RTCM Client

From "AllStarLink Wiki"

The Radio Thin Client Module (RTCM) is commercially available hardware for interfacing radios to an AllStarLink computer.

The Micro-Node RTCM and VOTER interfaces are typically used with AllStar in voting/simulcast applications. They MAY be used for ANY repeater interface application, through the chan_voter channel driver. The VOTER is the original through-hole board designed by Jim Dixon for this application. It is open-source, and the relevant Gerber files and BoM to build it are available. The Micro-Node Radio Thin Client Module (RTCM) is the commercial version of the VOTER. It uses surface mount parts (SMT), but is functionally equivalent to the original VOTER. In general, the two terms (RTCM/VOTER) are used interchangeably, as they operate the same, and use the same firmware (mostly, see below).

This page will highlight some of the operational concerns, quirks, bugs, and other items of interest that relate to these interfaces. Much of the information has been gleaned off the AllStar mail list, comes from personal experience, or comes from notes in the firmware source code.

Contents

Testing

RTCM's come preconfigured to login to voter-demo-allstarlink.org for out of the box testing.

Factory Reset

A factory reset can be performed by setting SW 1 down and then applying power to the RTCM. Wait for the green LED to stop blinking, power off and set SW 1 back up.

Firmware

The firmware in the RTCM and VOTER is the same, except it is compiled for the specific dsPIC that is installed in each. As such, the firmware is specific to the VOTER or the RTCM, and is NOT interchangeable (it won't boot in the wrong device). Firmware for the RTCM is denoted by "smt" in the filename.

The VOTER uses a dsPIC33FJ128GP802 and the RTCM uses a dsPIC33FJ128GP804.

There are **two parts** to the firmware, a bootloader, and then the actual firmware file. The bootloader starts when power is applied, and allows you to talk to the dsPIC and load new firmware files over ethernet. If the bootloader is not intercepted by the loading tool, it will continue to boot the current firmware file.

All new boards will need to have the bootloader installed first, followed by a firmware file. You **can** load a firmware file directly (.hex) in to the dsPIC, but then you will not have any of the bootloader remote loading features.

The current bootloader (.cof file) is available here: https://github.com/AllStarLink/voter/tree/master/voter-bootloader (the -smt file is for the RTCM). It needs to be loaded with a PICKit programmer.

Current firmware (.cry file) is available here: https://github.com/AllStarLink/voter/tree/master/board-firmware . They are loaded with the EBLEX C30 Programmer.

There are multiple "flavors" of firmware available. See below for further explanation of what all the options mean.

The DSPBEW versions have Jim's DSP/BEW feature enabled. Note that due to the size of this feature, the diagnostics menu is NOT available in this version.

The CHUCK versions have Chuck Henderson's RSSI and Squelch modifications enabled. This is probably the version that most user's will want to use in production.

Compiling Environment

If you look in the votersystem.pdf, you will find a procedure to modify and load the bootloader in to the dsPIC of a VOTER board.

Unfortunately, there is an important step **missing** in that procedure, which is covered below.

In addition, the old links for the MPLAB software are dead, so let's update this info and get you going.

Currently (Feb 2017), you can get the required software from:

> http://www.microchip.com/development-tools/downloads-archive

Download MPLAB IDE 32-bit Windows v8.66:

> http://ww1.microchip.com/downloads/en/DeviceDoc/MPLAB_IDE_v8_66.zip
> http://dvswitch.org/files/AllStarLink/Voter/MPLAB_IDE_v8_66.zip

Download MPLAB C Compiler for PIC24 and dsPIC DSCs v3.31 **NOT v.3.25**:

> http://ww1.microchip.com/downloads/en/DeviceDoc/mplabc30-v3_31-windows-installer.exe
> http://dvswitch.org/files/AllStarLink/Voter/mplabc30-v3_31-windows-installer.exe

Optionally, install Windows Virtual PC and XP Mode. This is getting to be pretty old software, so running it under XP Mode may be a good idea, so we can keep it isolated (install it in a virtual machine). It does run fine in Windows 7.

To setup the compile/build environment, follow these steps:

- Run the MPLAB IDE installer. You don't need to install the HI-TECH C Compiler at the end (click no)
- Run the MPLAB C Compiler installer
- Select Legacy Directory Name
- **Select Lite Compiler**
- Go to: https://github.com/AllStarLink
- Follow the links to: voter --> Clone or Download --> Download Zip. That will get you voter-master.zip which is a download of the whole VOTER tree from GitHub.
- Extract it somewhere (ie. in the XP Mode Virtual PC)
- Launch the MPLAB IDE
- Go to Configure --> Settings --> Projects and de-select one-to-one project mode.

Bootloader

If you need to load the bootloader in to a fresh board, you will need to follow these steps:

- Go to Project --> Open --> voter-bootloader.mcp --> Open (it is in the voter-bootloader folder of the GitHub source)
- Go to File --> Import --> voter-bootloader --> ENC_C30.cof --> Open **This step is missing from the original procedure.**
- Remove JP7 on the VOTER Board. This is necessary to allow programming by the PICKit2/PICKit3 device.

- Attach the PICKIT2/PICKit3 device to J1 on the VOTER board. Note that Pin 1 is closest to the power supply modules (as indicated on the board).
- If you have not already selected a programming device, go to Programmer --> Select Device and choose PICKit3 (or PICKit2, depending on what you are using).
- Go to Programmer --> Program. This will program the bootloader firmware into the PIC device on the board.

If you want to change the default IP address from 192.168.1.11 in the bootloader:

- Select View --> Program Memory (from the top menu bar)
- Hit Control-F (to "find") and search for the digits "00A8C0".
- These should be found at memory address "03018".

The "A8C0" at 03018 represents the hex digits C0 (192) and A8 (168) which are the first two octets of the IP address. The six digits to enter are 00 then the SECOND octet of the IP address in hex then the FIRST octet of the IP address in hex.

The "0B01" at 0301A represents the hex digits 0B (11) and 01 (1) which are the second two octets of the IP address. The six digits to enter are 00 then the FOURTH octet of the IP address in hex then the THIRD octet of the IP address in hex.

Once you have modified the address to your desired IP, follow the programming procedure (above).

Firmware

To compile the firmware (if you want to make custom changes):

- Go to Project --> Open --> navigate to board-firmware and open the .mcp file for the flavor of firmware you want to compile. They are in the board-firmware folder of the GitHub source.
- **NOTE**: the .mcp files with the "smt" suffix are for the RTCM (built with SMT parts). The non-smt files are for the ORIGINAL through-hole VOTER boards. The difference is that the VOTER uses a dsPIC33FJ128GP802 and the RTCM uses a dsPIC33FJ128GP804.
- Go to Project --> Build Configuration and select "Release". This may not be necessary (I don't believe that option is used in the firmware), but it removes the compiler option of __DEBUG being passed, so theoretically it would build "normal" firmware.
- Go to Configure --> Select Device and choose the appropriate device for your board from the Device list. If you don't select the right one, the board will not boot. Select dsPIC33FJ128GP802 for the VOTER, and dsPIC33FJ128GP804 for the RTCM.
- Now, if you go to Project --> Build All it should compile everything and show you "Build Succeeded".

A .cry file should be in the board-firmware folder. You can load this with the ENC Loader.

There is also a .hex file in there that you could load with a programmer... but that would wipe the bootloader... so don't do that.

Chuck Squelch

If you want to enable "Chuck Squelch", open the HardwareProfile.h and un-comment #define CHUCK.

You may also want to go down to Line 291 in Voter.c (right click on the window, go to Properties --> "C" File Types --> Line Numbers) and tack on CHUCK after 1.60 (the current version number) so that when you load this firmware, the version will be shown as 1.51CHUCK, and you'll know that Chuck Squelch is compiled in. We should probably make that more automagic in the future...

DSP/BEW

If you want to compile the DSPBEW version, open the DSPBEW project file instead.

DSP/BEW Firmware Version

DSP BEW Firmware is mutually exclusive with the diagnostic menu. There is not enough space for both, if you load the DSP/BEW firmware, you will **NOT** have a diag menu.

BEW stands for Baseband Examination Window.

Typically, the discriminator of an FM communications receiver produces results containing audio spectrum from the "sub-audible" range (typically < 100 Hz) to well above frequencies able to be produced by modulating audio. These higher frequencies can be utilized to determine signal quality, since they can only contain noise (or no noise, if a sufficiently strong signal is present).

For receivers (such as the Motorola Quantar, etc) that do not provide sufficient spectral content at these "noise" frequencies (for various reasons), The "DSP/BEW (Digital Signal Processor / Baseband Examination Window)" feature of the RTCM firmware may be utilized.

These receivers are perfectly capable of providing valid "noise" signal with no modulation on the input of the receiver, but with strong modulation (high frequency audio and high deviation), it severely interferes with proper analysis of signal strength.

This feature provides a means by which a "Window" of baseband (normal audio range) signal is examined by a DSP and a determination of whether or not sufficient audio is present to cause interference of proper signal strength is made. During the VERY brief periods of time when it is determined that sufficient audio is present to cause interference, the signal strength value is "held" (the last valid value previous to the time of interference) until such time that the interfering audio is no longer present.

The DSP/BEW feature is selectable, and **should not** be used for a receiver that does not need it.

A note on the Motorola SLR5700 per VE7FET. DSP/BEW definitely makes a difference on what AllMon reports for the signal strength of received signals. It is less reactive when DSP/BEW = 1 than with it set to 0. We'll have to see what the real world trials show. It might be better to leave it off.

Chuck Squelch

"Chuck Squelch" are a couple firmware changes made by Chuck Henderson, WB9UUS.

Pre-compiled firmware versions including this option are available on GitHub. See above on how it is enabled/compiled, if you are rolling your own firmware modifications.

One of the changes fixes an issue with weak signals producing RSSI readings all over the place. It is caused by a 16 bit value that was overflowing (it is the RSSI change in the firmware). It results in rock-solid RSSI values being reported, even on barely or non-readable signals. **This change will likely be rolled in to a permanent fix in a future firmware release.**

The other firmware change changes how the squelch responds (looks at the noise in the last two audio samples) and makes the "Micor squelch" action work better.

You may also want to consider the following changes in /etc/asterisk/voter.conf:

```
;Comment out:
;thresholds =

;and set:
linger = 0
```

Squelch Issues

- If anyone is off frequency a little bit, that will make the voice talk off worse. Double check that the repeater and the users are all on frequency.
- Don't use narrow bandwidth on the repeater receiver.
- Make sure that the discriminator audio is not rolled off even a little bit at the high end. There should not be resistors in series or capacitors to ground between the discriminator chip output pin and the voter board input, for best results.

Motorola Quantar

Some things to consider:

- Install JP1 on the Quantar's RTCM. The squelch should calibrate at around 4 blinks rather than the 12 blinks or so without JP1.
- Be sure you've done the diode and squelch calibration with the actual attached radio (no antenna).
- The Quantar firmware should be 20.14.48 as later versions have better noise output.
- Try the "Chuck Squelch" RTCM version.
- Each RTCM should have 3 to 5 turns past threshold to prevent the squelch form being too loose. Somewhere around the 350 level seems about right. Chuck Squelch seems to need fewer turns but YMMV.
- Don't "and" CTCSS with squelch. That may override the RTCM's squelch detection. Compare with CTCSS on and off to see effect if any.

RTCM/VOTER LED's

RX LED on the RTCM/VOTER will flash (same rate as ACT LED) if you have External CTCSS enabled, and the received signal has the wrong (or no) valid PL.

Menus

Main Menu

This is what a typical Main Menu looks like:

```
Select the following values to View/Modify:

1  - Serial # (0000) (which is MAC ADDR 00:04:A3:00:00:00)
2  - VOTER Server Address (FQDN) (44.128.1.1)
3  - VOTER Server Port (667),  4  - Local Port (Override) (0)
5  - Client Password (password1),  6  - Host Password (hostpass1)
7  - Tx Buffer Length (800)
8  - GPS Data Protocol (0=NMEA, 1=TSIP) (1)
9  - GPS Serial Polarity (0=Non-Inverted, 1=Inverted) (1)
10 - GPS PPS Polarity (0=Non-Inverted, 1=Inverted, 2=NONE) (0)
11 - GPS Baud Rate (9600)
12 - External CTCSS (0=Ignore, 1=Non-Inverted, 2=Inverted) (0)
13 - COR Type (0=Normal, 1=IGNORE COR, 2=No Receiver) (0)
14 - Debug Level (0)
15  - Alt. VOTER Server Address (FQDN) ()
16  - Alt. VOTER Server Port (Override) (0)
17  - DSP/BEW Mode (0)
18 - "Duplex Mode 3" (0=DISABLED, 1-255 Hang Time) (1/10 secs) (0)
19 - Simulcast Launch Delay (0) (approx 200 ns, 5 = 1us, > 0 to ENA SC)
97 - RX Level,  98 - Status,  99 - Save Values to EEPROM
i - IP Parameters menu, o - Offline Mode Parameters menu
q - Disconnect Remote Console Session, r - reboot system, d - diagnostics

Enter Selection (1-27,97-99,r,q,d) :
```

IP Parameters Menu

This is what a typical IP Parameters Menu looks like:

```
IP Parameters Menu

Select the following values to View/Modify:

1  - (Static) IP Address (44.128.1.2)
2  - (Static) Netmask (255.255.255.0)
3  - (Static) Gateway (44.128.1.254)
4  - (Static) Primary DNS Server (44.128.1.254)
5  - (Static) Secondary DNS Server (0.0.0.0)
6  - DHCP Enable (0)
7  - Telnet Port (23)
8  - Telnet Username (admin)
9  - Telnet Password (radios)
10 - DynDNS Enable (0)
11 - DynDNS Username (wb6nil)
12 - DynDNS Password (radios42)
13 - DynDNS Host (voter-test.dyndns.org)
14 - BootLoader IP Address (192.168.1.10) (OK)
15 - Ethernet Duplex (0=Half, 1=Full) (1)
99 - Save Values to EEPROM
x  - Exit IP Parameters Menu (back to main menu)
q  - Disconnect Remote Console Session, r - reboot system

Enter Selection (1-14,99,c,x,q,r) :
```

Offline Parameters Menu

This is what a typical Offline Parameters Menu looks like:

```
OffLine Mode Parameters Menu

Select the following values to View/Modify:

1  - Offline Mode (0=NONE, 1=Simplex, 2=Simplex w/Trigger, 3=Repeater) (3)
2  - CW Speed (400) (1/8000 secs)
3  - Pre-CW Delay (4000) (1/8000 secs)
4  - Post-CW Delay (4000) (1/8000 secs)
5  - CW "Offline" (ID) String (WB6NIL)
6  - CW "Online" String (OK)
7  - "Offline" (CW ID) Period Time (6000) (1/10 secs)
8  - Offline Repeat Hang Time (15) (1/10 secs)
9  - Offline CTCSS Tone (0.0) Hz
10 - Offline CTCSS Level (0-32767) (3000)
11 - Offline De-Emphasis Override (0=NORMAL, 1=OVERRIDE) (0)
99 - Save Values to EEPROM
x  - Exit OffLine Mode Parameter Menu (back to main menu)
q  - Disconnect Remote Console Session, r - reboot system

Enter Selection (1-9,99,c,x,q,r) :
```

Configuration Options

Main Menu Options

Serial

Sets the serial number for the RTCM/VOTER.

- This sets the MAC address of the unit.
- Each unit must have a unique serial number on your network segment.

VOTER Server Addr ess (FQDN)

This is the DNS name or IP address of the Asterisk server you are connecting to.

VOTER Server Port

This is the UDP port that the Asterisk server is listening on for clients.

- Typically it is left at the default of port 667.
- **MUST** match the *port* directive in *voter.conf*.

Local Port (Override)

Sets the local unit UDP port to be something different.

- This is typically not used and left at default (0).

Client Password

This is the **unique** password for this unit.

- Every unit **must** have a **unique** password.
- **MUST** match the password on the client configuration line in *voter.conf*.

Host Password

This is the password of the Asterisk server.

- Configured with the *password* directive in *voter.conf*.

Tx Buffer Length

This is the size of the TX buffer from the RTCM/VOTER to the Asterisk server.

- Tuned parameter, set via the tuning instructions based on network configuration.

GPS Data Protocol (0=NMEA, 1=TSIP)

Type of GPS connected.

- NMEA for all non-Trimble GPS units.
- TSIP to use with Trimble GPS that use Trimble Serial Interface Protocol (TSIP).

GPS Serial Polarity (0=Non-Inverted, 1=Inverted)

GPS dependent setting.

- Depends on if you are sending TTL or RS-232 data, adjust as needed.

GPS PPS Polarity (0=Non-Inverted, 1=Inverted, 2=NONE)

GPS dependent setting.

- Depends on if you are sending TTL or RS-232 data for the PPS signal, adjust as needed.
- If not using GPS or configuring as a mixed client, set to 2 (None).

GPS Baud Rate

Communication speed for the GPS, default is 9600.

External CTCSS (0=Ignore, 1=Non-Inverted, 2=Inverted)

If you are using the CTCSS input from an external CTCSS decoder, adjust this for your logic polarity, or set to 0 (ignore) if not used.

COR Type (0=Normal, 1=IGNORE COR, 2=No Receiver)

Typically, this is left at Normal.

- Set to 1 (Ignore COR), if you are using the External CTCSS input.

Debug Level

Set the Debug Level, if needed, based on the Debug Options below.

Alt. VOTER Server Address (FQDN)

If you are using redundant Asterisk servers, set the IP/DNS address of the redundant server here.

Alt. VOTER Server Port (Overr ide)

If the redundant Asterisk server UDP port is not 667, set it here.

DSP/BEW Mode

If the firmware supports DSP/BEW Mode, set this to 1 to enable it, if desired.

- DSP/BEW firmware versions use extra firmware space and will make the diagnostic menu unavailable.

"Duplex Mode 3" (0=DISABLED, 1-255 Hang T ime) (1/10 secs)

If you are using Duplex Mode 3, set the hang time here to enable the option. Most users leave this disabled.

Simulcast Launch Delay

If you are using Simulcast, set your launch delay here.

RX Level

This opens the RX Level tuning screen to adjust the receive audio level from the repeater.

Status

This opens the status menu, showing some of the current operating parameters.

Save Values to EEPROM

Save all the options to non-volatile EEPROM.

- Make sure to save after you make any changes.

IP Parameters menu

Open the IP Parameters Menu.

Offline Mode Parameters menu

Open the Offline Mode Parameters Menu.

Disconnect Remote Console Session

If you are connected via Telnet, this will exit your session.

Reboot System

Reboot the VOTER/RTCM.

- This should be done after making configuration changes.
- Some setting changes require a reboot to become effective.

Diagnostics

Open the Diagnostics Menu.

- Diagnostics are unavailable if you are using DSP/BEW firmware.

Network Information

RTCM/VOTER Bandwidth

Assuming ulaw... The RTCM will require approximately 80kbps of bandwidth for continual usage.

Network Quality

Your IP Network quality is important. You may wish to add a rule in your routers to prioritize traffic on UDP port 667 so that RTCM audio is given priority over other traffic. Also see the #Debug Options below for notes on how to tag your packets with ToS.

RX/TX Buffers are NOT Both Millisecond Values

You might assume an RX buffer (in voter.conf) of 120ms would be equivalent of a TX buffer (in the RTCM) of 120ms. That is not the case.

The TX buffer is a number of 125 **microsecond** intervals, where the RX buffer is in milliseconds.

If you follow the buffer setting instructions, you should be fine, in most cases.

Debug Options

The VOTER/RTCM firmware supports some additional debugging information that can be turned on.

From the source code, the different Debug Options are listed as:

```
1 - Alt/Main Host change notifications
2 - Ignore HWlock (GGPS only)
4 - GPS/PPS Failure simulation (GGPS only)
8 - POCSAG H/W output disable (GGPS only)
16 - IP TOS Class for Ubiquiti
32 - GPS Debug
64 - Fix GPS 1 second off
128 - Fix GPS 1 month off (WTF,O??)
```

Not sure what they all do, but that is what they are. Here are the most common ones used:

- "Alt/Main Host change notifications"
  - shows when the connection to the Asterisk server changes state.

- "IP TOS Class for Ubiquity"
  - marks the IP headers from the RTCM/VOTER **TO** the network with ToS C0/DSCP 48 (UBNT shows this as 802.1p Class 6 (Voice <10mS latency). Other sources show this as a Network Control TOS.) If you enable this, you also will want to have "utos=y" in your voter.conf to mark the packets from the network **TO** the RTCM/VOTER. We will probably change this option to mark the packets by default in a future version.
- "GPS Debug"
  - will print NMEA or TSIP debug strings from the connected GPS.
- "Fix GPS 1 second off"
  - this is for NMEA GPS **only**. It will add one second to the time.
- "Fix GPS 1 month off (WTF,O??)"
  - this is for TSIP GPS **only**. It will effectively add a month to the date. The mktime function normally needs one month subtracted from the raw month it converts to a time/date string. Maybe there are some GPS out there that do that automatically?

The way this works is you add together the options you want to enable.

Want to enable GPS Debug and IP ToS, set debug to 48.

Just want to turn on GPS Debug, set debug to 32.

Just want ToS turned on, set debug to 16.

Un-documented Menus/Features

- Menu 96 - If firmware compiled with DUMPENCREGS enabled, Menu 96 looks like it dumps the registers from the ENC (Ethernet) chip?
- Menu 111 - show the "hidden" option values (normally they should all be 0).
- Menu 11780 - set the "Elkes" value. Developed for Pete Elkes, this is for solar type sites. It shuts off the Tx after a period of no Rx activity.
- Menu 1103 - set the "Glaser" timer value? No idea what this does.
- Menu 1170 - set "Sawyer Mode". Developed for Tim Sawyer use on Yaesu VXR5000 repeater. It modifies de-ephasis behavior when on/off line.

Audio

The RTCM/VOTER is totally flexible regarding emphasis. Although the way it is set is completely obscure. It tries to automatically do the right thing for you, which is great most of the time. But when it’s not, it's hard to know what is going on.

If you are changing the COR Type settings, or nodeemp in voter.conf, make sure you save/reboot the RTCM/VOTER every time you make a change... changes are **not** effective until the RTCM/VOTER reboots!

Voting

Before we get in to the different ways audio is routed, there is an important consideration you need to make if you are using the RTCM/VOTER for voting.

The way the voting process works, it needs **discriminator audio** to determine the signal to noise level from each satellite receiver. As such, you will need to be feeding discriminator audio in to the RX audio pin, so that the hardware/software can vote properly. That means you also need to let the hardware/software do the squelch action.

If you need RX CTCSS, you'll need to feed logic from an external CTCSS decoder in to the CTCSS input pin.

Also, don't disable COR in the RTCM/VOTER... it will cause it to disable the squelch and it will report an RSSI of 255 (full quieting) for all received signals.

Receiver De-emphasis

On the RX side the COR Type setting in the RTCM/VOTER determines whether the de-emphasis filter is used for RX audio. 0=Normal means the RTCM/VOTER squelch circuit is in use and it is expecting discriminator audio on the RX pin (to be able to do the squelch action), and therefore it **will** provide de-emphasis (audio is routed THROUGH the de-emphasis RC filter circuit) to the receiver audio. COR Type 1=IGNORE COR uses the CTCSS input pin for COR or CTCSS *logic* and it then expects de-emphasized receive (line) audio on the RX pin, and therefore will **not** provide de-emphasis to the received audio (the RC filter is switched **out** and audio passes straight through).

Most of the time, you do NOT need to override the automatic filter selection. However, if you do, and you are sure you have a good reason to, you CAN switch the de-emphasis filter **out** of the circuit so that audio passes straight through to the encoder. You would do this by setting the nodeemp=1 option in voter.conf. When you set nodeemp=1, the *VOTER Protocol* tells the RTCM/VOTER to switch the filter **out**, so audio is passed straight through.

Transmitter Pre-emhpasis

On the TX side, the RTCM/VOTER expects the repeater to accept *mic audio*. In other words, the **repeater** is providing the pre-emphasis, not the RTCM/VOTER (you are **not** directly modulating the TX). This can be overridden by setting CTCSS tone in voter.conf. If a CTCSS tone is defined, the RTCM **will** provide pre-emphasis to the audio, and expects that you will be connecting to your repeater's flat audio input (direct modulation).

If you don’t want TX CTCSS tone but **do** need pre-emphasis, set an arbitrary (any) CTCSS tone in voter.conf, and set the level to 0. This will force the RTCM/VOTER to pre-emphasize the audio it generates on the TX pin, but it won't actually mix in a CTCSS tone.

Level Setting

Setting the audio levels for the RTCM/VOTER is pretty straight forward. Just follow these steps:

- Ensure you have a connection to your host Asterisk server/chan_voter instance
- Send a 1kHz@3kHz on-channel, full-quieting signal in to the repeater's RX
- Set the RX up so it reads 3kHz deviation on the RTCM's built in console meter
- Now set the TX level **pot** to get 3kHz out of the transmitter (No PL)

Now change the modulation from 1kHz tone to 800Hz followed by 1.8kHz and verify that the deviation level doesn't change as the tone frequency changes. **Changing levels indicates a pre/de-emphasis issue.** You will want to read the above sections on how audio is handled, and figure out where your issue is.

If using PL you have to account for that deviation, unless you filter it out with your IFR (test set).

That's it!

Optionally, if you are using the built-in "offline repeat" functions, fail the connection to the host Asterisk server, and make sure your repeat audio performs the same as above.

Resolution issues when setting levels using built VOTER boards

Sometimes, depending on what radio is being used getting a good enough resolution with R61 when setting levels is a problem. You move a small amount on the pot and it jumps several hundred KHz of deviation. Inserting a 4:1 voltage divider between the VOTER board and transmitter helps with this. Something such as a 1K and 330 ohm resistor will work. This allows for much finer adjustment on the transmit audio (R61).

Crappy Transmit Audio

Does your repeated audio sounded really bassy, muffled, and not very understandable?

There was a situation where, compared to a typical simplex radio-to-radio transmission, the audio through the repeater (RTCM and Asterisk) was unacceptable. What was discovered was that the RTCM has an internal pre-emphasis function that was disabled. The user had intentionally disabled "txctcss" and "txctcsslevel" in voter.conf because he didn't want the RTCM transmitting CTCSS (the Quantar was doing that already). He ultimately found a post on the mail list explaining the settings above that said enabling txctcss = some valid tone (114.8 in his case) and setting txctcsslevel = 0 would turn on the pre-emphasis function in the RTCM without transmitting CTCSS tones.

He did, and it worked like a charm! Audio now had more treble and was less bassy/muffled. So, future RTCM users, be sure not to comment out txctcss and txctcsslevel in voter.conf!

Just enable and set the level to 0. You'll thank me later.

Pulse Noise on Transmit Audio

It has been noticed sometimes on built VOTER boards that it has low level DAC noise which sometimes is apparent as a pulsing noise on the audio output. Adding a 4:1 voltage divider inline between the transmitter and VOTER board helps reduce or eliminate this as described above in Level setting.

GPS

The RTCM/VOTER will work with most GPS available. It requires either NMEA or Trimble TSIP binary data. It only receives data **from** the GPS (GPS TX), it does not send anything **to** the GPS.

The firmware is specifically written to talk to Trimble Thunderbolt receivers using Trimble's TSIP binary data interface, however, other Trimbles GPS receivers that talk TSIP are generally compatible.

NMEA Sentences

If you are using an NMEA GPS (as opposed to a Trimble using the TSIP binary interface), the RTCM/VOTER is looking for the following NMEA sentences:

```
$GPGGA
$GPGSV
$GPRMC
```

GPS Lock

The GPS led will go solid regardless of the connection LED. That **has** to happen or it won't connect to the Asterisk server.

It is not unusual for it to take up to 20 mins to get a GPS lock LED (ie. using a Trimble Thunderbolt) after any reboot.

GPS De-sense

If you are having odd loss of lock issues, consider you may have interference to your GPS antenna from strong RF nearby. A note from Jesse Lloyd:

```
I also had crazy problems with poor signal on my GPS when I set it up sitting in a window, and once installed at

I found the debug setting of 32 useful in the RTCM, you can see a hex output of the GPS status.
```

GPS Issues

Trimble Thunderbolt

You may find your Trimble Thunderbolt is showing the incorrect date at the moment. It could be showing the year as 1997. This is due to the date in the Thunderbolt being reported incorrectly.

This can cause some of your voting receivers to not connect, if they are used with other GPS in your system. If you have **ALL** Thunderbolts, or **NO** Thunderbolts, you are probably fine. If you have **ALL** Thunderbolts, the date/time is probably wrong, but they will ALL be wrong, so they will connect.

GPS Time is a continuous counting time scale beginning at the January 5, 1980 to January 6, 1980 midnight. It is split into two parts: a time of week measured in seconds from midnight Sat/Sun and a week number. The time of week is transmitted in an unambiguous manner by the satellites, but only the bottom 10 bits of the week number are transmitted. This means that a receiver will see a week number count that goes up steadily until it reaches 1023 after which it will "roll over" back to zero, before steadily going up again. Such a week rollover will occur approx. every 20 years. The last week rollover occurred in 1999 and the next one will be in 2019.

The Thunderbolt manual states:

```
The ThunderBolt adjusts for this week rollover by adding 1024 to any week number reported by GPS which is less t
```

As such, the Trimble Thunderbolt has a firmware issue with the GPS Week rollover that manifested itself on July 30, 2017, causing the date to become incorrect. The Thunderbolt thinks the week changed from 935 to 936 (actual week 1959-1024=935), so it stopped adding 1024 to the week.

We have added a brute-force fix starting in RTCM/VOTER firmware >=1.51. This fix adds 619315200 seconds (1024 weeks) to the time reported by the GPS. It fixes the Thunderbolts, we have not done extensive testing to see how it affects other TSIP receivers.

Garmin

Garmin 18x LVC Wiring Issues

If you have issues with your GPS 18x LVC not talking to the RTCM/VOTER, it may not be hooked up to the RTCM correctly.

The Garmin pin labeling is backwards to what you may think. See below. You probably need to swap pins 6 and 14.

```
RTCM                    GPS 18x LVC
6 GRX   <-- Rx Data      6 Green
7 GPPS  <-- Pulse Output 1 Yellow
8 GND   Ground           3 Black
8 GND   Ground           5 Black
13 +5V  -->Vin           2 Red
14 GTX  --> TX Data      4 White
```

Log into the RTCM and do 98 and you should see something like this:

```
Current Time: Sun  Apr 20, 2014  04:37:02.820
Last Rx Pkt System time: 04/20/2014 03:55:35.580, diff: 2487260 msec
Last Rx Pkt Timestamp time: 04/20/2014 03:55:32.064, diff: 3515 msec
Last Rx Pkt index: 160, inbounds: 1
```

Garmin and the RTCM

Beware when buying newer Garmin GPS's to use with the Micro-Node RTCM.

The RTCM expects a 5V PPS signal.

Newer Garmin's (GPS 18X, 18X LVC, etc.) MAY NOT output 5V, and can cause issues. **Check the Garmin datasheet**.

The VOTER is designed to accept both 3.3V or 5V signals, and *should* work fine.

- Some of the Garmin GPS's come with 4800 baud set as default. If you are getting a "Warning: GPS Data time period elapsed" error on your RTCM, change both the GPS and RTCM to use 9600 baud. To do this, interface the GPS to a DB9 connector as per page 8 of the manual (remembering to ONLY use 5V as the power source *facepalm*). Once done, download, extract and open SNSRXCFG_330.exe and run.

1. Select your GPS (in most cases GPS 18x PC/LVC). Press F10 to switch to NMEA mode (Config > Switch to NMEA Mode)
2. Select Config > Setup and choose the COM port your GPS is connected too. Leave baud rate as auto for now, OK.
3. Select Comm > Connect to connect to the GPS.
4. Go to Config > Get Configuration from GPS to download it's current configuration
5. Click File > Save to save the current configuration
6. In Config > Sensor Configuration change the Baud Rate to 9600. You can also check to make sure 1PPS is enabled here too. Click OK.
7. Hitting F7 when in the main window of the software also brings up the GPS sentences to output if that is of interest.
8. File > Save to a different file than step 5.
9. Press F9 or Config > Send Configuration to GPS. This will then send all the changes you made to the GPS unit (including baud rate so a reconnect may be needed)

No GPS/Mixed Mode

You do not need a timing source to use an RTCM if you don’t want to vote. That’s called Mix Mode.

Having the PPS Polarity set to 2 tells the RTCM you do not have a GPS. That forces the RTCM to become a mix (non-voting) client. Turn it on, save it, and restart. A simple voter.conf would look like this:

```
[general]
port = 667
password = BLAH

[1000]
Site1 = pswrd1,master,transmit
Site2 = pswrd2,transmit
```

Mixed Client Error

"I am getting this error in Asterisk":

```
WARNING[2368]: chan_voter.c:4511 voter_reader: Voter client master timing source mobile1 attempting to authentic
```

A mixed client error means the voter.conf file is expecting an RTCM to try and connect with a GPS IP packet (ie. has **master** in it in voter.conf), but the RTCM isn't sending a GPS IP packet. So its a mismatch between voter.conf and Option 10 in the RTCM.

If you want to use a mix client (non-voted), make sure that receiver's configuration line in voter.conf **does not** have the **master** option set.

Mix Clients with Voted Client Issues

Situation... "I have a private node with 6 voted receivers using RTCMs. I'd like to add a 7th RTCM to this node that is always mixed in rather than voted. I'm able to make this RTCM work as a 7th voted receiver with no problem. Everything I've read seems to indicate that if I change GPS PPS polarity to "none" this will achieve my desired results, however I am unable to get audio out of my transmit RTCM from this 7th site. It does change color to cyan in Allmon indicating it is non voted input but I do not get any indication or audio when that units COR goes active."

"Setting voter debug level 3 in Asterisk and I'm seeing the following message repeatedly scrolling by in a blur when the mix client detects COR (COR is active). Sequence numbers are continually incrementing by 1. I'm running software version 1.47 on all my clients:"

```
mix client (Mulaw) my_client index:0 their seq:629 our seq:629
mix client my_client outa bounds, resetting!!
```

If you have a similar situation to the above... check your buflen in voter.conf. Make sure it is at >=160 and see if that fixes it.

GPS Debug

To turn on GPS Debugging, set the Debug Option Level in the RTCM/VOTER to 32.

See #Debug Options levels for more information on how this works.

Trimble Debug Status Decoding

The VOTER/RTCM when in TSIP mode (Trimble), assumes it is a Trimble Thunderbolt and is looking for two packets:

- 0x8F-AB - Primary Timing Packet
- 0x8F-AC - Supplemental Timing Packet

Grab a copy of the Thunderbolt User Guide: http://leapsecond.com/pages/tbolt/Thunderbolt-2012-02.pdf

Pages 78-83 are the important ones.

Packet 0x8F-AB is what grabs the timing information, and packet 0x8F-AC is what it looks at for everything else (including debug).

The debug string that the VOTER/RTCM reports:

```
       printf("GPS-DEBUG: TSIP: ok %d, 9 - 14: %02x %02x %02x %02x %02x %02x\n",
                                        happy,gps_buf[9],gps_buf[10],gps_buf[11],gps_buf[12],gps_buf[13],gps_bu
```

So, the 1 after ok is the "happy gps" flag. The other 5 bytes are supposed to be bytes 9-14 from the packet... sort of. They are bytes 9-14 of the buffer, but they are actually bytes 8-13 of the binary message.

Looking at the message structure, bytes 8-13 are:

- 8-9 Critical Alarms
- 10-11 Minor Alarms
- 12 GPS Decoding Status
- 13 Disciplining Activity

Therefore, the messages you are seeing break down as follows:

```
GPS-DEBUG: TSIP: ok 1, 9 - 14: 00 00 00 00 00 00 - everything is good in the 'hood, Doing Fixes, Phase Locking

GPS-DEBUG: TSIP: ok 0, 9 - 14: 00 00 00 18 08 06 - not happy, Not Tracking Satellites, Not Disciplining Oscillat

GPS-DEBUG: TSIP: ok 0, 9 - 14: 00 00 00 08 08 05 - not happy, Not Tracking Satellites, No Useable Sats, Compensa

GPS-DEBUG: TSIP: ok 0, 9 - 14: 00 00 00 00 00 05 - not happy, Compensating OXCO (holdover)

GPS-DEBUG: TSIP: ok 1, 9 - 14: 00 00 00 00 00 08 - happy, Recovery Mode

GPS-DEBUG: TSIP: ok 0, 9 - 14: 00 00 00 00 00 04 - not happy, Initializing Loop Filter
```

The GPS is flagged as NOT HAPPY in TSIP mode if ANY of the following are TRUE:

- If GPS Decoding Status is anything other than "Doing Fixes".
- If Disciplining Activity is not Phase Locking or Recovery Mode.
- Any Critical Alarms.
- Any Minor Alarms.

Ubiquity ToS

See https://help.ubnt.com/hc/en-us/articles/205231750-airMAX-How-is-QoS-and-prioritization-handled-by-airMAX-

So, if we set ToS/DSCP in the header to C0, then Ubiquity (and other gear watching ToS) *should* prioritize the packets. This sets the DSCP to 110 000 aka 48. UBNT shows this as 802.1p Class 6 (Voice <10mS latency).

Other sources show this as a Network Control TOS.

There is an option in voter.conf to turn this on (utos=y). However, this only controls packets being sent from Asterisk **TO** the RTCM.

If you want to tag packets from the RTCM **TO** Asterisk, you need to set the RTCM debug option level to 16 (see #Debug Options for how this works).

At some point we should probably change the default behaviour of the RTCM firmware to mark the packets and use the debug setting to **disable** ToS. That change (if you wanted to compile your own firmware) would be:

```
Change line 90 in IP.c

From:
#define IP_SERVICE         ((AppConfig.DebugLevel & 16) ? 0xc0 : (IP_SERVICE_ROUTINE | IP_SERVICE_N_DELAY))

To:
#define IP_SERVICE         ((AppConfig.DebugLevel & 16) ? (IP_SERVICE_ROUTINE | IP_SERVICE_N_DELAY) : 0xc0)
```

This changes the conditional expression from if the debug level is 16 to mark the packets with DSCP 48 to if the debug level is 16, mark the packets routine (DSCP 0).

RTCM Simulcasting

The RTCM/VOTER boards do support simulcasting, however, there are a bunch of quirks that one needs to be aware of.

Radio Hardware

For best results, you should use all identical RF equipment between your voting RX sites and simulcast TX sites. If you don't, you can end up with strange audio artifacts when different receivers are used, and other strange audio issues when different transmitters are used. The most important thing when setting your levels is that ALL the RX and TX levels match from radio to radio. The best way this is achieved is using a "master" radio and reference from that - all radios might be identical models, but may not have identical audio characteristics so check those levels! Also check CTCSS and audio levels separately to make sure they all match.

9.6MHz Oscillator

If you want reliable simulcast with an acceptable level of overlap warble, you MUST inject a GPS-disciplined or OXCO 9.6MHz signal to the clock of the RTCM's, in lieu of their stock internal crystal.

Otherwise, the inherent internal clock jitter of the RTCM's will cause CTCSS warble, etc. in areas where your transmitters overlap. The 1 PPS feed that you normally feed the RTCM's is acceptable for voting, but naturally,

a typical crystal-driven clock will slightly drift in between the pulses every second due to inherent jitter, etc., and that ain't to good for situations like simulcast which demand sample-accurate performance.

Here are some comments from Joe, KC2IRV, on the subject. He has also created his own wiki, specific to simulcast issues, you can find it at: http://rtcmsimulcast.wikifoundry.com/

> I felt I needed to let those on this mailing list know my findings with simulcasting with the RTCM units since I have received a great deal of help and information from people on this list.
>
> For the past week I have had a two (2) site simulcast system up and running on UHF using the RTCM units. I have done a great deal of testing and proving on the bench and it is so far working beautifully.
>
> The one issue that was made apparent to me early on was that in order for the RTCM's to be suitable for simulcast, the processor inside it needed to be clocked to a more accurate source. This source is supposed to be the Programmable Clock Generator Module in order to produce a GPS locked 9.6 MHz reference for the processor. Unfortunately this unit is not currently available.
>
> Knowing this, I started to go down the road of possibly designing my own, but stumbled upon a cheap, plentiful supply of Symmetricom 9.6 MHz sinewave OCXO's. I took a chance and ordered a few to experiment with to see if the RTCM would be able to use this in place of the supplied crystal for a clock. I found they did, with that I decided to modify the RTCM with an SMA connector just above the audio adjustment pots and use this OCXO. I trimmed the frequency on the units to match each other and used them to produce the processor clock.
>
> After a week of testing, I have found the results in the overlap areas to be exactly as I would expect them on a public safety/commercial simulcast system. While in an overlap region where both transmitters are within 3dB of one another, to the ear you hear no audio phase wandering. Of course , this has only been a week of testing so I wouldn't call it case closed yet.
>
> I will say that if this system remains this way after testing for 6 months to a year than using a 9.6 MHz OXCO is a viable alternative to achieve the needed precision and accuracy needed for the processor clock to make it suitable for simulcast.

Comments from Jim Dixon on the issue:

> You have to use **something** that will take the precise 10MHz (or whatever it is) from the GPSDO and PLL (phase-locked) convert it to 9.6MHz (square wave at 3.3V for the CPU clock).
>
> We had to use 9.6MHz mainly because of the DAC in the dsPIC instead of 10MHz. The options for various divide/clock radios in the part are RATHER limited. As I recall, there wasn't even a way of getting the necessary 16kb/s sample rate on the ADC from 10MHZ, either.

Micro-Node RTCM Clock Issue

James, KI0KN, had some strange-ness with some of his RTCM's when used for voting. As soon as he changed menu item 10 on the RTCM to either a (1) or a (0) instead of (2), he **instantly** got this on the RTCM console:

```
04/05/2016 18:44:13.660  Lost GPS Time synchronization
04/05/2016 18:44:13.660  Host Connection Lost (Pri) (10.16.1.240)
```

And it sits there forever and never re-establishes connection to the host. The issue was finally traced to a bad batch of 9.6MHz crystals that affected a small run of the RTCM's. The issue was eventually resolved by Micro-Node, but we'll document it here just for record keeping.

James comments to the list:

> Well, after many of you offered your time and thoughts on my voter problem, Jim, WB6NIL, graciously donated a couple of hours of his time to remotely help me and he uncovered the problem.
>
> All 5 of my RTCMs were purchased within the last 6 months, and I suspect all 5 have the same problem (will verify that today). The problem is that the microprocessor crystal is running too fast! There is 9.6Mhz crystal that drives the MPU, on the particular unit that Jim helped me diagnose, it's running 2.5 khz too fast.
>
> There is a sanity check in the firmware that makes sure the correct number of samples were taken in the last second (it's supposed to be 8000) and fails if the sample is incorrectly sized (mine is taking 8003 samples). Jim helped adjust the code to be more tolerant of the sampling error and my whole system instantly worked.
>
> I have since tested it with ALL my GPSs (they all work great!). So I now have everything working, including the voting.
>
> I haven't contact Micro-Node about it yet. Jim told me that voting should work fine but this clock error would probably not be acceptable for Simulcast (something we have no plans on doing as of now). Since mine are deployed at mountain top sites, I'll probably pursue a crystal that is running at the correct frequency and use "standard" firmware so that if the decision is ever made to play with simulcast, I won't be right back in this same boat.
>
> Thanks to all that helped out, and a HUGE thanks to Jim for taking the time to troubleshoot this.
>
> BTW- This clock error will NOT effect the RTCM in mix client mode, it only affects it in voting or simulcast mode!
>
> In voter.c, the stock line is:

```
if ((samplecnt >= 7999) && (samplecnt <= 8001))

it needs to be changed to something like:

if ((samplecnt >= 7999) && (samplecnt <= 8003 ))
```

> That was just enough to make my first RTCM work. I've had a chance now to check a few more of my RTCMs on the bench this morning and the crystal frequencies are kind of all over. The next one I had on the bench was 5.2Khz fast. The above code was still not enough to fix it (as would be expected) so it looks like the quality control on whatever manufacturer that crystal is, isn't very good (or a low tolerance crystal is being used). I am going to pursue crystal replacement to get a highly accurate, stable, on-frequency crystal in there as my first choice. Changing the firmware 5 different times doesn't seem like the right answer.
>
> Jim put that sanity check in there for a reason. Working around it **may** allow the device to work, but to me, it seems getting the hardware operating the way it was supposed to originally is the better answer. I will follow up with Micro-node and let you all know how it goes.

After following up with Micro-Node:

They were all off, some were off a LOT. I talked to Mark @ Micronode. He told me there were 25 units that made it out the door with a crystal from an unapproved source and that was quite likely the reason for the problem. He gladly sent me 5 new crystals to replace (he offered to replace them himself if I sent the units in, but I am comfortable doing it myself to save the time and postage).

I haven't received them in the mail yet to show that the problem is fixed, but I will post here when that happens. He was great to work with and had no issue getting things set straight. If you are having the same issue and your RTCMs were bought about the same time, I'd suggest you contact him and have him help you resolve the issue!

Duplex Mode 3

Does delayed repeat audio bug you? Tired of hearing a bit of yourself after you unkey? Wish you could talk full duplex? Hate the echo chamber effect? Then we have the answer for you.

Duplex Mode 3 in app_rpt allows for "in-cabinet repeat" (where the radio hardware provides repeat audio) and app_rpt adds the hang time, courtesy tones, linking - all the things apt_rpt does sans repeat audio. Therefore no repeat audio delay. Cool, eh? This duplex mode has been in app_rpt for a while. Problem has been how to implement it in the RTCM environment.

Duplex Mode 3 support in the RTCM provides in-cabinet repeat functionality. Repeat audio loops through the RTCM and has almost zero delay because it does not have to traverse the network. The delay is not quite zero but it's plenty short enough to eliminate all of the above mentioned annoyances.

Of course Duplex Mode 3 support can't be used with voting or simulcast. You also loose Touch Tone muting, Time Out Timer and Repeater Disable functions because the repeat path is not through app_rpt.

Setting Voter Buffers

Voter ping is useful for end-to-end network evaluation when ICMP ping is turned off and/or the RTCM is behind a firewall and is not ICMP reachable. It can also help with finding the correct RTCM and voter.conf buffer settings.

Voter Ping Usage

The voter ping asterisk CLI syntax is:

```
*CLI>voter ping nameOfClient [packetCount]
```

If packetCount is not specified 8 pings will be sent. Use packetCount of 0 to stop an in progress voter ping. Set packetCount to at least 100 when evaluating link quality.

The result will be similar to:

```
...
PING (nameOfClient): Packets tx: 100, rx: 100, oos: 0, Avg.: 26.710 ms
PING (nameOfClient):  Worst: 38 ms, Best: 22 ms, 100.0% Packets successfully received (0.0% loss)
```

The output above is self evident except for oos which is a count of out of sequence packets. Voter ping requires RTCM firmware 1.23 or newer and chan_voter 2013-08-04 or newer.

RX Buffer Size

Ping all the receiver sites and look for the worst response of the worst client. As a rough rule of thumb the buflen setting in voter.conf should be set to the worst response + 40ms or 120 whichever is greater. Using the above case the buflen should be set to 120 (38+40

TX Buffer Size

Ping all the transmitter sites and look for the worst response of the worst client. The RTCM TX Buffer Length should be set to (worst response + 40ms) * 8 or 480 whichever is greater. Using the above case the RTCM TX Buffer Length should be set to 624 ((38+40)*8).

Assumptions

- The minimum TX buffer size is 480 (60ms) and the minimum RX buffer is 120ms. These were derived by testing on a LAN segment with chan_voter 2013-08-04 and RTCM 1.26.
- The ping times are not round trip times but they are in fact round trip times. Therefore the worst response could (should?) be divided by 2. Ie RX buffer = 38/2+40=59 and TX buffer = (38/2 + 40) * 8 = 472. Minimums still apply.
- The internet path to and from the RTCM under test is symmetrical.
- The added 40ms pad is an estimate of buffer ingress and egress.

As always your milage may vary. Some trial and error may be required to find the optimum settings.

Setup

The RX buffer is set with buflen=120 in /etc/asterisk/voter.conf. The value is in milliseconds. The TX buffer is set in the RTCM with menu item 7. The value is in 125 microsecond increments. To match the size of the TX to the RX buffer, use the RX buffer * 8 to get the TX buffer size.

Categories: How to | Node Configuration
