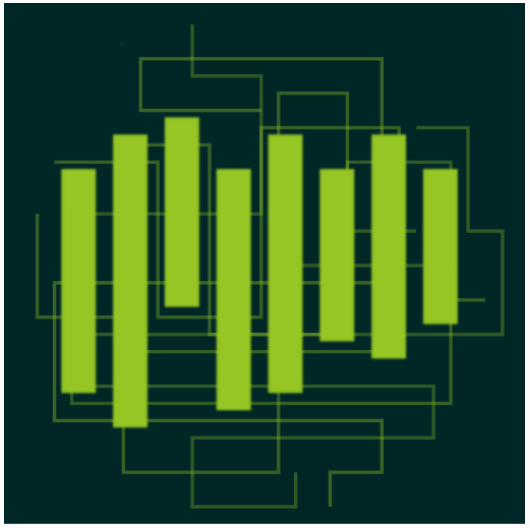
# SignalID

## Automatic Radio Signal Identification

**(V1.2)**



Neosama, TORTILLUM

@Rafios06
@tortillum

# Introduction

**There's a plethora of radio signals.**

**From simple Morse code to at least obvious radar signals...**

**An automatic classification/identification system of its signals would be a very useful tool.**

**SignalID is an Android application, but the algorithm used (Salome) can be implemented elsewhere.**

# How to use the Android application

**The interface is like Shazam.**

**There are two buttons useful for identification.**

**The main one, the one that moves, is to start the search and the second one is to indicate if the signal comes from the HF band or not.**

# How to use the Android application

**Once your receiver is well adjusted (frequency and bandwidth)**

**Place your Android as close as possible to the sound output.**

**In a quiet sound environment (no TV or car noise ...)**

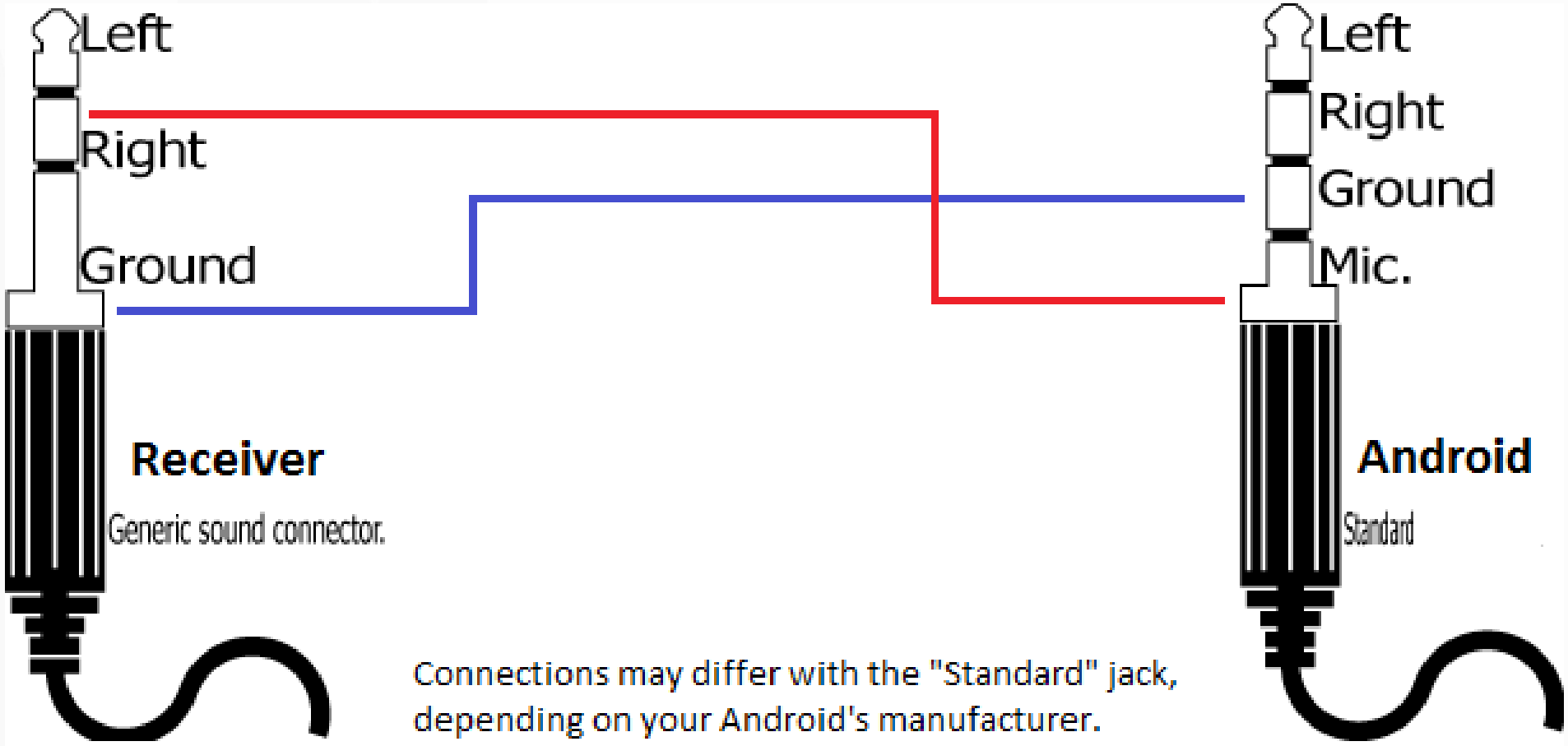
**Ideally make/purchase a cable jack (see schematic)**

**To connect the sound output directly to the microphone input.**

***Demonstration (video) :***

**https://youtu.be/E-y1Ts1q2RU**

# Schematic - cable jack



# How to use the Android application

**Next, indicate whether the signal comes from the HF band (0-30MHz) or not.**

**Press the big button and wait 5 seconds near the audio source.**

**There, it's done!**

# Reliability

**It is not yet very reliable with all signals, but I can easily recognize for example the RTTY and STANAG 4285 signals with.**

**Because they are the most accessible to listen to.**

**The use of the cable jack mentioned above greatly improves the results.**

# Technical Information

**The algorithm (Salome) uses as input WAV files.**

**With as parameters :**

***SampleRate = 44100***

***Format = 16-bit PCM***

***Duration = 5 seconds***

# Technical Information

**Then an FFT is performed thanks to the external library,**

**Jtransforms (https://github.com/wendykierp/JTransforms)**

**Then compare the result obtained with those present in the algorithm.**

# Technical Information : FSK

**For RTTY (85Hz_commercial) the characteristics are :**

- **First tone at 1040 hertz**
- **Second tone at 995 hertz**

**If both tones are dominant over the others, it is considered that the signal is probably RTTY.**

**This works for FSK signals.**

# Technical Information : OTHER MODULATIONS

**The principle is the same, I take the highest and lowest tone of the signal.**

**There is then a "score", incremented each time it encounters a tone (dominant, not noise) that lies between the highest and lowest frequency.**

# Technical Information : OTHER MODULATIONS

**For 4285 (GEN) the characteristics are :**

- **First tone at 2415 hertz**
- **Second tone at 1405 hertz**
- **SCORE between 25 and 35**

**If all the criteria are present, it is considered to be STANAG 4285 (GEN).**

# What's next?

- More recognized signals
- Pre-filtering when taking audio with the microphone
- Possibility to use a WAV file as source
- A PC version ?

# Open Source

**The code is open, be it the Android application, the algorithm, the tool used to determine the characteristic frequencies to add new signals.**

**All the project is open source !**

**The project being open source anyone can contribute to the development.**

**https://github.com/Neosama/SignalID**

# License

**This project is licensed under the GNU General Public License v3.0**

# Conclusion

**Lots of ideas, this is just the beginning.**

# Some links

***Twitter***

**@tortillum  @Rafios06**

***Google Play***

**https://play.google.com/store/apps/details?id=com.tortillum.signalid**

***Github***

***https://github.com/Neosama/SignalID***