try:

```
$ arecord -D plughw:1,0 -c1 -d 0 -r 22050 -f S16_LE | sudo ./fm_transmitter -f 100.6 -
```

MP3 to wav Converter

This is a simple and lightweight mp3 to wave converter for Raspbian. There are many out there, this one has comlete simplicity – even if your a newbie to terminal commands – that is why I chose to use it for my tutorial. You can easily look up other software for this task – others have a lot more options and formats available. This program runs from the command terminal and will convert a SINGLE audio file to .wav

Install with
```
sudo apt-get install mpg321
```

command to run the converter (from terminal) CONVERT TO WAV
```
mpg321 -w track.wav track.mp3
```

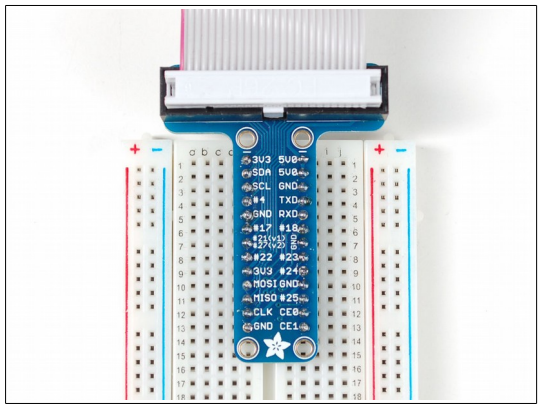
Replacing `track.wav` for the name you want the converted audio to have, and `track.mp3` for the name of the original mp3.

Hardware

As you may know by now, to transmit efficiently over more than a few cm's you need to utilise a GPIO pin as an antenna. All that is needed technically is a length of wire – about 20 cm – connected to GPIO pin 4, or number 7 as you look at the Pi. This can get your transmissions upto100 meters, depending on a few factors, such as the frequency used and obstacles between the transmitter and receiver. I can get about 10 meters through walls – to next door, or 100 meters through open space, providing the receiver has a suitable antenna for the frequenciy you transmit over.

Antennas can be improved to greatly improve performance, end audio quality, and range.

I use a breadboard, connected to my GPIO pins via a cobbler / breakout kit – like below but with more Pins (for a newer Pi) and it allows safe and efficient access to the GPIO pins n a range of projects.



More sophisticated antenna designs use a ground. A simple dipole antenna can be made for the Pi, using GPIO pin 4 and a Ground pin. The Rpi only has 5Vdc so improving the antenna design will achieve the maximum performance for your power rating. I have no idea if further amplification can be achieved, but as mentioned there are add-on boards for the Rpi – that have some astonishing RF capabilities. I hope to see a huge advancement in Amateur radio using Raspberry Pi's.

I have written a basic antenna calculator and included it with this project. Although I said this project works with a simple wire of around 20cm, this is a very crude antenna, it is only really good for testing that the software is configured to the GPIO pin before designing an antenna.

You have to take a frequency range that you want to use, then calculate the average frequency within that range (mid freq), then when you have a frequency you have to calculate it's wavelength, and decide what ratio or size you will use. Sometimes it is impractical to make a 10 meter full wavelength antenna, so the length is cut in half, or 5/8ths for example and this length is used.

I have a simple design for an adjustable antenna – pairing with the antenna length calculator, and another very simple design for a dipole for the 70 MHz to 108 MHz band. I take 2 vehicle antennas and construct a dipole – that can switch between horizontal and vertical.

Before you go ahead and knock up an antenna please see my guide – you should take care to use the proper connections and cable to provide the correct load of 50 ohms. Normal household coax cable will work but with limitations. I use RG8 and RG58 coax for my radio equipment. The braided shield is a LOT thicker, providing adequate ground supply.

If you don't have decent coax, I have used a guitar lead for an antenna cable.

Both antennas will be covered fully in seperate project PDF's