OmniRig, Icom CI-V and the K9JM CI-V router

James Michener K9JM
June 13, 2013

Introduction

This paper will discuss the use of OmniRig operation with the Icom CI-V bus and with the K9JM CI-V Router.

Disclaimer:

The author is not expert of OmniRig, if there are any false or incorrect statements in this paper, corrections would be greatly appreciated.

What is OmniRig:

http://www.dxatlas.com/omnirig/

OmniRig is a module of code that provides an interface between a users program and a radio. It is what is called a Microsoft COM/OLE component

http://www.microsoft.com/com/default.mspx

OmniRig can be thought of as a library, that can be included as part of another program to control a radio.

If used within a program, it provides an common interface to any radio. The conversion from this common interface to the specific radio is done through OmniRig layer of code. The definition of how to translate commands and messages that are passed to the radio is user programmable, through a standard Windows ".ini" file. These files are contained in a directory called "Rigs", with one .ini file per rig type. This ".ini" file contains information regarding what messages should be generated to query or set states within a radio, as well as how to validate the message.

Important Concept:

As far as the radio interface is concerned, OmniRig is the master. OmniRig is suppose to be the only thing on the bus sending requests to the radio, and the radio should only speak when asked a question by OmniRig. Any other messages will do nothing but confuse OmniRig.

Unfortunately, this concept goes against the entire concept that Icom uses to "sync" the radio to other equipment, such as the PW-1 on the CI-V bus.

The Icom radio and other equipment (eg. Icom PW-1, many amplifiers,. StepIRs and autotuners that listen to the CI-V bus) use the Icom's "transceive" messages to keep themselves synced. Whenever the frequency, mode or bandwidth of the radio changes, a "transceive" message can be sent out by the transciever. This message is addressed to Icom ID = zero, which indicates that it is alike a "broadcast" message that can be picked up by anything on the bus that is interested. Unfortunately, the first thing that OmniRig does for Icom radios is to send a command to the radio that turns off "transceive" messages. Once these messages are disabled, devices that are looking for these messages will no long "follow" the transceiver.

The PW-1, when it doesn't get a "transceive" message within about 8 seconds, will generate it's own query of the Icom radio, this query may confuse OmniRig and can cause a collision on the buss. OmniRig is at least smart enough to validate each message so it is more likely to detect a bad message than most Icom radios.

The good news is that OmniRig behavior is programmable, and the initialization of the Icom radio so that "transceive" messages are turned off can be easily defeated.

K9JM CI-V router and OmniRig

With most OmniRig configurations, there are two options that should be set with the K9JM CI-V:

1.) Transceive msg to USB = false
2.) USB Echo = true

Transeive msg to USB, permits having "transceive" be on in the radio, without having the "transceive" messages be sent over the USB to the application program, in this case OmniRig. This feature was added to support N1MM, where the band map operation is improved if transceive messages are turned off.   This eanbles the CI-V bus and the K9JM CI-V router to be informed immediately when the frequency, mode or bandwidth is changed, while keeping the application program running on the PC not to be bothered by these message.

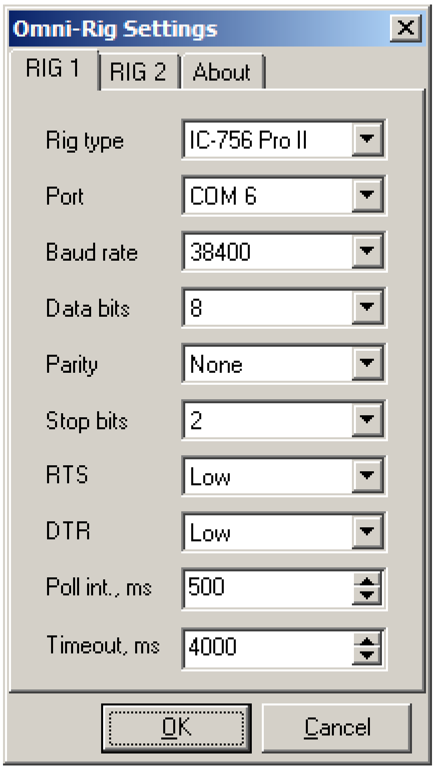
The USB Echo being true (on), causes any commands sent to the radio via the USB port to be echoed back to the computer. Most all of OmniRig validation checks for Icom radios include the initial command as part of the validation.   If one would want to operate the K9JM router with OmniRig, with USB Echo false (off), then the command must be removed from every validation entry in the OmniRig ".ini" file. While this can be done it is time consuming, it is easier to turn on USB Echo.

To learn how to change configuration of the K9JM CI-V router, read that configuration section of the user guide.

http://www.k9jm.com/CIV_Router/User%20Guide.pdf

The set up of OmniRig is more complex. First the basics. The set up of the basic program is done through a common GUI interface. Shown below is the dialog box for the author's set up.   Note: as with other programs, the COM port must match that for the CI-V router, and the baud rate is the USB rate as configured in the router. (See configuration in the User Guide)   The default is a baud rate of 38400. It is important that RTS and DTS are both low, as is 8 bit no parity and two stop bits. The polling rate and time out can be set as desired. Some discussion of these parameters will occur later in this document.

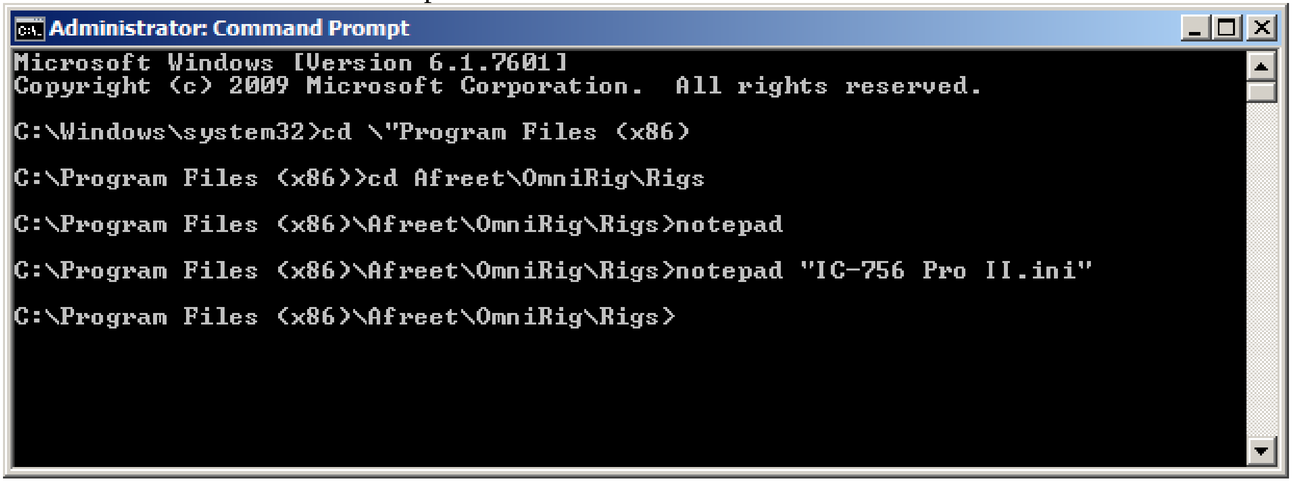
The selection of the *Rig Type* comes from a list of rigs that corresponds to all base file names in the dierctory called "Rigs".   OmniRig installs itself on C: drive in either the "Program Files" or the "Program Files (x86)" directories. In this directory is a sub directory called "Afreet" and in that directory is another directory called "OmniRig".   In this directory is the program as well as another directory called "Rigs".



Rig type: IC-756 Pro II
Port: COM 6
Baud rate: 38400
Data bits: 8
Parity: None
Stop bits: 2
RTS: Low
DTR: Low
Poll int., ms: 500
Timeout, ms: 4000

You will need to find the file in the "Rigs" directory that corresponds to your radio and edit it using your favorite Windows text editor. Notepad comes with Windows and it will work just fine.

If you are using Windows Vista, Windows 7 or Windows 8, contents of this directory is locked so only one with administrator privledge can edit the file. One way to acquire administrator privledge is to open a command prompt by right clicking on "Run as administrator". This will open an "Administrator: Command Prompt" window.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>cd \"Program Files (x86)

C:\Program Files (x86)>cd Afreet\OmniRig\Rigs

C:\Program Files (x86)\Afreet\OmniRig\Rigs>notepad

C:\Program Files (x86)\Afreet\OmniRig\Rigs>notepad "IC-756 Pro II.ini"

C:\Program Files (x86)\Afreet\OmniRig\Rigs>
```

Above is how I navigated on my machine to the proper .ini file.   Note the error I did when I ran notepad without an argument. In another window, Notepad opens and shows the the full .ini file. Take a look at the file. To help read the information

http://www.dxatlas.com/omnirig/IniStru.txt

The first few of my IC756ProII looks like:

```
;-------------------------------------------------------------------------------
;                     Icom IC-756 Pro II command set
;
;     File created by Alex Shovkoplyas, VE3NEA ve3nea@dxatlas.com
; Modified by Brendan EI6IZ ei6iz@oceanfree.net
;-------------------------------------------------------------------------------


;-------------------------------------------------------------------------------
;                                  initialize
;-------------------------------------------------------------------------------
[INIT1]
;disable transceive mode
Command=FEFE64E0.1A0542.00.FD
ReplyLength=15
Validate=FEFE64E01A054200FD.FEFEE064FBFD

[INIT2]
;disable 731 mode
Command=FEFE64E0.1A0543.00.FD
ReplyLength=15
Validate=FEFE64E01A054300FD.FEFEE064FBFD

; Some of us don't like this option one bit!
[INIT3]
;set CW normal to upper sideband
Command=FEFE64E0.1A0567.01.FD
ReplyLength=15
Validate=FEFE64E01A056701FD.FEFEE064FBFD
```

When the program initializes, it sends out INIT1 ... INITN, and the first command it sends out is one that disables transceive mode!   The quick and dirty way to fix this is change the command to one that enables transceive mode.

```
[INIT1]
;disable transceive mode
Command=FEFE64E0.1A0542.01.FD
ReplyLength=15
Validate=FEFE64E01A054200FD.FEFEE064FBFD
```

If you noticed, only one character was changed.

An alternate approach is to place a semicolon at the start of each line, or delete the entire section. In my opinion, all the INIT sections can be deleted.

With is set up, using the "Omni-Rig Client" from the dxatlas.com website seems to work.   Well mostly.   When you spin the knob, a flood of "transceive" messages are being generated, and OmniRig can't get a word in edgewise to the radio for some commands. It complains that the rig is not responding, and then a few seconds (Timeout, ms) after you stop turning the knob, it agains polls the radio and is successful, so all is happy.

This raises the question, what is it polling?   Looking at the read status section of the .ini file answers the question. (See the example below) The first two status commands are used to fetch the frequency and the mode.   In the K9JM CI-V router, these commands are not forwarded to the radio if the router already knows the anwer. The router will immediately answer the query regardless of the "transceive" traffic that is happening on the CI-V bus.   It are the commands for STATUS3 and STATUS4 that causes the time out. These commands are reading the CW pitch and is the transceiver is in transmit or not. The router does not cache these parameters, so it must query the radio to get the current response. So the question one can ask, does your application that uses OmniRig really need to know the CW pitch and if one is in transmit or not? If the answer to either of those are no, then just delete the STATUS3 and STATUS4 commands from the file.

Of course every Icom ".ini" file is different. Your mileage may vary.

```
;-------------------------------------------------------------------------------
;                               read status
;-------------------------------------------------------------------------------
[STATUS1]
Command=FEFE64E0.03.FD
ReplyLength=17
Validate=FEFE64E003FD.FEFEE064.03.0000000000.FD
Value1=11|5|vfBcdLU|1|0|pmFreq

[STATUS2]
Command=FEFE64E0.04.FD
ReplyLength=14
Validate=FEFE64E004FD.FEFEE064.04.0000.FD
;is filter byte appended to the mode byte?
Flag1=000000000000.0000000000.03.0000|pmCW_U
Flag2=000000000000.0000000000.07.0000|pmCW_L
Flag3=000000000000.0000000000.01.0000|pmSSB_U
Flag4=000000000000.0000000000.FF.0000|pmSSB_L
Flag5=000000000000.0000000000.04.0000|pmDIG_U
Flag6=000000000000.0000000000.08.0000|pmDIG_L
Flag7=000000000000.0000000000.02.0000|pmAM
Flag8=000000000000.0000000000.05.0000|pmFM

;we need an extended command to read the SSB D modes and this still doesn't tell
us which sideband,
; you have to read the Sideband THEN check to see What the 1A06 command returns
(0= normal SSB 1= D mode)
; how do we code for this? Does it even matter that we can't read this?

[STATUS3]
;need to read pitch offset in cw mode
;there's a fundamental problem here, anything between 0 and 9 = 300 10-19 =325
20-29 =350 etc
;how do we scale for this?
Command=FEFE64E0.1409.FD
ReplyLength=16
Validate=FEFE64E01409FD.FEFEE0641409.0000.FD
Value1=13|2|vfBcdBU|1|1|pmPitch

[STATUS4]
Command=FEFE64E0.1C00.FD
ReplyLength=15
Validate=FEFE64E01C00FD.FEFEE064.1C00.00.FD
Flag1=00000000000000.00000000.0000.01.00|pmTx
Flag2=00000000000000.00000000.0000.FF.00|00000000000000.00000000.0000.00.00|pmRx
```

It should also be noted that if USB Echo is turned off in the CI-V router, then the first part of each "Validate=" message must be removed from each "Validate=" line. That is quite a bit of work, and frankly isn't worth the effort in my opinion.

Feedback is welcomed. 73 de Jim K9JM