

EncoderCAT

1.12

G3VPX

Table of contents

EncoderCAT - project overview	2
EncoderCAT - introduction	3
EncoderCAT - some images	5
EncoderCAT - radio compatibility	8
EncoderCAT PCB	10
EncoderCAT - initial setup	12
EncoderCAT - processor programming	12
EncoderCAT - FT245RL USB device	13
xCAT - configuration introduction	18
xCAT - introduction	18
Configuration - Yaesu, Kenwood	20
xCAT - Yaesu, Kenwood	20
xCAT example - Yaesu	21
Command masks example	22
xCAT - Yaesu MENU command (EX)	24
Configuration - Icom	26
xCAT - Icom encoder cfg.	26
xCAT - CI-V option selectors	28
xCAT - examples Icom	30
EncoderCAT - scope view of CIV data	33
Configuration - OLED display	34
xCAT - OLED display option	34
xCAT - OLED display lookups	35
EncoderCAT and xCAT code	36
Source code	36
JTAG programmer	37
Construction	39
EncoderCat schematic	39
EncoderCAT bill of materials	40
Soldering SMD devices	41
Construction notes	43
Further development	44
encodercat@groups.io	44
piWebCAT	45
Miscellaneous notes etc	48
xCAT / EncoderCAT notes	48
Software / schematic etc	49
EncoderCAT - Downloads	49

EncoderCAT - rotary encoder control by CAT interface

EncoderCAT provides CAT control by rotary encoders

Most modern amateur radio transceivers have a CAT interface for external control by PC based software. The user manuals give details of the control code sequences for a wide range of controllable settings. I shall refer to these settings as rig **parameters**.

There exists a small number of PC based CAT control applications, some of which offer comprehensive control for a range of different radios. Some are free. Some cost money.

Rig control parameters such as RF power, noise reduction level, vox gain, audio gain etc are typically managed on a PC screen by slider controls.

EncoderCAT provides control for such parameters using low cost electro-mechanical rotary encoders. These are devices which are typically used as volume controls in car radios. They have remained available for purchase on eBay for the past 15 years or longer. The price can be as low as £20 for ten encoders.

EncoderCAT was initially developed using a Yaesu FTdx101D and then extended for Icom radios.

The **FTdx101D** has dedicated rotary controls for the most commonly used functions. In addition, a single 'Multifunction' control can be assigned to one of a range of other rig parameters. These parameters include DNR level and RF Power for which separate controls are desirable. EncoderCAT provides such separate controls.

Configuration

EncoderCAT is configured for a particular radio using the PC-based **xCAT** application.

- xCAT connects to the EncoderCAT PCB's micro-USB connector (which also supplies 5v power)
- The configuration is sent via USB to EncoderCAT and stored in the EncoderCAT processor's EEPROM.
- Each encoder can be configured to one of the radio's control parameters, eg: RF power.
- xCAT does not contain the large lists of parameter settings for a range of radios. Instead, the operator uses the radio's user manual to create a control parameter dataset for the chosen actions of EncoderCAT's eight rotary encoders. This dataset also includes baudrate and other serial transmission parameters and a title for the dataset.
- The dataset can be copied to and from EncoderCAT and to and from a PC disc file.

Note that there is lot more configuration information in this document for Icom CI-V configuration - because it is more complex than Yaesu. Kenwood, Elecraft.

Operation

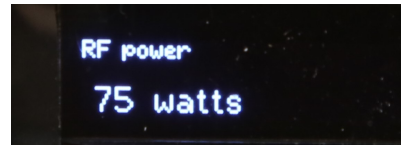
When an encoder is rotated by one notch:

- The processor identifies the encoder and the direction of movement (ie: up or down)
- The processor looks up the CAT command details for the assigned encoder parameter.
- An CAT command is sent to the radio to request a read of the current parameter value.
- The radio sends back a message containing the parameter value.
- EncoderCAT increments or decrements the value by the specified step.
- The modified value is transmitted back to the radio.

At 38400 baud, this all happens in 5ms, well before the next encoder notch movement occurs

OLED display panel

EncoderCAT has an optional small OLED display.
The firmware is designed to drive an I2C connected display which conforms to SSD1306 driver specification.



The display identifies the operated encoder and stays with that encoder until another is operated.

For each encoder, the user can configure for the OLED display (with xCAT) :

- **Description**, eg: "RF power" as above
- **Units** (eg: watts, Hz, kw, dB),
- **Scaling** of the CAT value ie: (value x multiplier) / divisor
- Optional **decimal point** eg: change 75 watts to 0.075 kw
- **Lookup table**, eg: value 1 -> 300Hz 2 -> 600Hz 3 -> 1000Hz

My prototype uses a small 128 x 32 pixel monochrome display.

These are available from a number of eBay suppliers and cost approximately £4.00.

The display unit does not contain font data.

I have two proportional font character sets in the firmware:

- one 16 bits high for the data and one 8 bits high for the description.

EncoderCAT cost

EncoderCAT components cost about £30 from Farnell electronics

(The cost may be more than this due to some minimum quantity constraints and post and packing)

G3VPX can supply the PCB for £4.

The main cost is a JTAG programmer.

The cheapest Atmel JTAG programmer is the Atmel-ICE supplied as PCB-only for £41 + VAT,

This is a fast programmer with built in debugging capability to work with Atmel's (free) Studio 7 system.

You need to make up a cable to use this device. The connectors are in the EncoderCAT bill of materials.

G3VPX can supply preprogrammed Atxmega192A3 processor.

However, this project is likely to evolve / improve with feedback from users

- you would then need a programmer to install the program updates.

See: [JTAG programmer](#)

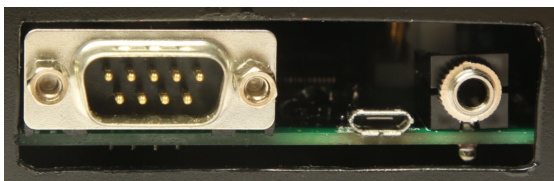
EncoderCAT - images of G3VPX prototype unit

My EncoderCAT unit is housed in a box constructed from 1.6 oz double-sided copper clad PCB material. I find this to be a versatile and easy to use construction material.

- Easy to cut and shape
- Panels are soldered together and quite complex 3 dimensional arrangements are possible.
 - see <http://www.g3vpX.net/picastar/box.htm>
- Mounting brackets are simply soldered to the box sides
 - see image - brass angle lid mounts (heat needed - preferably before you paint the box!)

Note: If you make a full length solder fillet between two otherwise unsupported panels, then the joint will become angled as the solder cools.

- Therefore the whole box should initially be assembled with small solder tacks.
- Then you can then fully fillet-solder the joints without risk of distortion.



Rear connectors

RS232 - male - exactly as on a PC.

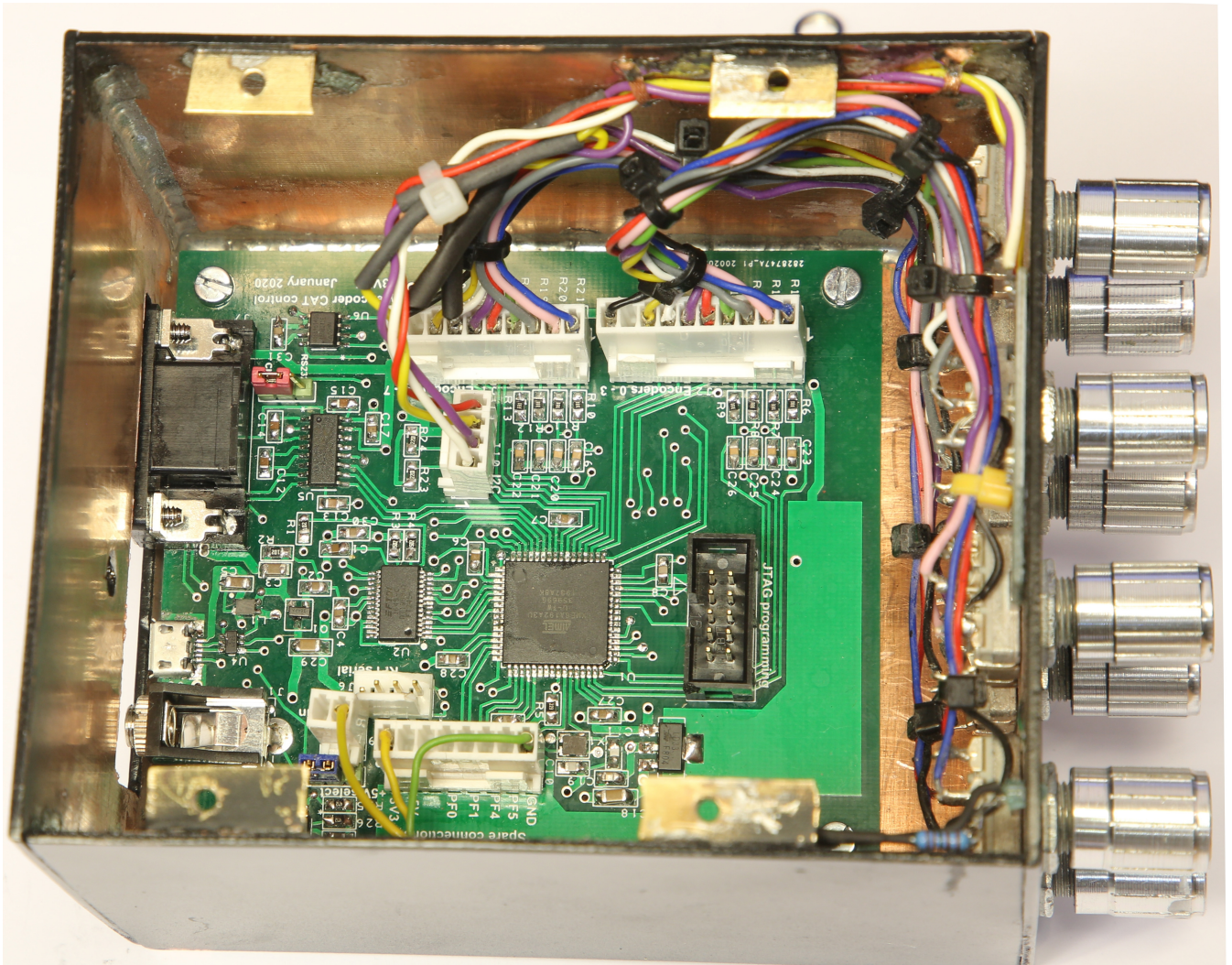
Micro USB for 5v power and xCAT configuration.

3.5 mm mono jack socket for Icom CI-V

In this development unit, I made a hole large enough to easily route in the temporary JTAG programming cable.

EncoderCAT - prototype - internal views - image

The PCB mounting points connect the ground plane to the box via 3mm threaded brass pillars. A 3mm front panel power indicator LED and series resistor are driven by the 'spare' connector.



Rotary encoders and OLED display - image

Below is a close up of the OLED display panel and some encoders.
These encoders are Song volume controls ... but there are alternative suitable similar devices.

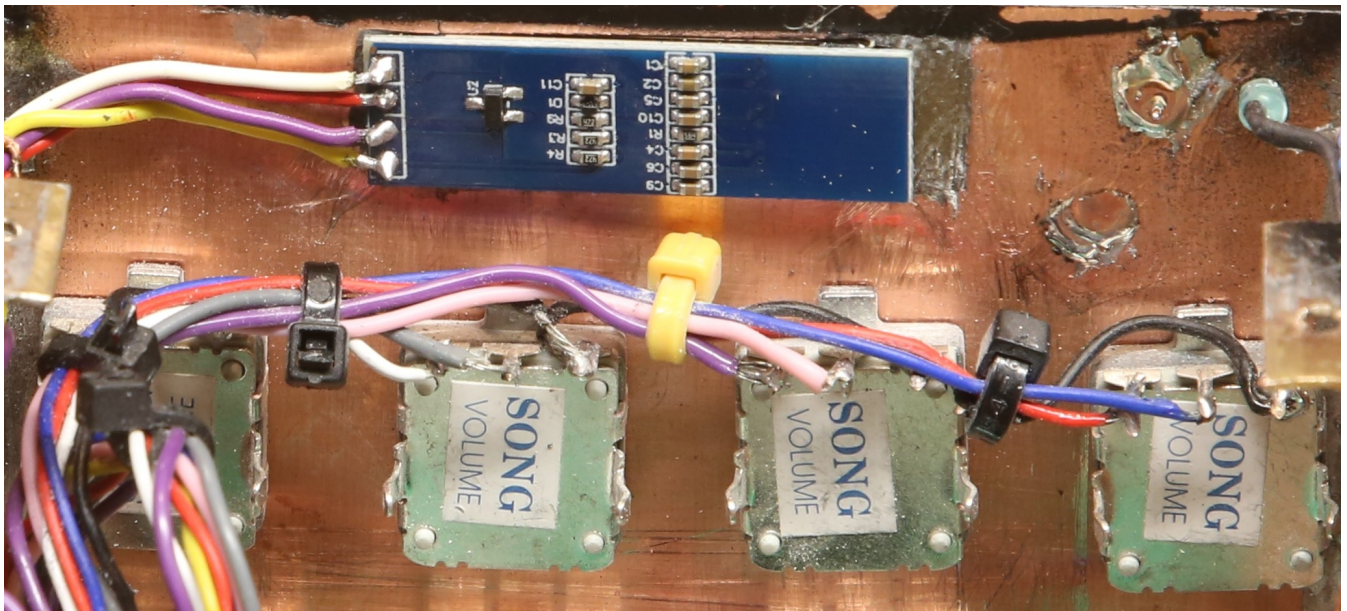
The **encoders** have common terminal (in this case: on the right) which is connected to ground.
I have referred to the other two encoder connections as **phase** and **quad**.
When the encoder is in a rest state (ie: sitting in a mechanical notch) then phase and quad are open circuit.
If the encoder is moved right or left, phase and quad are briefly grounded .. but not simultaneously.
Their contacts are offset from each other such that:

- Grounding phase triggers a processor interrupt.
- The firmware interrupt service routine checks the phase lines to identify the triggering encoder.
- It then examines the associated quad line to determine the direction of rotation.

It is possible to leave an encoder stuck between notches. The firmware has code to ensure that this does not inhibit the operation of a different encoder.

The **OLED display** is glued to the rear of the front panel with impact adhesive.
It has a 4 - wire I2C cable from J10 on the PCB. (0v, 3v3, SDA and SCL)

The copper under the four connections was filed away to avoid short circuiting.



EncoderCAT - radio compatibility

PC based CAT systems hold the control codes for a wide range of radios.
EncoderCAT does not do this.

- It only needs to store the control sequences for eight parameters (controlled by the eight encoders)
- The control code formats are entered by the user, via USB using my PC based application xCAT.

Encoder actions occur when the operator clicks an encoder up or down.

Most encoder actions will be **read - modify - write**, eg, the **RF power** action is:

- *Read power from rig. Increment or decrement the value. Write the modified RF power back to rig.*

EncoderCAT has been developed using a new **Yaesu FTdx101D** transceiver and an **Icom IC7000**.

Yaesu and Kenwood continue to provide a DB9, 9 pin **RS232 CAT interface** on their radios.

I understand that **Elecraft** use RS232 via a **RJ45** connector.

I read that the adapter cable used with a P3 panadapter can be used for PC connection.

These RS232 connected rigs all use CAT commands in the form of **ASCII character sequences**.

These are easy to understand and manage. For example, Yaesu RF power control uses **PC**.

eg: read power from rig is: **PC**; answer from rig is: **PC085**; (85w) set rig power is: **PC090**; (90w) .

The corresponding user-entered configuration **masks** are:

read mask = **PC**; answer mask = **PChtu**; set mask = **PChtu**; (**h** = hundreds, **t** = tens, **u** = units)

Icom transceivers' CAT interface is their **CI-V** system.

This is a single bidirectional wire with open drain drive and pullup resistors.

The supply voltage appears to be 3v3. (from test meter on IC7000 CI-V line and also from circuit diagram)

The connection is a **mono 3.5mm jack socket**.

The EncoderCAT PCB has a 3.5mm jack socket for direct connection to a rig.

Icom's CI-V adapters are not needed.

(Icom's RS232 adapter is discontinued The currently available adapter is USB)

There are a range of RS232 to CI-V adapters (Icom and also homebrew on the internet)

These connect RS232 Rx and Tx lines (through appropriate buffering) to the single CI-V line.

This results in Tx data returning down the Rx line during transmission.

EncoderCAT uses an NXP PCA9600 bidirectional buffer device which avoids this feature by having the input-low threshold voltage lower than then output pulldown level.

Icom CI-V coding is a little more complex than a simple ASCII system.

It uses CAT commands in the form of sequences of bytes specified in hexadecimal:

- **Hexadecimal** command codes, eg: 1A hex
- **Decimal data values** represented in **Binary Coded Decimal format** (BCD)

Hexadecimal bytes in this document are represented with the prefix 0x. (as used C and PHP languages)

eg: 0x2A is hexadecimal 2A = $2 \times 16 + 10 =$ decimal 42.

BCD format represents two decimal digits as a single hexadecimal byte.

eg: decimal 37 is 0x37 ... easy to read (but on hex to dec conversion = $3 \times 16 + 7 =$ 55 decimal)

EncoderCAT software does the decimal to BCD conversion for you.

This allows you to enter command 050014 as such in decimal

(EncoderCAT then transmits this as three bytes: 0x50 0x00 0x14)

Icom's command system is complicated by what would appear to be historical / evolutionary features.

The range of formats includes:

- single hex command byte.
- hex command byte + hex sub-command.
- hex command byte + hex sub-command byte + bcd data byte. (read only - but read/write in later rigs).
- hex command byte + two bcd data bytes (read/write). The bcd bytes referred to as sub-command.
- hex command byte + three bcd data bytes. (read/write). The bcd bytes referred to as sub-command.

Most read - modify -- write commands would use the last option.

EncoderCAT provides a CI-V user configuration system that supports these formats.

I got all this working on my IC7000 and then downloaded the manual for a modern Icom rig.

The hex + 3 BCD commands have a completely different numbering allocation.

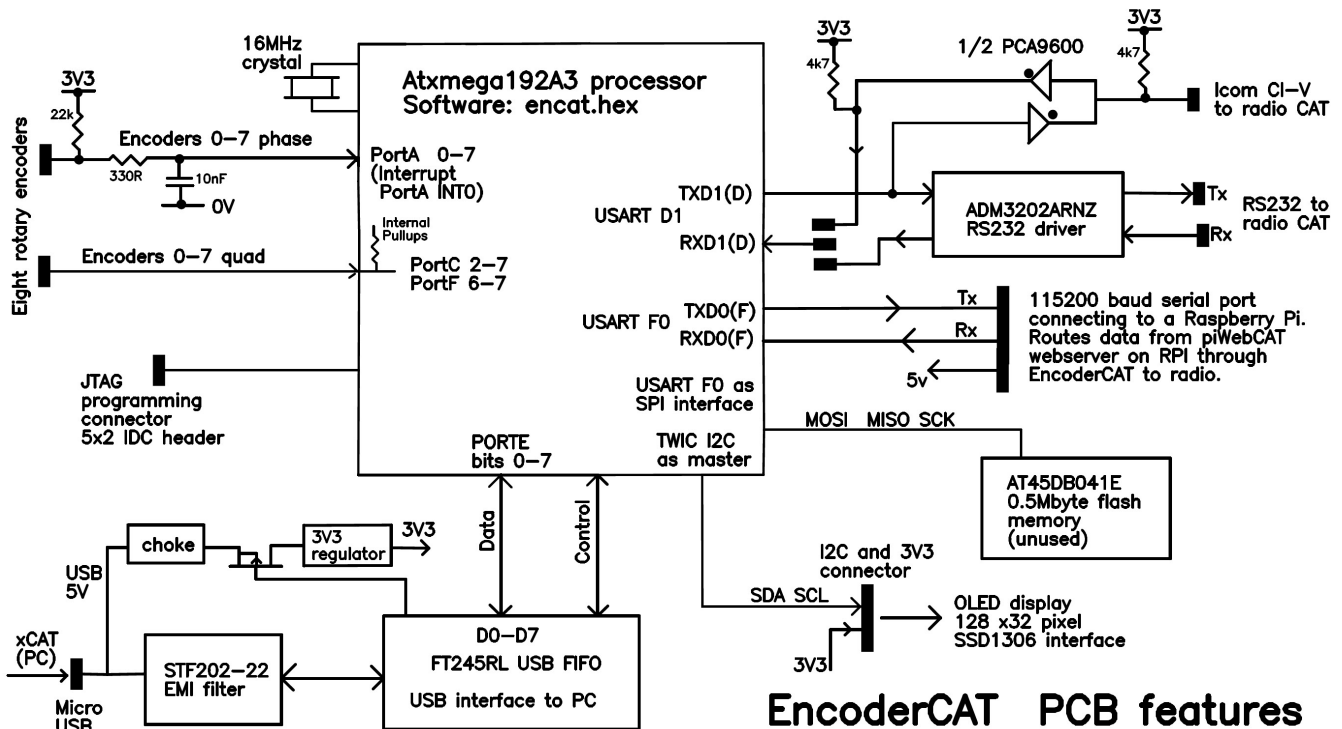
... *This doesn't matter*

- *the user of EncoderCAT configures the eight encoders using the CAT manual for his/her rig.*

EncoderCAT PCB

The PCB has the following features:

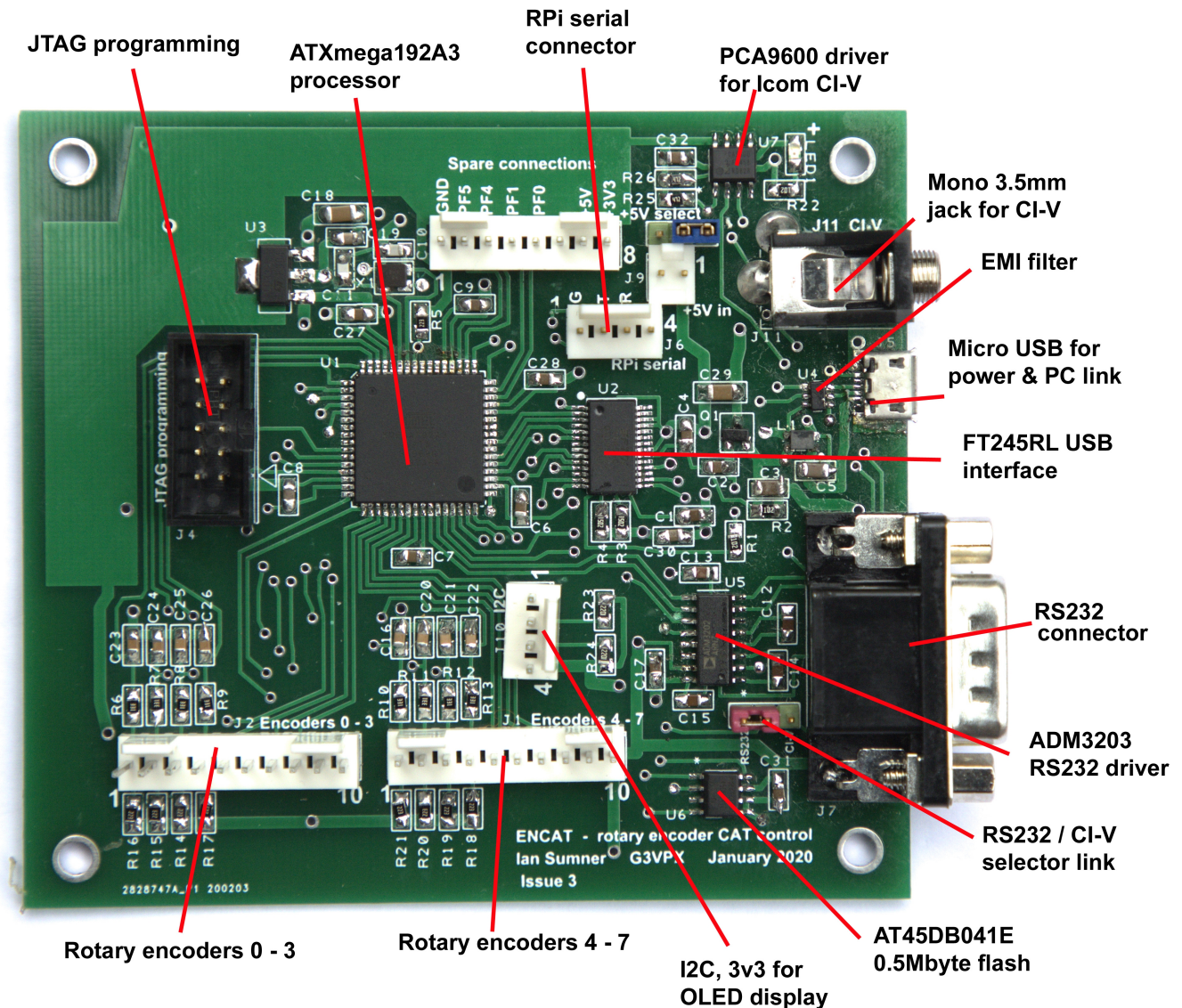
- PCB size is 8.5 x 9.5 cm. (My discarded issue 1 and 2 PCBs make good beer mats!)
- Atmel Atmega192A3 processor running at 32MHz. (16MHz crystal + PLL doubler enabled)
- RS232 (+5v / -5v) CAT interface drive using an ADM3203 device.
- Icom CI-V CAT interface - single wire bidirectional, wire-ored. Driven by a NXP PCA9600 device.
- Input connectors for eight rotary encoders (low resolution .. eg: car radio volume controllers)
- Power supply - 5v - via micro USB or dedicated power connector.
- Configuration via USB using xCAT PC application. (FTDI FT245RL device)
- Configuration stored in processor's 2kbyte EEPROM memory.
- 115200 baud serial connector to a Raspberry Pi for integration with piWebCAT.
- I2C connector (with 3V3 power) - used for 128x32 OLED display.
- Adesto AT45DB041 0.5 Mbyte flash memory device for future expansion or alternative PCB usage,
- JTAG programming connector.
- **The PCB could be used for other projects** - eg: controlling a home brew transceiver.
 - It has USB, fast serial, RS232 and I2C connectors plus sixteen bit digital I/O.
 - It has a 0.5Mbyte flash chip - which is unused in EncoderCAT.
 - The number of digital I/O lines could be increased using the I2C interface eg: using a PCA9555 I2C to 16 line decoder, see http://www.homebrew-radios.net/trxavr_picastar/I2C/PCA9555_board.htm



EncoderCAT PCB features

The PCB schematic is available as a downloadable PDF at: <http://encat.g3vpx.net/files/encat-schematic.pdf>

Note that the AT45DB041E device is not used and so can be omitted (They cost £1.05 + VAT from Farnell !).



The PCB shown above was designed using **Easy-PC** software and commercially manufactured.

Using Easy-PC:

Every component has a schematic symbol and PCB footprint (copper, drilling, top silk etc).

A component record then combines the schematic and PCB records and maps the pin numbering between them.

The schematic diagram is built using the schematic symbols.

'Forward design changes' then transfers the schematic design to the PCB layout placing all components in a 'component bin'.

Then, as components are dragged from bin to layout, connections are shown as straight, fine dotted lines.

These are then manually replaced by PCB tracks.

The software prevents the operator from making connections which are illegal according the schematic.

A final integrity check warns of any electrical discrepancies between layout and schematic.

At the time of writing, G3VPX had 50 bare PCBs available for sale at low cost.

.. more can be manufactured at 2-3 weeks notice.

Home fabrication would be possible using double sided construction with wired-through vias.

This would however require a totally new layout, mainly because of the wired vias:

- Wired via cannot be located underneath surface mounted ICs.
- Wired via on home brew boards need to be larger to cope with alignment inaccuracies between the two sides of the board.

EncoderCAT - Atmel Studio 7 - programming encat.hex

EncoderCAT code is written in the C programming language, using **Atmel Studio 7**.
Atmel is now owned by MicroChip and so the download of Studio 7 is from a Microchip site:

<https://www.microchip.com/mplab/avr-support/atmel-studio-7>

You do not need to compile enCAT code to program EncoderCAT.
You can simply program with the pre-built **encat.hex**.

xCatSetup.exe needs to be run *as administrator* (a right mouse click option)
xCatSetup.exe creates folder: Program Files (x86)\hbradios\xcat
A number of files and folders are placed in this folder including xcat.exe, xcat.chm, EncoderCAT.pdf.

A subfolder: encatcode is created. (ie: C:\Program Files (x86)\hbradios\xcat\encatcode)
In this folder is **encat.hex** and zipped EncoderCAT and xCAT source files.

In order to program the EncoderCAT processor with **encat.hex**:

- Install and run Atmel Studio 7
- Connect the programmer to PC USB and to the JTAG connector on the EncoderCAT PCB.
- Click **menu | tools | Device programming**.
- Select tool **Atmel_ICE** (or your programmer)
- Select device = **ATxmega192A3**.
- Select interface = **JTAG**
- Click **Apply**.
- Set interface speed to maximum (7.5Mhz) ... not essential.
- Click **Read** - device signature 0x1E744 should appear.
- Click **Memories** - we are programming flash (NOT EEPROM)
- Ensure **Erase flash before programming** and **Verify Flash after programming** are checked.
- Locate the **encat.hex** file into the file box (click the elipsis button on the right)
- Click **Program** ... the device should program and verify OK

Note: - I have have not mentioned **fuses** because the default settings are OK.

EncoderCAT - FT245RL USB device configuration

The EncoderCAT PCB contains an FT245RL USB interface device (made by FTDI)
FT_Prog is FTDI's configuration utility for the **FT245RL USB device**.

***xCatSetup.exe bundles FT_Prog and on completion offers to install it.
 This is version 2.8.2.0. It works fine on Windows 10.
 The latest version 3.6.88.402 on the FTDI website crashes for me on device scanning.
 A quick Google search reveals that I am not alone here. ... so use 2.8.2.0 !!***

USB device drivers

The installation needs:

- A **USB serial port driver**..
- The dynamic link library file **ft2dxx.dll** to communicate with EncoderCAT and FT_PROG.

xCatSetup.exe is the xCAT installer. It also has the following functions:

- Creates folder C:\Program files (x86)\hbradios\xcat\FTPprog and installs into it:
 - The FT_Prog installer: **FT_Prog_v2.8.2.0 Installer.exe**
 - The FT_Prog manual: **AN_124_User_Guide_For_FT_PROG.pdf**
- Creates folder C:\Program files (x86)\hbradios\xcat\FTDI_drivers and installs into it
 - A zipped file of USB driver files: **FTDI_drivers_2_10_00.zip**
 This can be unzipped for use in USB driver installation (see below).

Therefore please run xCatSetup.exe before carrying out USB device configuration

USB serial port driver

Connect EncoderCAT's micro USB connector to a PC USB port.

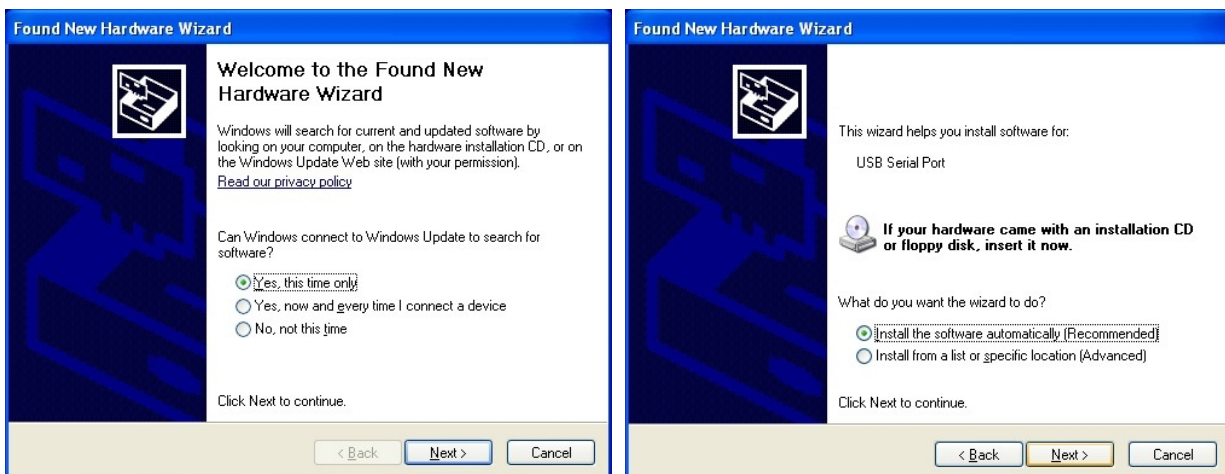
The familiar balloon popup window appears at screen bottom right. The identification text will be 'USB serial port found' (or similar) in a new gateway.

The driver may then install automatically with no further user action.

If not, then the windows below may appear:

Windows will offer to search Windows updates for the driver - this usually works ok after several seconds searching.

The screens below require answers: 'Yes this time only' and then 'Install the software automatically'



If no driver is found, then follow the on screen instructions (back button may be needed) to locate the **FTDI_drivers** folder on the EncoderCAT installer source device or **C:\Program files (x86)\hbradios\xcat\FTDI_drivers** and then click Next to find and load the driver from the CD.

Sometimes, for reasons known to Microsoft, Windows seems to lose interest in a failed USB device driver installation and no longer presents installation options.

If this happens, then you need to locate the device in Device Manager (or Hardware Manager). (use Start button > control panel > System > Device or Hardware manager)

The device will be under USB devices or Other devices.

It will have an exclamation mark against it. Right click, select properties and then Update driver.

Point the driver location Browser to the **FTDI_drivers** folder on the xCAT installation media.

or **C:\Program files (x86)\hbradios\xcat\FTDI_drivers**.

ftd2xx.dll (supplied by FTDI, the USB device manufacturer)

The correct version is in the i386 subfolder of the following download from FTDI:

<http://www.ftdichip.com/Drivers/CDM/CDM%20v2.10.00%20WHQL%20Certified.zip>

However, you do not need to download it because:

ftd2xx.dll is installed when **xCATSetup.exe** is run.

Its installed location is **C:\Windows\system32** on a 32bit machine

and **C:\Windows\sysWOW64** on a 64 bit machine

FT_Prog needs ftd2xx.dll but the FT_Prog installer appears not to install it.

However, FT_Prog uses the ftd2xx.dll installed by xCatSetup.exe

FT_Prog (supplied by FTDI, the USB device manufacturer)

The **FT_Prog** installer, **FT_Prog_v2.8.2.0 Installer.exe** is installed and optionally run by xCatSetup.exe as described above.

It can also be downloaded from:

http://www.ftdichip.com/Support/Utilities/FT_Prog_v2.8.2.0.zip

FT_Prog user manual is installed by xCatSetup.exe as described above.

It can also be downloaded from:

http://www.ftdichip.com/Support/Documents/AppNotes/AN_124_User_Guide_For_FT_PROG.pdf

Actions required are therefore:

- Run **xCatSetup.exe** from the installation media and opt to run **FT_Prog_v2.8.2.0 Installer.exe**

EEPROM

USB devices have an EEPROM containing an identifying text string and other configuration information.

This text pops up on the PC screen when the device is first connected.

Raymon is programmed to look for **FT245R** devices but only recognises them if 'EncoderCAT' appears somewhere in the text string. ('EncoderCAT' is not case sensitive here)

So the first task is to modify the EEPROM data.

This is done with **FT_Prog**.

Important:

If you make changes to the FT245RL EEPROM with FT_Prog and then rerun FT_Prog to view the settings, then you will see confirmation of your changes.

However, xCAT will not see these changes until the FT245RL device is powered down and up again.

Powering down the module does not achieve this! The FT245RL is always USB powered and so you must disconnect and reconnect the USB cable.

This is likely to be a Windows issue. FTDI's FT_Prog communicates with the FT245RL using the same FTDI dll (ftd2xx.dll).

Switch on the Gateway module and connect by USB to the PC

- You should see a pop up box informing you that an FT245R device has been found.

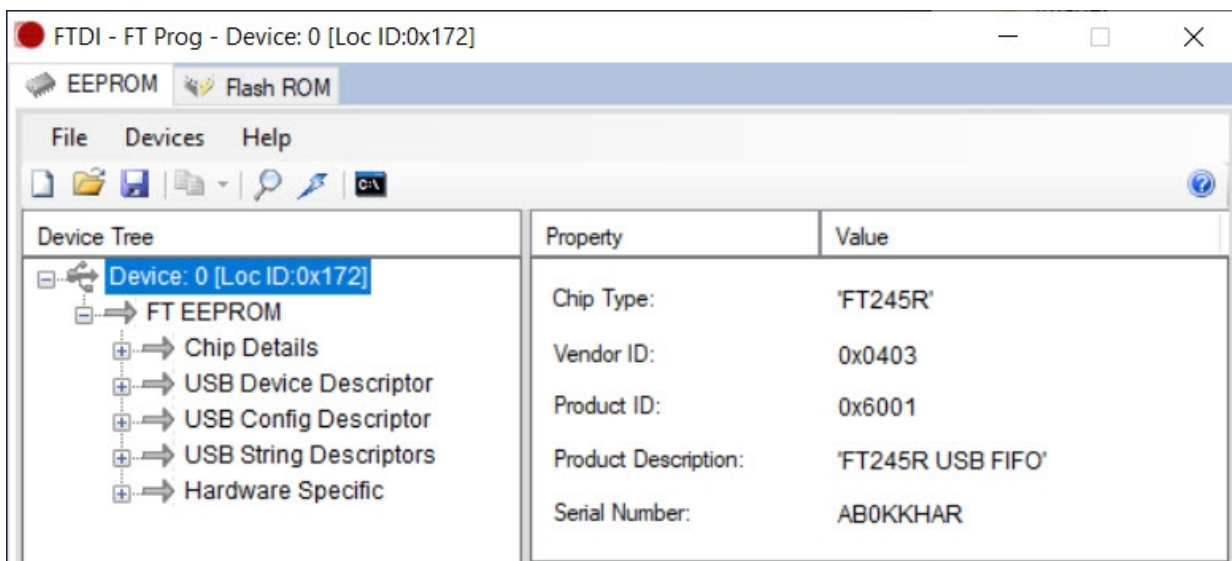
Run FT_Prog and click **Menu | Device | Scan and Parse**

You will see a tree control on the left and a summary configuration on the right.

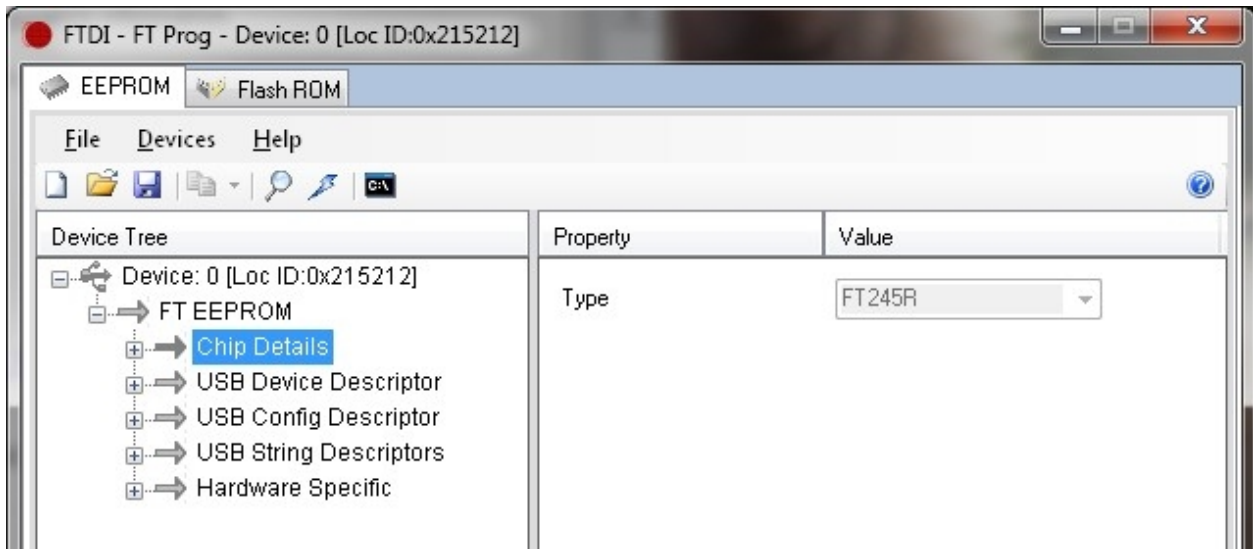
The images below were obtained from an FT245R device that had already been configured.

As such, they serve as guide for what you should configure.

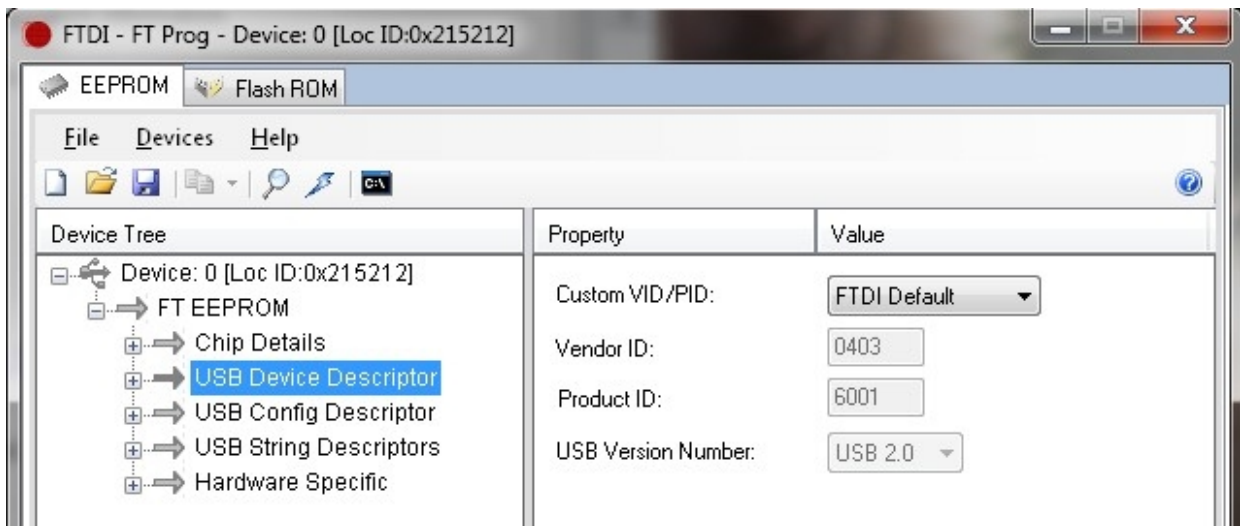
Please note: If you have other USB devices attached to your PC that use FTDI devices, then they will also be shown.



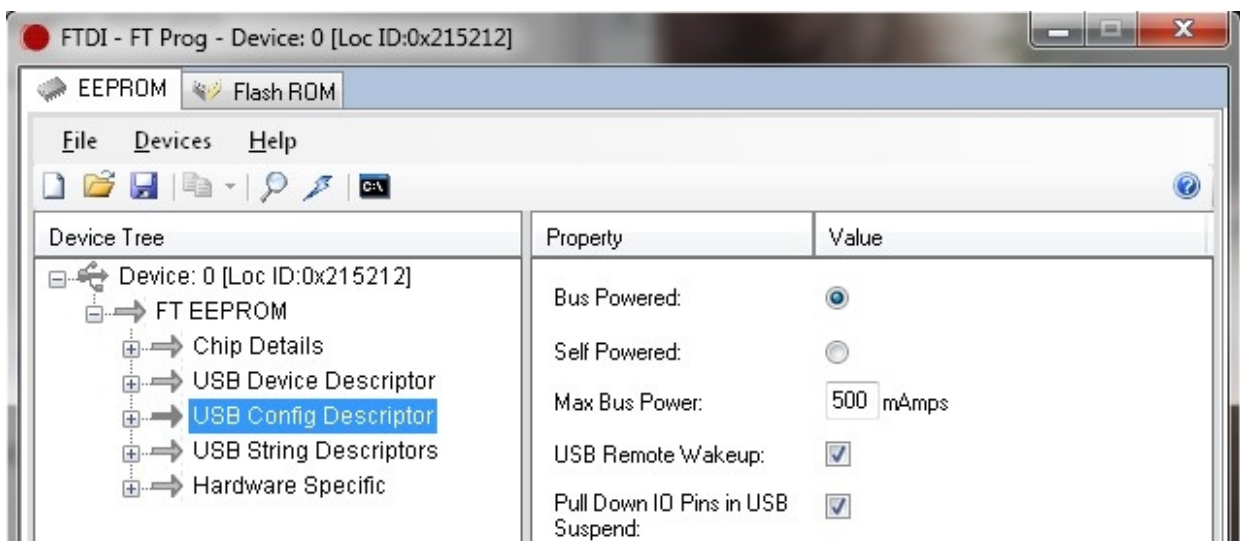
Initially the tree root is highlighted with a configuration summary showing on the right.



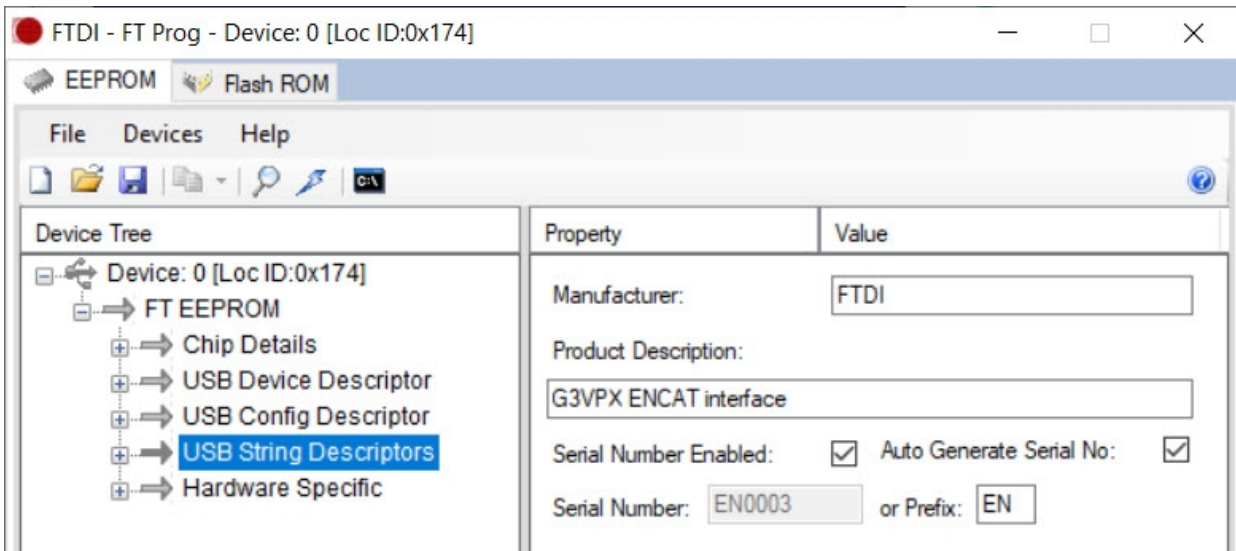
Click to select **Chip details**. FT245R - no action required



Click to select **USB Device Descriptor**: - no change needed



Click to select **USB Config Descriptor**: - Set as shown



Click to select **USB String Descriptor**:

The **Product Description** string must contain the letters: **ENCAT** or **encat** (case insensitive).

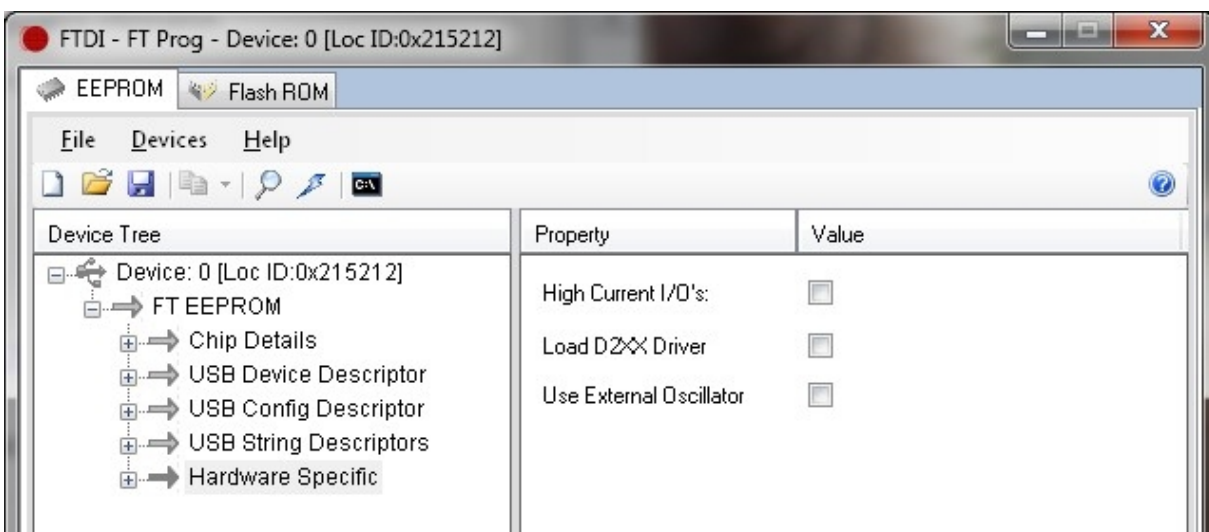
Auto Generate SerialNo: is unchecked in order to allow entry of a EncoderCAT serial number.

The serial number can be anything you wish, eg: EN0002


(Note that some of the screen images in this document show an earlier shorter serial number format.)

Please note that if you return to make a change, then **Auto-generate serial no** will be checked.

You must uncheck this, otherwise your serial number will be changed.



Click to select: Hardware Specific: These are left unchecked.

Now click the  program button OR **menu | Devices | Program**

Then click **Program** on the confirmation window and the device will be programmed.

Terminate FTProg

Unplug the USB cable

Wait 3 seconds.

Reconnect the USB cable

Start xCAT

xCAT's status bar should show:

USB: G3VPX ENCAT interface s/n: EN0003

And then, if EncoderCAT code is running:

USB: G3VPX ENCAT interface s/n: EN0003 enCAT v1.04 ready

xCAT - CAT command configuration - introduction

xCAT is a PC based application that is used to configure EncoderCAT via USB.

xCAT has three windows: **Encoder configuration**, **Lookup values** and **Debug**.

xCAT opens with the **Encoder configuration** window as shown below in **RS232** interface format:

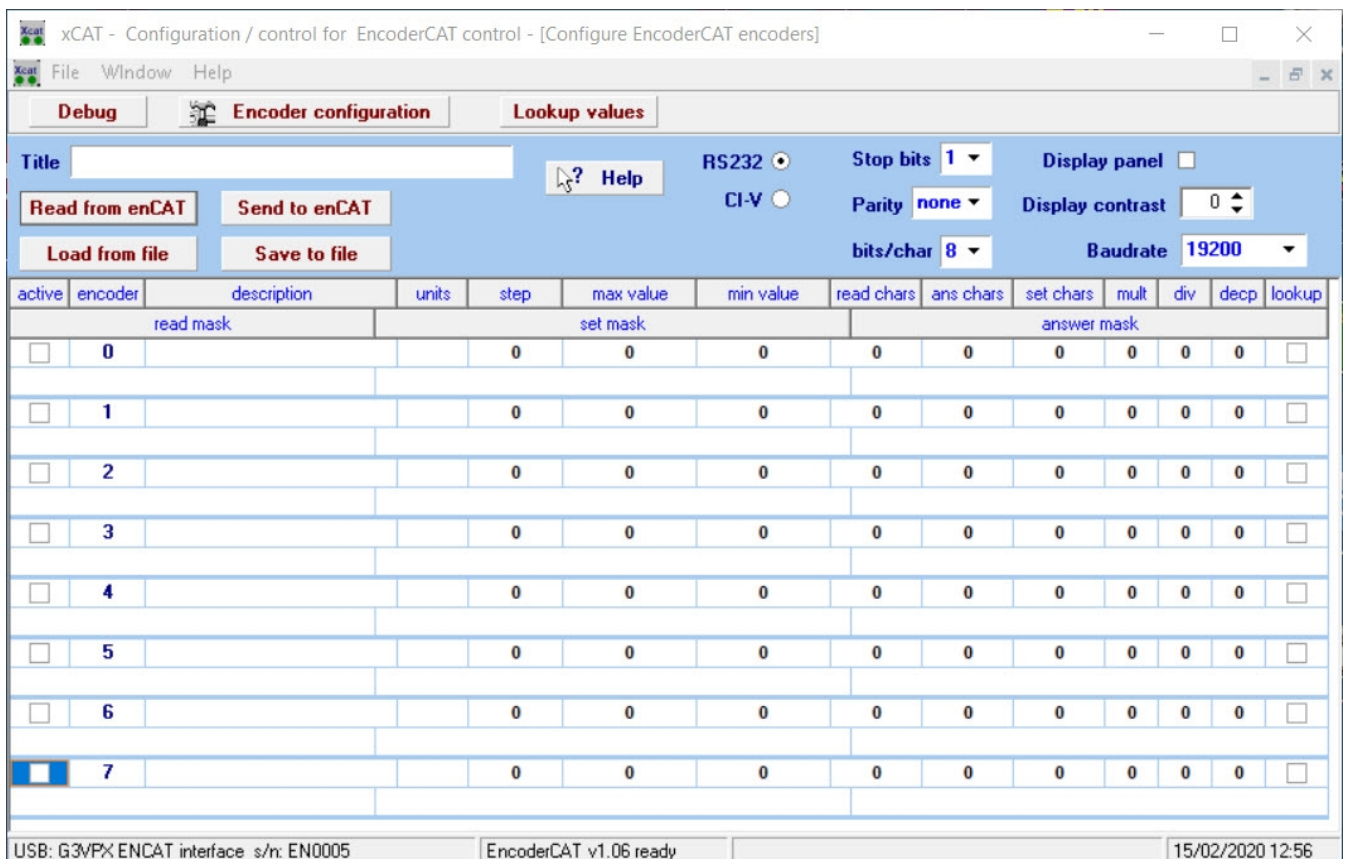
Each encoder has two rows of data. The column headers are in two corresponding rows.

The top bar data is saved to EncoderCAT and disc file with the encoder data.

ie: Title, serial settings, display enable and contrast and interface selection radio buttons (RS232 or CI-V)

All the encoder data fields are edited directly on the grid.

EncoderCAT is abbreviated to **enCAT** on the controls



If the EncoderCAT USB interface has been configured then USB... message appears status bar left.

If the EncoderCAT firmware is programmed and running, then 'EncoderCAT vN.NN ready' appears as above.

At startup, as shown above, the configuration grid is not populated with data. The RS232 interface option display by default.

- **Save** saves the grid and top bar data to a disc file (This is an editable text file in 'INI' file format.)
- **Load** loads the grid and top bar data controls from the disc file.

USB debug facility - the command bar **Debug** button opens a debug window.

xCAT / EncoderCAT have a USB debug system. EncoderCAT has debug variables MonitorA, MonitorB etc.

If MonitorA is assigned to some internal value (eg: MonitorA = bytecount) then it is displayed in xCAT debug.

Note that the Atmel Studio 7 debug system can still also be used (with the ATATMEL-ICE programmer)

RS232 and CI-V configuration displays

The RS232 / CI-V radio buttons switch between the two configuration data formats, ie:

- RS232 - ASCII character based configuration data for Yaesu, Kenwood and Elecraft radios.
(and ? for any other radio using RS232 + ASCII character data)
- CI-V - Hexadecimal / BCD based configuration data for Icom radios.

Below are images for RS232 and CI-V.

They show the column headers and encoder 0 configured for **noise reduction level**.

Some fields are common to both formats:

- Fields related to the *data value* and its increment or decrement: **max**, **min**, **step**
- Fields related to the *OLED display*: **units**, **mult**, **div**, **decp** (decimal point) and **lookup** (use lookup table)

The images below show noise reduction level settings for my:

- Yaesu FTdx101D - DNR - range 1 to 15. Encoder step set to 1.
- Icom IC7000. - NR - range 0 to 15. Encoder step set to 1.

Yaesu FTdx101D - RS232 - headers + encoder 0 - See [xCAT - Yaesu, Kenwood](#) for details

Load from file		Save to file		bits/char 8		Baudrate 38400							
active	encoder	description	units	step	max value	min value	read chars	ans chars	set chars	mult	div	decp	looku
read mask			set mask			answer mask							
<input checked="" type="checkbox"/>	0	DNR sub		1	15	1	4	6	6	1	1	0	<input type="checkbox"/>
RL1;			RL1tu;			RL1tu;							

Icom IC7000 - CI-V - headers + encoder 0 - See [xCAT - Icom](#) for details

active	encoder	description	units	step	max value	min value				mult	div	decp	lookup
rig address	encat address	subcom type	read / send	encoder mode	data digits	Com (hex)	subcom (hex)	level data	subcom 4 or 6 digit - bcd				
<input checked="" type="checkbox"/>	0	NR level		1	15	0				1	1	0	<input type="checkbox"/>
70	E0	BCD 6 digit	READ & SEND	UP / DOWN	2	1A	00	0		50114			

xcAT - Configuration - Yaesu, Kenwood

Below is the current configuration for my FTdx101D.

Clicking **READ from enCAT** produced the displayed data. (Clicking **Load** would do the same)

active	encoder	description	units	step	max value	min value	read chars	ans chars	set chars	mult	div	decp	lookup
<input checked="" type="checkbox"/>	0	DNR sub		1	15	1	4	6	6	1	1	0	<input type="checkbox"/>
<input checked="" type="checkbox"/>	1	RTTY shift	Hz	1	3	0	9	10	10	1	1	0	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	2	VOX delay	sec	1	33	0	3	5	5	1	1	3	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	3	Ext monitor ON / OFF		1	1	0	9	10	10	1	1	0	<input type="checkbox"/>
<input checked="" type="checkbox"/>	4	DNR main		1	15	1	4	6	6	1	1	0	<input type="checkbox"/>
<input checked="" type="checkbox"/>	5	Width main	kHz	1	21	1	4	7	7	1	10	2	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	6	VOX gain		5	100	0	3	6	6	1	1	0	<input type="checkbox"/>
<input checked="" type="checkbox"/>	7	RF power	watts	5	100	5	3	6	6	1	1	0	<input type="checkbox"/>

Summary of configuration data fields;

(CAT value is the numeric read from and written back to the radio. It may differ from the OLED display)

- **active** - Check this to make the encoder active.
- **encoder** - Encoder number 0 - 7. (Encoders are referenced to connections on the schematic)
- **description** - Descriptive text. Identifies the encoder in configuration, disc file and OLED display.
- **units** - Optional units for the OLED display.
- **step** - The increment (or decrement) to the CAT value (eg: RF power 0 - 100w in steps of 5w)
- **max value** - The maximum CAT value. (Keeps EncoderCAT in the radio's CAT range)
- **min value** - The minimum CAT value. (Keeps EncoderCAT in the radio's CAT range)
- **read chars** - The number of chars transmitted when requesting the CAT value from the radio.
- **ans chars** - The number of characters received in a CAT value response from the radio.
- **set chars** - The number of characters transmitted when sending a CAT value to the radio.
- **mult** - Mult and div default to 1 but can be used to scale the OLED displayed value.
- **div** - the displayed value is: $(\text{CAT value} \times \text{mult}) / \text{div}$ (Only affects the display)
- **decp** - Number of decimal places. Default = 0. eg: decp = 3 changes 75 to 0.075.
- **lookup** - If checked, then CAT value is used as a lookup index for the encoder's lookup list.
- **read mask** - The character pattern of a read request. eg: **PC;** (no data here) see: [masks](#)
- **set mask** - The character pattern of a set request. eg: **PChTu;** see: [masks](#)
- **answer mask** - The character pattern for received data following a read request, eg: **PChTu;**

RS232 parameters: **Baudrate**, **bits/char**, **stop bits** and **parity** must match the settings on the radio.

Display panel is left unchecked if there is no OLED display.

Display contrast for SSD1306 display drivers has range 0-255.

Title is used to identify the configuration. (27 chars max. Lookups have a separate title.)

EncoderCAT / xCAT - CAT command example

The examples here are taken from the Yaesu FTdx101D CAT manual.
Other Yaesu radios appear to use the same codes.
Kenwood radios use a similar system.

EncoderCAT's encoders modify the current level of their assigned radio parameter.

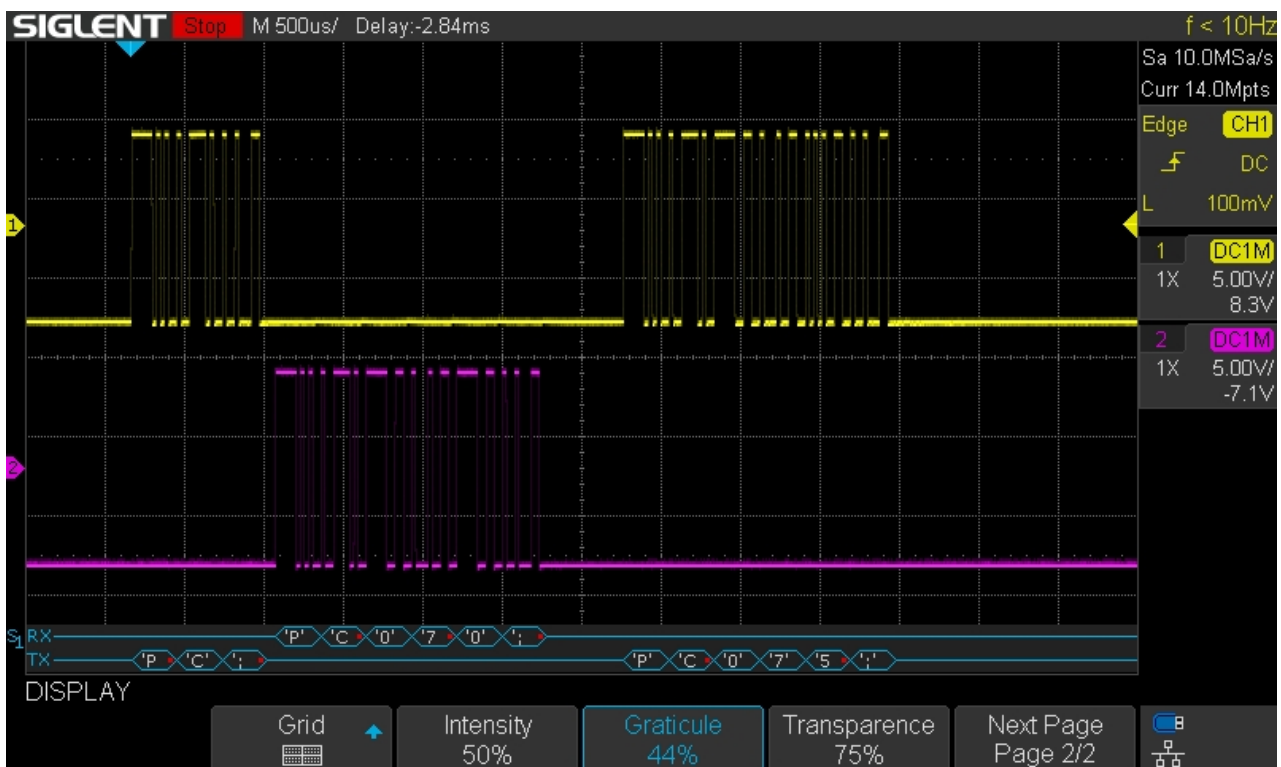
Three RS232 transactions are needed for each change. (**RF Power 70 > 75 as example**)

- **Read** - a command (with no data) is sent to read the current value. ("PC;" read chars = 3)
- **Answer** - the radio transmits back an answer containing the current value. ("PC070;" ans chars = 6)
EncoderCAT increments or decrements the value according to encoder direction and step setting.
- **Set** - EncoderCAT sends the modified value back to the radio. ("PC075;" set chars = 6)

The oscilloscope tracings below show EncoderCAT transmissions in yellow and radio response in purple.
RS232 settings are **38400 baud**, 2 stop bits, no parity, 8 bits/char.
The oscilloscope has decode switched on and set to serial data, ASCII presentation.
The decoded characters appear at the bottom of the image.

In this example, **the whole sequence takes 5ms**.

The implied speed capability of 200 encoder notches per second well exceeds realistic encoder click rates!!



EncoderCAT - Command masks examples *(Not used for Icom CI-V.)*

The examples are from the Fdx101D. Other Yaesu radios have a similar CAT system.

I have used the terms **read mask**, **set mask** and **answer mask** to describe the configuration strings.

A example from the FTdx101D CAT manual - Noise Reduction Level.

The CAT manual entry is below. There are two identical receivers, **MAIN** and **SUB**.

(Note that the terminating semicolon is part of the command)

RL	NOISE REDUCTION LEVEL										
Set	1	2	3	4	5	6	7	8	9	10	P1 0: MAIN Band 1: SUB Band P2 01 - 15
	R	L	P1	P2	P2	;					
Read	1	2	3	4	5	6	7	8	9	10	
	R	L	P1	;							
Answer	1	2	3	4	5	6	7	8	9	10	
	R	L	P1	P2	P2	;					

For a change from value 6 to 7:

- **MAIN** - read = "RL0;" answer = "RL006;" set = "RL007;"
- **SUB** - read = "RL1;" answer = "RL106;" set = "RL107;"

In **MAIN.Answer** above: **"RL0"** is the command and **"06"** is the variable data.

xCAT configuration for main receiver noise reduction level 'RL0'

active	encoder	description	units	step	max value	min value	read chars	ans chars	set chars	mult	div	decp	lookup
		read mask							set mask				
		answer mask											
<input checked="" type="checkbox"/>	4	DNR main		1	15	1	4	6	6	1	1	0	<input type="checkbox"/>
		RL0;							RL0tu;				

The answer mask is: "RL0tu;"

- **"RL0"** are **fixed** as the first three characters.
- **"tu"** indicates that the fourth and fifth characters are variable **tens** and **units**.

Further example: Main frequency

Yaesu CAT manual states:

FA	FREQUENCY MAIN BAND										
Set	1	2	3	4	5	6	7	8	9	10	P1 000030000 - 075000000 (Hz)
	F	A	P1	P1	P1	P1	P1	P1	P1	P1	
	11	12	13	14	15	16	17	18	19	20	
	P1	;									

xCAT configuration for main frequency control

active	encoder	description	units	step	max value	min value	read chars	ans chars	set chars	mult	div	decp	lookup
		read mask							set mask				
		answer mask											
<input checked="" type="checkbox"/>	0	freq main		10	75000000	30000	3	12	12	1	1	6	<input type="checkbox"/>
		FA;							FAgfedcmhtu;				

The mask string is: **FAgfedcmhtu;** which represents numbers 0 - 999 999 999 ie: 0 - 999 Mhz

The FTdx101D range is 30khz to 75Mhz - so this determines the **max** and **min** settings, 75000000 and 30000.

I have chosen a **step** of 10Hz.

The **decp** (decimal places) setting of 6 provides an OLED display in Mhz eg: 14.345700

I have provided this as an example of configuration and to prove that EncoderCAT can control main frequency.

I do not use EncoderCAT for rig tuning.

Main frequency control really needs a non-notched, high resolution encoder.

List of 'variable' characters

I have defined a small list of lower case characters that represent variable data. All other characters outside this list are interpreted as **fixed**.

The characters are g, f, e, d, c, m, h, t, u (u = units, t = tens, h = hundreds etc)
There is also s = sign - representing '-' or '+'

The codes are detailed below in an extract from EncoderCAT C source code:

EncoderCAT codes for character interpretation:

For decimal digit ascii characters, ie: '0' to '9'

```
switch(mb)
{
  case 'g': mult = 100000000; break; // g = hundred millions
  case 'f': mult = 10000000; break; // f = ten millions
  case 'e': mult = 1000000; break; // e = millions
  case 'd': mult = 100000; break; // d = hundred thousands
  case 'c': mult = 10000; break; // c = ten thousands
  case 'm': mult = 1000; break; // m = thousands
  case 'h': mult = 100; break; // h = hundreds
  case 't': mult = 10; break; // t = tens
  case 'u': mult = 1; break; // u = units
}
```

```
// Also s represents sign '-' or '+'
```

EncoderCAT / xCAT - MENU command examples

The FTdx101D has a **MENU** CAT command.

This is prefixed with the letters 'EX' and has 211 sub-commands which control menu settings.,
These commands include:

- Simple binary (ie two-state) settings, eg: external display on / off
- Settings with a few options: eg: screen saver time: OFF, 15min, 30min 60min.
- Settings controlling a value: eg: TFT contrast 0 - 20.

They are all commands that are accessed by the **FUNC** button.

This displays 29 buttons (eg: operation) which access many sub-settings:

eg: Decode Rx select, Headphone mix, ANT3 select, NB width, NB rejection.

Support for MENU was a late consideration, after the design of the standard read-modify-write design.

No design changes were needed to support it.

It just needed some testing and the addition of this documentation !

Two examples are given below:

RTTY shift

The CAT manual specifies:

- P1 = 01 RADIO SETTING
- P2 = 05 MODE RTTY
- P3 = 14 SHIFT FREQUENCY 0: 170 Hz 1: 200Hz 2: 425 Hz 3: 850 Hz

The **read** command mask is therefore: **EX010514;** read chars = 9.

The **answer** and **set** masks are: **EX010514u;** answer and set chars = 10.

u = units. The actual CAT values are 0,1,2,3 ... you have to infer from this that a single digit is needed.

I have set min value = 0 and max value = 3 (The range of the transmitted CAT value)

I have set step = 1 (We want the value to step 0 - 1 - 2 - 3 - 2 - 1 - 0)

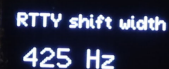
The above would display values 0, 1, 2 and 3

Using the lookup facility:

I set the lookup box to checked. I specified units = Hz.

I entered in the lookup table for encoder 1: 0 = 170, 1 = 200, 2 = 425 and 3 = 850.

The resulting OLED display is shown/



RTTY shift width
425 Hz

The configuration settings are shown below:

active	encoder	description	units	step	max value	min value	read chars	ans chars	set chars	mult	div	decp	lookup
		read mask			set mask		answer mask						
<input checked="" type="checkbox"/>	1	RTTY shift	Hz	1	3	0	9	10	10	1	1	0	<input checked="" type="checkbox"/>
EX010514;			EX010514u;				EX010514u;						

Ext monitor ON / OFF

The CAT manual specifies:

- P1 = 04 DISPLAY SETTING
- P2 = 03 EXT-MONITOR
- P3 = 01 EXT DISPLAY 0: DISABLE 1: ENABLE

The **read** command mask is therefore: **EX040301**; read chars = 9.

The **answer** and **set** masks are: **EX040301u**; answer and set chars = 10.

u = units. The actual CAT values are 0 and 1 ... you have to infer from this that a single digit is needed.

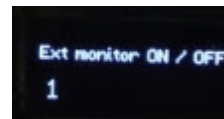
I have set min value = 0 and max value = 1 (The range of the transmitted CAT value)

I have set step = 1 (We want the value to step 0 - 1 - 0)

The above would display values 0 and 1

I specified no units and no lookup.

The resulting OLED display is shown. 1 = ON 0 = OFF



The configuration settings are shown below:

active	encoder	description	units	step	max value	min value	read chars	ans chars	set chars	mult	div	decp	lookup
		read mask					set mask	answer mask					
<input checked="" type="checkbox"/>	3	Ext monitor ON / OFF		1	1	0	9	10	10	1	1	0	<input type="checkbox"/>
EX040301;			EX040301u;				EX040301u;						

EX command in other Yaesu radios

I have not surveyed all the Yaesu radios' CAT manuals.

For the FT2000, the CAT system is very similar but with some different values.

(On the FT2000, the menus are navigated by a rotary encoder rather than by buttons and sub buttons.

This is consistent with it having one menu level containing 136 items, rather than sub-levels.)

The **MENU** CAT command uses the same **EX** prefix, but the structure of the following characters is different.

The first parameter, **P1** is specified as using three rather than two characters. It has values 000 to 136.

The second parameter, **P2** is the CAT value and appears to have between 1 and 6 or 7 characters.

eg: **50Mhz repeater shift** is **EX075**. Range specified as 0000 ~ 4000 kHz (10Hz / step)

For 10Hz resolution, we would need 6 digits (specified to 10Hz) OR 7 digits (specified to 1Hz)

The answer and set masks would therefore be **EX075dcmhtu**; (6 digits) or **EX075edcnhtu**; (7 digits)

I do not have an FT2000 to test this I traded it in for the FTdx101D !!

xCAT - Icom CI-V configuration

Icom CI-V uses CAT byte sequences are that specified in hexadecimal:

- **Hexadecimal** command codes, eg: 1A hex
- **Decimal data values** represented in **Binary Coded Decimal format** (BCD)

Hexadecimal bytes are represented in this document with the prefix 0x. (as used C and PHP languages)
eg: 0x2A is hexadecimal 2A = 2 x16 + 10 = decimal 42.

BCD format represents two decimal digits as a single hexadecimal byte.

eg: decimal 37 is 0x37 ... easy to read (but on hex to dec conversion = 3 x16 +7 = 55 decimal)

EncoderCAT software does the decimal to BCD conversion for you.

This allows you to enter command 050014 as such in decimal

(EncoderCAT then transmits this as three bytes: 0x50 0x00 0x14)

Icom's command system is complicated by what would appear to be historical / evolutionary features.

The range of formats includes:

- single hex command byte
- hex command byte + hex sub-command
- hex command byte + hex sub-command byte + bcd data bytes. (read only - but read/write in later rigs)
- hex command byte + two bcd data bytes (read/write). The bcd bytes referred to as sub-command
- hex command byte + three bcd data bytes. (read/write). The bcd bytes referred to as sub-command

EncoderCAT provides a CI-V user configuration system that supports these formats.

Most read - modify – write commands would use the last option.

Below is a development IC7000 configuration. Editing is performed directly on the grid.

This display is obtained by either **Read from EncoderCAT** or **Load from file**.

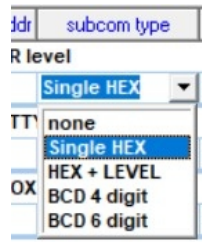
Note that hexadecimal values are shown in green.

active	encoder	description	units	step	max value	min value	mult	div	decp	lookup		
rig address	encat address	subcom type	read / send	encoder mode	data digits	Com (hex)	subcom (hex)	level data	subcom 4 or 6 digit - bcd			
<input checked="" type="checkbox"/>	0	NR level		1	15	0			1	1	0	<input type="checkbox"/>
70	E0	BCD 6 digit	READ & SEND	UP / DOWN	2	1A	00	0		50114		
<input checked="" type="checkbox"/>	1	RTTY shift width	Hz	1	2	0			1	1	0	<input checked="" type="checkbox"/>
70	E0	BCD 6 digit	READ & SEND	UP / DOWN	1	1A	00	0		50011		
<input checked="" type="checkbox"/>	2	VOX delay	sec	1	20	0			1	1	1	<input type="checkbox"/>
70	E0	BCD 6 digit	READ & SEND	UP / DOWN	2	1A	00	0		50117		
<input checked="" type="checkbox"/>	3	Swap VFO A/B (toggle)		1	0	0			1	1	0	<input type="checkbox"/>
70	E0	Single HEX	SEND ONLY	TOGGLE	0	07	B0	0		0		
<input checked="" type="checkbox"/>	4	NR ON / OFF		5	255	0			1	1	0	<input type="checkbox"/>
70	E0	HEX + LEVEL	SEND ONLY	ON / OFF	0	16	40	0		0		
<input checked="" type="checkbox"/>	5	Speech comp level		1	10	0			1	1	0	<input type="checkbox"/>
70	E0	BCD 6 digit	READ & SEND	UP / DOWN	2	1A	00	0		50094		
<input checked="" type="checkbox"/>	6	Frequency 03 > 05	MHz	100	29700000	1800000			1	10	5	<input type="checkbox"/>
70	E0	none	READ & SEND	UP / DOWN	8	03	00	0		0		
<input checked="" type="checkbox"/>	7	RF pwr 25% / 100%	%	191	0	0			100	255	0	<input type="checkbox"/>
70	E0	HEX + LEVEL	SEND ONLY	UP / DOWN	0	14	0A	64		0		

USB: G3VFX ENCAT interface s/n: EN0005 EncoderCAT v1.06 ready 15/02/2020 13:22

Icom CI-V - configuration field details

Fields: **subcom type**, **send / read**, and **encoder mode** are option selectors. eg: **subcom type** as shown here. The selectors are drop down lists located in the grid.



A full explanation of these fields is on the following page.

CI-V command byte sequence is as below (rig addr and EncoderCAT addr swap place in data from the radio);

byte 0	byte1	byte 2	byte 3	byte 4	0 - 3 bytes	0 to 5 bytes	end (oblig)
0xFE	0xFE	rig addr	EncoderCAT addr	com	sub com	data bytes	0xFD

Note that subcom, if present is specified as a single hex byte or two or three bcd bytes.

Summary of configuration data fields;

(CAT value is the numeric read from and written back to the radio. It may differ from the OLED display)

- **active** - Check this to make the encoder active.
- **encoder** - Encoder number 0 - 7. (Encoders are referenced to connections on the schematic)
- **description** - Descriptive text. Identifies the encoder in configuration, disc file and OLED display.
- **units** - Optional units for the OLED display.
- **step** - The increment (or decrement) to the CAT value for each encoder notch, (eg: RF power 0 - 100w in steps of 5w)
- **max value** - The maximum CAT value. (Keeps EncoderCAT in the radio's CAT range)
- **min value** - The minimum CAT value. (Keeps EncoderCAT in the radio's CAT range)
- **mult** - Mult and div default to 1 but can be used to scale the OLED displayed value.
- **div** - the displayed value is: $(\text{CAT value} \times \text{mult}) / \text{div}$ (Only affects the display)
- **decp** - Number of decimal places. Default = 0. eg: decp = 3 changes 75 to 0.075.
- **lookup** - If checked, then CAT value is used as a lookup index for the encoder's lookup list.
- **rig addr** - The hexadecimal address of the radio (IC7000 defaults to 0x70)
- **EncoderCAT addr** - An address transmitted with the command and received with responses.
- **subcom type** - Options: **none**, **Single HEX** , **HEX + LEVEL**, **BCD 4 digit**, **BCD 6 digit**.
- **read / send** - Options: **READ & SEND**, **SEND ONLY**,
- **encoder mode** - Action on the rig parameter value : **no data**, **ON /OFF**, **UP /DOWN**, **TOGGLE**.
- **data digits** - Number of **decimal** data digits sent and returned in a read - modify - write command
eg: RFpower 0-255 is 3 digits (resulting in 2 BCD bytes (eg: 215 coded as 0x02 0x15)
This is command= 0x1A subcommand = 050001 (subcom type = BCD 3 digit)
Note that 0-255 may represent 0 -100w so convert for display using mult=100, div=255
- **com (hex)** - Hexadecimal command byte (always the 5th byte)
- **subcom (hex)** - Hexadecimal subcommand if present
(Set to 0 for **BCD 6 digit** because the resulting 3 bcd digits **are the subcommand**)
- **level data** - Data byte specifically for **HEX + LEVEL** ... (which is Icom CAT manual terminology)
- **subcom 4 or 6 bcd** - Decimal value EncoderCAT converts to bcd for **BCD 4 digit** / **BCD 6 digit**.
eg; 500114 decimal ... transmitted as hex bytes 0x05 0x01 0x14

RS232 parameters: **Baudrate**, **bits/char**, **stop bits** and **parity** must match the settings on the radio.

Display panel is left unchecked if there is no OLED display.

Display contrast for SSD1306 display drivers has range 0-255.

Title is used to identify the configuration. (27 chars max. Lookups have a separate title.)

Icom CI-V option selectors

Some CI-V commands are simply not suitable for a rotary encoder action. I have tried to make provision for the user to configure all the commands that *are* potentially suitable. The configuration design is unavoidably complicated by the fact that CI-V uses a range of different command systems as listed in the previous page. The three selector fields **subcom type**, **read / send** and **encoder mode** provide configuration support for these different command systems and are detailed below.

My biggest surprise was that BCD coded frequency is sent with least significant byte (LSB) first, whereas other BCD coded data has most significant byte (MSB) first !

Icom CI-V read / send field

read / send = READ & SEND

The **read / send** field has options: **READ & SEND** and **SEND ONLY**.
(There is no use for **read-only** actions in this encoder unit.)

Most EncoderCAT encoder configurations will be **read - modify - write** where we select **READ & SEND**.

The majority of these use **subcom type = BCD 6 digit**.

Here, sub-command section of the byte sequence is three BCD bytes representing a 5 digit decimal number. eg: for IC7000 Noise Reduction level: the manual specifies **command = 0x1a**, **sub-command = 050114**.

You enter **50114** in xCAT's **subcom 4 or 6 digit field** and EncoderCAT transmits as **0x05 0x01 0x14**.

read / send = SEND ONLY

Icom CI-V has unidirectional command types that only send data to the radio.
(The radio responds with a simple 'OK' message ... on receipt of which the OLED display is updated.)

EncoderCAT holds no data to send.

However, the xCAT configuration provides for two useful applications for SEND ONLY. These provide binary switching between two states by turning the encoder right or left. They use **subcom type = HEX + LEVEL** and **read / send = SEND ONLY**

- It is possible to use an encoder as an on / off switch where rotate right = ON and rotate left = OFF. For this we set **encoder mode = ON / OFF** (The OLED display shows ON or OFF instead of 0 or 1)
- If we set **encoder mode = UP / DOWN**, then we can switch between **data level** and **data level + step**
This is shown for encoder 7 in the above window image. **level data = 64** and **step = 191**
It switches RF power between 64 and 255
- but we can use **mult** and **div** scaling to make the display show 25% and 100%. (or 25w and 100w)
See [xCAT examples - RF power switching](#)

Other '**SEND ONLY**' commands with more than two options can similarly be used to switch between two selected options or values. eg: IC7000 com = 0x16 subcom = 0x12 AGC 1=Fast 2=Mid 3=Slow.
With data level = 1 and step = 2 we switch between 1 and 3 ie: Fast and Slow

Icom CI-V subcom type field

This has already been partly discussed above
It is an option selector with options:

- **none** This will be rarely used as there are very few commands with no subcommand.
If you set read / send = READ & SEND and command 0x03 (read frequency) the EncoderCAT can tune the radio. EncoderCAT automatically changes the command to 0x05 (set frequency) on writing back to the radio (see test config. above).
This could be used for frequency fine adjustment but not really as a main tuner.
See [xCAT examples Icom - Frequency control](#)
- **SINGLE HEX** This refers to command + single hex subcommand byte and no data.
If configured, the command will be sent on encoder movement in either direction. (not very useful!)
- **HEX + LEVEL** Main use is with encoder mode = ON/OFF or UP / DOWN
and read / send = SEND ONLY for command = 0x16 as described above.
See example above of encoder 4 > NR ON/OFF switching.
- **BCD 6 digit** This is by far the commonest read - modify - write command.
Here, sub-command section of the byte sequence is three BCD bytes representing a 5 digit decimal number.
eg: for IC7000 Noise Reduction level:
the manual specifies **command = 0x1a, sub-command = 050114**.
You enter **50114** in xCAT's **subcom 4 or 6 digit field**
and EncoderCAT transmits as **0x05 0x01 0x14**.
- **BCD 4 digit** As BCD 6 digit but with a four digit decimal number translating to two BCD bytes.

Icom CI-V encoder mode field

- **none** unspecified - no action
- **ON / OFF** Use of the encoder as an ON /OFF switch by right / left rotation.
This is with unidirectional command codes.
We set **subcom type = HEX + LEVEL** and **read / send = SEND ONLY**
(The OLED display shows ON or OFF instead of 0 or 1)
- **UP / DOWN** With read / send = READ & SEND and subcom type - BCD 6 digit
this is the encoder mode for read-modify-write encoder action.
(It is also used read / send = SEND ONLY for binary switching between **two values or options - see above and also the examples**.)
- **TOGGLE** There are a few CI-V commands that toggle between two states.
eg: command = 0x07 sub command = 0xB0 VFO A/B toggle.

xCAT EncoderCAT configuration examples

All the example are for an IC7000 ... which is my only ICOM radio !!

I quote from the IC7000 instruction manual. You can download this from Icom.

The data presentation is the same for the more modern radios ... with different command code allocations.

All configurations have **rig address = 0x70** (IC7000 default) and **encat address = 0xE0** (any value should be ok)

NR level

Manual page 146: command 1A sub command 050114

Send/read NR level set (0 = 0 to 15 = 15)

active	encoder	description		units	step	max value	min value			mult	div	decp	lookup
rig address	encat address	subcom type	read / send	encoder mode	data digits	Com (hex)	subcom (hex)	level data	subcom 4 or 6 digit - bcd				
<input checked="" type="checkbox"/>	0	NR level			1	15	0			1	1	0	<input type="checkbox"/>
70	E0	BCD 6 digit	READ & SEND	UP / DOWN	2	1A	00	0	50114				

NR level
15

Encoder 0 min = 0 max = 15 step = 1

mult and div = 1 (ie no scaling) decp = 0 (no decimal point) no lookup

command = 0x1A subcommand(hex) = 0 - this is not used with BCD 6 digit

(the sub command is subcom 4 or 6 digit = 50114 which codes as 0x05 0x01 0x14)

subcom type = BCD 6 digit read /send = READ & SEND encoder mode = UP / DOWN

data digits = 2 (0-15 needs 2 decimal digits > one bcd byte). level data unused

NR on / off

Manual page 143: command 16 sub command 40

Noise reduction (0 = OFF; 1 = ON)

active	encoder	description		units	step	max value	min value			mult	div	decp	lookup
rig address	encat address	subcom type	read / send	encoder mode	data digits	Com (hex)	subcom (hex)	level data	subcom 4 or 6 digit - bcd				
<input checked="" type="checkbox"/>	4	NR ON / OFF			5	255	0			1	1	0	<input type="checkbox"/>
70	E0	HEX + LEVEL	SEND ONLY	ON / OFF	0	16	40	0	0				

NR ON / OFF
ON

Encoder 4 min, max, step and data digits unused

mult, div and decp are OLED display controls - unused. OLED displays 0 / 1 as ON / OFF.

Command = 16 subcommand (hex) = 40 level data unused subcom 4 or 6 digit unused

subcom type = HEX + LEVEL read / send - READ ONLY encoder mode = UP / DOWN

data digits = 0 (unused) level data = 0 (unused)

RTTY shift width

Manual page 144: command = 1A subcommand = 050011

Send /read RTTY shift width (0 = 170 Hz, 1 = 200 Hz, 2 = 425 Hz)

active	encoder	description		units	step	max value	min value			mult	div	decp	lookup
rig address	encat address	subcom type	read / send	encoder mode	data digits	Com (hex)	subcom (hex)	level data	subcom 4 or 6 digit - bcd				
<input checked="" type="checkbox"/>	1	RTTY shift width	Hz	1	2	0				1	1	0	<input checked="" type="checkbox"/>
70	E0	BCD 6 digit	READ & SEND	UP / DOWN	1	1A	00	0	50011				

RTTY shift width
425 Hz

Encoder 1 min = 0 max = 2 step = 1 div = 1 mult =

1 (no scaling) decp = 0 (unused) units = Hz

Lookup ticked See lookup table settings >>>

CAT value	Encoder 0	Encoder 1
0	0	170
1	0	200
2	0	425

command = 0x1A subcommand (hex) = 0 (unused). Subcommand is 6 digit decimal: 050011 which codes as 0x05 0x01 0x11

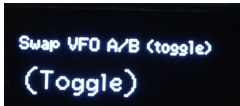
subcom type = BCD 6 digit read /send = READ & SEND encoder mode = UP / DOWN

data digits = 1 (0-3 needs 1 decimal digits > one bcd byte). level data unused

Swap VFO A / B (toggle)

Manual page 142: command = 0x07 subcommand = 0xB0
Exchange VFO A and VFO B.

active	encoder	description		units	step	max value	min value			mult	div	decp	lookup
rig address	encat address	subcom type	read / send	encoder mode	data digits	Com (hex)	subcom (hex)	level data	subcom 4 or 6 digit - bcd				
<input checked="" type="checkbox"/>	3	Swap VFO A/B (toggle)			1	0	0			1	1	0	<input type="checkbox"/>
70	E0	Single HEX	SEND ONLY	TOGGLE	0	07	B0	0	0				

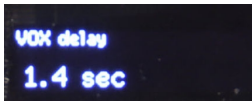


Encoder 3 min, max, step, mult, div, decp, lookup all unused
command = 0x07 subcommand (hex) = 0x0B
sub command 4 or 6 digit unused level data and data digits unused
subcom type = single hex read / send = SEND ONLY encoder mode = TOGGLE
data digits = 0 (unused). level data unused

VOX delay

Manual page 146: command = 1A subcommand = 050117
Send / read VOX delay (0 = 0.0 sec. to 20 = 29 sec.)

active	encoder	description		units	step	max value	min value			mult	div	decp	lookup
rig address	encat address	subcom type	read / send	encoder mode	data digits	Com (hex)	subcom (hex)	level data	subcom 4 or 6 digit - bcd				
<input checked="" type="checkbox"/>	2	VOX delay	sec	1	20	0			1	1	1	<input type="checkbox"/>	
70	E0	BCD 6 digit	READ & SEND	UP / DOWN	2	1A	00	0	50117				



Encoder 2 min = 0 max = 20 step = 1 decp = 1 to present in seconds Units = 'sec'.
command = 0x1A subcom (hex) = 0 = unused. - this is not used with BCD 6 digit
The sub command is subcom 4 or 6 digit = 50117 which codes as 0x05 0x01 0x17.
subcom type = BCD 6 digit read / send = READ & SEND encoder mode = UP / DOWN
data digits = 2 (0-20 needs 2 decimal digits > one bcd byte). level data unused

Speech comp level

Manual page 145 command = 1A subcommand = 050094
Send / read speech compressor level (0 = 0 to 10 = 10)

active	encoder	description		units	step	max value	min value			mult	div	decp	looku
rig address	encat address	subcom type	read / send	encoder mode	data digits	Com (hex)	subcom (hex)	level data	subcom 4 or 6 digit - bcd				
<input checked="" type="checkbox"/>	5	Speech comp level			1	10	0			1	1	0	<input type="checkbox"/>
70	E0	BCD 6 digit	READ & SEND	UP / DOWN	2	1A	00	0	50094				



Encoder 5 min = 0 max = 10 step = 1 decp = 0 = no dec point. . Units not set.
command = 0x1A subcom (hex) = 0 = unused. - this is not used with BCD 64 or digit
The sub command is subcom 4 or 6 digit = 50094 which codes as 0x05 0x00 0x94.
subcom type = BCD 6 digit read / send = READ & SEND encoder mode = UP / DOWN
data digits = 2 (0-10 needs 2 decimal digits > one bcd byte). level data unused

Frequency 03 > 05

Manual page 142: command = 03 read -- EncoderCAT changes to command = 05 to write back. no subcommand

Read operating frequency

active	encoder	description		units	step	max value	min value			mult	div	decp	lookup
rig address	encat address	subcom type	read / send	encoder mode	data digits	Com (hex)	subcom (hex)	level data	subcom 4 or 6 digit - bcd				
<input checked="" type="checkbox"/>	6	Frequency 03 > 05			100	29700000	1800000			1	10	5	<input type="checkbox"/>
70	E0	none	READ & SEND	UP / DOWN	8	03	00	0	0				

Frequency 03 > 05
3.76715 MHz

Encoder 6 min = 1800000, max = 29700000, step = 100

mult, div unused decp = 6 to present in MHz, eg 3.345100 units = MHz

command = 0x03 no subcommand EncoderCAT changes command to 0x05 on send back

sub command 4 or 6 digit unused level data and data digits unused

subcom type = none read / send = READ & SEND encoder mode = UP / DOWN

data digits = 0 (unused). level data unused

RF power 25% / 100% switching

... to illustrate binary switching between two preset levels

Manual page 143: command = 14 subcommand = 0A + Level data

RF power setting (0 = mini. to 255 = max.)

active	encoder	description		units	step	max value	min value			mult	div	decp	lookup
rig address	encat address	subcom type	read / send	encoder mode	data digits	Com (hex)	subcom (hex)	level data	subcom 4 or 6 digit - bcd				
<input checked="" type="checkbox"/>	7	RF pwr 25% / 100%	%	191	0	0				100	255	0	<input type="checkbox"/>
70	E0	HEX + LEVEL	SEND ONLY	UP / DOWN	0	14	0A	64	0				

RF pwr 25% / 100%
25 %

Encoder 7 min, max unused

step = 191 level data = 64

Power value switches between 64 on left rotation and 64+191 on right rotation

Units = '%' decp = 0 (no decimal point)

command = 0x14 subcom (hex) = 0x0A subcom 4 or 6 digit unused no lookup

For OLE display, mult=100 and div=255 scale: value x 100 / 255 to display 0 - 100w

RF power full control range

RF power 0 - 100w

Manual page 143 command = 1A subcommand = 050001

RF power setting (0 = mini. to 255 = max.)

active	encoder	description		units	step	max value	min value			mult	div	decp	lookup
rig address	encat address	subcom type	read / send	encoder mode	data digits	Com (hex)	subcom (hex)	level data	subcom 4 or 6 digit - bcd				
<input checked="" type="checkbox"/>	7	RF power	w	10	255	0				100	255	0	<input type="checkbox"/>
70	E0	BCD 6 digit	READ & SEND	UP / DOWN	3	1A	00	0	50001				

RF power
74 w

Encoder 7 min = 0 max = 255 step = 10 decp = 0 = no dec point. . Units = 'w'

command = 0x1A subcom (hex) = 0 = unused. - this is not used with BCD 6 or 4 digit

The sub command is subcom 4 or 6 digit = 50001 which codes as 0x05 0x00 0x01.

subcom type = BCD 6 digit read / send = READ & SEND encoder mode = UP / DOWN

data digits = 3 (0-255 needs 3 decimal digits > two bcd bytes). level data unused

For OLE display, mult=100 and div=255 scale: value x 100 / 255 to display 0 - 100w

EncoderCAT Oscilloscope views of CI-V data

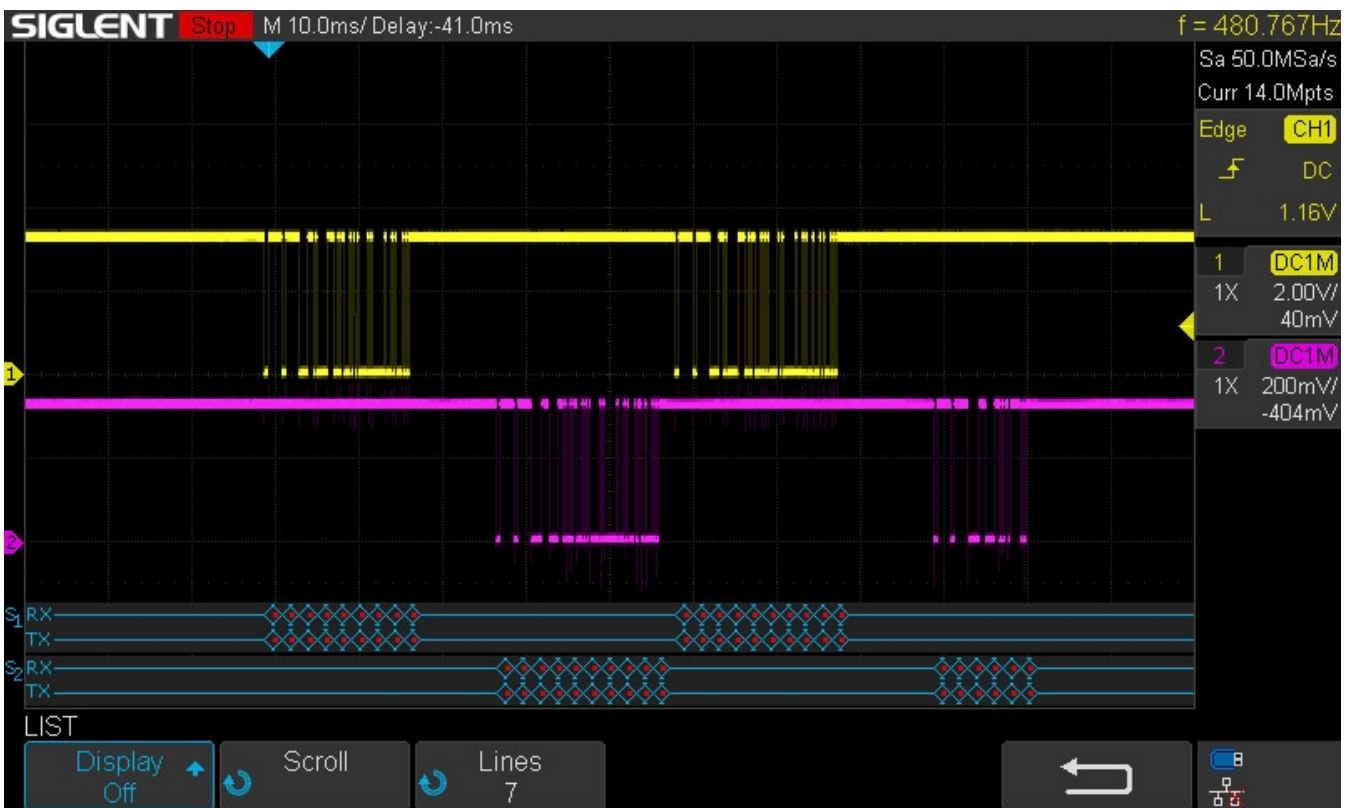
The traces below are from the Rx and Tx lines of the ATxmega192A3 microprocessor USART D0. A trace from the bidirectional, single wire CI-V line would have both Tx and Rx on the same trace.

Most CI-V interfaces would show Tx data on the Rx line because they are connected together (via buffers).

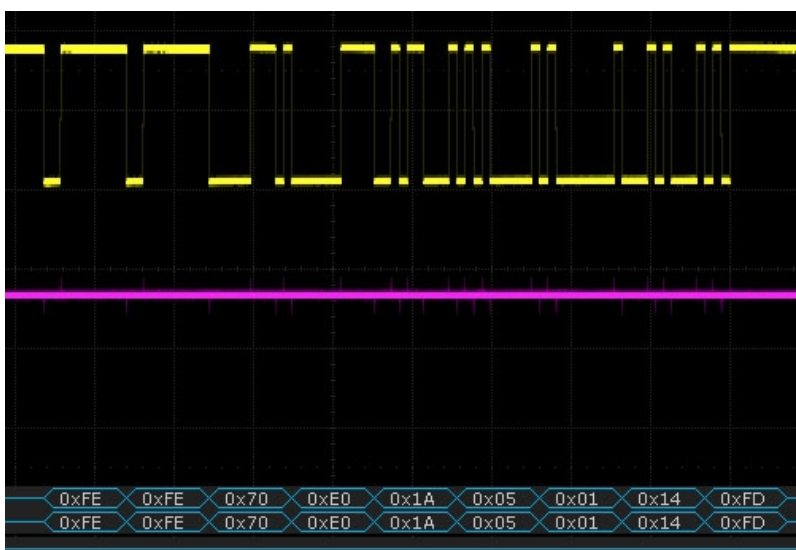
EncoderCAT uses half of an NXP PCA9600 buffer (This is a dual I2C CAT5 line driverreversed)

The Tx data does not appear on the Rx line because the internally connected Tx buffer output and Rx buffer input have voltage levels designed to prevent this from happening.

(The pull down low voltage of the Tx buffer output (0.5v) is higher than the low voltage threshold of the Rx buffer input.)



The scope screen shot above shows the sequence for IC7000 Noise reduction level changing from 9 to 8. The yellow trace is the Tx data from EncoderCAT and the purple trace is the responses from the ic7000



This is the TX request from EncoderCAT expanded to view the data decode.

The sequence is:

0xFE 0xFE - obligatory header bytes

0x70 - IC7000's address

0xE0 - address assigned to EncoderCAT

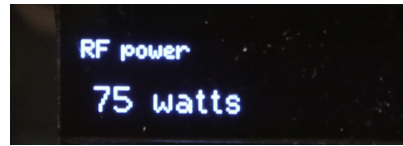
0x05 0x01 0x14 the subcommand
(three bcd bytes for 050114)

0xFD - obligatory terminator

xCAT - OLED display options

OLED display panel

EncoderCAT has an optional small OLED display.
The firmware is designed to drive an I2C connected display which conforms to SSD1306 driver specification.



The display identifies the operated encoder and stays with that encoder until another is operated.

For each encoder, the user can configure for the OLED display (with xCAT) :

- **Description**, eg: "RF power" as above
- **Units** (eg: watts, Hz, kw, dB),
- **Scaling** of CAT value ie: (value x multiplier) / divisor
- Optional **decimal point** eg: change 75 watts to 0.075 kw
- **Lookup table**, eg: value 1 -> 300Hz 2 -> 600Hz 3 -> 1000Hz

My prototype uses a small 128 x 32 pixel monochrome display.
These are available from a number of eBay suppliers and cost approximately £4.00.
The display unit does not contain font data.

I have two proportional font character sets in the firmware:

- one 16 bits high for the data and one 8 bits high for the description.

I2C address: All the SSD1306 display that I found had address = 0x70
... so this is fixed in the code. If other addresses are encountered, then the code can be changed or an address setting field added to the xCAT configuration data.

Display configuration options only affect the display.
They have no effect on the encoder's action on the radio.

xCAT encoder configuration has five fields that control the display:

- **Description** - this is a text description of up to 21 characters.
It is displayed at the top of the display in 8 bit high font.
- **units** - up to six characters displayed after the value, eg MHz, ms etc
- **mult** - this is a multiplier for scaling the value for display.
- **div** - this is a divisor for scaling of the data for display
- **decp** - if non- zero this put a decimal point in the displayed value.
- **lookup** - if checked, the CAT value will be used as the index for the encoder's lookup table. The displayed value is the value from the table.
The presented lookup value can be modified using **mult**, **div** and **decp**.

Scaling

The scaling calculation is **(CAT value) x mult / div**

Thus **mult = 100** and **div = 255** will scale values **0 - 255** to **0 - 100**
(as in the RF power example in [xCAT examples lcom](#))

The default value for no scaling are mult = 1 and div = 1

Decimal point

Setting **decp** to a value greater than zero introduces a decimal point in the display.

eg: **3.562300 MHz** **decp = 6**

1.8 ms **decp = 1**

Lookups - see next page: [xCAT - OLED display lookups](#)

xCat - Lookup list configuration

Example from **Yaesu FTdx101D CAT manual**:

Vox Delay Time: CAT value range is **0 - 33**

00 = 30msec 01 = 50 msec 02 = 100msec 03 = 150 msec 04 = 200msec 05 = 250msec
06 = 300msec 07 = 400msec 08 = 500msec 09 = 600msec 33 = 3000msec

It is observed that CAT value has a non-linear relationship to delay time:

A lookup table is needed ... this is purely for the OLED display.... The actual CAT value is managed as 0 - 33.

Click the **Lookup values** button to display the window below:

Vox delay is encoder 2.

CAT value	Encoder 0	Encoder 1	Encoder 2	Encoder 3	Encoder 4	Encoder 5	Encoder 6	Encoder 7
0	0	600	30	0	0	600	0	0
1	0	300	50	0	0	300	0	0
2	0	400	100	0	0	400	0	0
3	0	600	150	0	0	600	0	0
4	0	850	200	0	0	850	0	0
5	0	1100	250	0	0	1100	0	0
6	0	1200	300	0	0	1200	0	0
7	0	1500	400	0	0	1500	0	0
8	0	1650	500	0	0	1650	0	0
9	0	1800	600	0	0	1800	0	0
10	0	1950	700	0	0	1950	0	0
11	0	2100	800	0	0	2100	0	0
12	0	2200	900	0	0	2200	0	0
13	0	2300	1000	0	0	2300	0	0
14	0	2400	1100	0	0	2400	0	0
15	0	2500	1200	0	0	2500	0	0
16	0	2600	1300	0	0	2600	0	0
17	0	2700	1400	0	0	2700	0	0
18	0	2800	1500	0	0	2800	0	0
19	0	2900	1600	0	0	2900	0	0

USB: G3VPX ENCAT interface s/n: EN0005 EncoderCAT v1.06 ready 15/02/2020 13:25

A column of lookup values can be assigned to each of encoders 0 -7.

Each encoder can have up to 54 values. Unused values outside the CAT value range are ignored.

If the **lookup** field is checked in **encoder configuration** then the OLED displayed value will be from the lookup column.

The lookup values are edited directly on the grid.

The lookup values are stored in EncoderCAT processor EEPROM and in disc file.

This is managed as with encoder configuration data with buttons:

Read from enCAT, Send to enCAT, Save and Load.

*Please note that the order of events is lookup > scaling > decimal point.
Thus the **mult**, **div** and **dec** settings can be applied to the lookup values.*

eg: Encoder 5 in the above example controls main receiver IF passband width. I entered lookup values in Hz. However, if we set **div** = 10 then 2400 becomes 240. Then, if we set **dec** = 2 and **units** = 'kHz', this will display as 2.40 kHz.

width main
2.40 kHz

xCAT and EncoderCAT source code

The latest zipped source code is automatically loaded by xcatsetup.exe into folder: **C:\Program files (x86)\EncoderCat\xcat\encatcode**
It is also downloadable from the encat.g3vpx.net website. See [Downloads](#)

xCAT source code

xcat.exe is built using Borland Delphi 6.

Although of year 2000, Delphi 6 remains very effective and adequate for xCAT and generates 32 bit code which works well with Windows 10.

Low cost copies of Delphi 6 (or 7) may be available but you need other packages to build xCAT. It uses the Quantum Grid Suite and MemData from Developer Express and Raize components. Both of these were purchased by me in year 2000.

So, the source code is provided for information - but you cannot easily build it.

EncoderCAT source code

encat.hex is built using Atmel Studio 7 which you have installed to do the programming. So you can modify and rebuild the code.

Unzip the code (zipped encat folder) into C:\EncoderCAT ... or any other folder of your choice. Click on encat.atstn to launch in Atmel Studio 7 (which must be already installed !!)

encat.hex is generated into the ..\release subfolder

encat code uses **Atmel** supplied interrupt driven drivers for:

- I2C for OLED display **twi_master_driver.c** on TWIC (processor has two I2C ports)
- USART for RS232 port **usart_driver.c** on USARTD0 (in future for RPI with USARTF0 ?)
- SPI driver for the 0.5 Mbyte flash **eeeprom_driver.c** **enSPleeprom.c**

Note that the datasheet information on the processor is in two documents:

- **ATxmega192A3** and similar processors (specific to these devices)
- **Xmega A** datasheet - generic - has all the fine detail needed for coding.

EncoderCAT - JTAG programmer - minimising the cost

A **JTAG programmer** device is needed to program the ATxmega192A3 using Atmel Studio 7.

I have two Atmel JTAGICE USB programmers. These work fine but are no longer available.

Atmel's current programmers are available from Farnell, see <http://www.farnell.com/datasheets/1835727.pdf>

The **ATATMEL-ICE-BASIC** programmer / debugger has case and cables and costs £73.56 +VAT

A lower cost option is the **ATATMEL-ICE-PCBA** which costs £41.12 + VAT.
This is supplied as just the PCB (no cables, no case).

In order to be able to recommend a cheaper option for EncoderCAT, I purchased an ATATMEL-ICE-BASIC..
The PC USB cable is a standard micro USB cable as used on mobile phones etc.
I made up a programming cable as shown below.

The ATATMEL_ICE JTAG connector is a 1.27mm 2 x 5 way IDC header.

EncoderCAT's JTAG connector is a 2.54mm 2 x 5 way IDC header.

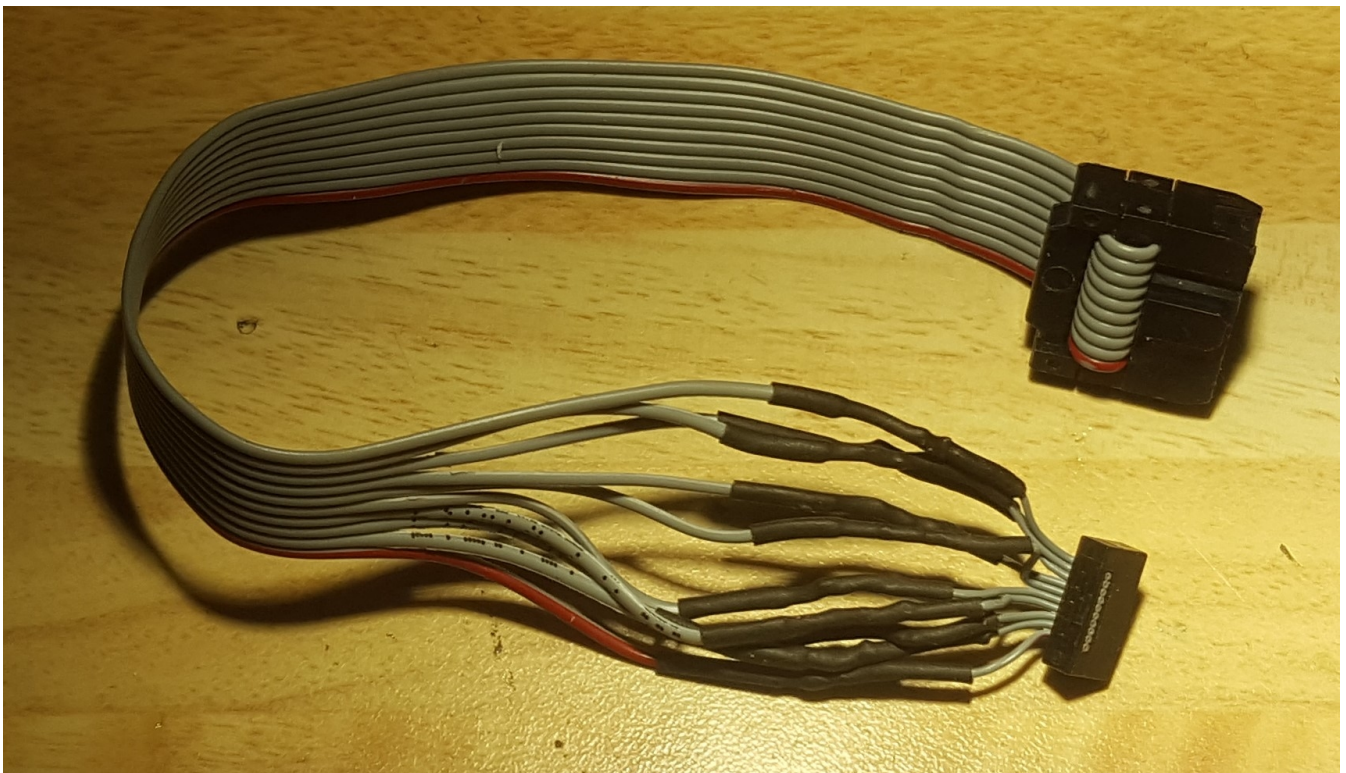
The two connectors have identical pin layouts.

I made up a cable as follows:

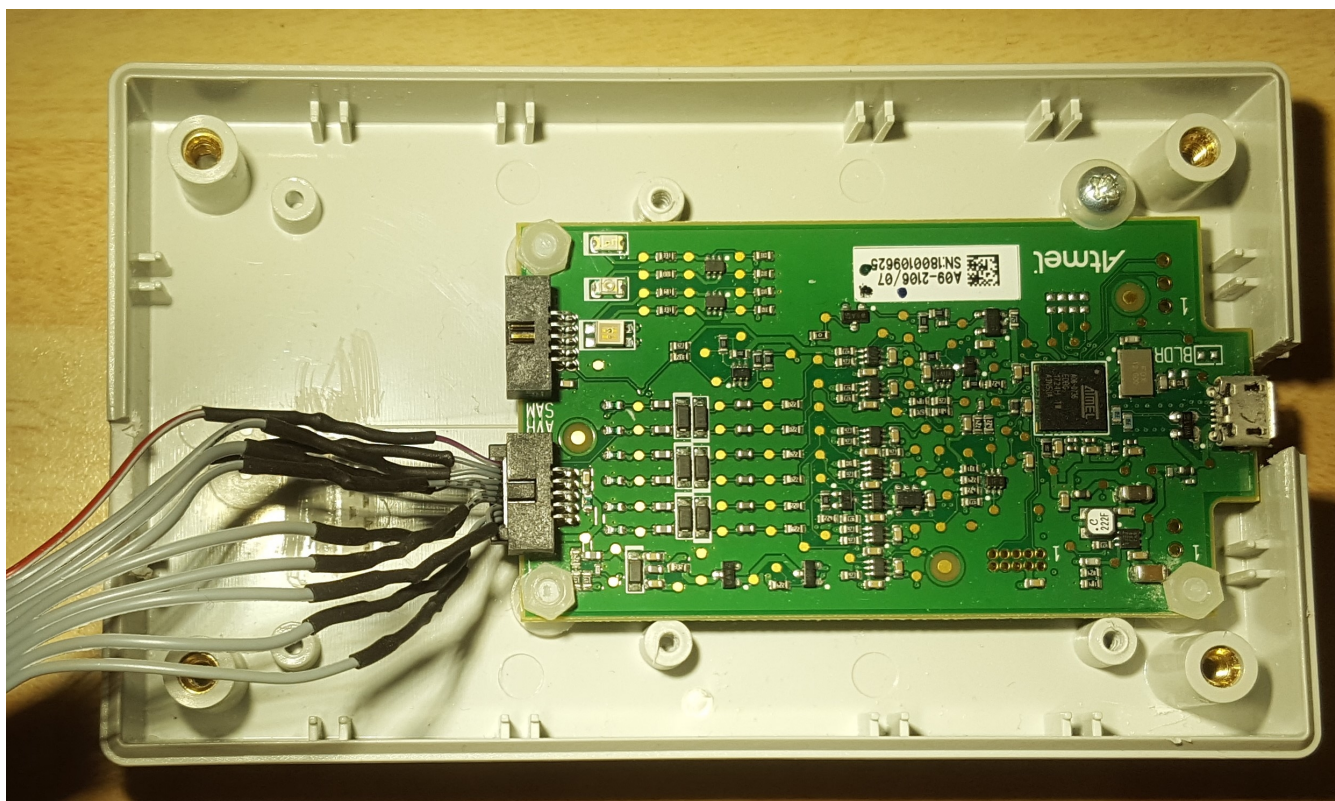
- install a 1.27mm connector onto a length of 10way 0.635mm IDC cable.
- Install a 2.54mm connector onto a length of 10way 1.27mm IDC cable.
- Connect the two together on a one to one basis. (Soldered and heatshrink sleeved)

This is shown below - and works fine.

I can supply short lengths of 0.635mm pitch cable with the PCB. (The only UK supply that I found was 30m reels)



The complete ATATMEL-ICE PCB in a box is shown below.
USB connection is a familiar micro USB B (and powers the unit)



EncoderCAT - PCB - Bill of materials

Component no	Description	Item	manufacturer	quant	Farnell #
EncoderCAT PCB					
U1	Processor	ATxmega192A3U-AU	Atmel	1	206-6303
	27.5.2020 G3VPX has 25. Farnell have 20 Mouser UK have 1500. Note that the 'U' in ATxmega192A3U is for USB interface on a serial port - we don't use this as USB.				
U2	parallel / USB interface	FT245RL	FTDI	1	114-6034
U3	3v3 LDO regulator	TLV1117-33CDCYR	Texas	1	300-7751
U4	Filter	STF202-22	ON Semiconductor	1	165-1958
U5	RS232 transceiver 3V0-5v5 supply	ADM3202ARNZ (narrow !)	Analog devices	1	133-3250
U6	512kbyte flash	AT45DB041E-SSHN-T	Adesto	1	290-9996
U7	PCA9600D dual bidiectioal buffer	PCA9600D	NXP	1	2400485
L1	USB power choke	ACM2520-102-2P	TDK	1	166-9299
J1, J2	10way header 2.54mm		Molex	2	275-1319
J3	8way header 2.54mm		Molex	1	973-1180
J4	JTAG connector 2x5 IDC boxed	2x5 IDC boxed	3M 30310-6002HB	1	211-2432
J5	Micro USB-B socket	FCI- 10103594 -0001LF	Amphenol FCI	1	229-3752
J6, J10	1x4way header 2.54mm	Molex 22-27-2041	Molex	2	973-1164
J7	D Sub 9pin Rt angled PCB connector	8LCM009P-30 1B-XX	Multicomp	1	984-6298
J8, J12	1x3 way header for jumper 2.54mm	Molex 90120-0763	Molex	2	973-3310
J9	1x2way header 2.54mm	Molex 22-11-2022	Molex	1	275-1315
J11	mono 3.5mm jack socket	Schurter 4832.222	Schurter	1	152.206
X1	crystal 16MHz	ABM8G-16.00 0MHZ-18-D2 Y-T	Abracon	1	246-7841
LED1	power indictor LED smd 2mA red	LED 0805 2mA red		1	852-9930
Q1	P enhancement Mosfet	FDN304P		1	235-2451
R1, R2, R22		resistor 0808 1k0		3	
R3, R4		resistor 0805 15k		2	
R6, R7, R8, R9, R10, R11, R12, R13		resistor 0805 330R		8	
R5, R14, R15, R16, R17, R18, R18, R20, R21		resistor 0805 22k		9	
R23, R24		resistor 0805 2k2		2	
R25, R26		resistor 0805 4k7		2	
C16, C20, C21, C22, C23, C24, C25, C26		capacitor 6.3v 10nF 0805		8	186-5556
C12, C13, C14, C15	capacitor 100nF 16v min 0805	capacitor 100nF 16v 0805		4	183-3888
C11, C12		capacitor cermic 18pF 0805		2	
C1, C2, C4, C6, C7, C8, C9, C17, C19, C27, C28, C30, C31, C32	capacior 1uF 0805 6.3v min	capacitor ceramic 1uF 16v		14	922-7792
C3, C5	capacitor 10uF 6v3 0805	capacitor 10uF 6v3 0805	15v - cheaper >	2	186-7958
C18 ,C29		capacitor ceramic 22uF 10v 1206		2	209-4045
JTAG programmer items	cables & connectors - alternatives				
JTAG 1	JTAG programmer	Atmel ICE-PCBA	Microchip	1	240-7171
IDC connector to ATAMET-ICE JTAG	2x5way 1.27mm IDC wire to board	M50-3300542	Harwin	1	228-9784
IDC connector to enCAT JTAG	2x5way 2.54mm IDC wire to board	0918 510 6813	Harting	1	110-6726
Ribbon cable	1.27mm pitch 10 way for IDC	3365-10	3M	1m	297-288
Ribbon cable (10cm with PCB)	0.635mm pitch 10 way for IDC	3756-10	3M		
	The 0.635 pitch ribbon cable is difficult buy in small quantities. G3VPX will supply with the PCB if required.				
	1.27mm connector - not stocked by Farnell.				
	RS Components: 909-4391 Harwin #M50-3300542. £7.50 for 5. stock 100. ? Buy and share with group if problem.				

Molex socket suggestions (for encoder cables etc) :

10 way: 22-01-3107 (Farnell 3107168) 8 way: 22-01-3087 (Farnell 2396394)
4way: 22-01-2041 (Farnell 2751622) 2 way: 22-01-3027 (Farnell 146 2825)
KK crimp contacts: 08-50-0113 (Farnell 2062734) ... I solder them and then shape with long nosed pliers.

EncoderCAT PCB - soldering SMD devices

You need:

- A **fine tipped** thermostatic soldering iron
- a **good** desoldering braid ... some are not very good. **Chem Wick** (eBay) is excellent.
- Rework Solder **Flux spray** from RS Components (or similar)
- A flux removal spray ... for when the board is completed. eg: Electrolube ULS (outdoors!!)
- A good magnifying glass.
- I use 60/40 lead/tin solder - its nicer to use than lead-free - and has a lower melting point.
(It doesn't comply with ROHS
... unless you interpret as **Real Old-fashioned Heavy-metal Solder !!**)

I first coat the whole component side of the board in the spray flux and let it dry.

Spray a small amount into a small container and apply to the areas for SMD soldering.
Place the PCB on a heating radiator to dry it until it becomes sticky.

Then when you place components on the board they stick to the sticky syrupy flux.
You can drag them around in a controlled manner.

I would suggest the following order of assembly:

1. Crystal (X1), choke (L1) and micro USB connector
2. Resistors, capacitors, LED, voltage regulator.
3. The the ICs
4. Finally the through-board mounted connectors.

Specific comments:

Crystal X1

This is possibly the most tricky component to solder.
It has contact pads underneath only.

My plan:

- Apply solder flux and then tin the four pads -
- remove excess solder ... but aim to leave a very thin blob of solder on each pad
(if it is not very thin then you will have different heights.)
- Apply a blob of solder flux and dry until sticky
- Position X1 on the pads ensuring that a thin area of pad is visible all round.
- Wipe the iron and then get a blob of fresh solder on the tip.
- Carefully hold X1 fixed with fine screwdriver tip and then apply the blob of solder to one pad holding the iron in place for 2-3 seconds.
- Check that X1 is now fixed by this pad. You now no longer need to hold it in position.
- Repeat the process on the other four corners.
- Remove excess with desoldering braid and inspect.

Choke (L1)

- similar to X1 but a bit easier as the contacts are at the side as well as underneath.

Micro USB connector

- This has four ground pads and 5 contacts.
- The good news is that it has four tiny lugs which locate it on the holes in the PCB.
- Fix it in position by soldering the grounded casing first.
 - .making sure that the five contacts are aligned with their PCB tracks
 - *Be very careful not allow solder to run inside the connector (or else the USB plug wont go in !)*
- Then solder the five contacts .. they are slightly recessed inside the casing
 - . but are easily soldered by poking the solder iron (with a solder blob) and then and stroking outwards.
- Then clean up with desoldering braid.
- Then inspect with a magnifier

ICs

The **ATxmega192A3** is the easiest because of relatively wide pin spacing. (0.8 mm)

Be careful with orientation

- It has a small location dot at pins 1 and 64 but may have a larger dot at other corners!!
- The pin 1 /64 corner of the device is angled.

All the ICs

- Use sticky solder flux. Place the device on the PCB.
 - If the flux has a treacle-like viscosity, then you can slowly drag the IC into position.
- Hold with a screwdriver tip and apply a solder blob (2-3 seconds) to 2 or 3 pins of one corner.
 - this will fix the device - check with a magnifier that it hasn't moved.
- If ok - then fix another corner with a solder blob.
- If ok then flood solder all sides starting away from the fixing points.
 - ... dont worry about the total short circuiting of all pins - the desolder braid will remove the excess.
 - .. You need to apply heat to all areas for 3 seconds or so to ensure good joints.
 - .. Gently stroke outwards with the fine soldering tip.
- Then lay clean desolder braid along one side at a time and apply the iron to soak up excess solder
 - inspect carefully - identify solder bridges and reapply the braid to remove them.

Be very careful when soldering and particularly when using the braid so that:

- ***You do not bend the pins by dragging them sideways.***
- ***You do no lift the pins by pulling off the braid when the solder has cooled to the pins***

I often finally neaten the pin joints by stroking them outward with a fine soldering iron tip with a tiny amount of solder.

(This may cause a recurrence of inter-pin solder bridges - which I can remove with the braid)

Construction notes

J6 - raspberry Pi connector

You may wish to omit connector J6 ... to avoid accidentally plugging the OLED display onto it !
(This would only be harmful if +5v was supplied from J9 rather than from USB)

Flash memory device: AT45DB041E-SSHN

This must be the ... SSHN version which has foot print SOIC8 narrow. (the wide ones don't fit!!)

Construction order - testing - suggestions

- Install the tricky components: 16Mhz xtal, choke, EMI filter, micro USB connector.
- Apply USB power - check for 0v / 5v at the output of the choke.
- Install all the capacitors.
- Install all the resistors.
- Install Q1 and the FT245RL device.
- Install the 3v3 regulator, U3 and LED1 (LED1 has a directional arrow on its underside)
- Jump J8 for USB power.
- Power up using the micro USB connector
 - LED should illuminate.
 - check for 3v3 on the output tab of the regulator.

You can now, if you wish configure the FT245RL's internal EEPROM using FTProg.
This includes entering an identifier string which include the letters 'ENCAT' or 'encat'.

Connect the USB to your PC and start xCAT.
xCAT should now detect and report your FT245RL on the left status bar.
(It will not report EncoderCAT version ... as you haven't installed the processor yet.)

- Now install the processor and all the other chips and the connectors.
- Power up again.
- Program the processor with encat.hex via JTAG.
- Connect to the PC - xCAT should now report the EncoderCAT version on the status bar.

Use xCAT to load one of my encoder configuration files from the PC disc.
If you have no OLED display connected, then uncheck the display box.
Check that you can write it to EncoderCAT and then read it back again.

Do the same for my corresponding Lookup file.

OLED display - Note that this is updated AFTER a successful transaction with the radio.
... if EncoderCAT is not talking to the radio, then the display will not change.

CI-V connector

The three holes in the PCB are a little larger than needed for the jack socket's pins.
Bend them to the edge of the holes.
Do not flood the holes with excessive solder. I shorted the signal pin by doing this.

Further development

EncoderCAT support group

EncoderCAT has been developed using only two radios:

- Yaesu FTdx101D
- Icom IC7000

We need a support /messaging group to deal with any problems arising from construction issues, radio compatibility and development ideas.

I have therefore created a support group: encodercat@groups.io

Click the link below to join or connect to the EncoderCAT support group.

[EncoderCAT support group link](#)

piWebCAT - CAT control from a web-browser

I have recently developed **piWebCAT** (May 2020)

piWebCAT provides CAT control from a web browser.

It uses a **Raspberry Pi 4B** computer with Apache web server, MYSQL database and PHP7 server programming. It works well on PC based browsers.

It is fine on my fast modern Android 7 tablet. (Touch tuning a little slower on an older tablet)

A piWebCAT RPi can optionally communicate with your radio via a 115 kbaud serial link to EncoderCAT. This allows simultaneous use of piWebCAT and EncoderCAT via your radio's single CAT connection. You may wish to leave space in your EncoderCAT box for a Raspberry Pi !

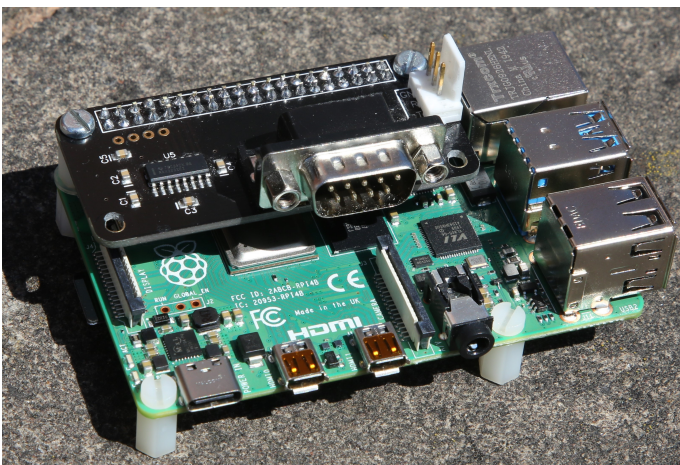


piWebCAT's control window. (The other windows are database editor and meter calibration)

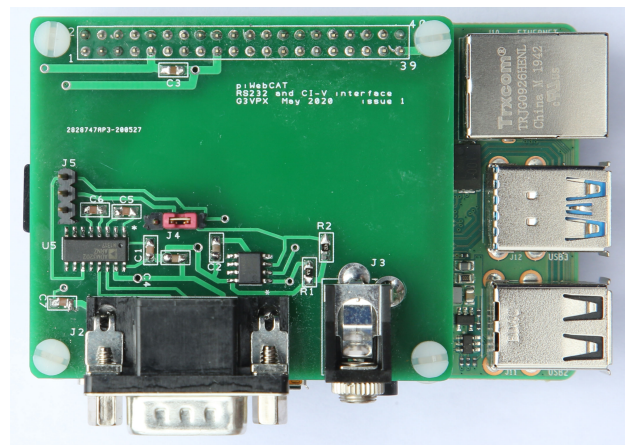
The footer bar shows interface = ENCAT baudrate = 115200.

This is because the RPi serial Tx and Rx on GPIO pins 8 and 10 can optionally connect through EncoderCAT to the radio.

piWebCAT hardware



RPi 4B web server + Serial piZero Rs232 card (£9)



RPi 4B web server + G3VPX piWebCAT PCB

This has RS232 and Icom CI-V connections.

The full width PCB brings connectors to the side of a box.

piWebCAT specification

- **Extensive, flexible user configurability.** Configuration stored on RPi server in a MYSQL database. The database can store configurations for multiple radios.
- **Hardware:** Raspberry Pi 4B computer + RS232 or RS232/CI-V GPIO card.
- Downloadable totally preconfigured **SD card image.** Almost 'plug and play' for a modern Yaesu radio.
- **Frequency control:** VFOA / VFOB, swap, B to A, split (or whatever else you want to configure.)
- **Tuning:** A dedicated tuning scale for each band with marker. Touch or click the scale for coarse positioning. A tuning window with three horizontal bands: fast, medium and slow. (user definable tuning rates)
 - tune by mouse (or finger) drag along these bands
 - OR three mouse wheel tuning rates according to which band the mouse pointer is in.
 - On an android tablet, tune by finger drag or by mouse wheel on bluetooth or OTG mouse.
- **Bands:** 160m, 80m, 60m, 40m, 30, 20m, 17m, 15m, 12m, 10m, 6m, 4m, 2m and 70cm.
- Tuning by **UP** and **DOWN** buttons for **VHF/UHF channels**, eg: +12.5kHz and -12.5kHz
 - This is one of the few fixed coded functions -
 - but you retain control of frequency shift, button choice and captions and colour.
- **Memories.** Button configured to popup a memory selector keypad. Memory to VFO CAT command sent.
- **27 slider controls** - some predefined + some spare for user allocation ... **but you can redefine them all.**
 - Each slider has an adjacent value display (with scaling, dec.point, optional table lookup and units)
 - Sliders are in three groups:
 - COMP, Mic Gain, Vox gain, RFpower, AF gain and squelch (probably unchanged) + 2 spare.
 - Nine sliders with a reset to default button - default defined by you in the database -
 - Nine sliders with an adjacent button: mainly on/off.
 - the association of button and slider is database defined and optional.
- **66 button controls:**
 - A few need to be of fixed function: eg: Band buttons and VFOA and VFOB.
 - Some are associated with sliders (eg: DNR on/off - DNR level.) but you can change what they all do.
 - Five are dedicated TX meter function selectors - but you choose which five Tx metering values to offer.
 - The rest are under user configuration control for:
 - the CAT commands that they generate.
 - their action: S = single momentary, T = toggling, G = grouped (share a common code) etc
 - their captions and the background colour of the button.
- **Default settings of controls:**
 - At start up, frequency and mode and the settings of all buttons and sliders are loaded from the radio.
 - Thereafter, only frequency, mode and MOX update automatically on piWebCAT, if changed on the radio.
 - A 'Reconnect' button is therefore provided to reload the settings.
 - However, this is only needed if changes are made on the radio:
 - If you use piWebCAT to change band or VFO on the radio, the radio responds immediately and piWebCAT's controls are updated automatically within 5 seconds. However, piWebCAT remembers the last button and slider values for each VFO. This avoids the reload delay when alternating VFO A/B. (piWebCAT does not use CI-V transceive)
- **S meter / Tx meters**
 - On receive this is an S meter with an appearance similar to the FTdx101D.
 - On transmit, the meter displays one of five button selectable options.
 - the Tx values on offer for display and their CAT controls are defined in the database for your radio.
 - I provide five Tx meter backgrounds for Power, ALC, Compression, PA ID and SWR.
 - These are simple 250 x 110 bitmap images ... you can change them (please keep the arc !!)
 - Accuracy of display is achieved for each of the six installed metering functions by the provision of 20 point interpolated calibration tables in the database (and an easy calibration procedure)
- **Database editing.**
 - piWebCAT has a spreadsheet-like editor grid for each of the eight database tables.
 - Editing is directly onto the grid. Only records for the selected radio are presented.
 - Drop down list selector are used whenever possible.
 - The database tables can be exported to a CSV file for import into Excel. They can then be printed.
 - The database is standard MYSQL (MariaDB) and so can be edited by other database tools.
 - I use the excellent, free MYSQL Front. This can export radio configurations as a backup of your settings and to facilitate the exchange of configuration data between users.

Design philosophy

I have attempted to maximise the use of a user editable database in configuring piWebCAT.

This means that there are very few controls tailored to the features of particular radios.

A example of an exception to this rule is **Rx clarifier on the Fdx101D**.

If we compare the FTdx101D **IF shift** and **clarifier** commands.

- **IF shift is not a problem.** A single read/write command is used (IS00-0340; for -340Hz with the current VFO.)
piWebCAT can display a centre-zero slider with text -340 Hz and a centre zeroing reset button.
- **Rx Clarifier is more difficult.** The FTdx101D has an on/off command and a clear command.
There are separate write-only commands for shift up increment and shift down decrement.
- these can be combined into a single control, but doing it via a database is potentially messy (not done ..yet!)

The project provides:

- A an SD card image download for the Raspberry Pi
- Extensive documentation in a website and as an 86 page PDF.
- My FTdx101D and IC7000 database configurations as examples and as templates.

Software

piWebCAT was built using **Microsoft's Expression 4** - a free downloadable web development package.

The design was done 100% in code. This kind of web page is too complex for WYSIWYG graphical layout.

You can use an FTP client program (eg: FileZilla) to download the complete piWebCAT code from the usual Linux `/var/www/html` folder on the RPi. You can then edit it and play with it using Expression 4.

You need to keep a copy of the original. Javascript and PHP code can be killed by a single wrong character which can be very difficult to locate across multiple coded sections!!

The piWebCAT data path is:

Javascript code in the web browser < LAN > PHP code in RPi webserver < RS232 serial > FTdx101D.

Note that the client javascript makes extensive use of **jQuery** (open source).

Communication with the server is by jQuery **Ajax** commands.

The editing grids use **phpGrid** - a purchased Chinese product that is licensed to me for OEM distribution.

phpGrid makes extensive use of javascript **jqGrid** which is open source.

Miscellaneous notes etc

xCAT / EncoderCAT notes

IC7000 baud rate

During development for Icom CI-V, when operating at 19200 baud, I noted that the oscilloscope was failing to properly decode the byte sequences from the IC7000. The outgoing bytes sequences from EncoderCAT were decoding ok.

The problem turned out to be that my IC7000 is transmitting at 18200 baud!!
- EncoderCAT copes with this but the scope flags errors.

This is an unexpected type of problem. One expects baudrate to be scaled from a crystal controlled processor clock.

I telephoned Icom uk.

The engineer was not aware of the problem and could not advise any cause nor cure.

When I explained that I had found internet discussions on IC7000 CAT failure / dropout, he conceded that it might be a feature.

I added 18200 baud at the bottom of the available baudrate list !!

EncoderCAT - Downloads

xcatsetup.exe is an installer script that will install / update and configure on your PC. The installer creates C:\Program Files (x86)\hbradios\xcat

This folder contains xcat.exe, xCAT.chm (help file) xCAT.pdf (pdf version of help) and also subfolders contain source code.

Download links:

xcatsetup.exe <http://encat.g3vpx.net/updater/xcatsetup.exe>

xcat source code <http://encat.g3vpx.net/files/xcat-source.zip>

EncoderCAT source code <http://encat.g3vpx.net/files/encat-source.zip>

EncoderCAT website as pdf <http://encat.g3vpx.net/files/encodercat.pdf>

EncoderCAT schematic <http://encat.g3vpx.net/files/encat-schematic.pdf>

EncoderCAT PCB layout <http://encat.g3vpx.net/files/encat-pcb.pdf>

EncoderCAT parts list <http://encat.g3vpx.net/files/encat-bom.pdf>

EncoderCAT hex and ini files <http://encat.g3vpx.net/files/encatfiles.zip>
(xcatsetup.exe also places these in C:\Program files (x86)\EncoderCAT\files)

email g3vpx [click to email](#)

xcatsetup.exe

xcatsetup.exe needs to be 'run as administrator' because it places files and folders in C:\Program files (x86).

('Run as administrator' is a file right mouse option in Window explorer)

Anti virus software

The software is virus checked with AVG before uploading. However, the running of **xcatsetup.exe** and **xcat.exe** may be delayed by your antivirus software. My PCs use AVG Internet Security etc.

When I first ran **xcat.exe** on a second machine, it initially failed to run properly and an error message box reporting unfound Windows Classes appeared. An AVG popup window indicated that this was because *xcat.exe was a 'rare' program*. AVG was clearly reacting because of no previous encounters with the program. The file appeared to be copied off-site to AVG for checking. After about 30 seconds, a message can back from AVG saying that there was no problem - the program was allowed to run.

This process might take longer with a slower broadband connection. (Mine is 400 Mbits/sec download and 30 Mbits/sec upload) If this happens recurrently, then you need to follow the instructions to have the program accepted as ok by your anti-virus software

11.06.2020 - Testing xcatsetup.exe... AVG anxious .. blocked ,,off- site analysis Ten minutes later a message was returned saying there was nothing wrong with it! - Ran ok. It's a big file - maybe AVG worries about the size. I removed the PDF (you can download it separately from this page) AVG was then happy with a 15 second local scan.

