Latest FOX Transmitters

on in a looooong line of transmitters

KC0JFQ: W. Robison

February 27, 2025

Job: fox˙present˙2

File: fox˙present˙2.tex

Outline

Iowa City Amateur Radio Club

Why

Because we can!

- More Flexibility (extremely programmable)
  - Uncanny ability to Fool and Frustrate the hunters
- All setup performed day(s) before the hunt
  - Ususlly set time and date and check battery condition
- No timing critical tasks at the start of the hunt
  - Turn it on when you hide it
  - Turn it on again if you bump the power switch
- Easy on batteries
  - 24 hours run time with CHiRP amplifier.

Hardware Genesis

ICARC Fox hunts started up again in 2018/2019

- WB6EYV MicroHunt Foxhunting Transmitter
  - Uses the ICS525 synthesizer.
  - Fixed Frequency, very low power.
- ICARC 73161 series transmitters
  - Three hardware revisions (all using the same 525 synthesizer).
  - W0PPF (George) asks: "does it talk?".
- ICARC 73176 series transmitters
  - Yes, it talks! (Raspberry-PI based FOX Transmitter).
  - Power pig. Boots up slowly. Susceptible to SD card corruption (dead)!
- ICARC 73181 series transmitters Add PWM audio feature
  - Again, three hardware revisions (ICS307 then SI5351).
    - ICS307 is *end-of-life*; Renesas just keeps on hosing me :-(
    - SI5351 is far more capable; we get everywhere in the band:-)
    - zNEO package change (80-pin package not readily available)
  - Add second FLASH device to store audio. (low cost)
    - (We retrofit the PWM audio feature to the 102-73161-25).

Iowa City Amateur Radio Club

Programming Flexibility

- Frequency: 2M/VHF, 70cM/UHF, *and HF*!
  - Frequecy selection programmable within band
  - SI5351 can generate VHF and HF frequencies
  - UHF requires SA818U/DRA818U tranceiver module
  - SI5351 is lower power than SA818/DRA818
- Transmit Power
  - SA818/DRA818 may run 500mW or 1000mW
    - although we seem to get less than 250mW out
  - SI5351 uses several RF daughter-boards (up to around 175mW)
  - Matching network on RF daughter-board
  - Attenuator network on RF daughter-board
- CW and voice
  - CW audio tone programmable (Frequency)
  - CW chipping rate programmable (Spacing/Timing)
  - Voice sample rates 4KHz, 5KHz, or 8KHz
    - Rules imposed bandwidth limits suggest
    - 4KHz or 5KHz is all thats necessary

Iowa City Amateur Radio Club

Timing Flexibility

Based on modular arithmetic using time from TOY clock

- Scheduling Parameters:
  - TOY (Time of Year, seconds from some epoch) DS1672
  - TOD (calculate from TOY clock) SYSTEM
  - Period (seconds, from setup in FRAM) MODS
  - Offset (must be less than seconds, from setup in FRAM) MODS

Calculate Time of Day: **(TOD = TOY % 86400)**
Transmit when (**(TOD % Period) == Offset**)

- Divide time-of-day by the scheduling-period taking only the remainder
- Compare the remainder with the scheduling offset
- Run transmit sequence (program) when they match!
- This calculation/comparison occurs at the RTI rate (10mS)

Start hunt (**STAR** 10:00:00)

- Scheduling is suspended until specified time occurs
  - At 10 A.M. in this example
- Early setup while avoiding early detection!
  - Keep quiet as the hunters register and prepare for the hunt
  - Useful for a formal hunt, where we're giving out prizes!

Iowa City Amateur Radio Club

Can we achieve sub-second resolution?

- Host side *tricks*
  - Host system synchronized to UT using NTP
    - This achieves sub-millisecond precision on the host.
  - Time setting utility wait for seconds to roll and sends time update mesage to target. (Target within milliseconds)
- Fox transmitter side *tricks*
  - Assume writing to the DS1672 resets sub-seconds to zero. Seems to work, datasheet isn't clear on side-effects from write to seconds register.
  - Read DS1672 time register until LSB rolls
    - (Fox transmitter repeats this for up to a bit over one second)
  - Fox system runs with a 10mS *tick*.

In the end, we get reasonably tight timing, allowing for some interesting operating schedules.

- Transmitters can, in effect, carry on a *conversation* amongst themselves.

Hardware

ZiLOG zNEO. 16 bit expansion of ZiLOG Z8/eZ8

- Hybrid Von_Neumann/Harvard architecture
  - Compiler friendly Architecture; address space is **not** split
- 128 K Byte program flash
- 4 K Byte SRAM area

SMPS Regulator. switch-mode: more efficient than linear, longer battery life

Battery Current and Voltage Monitors.

USB or logic-level interface. programming/setup serial channel from host computer.

Second serial interface. controls the DRA818/SA818 module.

- Interface for external radio, (i.e. a hand-held tranceiver)

TOY clock. Synchronize all transmitter schedules.

PWM Channel. Voice for identification and status reporting.

SI5351 synthesizer. HF and VHF carrier, FM modulation through reference crystal.

RF Daughterboard. RF amplifier on daughter-board allows for experimentation.

RF Daughterboard power switch. unpowered when idle.

Output Filter.

- Lowpass filter between RF stage and the output (BNC) connector.

Hardware: RF Amplifier

Amplifier 102-73181-28. A1A and F1A/F3E up to around 100mW

- MMIC gain element in SOT89 package (IF amplifier: Class-C).
- CHiRP specific amplifier (RF power switching using**TX ENA** net).
- Controlled rise time power switch (C15/R4 next page).
- default mode F1A/F3E, **CONF CW** to operate A1A
- Wildlife tracker mode **CHRP** *tone.,per.,dur.,count*

Amplifier 102-73181-36. A1A and F1A/F3E up to around 1000mW

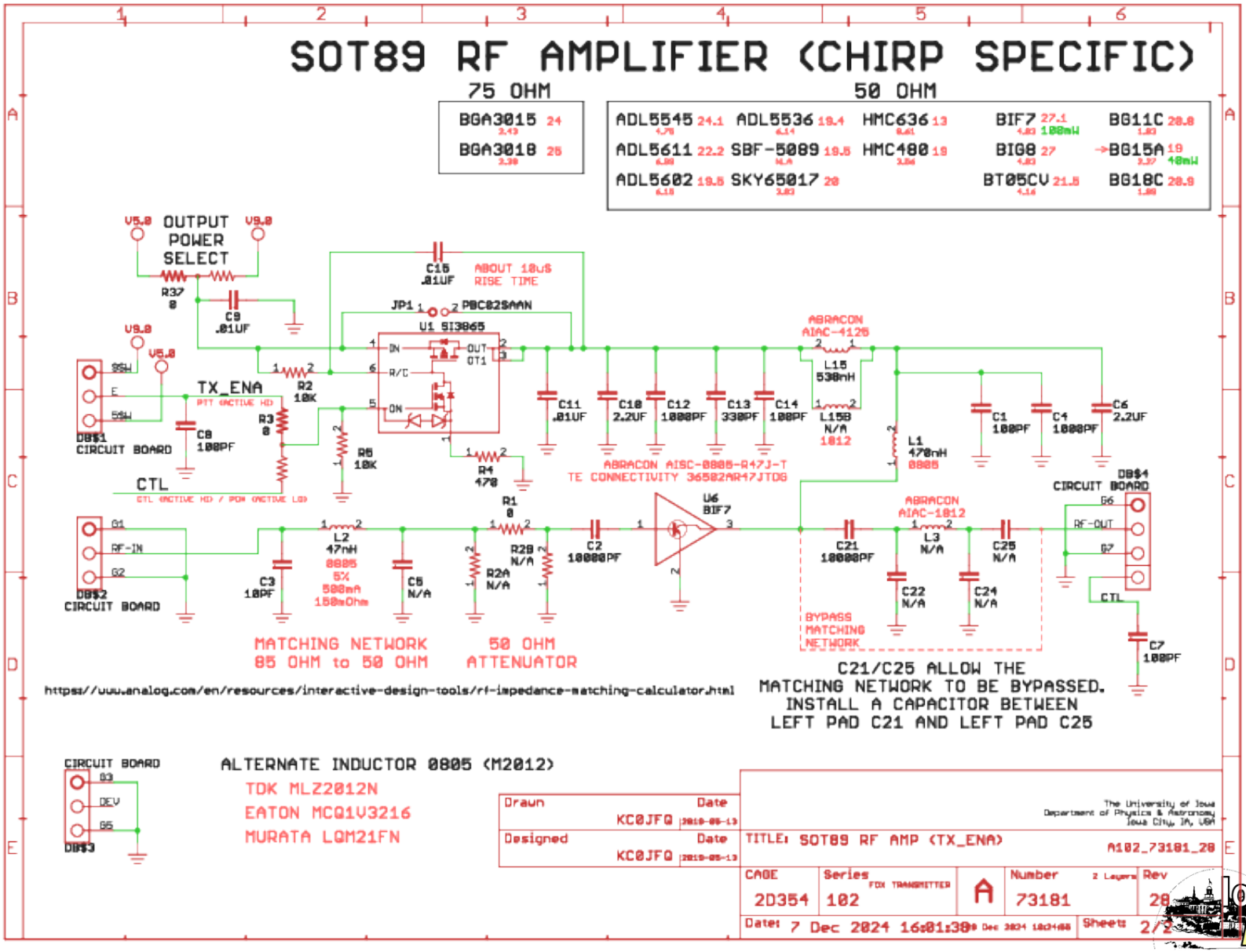
- DRA818/SA818 VHF or UHF tranceiver module.
- Works with CHiRP (PTT* using **TX ENA** net).
- default mode F1A/F3E, **CONF CW** to operate A1A
- Wildlife tracker mode **CHRP** *tone.,per.,dur.,count*

Amplifier OTHERS! Daughter-board

- Experiment with RF designs without having to rebuild the digital section
- Easy and quick to swap out the RF amplifier

Iowa City Amateur Radio Club

Hardware: MMIC Schematic



SI3865

Pwr Swt

C15/R4

Slow St

BIF7

Pwr Amp

C3/L2

Match

C21

DC Block

Iowa City Amateur Radio Club

Software

Entirely written in "c". Small number of in-line assembly instructions.

Very modular. 40+ individual *source units* make up the load image.

Architecture is a simple loop and a few interrupt handlers.

- Look for incoming command buffer.
- Look for scheduling match (**(TOD % Period) == Offset**)
- HALT (zNEO instruction stops processor). Low Power!

Clock Interrupt.

- TOY clock sets system clock at startup.
- RTI interrupt updates system clock (100 ticks/second).

UART Interrupt.

- buffer incoming commands until a **0x0D** is detected.
- *special case* InTEL HEX records.
  - HEX records loaded into FLASH (Checksum must be valid!)

CW Interrupt. Controls **TONE ENABLE** net.

- Interrupt period is set to CW chipping rate (one *dit* time)
- Interrupt routine counts out longer periods (nominal *dah* is 3 interrupts).
- CW timing controlled/changed by **CWPM** command

Iowa City Amateur Radio Club

Seems to be constantly expanding

Currently take up about 120KB out of 128KB)

Latest additions: CHiRP features.

Chirping like a bird **and** chirping like a RADAR!

Wildlife Tracker:

Sends out a short tone burst (like a CW dit or dah).

Repeats on a schedule defined in the command.

RADAR-like CHiRP:

Sends out an audio file. We will see about audio files in a bit

Repeats on a schedule defined in the command.

Prototype audio chirp from 300Hz to 1500Hz in 500mS.

System timing is tight enough to allow sending these chirps every two seconds from a transmit group (i.e. In a group of 6 transmitters, each one transmits every 12 seconds).

Synthesizer: SI5351

SI5351 exists only on 102-73181-5 and 102-73181-10 boards.

- I2C device with very large register space.
  - Many bits to load!
- Small table built into zNEO. (only for initial setup and characterization)
  - Internal table for frequency error measurement, then load an external table.
  - Table entry has frequency and three register values.
    - many other SI5351 fields need to be written with *mostly* static values!
  - Working frequency table stored in FRAM . (i.e. the external table)
- Three outputs from SI5351. Only one of them can be selected.
  - CLK0: directly to RF daughter-board. 85 Ohm, Drives SOT89 amplifiers
  - CLK1: buffered high-speed CMOS logic gate. Not normally populated
  - CLK2: buffered LVDS driver. LVDS pair sent to daughter-board
- Command path to directly load the *Multi Synth* registers.
  - **NO sanity checking.**
  - Allow configuring the SI5351 for any frequency.
  - Easy to generate out-of-band signals. (like 10M or 6M)

Synthesizer: DRA818/SA818

Functions correctly only with 102-73181-10 boards.

- Earlier boards are missing second on/off control

Low-cost tranceiver module. (Rx channel as well as Tx channel.)

- Daughterboard 102-73181-36 used to mount DRA818/SA818 module.
- Serial command interface for frequency selection.
- Digital levels for power-down (**PD***) and Push-to-Talk (**PTT***).

Daughterboard 102-73181-36 connects receive channel to tiny speaker.

- Low power audio amplifier.
- Amplifier disabled during transmit.
- Audio path not populated for typical fox hunt application.
- Audio modulation from motherboard connects to audio-in.
- Board can be built as a software test fixture (speaker and LEDs).

The 102-73181-5 doesn't cut it!

- This artwork doesn't split the power down (**PD***) and the push-to-talk (**PTT***) signals.
- 80-pin zNEO package is un-obtanium!!!

Doesn't seem to produce advertised power.

- DRA818 and SA818 seem to be six to eight dB down from spec.

Synthesizer: ICS525

Exists only on 102-73161-25 boards.

ICS525 no longer available

Current software release provides an upgrade path for older transmitters

The new software is based on the software from these units.
The new software is more modular. Streamlined command decoder.
Most existing commands carry forward unchanged.

Frequency Selection.

Table based, much like SI5351 implementation.
Support for directly programming the 3 registers.
Frequency selection much more limited due to ICS525 architecture.
19 discrete bits set the frequency (the 3 registers).
RF modulation achieved by varying the load on the reference crystal (same method as the SI5351 modulator).

Poor RF performance.

power from ICS525 spotty, some good, some bad.

DEPRECATED, KAPUT; NO-LONGER-BUILDABLE

Programming (serial access)

The primary design philosophy with respect to loading the **fox transmitter** is that it is simple and must not require vendor specific tools.

No programming dongles, **no special USB drivers**.

The USB interface is ubiquitous these days, so we can choose between an on-board USB serial (from **FTDIchip**) or a logic level serial interface that makes use of a USB serial cable. We can save the cost of multiple USB serial chips by leaving the USB serial chip off of the fox transmitter and using a single USB serial cable to service multiple units.

The USB cable I use is from **FTDIchip**. It terminates in a standard 3.5mm TRS jack (i.e. stereo headphone jack). This cable runs around $25 compared to about $5 per *fox transmitter* for the **FT232R** USB UART device. We break even with a full set of 5 *fox transmitters*.

Programming

Access to the *FOX Transmitter operating sequences (programs)*

therefore, uses the 3.5mm serial port.

I use a Linux utility to set the time and

load *FOX Transmitter operating sequences (programs)*

Early boards (102-73161) have USB UART on board.

Standard USB-B connector (not mini or micro).

Therefore each board has unique USB port (COM492 on Windoze)

Linux uses an ID string generated from fields in the USB device.

FOX10: /dev/serial/by-id/usb-UIowa_KC0JFQ_FOX_V2_2078-0-0105-if00-port0

FOX14: /dev/serial/by-id/usb-UIowa_KC0JFQ_FOX_V2_2078-0-0109-if00-port0

Night before we open every enclosure (USB cable) to update time

UUGH! Open each box, update time, close box.

Later boards switch to a logic-level port using same zNEO serial port

All stations share a single USB serial cable (reduce overall cost).

ALL: /dev/serial/by-id/usb-KC0JFQ_KC0JFQ_Debug_5000-0-0115-if00-port0

Night before we just switch the USB-UART cable from box-to-box.

Use the **fox_simple** utility to set the time from the Linux host!

FTDI Chip part number: **TTL-232R-3V3-AJ**

Retain pads for USB UART (not populated).

Iowa City Amateur Radio Club

System Commands

System ConfigurationCommands

- **ONCE** Execute sequence (program) one time. (for testing)
- **RUN0** Enable the specified schedule.
- **STAT** Status Report.
- **CONF** Hardware Configuration.
- **TIME** Read or write TOY clock.
- **STAR** Start scheduling at specified time.

System Setup Commands

- **CALL** Set FCC Callsign. (W0IO W0JV KC0JFQ, etc.)
- **NAME** Set unit "nickname". (FOX1, FOX2, FOX3, ...)
- **TIME** Set system time from TOY clock. (Scheduling ignores days)
- **EPOC** Set local time zone. (In Iowa we use -5.0 or -6.0)

Remember that **STAR** command for starting at a specific time?

Both **CALL** and **NAME** can be substituted into the **CODE** and **TALK** commands using the <CALL> and <NAME> construct. These *System setup commands* should only appear in the **INI=** file.

Iowa City Amateur Radio Club

Program Commands

Program Commands

- **BEGN** Enable transmitter, send signon message.
- **DONE** Disable transmitter, send signoff message.
- **CODE** Send CW message.
- **TALK** Send Voice message.
- **CHRP** Emulate wildlife tracker (or RADAR-like chirp).
- **BATV** Battery Report (Voice).
- **BATC** Battery Report (CW).
- **BATR** Battery Report (operating time analysis).

These *Program Commands* appear in the *operating sequence*

Program Scheduling Commands

- **MODS** Load (or set) a schedule.

Iowa City Amateur Radio Club

FRAM and FLASHCommands

FRAM Commands

- ESAV Save a command string to the FRAM device.
- EZER Zero out a command. Allows for overwrite.
- ERAS Erase a command. Changes it to a dummy command.
- EDMP Dump FRAM.
- EDID Dump FRAM and FLASH JEDEC ID bytes.

FLASH Commands

- HERA Erase entire FLASH device.
- HDMP Dump all or parts of the FLASH device.
- :hex Load FLASH device using InTEL HEX records.
  - Only mechanism to write FLASH device.
  - Standard InTEL HEX records (extended address record).
  - ignores whitespace (to improve readability).
  - InTEL HEX file checksum **must** be valid!

TEST Commands

Commands

- **HALT** Halt processor asm(" HALT");.
- **STOP** Stop processor asm(" STOP");.
- **REST** Reset Processor.
- **TEST** Test routines.

STOP requires a hardware reset or power cycle!
The **STOP** command is benign.

Test routines are used to exercise various partsof the system during hardware and software debugging. There is room in program flash to leave these diagnostic and testing routines in place.

The **TEST** commands have the potential to damage hardware if used incorrectly. If you aren't using an oscilloscope, don't run the **TEST** cmmand!

Iowa City Amateur Radio Club

TALK Directory

Listing 1: TALK directory

```
esav TALK=BATTI 0
esav TALK=BATTV 4224
esav TALK=REG5 8704
esav TALK=POINT 13824
esav TALK=V_HZ 15232
esav TALK=V_KHZ 17664
esav TALK=V_MHZ 20864
esav TALK=V_N0 24064
esav TALK=V_N1 26752
esav TALK=V_N2 28544
esav TALK=V_N3 30720
esav TALK=V_N4 32640
esav TALK=V_N5 34560
esav TALK=V_N6 36736
esav TALK=V_N7 38528
esav TALK=V_N8 40448
esav TALK=V_N9 41984
esav TALK=V_MAMP 44416
esav TALK=V_VOLTS 48128
```

Directory entries for the audio clips.

The begining of the **TALK directory**.

Name and starting address in FLASH.

INI File

Listing 2: INI File begin

```
#       Our Epoch is CDT: -5 Hours from Zulu
#       Set system time from DS1672
#
esav INI=TIME
esav INI=WAIT 0.5
esav INI=TIME
esav INI=EPOC -5.0
esav INI=NAME 'name'
esav INI=CALL 'call'
#
#esav INI=CONF BMON 12.5V
esav INI=CONF 'synth_dev'
esav INI=CONF 'synth_set1' 'synth_set2'
#esav INI=CONF DRA818
```

The INI= file; Define who we are, our personality.

I use a single setup file to load FOX20..FOX32

'call', **'name'**, and **'run'** are substituted from the *fox_simple* utility command line.

This file runs when one or no jumpers installed

We start all stations on the same announce frequency,

then change to the operating frequency a bit later...

Iowa City Amateur Radio Club

INI Schedules

Listing 3: INI File end

```
esav INI=CONF 'spare1' 'spare2'
esav INI=FREQ 144.150
esav INI=BATR
#
#       Set schedules,  leaving ONLY
#
REM- 01234567890123456789012345678901
esav INI=MODS S0 'run'
esav INI=MODS S1 'run'
esav INI=REM- MODS S2 'run'
esav INI=REM- MODS S3 'run'
esav INI=REM- MODS S4 'run'
esav INI=REM- MODS S5 'run'
esav INI=MODS S6 'runs6'
esav INI=MODS S7 'run'
```

The INI= file; Define our operating schedule.

Finishing up the **INI=** file.

Define up to ten schedules.

We take the primary schedule **S0=** *period* and *offset* from command line supplied to the *fox_simple* utility.

The **STAT** command provides visibility for debugging

ANN File

Listing 4: ANN File

```
#
#
esav MAS=CWPM 35,-1,-1,-1,-1
esav MAS=STAT
#
#
#
#  We're making use of the <CALL> and <NAME> substitution
#     inside the fox transmitter!!!
#
esav REM- fox_ann_V2025.fox
esav ANN=TONE 1.0
esav ANN=CWPM 30,-1,-1,-1,-1
esav ANN=BEGN
esav ANN=BATR
esav ANN=TALK <CALL>
esav ANN=TALK <NAME>
esav ANN=WAIT 1.0
```

The ANN= file The system announce message.

- Runs **after** INI= when **no** jumpers are installed. Tell 'em we're alive!
- More parameter substitution from command loader: **'freq..'** and **'sched..'**
- Parameter substitution from INI= setup: <CALL> <NAME>
- Frequency change to operating frequency at **'freq'**
- Schedule **S0** enabled in the last command

Iowa City Amateur Radio Club

TEST File

Listing 5: TEST File

```
esav INI=MODS S9 360,15
esav INI=STAT
```

The TEST= file; system test

Runs **after** INI= when TEST jumper installed
You are free to do whatever you want here
ANN= message is **not** sent

Perhaps a comprehensive performance test?

Or a complete functional test?

Iowa City Amateur Radio Club

MAS File

Listing 6: MAS File

```
#
#
```

The MAS= file; alternate system test

Runs **after** INI= when MAS jumper installed
You are free to do whatever you want here
ANN= message is **not** sent

The label (and function) is a leftover from earlier designs where we had the notion of updating time in the field.
One station, with this jumper installed, would emit time messages out the network port while all other station would listen for the time update message,
Well, we scrapped that to control the DRA818/SA818

So this jumper simply causes the **MAS=** sequenct to run.

System Recovery

System Recovery (error recovery)

Install both **MAS** and **TEST** jumpers
Nothing is read from either the **sequence** or **waveform** memory.
Use to recover from rotally fouled up sequences

When you really screwed it, so it won't even talk to you!
With both jumpers in, the software skips all setup files...

Yes, it has been used to recover from a *FUBAR*
That *FUBAR* triggered the software update
to implement this recovey feature.

Help Pages

System Help Pages

There is a fairly comprehensive list of help items built into the software as part of the command decoder.

Enter the **HELP** command to get a list of all commands implemented in the operating software.

Most commands you will use will have a brief synopsis of the command arguments.

You can supply a small bit of text to the **HELP** command to limit the volume of the response. Something like **HELP SYS** to see commands the contain the string "SYS" in them.

Help 1

Listing 7: fox27.help_1

```
sts01,00* TEST HELP **  TEST HELP **  TEST HELP **
sts01,00* Idx MNE  Class     ————————Arguments———————— Command Function
sts01,01*   1 HELP SYS                                 Help Menu and Items
sts01,02*   2 HELP SYS       <string>                  matching help items
sts01,03*   3 ONCE SYS       <name>                    Test run the named seqwuence
sts01,04*   4 REM– SYS                                 Remark, (side–effect: stops schedules)
sts01,05*   5 RUN0 SYS                                 RUN ALL Schedules
sts01,06*   6 RUN0 SYS       <name>                    RUN Specific Schedule
sts01,07*   7 STAR SYS       <time>                    Start running schedules at specified ti
sts01,08*   8 IDLE SYS                                 STOP ALL Schedules
sts01,09*   9 STAT SYS       <flag>                    System Status, (I)ident scan
sts01,10*  10 CONF SYS       <keywords>                Hardware Configuration
sts01,11*  11 TOYC SYS       <res> (250 2K 4K NONE)    Hi chg rte DS1672 bat
sts01,12*  12 TIME SYS       <time value>              Set Time (set DS1672)
sts01,13*  13 D525 SYS       <sub–command>             ICS525 debug routines
sts01,14*  14 TIME SETUP                               Time from DS1672 to System (NO Argument
sts01,15*  15 EPOC SETUP     <hours>                   Epoch offset (i.e. time zone)
sts01,16*  16 CALL SETUP     <call>                    FCC Assigned Callsign
sts01,17*  17 NAME SETUP     <nick>                    Local Nickname
sts01,18*  18 NICK SETUP     <nick>                    alias for "NAME", but don't use it!
```

Help 2

Listing 8: fox27.help_2

```
sts01,19*   19 TONE PGM       <freq>)                        Audio Tone (in KHz)
sts01,20*   20 CWPM PGM       <wpm gap1 gap2 gap3>           CW Chipping Rate
sts01,21*   21 FREQ PGM       <freq>                         Frequency (in MHz)
sts01,22*   22 5351 PGM       <key>,<value>,<value>,...SI5351 setup group
sts01,23*   23 BEGN PGM                                     Key TX and Send Callsign (CW)
sts01,24*   24 CODE PGM       <message>                      Send Message (CW) up to 22 char
sts01,25*   25 TALK PGM       <file-name>                    Play Voiced Message (EDMP TALK)
sts01,26*   26 WAIT PGM       <secon.ds>                     Wait (simple delay)
sts01,27*   27 CHRP PGM       <tone> <dur> <del> <cnt>Send carrier chirp
sts01,28*   28 DONE PGM                                     Send Callsign (CW), SK (CW), and unkey
sts01,29*   29 BATC PGM       <mod>,<key>,<setpoint>         Transmit Code Battery Report
sts01,30*   30 BATV PGM       <mod>,<key>                    Transmit Vocal Battery Report
                                        mod: "E" encode (not CW) for BATC
                                        mod: "B" battery reading taken before BEGN
                                        mod: "A" battery reading taken after BEGN
                                        key: "V" battery voltage,
                                             "I" battery current,
                                             "R" 5V rail
sts01,31*   31 MODS SCHED     <Sname period offset>          Modulus Schedule Set
sts01,32*   32 MODC SCHED     <Sname=>                       Modulus Schedule Clear
```

Iowa City Amateur Radio Club

Help 3

Listing 9: fox27.help_3

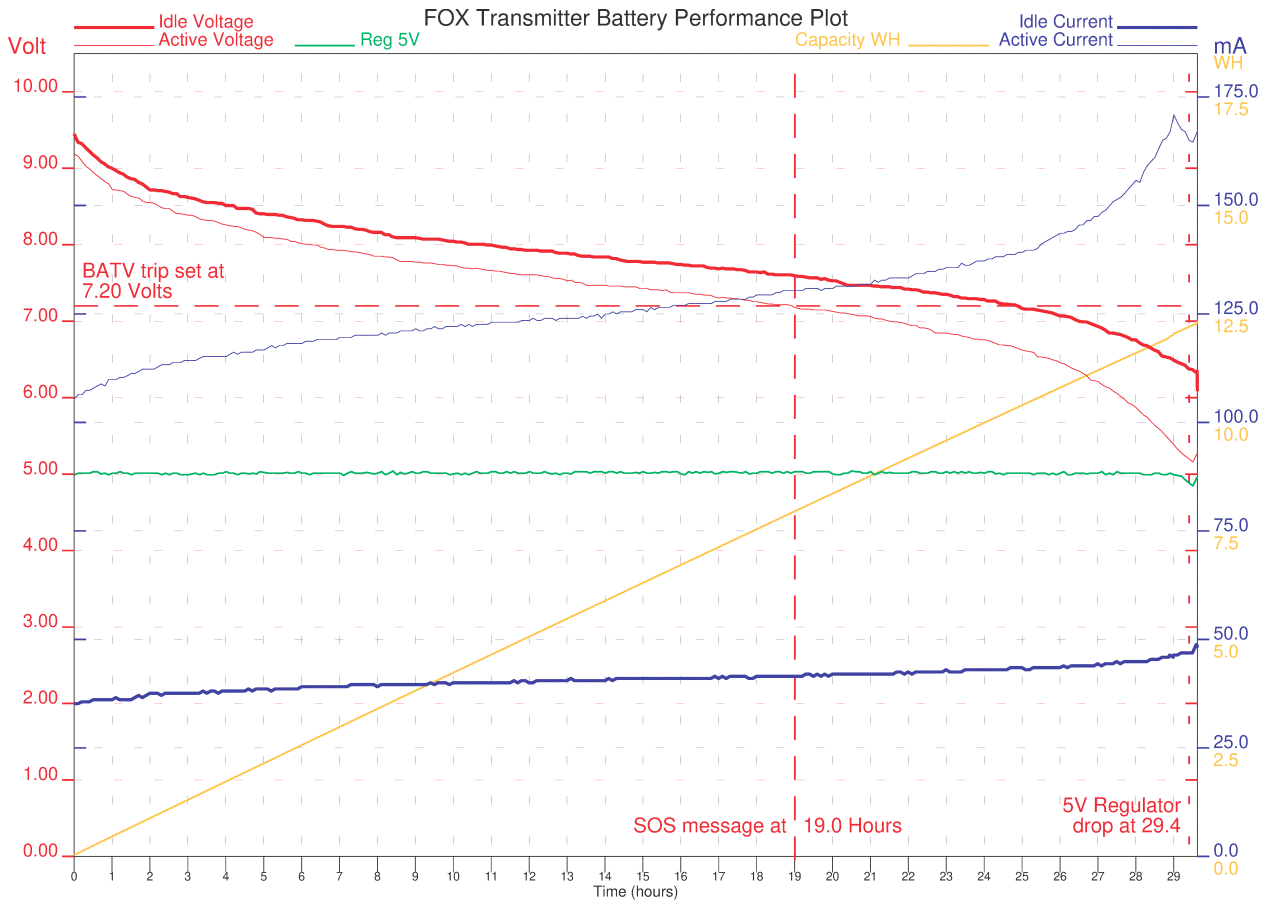
```
sts01,33*   33 TALK DIRECTORY esav TALK=name,Strt,Len,rate      (appears in FRAM as the TALK=
                                                               Waveform Directory Entry
                                                               rate keys: 4K 5K 8K 10K 16K
sts01,34*   34 freq DIRECTORY esav 144.150=13BF,70E40,F4240,100 (appears in FRAM as frequency
                                                               Register Parameters are Synthesizer dep
sts01,35*   35 ESAV FRAM       NAM=<text>                      Save named record in next free location
sts01,36*   36 EDMP FRAM       "match string"                  Dump active records
sts01,37*   37 EDID FRAM                                       Flash JEDEC-ID table dump (PROG & WAVE)
sts01,38*   38 ERAS FRAM       <number> or "DEV"               Rewrite <record> to REM- (DEV, QTR, HA
sts01,39*   39 EZER FRAM       <number>                        Erase <record> to ZERO
sts01,40*   40 ETAB FRAM                                       Dump JEDEC-ID device table
sts01,41*   41 HERA FLASH      ALL                             Hex erase (entire WAVE device)
sts01,42*   42 HDMP FLASH      <len-32B-lines <hex-start <*>>> Hex dump (WAVE device)
sts01,43*   43 H56K FLASH                                      Fast terminql bit rate
sts01,44*   44 :hex FLASH-HEX :llaaaattddddddddcc             Intel HEX loader (WAVE device)
sts01,45*   45 HALT TEST                                      Halt Processor
sts01,46*   46 STOP TEST                                      Stop Processor
sts01,47*   47 REST TEST                                      Reset System
sts01,48*   48 TEST TEST                                      Hardware Test Subsystem
STS01,49* Handler_HELP (cmd_help.c*)  3.00 Sec
```

Battery Plot

BATR plot

Battery Runtime test

- Fresh Battery
- After 19 hours battery drops below the 7.2V trip point
- Near the end, the 5V rail starts to drop out of regulation
- Also note that all of the currents rise as battery voltage collapses



- With fresh AAA (cheapest ones from Amazon) expect around 19 to 20 hours of operation using the CHiRP amplifier. Perhaps 12 to 14 with the DRA818.
- Estimate 1 hour setup, 2 hour hunt, 1 hour teardown and you will get 4 or 5 hunts on a set of batteries

Iowa City Amateur Radio Club

Build Support

Sometimes **Linux** sucks

- Fedora40 upgrade took out the IDE!
- ZDS-II (ZiLOG development tool) runs under WINE
  - **NO** support for USB programmer
  - The **ZENETSC** ethernet programmer worked
- 102-73220-20 and 102-73220-32 provide hardware interface for the job!
  - 102-73220-20 USB to RS4825 for sensor network
    - connector for Raspberry PI Zero
    - R-PI connector provides path to attach level shifter board
  - 102-73220-32 ZiLOG *Z-DBG* Programming interface
    - 6-pin flat cable to target
    - lets make more software!
    - not as fast as ZiLOG **ZENETSC** programmer

Iowa City Amateur Radio Club