## Project Motivation, Attribution and Other Thoughts, May 2020

I have had some success in executing SDR projects using STM32 Discovery Boards which combine a 32 bit processor with a graphic touch screen display. These boards make for a compact stand alone QRP transceiver. My past projects made use of IQ DSP processing to integrate the Discovery boards with popular SDR radios such as the SoftRock RxTx and UHFSDR transceivers.

Kees Talen provoked me to do an FT8 project. I tried several times with NIL results. Then I came across the work done by Karlis Goba which you may review on this website: https://github.com/kgoba/ft8_lib . Karlis is a ham, YL3JG.

And, I decided to port my previous FT8 work over to the STM769 Disco board which has a lot more RAM than the STM746 Board. ShaZam I finally got the FT8 stuff to work! Not only does iit work, it works well due to the great work done by Karlis. I have had several email exchanges with Karlis and he has been most helpful and supportive in my effort to produce another DSP project.

One of my colleagues, Jim Reagan (W0CH)L, shared with me an interesting Silicon Labs chip, the Si4753. Our local QRP Group, Austin QRP Club, are quite experienced with several Silicon Labs products to include the Si570 and Si5351. Silicon Labs headquarters are located in Austin and they have been supportive by allowing several of their engineers to make technical presentations at our meetings.

I found the Si4753 to be most intriguing and well behaved. I made a crude lash up and fed the audio into one of my STM board projects and found it to really works well. So, I decided to build a transceiver for FT8 which uses the Si4735 for receive and the Si5351 for generating the FT8 signal directly at the chosen operating frequency. Hence the name for my project, Silicon Laboratories FT8 Transceiver.

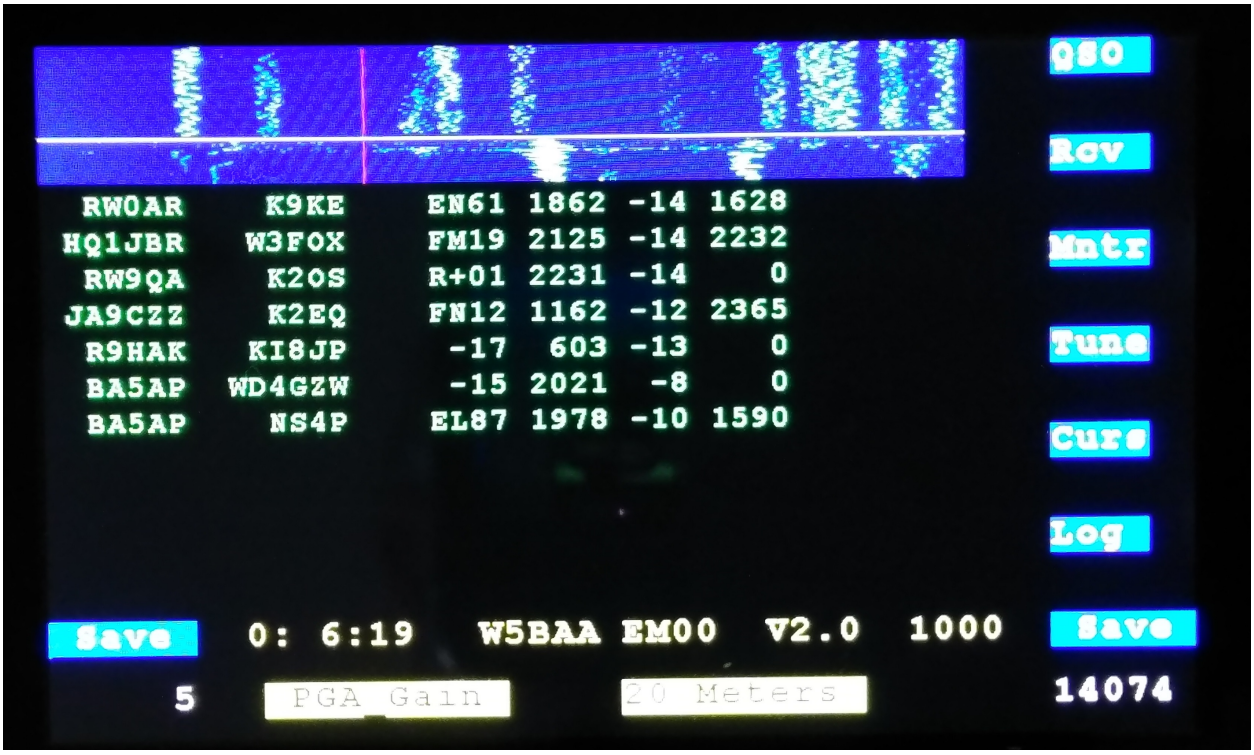
At, present the operating mode of the application is quite limited. However, as I learn more about FT8 operations and experiment with the code I may add additional operation features.

## Time Synchronization

One of the main items that I learned from Karlis is that it is quire easy to synchronize the application in time without relying upon a GPS or network clock. To do this, we set up an internal clock in software with millisecond resolution to create what I call FT8 Relative Time. This clock is synchronized by simply watching the waterfall and depressing the Blue User Button on the STM769 board during the lull in FT8 traffic. My experience is that FT8 application will remain synchronized for hours using this method.

## Simple Operating Modes

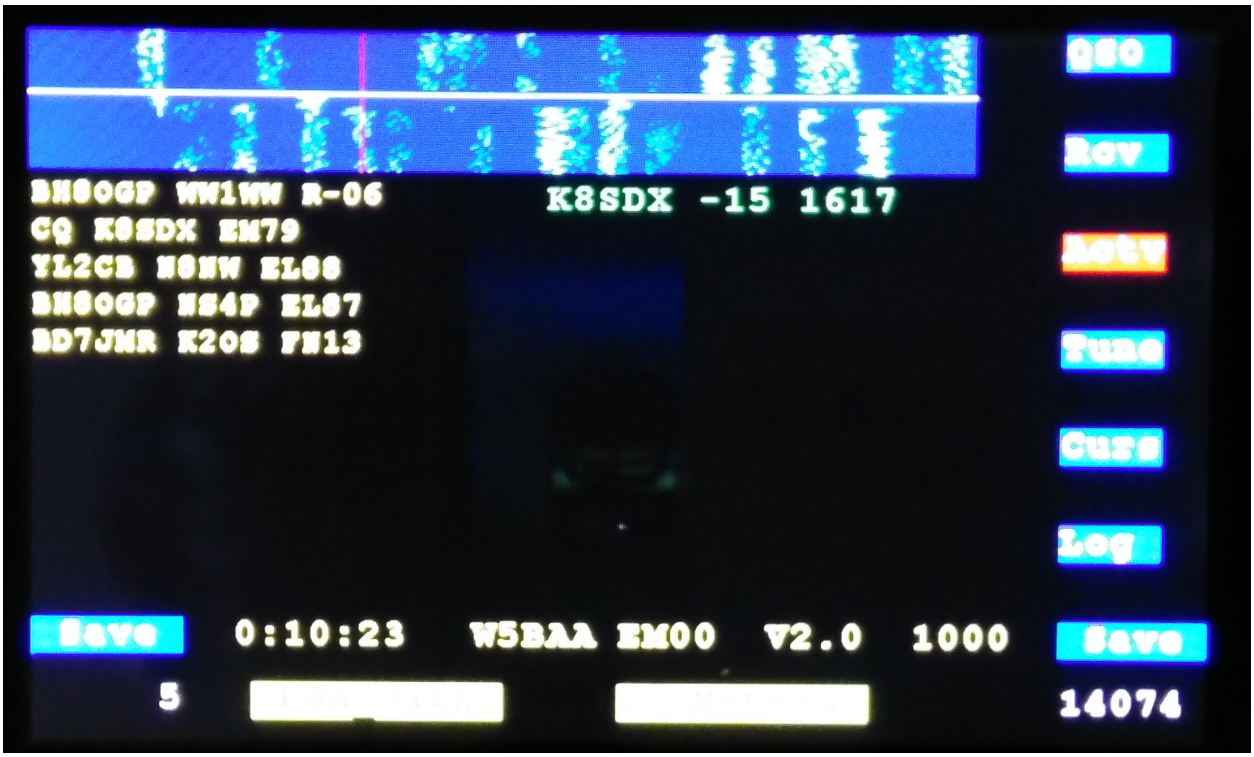
The transceiver has three operating modes: Monitoring FT8 Traffic,  FT8 QSO and transmitting a FT8 Beacon Message.




When the transceiver starts up it is in the Monitoring Mode.

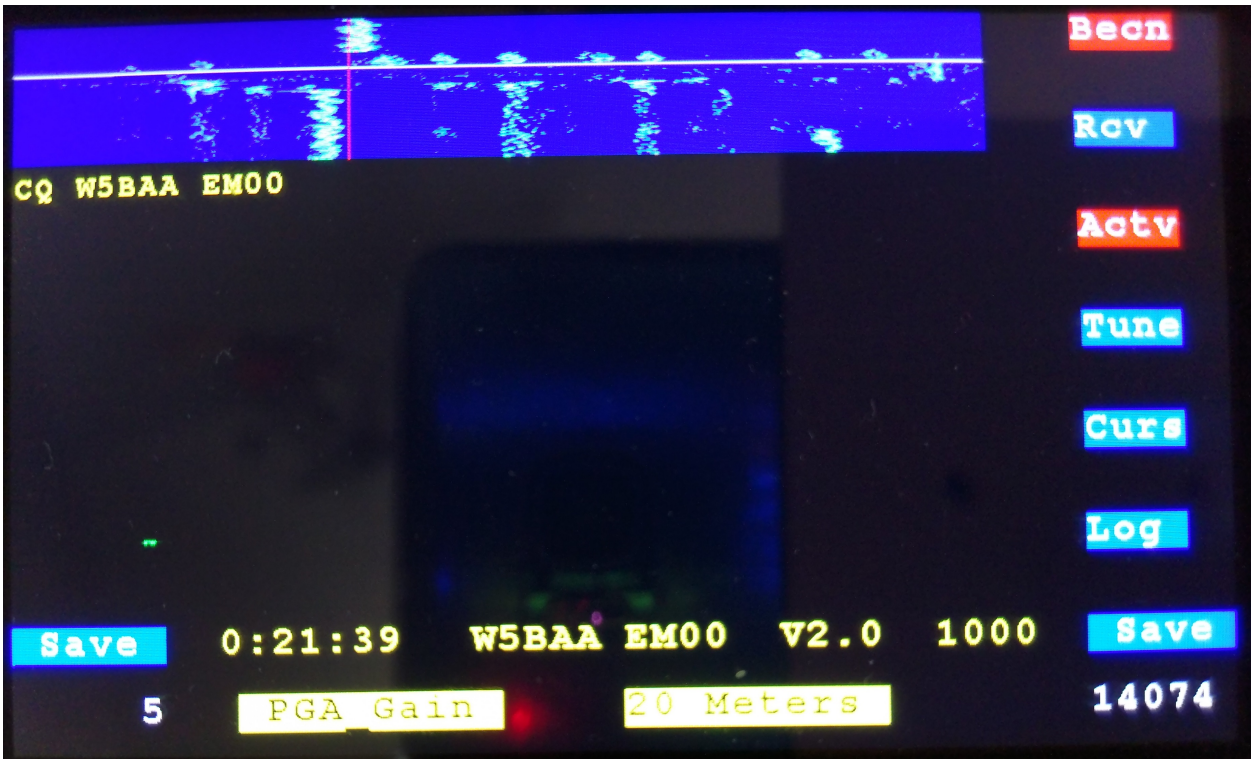
Up to ten decoded FT8 messages and data are displayed in green immediately below the Waterfall. This display is organized as six colums: **FT8 To Call Sign**, **FT8 From Call Sign,  FT8 Message,  Received Audio Frequency**, **Received Signal Level**,  and **Distance** between your station Maidenhead Locator and the received station Locator in Km.

A blue button on the right is shown labeled as “Mntr”. When this button is touched, it turns red and is labeled “Actv”. This mode allows transmission of FT8 messages.




At this point the unit is in the Reply to CQ mode as indicated by the blue Touch Button at the far right hand side of the display which is labeled “QSO”. In this mode up to ten FT8 messages are displayed in the left hand pane. In the right hand pane up to seven FT8 CQ messages are displayed as Call Sign, Received Signal Level and Distance between stations. Touching one of the seven messages will generate a message to the selected station. A call will be issued to that station on the next FT8 cycle.  This message is displayed at the top of the right hand pane .The application will then listen for a reply from this station for two FT8 cycles. If a reply is heard it will be displayed at the bottom of the right hand pane. The application will repeat calling the station up to three times before giving up.

When the QSO Touch Button is touched the button turns red and is labeled “Becn”.  In this mode the application monitors the FT8 Relative Time clock and will issue an FT8 CQ that includes Your Call and Maidenhead Locator during the next FT8 cycle. The application then monitors the FT8 traffic and searches for Your Call at the beginning of each FT8 decoded message. When if finds the first decoded message with Your Call, it composes an FT8 reply that includes the Responding Station Call, Your Call and the other stations Received Signal Level. . This reply is then transmitted in the next FT8 cycle and is shown as green text at the top of the right hand pane.  At the end of this FT8 transmission, the Becn is reset to issue another CQ on after listening to the FT8 traffic for two FT8 cycles.



# Silicon Labs FT8 Transceiver Project Hardware Details

The Silicon Labs FT8 Project uses an STM32F769 Disco Board to provide audio processing and control so that the user may communicate with other radio amateurs using the FT8 mode of text message transmission and reception.

The RF side of the project employs an Si4753 receiver IC for SSB HF reception and an Si5351 clock IC for generation of an FT8 signal directly at the chosen frequency of FT8 operation. In addition, The Si5351 IC also provides the clock signal for the Si4753 receiver IC at 32768 hz. The FT8 signal generated by the Si5351 is passed thru a low pass filter to remove unwanted harmonics. The filtered signal is fed into a MiniCircuits GVA 84 – I monolithic amplifier. The output of the amplifier is fed into another low pass filter. The output low pass filter is directly connected to your antenna. The power output to the antenna is 100 milliwatts ( about 6.5 volts peak to peak into a 50 ohm load).

The receive signal is taken at the input terminals of the low pass filter connected to the amplifier. So, the low pass filter connected to your antenna is used on both transmit and receive. Two 2N7000 FETs are used to interface the Si4735 with the output low pass filter.

The power for the GVA 84-I is controlled by a 2N3906 PNP transistor. This transistor is turned on and off by a 2N3904 NPN transistor.

A Teensy 3.2 board is used to control the two Si ICs via a common I2C bus. Further, the Teensy provides a PTT signal which controls the 2N3904 transistor described above.

The Disco Board includes a 3.5 inch color touch display with a resolution of 800X 480 pixels. The F7_FT8 project includes two rotary encoders for user adjustment of operating parameters and frequency.

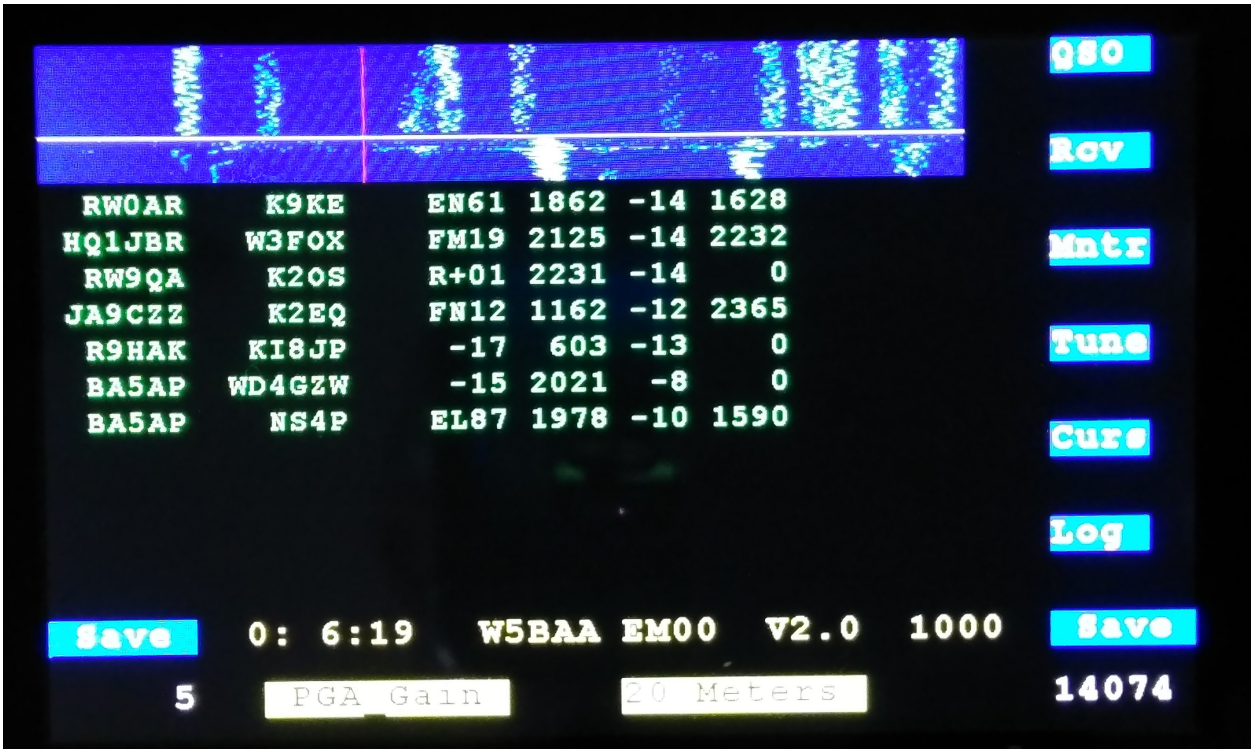
The RF board, Teensy 3.2, and the STM32F769 Disco Board are powered at 5 volts via a single power connection. The Si4735 and Si5351 are powered at 3.3 volts via the Teensy 3.2.

The interface between the F769 board and the Teensy board for frequency control and PTT control is via a serial UART connection. Source code file is provided for programming the Teensy board. The only other connections required for the project are the audio output of the Si4735 to the audio input of the F769 board. Although this connection is shown as a two channel (stereo) connection, the audio processing is applied to only one audio channel.

The project uses an SD Card inserted into the SD Card Socket on the F769 board for storage of operating parameters, band frequency and station data. Also, operation log data imay be stored to the SD Card.

The **Blue User Push Button** on the lower left corner of the Disco board is used to time synchronize the FT8 coding and decoding algorithms with the rest of the World.

All control of the transceiver is done thru the encoders, the touch screen or the User Push Button. The secrets of these controls are revealed below.




## Screen Layout

A realtime Waterfall display of received audio is displayed at the top. The Waterfall covers the frequency range of 300 to 2300 Hz. A red cursor is shown which shows the frequency chosen for transmission in the QSO and Beacon modes. The location of the cursor may be set by touching the Waterfall or by rotating the Frequency Encoder when the **Curs** Touch Button is blue.

A column of Touch Buttons are provided along the right hand side which are explained in following sections.

The decoded FT8 messages and data are displayed in green immediately below the Waterfall. This display is organized as six columns: **FT8 To Call Sign**, **FT8 From Call Sign, FT8 Message, Received Audio Frequency**, **Received Signal Level**, and **Distance** between your station Maidenhead Locator and the received station Locator in Km.

A line of two Touch Buttons and general data is shown below the FT8 message area.

And, finally, at the bottom of the screen are two Menu Bars and associated user input data displays.

## Touch Buttons

There are eight touch buttons on the user screen which are shown as **blue** with text labels in white upon boot up and will change to **red** background when touched. The operation of each button is described below:

**QSO / Becn:** When this button is **blue** and the Mntr Button is **red** (in the Active State) the FT8 traffic is monitored for stations calling CQ. Stations calling CQ in the last FT8 period are listed in the right hand pane. You may select any of the stations as your target for calling. The call and response messages of the QSO are displayed in the right hand pane. At the end of the session you may reset the QSO state by touching anywhere in the right hand pane.

When this button is r**ed** the unit will automatically issue a CQ transmission with your call and locator every other FT8 period So, when in the beacon mode the unit issues a CQ transmission once every 30 seconds. The CQ call is made at the audio frequency indicated by the red cursor line displayed in the waterfall. When the application detects a station answering your CQ a reply to that station is transmitted and displayed in the right hand pane.

**Rcv / Xmit:** This button does not respond to your touch. However, it is linked to the QSO / Becn button so that when it is **blue** the unit is in the receive mode. When it is **red** the unit is transmitting in either the QSO or Beacon mode.

**Mntr / Actv:** When this button is **blue, t**he unit is in the receive only mode**.** When it is red the unit will transmit in either the QSO mode or the CQ(Beacon) mode.

**Tune:** When this button is **blue**, nothing exciting happens. When it is **red** a continuous Si5351 signal is generated but not transmitted at the frequency indicated by the red cursor line displayed in the waterfall. This allows you to see precisely where in the FT8 spectrum you will be transmitting.

**Curs / Frequency:** When **blue**, the Frequency Encoder adjusts the transmit audio frequency which is shown in the waterfall and as a numerical value displayed immediately to the left of the right hand Save button. When **red**, the Frequency Encoder adjusts the RF frequency of the Si4735 receiver and the base frequency of the Si5351 transmit clock. The RF base frequency is displayed in the extreme lower right hand corner of the display. The actual transmitted frequency is the sum of the audio frequency and the RF base frequency.

**Log:** When this button is **red** each FT8 QSO conducted by your unit will be saved to a log file on the SD card.

**Right Hand Save:** When this button is touched it will briefly flash as red to indicate that the RF frequency displayed immediately below is saved to a file on the SD card.

**Left Hand Save:** When this button is touched it will briefly flash as red to indicate that the Parameter Value displayed immediately below is saved to a file on the SD card.

**Data Display Line:** Nestled between the Left and Right Save buttons five data items are displayed:

FT8 Session Relative Time in Hours, Minutes & Seconds (FT8_Time)

Your Station Call which you enter via a file you place on the SD Card

Your Station Maidenhead Locator which you enter via the SD Card

The revision level of the firmware.

The current transmit audio frequency in Hz.

## Touch Menus

There are two touch menus on the user screen which are shown as yellow with text labels in black upon boot up. Pressing the left side of a menu will cause the selection to go backwards. Pressing the right side of a menu will cause the selection to go forward. The left hand menu controls the unit's operating parameters while the right hand menu controls the RF Frequency Band. The operation of each menu is described below:

**Parameter Menu:** This menu displays the current parameter that can be modified or saved. The parameter selected can be changed with the Parameter Encoder.

**Band Menu:** The menu displays the current band. The current frequency associated with the band selected is displayed immediately to the right of the menu. The RF frequency of the Si4753 Receiver is immediately changed when this menu is manipulated.

## Parameter  Data Items

The data items associated with the Parameter Encoder are described below:

**PGA_Gain:** This sets  the gain of the Codec first gain stage of the analog  audio input.

**ADC_Gain:** This sets the gain of the Codec Analog to Digital Converter (ADC) of the analog audio input.

**RCVR_Gain:** This value controls the audio level  of the audio output sent to the headphone port when the transceiver is in the receive mode.

**Sidetone_Gain:** This value controls the audio level  of the audio output sent to the headphone port when the transceiver is in the transmit mode.

**Xmit_Set:** This value is used to adjust the exact frequency of your transmitted FT8 signal

**5351_Set:** This value is used to adjust the Si5351 Clock Correction value.

**Day:** This value is used to adjust the numerical day used in the Log File name.

**Month:** This value is used to adjust the numerical month used in the Log File name.

**Year:** This value is used to adjust the numerical year used in the Log File name.

## Frequency Calibration

This project relies on the Si5351 Clock Chip to provide the required clock signals for both receive and transmit. Clock Output 2 provides the 32768 Hz clock required by the Si4735 Receiver Chip. Clock Output 0 provides the clock used to generate the FT8 directly at your selected frequency.

The Si4735 receive frequency is set directly by the application, what you see displayed in the very lower right hand corner of the display is what is sent to the receiver. The Si4735 step frequency resolution is one kilocycle.

The Si5351 transmit frequency is calculated by the application as the sum of the displayed receive frequency plus the audio frequency indicated by the red cursor in the waterfall display and as an analog value displayed immediately to the left of the Right Save button.

Two values are provided for frequency calibration:

**5351_Set:** This value is used to adjust the Si5351 Clock Correction value. The correction factor affects both clock outputs.

**Xmit_Set:** This value is used to adjust the exact frequency of your transmitted FT8 signal by adjusting the Clk 0 frequency.

The frequency calibration procedure requires two steps.

First, set the receiver to the 15 Mhz WWV  band. Set the exact receive frequency to 14999 Hz. This should give a beat tone of 1000 Hz. Adjust the  Si5351 Clock Correction value using the parameter encoder so that the WWV beat audio matches 1000 Hz.

Secondly, set the unit to the desired FT8 band and the touch the Tune button. The Si5351 signal will be displayed as a received signal on the waterfall. Adjust Xmit_Set value so that the Si5351 signal matches the red cursor.

Save both the 5351_Set and Xmit_Set values to the SD Card.

## SD Card Installation

The application  utilizes a micro SD.  A micro SD must be installed in the micro SD slot.  The system displays an error message if no micro SD disk is found.

For proper operation, three files must exist on the SD:

Frequencies.txt

SaveParams.txt

StationData.txt

Upon initial startup of the application you do not need to have the Frequencies.txt or SaveParams.txt files installed. The application will recognize that these files are not present and will automatically generate them using default data stored in the firmware. After these files are initialized you may modify them using the menus and encoders as described in the previous sections.

However, you must create and install the StationData.txt file manually. Notepad is a good Windows application for creating this simple file The file requires a single line of data as shown below:

W5BAA:EM00

Replace my call with your call and the EM00 with your Maidenhead Locator.

## Logging QSOs

When you use the application you may log your FT8 transactions by using the Log Touch Button.

The log file will be automatically generated and it will have the file name daymonthyear,txt,

An example of the contents of a log file are shown below:

W5BAA KA7QJN +07

KA7QJN W5BAA EM00

W5BAA AB6OR DM13

AB6OR W5BAA EM00

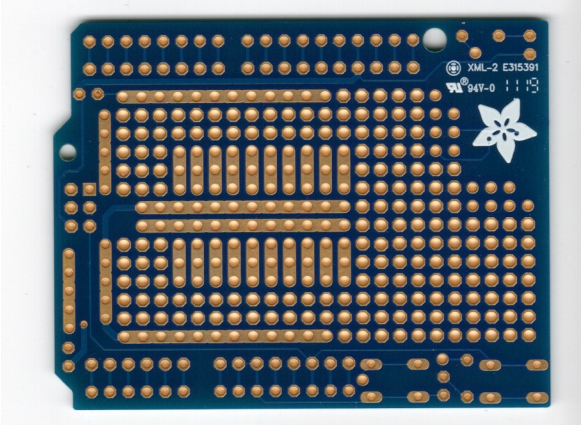
W5BAA W4LRN EM90

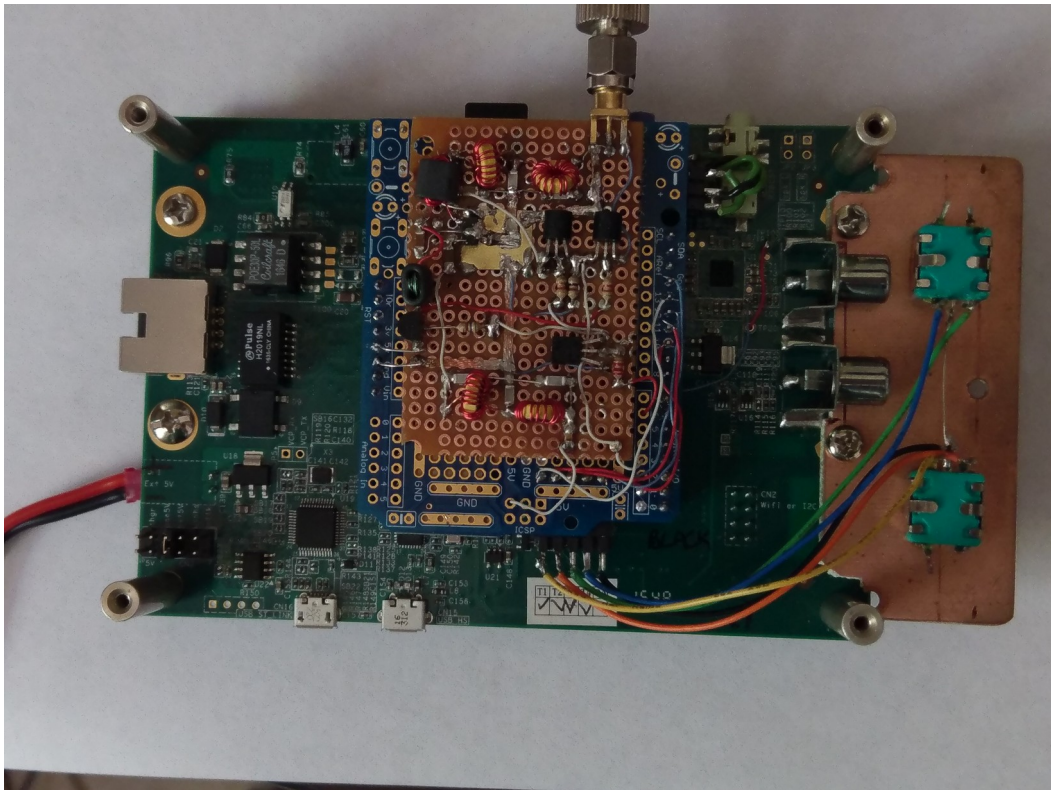
W4LRN W5BAA EM00

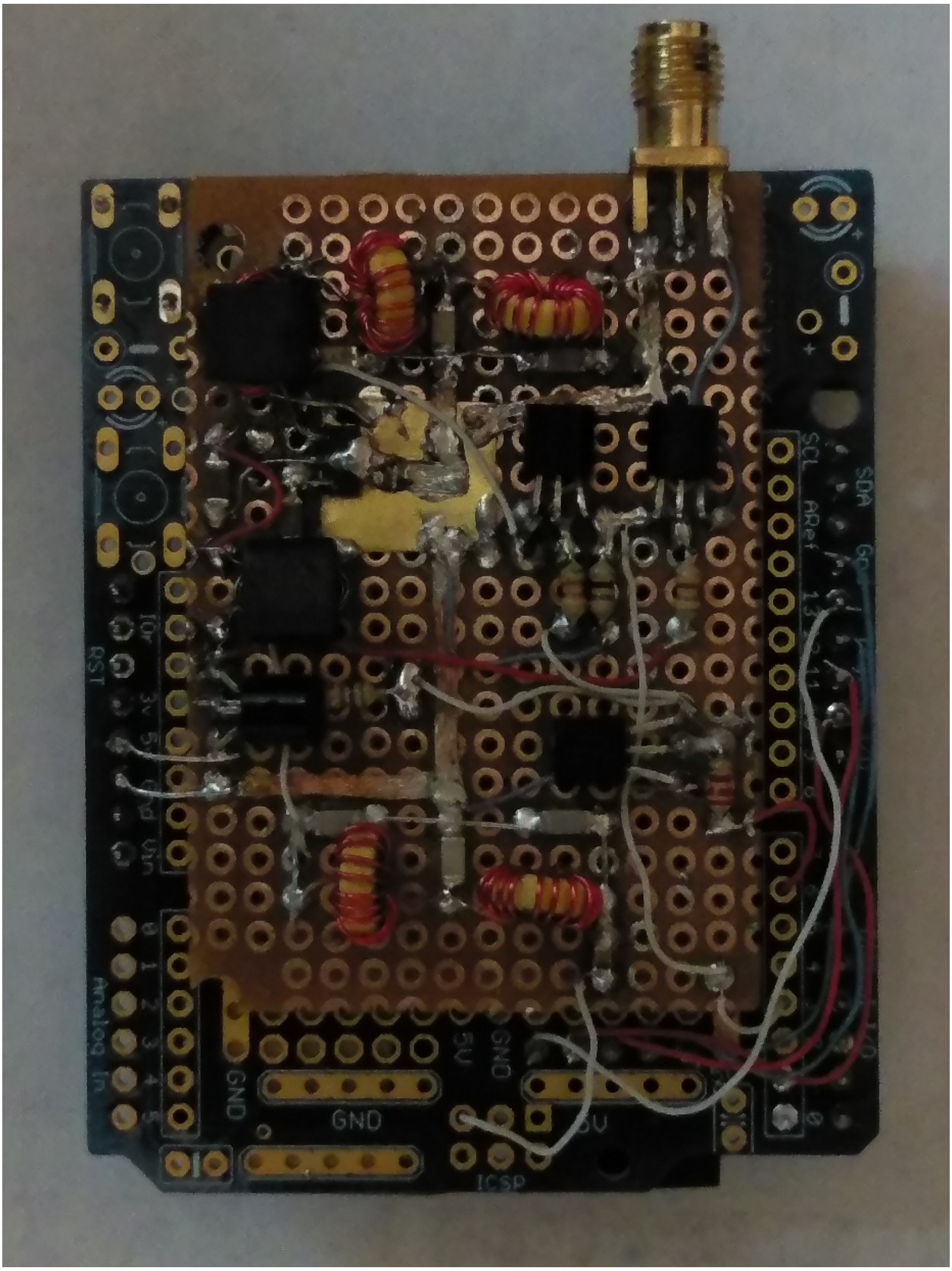
W5BAA W4LRN EM90

## Construction

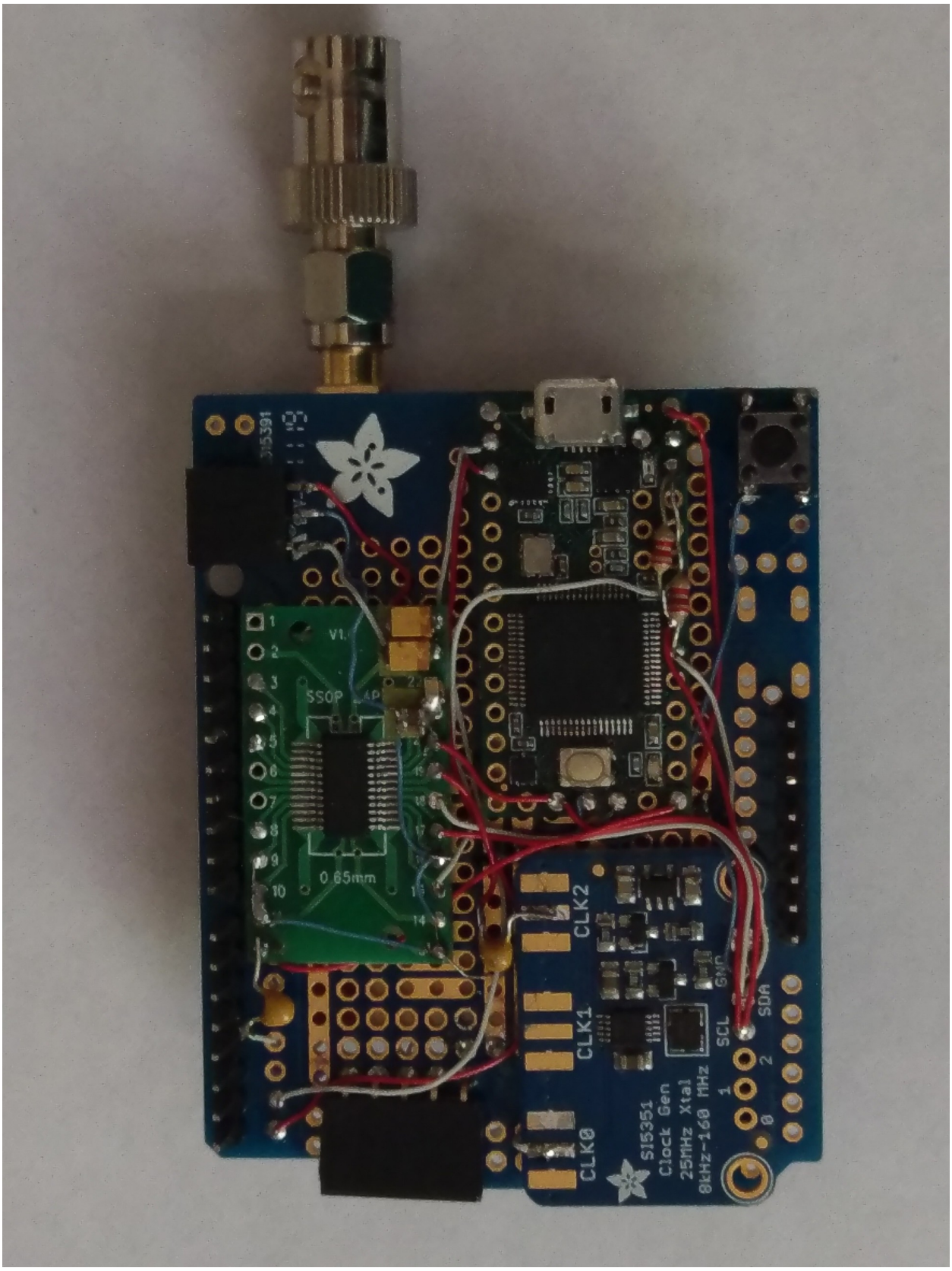
The STM user manual for the F769 Disco Board is provided as a PDF file. Described in the manual are four header pin connectors which are laid out to match the connectors on the Adafruit Shield Board shown below. All of the interface connections required for this project except for the receiver audio output signal are brought out on the four header pin connectors. Also, photos of my project board are shown.






Front Side


SSOP
0.65mm
CLK0
CLK1
CLK2
SCL
SDA
Si5351
Clock Gen
25MHz Xtal
8kHz-160 MHz

Back Side

## Installing Firmware on F769 Disco Board

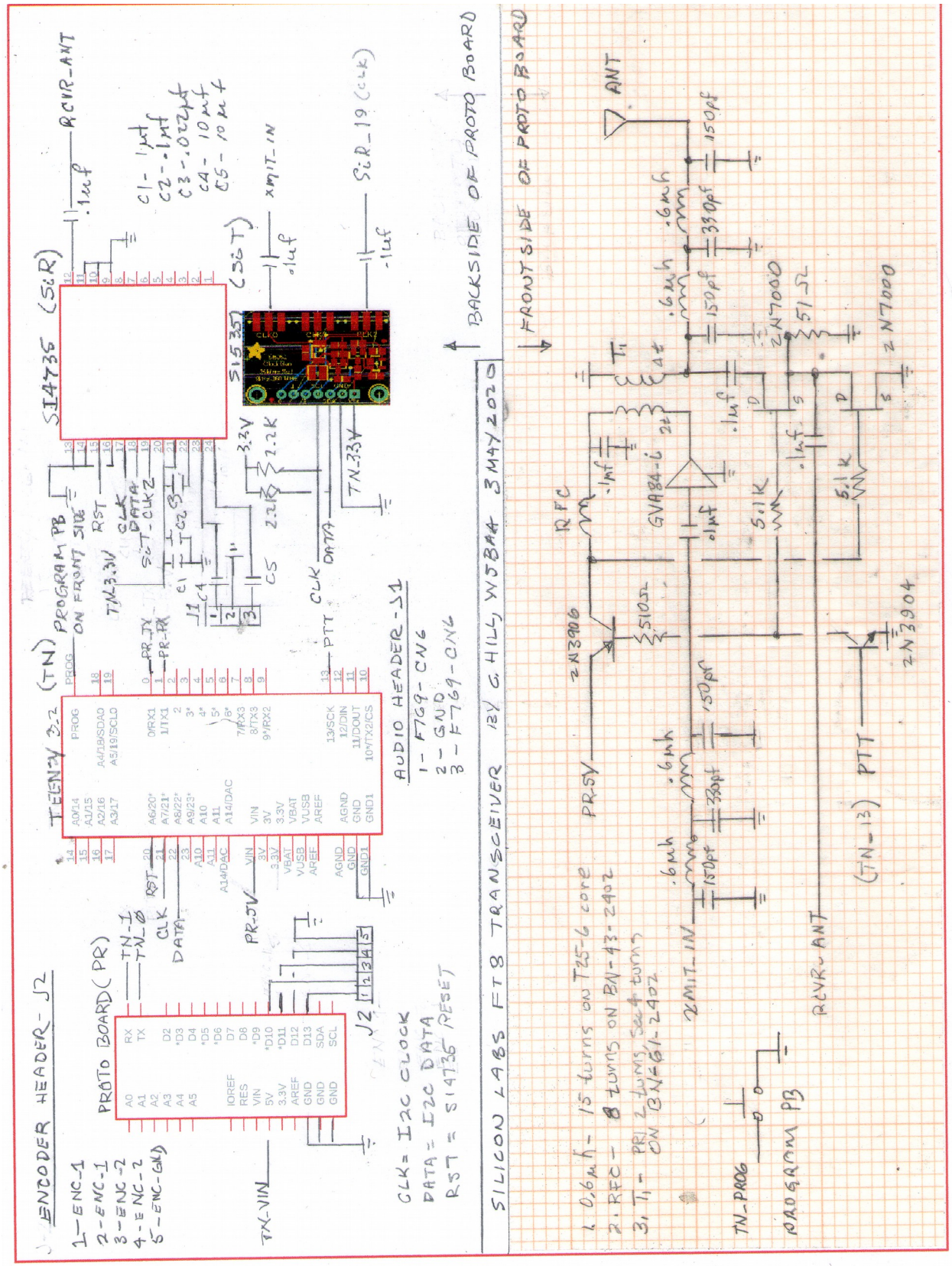
The firmware is supplied as a STM32f  binary file labeled “Velvet.bin”.

This file is installed on the STMF769 board using the STM Utility called ST Link which may be obtained at this link: https://www.st.com/en/development-tools/stsw-ink004.html .

ST Link is free and only requires an USB cable for connection between your PC and the F769 Disco board.

## Installing  Firmware on your Teensy 3.2 Board

The Teensy 3.2 application is provided as an INO file with the required drivers which you may used with your Arduino IDE to compile and download to your Teensy 3.2.




**Project Schematic**