

Latest ICARC FOX Transmitters

in a looooong line of transmitters

KC0JFQ: W. Robison

August 4, 2024

A dive into the hardware and software used to implement the ICARC Fox Hunting FOX Transmitters.



Latest ICARC FOX Transmitters

Presentation Outline

Why

Hardware Genesis

Programming Flexibility

Timing Flexibility

Hardware

Software

Synthesizers

Programming

Help Pages



Latest ICARC FOX Transmitters

Why

Because we can!

More Flexibility (extremely programmable)

Uncanny ability to Fool&Frustrate the hunters

All setup performed day(s) before the hunt

Ususlly set time&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

Latest ICARC FOX Transmitters

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 ICS525 synthesizer).

W0PPF (George) asks: "does it talk?".

ICARC 73176 series transmitters

Yes, it talks! (Raspberry-PI FOX Transmitter).

Power pig. Boots up slowly.

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 73161 models).



Latest ICARC FOX Transmitters

Programming Flexibility

Frequency 2M/VHF, 23cM/UHF, *and HF!*

- Frequency selection programmable within band

- UHF requires SA818U/DRA818U transceiver module

- SI5351 is lower power than SA818/DRA818

- SI5351 can generate HF frequencies

Transmit Power

- SA818/DRA818 may run 500mW or 1000mW

- SI5351 uses several RF daughterboards (up to around 100mW)

- Matching network pads on RF daughterboard

- Attenuator pads on RF daughterboard

CW and voice

- CW audio tone programmable

- CW chipping rate programmable

- Voice sample rate 4KHz, 5KHz, or 8KHz

Latest ICARC FOX Transmitters

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)

Period (seconds, from setup in FRAM)

Offset (must be less than seconds, from setup in FRAM)

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 program when they match!

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

Scheduling is suspended until specified time occurs

Early setup while avoiding early detection!



Latest ICARC FOX Transmitters

Hardware

ZiLOG zNEO. 16 bit expansion of ZiLOG Z8/eZ8

128 K Byte program flash

4 K Byte RAM area

SMPS Regulator. switch-mode: more efficient than linear.

Battery Current and Voltage Monitor.

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

Second serial interface. External transceiver or DRA818/SA818 module.

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

TOY clock. Synchronize all transmitter schedules.

PWM Channel. Voice for identification and status reporting.

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

RF Daughterboard. RF amplifier on daughterboard allows for experimentation.

RF Daughterboard power switch. unpowered when idle.

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

Latest ICARC FOX Transmitters

Hardware

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).

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 transceiver module.

Work with CHiRP (PTT* using **TX_ENA** net).

default mode F1A/F3E, **CONF CW** to operate A1A

Wildlife tracker mode **CHRP** *tone.,per.,dur.,count*

Latest ICARC FOX Transmitters

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).

Clock Interrupt.

- TOY clock sets system clock at startup.

- 10mS interrupt updates system clock (100 ticks/second).

UART Interrupt.

- buffer incoming commands until a **0x0D** is detected.

- buffer InTel HEX records.

CW Interrupt. Controls **TONE_ENABLE** pin.

- Interrupt period is set to CW chipping rate (*dit*).

- Interrupt routine counts out longer periods

 - (*dah* is 3 interrupts).



Latest ICARC FOX Transmitters

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 (unit setup) table built into pgm flash when software is built.

Internal table used to measure frequency error, then select external table.

Table entry has frequency and six register values. many other SI5351 fields written!

Working frequency table (i.e. the external table) stored in FRAM .

Three outputs from SI5351. Only one of them can be selected.

CLK0 goes directly to RF daughterboard (50 ohm!). Drives SOT89 amplifiers

CLK1 is buffered by a high-speed CMOS logic gate. gate addition required

CLK2 is buffered by an LVDS driver. LVDS pair sent to daughterboard

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)



Latest ICARC FOX Transmitters

Synthesizer: DRA818/SA818

Functions correctly only with 102-73181-10 boards.

Low-cost transceiver 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.

Doesn't seem to produce advertised power.

Six to eight dB down from spec.



Latest ICARC FOX Transmitters

Synthesizer: ICS525

Exists only on 102-73161-25 boards.

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.

Latest ICARC FOX Transmitters

Programming

FOX Transmitter operating programs.

Loads through 57,600 b/S serial channel.

Stored in FRAM device.

FRAM does not require device erase, we can change single commands when developing new operating programs.

FRAM size somewhat limited, but doesn't seem to be a problem (other than cost for a large device).

Simple verb-object structure

Four character command stem (the *verb*).

A variable number of parameters (the *object*) (command dependant).

Small subset actually used to implement a fox message.

Waveform data (audio) stored in FLASH.

Large device for not too many dollars!

Can boost serial rate to 115,200 b/S.



Latest ICARC FOX Transmitters

Programming

Access to the *FOX Transmitter operating programs* uses serial port.

Linux utility to set time and load *FOX Transmitter operating programs*.

Early boards have USB UART on board.

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

Each board has unique USB port (COM492 on Windoze)

Linux uses ID string in the USB device.

FOX10: /dev/serial/by-id/usb-Ulowa_KC0JFQ_FOX_V2_2078-0-0105-if00-port0

FOX14: /dev/serial/by-id/usb-Ulowa_KC0JFQ_FOX_V2_2078-0-0109-if00-port0

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

Later boards have logic-level serial port.

3.5mm stereo audio connector.

All stations share a common USB serial cable (mine is called TACH).

TACH: /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.

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

Retain pads for USB UART (not populated).



Latest ICARC FOX Transmitters

System Commands

System configuration commands.

ONCE Execute 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)

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.

Latest ICARC FOX Transmitters

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.

BATV Battery Report (Voice).

BATC Battery Report (CW).

BATR Battery Report (operating time analysis).

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

Program Scheduling Commands.

MODS Load (or set) a schedule.

Latest ICARC FOX Transmitters

FRAM and FLASH Commands

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!.

Latest ICARC FOX Transmitters

TEST Commands

TEST Routines. Used for hardware test and debug.

HALT Halt processor `asm(" HALT");`.

STOP Stop processor `asm(" STOP");`.

RESET Reset Processor.

TEST Test routines.

STOP requires a hardware reset or power cycle!

Test routines are used to exercise various parts
of 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.

Latest ICARC FOX Transmitters

Talk Directory

Listing 1: TALK directory

| | | | |
|------|--------------|-------|----|
| esav | TALK=BATTI | 0 | 1 |
| esav | TALK=BATTV | 4224 | 2 |
| esav | TALK=REG5 | 8704 | 3 |
| esav | TALK=POINT | 13824 | 4 |
| esav | TALK=V.HZ | 15232 | 5 |
| esav | TALK=V.KHZ | 17664 | 6 |
| esav | TALK=V.MHZ | 20864 | 7 |
| esav | TALK=V.N0 | 24064 | 8 |
| esav | TALK=V.N1 | 26752 | 9 |
| esav | TALK=V.N2 | 28544 | 10 |
| esav | TALK=V.N3 | 30720 | 11 |
| esav | TALK=V.N4 | 32640 | 12 |
| esav | TALK=V.N5 | 34560 | 13 |
| esav | TALK=V.N6 | 36736 | 14 |
| esav | TALK=V.N7 | 38528 | 15 |
| esav | TALK=V.N8 | 40448 | 16 |
| esav | TALK=V.N9 | 41984 | 17 |
| esav | TALK=V.MAMP | 44416 | 18 |
| esav | TALK=V.VOLTS | 48128 | 19 |

Directory entries for the audio clips.

The beginning of the **TALK** directory.

Name and starting address in FLASH.



Latest ICARC FOX Transmitters

Talk Directory

| | | | |
|------|-------------|--------|----|
| esav | TALK=KC0JFQ | 51200 | 20 |
| esav | TALK=W0JV | 56960 | 21 |
| esav | TALK=FOX20 | 63104 | 22 |
| esav | TALK=FOX21 | 66048 | 23 |
| esav | TALK=FOX22 | 70272 | 24 |
| esav | TALK=FOX23 | 74624 | 25 |
| esav | TALK=FOX24 | 79872 | 26 |
| esav | TALK=FOX25 | 84608 | 27 |
| esav | TALK=FOX26 | 90752 | 28 |
| esav | TALK=FOX27 | 95744 | 29 |
| esav | TALK=FOX28 | 100352 | 30 |
| esav | TALK=FOX29 | 104832 | 31 |
| esav | TALK=FOX30 | 109824 | 32 |
| esav | TALK=FOX31 | 113664 | 33 |
| esav | TALK=FOX32 | 118400 | 34 |

More Directory entries...

The *NickName* clips for all Fox stations are present

Both club callsign and personal callsign are in the FLASH.



Latest ICARC FOX Transmitters

Talk Directory

| | | |
|------------------|--------|----|
| esav TALK=V.F144 | 123264 | 35 |
| esav TALK=V.F145 | 128000 | 36 |
| esav TALK=V.F200 | 134016 | 37 |
| esav TALK=V.F225 | 139776 | 38 |
| esav TALK=V.F250 | 145792 | 39 |
| esav TALK=V.F275 | 152192 | 40 |
| esav TALK=V.F300 | 158080 | 41 |
| esav TALK=V.F325 | 163968 | 42 |
| esav TALK=V.F350 | 168704 | 43 |
| esav TALK=V.F375 | 173312 | 44 |

Even more Directory entries...

Frequency clips for the ANN= message at startup

(ANN announces operating frequency!)

As you should expect, order is not critical

When a **TALK** command is executed,

we match names to get the start address in FLASH.

We then look for a **RIFF/WAVE** header to get the data size and sample rate.

SPI clock rate is set to 8x the audio sample rate.

This drives the target sample rate (1 sample every 8 SPI clocks).



Latest ICARC FOX Transmitters

INI File

Listing 2: "INI File"

| | |
|---------------------------|----|
| # | 38 |
| esav INI=TIME | 39 |
| esav INI=WAIT 0.5 | 40 |
| esav INI=TIME | 41 |
| esav INI=EPOC -5.0 | 42 |
| # | 43 |
| esav INI=NAME 'name' | 44 |
| esav INI=CALL 'call' | 45 |
| # | 46 |
| #esav INI=CONF BMON 12.5V | 47 |
| esav INI=CONF SI5351 | 48 |
| esav INI=CONF 8MA CLK0 | 49 |
| esav INI=CONF DRA818 | 50 |
| esav INI=FREQ 144.150 | 51 |

The INI= file; system initialization.

I use a single setup file to load FOX20..FOX32
'call', 'name', and 'run' substituted from the
loader utility command line

This file runs when one or no jumpers installed

We start all stations on the same announce frequency,

We will change to the operating frequency a bit later...



Latest ICARC FOX Transmitters

INI schedules

Listing 3: "INI schedules"

| | |
|--------------------------------------|----|
| # | 52 |
| # Set schedules , leaving ONLY | 53 |
| # | 54 |
| REM- 0123456789012345678901234567890 | 55 |
| esav INI=MODS S0 'run ' | 56 |
| esav INI=MODS S1 30 0 | 57 |
| esav INI=MODS S5 60 0 | 58 |
| esav INI=MODS S6 10 0 | 59 |
| esav INI=MODS S7 300 0 | 60 |
| esav INI=MODS S9 300 0 | 61 |
| # | 62 |

The INI= file; define schedules.

Finishing up the INI= file.

Define up to ten schedules.

We take the primary schedule **S0=** from the command line.

Latest ICARC FOX Transmitters

ANN File

Listing 4: "ANN File"

| | |
|---------------------------------|----|
| esav ANN=REM- fox_ann_V2023.fox | 73 |
| esav ANN=TONE 1.0 | 74 |
| esav ANN=CWPM 30,-1,-1,-1,-1 | 75 |
| esav ANN=BEGN | 76 |
| esav ANN=TALK <CALL> | 77 |
| esav ANN=TALK <NAME> | 78 |
| esav ANN=WAIT 1.0 | 79 |
| esav ANN=BATV V | 80 |
| esav ANN=BATV I | 81 |
| esav ANN=WAIT 0.3 | 82 |
| esav ANN=TALK 'freqM' | 83 |
| esav ANN=TALK 'freqK' | 84 |
| esav ANN=TONE 1.0 | 85 |
| esav ANN=CWPM 30,-1,-1,-1,-1 | 86 |
| esav ANN=DONE | 87 |
| esav ANN=FREQ 'freq' | 88 |
| esav ANN=STAT | 89 |
| esav ANN=RUN0 S0 | 90 |

The ANN= file; system announce message.

Runs **after** INI= when **no** jumpers are installed. Tell 'em we're alive!

More parameter substitution from command line: '**freq..**'

Parameter substitution from INI= setup: <CALL><NAME>

Frequency change to operating frequency in line 88

Schedule **S0** enabled in the last command in line 90



Latest ICARC FOX Transmitters

TEST File

Listing 5: "TEST File"

```
esav TEST=CVPM 35,-1,-1,-1,-1
esav TEST=TIME
#
```

64
65
66

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

Did you notice

that we changed frequency!

In the **INI=** command and again in the **ANN=** command?

We can do that! (even in the middle of a message)



Latest ICARC FOX Transmitters

MAS File

Listing 6: "MAS File"

```
esav MAS=CWPM 35,-1,-1,-1,-1  
esav MAS=STAT
```

67
68

The MAS= file; alternate system test.

Runs **after** INI= when MAS jumper installed

You are free to do whatever you want here too!

ANN= message is **not** sent

Latest ICARC FOX Transmitters

System Recovery

System Recovery (error recovery)

Install both **MAS** and **TEST** jumpers

Nothing is read from either the **program** or **waveform** memory.

Use to recover from fouled up programs

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 recovery feature.

Latest ICARC FOX Transmitters

S0 File

Listing 7: "S0 File"

| | | |
|------|---|-----|
| # | | 91 |
| # | Operating Schedules | 92 |
| # | The power draw of the DRA818/SA818 may cause the | 93 |
| # | "BATC" form of the report to take too long, so we | 94 |
| # | ALWAYS use the vocal report. | 95 |
| # | ALSO We're making use of the <CALL> and <NAME> substitution | 96 |
| # | inside the fox transmitter!!! | 97 |
| # | | 98 |
| esav | S0=BATR | 99 |
| esav | S0=CONF -AM | 100 |
| esav | S0=TONE 1.0 | 101 |
| esav | S0=CWPM 30,-1,-1,-1,-1 | 102 |
| esav | S0=BEGN | 103 |
| esav | S0=TALK <CALL> | 104 |
| esav | S0=TALK <NAME> | 105 |
| esav | S0=WAIT 0.5 | 106 |

The S0= file; operating program for schedule 0.

Set audio tone and CW chipping parameters to:

1KHz and 30WPM (fast signon message)

BEGN enables carrier and sends signon message

TALK verbalizes our callsign and name

WAIT for half a second, unmodulated carrier



Latest ICARC FOX Transmitters

S0 File

| | |
|---|-----|
| # | 110 |
| # Fill time so they have a chance of finding me | 111 |
| # | 112 |
| esav S0=TONE 1.5 | 113 |
| esav S0=CWPM 25,-1,-1,-1,-1 | 114 |
| esav S0=WAIT 0.15 | 115 |
| #esav S0=CODE hi hi hi | 116 |
| esav S0=BATC EV 7.2 | 117 |
| esav S0=WAIT 0.5 | 118 |
| esav S0=CODE IOWA CITY | 119 |
| esav S0=CODE AMATEUR RADIO | 120 |
| esav S0=CODE CLUB FOXHUNT | 121 |
| esav S0=CODE F W KENT PARK | 122 |

Change what we sound like, 1.5KHz and 25WPM

WAIT for 150 milli-seconds, more unmodulated carrier

Line 117, send *encoded* battery report (count the dah/dits)

When above 7.2V, send "**HI HI HI BATV ttttttt eee**"

When below 7.2V, send "**SOS SOS BATV ttttttt e**"

Lines 119-122; send our message in code



Latest ICARC FOX Transmitters

S0 File

| | |
|--|-----|
| # | 124 |
| # | 125 |
| # Prepare (kinda...) for Signoff | 126 |
| # these extr REM- commands can be deleted (EZER) | 127 |
| # and replaced with more CODE commands to adjust time... | 128 |
| esav S0=REM- | 129 |
| esav S0=REM- | 130 |
| esav S0=REM- | 131 |
| # | 132 |
| # Signoff | 133 |
| # | 134 |
| esav S0=BATR | 135 |
| esav S0=TONE 1.0 | 136 |
| esav S0=CWPM 30,-1,-1,-1,-1 | 137 |
| esav S0=DONE | 138 |

lines 129-131 can be replaced to fix *problems*.

lines 136-138 return the code generator to the same

state it was in for the **BEGN** message

DONE sends our callsign in the signoff message

and removes carrier (ID at end-of-message).



Latest ICARC FOX Transmitters

Help Pages

HELP list from current software:

Starting with the **System** commands.

Listing 8: fox27.help_1

| sts01 ,00* | TEST HELP ** | TEST HELP ** | TEST HELP ** | | | 1 |
|------------|--------------|--------------|--------------|-----------|--|----|
| sts01 ,00* | Idx | MNE | Class | Arguments | Command Function | 2 |
| sts01 ,01* | 1 | HELP | SYS | | Help Menu and Items | 3 |
| sts01 ,02* | 2 | HELP | SYS | <string> | matching help items | 4 |
| sts01 ,03* | 3 | ONCE | SYS | <name> | Test run the named sequence | 5 |
| sts01 ,04* | 4 | REM- | SYS | | Remark, (side-effect: stops → schedules) | 6 |
| sts01 ,05* | 5 | RUN0 | SYS | | RUN ALL Schedules | 7 |
| sts01 ,06* | 6 | RUN0 | SYS | <name> | RUN Specific Schedule | 8 |
| sts01 ,07* | 7 | STAR | SYS | <time> | Start running schedules at → specified time | 9 |
| sts01 ,08* | 8 | IDLE | SYS | | STOP ALL Schedules | 10 |
| sts01 ,09* | 9 | STAT | SYS | <flag> | System Status, (l)ident scan | 11 |

ONCE Sequence testing feature

RUN0

IDLE

STAT Verbose dump of system status.

Adding the **I** dumps the version/date strings
from all the software modules.



Latest ICARC FOX Transmitters

Help Pages

More **System** commands.

Listing 9: fox27.help_12

| | | | | | | |
|-----------|----|------|-----|------------------------|------------------------|----|
| sts01,10* | 10 | CONF | SYS | <keywords> | Hardware Configuration | 12 |
| sts01,11* | 11 | TOYC | SYS | <res> (250 2K 4K NONE) | Hi chg rte DS1672 bat | 13 |
| sts01,12* | 12 | TIME | SYS | <time value> | Set Time (set DS1672) | 14 |
| sts01,13* | 13 | D525 | SYS | <sub-command> | ICS525 debug routines | 15 |

CONF defines the hardware we are running on.

TIME with argument sets the DS1672 TOY clock.

Latest ICARC FOX Transmitters

Help Pages

Next in line are the **Setup** commands.

Listing 10: fox27.help_16

| | | | | | |
|------------|----|------------|---------|---------------------------------|----|
| sts01 ,14* | 14 | TIME SETUP | | Time from DS1672 to System (NO | 16 |
| | | | | Argument!) | |
| sts01 ,15* | 15 | EPOC SETUP | <hours> | Epoch offset (i.e. time zone) | 17 |
| sts01 ,16* | 16 | CALL SETUP | <call> | FCC Assigned Callsign | 18 |
| sts01 ,17* | 17 | NAME SETUP | <nick> | Local Nickname | 19 |
| sts01 ,18* | 18 | NICK SETUP | <nick> | alias for "NAME", but don't use | 20 |
| | | | | it! | |

TIME with no argument sets system time from TOY clock.

EPOC sets a timezone offset (TOY is kept in UT, not local) (negative west of Z).

CALL save the FCC assigned callsign for later use.

AA0AAA/5 is OK, but must have audio file for each station!

NAME & NICK save a tactical callsign (nickname) for later use.

Use this to distinguish stations!



Latest ICARC FOX Transmitters

Help Pages

Followed by **Program** commands.

Listing 11: fox27.help_21

| | | | | | |
|------------|----|----------|----------------------------|--|----|
| sts01 ,19* | 19 | TONE PGM | <freq> | Audio Tone (in KHz) | 21 |
| sts01 ,20* | 20 | CWPM PGM | <wpm gap1 gap2 gap3> | CW Chipping Rate | 22 |
| sts01 ,21* | 21 | FREQ PGM | <freq> | Frequency (in MHz) | 23 |
| sts01 ,22* | 22 | 5351 PGM | <key>,<value>,<value> ,... | SI5351 setup group | 24 |
| sts01 ,23* | 23 | BEGN PGM | | Key TX and Send Callsign (CW) | 25 |
| sts01 ,24* | 24 | CODE PGM | <message> | Send Message (CW) up to 22 char | 26 |
| sts01 ,25* | 25 | TALK PGM | <file -name> | Play Voiced Message (EDMP TALK) | 27 |
| sts01 ,26* | 26 | WAIT PGM | <secon.ds> | Wait (simple delay) | 28 |
| sts01 ,27* | 27 | CHRP PGM | <tone> <dur> <cnt> | Send carrier chirp | 29 |
| sts01 ,28* | 28 | DONE PGM | | Send Callsign (CW) , SK (CW) , → and unkey TX | 30 |

TONE Audio frequency for code generator.

TONE 0.0 will disable tone generator for unmodulated carrier.

CWPM Code rate and timings.

FREQ Nominal Carrier Center Frequency.

BEGN Turn on RF and send signon message.

CODE Send text in code.

TALK Send audio file.

WAIT Simple delay.

DONE Turn off RF and send signoff message.



Latest ICARC FOX Transmitters

Help Pages

Add some battery status reports.

Listing 12: fox27.help_31

| | | | | |
|------------|-------------|--|-------------------------------|----|
| sts01 ,29* | 29 BATC PGM | <mod>,<key>,<setpoint> | Transmit Code Battery Report | 31 |
| sts01 ,30* | 30 BATV PGM | <mod>,<key> | Transmit Vocal Battery Report | 32 |
| | | mod: "E" encode (not CW) for BATC | | 33 |
| | | mod: "B" battery reading taken before BEGN | | 34 |
| | | mod: "A" battery reading taken after BEGN | | 35 |
| | | key: "V" battery voltage , | | 36 |
| | | "I" battery current , | | 37 |
| | | "R" 5V rail | | 38 |

BATC Battery Report, CODE.

Adding **E** to command changes from using proper code
to using 'T' and 'E' characters to encode the voltage or current.

BATV Battery Report, Voice.

Reporting audio clips must be loaded in FLASH for this to function.

BATR Battery Report, Text.

Report for analyzing system operating time.
(Does not produce any RF report).



Latest ICARC FOX Transmitters

Help Pages

A few **Scheduling** commands.

Listing 13: fox27.help_39

| | | | | | | | | |
|-----------|----|------|-------|-----------------------|---------|----------|-------|----|
| sts01,31* | 31 | MODS | SCHED | <Sname period offset> | Modulus | Schedule | Set | 39 |
| sts01,32* | 32 | MODC | SCHED | <Sname=> | Modulus | Schedule | Clear | 40 |

MODS Modular Schedule.

Schedule name: S0 through S9 (up to 10 of them).

Period is the cycle time, expressed in seconds.

Offset is offset into the scheduling Period.

Always less than the Period.

Latest ICARC FOX Transmitters

Help Pages

A single **Directory** command.

Listing 14: fox27.help_39

| | | | |
|------------|---|---------------------|----|
| sts01 ,33* | 33 TALK DIRECTORY esav TALK=name, Strt , Len , rate | (appears in FRAM as | 41 |
| | → the TALK= file) | | |
| | Waveform Directory Entry | | 42 |
| | rate keys: 4K 5K 8K 10K 16K | | 43 |

TALK= Audio File Directory.

This is the directory of audio clips stored in FLASH.

You will see one for each fragment of speech.

Number of them limited by space in FRAM to store the directory
and space in FLASH to store data samples.

Data rate can be one of three rates (mind bandwidth limitations!)

4KHz, 5KHz, 8KHz, 10KHz, 16KHz

RIFF/WAVE files have length and rate information in the header.

So they only require the starting address in the directory entry.

This **TALK** key is context sensitive. This keyword defines a directory entry and a command.

Latest ICARC FOX Transmitters

Help Pages

Additional Frequency Parameters.

Listing 15: fox27.help_44

| | | | | | | | |
|-----------|----|------|-----------|------|------------------------------|---------------------|----|
| sts01,34* | 34 | freq | DIRECTORY | esav | 144.150=13BF,70E40,F4240,100 | (appears in FRAM as | 44 |
| | | | | | ↪ frequency record) | | |
| | | | | | Register Parameters are | | 45 |

Frequency setup for SI5351/ICS525

The frequency entry is a text record (so think string match)

The *freq* is the frequency in MHz (3 decimal digits)

For the SI5351 we have four values to be sent to the synthesizer.

For the ICS525 we have the 3 register values.

Used to fill in the external frequency table.

External utility to generate a table of frequencies

Latest ICARC FOX Transmitters

Help Pages

FLASH and FRAM Control COmmands

First of them are the **FRAM** commands.

Listing 16: fox27.help_46

| | | | | | |
|------------|----|-----------|-------------------|---|----|
| sts01 ,35* | 35 | ESAV FRAM | NAM=<text> | Save named record in next free location | 44 |
| sts01 ,36* | 36 | EDMP FRAM | "match string" | Dump active records | 45 |
| sts01 ,37* | 37 | EDID FRAM | | Flash JEDEC-ID table dump (PROG & WAVE) | 46 |
| sts01 ,38* | 38 | ERAS FRAM | <number> or "DEV" | Rewrite <record> to REM- (DEV, QTR, HALF) | 47 |
| sts01 ,39* | 39 | EZER FRAM | <number> | Erase <record> to ZERO | 48 |
| sts01 ,40* | 40 | ETAB FRAM | | Dump JEDEC-ID device table | 49 |

ESAV Save record.

EDMP Dump records. The *match string* will report only matching strings.

EDID Read and report of the ID bytes in the FRAM and FLASH device.

ERAS Rewrite the numbered record to be a *REM-* command.

EZER Rewrite the numbered record to the empty value.

Records after the **zeroed** record are lost until something is written to this replace the **zeroed** record .

ETAB List of recognized memory devices

Latest ICARC FOX Transmitters

Help Pages

And then the small number of **FLASH** commands.

Listing 17: fox27.help_52

| | | | | | |
|------------|----|----------------|---------------------------------|--------------------------------|----|
| sts01 ,41* | 41 | HERA FLASH | ALL | Hex erase (entire WAVE device) | 50 |
| sts01 ,42* | 42 | HDMP FLASH | <len-32B-lines <hex-start <*>>> | Hex dump (WAVE device) | 51 |
| sts01 ,43* | 43 | H56K FLASH | | Fast terminql bit rate | 52 |
| sts01 ,44* | 44 | :hex FLASH-HEX | :llaaaattdddddcc | Intel HEX loader (WAVE device) | 53 |

The FLASH is used to store audio data (i.e. no commands stored here).

These are 8-bit mono *reduced sample rate* WAV files (4KHz to 16KHz).

The *TALK*= entries point to a starting point for each audio fragment.

Length and sample rate in RIFF/WAVE header, otherwise in directory entry (see sheet 19)

Start pointer points at RIFF/WAVE header, otherwise to the start of the waveform data

Only write path to FLASH is using an Intel HEX file to store audio data.

Intel HEX file may have embedded spaces.

character text may appear following the checksum, but that may cause buffer overflow.

Intel HEX record length limited to 32 data points.

There is very little SRAM in the zNEO to buffer longer records

H56K is now H115 to buzz-up the transfer rate (to 115,200)!

Nominal rate 57,600 b/S. Input channel buffered (ISR).

Write timing open-loop, zNEO allows 10mS to 20mS before status report.

HERA can be sloooooooooooooooooooooooooooooW

Sends device erase to FLASH device, reports expected erase time but doesn't wait!

Some flash devices take hundreds of seconds to erase!

Flash device appears dead or broken until erase function completes.



Latest ICARC FOX Transmitters

Help Pages

Finally, some **Test** commands.

Listing 18: fox27.help_56

| | | | | |
|------------|----|-----------|-------------------------|----|
| sts01 ,45* | 45 | HALT TEST | Halt Processor | 54 |
| sts01 ,46* | 46 | STOP TEST | Stop Processor | 55 |
| sts01 ,47* | 47 | REST TEST | Reset System | 56 |
| sts01 ,48* | 48 | TEST TEST | Hardware Test Subsystem | 57 |

HALT Executes the zNEO HALT instruction. Next interrupt releases the halt state.

STOP Executes the zNEO STOP instruction. Causes the system to hang.

REST Sets the *USER_RST* bit into the zNEO *RSTSCR* register.

Causes the zNEO to reset (like a power cycle or mashing the reset button).

TEST This is a gateway command to a set of hardware debugging routines.

TEST HLP Produces a list of test routines.

Used for hardware test and debug.