Table of Contents

Part I: Riding the CI-V Bus - Introduction

November 9, 2010
Larry, K8UT

For over a month I have been studying this PW1 and CI-V issue, but have not had time to complete all the work and get the results down on paper. The completion of this study has been delayed by the failure and repair of my own PW1 and the learning curve I've had to climb while trying to understand what's going on between PW1, radio, and logging software.

Understand that the CI-V protocol is a bus architecture, and thus is prone to collisions. However, before condemning the CI-V protocol as inherently evil, keep in mind that another popular protocol, Ethernet, is also a bus architecture and also prone to collisions. CI-V - again like Ethernet - defines collision declaration and recovery mechanisms within its protocol. Fortunately, the CI-V and Ethernet protocols work properly; unfortunately, the PW1 does not. The issue then becomes whether today's logging software developers can overcome the PW1's flaws and build systems that function safely and reliably. I hope that the information discovered in this study assists them toward that end. In the mean time, I also hope that the information discovered in this study assists Icom transceiver and PW1 owners in configuring their equipment so as to minimize risks and maximize their on-air operating experience.

Part II: Riding the CI-V Bus - FAQ

Frequently Asked Questions

1. Why perform this study of the PW1 amplifier and the CI-V Bus?

The author owns an Icom transceiver and a PW1 amplifier, and at times has observed discrepancies between his radio's band/frequency/mode display and that of his logging programs. He also experienced a recent PW1 failure as the result of "hot-switching" between bands while transmitting. This study is his attempt to better understand the CI-V bus, take whatever steps possible to avoid a future PW1 failure, and advise end-users and software developers of the nuances of operating a PW1 amplifier on the CI-V bus.

2. I do NOT own a PW1 amplifier, but control an Icom transceiver from a computer program. Should I worry?

You probably don't need to worry. I never witnessed collisions on a CI-V bus with just a PC and a radio (no PW1) when the radio was set for TRANSCEIVE=OFF. But CI-V bus collisions were observed on a simple, two-device bus with only a computer and an Icom radio configured for TRANSCEIVE=ON. At the very least, this study should explain why your radio sometimes does not obey commands from your logging program; or why your log sometimes records strange band and mode information from a QSO.

3. I own a PW1 amplifier, and control the amplifier and an Icom transceiver from a computer program. Should I worry?

Yes, you should worry. If you are not experiencing any of the symptoms described below, then consider yourself lucky and take the necessary steps to extend your luck. If you are experiencing the symptoms, you should take precautions to avoid damaging your amplifier.

4. I heard that the CI-V architecture is inherently broken and should be avoided at all costs. Is that true?

No. The CI-V bus architecture uses a scheme similar to Ethernet that allows multiple nodes to compete for access to the bus. The number of collisions on a CI-V bus is generally a small fraction of the total number of successful packets, and collision counts depend on the devices connected, software program used, polling rate, packet intervals, baud rates, and the operator's practices. Collisions do occur, but a reliable means of dealing with collisions is provided, and is implemented in Icom transceivers. However, the combination of potential CI-V collisions, coupled with the PW1's design flaws, requires special attention from Icom PW1 owners.

5. What is the primary issue with using the CI-V bus for PC-to-radio-to-PW1 communications?

The CI-V architecture is prone to collisions when two or more devices are connected and competing for access to the bus. Although Icom transceivers are designed to ignore and recover from collisions, the same is not true for the PW1 amplifier nor some logging programs.

6. What is likely to happen as a result of a collision on my CI-V bus? What's the worst thing that could happen?

In many megabytes of test logs, most collisions were innocuous and did not occur at times when they would have any impact on radio operations. The most frequent visible result is a temporary disagreement between the settings on your Icom radio and the display on your computer logging program. The most rare - but worst-case - result would be a failure of your PW1 amplifier's output transistors caused by band-switching while transmitting.

7. How would I know if I have a problem with my CI-V bus?

Several symptoms indicate when CI-V collisions are interfering with the communications between your computer, radio and PW1 amplifier: the frequency display in your logging program jumps erratically; clicking on packet spots sends your radio to strange bands/frequencies/modes; logging a QSO results in a record with the wrong band or mode; spinning the VFO across a band causes the relays in the PW1 to chatter.

8. Is there anything I can do to reduce, or eliminate, problems with my CI-V bus?

Yes, you can take steps that will reduce the CI-V collision risk to zero, although you may find the loss of automation unacceptable. The following are listed in the order of simple & obvious -to- drastic & shack-altering:

- Never change bands or spin your VFO dial while transmitting
- Never perform tasks in your logging program (packet clicks, band changes) that will change bands, frequencies or modes while transmitting
- Use the highest CI-V baud rate at which your equipment operates reliably
- Configure your radio for TRANSCEIVE=OFF (CAUTION! See Details about slow band changes)
- Operate your PW1 in manual (non-Auto) mode
- Unplug your PW1 from the CI-V bus
- Select a new logging program
- Buy a different amplifier

9. What products were involved in this CI-V bus study?

The Icom equipment included an Icom IC-7800, a 756 Pro 3 and two Icom PW1 amplifiers. Three logging programs were used in creating CI-V logs: N1MM Logger, DX4Win, and DXLab Commander. The CI-V bus connections were accomplished using a direct serial -to- serial cable between the computer and 7800; a direct remote -to- remote connection between 7800 and PW1; a Radio Shack USB -to- serial converter between the computer and the CI-V bus; and a homebrew serial -to- CI-V converter between the computer and the CI-V bus. The study and its logs were created using four different computers.

10. Which logging programs are better at avoiding, or eliminating, problems with the CI-V bus?

Of the three programs used in this study, DXLab Commander stands out for its two-tier approach to eliminating collisions on the CI-V bus. First, at the hardware layer, Commander creates separate networks for computer-to-PW1 and computer-to-radio communications, thus eliminating the PW1 collision risk. Second, at the software layer, Commander silences its normal software polling when receiving TRANSCEIVE=ON VFO packet floods from the radio (see Detail document for more about this) and allows for greater configurability by permitting user-adjustable polling and packet delivery confirmation.

11. Did theories and opinions about the PW1 and CI-V bus change much from the beginning of the study to the end of the study?

Huge! I began this study thinking there were no collisions; that problems were caused by serial port and UART overruns in the radio and PW1; that merely slowing polling rates and extending the intervals between commands would fix things; and that higher baud rates were more likely to cause problems than slower baud rates. I ended the study believing that it's all about collisions; that faster baud rates and shorter polling times have little effect on collision counts; that the ill-behaved PW1 can be tamed by understanding the CI-V protocol; and that a single network CI-V bus could function safely and reliably by applying new information emerging as a result of this work. This optimistic attitude on my part is purely the result of this study – a lot of programming and testing needs to be conducted before a logging program emerges to fit that description.

12. Could I duplicate K8UT's tests at my QTH?

Yes. Add another computer to your current CI-V bus by connecting a serial or USB CI-V adapter from that second computer. Download and Install Dxlab Commander software on the second computer, and configure Commander to be silent on the CI-V bus by disabling >Config "Continuous interrogation". Use Commander's <Msgs> feature to observe and capture the CI-V bus traffic between your radio, computer, and PW1.

13. Will the packet logs make any sense to me?

At first, the barrage of data will seem overwhelming. But after a while you will recognize simple patterns of packets repeated over-and-over. Look especially for packets containing the string "FC FC FC FC FC" – which is CI-V parlance for a collision. Refer to the lousy explanation of the CI-V protocol in the back of your Icom manual, or read a good primer on CI-V written by DF4OR at www.plicht.de/ekki/civ, or fully immerse yourself with the complete Icom CI-V reference manual at http://www.qsl.net/icom/download/civ3.zip.

14. Who assisted K8UT with the development of this material?

K8UT is very grateful for the assistance provided by Dave (AA6YQ) of DXLab, John (K3CT) of N1MM Logger, Paul (KK4HD) of DX4Win, and Adam Farson (VA7OJ/AB4OJ).

Part III: Riding the CI-V Bus - Details

PW1 and CI-V Details

Understand that the CI-V protocol is a bus architecture, and thus is prone to collisions. However, before condemning the CI-V protocol as inherently evil, keep in mind that another popular protocol, Ethernet, is also a bus architecture and also prone to collisions. CI-V - again like Ethernet - defines collision declaration and recovery mechanisms within its protocol. Fortunately, the CI-V and Ethernet protocols work properly; unfortunately, the PW1 does not. The issue then becomes whether today's logging software developers can overcome the PW1's flaws and build systems that function safely and reliably. I hope that the information discovered in this study assists them toward that end. In the mean time, I also hope that the information discovered in this study assists Icom transceiver and PW1 owners in configuring their equipment so as to minimize risks and maximize their on-air operating experience.

1. PW1 DEVICE ADDRESS

The PW1 amplifier's only means of communication is through a rear panel Remote connector, which exchanges CI-V packets with other devices on the bus - usually an Icom transceiver and a logging program. The CI-V protocol requires packets to include two addresses: the address of the destination device and the address of the sending device. Within the amplifier are two hex rotary switches that set its device address. The default device address is 54h, but by turning those switches you can select any address from 01 through 7F. Although the labeling of the switches suggest that you could set an address of 00 or an address higher than 7F, selecting an address of 00 sets the amp to 54h, and any setting higher than 7Fh will have the high bit stripped from the address (For example: 81 = 01, 91 = 21.). It is important to understand that this is the address the amp will use as its device address when it transmits packets. The amplifier's device address is not related to the device address that you set in your Icom transceiver, an address that the PW1 will learn when performing the Programming step listed on page 12 of the PW1 manual. For obvious reasons, you should never set your Icom transceiver device address the same as your PW1 device address. In practical operation, there is no reason to ever change the PW1's device address from its default value of 54h.

2. RADIO TO COMPUTER COMMUNICATION

Computer logging software polls the radio for frequency and mode information at regular intervals. In N1MM Logger, that rate is 4 times per second and is defined for each rig by the software developers. In DXLab Commander, that rate is 5 times per second but is user-configurable. In DX4Win, the rate is user-configurable but only down to 2 seconds (any setting less than 2000 ms still gives you 2 seconds). These logging software polls include requests for at least two pieces of information: the current frequency and the current mode/filter setting. Thus, under even the slowest of circumstances, no more than 2 seconds transpire before your radio responds to a poll and tells your logging software the frequency and mode.

3. RADIO TO PW1 COMMUNICATION

That's good for the radio and the computer, but how does the PW1 learn about band and frequency? The PW1 gets its updates one of two ways: by responding to a broadcast Set Frequency message sent from a radio with TRANSCEIVE=ON; or by polling the radio with one of two polling methods - an Asynchronous 10 Second Poll or an Event-Based Poll. Unfortunately, all three of these possess the potential to cause collisions on a CI-V bus.

3.1. TRANSCEIVE = ON or OFF? Trouble either way!

The CI-V protocol defines special address 00 as a broadcast message address, and special commands 00h as the SET Frequency command and 01h as the Set Mode / Filter command. When an Icom radio is configured for TRANSCEIVE=ON, it sends an address 00 broadcast message when one of the radio's front panel buttons changes the band, frequency, mode or filter. The PW1 hears the broadcast message and immediately changes bands as necessary. This communication method works fine until you spin the VFO dial across a band, at which time the Icom radio floods the CI-V bus with broadcast Set Frequency messages. This message flood may collide with other traffic and cause collisions on the CI-V bus, resulting in the PW1 mis-interpreting the Set Frequency message, and causing the PW1's relays to chatter as the PW1 switches between erroneous band changes. The sound of the relays chattering in your PW1 is unsettling, and indicates a communication failure caused by the PW1's inability to detect and ignore collisions. However, as long as you don't put your rig in transmit mode while spinning your VFO (!) no harm will come from this. (well, perhaps it's a little tough on the relays ;-)

What happens if you set TRANSCEIVE=OFF to avoid collisions? TRANSCEIVE=OFF operation branches into two scenarios – one with changes initiated from the logging software, the other with changes initiated from the radio.

A change that begins in your logging software will send an update command to the radio, which will trigger an "FB" PW1 Event-Based Poll, which will keep the PW1 synchronized (See Polling Methods, below). This synchronization should occur quickly – within one polling interval of your logging software. (See earlier description of logging polling times)

However, with TRANSCEIVE=OFF, a change initiated from the radio (pushing a mode, filter, band button) does not trigger any poll from the PW1. The PW1 will not learn about a radio-initiated change until its next Asynchronous 10 Second Poll. This means that as many as ten seconds could elapse during which the PW1 remains on the previous band when you might begin to transmit on the radio's new band. The PW1's protection circuits will detect this cross-band situation and your PW1 will go off-line on "Amp/Protect". It's kinda' like creating a short circuit to see if a circuit breaker is going to work. Safe, but still not something you like to see happen.

4. PW1 POLLING METHODS

Aside from special address 00 broadcast messages, there are no CI-V commands that directly instruct the PW1 to change bands. Instead, the PW1 learns about band, frequency, and mode changes by sending interrogatory polls to the radio and acting upon the radio's reply. The PW1 has two polling methods: an Asynchronous 10 Second Polling method; and an Event-Based Polling method.

4.1. The Asynchronous 10 Second Polling Method

Within the PW1 is a timer that, in the absence of inhibiting messages (more on inhibiting messages later), triggers the PW1 to poll the radio every 10 seconds. This poll contains two CI-V instructions: a request for VFO frequency (03h) and a request for mode and filter settings (04h). The radio replies to each instruction with the current information, and the PW1 changes bands accordingly.

This Asynchronous 10 Second Poll is like the Energizer Bunny – is just keeps polling every ten seconds, regardless of competing traffic and the collisions it may cause on the CI-V bus. There is nothing the end-user can do to stop it – well, except unplug the CI-V cable. The PW1 polls under the following circumstances – not all of which are good ideas:

- Asynchronous poll when PW1 is in Auto mode. Which it should. PW1 reacting to radio changes

- Asynchronous poll when PW1 is in "manual" (non-Auto) mode. Which it shouldn't. You're running the PW1 manually from front-panel band switches. Why poll? Do you want the amplifier to react if it learns something from the radio?

- Asynchronous poll when the PW1 is turned off. OK, this makes sense. If you're only using the PW1 for its tuner and the amplifier is shut off, you'd want its tuner to follow band changes

- Asynchronous poll when the PW1 is in transmit mode. This is the spooky part! You're transmitting. The PW1 polls the radio. Should the amplifier react based on the radio's reply? If the PW1 Asynchronous 10 Second Poll creates a collision and there are fragmented packets on the bus, could the PW1 mis-interpret the data and switch bands while you transmit? Not good.

The only way to inhibit the Asynchronous 10 Second Energizer Bunny is through the CI-V bus – by sending an address 00 broadcast message with either 00h Set Frequency or 01h Set Mode/Filter commands. The PW1 will react to that broadcast message, change bands if necessary, and reset its Asynchronous 10 Second Poll timer to zero seconds. If the PW1 receives a constant dose of these broadcast messages, the PW1's Asynchronous 10 Second Poll will remain silent until ten seconds after it receives the last broadcast message. A constant dose of broadcast messages could come from the radio (such as when spinning the VFO with TRANSCEIVE=ON), or from a computer logging program as part of its regular computer-to-radio polling. Including a broadcast message in the logging software's regular poll could be a creative method for logging programs to silence the PW1 Asynchronous 10 Second Poll traffic and reduce the risk of collisions on the CI-V bus.

4.2. The Event-Based Polling Method

When you click on a packet spot, or select a new band in a bandmap window, or invoke a macro that selects a different VFO, your logging program sends a message to the radio requesting that the radio change something (band, frequency, mode, filter, VFO, split/no split). The radio acknowledges receipt of that request with the CI-V ACK reply (FBh). The PW1, eavesdropping on the CI-V bus, sees that "FB" traffic and – suspecting that the radio might have changed something – reacts by issuing an Event-Based Poll to the radio. The contents of the Event-Based Poll are the same as the Asynchronous 10 Second Poll: a request for VFO frequency (03h) and a request for mode / filter settings (04h). The radio replies with the current information, and the PW1 changes its settings accordingly. Thus, the PW1 remains synchronized with the radio by listening for computer-to-radio traffic that may require the PW1 to change frequencies.

Like the Asynchronous 10 Second Poll, the Event-Based Poll transmits on the CI-V bus regardless of other traffic and the collisions it may cause. And, like the collision problem caused by spinning your VFO with TRANSCEIVE=ON, the Event-Based Poll poses a collision hazard by generating a short blast of PW1 traffic while the radio and computer may still be exchanging messages. However, unlike the Asynchronous 10 Second Poll, the Event-Based Poll appears to be predictable. Your logging software initiated the traffic that caused the Event-Based Poll, and that same logging software should be able to avoid colliding with the subsequent exchange between the radio and the PW1. Although not a perfect scenario, this new understanding of the behavior of the radio and the PW1 gives software developers the opportunity to anticipate CI-V bus traffic and insert timing delays to avoid collisions when Event-Based Polls occur.

5. PW1 DESIGN FLAWS

During the course of this study I’ve observed three PW1 design flaws: two serious and one minor. The first serious flaw involves hot-switching bands while in transmit. If a command to change bands arrives on the CI-V bus while the amplifier is transmitting, the amp will obey that command and hot-switch to that new band. Depending on several variables (power level, antenna selection, tuner position, SWR of new band.) that live transfer between bands may be fatal - causing an amplifier failure that could destroy its output transistors. In my PW1’s failure case, I inadvertently pushed a button that told the PW1 to change bands. Foolish move; self-inflicted; my fault. But, Icom is also at fault for producing a product that would allow me to pull such a stunt. How difficult would it have been for the PW1 designers to practice defensive programming by inserting a few lines of code inhibiting CI-V communication and preventing band-changes while transmitting? This design flaw, and the risk of CI-V collisions described previously, should compel Icom PW1 owners to be extra cautious in the configuration and operation of their amplifiers. However, lesson learned: be careful when transmitting. Do not press any buttons on the radio or make any changes to your logging software that will direct the PW1 to change bands while transmitting.

The second serious flaw involves the PW1’s inability to detect and ignore CI-V bus collisions. Evidence of this flaw is the relay chatter heard while spinning the VFO in TRANSCEIVE=ON mode. This study has uncovered some creative logging program techniques that could inhibit the traffic that causes those collisions, but we shouldn’t need to retrofit in logging software for defects in amplifier hardware.

The third flaw - a minor one - is the aggressive nature of the PW1’s Asynchronous 10 Second Poll. As per the examples of the Energizer Bunny cited earlier, there are two circumstances – the amp in manual mode and while transmitting - when polling is unnecessary and any subsequent poll results should be ignored by the amplifier. So, why contribute to possible collisions – which the PW1 cannot handle - by sending unnecessary traffic on the CI-V bus?

6. DISCLAIMERS

The information and opinions emerging from this study are mine and are intended to assist end-users and software developers in improving the fidelity of their interactions with Icom radios and the PW1 amplifier. However my operating habits, equipment, network, and test results may differ from those at your QTH. If you control an Icom transceiver and PW1 amplifier with a computer logging program, be careful with whatever configuration you elect to use.

This study was not intended to serve as a “Consumer Reports” software comparison nor to recommend any product over another. I studied these programs because they are the three logging programs that I use. I chose to highlight the DXLab Commander program in one paragraph because its developer has taken a creative approach to solving the PW1’s CI-V collision problem – with good results.

This study is the result of my on-air experience with Icom transceivers and the PW1 amplifier, plus empirical data gained by tapping the CI-V bus. There has been no communication with Icom employees nor access to Icom engineering information, other than standard product user manuals. As such, the results of this study may contain errors due to incorrect observation and interpretation.

This study should also not be construed as praise nor condemnation of the CI-V bus and the PW1 amplifier. My PW1 has served me well for 8 years and 30,000 QSOs, and I use it daily with a typical single-bus configuration. My self-inflicted PW1 failure was not caused by any of these software packages, and the damage was repaired by replacing one diode and one integrated circuit.

Part IV: Riding the CI-V Bus - Solutions

Dateline: April 2012

Good News! PW1 owners now have a choice of several solutions to overcome problems with the Icom amplifier. Although they all attack the same PW1 deficiencies, no two products are alike. This diversity gives PW1 owners a wide range of choices.

Two of the choices are hardware products; three are software programs. The following descriptions should assist PW1 owners in making a decision about which solution best meets their requirements.

HARDWARE

microHAM Station Master (and other products) - supported in the USA by Joe, W4TV

For many years microHAM have provided solutions which allow the use of automatically tuned amplifiers made by one manufacturer with transceivers by other manufacturers. This experience has been leveraged to bring support for the PW-1 to microHAM microKEYER II, Digikeyer II and MK2R/MK2R+ as well as microHAM Station Master and Station Master Deluxe. All of these products provide an optional, isolated CI-V bus emulating an Icom transceiver that will provide frequency control over a PW-1 (or any other accessory using CI-V protocol). When properly configured, all of the microHAM solutions function even if the computer is not running. All of the interfaces poll any connected transceiver for operating frequency.

website: http://microham-usa.com/store/cmspage.php?page_id=27

Arduino Based Icom CI-V Router - designed by Jim, K9JM

The K9JM Arduino based CI-V Router is a small, low cost hardware device that provides a high speed USB interface to the host computer and three isolated CI-V ports. Functionally, it is an intelligent CT-17 replacement. Message routing eliminates collision. The device operates independent of the host computer, host software and works with PC, Linux or Mac computers. Relay bypass permits CI-V communications even when the router is powered down. The device is an open source hardware and software solution that permits easy customization to provide band decoding, internet connectivity and other features. Full source code is available. Available as a full device or as parts for customization.

website: http://www.k9jm.com/CIV_Router/CI-V%20Router.html

SOFTWARE

DXLab Commander - written by Dave, AA6YQ

Commander is the transceiver control component of the full-featured, freeware *DXLab Suite*. When configured as described in http://www.dxlabsuite.com/dxlabwiki/PW1Control Commander prevents your PW-1 from encountering CI-V bus collisions.

website: http://www.dxlabsuite.com/

Data Decoder Utility (DDUtil) - written by Steve, K5FR

This freeware program was written to allow users of Flex radios to connect peripheral equipment and Radio Control Programs (RCP's) to the radio in order to complete the operator experience and have a degree of station control not previously possible. See the link for PW1 connection and setup information.

website: http://k5fr.com/ddutilwiki/index.php?title=Setup#IC-PW1

CI-Vfilter Utility - written by Larry, K8UT

CI-Vfilter is a small freeware utility that splits the CI-V bus into two segments and controls the flow of traffic between a PC-to-Radio bus and a PC-to-PW1 bus. By isolating the Icom PW1 amplifier on its own bus, CI-Vfilter overcomes the inherent design flaws of the PW1 amplifier involving asynchronous polling and packet collisions. The software has been tested with DX Labs Commander, N1MM Logger, and DX4Win - but should work with any Windows logging software.

website: www.k8ut.com/tiki-index.php?page=CI-Vfilter+Utility+Documentation