**Brief of technology**

Project Name : **Decoli - DEvice of COgnitive Listening**

Author: Suryasaradhi (Btech Eng.Physics 2018 - 2022)

*Explanation of the technology:*

The technology implemented here is a basically a digital radio that is very compact and portable, This radio is able to pick up Radio frequencies from 500khz – 1.7Ghz. It is able to translate unknown complex signals to human understandable speech. Furthermore our technology has integrated cutting edge AI, Cloud services allowing it to provide the user with more advanced features like, Audio translation, Modulation Detection, Signal Decomposition etc.

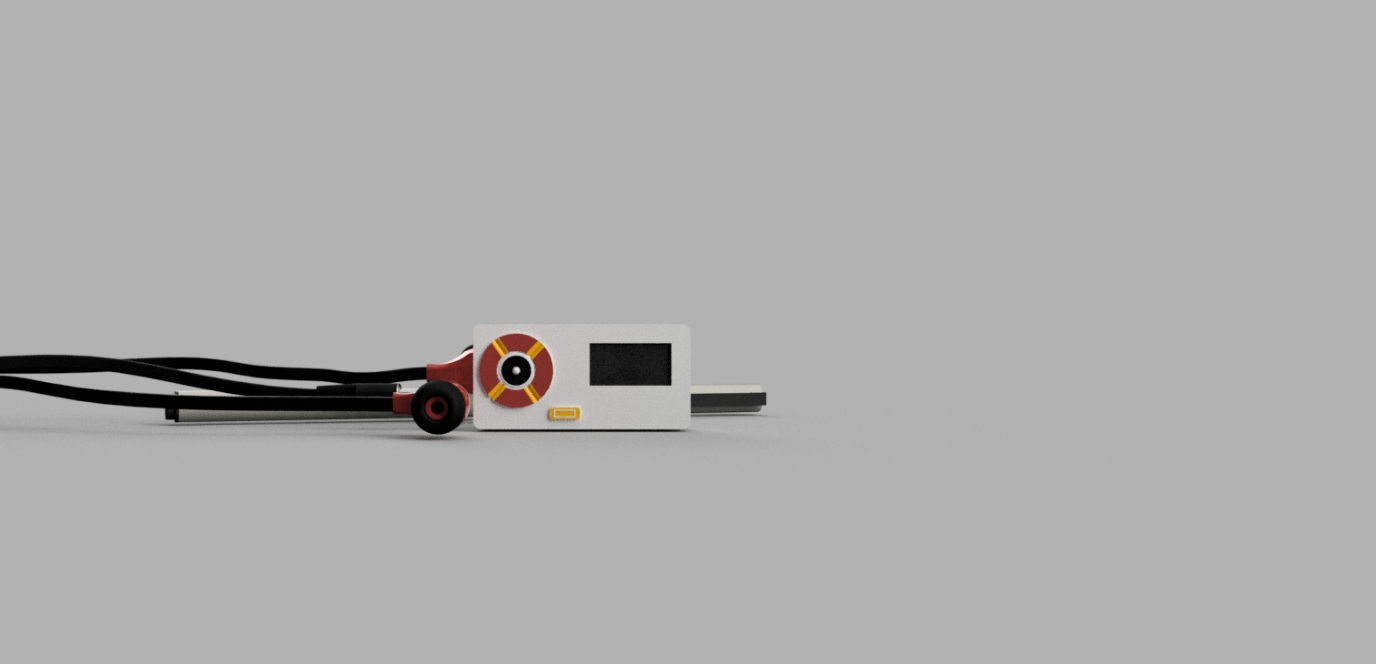
See Realtime updates of the project at – [**https://github.com/thesunRider/decoli**](https://github.com/thesunRider/decoli)

* My Approach

I had the idea of starting out with only covering a small range of frequencies in the SDR domain, a Typical RTL sdr covers 24Mhz – 1.2Ghz perfectly so I thought of using an RTL-SDR with a raspberry pi zero Wireless, which was already compact in size, I added a display module ,a battery management system and a few navigation buttons. Hence born was the first prototype. This was the starting point of my creation. I wanted to make sure that I wanted to make my code portable So reusable code was used for programming raspberry pi which would remain same in any embedded Linux processors. The problem with using products like raspberry pi and RTL sdr are they are copyrighted for use and they are huge in size. So I started working on creating a single board capable of the work of RTL SDR and raspberry pi, Hence the second advanced prototype of Decoli was envisioned.

* How it works

Our Innovation is basically a Portable SDR (Software Defined Radio) with some extra buffs. We are planning to build a compact device of the form factor of an ipod.



In our system We are using a RTL2832 chip as the DSP, A R820T chip as the Tuner , a NUC980D as the processing unit with an Esp8266 acting as the Wifi card which provides Internet.

Diagram

Description automatically generated

Figure 1 Data flow in Decoli

R802T amplifies the incoming signal, and mixes it down into a lower frequency 3Mhz, and feeds it to RTL282

Advantages of the technology:

The innovation’s uniqueness

Use cases:

*Technical Details and Complexities*

* Antenna

For Implementation of antenna we have gone with a monopole where a wire will run through the earphone the user puts in, We have attached a SMD SMA connector if the user wants to connect to a better antenna and receive in better bandwidth. An external amplifier can be directly connected to the antenna due , 150ma of current can be drawn from the Antenna’s bias tee after which the LDO’s supplying the current may overheat.

* Battery

We have three different processors running in the board in parallel with more than 8 accessory chips. We have estimated for the power usage to be around 1W for the entire board at 3.7V battery supply. Hence -

Battery Voltage: 3.7V

Battery Capacity: 1000mAh = 1000mAh

Efficiency : 80%

Battery Life (FULL USE): \*\*Discharging Time = Battery Capacity x Battery Volt / Device Watt.\*\*

1Ah x 3.7V \* efficiency / 1W(usage-max) = 2.9Hr

We are expecting a battery back up of 3 hours for a 1000maH lipo battery which should well fit in the device.

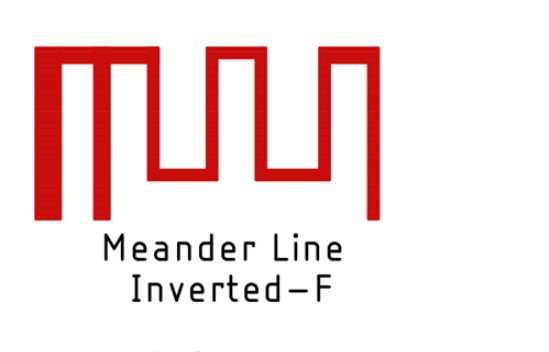
* Extra Additions

I have implemented **Bias Tee** , **Direct HF sampling** , **<1ppm TCXO crystal** which negates any temperature related frequency drifts, **Left LNA provisions** in circuit, etc as part of my own feature additions. Furthermore we have direct control over the R820T registers apart from the normally available 4 registers. This is achieved by directly extending the i2c bus over to the microprocessor from the R820T chip. I have implemented feedback AGC between RTL2832 and R820T which allows for better dynamic range of tuneable frequencies. All these additions can be seen in the circuit

* Circuit Schematics

I have created a 6 layer PCB for the device, There are three faces to the board, The SDR one,The Microprocessor one and the wifi one. Each having their on Complexities in design.

We had to create 4 layers of alternating powerplanes which support the SDR layer,with matched impedance in the surface. We have RF routed with impedance match in consideration to avoid signal reflections.

For the wifi processor we created a *Meander Line Inverted F antenna* and Impedance matched it with the ESP8266, thus Wifi was made available , ESP8266 interfaced over with the microprocessor through i2c.

For the design ,all these chips are available as separate evaluation modules and can be tested together before being soldered into the single motherboard. We have individually tested for the functions of each chip.

Diagram, schematic

Description automatically generatedDiagram, schematic

Description automatically generatedDiagram, schematic

Description automatically generated

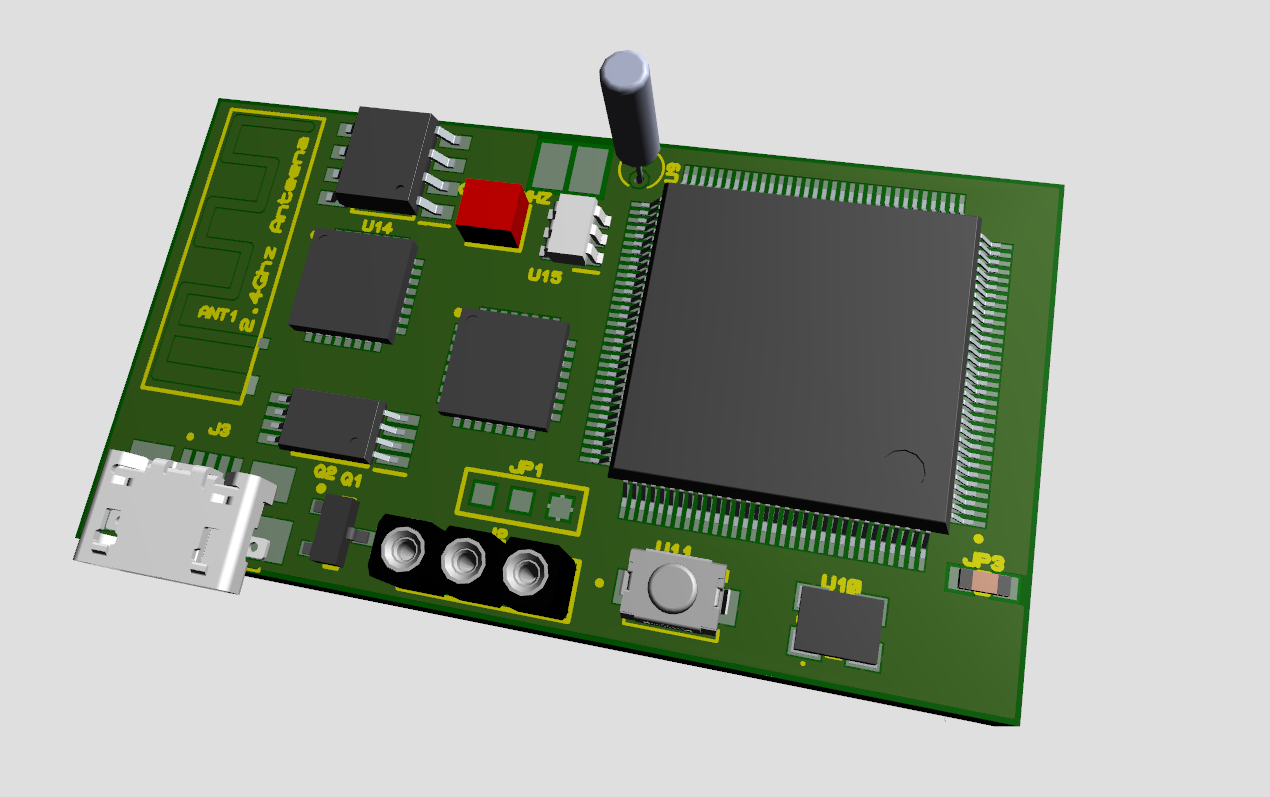


Figure 2 PCB 3d View

* Software

We have booted to LEDE Linux distribution using NUC980 OpenWrt, For programming we have opted to use python, As this is related to numerical data management And python has ready library available for RTL2832chips called as the pyRTLSDR. We have used GCP in python for speech translations. The GUI for Oled display was created using GUISlice,which helped in creating modular code for displaying.

We have used URH (universal Radio Hacker) for analysing incoming signals,to decode and interpret them. leveraging its python abilities underneath is still under works.

For Identifying modulation schemes of unknown data , We have used tensorflow for microcontrollers a light weight implementation of tensor flow to run a SVM based classifier which showed 60% accuracy results when it was trained on 20,000 signals ,128bits ,2 second in length. On the software end we haven’t moved forward with decoding random signals automatically currently AM and FM gets decoded and played back.

Once our hardware strengthens we will move to software to accomplish random signal decoding tasks.

A picture containing indoor

Description automatically generated

Figure 3 Decoli Side view