

ICARC Fox Transmitter

William Robison
KC0JFQ

October 27, 2025

A proposed article for publishing in QST.

L^AT_EX

This is pdf-TeX, Version 3.14159265-2.6-1.40.18

kc0jfq@n952.ooguy.com.

Documents webpage:
<http://n952.ooguy.com/HamRDF>

Contents

1	Genesis of a New Fox Transmitter	1
1.1	The Evolution of a Fox Transmitter	2
2	Hardware	3
2.1	Packaging	3
2.2	DC Power	3
2.3	Frequency Control	3
2.4	RF Amplifier	4
2.5	Audio	4
2.6	HT Control	4
2.7	TOY Clock	5
2.8	FRAM and FLASH	5
2.9	Configuration Port	6
2.10	Battery monitor	6
3	RF Amplifiers	6
3.1	HF Amplifiers	7
3.2	GPS Module	7

4	Software	8
4.1	Message Traffic	8
5	A sample Setup	9
5.1	INI=, Initialization commands	9
5.2	ANN=, Announcement Message Commands	11
5.3	TALK directory	12
5.4	Frequency table	13
5.5	Schedule 0	14
6	Conclusion	15
6.1	Acknowledgments	15
6.2	Authors email	15
6.3	Additional Documents	16
6.4	Board Availability	16

List of Figures

1	102-73181-10 Regulators, Reset, HT connector	17
2	102-73181-10 Peripherals	17
3	102-73181-10 Modulation and VCO	18
4	102-73181-10 Amplifier Interface and Output Filter	18
5	102-73227-21 MMIC Module Schematic	19
6	102-73181-36 DRA818 Module Schematic	20
7	102-73181-10 Artwork, TOP	21
8	102-73181-10 Artwork, BOT	21
9	102-73181-10 Board Image	22
10	102-73181-36 Board Image	22
11	102-73227-21 Board Artwork	22

Supporting graphics and image files:

Schematics 102.73181.10

Figure 1, Schematic, Sheet 2
 Figure 2, Schematic, Sheet 3
 Figure 3, Schematic, Sheet 4
 Figure 4, Schematic, Sheet 5

Boards

Figure 7, artwork Top
 Figure 8, artwork Bottom
 Figure 9, Completed FOX Transmitter

Schematics 102_73227_21

Figure 5, MMIC Amplifier Schematic, Sheet 1

Boards

Figure 11, MMIC Amplifier artwork Top

Schematics 102_73181_36

Figure 6, DRA818 Module Schematic, Sheet 1

Boards

Image 10, Completed DRA818 Module

William Robison KC0JFQ

1 Genesis of a New Fox Transmitter

The Iowa City Amateur Radio Club resumed foxhunting activities following a long pause, a pause long enough for useful skills to have been lost to the mists of time. We resumed our hunts in a county park near town. The park provided a lodge that was used as a comfortable base of operations. We started our hunts with a set of three transmitters that all operated at 146.565MHz. These are typical fox transmitters that are time multiplexed so that their transmissions do not occur concurrently.

Although these early hunts were comfortable for our event organizers, the novice fox hunters were having a rather difficult time locating the transmitters. With each transmitter being active for only one minute out of five, obtaining fixes was proving rather difficult for these novices. The success rate, that being the ability to find the transmitters, was disappointing.

To address this, I started a design effort (all the way back in 2018) to come up with a richly featured fox transmitter that would be able address the shortcomings seen with our hunts. The fox transmitter architecture must be able to reproduce the operating modes used by other fox hunting products as well as satisfy the needs of novice hunters.

We also want to support, to the best of our ability, a formal IARU ARDF event. This requires operating on both the 2M band and the 80M band. Some of the requirements for impounding receivers prior to the start of an event can be addressed in the Fox Transmitter itself, where message transmission doesn't begin until the start of the event!

A casual set of requirements emerged somewhat as follows:

- Easy to use command and configuration language
(using a simple verb/noun structure)
- Easily configured and synchronized schedules
(using a simple serial connection to a host computer)
- Software selectable frequency (dynamic)
(we don't want to deal with jumpers for this)
- Convenient mechanical packaging
(easy to transport, setup and physically find; not too small)
- Ability to configure a training mode
(for my novice hunters to practice with)
- No vendor specific programming adapters
(access using standard serial interface)

Some wag even asked "Why doesn't it talk?". Although said in jest at the time, it turned out to be difficult to resist (and easy to implement).

The design that emerged makes use of a single chip processor (SOC or a System-On-Chip) from Zilog, the 16 bit zNEO. The ICARC units all make use of the Z16F2810 device. The Z16FMC28 variant should work as well. The board outlines and the external mechanical interface of the several revisions have been kept identical in order to allow the use a common enclosure.

1.1 The Evolution of a Fox Transmitter

The design has evolved through multiple iterations, settling on that described here. An early iteration used a Raspberry-PI but this effort proved too power hungry and slow to start to be of real merit.

The hardware for the zNEO has also changed, primarily by moving the RF amplifier off of the main board and on to a daughter-board to allow experimenting with the RF section.

Earlier iterations used an ICS525 or ICS307 produced by IDT to generate an RF carrier. These devices are no longer available and were rather limited in the range of frequencies they could produce.

The latest iteration switched to using the SI5351A from Skyworks for generating a carrier. This is a far more capable device that operates throughout the 2M band as well as allowing the software to easily account for errors in the reference crystal.

2 Hardware

As mention earlier, the design makes use of a ZiLOG zNEO. The zNEO has a comfortably sized program memory (128KB) and a spartan RAM (4KB). These days the only readily available package is a 64-pin flat pack which, of course, is what the latest revision makes use of.

2.1 Packaging

The board fits in a Hammond 1599E enclosure along with a 6-cell AAA pack. The 124mm by 72mm board dimension is directly derived from the 1599E enclosure mechanical drawing . The enclosure may be obtained in several mechanically compatible variations (i.e. varying in color and cost). The final layout for all of the designs (figure 9) makes use of a drilling jig to locate the case holes required for the antenna, power switch, and programming jack.

2.2 DC Power

The design quickly moved to a switch-mode regulator to improve battery life. This also allows us to deal with a higher voltage battery pack without having to manage regulator power dissipation. The primary regulator (the switchmode device) provides 5 volts to the RF subsystem. A secondary linear regulator provides a local 3.3V rail for the digital logic.

Nominally power would be supplied using a 6 cell AAA alkaline pack. Although a simple 9V radio battery would work, in practice six AAA cells seems to be a bit less expensive with the added benefit of longer run time. Primary cells are used to simplify battery management, simply replacing cells when necessary. A 3 or 4 cell Li-Po pack that will mechanically fit would also work well as the first stage regulator will tolerate input voltages up to around 24V.

The battery monitor channel, as populated in the schematic, will work with up to a 12.5V battery, so we can use an 8-cell alkaline pack or a 3-cell LiPo pack. One resistor in the voltage divider may be changed to extend the range of pack voltages that can be measured.

2.3 Frequency Control

The current revision makes use of the clock synthesizer produced by Skyworks as mentioned earlier. The SI5351 isn't terribly difficult to program, it appears in many amateur radio designs. This device gives us access to the entire 2M band in frequency steps compatible with handheld transceivers.

The output from the SI5351 is FM modulated by varying the load capacitance on the SI5351 reference crystal using a pair of varactor diodes. The audio from the zNEO is used to modulate the bias on these varactor diodes, in turn varying the capacitive load across the crystal.

2.4 RF Amplifier

With the RF amplifier moved from the main board to a separate daughter-board, we can experiment the RF amplifier design. For most applications we use an MMIC (IF amplifier) based design. As discussed here we can easily get up to 150mW of output power.

We can also make use of a commonly available transceiver module that is available on eBay. This module (SA818 or DRA818) has an output of up to one watt and is programmed using a serial connection. All we supply to the module (located on the RF daughter-board) is power, audio, and setup commands (through the serial connection). The SA818 and DRA818 are also available in UHF variants.

The output filter is located back on the main board. It is a 7-pole Chebyshev or Elliptic filter (populated for either topology). The antenna connector (BNC) is also located on the main board.

2.5 Audio

There are two audio sources coming from the zNEO. The first is a simple programmable timer channel that produces a square wave. A second channel comes from a PWM controller. Audio data, stored in an external memory device, is transferred into the PWM control register as needed on a byte-by-byte basis. This, of course, allows the transmitter to *talk*.

These two audio sources are combined (wire-or) before being filtered and passed along to the varactors or the DRA818/SA818 module.

2.6 HT Control

Controlling the RF section of the fox transmitter is, for all practical purposes, identical to controlling a handheld transceiver. The audio, a separate Push-To-Talk control, and a serial channel are routed to a header that provides the capability to control an external handheld transceiver. This connector also provides connections to the power rail to allow external power to be delivered to the board through the HT connector.

The system software can control the HT as is, keying the PTT line and providing audio. HT frequency selection is not present in the software at the current revision level. Updates would be required to directly control the HT operating frequency with the FREQ command.

2.7 TOY Clock

A Time-Of-Year clock (an Analog/Maxim/Dallas DS1672) is provisioned on the board. This clock is a simple 32 bit seconds counter with a backup battery. The software keeps its system time in much the same way that the Linux system maintains its system time, as a count of seconds from some epoch.

The battery for the TOY clock is a small lithium coin cell. Although the DS1672 has a built-in charge circuit, its main power pin must be biased to supply current to the backup battery so this feature is not used. To deal with the low on-time seen by the fox transmitter system, the main battery runs directly through a low power regulator and then on to the DS1672 battery to keep it charged. A small current is supplied to the backup battery, on the order of 0.5 micro-amp, which should be compatible with most lithium chemistry cells. The regulator circuit draws roughly 30 microamps from the main battery when it is connected.

2.8 FRAM and FLASH

All of the audio, configuration and sequencing data is held in memory external to the zNEO. The zNEO never attempts to rewrite its own program flash. Two external devices are provisioned to make the data management task a bit simpler and also to reduce cost.

The first device, an FRAM, holds configuration and setup commands. A 64Kb device will hold 256 setup/control commands which is usually sufficient to configure the system for several types of fox hunt. A larger device may be used to increase the number of setup/control commands. The FRAM allows for easy replacement of individual (32 byte) records in the device.

The second device, a FLASH, holds audio waveform data. The audio files are stored in this much larger (and less expensive) device. The FLASH device performs erase operations (slowly) at the sector or device level.

An 8Mb FLASH, holding about 3 minutes of audio, can be had for less than a dollar whereas this size FRAM would cost over \$30.00. The software also deals with larger FLASH devices, should the need arise.

2.9 Configuration Port

Loading audio data and setup/control commands is accomplished using one of the serial ports. Physically it appears as a 3.5mm jack that is mechanically and electrically compatible with an FTDI USB-to-serial cable (TTL-232R-3V3-AJ).

Audio data is stored in the FLASH using a file containing Intel-HEX records. The command decoder recognizes an Intel-HEX record, loading it into the FLASH device.

Setup/control commands are written to the FRAM in a similar fashion. A command may be entered directly to be immediately executed. A "save" command prefix is used to store a setup/control commands into the FRAM memory for later use.

Commands (in the zNEO) are provided to erase and dump both FLASH and FRAM memory. Audio data and setup/control data, being kept in separate devices, are erased using two different commands. The net result being, of course, one does not need to erase audio data when erasing setup/control data.

There is also a high speed loader built into the zNEO software. This loader operates the serial channel in a binary mode to reduce the volume of traffic needed to load the FRAM/FLASH devices. The high speed loader may also operate at elevated bit rates to speed things up a bit more.

The high speed loader becomes vital when loading a large FLASH with audio data.

2.10 Battery monitor

Both battery voltage and battery current monitors are present in the design. The zNEO SOC provides the A/D for these measurements. Voltages are obtained using a simple voltage divider.

Current, on the other hand, is measured using a sense resistor (50m Ω) and a Zetex ZXCT1009. The ZXCT1009 is across the sense resistor in the battery positive. The ZXCT1009 provides an output current proportional to the voltage across the sense resistor. The output of the ZXCT1009 is placed across a resistor to develop a voltage that is, in turn, measured by the zNEO.

3 RF Amplifiers

We only present a brief description of the RF amplifiers here.

An image of the SA818/DRA818 RF module is shown in figure 10 on page 22. A schematic is shown in figure 6 on page 20.

The artwork for the MMIC RF daughter-board is shown in figure 11 on page 22. A schematic for this board is found in figure 5 on page 19.

The common mounting points are evident in figures 9, 10 and 11. The main board, of course, has sockets for all the inter-board signals whereas the daughter-boards only have the required connections.

3.1 HF Amplifiers

The SI5351 will easily produce a clock down in the HF bands. We can make use of this to operate in the lower bands if we add a low pass filter after the amplifier on the RF module.

Such an RF module has been successfully implemented and tested, although it will not be discussed here.

3.2 GPS Module

You will notice there is reference to a GPS Module on the 102-73227-21 Board 11 on page 22. The software has commands that may be used to read time and use the PPS signal from an attached GPS module. This is to allow better time synchronization in the field.

4 Software

The software design aims to eliminate the need for any type of field configuration operations. We really want to be able to simply turn the device on as it is placed in its hiding spot in the field and move on with the next Fox Transmitter.

We expect the *Fox Transmitter* to report on its condition when it is turned on and then settle in to performing its *foxly* duties.

The TOY clock provides the time synchronization required to operate multiple units in a group. This TOY clock is synchronized by a host system the night prior to a hunt (i.e. the time in the TOY clock is updated).

When powered on the system looks for setup commands (in the FRAM) to configure the system. One of the setup commands will copy the current time from the TOY clock into the system time field. The setup commands must also set the callsign, nickname, and other necessary configuration information.

Thus we configure the transmitters *identity* in preparation for transmitting over the air.

The transmitters status may be reported at turn-on. Typically we would vocalize the callsign, nickname, and battery condition. This status report, from every transmitter used in the hunt, would typically be sent on a common *startup frequency*. The transmitters will then switch away from the common *startup frequency*, moving to a unique operating frequency.

Consider, for a moment, a multi-group foxhunt where there are multiple hunt groups. Each group (5 or 6 transmitters) will require a unique frequency to operate on. The hunt organizer would have his H.T. set to the common *startup frequency*, as the transmitters are set out in the morning, listening for transmitters that don't work or have low batteries.

4.1 Message Traffic

Each transmitter will periodically send out a message. The delivery schedule, having been specified as part of the initialization commands, determines the *when*, providing for the regular delivery of *message traffic*. This delivery schedule consists of a cycle time (called "period") and a point in the cycle (called "offset") when the *message traffic* will be delivered.

The *message traffic* is generated by a set of commands that are executed at the scheduled time. This *message traffic* program would send a sign-on message, some message traffic, and a sign-off message. The sign-on/sign-off messages send the station callsign (rules, rules, rules). The rest of the traffic is defined by the program commands, either voice traffic or code traffic.

5 A sample Setup

Operation fox transmitter (hence the "personality" of the station) is entirely controlled by the commands stored in the FRAM. The FRAM is managed as a set of fixed length records, each one holding 32 bytes (thereby limiting the size of any one individual command). Here is a sample setup.

This is a working example, it is complete.

5.1 INI=, Initialization commands

The first group of commands are the initialization commands. This group of commands runs when the zNEO is reset (power-on or by mashing the reset button). The operating software scans the FRAM looking for initialization records (those that begin with "INI="). These commands are executed in the order that they are encountered.

```
INI=TIME
INI=WAIT 0.5
INI=TIME
INI=EPOC -5.0
INI=NAME FOX21
INI=CALL KCOJFQ
INI=CONF SI5351
INI=CONF 8MA CLK0
INI=FREQ 144.150
INI=MODS S0 360,60
INI=MODS S1 360,90
INI=STAT
```

The TIME command sets the system time from the DS1672 (with no arguments as shown here). The command is sent twice to mitigate an issue with the DS1672.

The EPOC command establishes the local time zone. It is expressed as hours from "ZULU" and assumes that time is stored as "ZULU" (not local time). The author is in the Midwest and using CDT.

The NAME command defines the stations nickname.

This stores the nickname which will be substituted into commands when <NAME> is found in a command. This nickname will be unique for each station.

The CALL command define the stations callsign.

The callsign, stored here, will always sent as part of the sign-on message and the sign-off message to comply with part 95 identification rules.

The callsign which will be substituted into commands when <CALL> is found in a command.

The **FREQ** command selects the *startup frequency*.

The MODS (modular schedule) command defines the message schedule. Up to 10 schedules can be stored (named S0..S9).

5.2 ANN=, Announcement Message Commands

These ANN= commands also run when the system is powered on or reset. The reason for this separation is not discussed here, it is sufficient to know that these command run after the INI= commands.

```
ANN=REM- fox_ann_V2023.fox
ANN=TONE 1.0
ANN=CWPM 30,-1,-1,-1,-1
ANN=BEGN
ANN=TALK <CALL>
ANN=TALK <NAME>
ANN=WAIT 1.0
ANN=BATV V
ANN=BATV I
ANN=WAIT 0.3
ANN=TALK 144
ANN=TALK 225
ANN=TONE 1.0
ANN=CWPM 30,-1,-1,-1,-1
ANN=DONE
ANN=FREQ 144.225
ANN=STAT
ANN=RUN0 S0
```

These **ANN=** commands cause the Fox Transmitter to send a status report over the air. The **TONE** and **CWPM** commands set the CW operating parameters. The **BEGN** command keys the transmitter and send the station callsign. The **TALK** commands verbalize the station callsign and nickname. The **BATV** command verbalizes current battery state. And the **DONE** command sends the callsign again and shuts down the transmitter.

The **RUN0** command enables the S0 schedule which was defined up in the **INI=** commands. This tells the Fox Transmitter to run the S0 schedule.

5.3 TALK directory

A small sample showing a few *TALK Directory* entries from the FRAM.

The audio file name (KC0JFQ, FOX21, etc.) is followed by a single parameter, the data starting address in FLASH.

In this example the data stored in the FLASH is an 8 bit mono WAV file. As such, the WAV header (stored in FLASH) provides additional information necessary to process the file (i.e. length and sample rate).

```
. . .
  TALK=KC0JFQ 51200
. . .
  TALK=FOX21 66048
  TALK=FOX22 70272
  TALK=FOX23 74624
  TALK=FOX24 79872
. . .
```

When we see a TALK (or BATV) command, we scan for a matching TALK= records to find the starting address of the WAV fragment in the FLASH memory.

The waveform data called out is then used to modulate the frequency synthesizer so we can hear it on our receiver.

The waveform data is loaded into the FLASH using a standard Intel HEX file. The utilities used by the author generate the example records with white-space between the fields to make it easier to inspect.

```
:02 0000 04 0000 FA
:20 0000 00 524946465010000057415645666D74201000000001000100A00F0000A00F0000 4F
:20 0020 00 01000800646174612B10000080808080807F7F7F808080808180808080818080 E3
. . .
:00 0000 01 FF
```

5.4 Frequency table

There is an abbreviated frequency table for the SI5351 stored in the zNEO program flash, just enough to determine the frequency offset error. An external table for the SI5351 is then generated that corrects the error. This is then loaded into FRAM to operate the fox transmitter.

```
INI=F0FF -15.000
144.F0FF -15.000
144.100=139C,D2EFF,F4240
144.105=139D,0DAC0,F4240
144.110=139D,3C8BF,F4240
. . .
144.150=139E,BF680,F4240
. . .
144.225=13A1,A21BF,F4240
. . .
```

The **FREQ** command performs a simple text scan of the FRAM for a matching frequency line to find the data to be loaded into the SI5351 registers. As you expect, the frequency is followed by the values to be loaded into the frequency synthesizer registers. It should be clear that you may load as many or as few records as you need, as long as the frequencies you select are in the table.

5.5 Schedule 0

We will now move on to look at an example command sequence that implements the fox transmitters on-air "personality".

This example is from a 6 unit group used for the ICARC hunts. Each transmitter is allocated a 60 second window in a 360 second cycle. The **MODS S0 360,60** from the **INI=** above runs a cycle period of 360 seconds with an offset of 60 seconds. Other schedules in the group will all have the same 360 second cycle period but offsets that increase in 60 second steps.

The S0= in this example corresponds with the S0 in the MODS command. These are the commands that are issued when the S0 scheduling point occurs.

```
S0=BATR
S0=TONE 1.0
S0=CWPM 30,-1,-1,-1,-1
S0=BEGN
S0=TALK <CALL>
S0=TALK <NAME>
S0=WAIT 0.5
S0=TONE 1.5
S0=CWPM 25,-1,-1,-1,-1
S0=WAIT 0.15
S0=BATC EV 7.2
S0=WAIT 0.5
S0=CODE IOWA CITY
S0=CODE AMATEUR RADIO
S0=CODE CLUB FOXHUNT
S0=CODE F W KENT PARK
S0=BATR
S0=TONE 1.0
S0=CWPM 30,-1,-1,-1,-1
S0=DONE
```

These commands are handled in the same manner as described in the **ANN=** section above.

The **BATR** command causes a battery report to appear on the serial port to allow characterizing battery performance. This allows an accurate estimate of run time to be measured.

The **CODE** command produce a CW message at the previously defined audio frequency and word rate. Code can be sent at any rate between 3WPM and 45WPM.

6 Conclusion

The first hunt where these new units were deployed resulted in a dramatic increase in the number of transmitters that were actually located by the hunters. Managing transmitters at the event becomes rather trivial, simply turn the unit on as it is dropped in its hiding place and move on. A brightly colored antenna provides an easy to find visual for the hunter as well as for the event organizer at the end of the hunt. No synchronization procedure is required on the day of the hunt.

Recovering from an accidental power-off requires only that the unit be switched back on. It goes through start-up reporting and then resumes operations on its assigned schedule.

For the hunt, we have moved away from a paper punch that was used to validate a find by punching a log card. Rather, we simply have the hunter pick a *found card* from the transmitter when it is located. The hunter may also record an ID number (unique to each hunt and unit) that appears on each unit if the supply of *found cards* is exhausted. This method requires that a set of ID labels and *found cards* be generated for each hunt. It eliminates the need to provide additional hardware (i.e. a paper punch) at the fox transmitter.

6.1 Acknowledgments

The author would like to offer thanks to all those that have helped in bringing the project into an operational state.

In particular, George Carsner, W0PPF, for the simple question "Why doesn't it talk?". This triggered the voice capability!

Rich Haendel, W3ACO, for advice on implementing and testing the RF section.

And finally, Don Kirchner (KD0L), who I've had the pleasure to work with professionally for many years. Don and I, along with a highly skilled team at the Department of Physics and Astronomy at the University of Iowa have sent many instruments to the outer planets.

6.2 Authors email

kc0jfq@n952.ooguy.com

6.3 Additional Documents

See the authors web site at:

<http://n952.ooguy.com/HamRDF>

The boards are predominantly surface mount. Assembly requires some skill with an iron and a static safe work area. An inexpensive bench microscope (from eBay) is usually necessary to find pin 1!

A complete set of build documents are available on the web site as a compressed tar archive. The latest software for the zNEO is included in the tar archive.

6.4 Board Availability

Transmitter boards may be obtained from the author for \$10 each plus actual shipping costs.

Boards can also be ordered from JLCPCB using the zip files included in the tar archive.

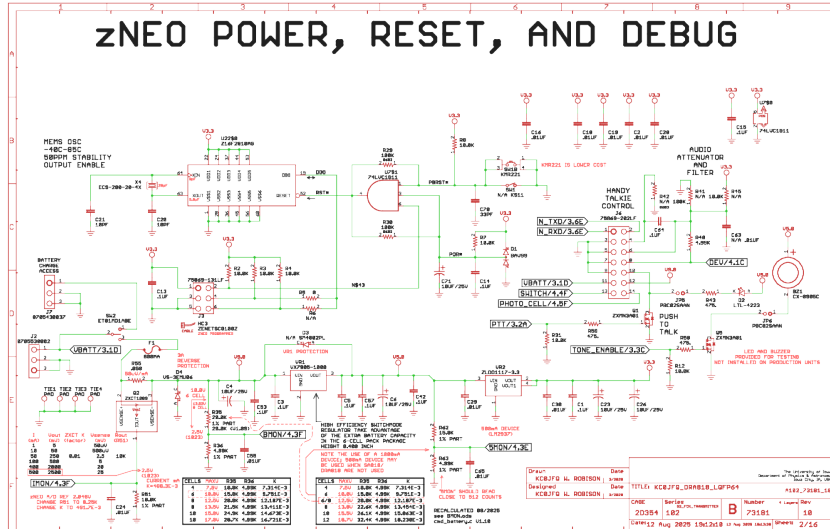


Figure 1: 102-73181-10 Regulators, Reset, HT connector

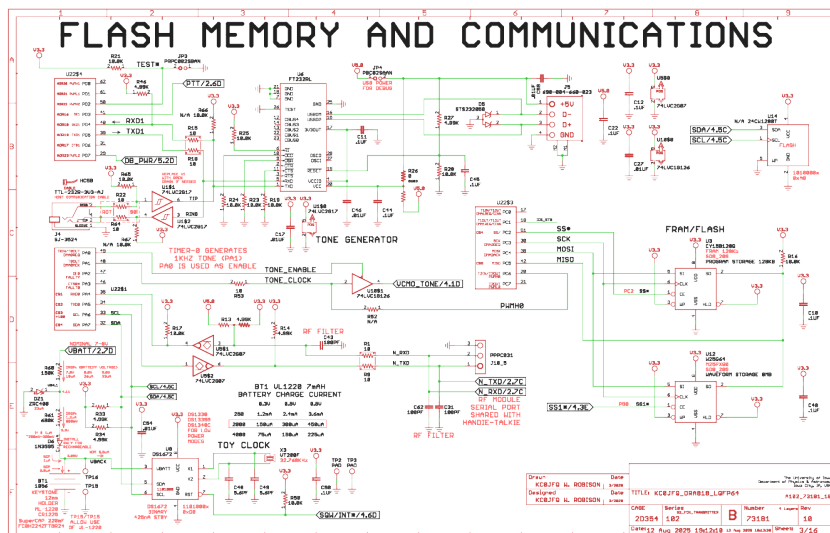


Figure 2: 102-73181-10 Peripherals

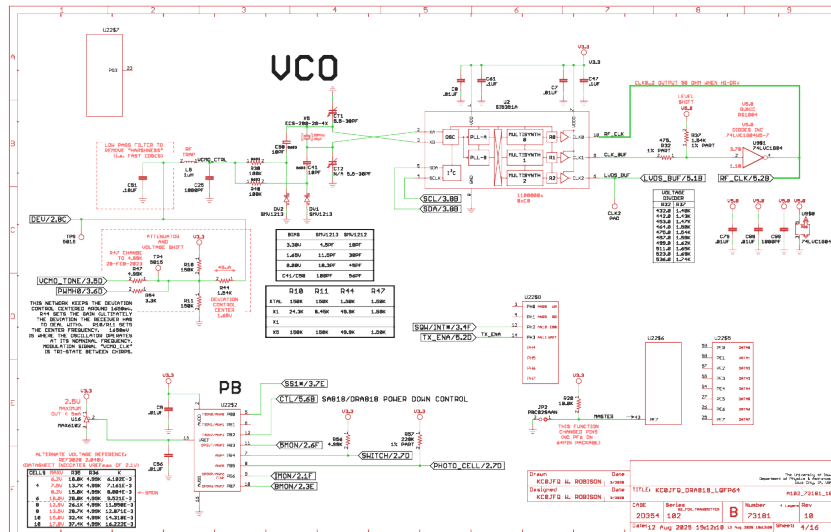


Figure 3: 102-73181-10 Modulation and VCO

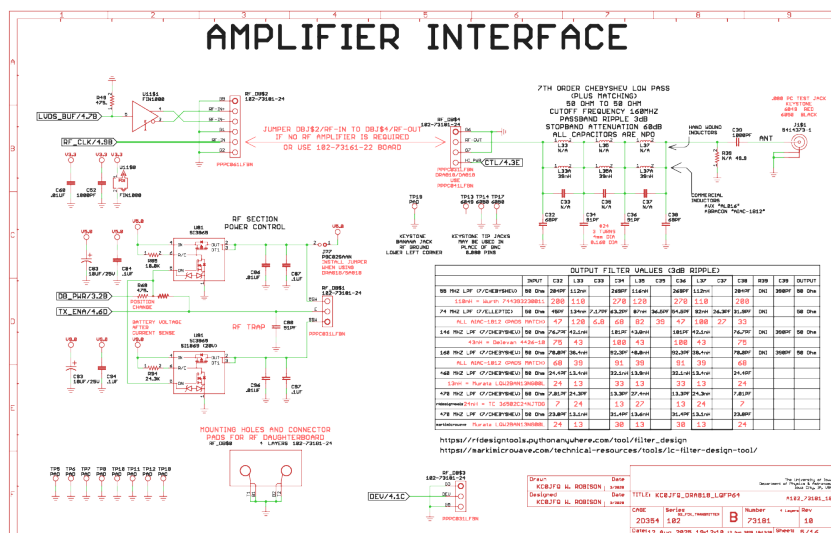


Figure 4: 102-73181-10 Amplifier Interface and Output Filter

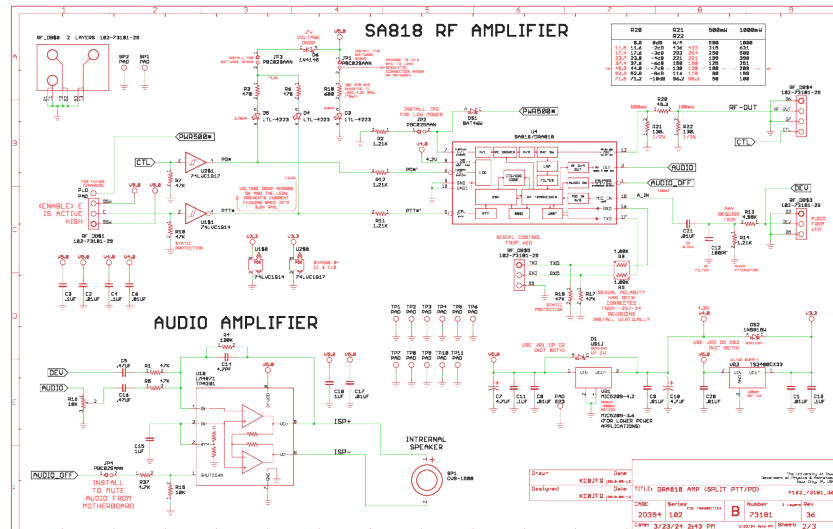


Figure 6: 102-73181-36 DRA818 Module Schematic

The LEDs and the Audio Amplifier would be populated only for software development and testing. Field deployed units have only the SA818 and associated parts installed.

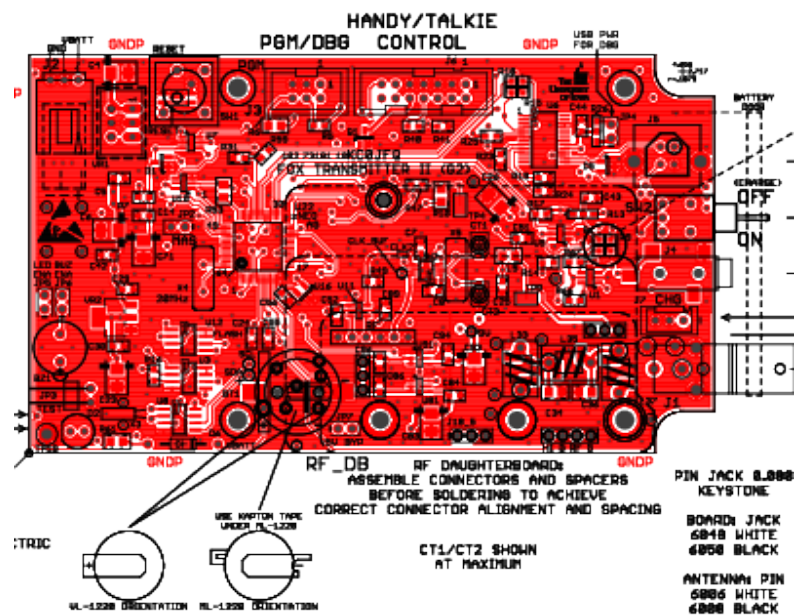


Figure 7: 102-73181-10 Artwork, TOP

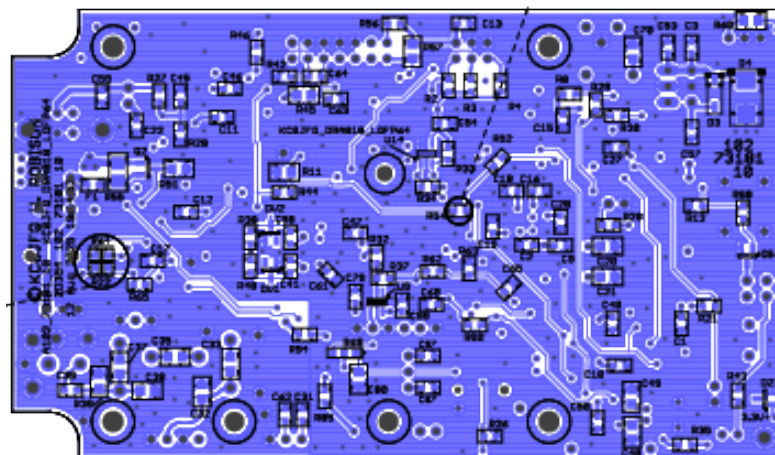


Figure 8: 102-73181-10 Artwork, BOT

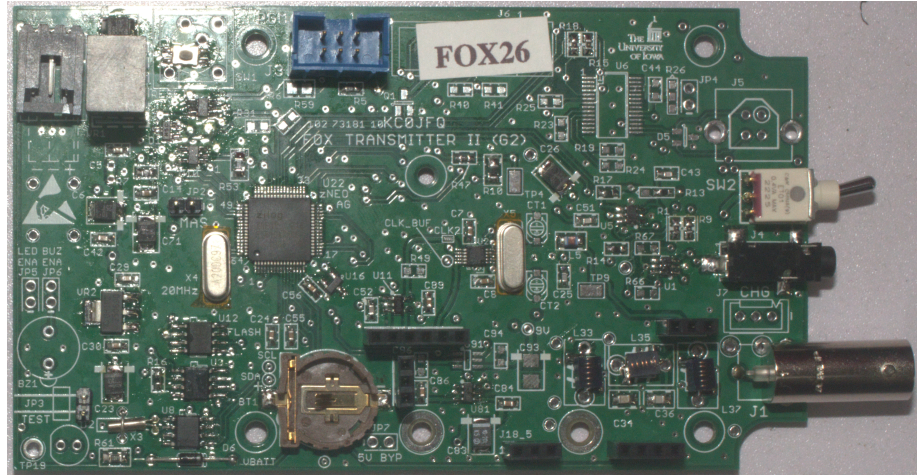


Figure 9: 102-73181-10 Board Image

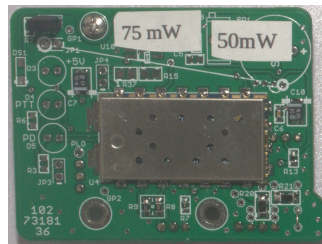


Figure 10: 102-73181-36 Board Image

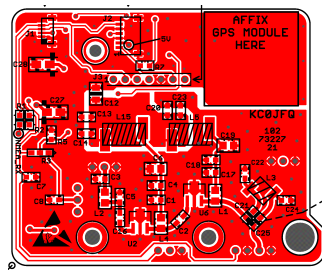


Figure 11: 102-73227-21 Board Artwork