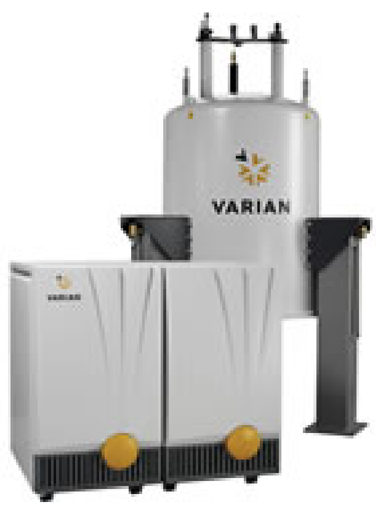
VnmrS Troubleshooting Guide

Varian, Inc. MR Systems

IRD – MR Tech Support
Rev C0310



VARIAN

VnmrS Troubleshooting Guide – Rev C0310

Technical contributors: Al Ramirez, Christine Hofstetter, Eric Frick, and Phil Hornung

Technical editor: Ken Kezeor

2700 Mitchell Drive, Walnut Creek, California 94598

1-800-356-4437

www.varianinc.com

Table of Contents

Chapter 1 Purpose

1.1 Purpose

The purpose of this kit is to assist in trouble shooting various sections of the VNMR system. The trouble shooting sections include the following:

Chapter	Topic Covered
2	Console Checks of the RF Transmit and Receive Paths
3	How to Measure the Signal to Noise Figure
4	Console Communication
5	Shims Supply
6	Software Diagnostics
7	Tuning
8	VT & Pneumatics
9	

1.2 Equipment required for all sections

- A calibrated oscilloscope:

Note: Please note that using an oscilloscope to measure frequencies higher than its capability results in attenuated signal measurements. All RF measurements must be done with the 50-ohm termination. All TLL signals must be measured using the 1-Mohm termination. All signals should be made with the shortest length possible to minimize signal loss. A good choice is the lock cable.

- A calibrated frequency signal generator
- A calibrated digital volt meter (DVM)
- The shim resistor board part#: 01-905888-00 is a 40 channel version
- A 9 pin F-9 pin F null modem cable
- 30 dB 300 Watt attenuator

1.3 Documentation required for all sections

- Varian NMR System Signal Paths_user (power point)
- Console Checks for VnmrS
- VnmrS Troubleshooting Communication
- VnmrS Troubleshooting Shims
- VnmrS Software Diagnostic Tools
- VnmrS Troubleshooting Tuning
- VnmrS Troubleshooting Variable Temperature Issues

Section	Equipment Required	Documentation Required
1 RF transmit/receive	1. Scope	VnmrS signal paths_user
	2. Signal generator	Console checks
	3. 30dB attenuator	
2 Mixer control lines	Mixer Jig	VnmrS Mixer Troubleshooting
3 Preamp control lines	Preamp Jig	VnmrS Preamp Troubleshooting
4 Console communication	9F-9F null modem cable	VnmrS Troubleshooting Communication
5 Shims Supply	1. Shim resistor board	VnmrS Trouble shooting Shims
	2. DVM	
	3. 9F-9F null modem cable	
6 Software Diagnostics	9F-9F null modem cable	Software Diagnostic Tools
7 Tuning	1. Scope	VnmrS Troubleshooting Tuning
	2. Signal Generator	
	3. DVM	
8 VT/pneumatics	1. DVM	VnmrS Troubleshooting Variable Temperature Issues
	2. Scope	

Chapter 2 Console Checks for VnmrS

2.1 Equipment needed

The following equipment is required to perform these tests:

Oscilloscope

Signal generator

30dB attenuator

2.2 Transmitter Paths Tests

Note: ***The results of the transmit path tests below can vary depending on the length of cable used & max frequency of the scope. Measuring higher frequencies (H1) than the scope can handle or excessively long cables (try to use the lock cable) will result in lower measurements.***

To test the High band and low band paths

1. Retrieve the H1sn parameters from /home/vnmr/parX00/tests & set these parameters (the par X00 refers to par400, par500, etc):
   pw=30, *d1=0.1*, at=0.2, nt=1e6, in='n', dp='y'
   tn='H1' when measuring proton (for VNMRS also set dn='C13')
   tn='C13' when measuring carbon (for VNMRS also set dn='H1')
   (for VNMRS: tn cannot equal dn)

Note: ***Set tpwr=31 when NOT using a 30dB attenuator***
Set tpwr=63 when using a 30dB attenuator or when measuring the attenuator outputs.

Caution: ***Please use caution so as not to damage the scope or NMR probe***

2. After typing go, measure the following (Vpp) using the 50-ohm scope setting:
   a. Measurements at the TX (all systems)
      - H1/C13: 1.3 – 2Vpp
   b. Measurements at the attenuator outputs (all systems tpwr=63)
      - J297 (H1) or J298 (C13): >600 mV
   c. Measurements at the cable (H1 or C13) that attaches to the probe
      - H1: >= 4V (typical: 4.5 –5 V)
      - C13: >= 9V (typical: 11v)

To Test the Lock Transmit Path

1. In the lock screen set lockpower to max & measure
   - Lock Transmitter output: > 3Vpp
2. Lock signal at the probe: >1.1V (50-ohm setting) and pulse has a width of 5 msec when lock scan ON or 50usec when off

2.3 Receive Paths Tests using a signal generator

Testing the high band & low band entire receive paths

Note: The results of the signal-to-noise tests below can vary depending on the amplitude accuracy of the signal generator & length of cable used

1. Use H1sn.par & set:
   a. gain=30 at=2 nt=1 pw=0 d1=0.1
   b. tn='H1' for proton test or tn='C13' for carbon test. *Don't change sw*
2. Set the signal generator:
   a. frequency to sfrq
   b. level= -110 dB (make sure there is *no delta sign* next to the frequency or the -110dB)
   c. Connect it to either the H1 preamp for proton test or the C13 preamp for carbon test. Signal is injected into the preamp.
3. Acquire a spectrum & type:
   a. f full aph0 dc vsadj (return)
   b. put up 2 cursors & type delta=1k (return)
   c. move 2 cursors to the extreme right
   d. type dsnmax(1k) (return).
4. Signal-to-Noise should be as follows:
   a. H1 & C13 > 650

To test the mixer & observe receiver (exclude the preamp)

1. set the sig gen to sfrq @ -100 dB & inject into the mixer HB or LB input
2. acquire a spectrum & type: f full aph0 dc vsadj (return) then type delta=1k (return)
3. move the cursors to the extreme right and type dsnmax(1k)
4. signal-to-noise should be as follows
   a. H1> 750 & C13> 800 (best)
   b. H1 & C13 (typical) > 650

To test the observe receiver only

1. Set the sig gen to 20 MHz @ *-60 dBm* & connect it to the obs receiver input
2. Acquire a spectrum & type: f full aph0 dc vsadj (return); type delta=1k (return); move the cursors to the extreme right; type dsnmax(1k)
3. Signal-to-noise should be as follows:
   - H*1 & C13 > approx 6500*

2.4 Lock Receiver Path

To test entire lock receive path

1. Open the lock screen window & set lockpower=0, lockgain=max

2. On the command line type: lockfreq?

3. Set the signal generator to that freq at a level= -110dB

4. Inject the signal generator into the lock cable at the probe (remove the cable from the probe's lock port).

5. Make sure you see a saturated/clipped, slightly off-resonance lock sinewave

    - (If sinewave is not clipped, receive path may have a problem)

To test only the lock receiver

If the signal injection at the probe does not give a clipped signal or gives no signal (flat line), do the following:

1. Set the signal generator to –110 dBm at lockfreq
2. Inject into lk preamp, then measure any J223, 224, or 225 (0/90 outputs mid pin), you should get > 10Vpp using 1Mohm at max lock gain (also nmr lock signal is clipped)
3. Inject –70 dBm @ lockfreq into lock receiver, then measure any J223, 224, or 225 (0/90). Each connector has 6 pins: 3 on top & bottom of the notch. The middle pin (at 12 o'clock & 6 o'clock) you should get >4Vpp using 1Mohm at max lock gain

2.5 Mtune

Mtune is a very quick check of the console. If you see a clean tune signal on screen, then most of the console is functioning. It *does not* test the mixer's HB, LB, or lk inputs—only tune input. It also does not use the preamp receive path.

2.6 VnmrS Self-Tests (No scope or signal generator required)

The results of the signal-to-noise tests below can vary depending on the length of cable used & the amplitude of the transmitter and attenuator.

For this test use the lock cable (00958298-14) approx 9ft long. The VnmrS LB preamp used the standard filters (LB + lk reject or barrel LPF) and universal (protune) quarterwave cable. For 400MR, use the lock cable (1916955-13). An average of these measurements on 3 systems is shown below. All NMR peaks should be in the center of the window.

If your measurement varies a couple 1000 units, it is ok. If it varies by half, then there is a problem.

Note: Important Note to avoid confusion: In normal NMR operation, the attenuator J297 is the high band output & J298 is the low band output. But for these console checks we are forcing J297 to be low band & J298 to be high band by forcing transmitter 1 to output C13 & transmitter 2 to output H1 (attenuator board is not being switched). These tests are injecting the output of the attenuator into either the H1 cable or C13 cable that was attached to the probe. This signal travels into the directional coupler, filter (if any), preamp, mixer, and then receiver (we are simulating a small probe signal).

H1 TX/RX path

This test uses TX2 to pulse H1 into the console but uses TX1 LO

1. Retrieve H1sn.par from /home/vnmr/parX00 & change the following (parX00 refers to par400, par500, etc): dfrq=sfrq, dn='none'. pw=0, tpwr= -16, dpwr= -16, dm='nny' dmm='c', gain=30, d1=0.1
2. Connect the lock cable from the attenuator J298 out 2 (2nd tx/HB output) to H1 cable that was connected to the probe H1 port.
3. Aquire a spectrum then type: aph0 nm dc f full vsadj
4. Put up 2 cursors, type delta=1k & move the cursors as a pair all the way to the right.
5. Type dsnmax(1k) and expect to get:
   - H1 s/n > 30k (V400s) or >26k (500s/600s)

C13 TX/RX path

This test uses TX1 to pulse C13 into the console but uses TX2 LO

1. Retrieve H1sn.par & change the following: tn='C13', dn='none', dfrq=sfrq, pw=0, tpwr= -16, dpwr= -16, dm='nny' dmm='c', gain=30, d1=0.1
2. Connect the lock cable from the attenuator J297 out 1 (1st tx/LB output) to C13 cable that was connected to the probe C13 port.
3. Acquire a spectrum then type: aph0 nm dc f full vsadj
4. Put up 2 cursors, type delta=1k & move the cursors as a pair all the way to the right.
5. Type dsnmax(1k) and expect to get:
   - C13 s/n > 36K (V400s) or >30k (500s/600s)

To test the mixer & receiver

1. Same parameter setup as the H1 or C13 above
2. For H1, connect the lock cable from the attenuator J298 out 2 (2nd tx/HB output) to the High Band port on the mixer
   - After the dsnmax(1k) expect:
     H1 > 9200 (400s) or >6k (500s/600s)
2. For C13, connect the lock cable from the attenuator J297 out 1 (1st tx/LB output) to the Low Band port on the mixer
   - After the dsnmax(1k) expect:
     C13 > 12200 (400s) or >6k (500s/600s)

To test the receiver only (excludes preamp & mixer)

1. Retrieve H1sn.par & change the following: tn='none', dn='none', sfrq=20, dfrq=sfrq, dn='none', pw=0, tpwr= -16, dpwr= -16, dm='nny' dmm='c', gain=30, d1=0.1
2. Connect the lock cable from the attenuator J297 out 1 (1st tx/LB output) to the DDR "IF in" port.
3. Acquire a spectrum then type: aph0 nm dc f full vsadj
4. Put up 2 cursors, type delta=1k & move the cursors as a pair all the way to the right.
5. Type dsnmax(1k) and expect to get:
   - 20 MHz peak s/n > 1200 (all systems)

Chapter 3 How to Measure System Noise Figure

3.1 Equipment needed

- a noise source such as the Agilent (HP) 346A or equivalent
- a +28V power supply to turn-on the noise head.
- a temperature monitor to measure the room temperature.
- the eafmpar1.tar file

3.2 Hardware Connections

1. Directly connect noise head male end to either the “probe” port on the high band or low band preamp. The male end is at the top of the noise source when looking at the noise source numbers (dB & GHz) right-side up. Then connect the +28V supply to the other end (bottom) of the noise head. The supply must be turned off.
2. Disconnect the connection from the power amplifier to the preamplifier at the “XMIT” port of the preamp under test. This is to protect the $5K noise head from being blown up! ! !

3.3 Software Setup

1. On the host computer do the following:
   - Put the eafmpar1.tar file in the vnmr1’s home directory.
   - Use tar xvf eafmpar1.tar to extract.
   - Note: Macros will overwrite predecessors. Five macros get copied into the local maclib and a parameter set into the local parlib.
   - On the command line type: rtppar(‘eaf’) and check these parameters:
   - pw=0, np = 500000, sw = 100000 Hz, at = 2.5 s, tpwr = -16, d1 = 1s, d2 = 0.001s
   - tof = 0 s, nt = 1, lb = not used, seqfil = ‘s2pul’, gain = 54 (may vary)
2. Select preamp under test with tn (observe) = H1 or C13
3. On the noise head, look at the dB values next to the 10, 100, 1000, & 2000MHz. The top 4 values (ENR) will be used.
4. On the command line type in the following:
   - enr10MHz = X.XX (where X.XX is the dB value next to the 10MHz…such as 15.34)
   - enr100MHz = X.XX
   - enr1000MHz = X.XX
   - enr2000MHz = XXX
5. Using a thermometer, measure the ambient air temperature of the air near the noise head. Allow a minute or two for settling of the thermometer after handling.
   - On the command line type in:
     tambient = XX.X (where X.XX is the measured air temperature in °C)
6. On the command line type in: ef0

Note: The interpolated ENR is displayed

   a. When the message “reply Cold when noise head cold” appears, make sure the 28V power supply is OFF. Then hit enter. The system will acquire.
   b. When the message “reply Hot when the noise head hot” appears, turn ON the 28V power supply then hit enter. The system will acquire again and show the system noise figure.
      Noise figures specs are:
         H1< 1 (typical is ~ 0.96)
         C13< 2 (typical is ~ 1.53)

If you get “ADC overflow”, reduce gain from 54 to 52 by typing on the command line and re-run the test by re-typing ef0 (step 6a & b).

7. Record the Noise Figure and Gain value reported both in results window and on text page. To get a print out, type (on the command line): printon efcalc printoff page
8. Remove the noise head and reconnect the probe connection.
9. Reconnect the power amplifier to the XMIT of the preamplifier.

Chapter 4 Communication

This section discusses VnmrS communication issues with the host PC running Linux OS. The “Controller Initialization” document is needed to do some of the trouble shooting discussed here.

4.1 Equipment needed

The following equipment is required to perform these tests:

- 9F-9F null modem cable

4.2 How to trouble shoot a communication issue:

Controller bootup process

Each VnmrS controller has a unique IP address. At bootup the PPC executes software in flash that initializes the PPC GPIO lines. These lines read the type bits of the IF card to dynamically configure the MAC address of each controller. VxWorks then uses RARP to obtain an IP for the controller. Controller firmware is transferred from the host to all controllers simultaneously storing them in flash memory. The PPC then executes nvlib.o, which uses the GPIO lines to read the type bits (set by the IF card or 3 jumpers) to configure the FPGA, which completes the bootup.

General tips to keep in mind

1. Info about setacq:
   - Setacq cannot load files into a blank controller. All controllers need to boot up completely for setacq to download files. If a controller does not boot, setacq will hang and not download to the rest of the controllers.
   - Setacq will not re-flash a controller a second time if it has been done once on the same system and patch level. It simply *verifies* if a controller’s firmware matches the version in /vnmr/acq/download. Verification is indicated by the DS5 flashing once per file checked & takes about a minute to check all files on all controllers.
   - A full download of 9 files for all controllers takes about 7-8 minutes. During the downloading, the yellow LEDs on all controllers blink in sync in a unique pattern per file downloaded. *During this process you must not reset the master as this can erase some or all firmware in the controllers.*

2. All controllers can bootup without the 80 MHz clock, but the clock is needed to get “setup complete”. If the clock is missing to any controller, an “su” results in a hangup with the message:
   - *“pulse sequence required controller(s) are unresponsive in initial synch”*

3. If a controller did not bootup or “cpu_err” LED is on:
   - Ensure the IF card power LED is lit. If it is off, this indicates improper seating.
   - Reseat the controller
   - Re-flash the controller using the “Controller Initialization” procedure

4.3 Issue: System is “inactive”

Check the setup and status of the network cards

1. eth1 (mother board card) should be connected to the console’s Ethernet port which connects to the 1 MB port on the hub. Its IP should be 172.16.0.1 (or 10.0.0.1) and subnet is 255.255.255.0
2. eth0 (card slot) should be connected to the LAN with their network info. The eth0 cannot be named wormhole and should be active even if not connected to any LAN.
3. Check eth0 by pinging it. It should respond with a repetitive line that starts: *“64 bytes from wormhole (172.16.0.1)….”*

Check if you can ping wormhole, master1 & DDR1

In a terminal window type:

1. ping wormhole : should result in “64 bytes from wormhole (172.16.0.1): icmp…….” repeatedly
2. ping master1 (or ddr1) : should result in “64 bytes from master1 (or ddr1)…..” repeatedly
   - If ping fails you see: “from wormhole (172.16.0.1) …. Destination Host unreachable”.
   - If rlogin fails, you see: “….no route to host”

TO PING OR RLOGIN INTO A CONTROLLER, IT NEEDS 3 THINGS

- the rarp demon must be running (see below)
- boot.ini must have the correct network info (see “Controller Initialization” procedure)
- VxWorks must be running

Note: Note that you may successfully ping or rlogin into a controller, yet it may not bootup completely if some files are missing such as nvScript, nvlib.o, & its exec.o file.

Check if master1 and DDR1 have booted completely

To achieve “idle” status, <u>both</u> the master and DDR must bootup. If any other controller does not boot up, the system will still go “idle”.

FOR A CONTROLLER TO BOOTUP & FUNCTION, IT NEEDS

1. Power: 5V & 3.3V for the controller and 7.6V & +/-15V for the mezzanines
2. Certain firmware files: boot.ini, vxWorks405gpr.bdx, nvlib.o, nvScript and X (where X is one execution file such as masterexec.o, rfexec.o, gradientexec.o, pfgexec.o, lockexec.o, ddrexec.o)
   - All files are present to enable them to change function depending on slot location.
3. Aproperly seated IF card (its power LED is on), whose type bits setting determine controller functionality

Use the “Controller Initialization” document to check the bootup messages and re-flash the master1 and/or DDR1 if they do not boot.

IF THE MASTER1 OR DDR1 REBOOT THEMSELVES, CHECK

1. Power to the board is within specs as measured on the side of the power supply.
2. Multiple interrupts from other sources such as the current driver or RFFE board will reset the interrupted controller. This can be monitored by rlogin then typing: *DebugLevel=1*.

   You will clearly see “interrupt….” if that is happening.

If setacq fails, try these items

1. Re-run setacq a 2nd time. The tftp demaon may not have started the first time.
2. Swap network cards: connect the console to the card slot instead of the mother board card.
    - Swap the network info on both cards using the network GUI. Run setacq by using */vnmr/bin/setacq eth0*
3. Connect the host directly to the Ethernet switch (bypassing back connector)
4. Remove any external router or hub
5. Use another Ethernet cable. It can be straight thru or with crossed wires (such as the one with blue ends)
6. Check for TCP wrappers: If the */etc/hosts.deny* file exists, it may list some or all IP addresses that are not allowed to connect to the PC. If it exists, makes sure that the */etc/hosts.allow* file exists having the IP addresses of all controllers & wormhole.
7. If a controller could not be initialized using the "Controller Initialization" procedure, remove the Ethernet cable from this controller so setacq will not hang and can download to the rest of the controllers.

Check if setacq does the following

1. Check /etc/hosts: it should have controllers' IP & name. IP of the controllers is based on IP of eth1.
2. Check /etc/hosts.equiv: it should have all controller names
3. Check /etc/ethers: it should have controllers' hard-coded Ethernet MAC addresses
4. Check if /tftpboot is created. Setacq copies vxWorks405gpr.bdx, all exec.o files, nvlib.o, & nvScript from /vnmr/acq/download. If setacq fails, these files remain in /tftpboot. If setacq completes, /tftpboot should be empty.
5. Check if tftpd is enabled by typing:
    more /etc/xinetd.d/tftp
   should see a line: disable=no
   setacq enables tftpd to be started at host bootup or when you run setacq
   (when setacq fails, you re-run to start tftpd)
6. Check if /etc/notrouter exists: setacq creates it to ensure the console does not function as a router
7. Check if the rarp demon is running by typing:
    ps –ef | grep rarpd
   (should respond with: root ……. /usr/sbin/rarpd –e)
   setacq enables rarpd to start at bootup
   if rarpd is not running you cannot ping some/all controllers or they may not bootup

Last resort: reload Linux OS

If communication with the system cannot be established with the above process, most likely the IS department has modified the software for security reasons. Often times, they made a change a while ago but do not take effect until the host PC is rebooted, which is when the user reports an issue. In this case, it is best to back up any data or configuration/calibration files and reinstall the OS Linux.

Note: Once the system is "idle", a "setup complete" can be achieved only if all controllers have the 80 MHz clock input. You can even achieve setup without the pneumatic box, RFFE box, or MIF connected.

Chapter 5 Shims Supply

VnmrS supports the 14, 23, 28 & 40 channel shim set. It will also support the RRI shim set (Ultrashims –39 in config) in VJ2.1B with patch 2.1BallANYvnm115 or higher. VnmrS imaging supports other shim sets.

5.1 7.1 Equipment needed

The following equipment is required to perform these tests:

1. 9F-9F null modem cable
2. Shim resistor board
3. DVM

5.2 How to trouble shoot a problem with the shim supply

1. Check if master recognizes it with determineShimType
2. Check led status for MIF to shim communication & power
3. Check output current of the shims using the resistor board
4. Other suggestions
5. How to increase the current output of a shim
6. Check Varian (Magnex) RT shim coil power dissipation

5.3 Check if master can read shim EEPROM

- rlogin into master1 & issue the command: determineShimType

  If the EEprom is NOT programmed, the output is

  read EE14 shows -1

  read EE40 shows 2277

Note: In a non-programmed EEPROM, the master can still address the shims properly and shims will respond. Programming simply allows a user to be able to determine the type of shims in the system via the determineShimType command.

If the EEprom is programmed, the output for 28 channel is:

read EE14 shows -1

read EE40 shows 4

value = 4 = 0xa

If the EEprom is programmed, the output for 40 channel is:

read EE14 shows -1

read EE40 shows 9

value = 9 = 0xa

The 4 is a value corresponding to the value of shimset for 28 shims or 9 for 40 and -1 means unknown.

5.4 Check led status to see if MIF communicates with all shims LEDs

12 LEDs are visible through a small opening at the bottom of the shim supply grill. Each time a shim is clicked; a corresponding LED will light & remain ON (not flash as in Inova).

LED indications from right to left on 28 channel shim supply

- 1 (right-most led): z0,1,2,3
- 2: z4, z5, z6, z7
- 3: x, y, zx, zy
- 4: xy, x2y2; xz2, yz2
- 5: x3, y3; zxy, zx2y2
- 6: z3x, z3y, z2x2y2, z2xy
- 7 (left-most led): zx3, zy3, z4x, z4y
- TC: short (if in lock or shim window) ; long (if system idle/acquiring)
- lock on /off: ON when click lock on
- lock hold
- homospoil ON
- 4 Power ok LEDs to 4 shim sections: ax ok, CPU ok, z0 ok, & xv ok (for axial, cpu, z0, & radial power sections) the xv may be off if shim cable not connected to shims

5.5 Check output current of the shims using the resistor board

Measurements with the resistor board

There are 2 different boards that will work: 00-993894-00 is a 32 channel version and 01-905888-00 is the 40 channel version. If you use a 40 channel version on the newer shim power supply you, need to cut out a corner of the board so it does not interfere with the back of the cabinet.

Zs use 0.1 ohms & others use 0.01 ohms resistors on the board

Typical Measurements

- z0 expect ~ 2 mV at max dac=32767
- z1 expect ~ 10 mV (or 20 mV)
- Depending on the version the Z0/Z1 board the voltage will either read approximately 10mV (100mA) or 20 mV (200mA). The –03 z0/z1 was ECO'd 6/06 to have100mA output. (The –02 had 200mA output)
- Old Inova vertical z0/z1 had z1=100mA max. When "low cost shims" came out, z1 max was 200mA & was kept for VNMRS shim supply. Since Magnex coil is stronger, shim algorithm had issues so now z1 max (-03) is down to 100mA.
- z2-z4 expect ~20 mV
- z5-z7 expect ~30 mV
- For All radials expect ~10 mV

5.6 Other checks/suggestions

1. Make sure 5V is at the correct level. When measured at the front panel, it should be 5.1V (side is 5.39V)
   a. If not, locate the right-most pot on the power supply and turn CW until getting the correct level. Symptoms of low voltage include shims resetting with “su” or “ga”, especially when tn=’lk’. The on screen shims show DAC values, but they are being reset. A load=y su restores lock level.
2. Check all 8 fuses plus the AC input fuse
3. Try another controller as master
4. Re-flash the master
5. Ensure continuity of the shim cable
6. Ensure continuity of the shim coil by checking all resistances
7. Re-seat or check integrity of the SPI cable from MIF to shims (any broken pins?)
8. Re-load the VJ software/patches and run setacq

5.7 Increasing shim current output

You can increase the output current of an RT shim to correct an abnormally high shim residual in the magnet. The axial (Zs) resistors are the same as the vertical boards shim supply, but the transverse/radial (non-spin) resistor is different. The shim supply upgrade kit (#0191571900) consists of the following:

0191572000	Upgrade instructions
3123941100	4.7 ohm axial resistor
3123941700	3.3 ohm axial resistor
3140130600	0.5 ohm transverse resistor

Example of increasing z3

You can change the sense resistor to increase the Z3 shim strength. There is enough supply voltage to double the strength.

There is a shim supply upgrade kit part number 0191571900 than contains the required resistors to increase the strength. By changing R104 on the shim driver board from 4.7 to 3.3 ohms will increase the output from 200 to 300mA. Also you can add another 4.7 ohm in parallel to the existing one to double the strength to 400 mA. I also checked the heat dissipation and even at 400 mA the shim heating from Z3 would only be about 1.75 Watts.

The shim supply upgrade kit is available in the Palo Alto service stock. For a quick solution on site you could always switch R104 with R108. R108 is the Z7 sense resistor. Z5,Z6 and Z7 have 3.3 ohm sense resistors giving these channels an output of 300 mA whereas Z2, Z3 and Z4 use 4.7 ohm sense resistors giving a 200 mA current output. Limiting Z7 to 200 mA should not cause you any problems. You could always change it back later after you receive the resistor upgrade kit.

Chapter 6 Software Diagnostic Tools

6.1 Equipment needed

The following equipment is required to perform these tests:

- 9F-9F null modem cable

6.2 To Trace Errors in the Start Up of VNMR

To trace errors in the start up of vnmr:

See example #4 in examples section

- In a terminal window type: vnmr –debug OR vnmrj –debug

6.3 In Linux to spy on processes (for inactive problems)

1. Open a terminal window as root
2. Type: *ps -ef |grep proc* and note the PID # for Sendproc & Recvproc
3. Type: *strace -p* PID# of Sendproc
4. Open another window as root and do the same for the other PID (this is lost on log out)
   The equivalent strace command on Sun/Solaris is *truss*

6.4 Monitor automations events in real-time with Linux 4 or higher

As vnmr1 in a terminal type: (create file called acqlog then all automation run logs will be stored here):

- *touch /vnmr/tmp/acqlog*
- *chmod 666 /vnmr/tmp/acqlog*

Then open a terminal window and type:

- *tail –f /vnmr/tmp/acqlog*

This window will show automation events in real time

Note: Since this acqlog file can grow very large over time, you should reset it by typing this (or delete it & re-create it when needed): cat /dev/null > /vnmr/tmp/acqlog

6.5 To monitor the network communication

1. On Dells you can monitor messages on the network cards eth0 and eth1. As root in a terminal window type:

   tcpdump -i ethX (where X is 0 or 1 depending on which card to monitor)

   You should see various messages going back and forth between the eth0 & LAN as well as between eth1 & the controllers (VnmrS) or acqusition cpu (Inova/MVX)

   See example #5 in examples section

2. On Sun computers you can also monitor communication on the network cards. As root type in the following in a terminal window:
   snoop -d network card name
3. The network card name depends on your Sun model. To find out what you have, type this in a terminal:
   ifconfig –a
4. You should see all network cards on the computer. Snoop reads the messages that the sun's initd sees via the network card; thus,
   a. For the LAN communications: if no messages are shown, there is No communication
   b. For the console: communication is ok when you see various messages such as:
      - "wormhole ----> inova (ip address) .... TCP=# .... ACK=# "
      - "inova (ip address) -----> wormhole ... "
   c. For the console: communication is NOT ok if you see
      - "wormhole (broadcasting) ARP C who is (ip-address) inova?"
   d. This means that there is a connection, but inova cpu is not responding.

6.6 To spy on the controllers use *acqdebug(11)*

See example #1 in examples section:

1. From a terminal window, rlogin to the controller (this can be repeated for any and all controllers)
2. Type acqdebug(11) in the Vnmr command line. This will start debug for all controllers that have been logged into. Alternatively, you can type DebugLevel=1 in the controllers window at the arrow prompt. You will have to do this for each controller.
3. To save the data to a file, type script filename in a terminal window. Then rlogin to the controller from that window.
4. If you are monitoring several controllers at the same time, create a directory in a convenient place. Start the script file for each controller from within that directory, and all the scripts will be in an easy-to-tar location.
5. To end the session, type exit at the controller's arrow prompt, and then cntl-D to end the script file. You will have to do this for each controller you are logged into.

6.7 To spy on proc family use *acqdebug(01)*

See example #2 in examples section

1. In terminal window:
   - su acqproc (to kill)
   - script filename (to start a script file called filename)
   - su acqproc (to start)
   - On the command line type: acqdebug(01)
   - When you are done, save script filename with control-D

6.8 To see commands and/or macros being executed in VnmrJ

See example #3 in examples section:

1. In a terminal window type: tty (note the output number)
2. On the VJ command line type:
   - *jFunc(55, '/dev/pts/#')* [/dev/pts/## is found in step #1]
   - *debug('c')* [lower case 'c' turns on tracing; upper case 'C' turns it off]
3. Run experiments/commands
   - messages are echoed in the terminal window
   - commands or macros that execute properly will show "started" and "returned"

- look for macros or commands that are “aborted”

4. This is also very, very chatty. You can save this output to a file as well. On the command line, type:
   - jFunc(55,’filename’)
5. It will write this file where ever you happen to be, so before typing the command, type pwd in the Vnmr command line to make certain you are in a reasonable (and writeable) spot. Typing cd in the command line before starting is always safe.

6.9 Use tip hardwire (on Sun only) or minicom (on Linux only)

Use these tools to:

- observe the acquisition cpu, msr, or controller bootup
- error messages or info
- to re-initialize the acquisition cpu or controller

For Sun systems use “tip hardwire”

1. Check the /etc/remote file so that line:
   hardwire:\ :dv=/dev/term/X
   Where X is set to port a or b (wherever you connected a cable from the diagnostic port on console to serial port a or b on the Sun).
2. Open terminal window and type:
   tip hardwire
3. Reboot the console to see bootup messages

For all systems with Linux use the “minicom” program

For any system using Linux 4.0 or higher (VnmrS, Inova, or MVX), use the minicom program. On VnmrS you can only initialize a controller using minicom (a “cu” does not allow an abort so you cannot initialize). On Inova or MVX, you need minicom (not “cu”) to view the Motorola CPU bootup & to initialize it.

Minicom is a serial communication program.

TO SET UP MINICOM DO THE FOLLOWING

1. In a terminal as root type
   - minicom -s (setup option to edit the defaults in /etc/minirc.dfl)
     i. hit enter (ignore any warning message)
     ii. hit enter (a window pops out)
     iii. hit enter (new window appears)
2. Scroll to “serial port setup” & hit enter
   - In the new window:
     i. Select “A” to change the serial device to /dev/ttyS0 then hit enter
     ii. Select “E” to change the “bps bits”
   - In the new window:
     i. Select “I” to set speed to (115200 on vnmrs but 9600 on Inova/MVX)
     ii. Select “L” to set parity to none
     iii. Top line should now read “current: 115200 (or 9600) 8N1”
     iv. (if not then also Select “V” to set data bits to 8 Select “W” to set stop bits to 1)
   - Hit enter to exit this window
     i. Select “F” to toggle hardware flow control to NO
     ii. Hit enter to exit
3. Scroll to “modem & dialing” then hit enter
   a. In the new window:

        i. Select “A” & delete all characters to the left but not the dots (periods) then hit enter
        ii. Hit enter to exit this window
    b. Scroll to “save setup as dfl” and hit enter
    c. Scroll to “exit from minicom” and hit enter 2x to get the root prompt
4. At the root prompt type in:
    - chmod 666 /dev/ttyS0 (so that any user can run minicom by accessing the serial port)
5. In a terminal as any user type in
    - minicom -o
    - then hit enter once or twice to get the “Welcome to minicom …”screen with a blinking cursor
    - may see message such as “Parameter xxx is public, but is marked private in global config file” where xxx= bits or parity or stop bits (you see this only if you enter or change the bits, parity, or stop bits in selecting V or W during the minicom setup)

After connecting the host computer serial port to the CPU (Inova/MVX) or controller (VNMRS), press the reset button to view bootup messages, and you can also preempt the bootup by hitting the space bar (to initialize the board).

6.10 Hardware Connection

1. Connect a serial NULL modem cable from the PC serial port to any controller diagnostic port
2. If using a 9F – 9F serial null modem (pin 1 is top right on the longer edge), then the following pins are tied:
    - 2 to 3; 4 to 6; 5 to 5; 7 to 8 (end to end)…also 1 to 6 on same side of connector on both ends of the cable for Data Carrier Detect

6.11 Examples Section

Example #1: Spy on the controllers output

```
[vnmr1@pt21 ~]$ script controller

Script started, file is controller

[vnmr1@pt21 ~]$ minicom -o


Welcome to minicom 2.00.0


OPTIONS: History Buffer, F-key Macros, Search History Buffer, I18n

Compiled on Jun 15 2004, 21:09:30.


Press CTRL-A Z for help on special keys


-> 00:03:09  'nexus.c' line: 369, CntlrPlexus(): SET_DEBUGLEVEL: DebugLevel = 1

00:03:27  'NDDS_SubFuncs.c' line: 56, A new publication took over my Subscripti.00:03:27 'NDDS_SubFuncs.c' line: 365, PubId: 259 (0x103), topic: 'h/rf3/dwnld/'00:03:27  'NDDS_SubFuncs.c' line: 366, Time Sent: 1181085560.352110999, Time Re800:03:27  'NDDS_SubFuncs.c' line: 368, Pub Seq#: 0, 1619, Recv'd Seq#: 0, 1

00:03:27  'NDDS_SubFuncs.c' line: 369, senderAppId: 0x801801, senderHostId: 0xa100:03:27
```

'NDDS_SubFuncs.c' line: 370, status: 'A new issue received'

00:03:27 'downLink.c' line: 179, -------- C_DNWLD_START Received, give Semap800:03:27 'NDDS_SubFuncs.c' line: 365, PubId: 259 (0x103), topic: 'h/rf3/dwnld/'00:03:27 'NDDS_SubFuncs.c' line: 366, Time Sent: 1181085560.352170999, Time Re800:03:27 'NDDS_SubFuncs.c' line: 368, Pub Seq#: 0, 1620, Recv'd Seq#: 0, 2

00:03:27 'NDDS_SubFuncs.c' line: 369, senderAppId: 0x801801, senderHostId: 0xa100:03:27 'NDDS_SubFuncs.c' line: 370, status: 'A new issue received'

00:03:27 'downLink.c' line: 276, Receiving: 'exp7lc0', total size: 396, size s000:03:27 'downLink.c' line: 247, Recv'd: Acodes: 0, Patterns: 0, Tables: 0

00:03:27 'nexus.c' line: 362, CntlrPlexus(): ROLLCALL

00:03:27 'NDDS_SubFuncs.c' line: 365, PubId: 259 (0x103), topic: 'h/rf3/dwnld/'00:03:27 'NDDS_SubFuncs.c' line: 366, Time Sent: 1181085560.364804999, Time Re000:03:27 'NDDS_SubFuncs.c' line: 368, Pub Seq#: 0, 1621, Recv'd Seq#: 0, 3

00:03:27 'NDDS_SubFuncs.c' line: 369, senderAppId: 0x801801, senderHostId: 0xa100:03:27 'NDDS_SubFuncs.c' line: 370, status: 'A new issue received'

00:03:27 'downLink.c' line: 276, Receiving: 'exp7f0', total size: 176, size se000:03:27 'downLink.c' line: 247, Recv'd: Acodes: 1, Patterns: 0, Tables: 0

00:03:27 'NDDS_SubFuncs.c' line: 365, PubId: 259 (0x103), topic: 'h/rf3/dwnld/'00:03:27 'NDDS_SubFuncs.c' line: 366, Time Sent: 1181085560.377510999, Time Re300:03:27 'NDDS_SubFuncs.c' line: 368, Pub Seq#: 0, 1622, Recv'd Seq#: 0, 4

00:03:27 'NDDS_SubFuncs.c' line: 369, senderAppId: 0x801801, senderHostId: 0xa100:03:27 'NDDS_SubFuncs.c' line: 370, status: 'A new issue received'

00:03:27 'downLink.c' line: 202, ---------- C_DNWLD_CMPLT Recieved, give Semae00:03:28 'nexus.c' line: 387, CntlrPlexus(): APARSER: cmd: 5, Acodes: 1, Table'00:03:29 'AParser.c' line: 318, APstart()- ExpName: 'exp7' Num Acodes: 1 Num000:03:29 'AParser.c' line: 328, APstart(): Waiting for at least 1 codes to be d00:03:29 'downLink.c' line: 492, nDownldCodes: 1, nDownldTables: 0, nDownldPat000:03:29 'AParser.c' line: 549, getAcodeSet: Find acode buffer: exp7f0

00:03:29 'A32Interp.c' line: 1235, A32Interp(): pAcodeId->cur_acode_size: 152 s00:03:29 'A32Interp.c' line: 1391, WAIT4ISYNC, postdelay ticks: 320

00:03:29 'shandler.c' line: 381, addPendingSWItr: add Acode 0xacd00004 to pendt00:03:29 'A32BridgeFuncs.c' line: 897, SETVT: type=2 pid=440 temp=30000 tmpoff000:03:29 'shandler.c' line: 381, addPendingSWItr: add Acode 0xacd00004 to pendt00:03:29 'A32Interp.c' line: 2862, RFINFO band=0, xmtr=0, power=-1

00:03:29 'A32Interp.c' line: 2231, END_PARSE 1

00:03:29 'shandler.c' line: 381, addPendingSWItr: add Acode 0x11 to pending lit00:03:29 'cntrlFifoBufObj.c' line: 761, fifoBufForce(): Send Buf: *0x3c87ba0=0100:03:29 'A32BridgeFuncs.c' line: 275, startCntrlFifo(): fifoStarted4Exp: 'NO'00:03:29 'A32BridgeFuncs.c' line: 290, Cntlr used cntrlFifoStartAndSync()

interrupt: =========== Cntrl FIFO - Started Duration: 0 | 1f8

interrupt: ========= Levels Instr: 0, Data: 20

interrupt: =========== Cntrl FIFO - Stopped Duration: 0 | 280

00:03:29 'shandler.c' line: 271, sHandler(): SW Itr 4, signal

00:03:29 'shandler.c' line: 367, Tell Master I'm Ready4sync

00:03:29 'nameClBufs.c' line: 731, nClbFreeByName: 'exp7f0'

00:03:29 'nameClBufs.c' line: 945, nClbFreeByRootName: delete: 'exp7lc0'

00:03:29 'downLink.c' line: 913, number deleted: 1

interrupt: =========== Cntrl FIFO - Started Duration: 0 | 2d8

interrupt: ========= Levels Instr: 0, Data: 16

interrupt: =========== Cntrl FIFO - Stopped Duration: 0 | 3c0

00:03:29 'shandler.c' line: 271, sHandler(): SW Itr 4, signal

00:03:29 'shandler.c' line: 367, Tell Master I'm Ready4sync

interrupt: =========== Cntrl FIFO - Started Duration: 0 | 280

interrupt: ========= Levels Instr: 0, Data: 14

interrupt: =========== Cntrl FIFO - Stopped Duration: 0 | 7a15ec

00:03:30 'shandler.c' line: 257, sHandler(): SW Itr 1, signal

00:03:30 'shandler.c' line: 533, SETUP_CMPLT

00:03:30 'shandler.c' line: 538, >> Fifo Instructions Sent: 41, --> Recv'd: 41000:03:30 'shandler.c' line: 638, Sent Master Setup Completed State

Example #2: Spy on the procs output

```
Script started on Tue 05 Jun 2007 04:32:47 PM PDT

[vnmr1@pt21 ~]$ su acqproc Starting Acquisition communications.

[vnmr1@pt21 ~]$ Expproc states:

Authorization Records:


Record: 0, Count: 0, Level: Acquire, User: '', Host: '', PID: 0

This Process ID is Non-Active


Record: 1, Count: 0, Level: Acquire, User: '', Host: '', PID: 0

This Process ID is Non-Active

16:33:13: 'expQfuncs.c' line: 699,

 Exp Queue, Priority: 0, In Queue: 0, 1st Entry Addr: 0xf72f1004

16:33:13: 'expQfuncs.c' line: 699,

 Exp Queue, Priority: 1, In Queue: 0, 1st Entry Addr: 0xf72f7408

16:33:13: 'expQfuncs.c' line: 1016,

Active Queue: In Queue: 0, 1st Entry Addr: 0xf72fe004

16:33:13: 'recvfuncs.c' line: 1645, debugLevel: New DebugLevel: 1

16:33:13: 'procfuncs.c' line: 448, debugLevel: New DebugLevel: 1

16:33:13: 'sendproc.c' line: 283, processMsge(): received -1 bytes, No Message

16:33:13: 'process.c' line: 80, Procproc: start procQtask()(chk processing Q)

16:33:33: 'expproc.c' line: 246, Expproc: Received SIGIO, processIO()

16:33:33: 'expsocket.c' line: 197, Accepted Sockets in Queue: 1

16:33:33: 'expsocket.c' line: 277, processAcceptSocket: size: 55,  msge: 'add2Qtail
/vnmr/acqqueue/exp7.vnmr1.074427, 7 vnmr1 1,'

16:33:33: 'msgehandler.c' line: 694, add2QTail: filename:
```

```
'/vnmr/acqqueue/exp7.vnmr1.074427', info: ' 7 vnmr1 1'
16:33:33: 'msgehandler.c' line: 402, ChkQ.
16:33:33: 'expQfuncs.c' line: 699,
 Exp Queue, Priority: 0, In Queue: 0, 1st Entry Addr: 0xf72f1004
16:33:33: 'expQfuncs.c' line: 699,
 Exp Queue, Priority: 1, In Queue: 1, 1st Entry Addr: 0xf72f7408
16:33:33: 'expQfuncs.c' line: 705,      (1): ExpId:   '/vnmr/acqqueue/exp7.vnmr1.074427'
16:33:33: 'expQfuncs.c' line: 706,      (1): ExpInfo: ' 7 vnmr1 1'
16:33:33: 'msgehandler.c' line: 424, chkExpQ: 0 Processing Q entries
16:33:33: 'msgehandler.c' line: 425, chkExpQ: 0 Active Processing Q entries
16:33:33: 'semLib.c' line: 433, semOpen: SEMID: 196614
16:33:33: 'ipcMsgQLib.c' line: 126, msgQCreate: created msgId = 32769
16:33:33: 'semLib.c' line: 433, semOpen: SEMID: 196614
16:33:33: 'ipcMsgQLib.c' line: 126, msgQCreate: created msgId = 98307
16:33:33: 'msgehandler.c' line: 458, Send Recvproc: 'recv /vnmr/acqqueue/exp7.vnmr1.074427'
16:33:33: 'msgehandler.c' line: 464, Send Sendproc: 'send /vnmr/acqqueue/exp7.vnmr1.074427
exp7'
16:33:33: 'msgehandler.c' line: 481, chkExpQ: # Acodes: 1, # FIDs: 1,  Jpsg flag: 0,
startAcode: 0
16:33:33: 'recvproc.c' line: 253, Recvproc: Received SIGUSR1, processMsge()
16:33:33: 'msgehandler.c' line: 483, chkExpQ: ActiveExpInfo.ExpInfo->PSGident = 1 (100-JPSG, 1-C PSG)
16:33:33: 'recvproc.c' line: 325, received 37 bytes, MsgInbuf len 37 bytes, Msge: 'recv
/vnmr/acqqueue/exp7.vnmr1.074427'
16:33:33: 'recvfuncs.c' line: 264,
||||||||||||||||||||||||||||||||||||||||||||||||||||


 >>>>>>>>> recvData Invoked.


16:33:33: 'semLib.c' line: 433, semOpen: SEMID: 196614
16:33:33: 'ipcMsgQLib.c' line: 126, msgQCreate: created msgId = 196610
16:33:33: 'semLib.c' line: 433, semOpen: SEMID: 196614
16:33:33: 'ipcMsgQLib.c' line: 126, msgQCreate: created msgId = 163840
16:33:33: 'recvfuncs.c' line: 293,
--------------------


 ----- Preparing for: '/vnmr/acqqueue/exp7.vnmr1.074427' data.
```

Example #3: jFunc output when executing “su” command

[vnmr1@pt21 ~]$ tty

/dev/pts/1

[vnmr1@pt21 ~]$ terminal /dev/pts/1 opened.

command vnmrjcmd started

command vnmrjcmd returned

```
macro ga started
 macro isspincad started
  command exists started
  command exists returned
  command exists started
  command exists returned
  command exists started
  command exists returned
  command exists started
  command exists returned
 macro isspincad returned
 command exists started
 command exists returned
 macro setdsp started
  command exists started
  command exists returned
  command on started
  command on returned
 macro setdsp returned
 command on started
 command on returned
 command exists started
 command exists returned
 command exists started
 command exists returned
 command exists started
 command exists returned
 command exists started
 command exists returned
 command exists started
 command exists returned
 macro reqpartest started
  command exists started
  command exists returned
  command exists started
  command exists returned
  command length started
```

```
 command length returned
 command substr started
 command substr returned
 macro gettoken started
  command length started
  command length returned
  command length started
  command length returned
  command strstr started
  command strstr returned
 macro gettoken returned
 command length started
 command length returned
 command substr started
 command substr returned
 macro gettoken started
  command length started
  command length returned
  command length started
  command length returned
  command strstr started
  command strstr returned
 macro gettoken returned
 command length started
 command length returned
 command substr started
 command substr returned
 macro gettoken started
  command length started
  command length returned
  command length started
  command length returned
  command strstr started
  command strstr returned
 macro gettoken returned
macro reqpartest returned
command exists started
```

command exists returned

command on started

command on returned

command exec started

command Acq started

command Acq returned

command exec returned

command exists started

command exists returned

macro ga returned

Example #4: vnmrj –debug output

[vnmr1@pt21 ~]$ vnmrj -debug

[vnmr1@pt21 ~]$ DEBUG Jun 5

file:/vnmr/shuffler/locator_statements_default.xml DEBUG Jun 5, 2007 16:49:04 open: file:/vnmr/shuffler/lo

DEBUG Jun 5, 2007 16:49:04 open: file:/vnmr/shuffler/locator_statements_protocols.xml

DEBUG Jun 5, 2007 16:49:04 open: file:/vnmr/shuffler/locator_statements_trash.xml

DEBUG Jun 5, 2007 16:49:04 building : /vnmr/templates/vnmrj/interface/VpMenu.xm

DEBUG Jun 5, 2007 16:49:04 building :

/home/vnmr1/vnmrsys/templates/vnmrj/inDEBUG Jun 5, 2007 16:49:04 building :

/home/vnmr1/vnmrsys/templates/vnmrj/inDEBUG Jun 5, 2007 16:49:04 building : /vnmr/templates/vnmrj

DEBUG Jun 5, 2007 16:49:04 building : /vnmr/templates/vnmrj/interface/MainMenu.xm

DEBUG Jun 5, 2007 16:49:04 building : /vnmr/templates/vnmrj/interface/ToolMenu.xml

DEBUG Jun 5, 2007 16:49:04 building : /vnmr/templates/vnmrj/interface/SysToolMenu.x

DEBUG Jun 5, 2007 16:49:05 building : /vnmr/templates/vnmrj/interface/HardwareBar.xml

DEBUG Jun 5, 2007 16:49:05 building : /vnmr/templates/layout/toolPanels/VertPanCryo.xm

DEBUG Jun 5, 2007 16:49:05 building : /vnmr/templates/layout/toolPanels/VpLayout.xml

DEBUG Jun 5, 2007 16:49:05 building : /vnmr/templates/layout/toolPanels/VertPanArray.x

DEBUG Jun 5, 2007 16:49:05 building : /vnmr/templates/layout/toolPanels/Display.xml

DEBUG Jun 5, 2007 16:49:05 building : /vnmr/templates/layout/toolPanels/VertPan1D.xm

DEBUG Jun 5, 2007 16:49:05 building : /vnmr/templates/layout/toolPanels/VertPan2D.xml

DEBUG Jun 5, 2007 16:49:05 building : /vnmr/templates/vnmrj/interface/TopPanel.xml

DEBUG Jun 5, 2007 16:49:07 building : /vnmr/templates/vnmrj/interface/MainMenuFile.

DEBUG Jun 5, 2007 16:49:07 building : /vnmr/templates/vnmrj/interface/MainMenuEdit.xml

DEBUG Jun 5, 2007 16:49:07 building : /vnmr/templates/vnmrj/interface/MainMenuView.xml

DEBUG Jun 5, 2007 16:49:07 building : /vnmr/templates/vnmrj/interface/MainMenuExpts.xml

DEBUG Jun 5, 2007 16:49:07 building : /vnmr/templates/vnmrj/interface/HadamardExpts.xml

DEBUG Jun 5, 2007 16:49:07 File not found: INTERFACE/DPFGSE_Expts.xml

DEBUG Jun 5, 2007 16:49:07 File not found: INTERFACE/ProtonFluorineExpts.xml

DEBUG Jun 5, 2007 16:49:07 File not found: INTERFACE/DosyExpts.xml

DEBUG Jun 5, 2007 16:49:07 File not found: INTERFACE/SelectiveH1Expts.xml

DEBUG Jun 5, 2007 16:49:07 File not found: INTERFACE/ProteinPackWater.xml

DEBUG Jun 5, 2007 16:49:07 File not found: INTERFACE/ProteinPackTriple.xml

DEBUG Jun 5, 2007 16:49:07 File not found: INTERFACE/ProteinPackHSQC.xml

DEBUG Jun 5, 2007 16:49:07 File not found: INTERFACE/RnaPack.xml

DEBUG Jun 5, 2007 16:49:08 building : /vnmr/templates/vnmrj/interface/SolidStateExpts.xml

DEBUG Jun 5, 2007 16:49:08 building : /vnmr/templates/vnmrj/interface/Solids1DExpts.xml

DEBUG Jun 5, 2007 16:49:08 building : /vnmr/templates/vnmrj/interface/HX2DExpts.xml

DEBUG Jun 5, 2007 16:49:08 building : /vnmr/templates/vnmrj/interface/HXYExpts.xml

DEBUG Jun 5, 2007 16:49:08 building : /vnmr/templates/vnmrj/interface/QuadrupoleExpts.xml

DEBUG Jun 5, 2007 16:49:08 building : /vnmr/templates/vnmrj/interface/MultipulseExpts.xml

DEBUG Jun 5, 2007 16:49:08 File not found: INTERFACE/LCNMRMSExpts.xml

DEBUG Jun 5, 2007 16:49:08 File not found: INTERFACE/UserDefinedExpts.xml

DEBUG Jun 5, 2007 16:49:08 building : /vnmr/templates/vnmrj/interface/MainMenuAcquire.xml

DEBUG Jun 5, 2007 16:49:08 building : /vnmr/templates/vnmrj/interface/MainMenuProcess.xml

DEBUG Jun 5, 2007 16:49:08 building : /vnmr/templates/vnmrj/interface/MainMenuTools.xml

DEBUG Jun 5, 2007 16:49:08 building : /vnmr/templates/vnmrj/interface/MainMenuHelp.xml

DEBUG Jun 5, 2007 16:49:08 File not found: INTERFACE/UserDefinedHelp.xml

DEBUG Jun 5, 2007 16:49:08 building : /vnmr/templates/vnmrj/interface/ToolMenuFid.xml

DEBUG Jun 5, 2007 16:49:08 building : /vnmr/templates/vnmrj/interface/ToolMenuSpec1D.xml

DEBUG Jun 5, 2007 16:49:08 building : /vnmr/templates/vnmrj/interface/ToolMenuSpec2D.xml

DEBUG Jun 5, 2007 16:49:08 building : /vnmr/templates/layout/default/sample_action.xml

DEBUG Jun 5, 2007 16:49:08 building : /vnmr/templates/layout/default/sample.xml

DEBUG Jun 5, 2007 16:49:08 building : /vnmr/templates/layout/default/Standard.xml

DEBUG Jun 5, 2007 16:49:08 ### appDir 0 /home/vnmr1/vnmrsys

DEBUG Jun 5, 2007 16:49:08 ### appDir 1 /vnmr

DEBUG Jun 5, 2007 16:49:08 ### appDir 2 /vnmr/autotest

DEBUG Jun 5, 2007 16:49:08 initVp

Example #5: tcpdump output using eth1 on VnmrS

[root@pt5 vnmr1]# tcpdump -i eth1

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode

```
listening on eth1, link-type EN10MB (Ethernet), capture size 96 bytes
12:25:29.980003 IP master1.1039 > wormhole.34595: UDP, length 320
12:25:30.003143 IP rf2.1034 > 225.0.0.15.7401: UDP, length 76
12:25:30.039472 IP rf4.1034 > 225.0.0.15.7401: UDP, length 76
12:25:30.158077 IP master1.1037 > 225.0.0.15.7401: UDP, length 76
12:25:30.272927 IP lock1.1034 > 225.0.0.15.7401: UDP, length 76
12:25:30.282879 IP master1.1039 > wormhole.34595: UDP, length 320
12:25:30.321978 IP pfg1.1034 > 225.0.0.15.7401: UDP, length 76
12:25:30.444555 IP ddr1.1034 > 225.0.0.15.7401: UDP, length 76
```

Chapter 7 Tuning

Several events occur when pushing the tune box button. The RFFE interrupts the master who then activates the meter. Simultaneously, the TX generates a frequency at a fixed amplitude sending it along the transmit path to the probe. The probe's reflected power & LO are processed by the mixer, which generates a signal for the meter to display.

7.1 Equipment needed

The following equipment is required to perform these tests:

- Signal Generator
- DVM
- Oscilloscope

7.2 How does tuning with tether box work?

The RFFE CPLD generates a tune interrupt anytime the tune box switches are pushed. If the CPLD generates the interrupt, you will see the SPI ON led flicker. Also, you can measure approx 4.5V dc at TPS11 (lower left corner on board). This line flickers positive (~4.5V) momentarily (and may be difficult to see on a scope) when any tune switch is pressed. This interrupt is sent to the master controller via the MIF. The master then sends signals on the SPI cable via the MIF to the RFFE CPLD chip. The CPLD chip then sets the 3 tune_ctl line bits, which in turn will activate the meter's power for the backlight and display bars.
If the CPLD generates this interrupt (as seen on TPS11 or the SPI ON led) but the console does not go into tune mode (tune box doesn't light up), then the issue can be the SPI cable, MIF, or master. If the CPLD does not generate an interrupt, then the RFFE board or power input (RFFE power supply) can be the issue.

7.3 What to check in case of tether box tuning problems

Check the TX amplitude

1. TX amplitude is fixed by software by type of RF amplifier. Check if amp is correctly identified:
2. rlogin to rf controller & type AMTbits
    - expect to see this:

Expect Results	Description
AMT 3900-11 Lo-Lo band	for dual 300W 10-245 MHz (600s)
AMT 3900-12 Hi-Lo band	for the 50W 200-500MHz/300W 6-220MHz (500s)
AMT 3900-2 Hi band	for single 200-500 MHz 50W
AMT 3900-1S Lo band	for single 300W 10-245 MHz (500s)
AMT 3900-55 Hi band(100/100W)	
AMT 3900-1 Lo band	for single 300W 6-220MHz (400s)
AMT 3900-1S4 HI-Lo band	for 50W 550-640MHz/ 300W 10-245 MHz (600s)
AMT 3900-1S7 Hi-Lo band(100W)	for 100W 550-640MHz/300W 10-245 MHz (600s)
AMT 3900-15B Hi-Lo band(100W)	for 100W 200-500 /300W 6-220 MHz(300s & 400s)
AMT3900-7 Hi-band	for the 100W used in high band VnmrS 700-900s
Unknown	for the 1kW amps

For all 3900 series amps, by default tune levels are set according to the ID bits of the amplifier. These ID its are on the 25-pin D-type connector. Only the 3900 series has these bits defined. If the amplifier is not identified the software, by default, assumes a 1kW amplifier.

Note: ***For V900s ONLY:***

1. If the AMTbits returns “none”, the amplifier control cable is incorrect or faulty and needs replacement. The cable part number is 01916637xx
2. If working on V900s, please consult with TAC or the website on proper use of tpwrt.

Check the TX frequency

1. TX frequency is a mix of the VFS frequency and the 80 MHz on the backplane

Check attenuator board output

The output is set by “atten” switches in tune box [mtune “power” parameter adjusts attenuator board output]. Each click from 9 to 0 will attenuate the signal by 5 dBm at the attenuator board. Expected levels (may be too small to measure on a scope):

Atten switch position	H1 (dBm / V)	C13 (dBm /V)
9	-39	-42
8	-43	-47
…		
2	-72	-76
1	-77	-82
0	-82	-87

Check the LO amplitude and frequency output from the VFS

1. check config
2. check rf controller or serial cable
3. check LO from transmitter
4. check all required inputs for the TX such as power and 80 MHz on the backplane

Check if the LO @ 1V - 2V is at the mixer (same level in mtune)

If not, check MIF control signals to LO selector, which sends LO into mixer.

Check if the probe tune return signal is at the mixer, if not, do this

1. check tune selector is outputting the correct tune return signal into the mixer
2. check the control signals the mixer gets from RFFE interface (which comes from master)
3. check SPI cable for broken pins (or connector where it plugs into)

Check the directional couplers (DC)

1. Are they correctly oriented? They are last in the TX path right before the probe. Make sure the *arrow points away from the probe* (towards the preamp). The DC for high band is not swappable with the low band. The high band part# ends in 800, but the low band ends in 900.
2. Check cable tightness & integrity.

Check each return signal connection on the tune selector

1. An easy way to remember this is: looking at the tune selector, the 4 BNC ports on one side are ch2 (top port), ch3, ch4, ch1 (bottom port).

7.4 Measurements

Tune Box Amplitude Levels

- Measured on a 500 (with 50W/300W) and 300 (with 100W/300W)
- AMTbits on 50W shows: AMT 3900-12 Hi-Lo band
- AMTbits on 100W shows: AMT 3900-15B Hi-Lo band(100W)

CW tuning:	H1(500)	C13(500)	H1(300)	C13(300)
TX:	3.7dBm/450mV	-1.9dBm/560mV	2.0dB/ 700mV	-2.2dB/ 615mV
atten:	-38 / 5mV	-42 / 6mV	-39/ 7mV	-42.3/ 6mV
AMP	x	x	x	x
H1 preamp:	9 / 720mV	13 / 3V	9.3/ 1.7V	13.0/3.15V
probe				
(thru DC/BPF):	8.8 / 700 mV	12.8 / 2.9V	9.0/1.65V	12.4/ 3V

Test Point Measurements on RFFE interface (NSR) board

Connection	CH switch position	Measure
U7H pin 6 & 14	0	0V
	1	5V
	2	0V
	3	5
(pattern repeats: 0,5,0,5 for next positions)		
Connection	Idle	Tune mode
U3E pin 192	H(4V)	L (0V)
193	L	H
194	H	L
Connection	Idle	Tune mode
Q18- G	H	L
Connection		
TPS11	Flickers momentarily to 4V (then back to 0) when pushing any button on tune box	
TPS11	is where the tune interrupt leaves the RFFE board	

Test Point Measurements on MIF board

On the MIF, TPS2 is the tune interrupt exiting the MIF going to the master (DS5 on MIF also flashes each time the tune box button is pressed).

TPS2 Flicks momentarily to 4V (then back to 0) when pushing any button on the tune box

Tune box in-house systems readings (unloaded with atten switch=9)

Readings are system and frequency dependent

- H1 (300): > 2000 units (off scale)
- C13(300): > 2000 units (off scale)
- H1(500): ~420
- C13(500): ~1000

7.5 General Notes

1. If you power cycle the RFFE or the AC power is flaky, the tune box will NOT light up. You must reset the master controller.
2. Tether Box Tuning & parameters:
    a. Tune Box: position #1 always HB & #2 always BB (regardless of tn/dn parameters)
        i. If tn='H1' & dn='C13', pressing 1 on the tune box channel switch enables H1 tuning using TX1 & pressing 2 enables C13 tuning using TX2
        ii. If tn='C13' & dn='H1', pressing 1 on the tune box channel switch enables H1 tuning using TX1 & pressing 2 enables C13 tuning using TX2
        (TX1 always used for high band tuning & TX2 for low band)
3. Tune meter offset
   When a probe is well tuned, the meter will *not read close to zero* (as on Inova). The "zero" reading will be 40-50 units. This "zero" reading on the meter reflects a ground difference between the RFFE Interface board and the mixer box. Because we have less signal than we did on Inova (directional coupler is ~ 10 dB less), we had to increase the gain on the RFFE Interface board to the tune meter. When we increased the gain by a factor of 10 (so you get good signal swing between tuned and not tuned), this unfortunately also amplified the ground difference. Most systems appear to have 40 to 50 counts of offset. This cannot be calibrated out.
4. Differences between tether box tune & mtune
    a. Tether box tuning
        i. The mixer DC signal to display tune value on the box display & can be measure on RFFE interf TPS16. The DC level varies with the reading on the tune box when the RF cable is removed from the probe. In-house systems showed:
        ii. 85mV with a reading of 1028
        iii. 172 mV with a reading of 1850
        i. 250 mV with a reading of full scale
        ii. A set of parameters is required to be setup
    b. Mtune
        i. The DDR uses the mixer output to digitize the tune signal on screen
        ii. Sweeps the frequency of the VFS (not cw as with tune box) via control from a controller & its IF board loads all parameters needed except you select the channel, nuclei, & scaling parameters (may need tpwr and gain)
        iii. Status LEDs that blink while mtune is running are the remote status TX and ACQ (almost in sync), DDR ds1, RFIF green & yellow where DDS plugs in, MIF ds1 & ds5, and RFFE T/R led.
        iv. Mixed signal passes through IF gain stages in the mixer (but not with tether box)
        v. RF amp: the CH2 "cw" LED is always on regardless of what nuclei is tuned

7.6 mtune trouble shooting

1. Inject 20 MHz @ -20 dBm into DDR IF input while running mtune with gain=60. You should see, after autoscaling, a clean, flat line across the screen, except the edges are jagged.
2. If mtune is running, the following LEDs are blinking about once a second simultaneously:
    - Remote Status: TX & ACQ
    - DDR: DS1
    - RFIF: green & yellow LEDs where PTS/DDS plugs in
    - MIF: DS1 & DS5
    - RFFE: SPI ON & T/R
    - TR gate is always 5V
    - Mixer gate is mainly 5V (mixer off) except for 0V (mixer on) during short TX on time

   If TR gate and mixer gates are swapped, you can acquire NMR & tune with the box, but mtune is all noise (because 5V shutsoff the mixer)

The flicker that happens about once a second indicates that the DDS/PTS is being swept by the RF controller. This is once major difference between mtune & tether box tune. The DDS is swept in mtune, but it is in CW (continuous wave) in tether box tune.

3. If any of these mixer inputs are missing, you will see noise:
   a. P5101 (ribbon cable): mixer power & control:
      - control lines for tune:
        - input select   8   15
        - high band   0V   0v
        - lk   0   5
        - low band   5   0
        - tune   5   5   (same using tune box or mtune)
      - control for mixer select:
        - pin 12 5V= High band; 0V=Low band on 500/600s
        - pin 12 0V for high & low on 200-400s
      - power is on pins 1=+18V & 3= -18V with 2=GND

   b. P5103: LO from LO selector: >450mV & flickers once a sec (signal level depends on the scope & system freq. Using a spectrum analyzer should show correctly at 1.5V – 2V)
      - trace this signal to the LO selector
      - On MIF J11 pin12
      - 0V when tn=H1 or F19
      - 5V when tn= C13 or any low band
      - trace back to the TX being selected: >550mV
      - trace back to the VFS (which is sweeping the freq)

   c. P5111: tune input from tune selector J5502: 6-10 mV using tune box but too small to measure in mtune (3mV), but it flickers once a second
      - tune input sugnal shape resembles mixer output to DDR. Increasing power to 30-40and gain=60 shows a better signal.
      - trace back to the directional coupler (J5511 from high band directional coupler, whose # ends in 800)
      - trace back to the RF amp, attenuator board, TX, VFS (which is sweeping the freq)
   d. J5102: Output to DDR (or DDR IF input)
      - When running mtune set gain=60, power=0, span=20MHz. Disconnect the console RF cables from the probe.
      - If you "T" the DDR IF input to a scope: you should see the *mtune signal is the outline of the positive IF signal you see on the scope. The amplitude depends on power & gainsettings. Expect 3.5Vpp (1 Mohm) with the above settings.*

   e. J5104: Mixer gate turns mixer on & off. The gate is mainly 5V (mixer is off) except during the short time where gate is 0V (mixer is ON). If you remove the gate, the mixer defaults to ON and mtune will still function.

4. H1 transmitter signal:
   - in mtune TX output looks like a pulse that flickers once a second
   - at the TX H1 is >450 mV
   - at the probe H1 is >20mV

5. mtune display:
   - Disconnect the probe. In mtune set power=0, gain=60, span=10 MHz
   - You should see a clean outline. You will see noise if any input to the mixer is missing.

Chapter 8 Variable Temperature Issues

The MIF controls the temperature setting on the VT unit via a serial line. The VT itself controls signals to switch the "high/low" VT valve. When a pneumatic fault occurs (for various reasons), the CPLD in the current driver generates an interrupt to the master to abort the experiment (independent of the settings in the spin/temp panel) and to the VT to shut down.

As long as the pneumatic fault exists (not cleared) or the VT is latched in error, the VT will be in shutdown mode and will not regulate. A red LED (left one) on the VT unit indicates shutdown when you try to set a temp.

8.1 Equipment needed

The following equipment is required to perform these tests:

- DVM
- Oscilloscope

8.2 Things to remember about the VT features in the pneumatic router

1. The purge regulator has a 20 psi pressure sensor. If the air pressure falls below 20 psi (or the regulator is set below 20 psi), the pneumatic router issues a fault.
2. The "pre-reg" and "temp" regulators are in series. In order for the "high temp/low temp" valve to switch properly, the "pre-reg" must be set to 5psi higher (or more) than the "temp" regulator. Set them so that the ball floats near the rate you specify in the vtairflow setting.
3. The VT flow meter is composed of 10 dual color LEDs. Some top & bottom LEDs will be yellow (set by software parameter "vtairlimits"), indicating the upper & lower limits of the VT air flow. The LEDs in the middle are green, indicating the allowed flow rates. A floating ball blocks an infrared emitter/phototransistor pair, so the LED is off where it floats. If the ball crosses into the top or bottom yellow LEDs, the LED becomes red and the pneumatic router issues a fault.

4. The setup/run switch: The normal position must be "run". If in the "run" position, all safety features are in place and protects probe by shutting down the VT if a pneumatic fault occurs. The "setup" position bypasses the safety and should only be used when setting things up at install time.

8.3 Causes of a pneumatic fault (what to check)

Run the "pneuTest" after login into the master1 and select #5:

- Look for the *"pneu fault"* status:
  - If status is "ok", it means the console did not generate a pneumatic fault. In this case the VT unit could be the problem if it shutsdown. To verify this, remove the temp cable from the pneumatic router and set a temp. If the VT still does not regulate (red LED on), the VT is the issue.
  - If the status is "FAULT", then the console did generate a pneumatic fault. The cause can be anything in the list below.

Examples of pneuTest output

```
pt3:vnmr1 1>rlogin master1

-> pneuTest
P n e u   T e s t

1  write/read bearing air    a  Ramp bearing DAC
2  write/read  drive  air    b  Ramp  drive  DAC
3  write/read  eject  air    c  Ramp  eject  DAC
4  write/read    vt   air    d  Ramp    vt   DAC
```

```
5  display status
6  switch to solids
7  walk flow LEDs
8  clear VT threshold error latch
9  sample present status
0  Done.
```

8.4 Below is an example of a normal status

Enter selection:5

Status (0xfbe257):
power supply: OK   (All voltages within 10% tolerance)
close loop: ARMED (setup/run in "run" position)
upper barrel: OLD (new upper barrel is for sensor at top)
liquids meter: PRESENT
solids meter: NOT PRESENT
VT threshold: OK (ball floating within green zone)
NB stack: OK (connection to stack ok)
pressure switch: OK (purge reg>20 psi & switch is closed)
pneu fault: OK (no fault exists)
config: 0x1 (1)
LED state: 0x3df (991)[991=1111011111:shows 5th led from bottom is off where ball floats]
state: LIQUIDS (if solids flow meter present, it's LEDs are off)

Below is an example of a pneumatic fault status caused by pressure switch being open/ faulty and ball crossed into the yellow zone:

Enter selection:5

```
Status (0xfbe3f7):
     power supply: OK
       close loop: ARMED
    upper barrel: OLD
   liquids meter: PRESENT
    solids meter: NOT PRESENT
    VT threshold: EXCEEDED
        NB stack: OK
 pressure switch: FAULT
      pneu fault: FAULT
          config: 0x1 (1)
       LED state: 0x3df (991)
           state: LIQUIDS
```

What to check

1. Ensure the purge regulator is >20psi (on first generation pneumatic box) or main input is >20psi
2. Ensure the 20 psi sensor is in the closed position
3. Check the sensor cable to the current driver (should be a short on J15)
4. Ensure the ball is floating in the middle of the VT flow meter green zone.
5. Ensure that the VT flow meter has no flashing LEDs. If flashing, press "reset pneumatic fault" button to clear.
6. Check the pneumatic router power supply to make sure it's still within tolerance. Test points 8, 11, 12, 13 should all be within +/- 10% tolerance

7. Ensure the VT flow meter does not show a red LED (unless ball is in the yellow zone) It could have a defective LED such that only a red one is lit. If ball is in the yellow zone, change vtairflow to a different value in the spin/temp panel.
8. Check the probe VT connector for proper resistances
    a. pins 1-3: 10-35 ohms is thermocouple resistance
    b. pins 6-7: 20-26 ohms (older probes:18-22) is heater
    c. pins 8-9: 170-180 ohms (older probes: 107-111) is sensor
    d. pins 4 open
    e. pins 2, 5 are shorted on Unibody style probes (TR, AutoX, etc) but they are open in older probes
    f. ensure the wafer is not shorting out (if removed, it must be replaced if damaged with capacitor side in towards the probe)

8.5 VT not switching from "high" to "low" temperature ?

The VT high/low valve is used to switch between "high" and "low" temperature ports. Air should always be flowing from the "high" if temp > vtc or temp is not used. When temp is less than or equal to vtc, air flows from the "low" port. If the valve is not switching, 2 things to check:

1. Ensure the "Pre_temp" regulator is set *5psi or more* than the "Temp" regulator.
2. Ensure if the VT unit is putting 12-18 V at the valve's power connector to switch ports (high to low) when temp is < vtc.
3. Temp cable from VT:
    - pin 8 is ground
    - pin 15 is 20V (always present) & measured with pin 8
    - pin 1 is floating & gets grounded to activate valve when temp<vtc. Don't disconnect the cable because it can cause the VT to lose its 20V (without the 20V, the hi/low relay will not switch.
4. The VT hi/low valve has a label (cannot see due to its orientation, but can pull bottom ring & rotate bottom portion) & should read "code=12J, 12V, 0.17A.

8.6 No air from high or low temp ports

If no air is coming out at all from any port, you must check the current driver board inside the pneumatic driver box. Test point J3 should have 9-12V and TP2 should have 5V to power the VT solenoid. Air from the VT solenoid passes through the VT flow meter (color LEDs) & then to the VT high/low valve out to the probe.

8.7 How to isolate if the VT unit is faulty

If you remove the "temp" cable from the pneumatic router, this removes the pneumatic fault to the VT. You should be able to "reset VT" and set a temp (make sure air is flowing to the probe). If the VT does regulate, the VT is ok (the console was causing it to shutdown due to a pneumatic fault when the cable was connected). If the VT does not regulate (shuts down), then it may be faulty.
By pressing the left arrow once, then the up arrow 29 times, you will get to the "last error" screen. If it shows "ext swch", this means either of the 2 external shutdown switches has closed (i.e, the console sent a shutdown signal due to a pneumatic fault).

8.8 How to clear pneumatic faults

If the power goes out

If the power goes out, VT solenoid closes (no air from high or low temp ports) & all LEDs will be off. After power returns, the ball will not be floating & all LEDs will be green. To restore operation, click buttons in this order:
1. press "reset VT" to start airflow and ball floats in the green zone

2. press “reset pneumatic fault” to clear flashing LEDs
3. then “su” to get “setup complete” and be able to set a temperature

If the air goes out

If air cuts out in the middle of an experiment, the ball will drop into the yellow zone. This triggers a pneumatic fault to abort the experiment and shutdown the VT. If the air returns, the VT will still be *latched in error*. The ball will be floating, but the LEDs on the VT flow meter will be flashing to indicate that a fault occurred and *needs to be cleared*. To restore operation, do the following:

1. ensure ball floating in the middle of the green zone
2. click “reset VT” button which does the following:
    a. remote status display for VT should show “pass” then room temp
    b. remote status VT green LED stops flashing & turns off
    c. removes latched error on VT so red (left) LED on VT turns off
3. click “reset pneumatic fault” button to clear the fault/flashing LEDs
4. then “su” to get “setup complete” and be able to set a temperature

Common error messages or problems:	How to clear/resolve:
“pneumatic pressure fault (<20 psi)”	• Increase “purge” regulator to >25 psi • Ensure pressure sensor is closed • Ensure J15 is a short on current driver
“vt airflow threshold exceeded”	• Ensure ball is floating in the green zone, not the upper/lower yellow zones • Ensure LEDs are not flashing by pressing reset pneumatic fault
“narrow bore stack fault”	• Ensure 15 pin cable is connected to upper stack • In wide bores ensure the shorting plug is installed on the pneumatic router
“VT controller has no gas flow”	• This occurs if a fault occurred & then cleared, but the VT is still latched in error when trying to set a temp. Press “reset VT” to clear the VT then set a temp.
“VT controller main sensor on top limit”	• This occurs when trying to set a temp that is above the “max temp’ setting of the VT. Increase the max temp setting.
“Narrow bore stack fault”	• Ensure 15pin cable is attached from the upper barrel to pneumatic router • Ensure the VT shorting plug is used on pneumatic router upper stack connection in wide bore solids • Correct problem with narrow bore solids tack getting too hot or cold
VT does not shutdown after a fault	• Ensure ‘setup/run’ is in the ‘run’ position
VT flow meter LEDs flashing	• Ensure ball is within the green zone and press “reset pneumatic fault” button
VT gas exiting from the high/low	• Ensure that high & low temp lines are
valve inside the box	• NOT tied together. Refer to install instructions in pneumatic manual