Skywave Documentation

Zotero | Github | Drive

March 17

Done this week

- Brainstorming: see documentation files, skywave_brainstorm.jpg
    - Listed our project's core concepts/keywords and their relationships to one another
    - Refined more key ideas
    - Focused on developing concepts of the material/tangible aspects of the project along with thematic concepts
- Continuing work on our final report and presentation

March 24

Done this week

- Bought and started reading Technosis
- Proposal report & presentation (See: Proposal Folder)

Glitch Feminism & Chronogenica

- Regarding the theory & theming behind this project, Glitch Feminism provides excellent inspiration; tying in to our themes of witchcraft, coexistence with humans and non-humans, and sisterhood.
- Chronogenica provides similar inspiration in their co-op consisting of mechanic workers, with rights and employment benefits working in collaboration with human colleagues.

Outputs

- Looking to refine between digital and analog signals.
- Digital:

        - PSK31 (text-based protocol, “texting”) provides a data-rich a/v output in the form of a ‘waterfall’. This data is rarely archived and allows for open listening to transmissions from multiple users.
- Analog:
    - Cosmic radiation often appears as an audio/visual output in the form of noise. A percentage of noise received from radios or tvs is of cosmic origin, as many cosmic objects emit EM radiation. This noise can be received. With very precise tech, very precise readings are possible as in radio astronomy. Some of this noise is actually remaining from the big bang!
        - NASA expresses some telescope info in the form of sonographs which communicates the data audibly. Similar data processing protocols could be used to produce a data rich A/V output.

Readings

- NASA Sonographs
- Legacy Russell, Glitch Feminism
- Chronogenica Co-op
- Davis, Erik. 2015. TechGnosis: Myth, Magic, & Mysticism in the Age of Information. Berkeley, [California]: North Atlantic books.
- Schröder, Frank Gerhard. 2012. Instruments and Methods for the Radio Detection of High Energy Cosmic Rays. Springer Theses. Berlin Heidelberg: Springer.

March 31

Done this week

- Interview with Cheyda
- Met with Sabine, Gabriel
- Proposal feedback
- PSK31 TouchDesigner demo (with retained data!)

Notes from Ceyda Interview

Readings

- https://www.vice.com/en/article/z3meny/artificial-intelligence-cult-tech-chatgpt
- Hayles, N. Katherine. Unthought: The Power of the Cognitive Nonconscious. University of Chicago Press, 2017, https://press.uchicago.edu/ucp/books/book/chicago/U/bo25861765.html.

Radio Waves and the Ionosphere
https://www.arrl.org/files/file/Technology/pdf/119962.pdf

April 7

Done this week

- Bought a radio, radio kit, radio chips
- Readings & research: technopaganism, technosis, witchcraft
- Met with Sabine, Gabriel

Hardware

- Shopping
    - Bought a radio. AM/FM SSB
        - SI-4732 radio chip
        - Potential for modifications - arduino? Maybe bypassing certain mechanisms, like the tuner to allow for sensor data inputs?
            - The radio should be capable of transmitting data to a computer via usb
        - Radio tutorial with SI4732
    - Bought Radio assembly kit from abra ^o^
    - Also bought some additional radio chips

Current Goals

- Begin recording data with the radio, outputting to td if possible
- Play with radio kit
- Continue research on biometric sensors: heartbeat, temperature, breathing
    - One emotibit available, checking with elio for available sensors

Readings

- (Techno)Paganism: An Exploration of Animistic Relations with the Digital
- Cyberspace as Sacred Space: Communicating Religion on Computer Networks
- Gods in the machine? The rise of artificial intelligence may result in new religions

April 14

Done this week

- Radio building in progress
- Researching radio electronics & hardware
- Sensors, wearables and ritual
- Continued work on research paper draft

Hardware

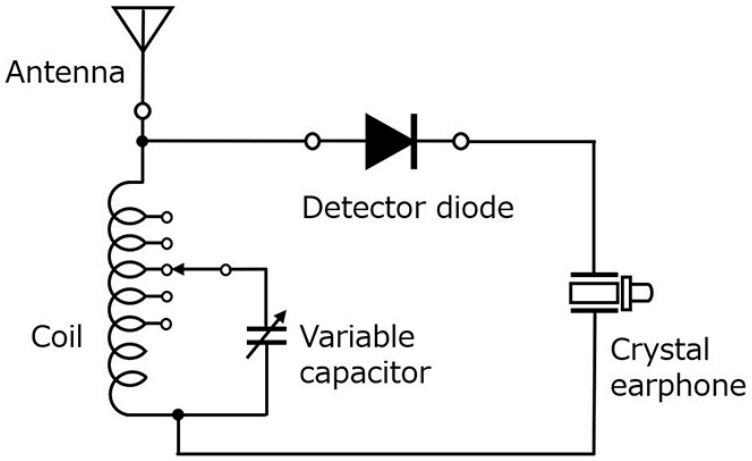
- We opened up the Si4732 radio:
    - [images]
    - The radio consists of an Arduino Nano connected to the Si4732 receiver module. It contains a potentiometer for tuning, several button inputs for bandwidth and mode, a speaker, and a display screen. It allows for an antenna connection, and allows reading via USB download or audio jack.
    - Datasheet
    - Library for Si47xx chips, documentation website
        - Code example from library for ATS-20
- Looking into ways to bypass the tuning mechanism (potentiometer) with biometric sensor data. The library appears to provide some simple functions for controlling the receiving frequency and mode directly via the nano microcontroller with the Si4732 module.
- Unsure of how best to bypass the existing protocols. The Nano may allow for shield attachment for additional modules, sensors, or we may opt only to modify the software. Seeking a method to decompile the existing code on the Nano.
- Narrowing between digital receiving (PSK31) and analog receiving (background radiation). Info on CMB (Cosmic Microwave Background) i.e., radiation remaining and redshifted to the microwave spectrum leftover from our universe's primordial era.
    - On listening to CMB
    - Notes on CMB & its discovery
    - Great video on CMB & the early universe

Radio Electronics: Research

Basic Principles of Radio

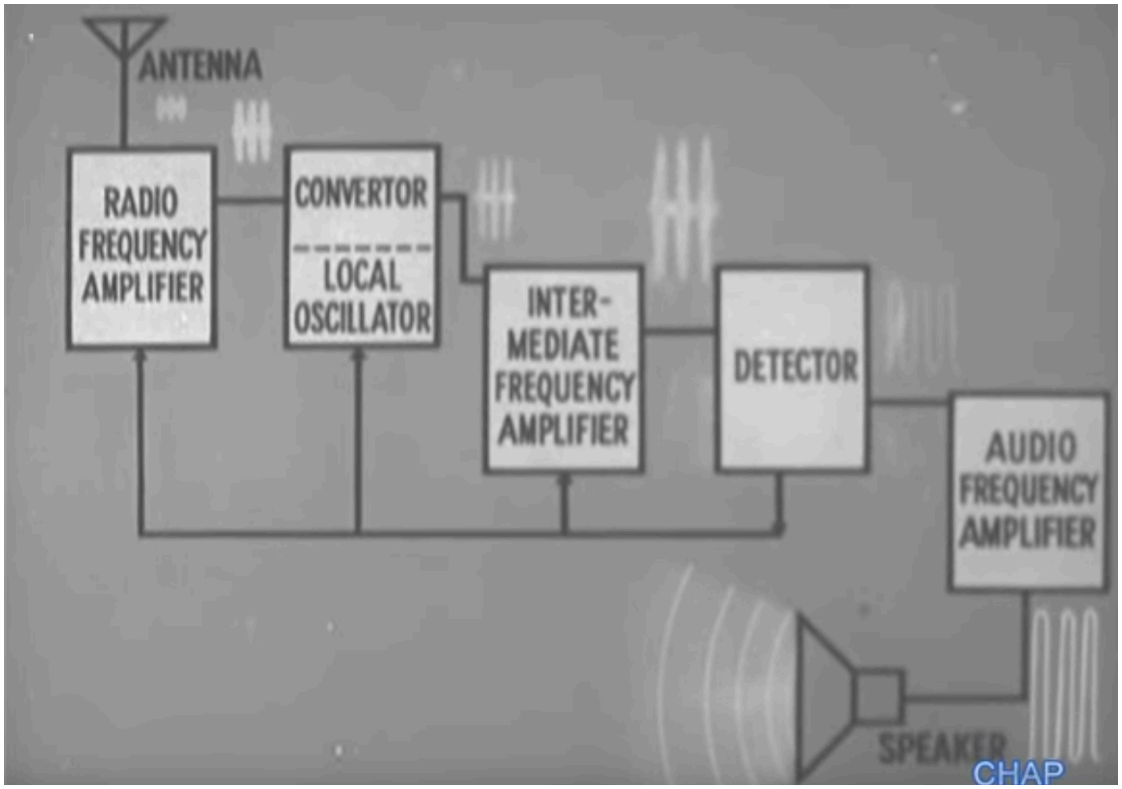
- Antenna to receive signals, tuner to select the signal to be received
- Detector differentiates high and low frequencies. One is a carrier (high), discarded. Low frequency (message) is passed to a reproducer which converts the information to an audible signal.
- Crystal set:

- Antenna led thru coil to ground, to tank (carrier) crystal semiconductor diode acts as a rectifier (detector). Small capacitor allows for low frequency carrier signals to be conducted. The high frequency signal is then redirected to the earphones to produce an audible signal.



  - ([Schematic])
- All modern radio receivers (1949) evolve from this basic schematic

Anatomy



- Antenna: receives many frequencies ->

- The tuner (capacitor?)dictates which frequency should be listened to ->
- The RF amplifier amplifies the desired frequency ->
- The frequency (RF signal) is passed through a local oscillator to a converter which outputs an intermediary signal (IF) at a constant frequency
- The IF is passed through the intermediary frequency amplifier and is strengthened
- The frequency is passed through a detector, where the low carrier frequency is split from the high frequency message ->
- The message is passed through the audio frequency amplifier and output to a speaker.

Ritual

- Use of crystal oscillators in radio
- Thinking of options for wearable sensors: potentially a wrist/hand cuff with a sensor (heartbeat?) that extends to finger cuffs and out to the radio/machines via wires extending from the hand
- Pinterest

Current Goals

- Digest data this week
- Transmit data vis osc (TD or Wekinator) this week
- Prototype TD output
- Continue project report
- Draft wearable sensors & methods for bypassing radio mechanisms, meet with Elio to discuss

April 17

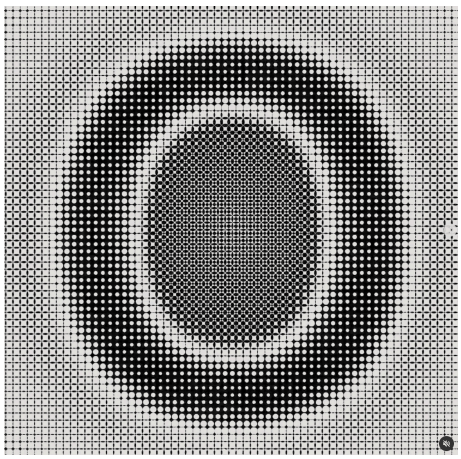
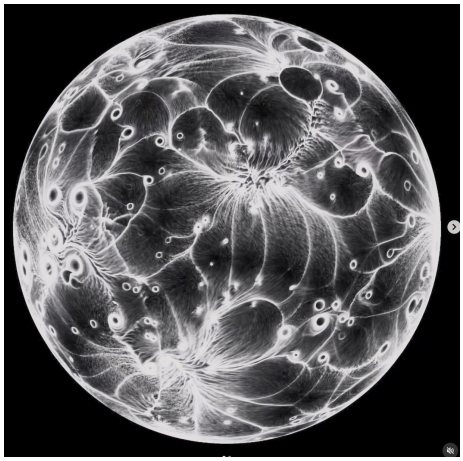
Hardware

- Not opening Tina Tuner - we have one working radio right now that functions great and not enough time to find a backup if something goes wrong.
    - Opting for an external machine (arduino) to sense biometric data. Both machines (radio + sensor) will have a wired connection to the computer containing the ML model.
    - Still considering a greater conjoined machine for this project's future: several Si4732 chips are incoming from AliExpress (likely not arriving until after the project)
        - Similarly, note that Tina uses that same chip and an Arduino Nano, so there is potential for her modification in the future <3

- Investigating possible biometrics sensors:
    - Option 1: Muse Mu-01 EEG headband
        - Using this in combination with Metrics software (PC) or Mind Monitor (iOS) for OSC transmission
        - Allows the sensing of brain waves. Designed to be used as a meditation tool to sense between stressful and relaxed states.
    - Option 2: Emotibit
        - Potentially borrowing one from Illiez. Provides lots of biometric sensing options that could be useful. Transmits over OSC.
        - Emotibit Features include: temperature, respiration, perspiration, heart rate, electrodermal (skin conduction) response

ML & Touch Designer

- Considering a variety of outputs via TouchDesigner. Priorities: an updating audio/visual display, data-rich, decryptable
    - Options: character-based encoding, reconfiguration of the spectrogram in different contexts (harmonics, multidimensional)
    - Inspirations (photos link to source):

April 21

Hardware & Data

Emotibit Paper

- Opting to work with the Emotibit as our biometric sensor - the Muse headband we rented has a fast-draining battery and the sensors frequently disconnect
- Currently missing the microsd card needed to read data, should have one by tomorrow. This prevented us from digesting data today
- The Emotibit:

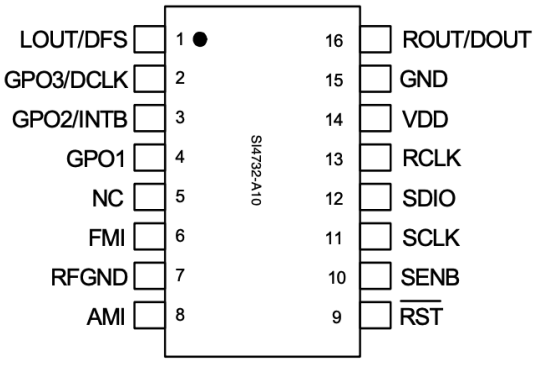
    - is wireless (it operates over wifi, which may be slightly problematic for a live demo at Concordia, but is enabled with other data transmission methods)
    - can be worn on the arm with adjustable attachments.
    - transmits a wide range of features made to describe a person's emotional state and/or other biometrics
- We will be transmitting these features to Wekinator, the output of which will then inform an audio/visual output in TouchDesigner.

Radio

- We (hesitantly) began uploading code to Tina Tuner, primarily from the Si374x Arduino Library on which her software is based.
- We were able to call methods to change features on the radio - eg. change the frequency, mode or bandwidth - from the Arduino IDE Serial Monitor.
    - This allows for much more interesting interaction between agents! The interaction between human->radio->ml->audio/visual can now loop and interact with itself.
    - Previously we thought bypassing radio tuning mechanisms (i.e., controlling the radio externally) was going to be much more difficult than reading its information, but the opposite seems to be true.
- We could also call some methods to read data from the radio, such as its current frequency, bandwidth, station, etc. Other than reading the quality of the signal, we are still having trouble reading the signal actively being received.
- We are reassessing the role of the radio as an input v. output in relation to our ML model. It was initially intended as an input, carrying data from the received signal, but it may now be an audio-based output as its frequency changes
- Software like FLDigi can only decode digital radio communication modes, and listen on multiple frequencies at a time using the software's modem. Although they can input interesting (cosmic) noise data as well, they will not be favoured for the scope of this project. They require more tuning, and we will focus on analog signal.
- Getting analog signal from serial port (USB download cable) to something usable in (wekinator?), using an SDR (Signal Defined Radio) software or a bootleg version we could make ourselves (with TD, Wekinator, …)
- RTL-SDR: SDR receiver written in python
- Experimental Python SDR : Python scripts
- SI374x Arduino Library: https://pu2clr.github.io/SI4735/#si4735-arduino-library-features
    - References: https://pu2clr.github.io/SI4735/#references
    - Examples: https://pu2clr.github.io/SI4735/examples/
    - Pinout:
- Using the PU2CLR Library for Si47xx radio chip, we were able to upload a new code onto Tina Tuner to have more information on what she receive, and more control on certain modification we could apply to her functionalities.
  - We can namely get information on the current frequency the receive is set on (si4735.getFrequency()), on the RDS status, the Station Name, the Program Information, and the RSSI (Received Signal Strength Indicator), and quality of signal.

```
  Checks the station name is available
*/
void checkRDS() {
  si4735.getRdsStatus();
  if (si4735.getRdsReceived()) {
    if (si4735.getRdsSync() && si4735.getNumRdsFifoUsed() > 0) {
      if ((millis() - delayMsgTurn) > 30000) {
        bShowStationName = !bShowStationName;
        progInfoIdx = 0;
        delayMsgTurn = millis();
      }
      if (bShowStationName) {  // time to show Station Name
        stationName = si4735.getRdsStationName();
        showStationName();
      } else {  // time to show Program Information
        programInfo = si4735.getRdsProgramInformation();
        showProgramInfo();
      }
    }
  }
}


/*
  Goes to the next band (see Band table)
*/
void bandUp() {
  Serial.print(si4735.getFrequency());
  // save the current frequency for the band
  band[bandIdx].currentFreq = currentFrequency;
  band[bandIdx].currentStepIdx = idxStep;  // currentStep;

  if (bandIdx < lastBand) {
    bandIdx++;
  } else {
```

```
    // Nothing to do if you are in FM mode
    band[bandIdx].currentFreq = currentFrequency;
    band[bandIdx].currentStepIdx = idxStep;
    useBand();
  }
}

// Show RSSI status only if this condition has changed
if ((millis() - elapsedRSSI) > MIN_ELAPSED_RSSI_TIME * 9) {
  si4735.getCurrentReceivedSignalQuality();
  int aux = si4735.getCurrentRSSI();
  if (rssi != aux) {
    rssi = aux;
    showRSSI();
  }

  if (countRSSI++ > 3) {
    disableCommand(NULL, false, NULL);  // disable all command buttons
    countRSSI = 0;
  }
  elapsedRSSI = millis();
```



- Got Tina a lift! And uploaded a new RDS code on her (code here). Buttons and functionalities are a little different now, but we are getting more information from the screen (data, hour, FM broadcast channel being aired), and we also get a serial monitor print on each button clicked! We're working with something now.

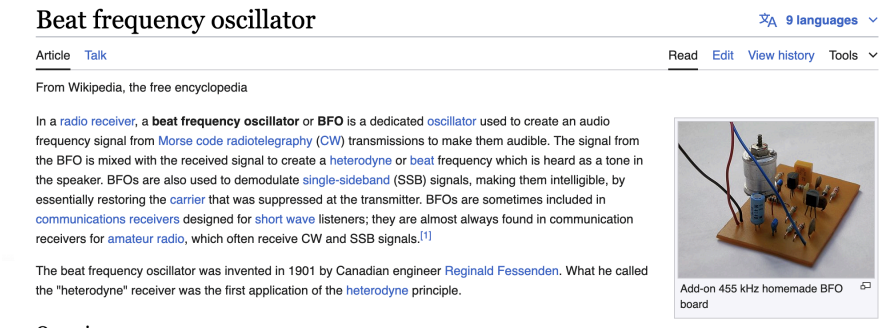
Final Project & Goals

- Approaching the project deadline, we're continually addressing the realistic scope of this project. We want to continue developing this after the class finishes, but recognize the necessary deadlines to meet for our presentation next week. We're considering focusing more heavily on the research paper than producing a live demo. The research paper may become the focus of this in its scope as a class project with our documentation on our prototype supplementing it.
- Our primary focus is in producing a research paper that could be submitted for publishing and/or be used in future grant applications for this project. The paper will include a lengthy plan & methodology section outlining the technical specifications of this project & our plans for its future. It will be supplemented by a 'proof-of-concept' prototype including the emotibit

April 23rd

Radio:

- BTO/VFO is a function that can be found in the new code Tina has. More research needed to know how to use it on our current chip, but BTO could be an interesting avenue to map a radio's internal "beat" to a human's, in addition to their respective "frequency". BFOs are used to demodulate SSB signals to make them intelligible, but can be found as 'noise' on all frequencies and modes (I believe), though in the current code Tina has, the BFO setting can only be found when we are on SSB mode. So more research is needed



Beat frequency oscillator

9 languages

Article Talk Read Edit View history Tools

From Wikipedia, the free encyclopedia

In a radio receiver, a **beat frequency oscillator** or **BFO** is a dedicated oscillator used to create an audio frequency signal from Morse code radiotelegraphy (CW) transmissions to make them audible. The signal from the BFO is mixed with the received signal to create a heterodyne or beat frequency which is heard as a tone in the speaker. BFOs are also used to demodulate single-sideband (SSB) signals, making them intelligible, by essentially restoring the carrier that was suppressed at the transmitter. BFOs are sometimes included in communications receivers designed for short wave listeners; they are almost always found in communication receivers for amateur radio, which often receive CW and SSB signals.[1]

The beat frequency oscillator was invented in 1901 by Canadian engineer Reginald Fessenden. What he called the "heterodyne" receiver was the first application of the heterodyne principle.

Add-on 455 kHz homemade BFO board

- Arduino Si4732 (Tina) code:
  Method *si4735.frequencyUp();* and *.frequencyDown();* allows you to control the frequency of the radio easily given any event

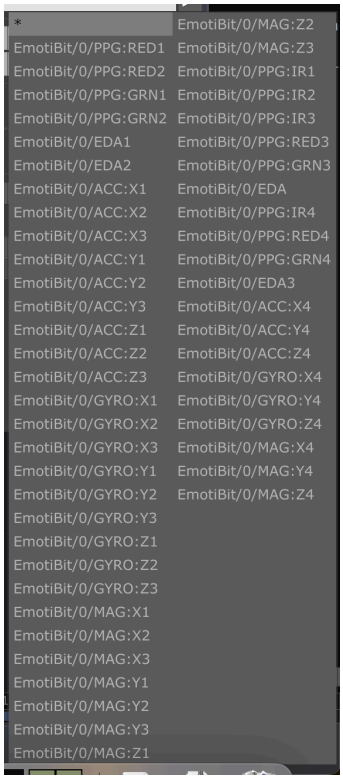
Add “si4735.seekStationProgress(showFrequencySeek, checkStopSeeking, seekDirection);” to add the new station information to code

```
- if (currentMode == FM) {
-         float f = round(si4735.getFrequency() / 10.0);
-         currentFrequency = (uint16_t)f * 10;  // adjusts band space from 1
  (10kHz) to 10 (100 kHz)
-         si4735.setFrequency(currentFrequency);
-       } else {
-         currentFrequency = si4735.getFrequency();  //
-       }
-       showFrequency();
-
```

Emotibit:

- Using Thomas’ Emotibit ML library here.
- Emotibit to thomas’ ML parser -> Wekinator -> beyond….

- OSC in CHOP from TouchDesigner receives all this channels from EmotiBit:



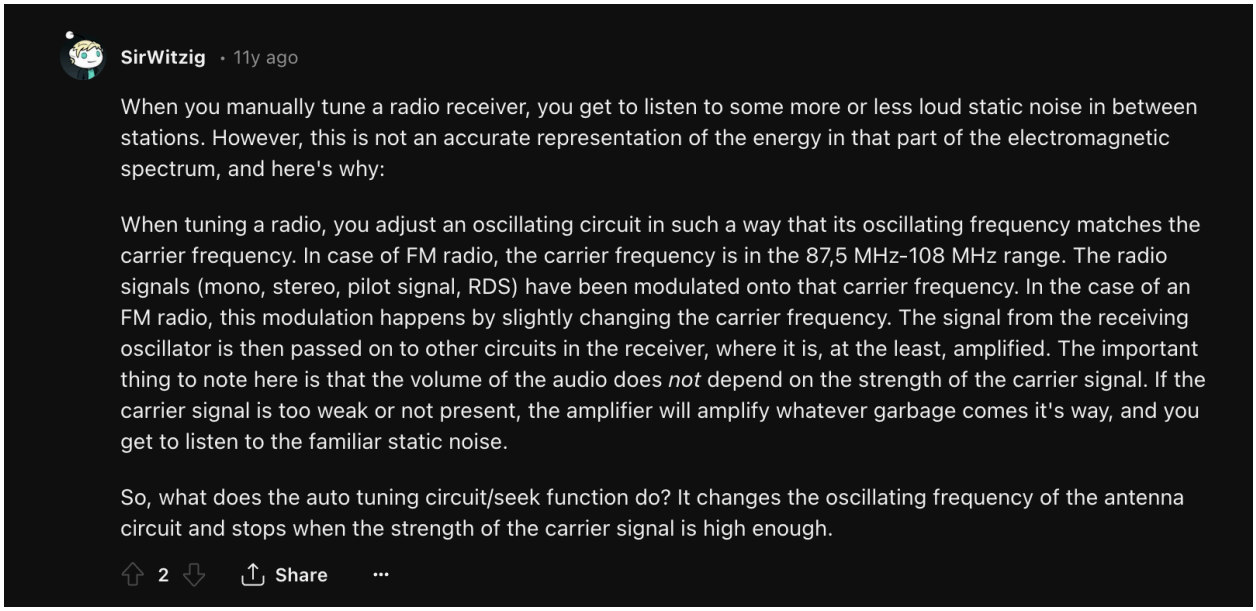
- Arduino IDE reads radio data (prints to serial), using OSC arduino library, echoes the serial data to OSC, caught in touchDesigner,

Interaction loop
https://miro.com/app/board/uXjVKQICCFI=/
-> join board link:
https://miro.com/welcomeonboard/ejhGMWhsWXpzV2JUMUY5YmR0clFtN05aRWtybkNQNVVEWDNGQmpaeFR5T2hCZlBheHZCWEVqMzI3TVpiQUdneHwzMDc0NDU3MzQ5ODE5MDQ4Nzk5fDI=?share_link_id=874669076461

Function documentation API
https://pu2clr.github.io/SI4735/extras/apidoc/html/functions_g.html

How does the seek button of a radio work (full thread here):

-



Potentials for ML:

- Human side:
    - Emotibit, decoding the biometrics to extrapolate an emotion and further extrapolate a frequency/wave function. The extrapolated emotion is placed upon a spectrum of possible emotions, with a meditative/mindful state centered in the middle. The aim is to align with Tina’s frequency
        - This wave function can then express itself audibly or visually as light/color/audio
- Tina side:

- When seeking, it is difficult to determine *why* she chooses to stop on a particular frequency. Also unclear why she scans certain frequencies and not others
- ML to understand her perspective and what she deems as similar or good enough from an audio signal

April 24rd

Emotibit and emotions: https://www.emotibit.com/sensing-emotions-with-emotibit/

EVERYTHING IS MADE OF FREQUENCIIIIIIIES!!!!! Proof

Most important video: The whole universe is vibrating (8min video)

- Explains how atoms make everything vibrate on some kind of frequency.
- We can calculate the human brain's frequency with an EEG, using 4 different channel frequencies. Two of them specific target calm and mindful brain activity, while the other increase when the brain focuses on specific tasks and goals (gets activated).
- 432Htz vibrations will make you feel calm, 19Hz vibrations will make you feel angry.
- "After a short while, it is quite common for different vibrating things to start vibrating at a the same frequency when they come together. Their synergy appears to be mysterious at times. This phenomenon is described as spontaneous self-organization" (very end).
- This is an important breakthrough for the future of this project. Whereas an Emotibit can infer an emotion from the human participant, an EEG device could give us an actual frequency to work with. However, human frequencies are usually in the range of 7.5Hz (similar to that of nature, according to the video), so we would still need some sort of mapping to put the radio signal, cosmic noise, human frequency, etc, on the same "range".

More on Cosmic Microwave Background Radiation here

Link to interbeing ISBN (pdf in drive)

Full journal from 1999 on ionosphere, 15 articles about physics of ionosphere can be found here