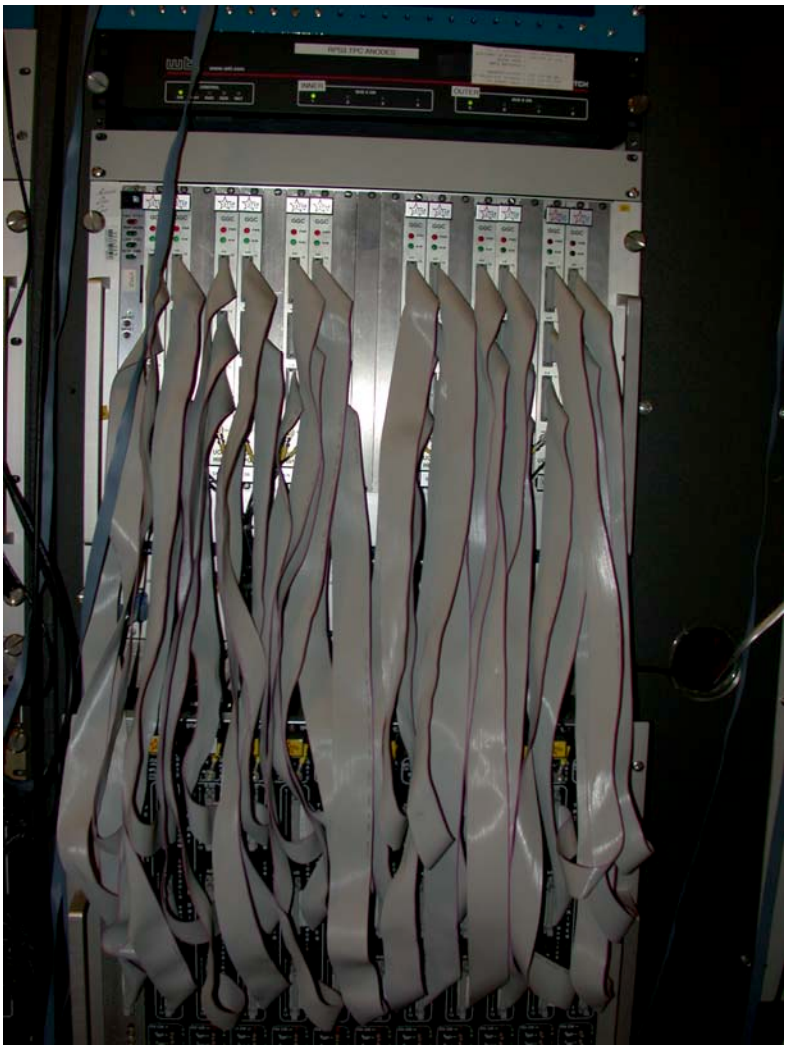
GATED GRID

HARDWARE

The gated grid system consists of two crates located in Rack Row 2A6: a 6U VME crate for the processor and 12 control modules and a special 9U crate for the actual GG drivers (also 12 modules). See picture:



The control crate is at the top, with the processor in the left hand slot. A set of three ribbon cables go from each control module to the corresponding driver module in the crate below. In addition there is a lemo cable that goes from each control module to each driver – this is the “gate open” signal (TTL), whose width determines the gate open time (typically 39 microseconds.) The primary gate open signal comes from the first floor platform as an output from the TPC TCD (rack 1A2). This NIM signal plugs into the left- most control module, which converts it to TTL and puts it on the VME back plane, where the other 11 control modules pick it up. Each control module then outputs the TTL signal to its corresponding driver. Note that this means that the left most control module is different from the other eleven – it has the additional NIM input lemo. We thus have two flavors of spares.

The gate open signal comes from the TPC TCD in Rack 1A2. The signal width (gate open) is set by DAQ software and is output from a front panel Lemo labeled “Grid Out”. Since we ran a mixed TPC and TPX system this year and the TPX had its own TCD I “ORed” the two Grid Out signals and then level shifted this OR to NIM. This was done in a NIM bin just above the TCD crate. If all the old TPC electronics is replaced before Run 9, this OR can be removed (but the level shifter has to stay!).

Each of the grid driver modules has four individual outputs, each of which drives one sector. The output cables are connected inside the modules on screw lugs, pass through a grommit on the back panel and then terminate after ~ 2 ft with a LEMO twinex connector. These are in turn connected to the 100 ft twinex cables which go to the face of the TPC (one cable per sector). From the front of the crate, the first four modules are for outer sectors 1-12, the next four are inner 1-12, the next four are outer 13-24, and the last four inner 13-24.

The gate open voltage (i.e., the voltage for both sets of wires when the gate is open) is determined by the TPC geometry and the cathode operating voltage. To maintain the uniform electric drift field the voltage on the gate needs to match the field cage at that point in z. Since we run the cathode at - 28 kV it has been calculated that the gate open voltage is -115 V. For documentation on this geometry and calculation see the TPC hardware web pages:

http://www.star.bnl.gov/public/tpc/tpc.html

And click on Hardware, then “Drift Defining Hardware”, then “Tuning the Anode planes” or “Gating Grid/MWC”

Tests were made initially to determine what the swing voltage needed to be to completely close the gate 100%, with some safety margin. This swing voltage has been set to plus/minus 75 volts for all runs since 2000. Thus alternating wires have -40 and -190 volts when the gate is closed.

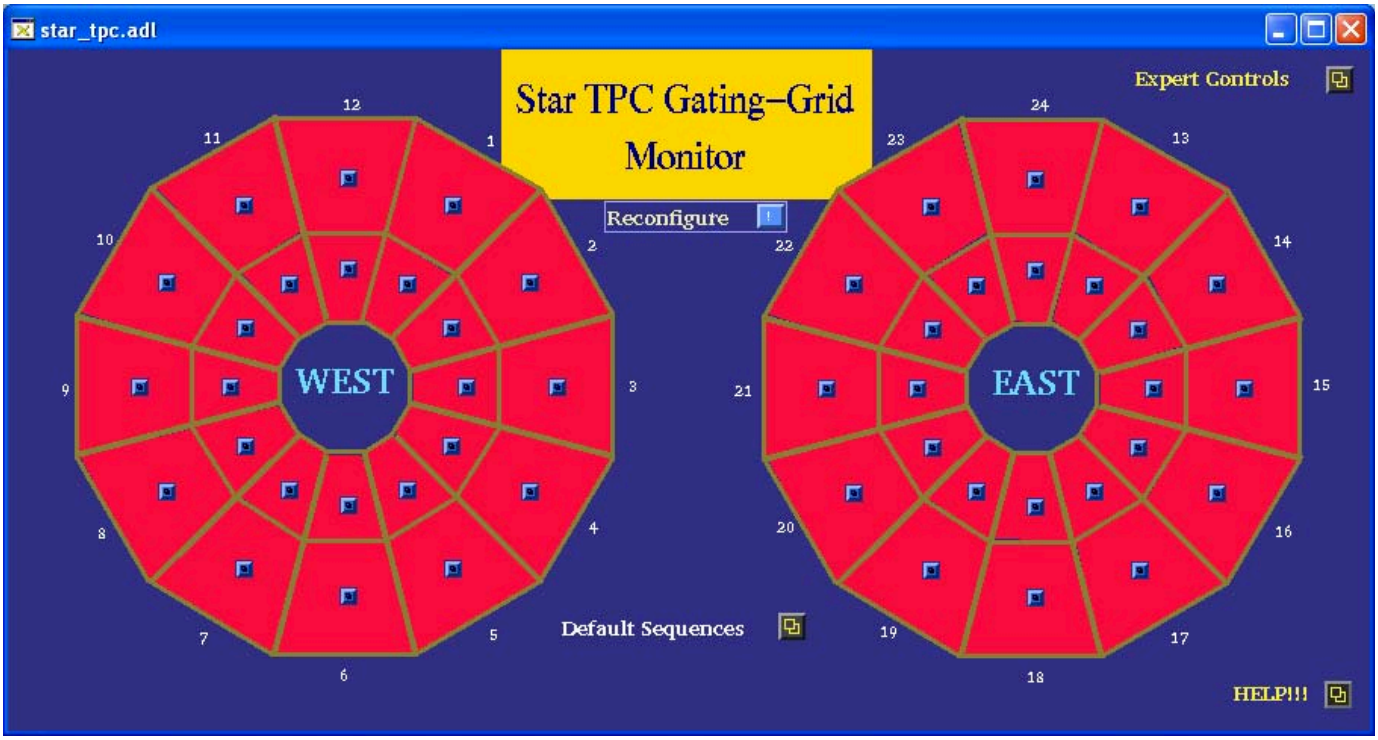
For monitoring purposes and when needed for calibration we have a breakout box which is kept inside rack 2A6. It consists of a box with female and male twinex Lemo connectors where one can plug in the cable from the driver and the cable from the chamber. It also has three banana plug test points for a DVM and two probe hoods for a scope probe. Thus the voltages can be measured safely when the chamber is still attached. Risetimes and ringing can also be viewed with a scope.

In addition, there are lemo monitor outputs on the front of each driver card for the four channels – this is useful for checking the gate width etc. The monitor output is 100:1. See pages 85-86 in my notebook #1.

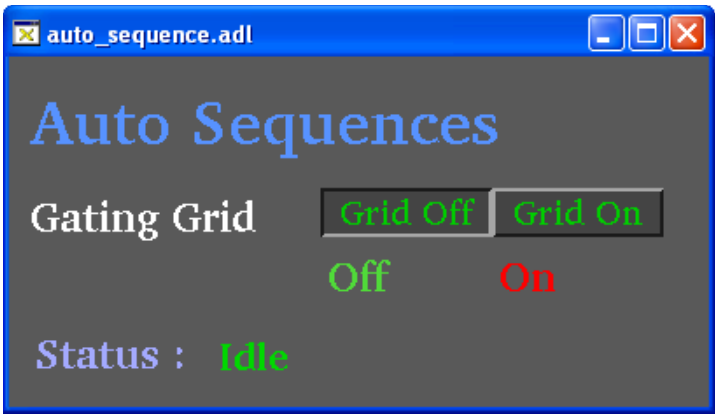
The gated grid is interlocked with the TPC AB interlock system – the GG voltage is turned off for gas system problems, water leaks etc. The interlock connection comes from the TPC AB system in Rack 2A8 and connects to a DB15 connector on the fan tray of the driver crate. This interlock can be over ridden by using a key on the main AB panel in the gas room for testing purposes if there is no gas in the TPC.

SLOW CONTROLS & GUI

The gated grid control GUI is accessible from the top level TPC GUI:

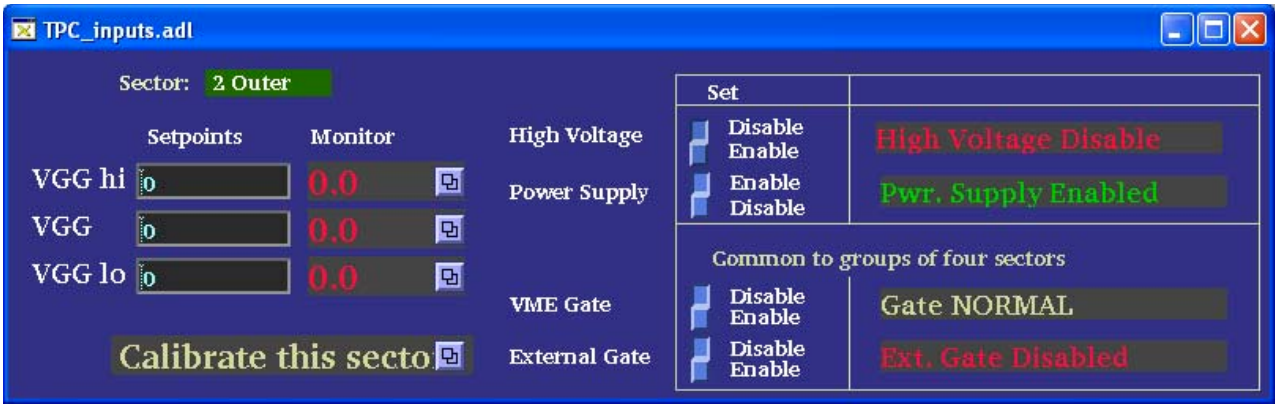


When starting up the voltage is initially off. The GG processor is connected to port 9002 on the terminal server. To turn the GG voltages on, click on the “Default Sequences” button on the GG GUI:



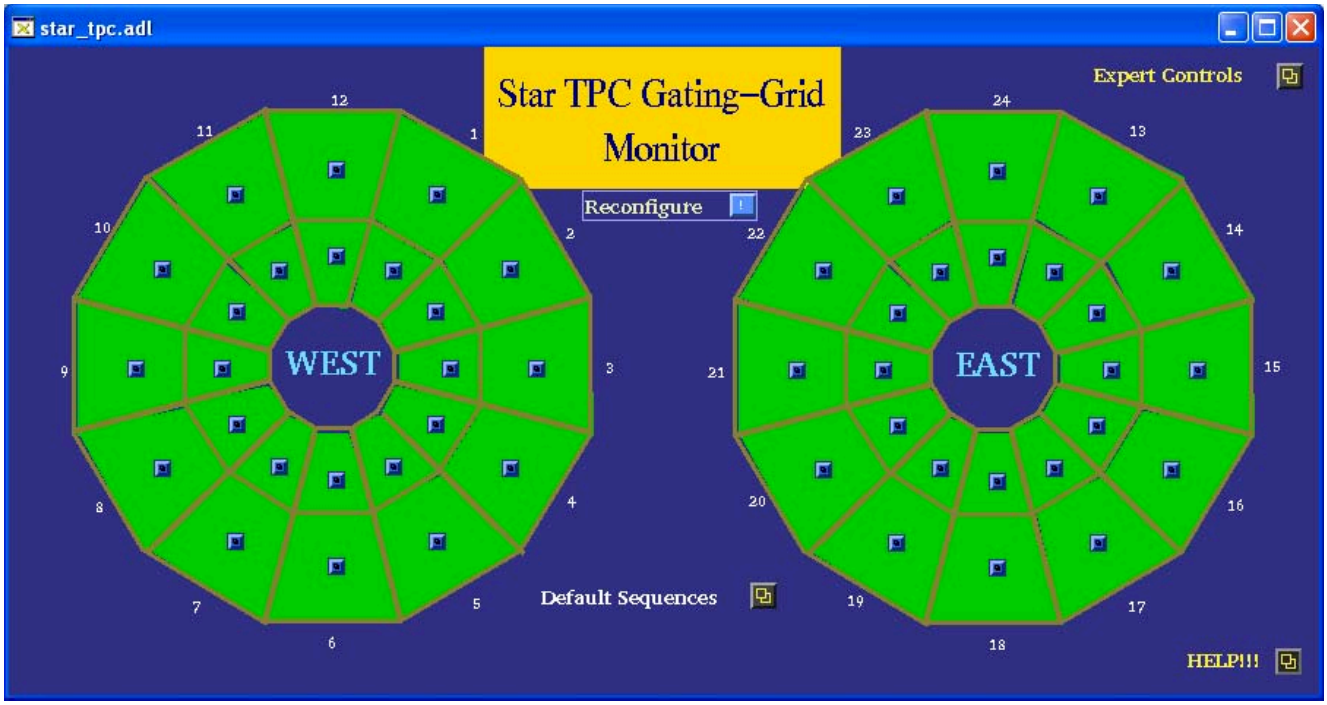
To turn the gating grid voltage on (all sectors), click on the “Grid On” button. If the processor is alive (see below), the status should change to “Downloading Setpoints” and the slow controls script will set all the driver voltages on in turn and enable the gate. If the Status continues to read “Idle”, then you’ll need to reboot the processor and then try again.

The turn on sequence takes a few minutes – to watch the progress, click on one of the buttons in the TPC sector display on the GG GUI. This brings up the sector control GUI:



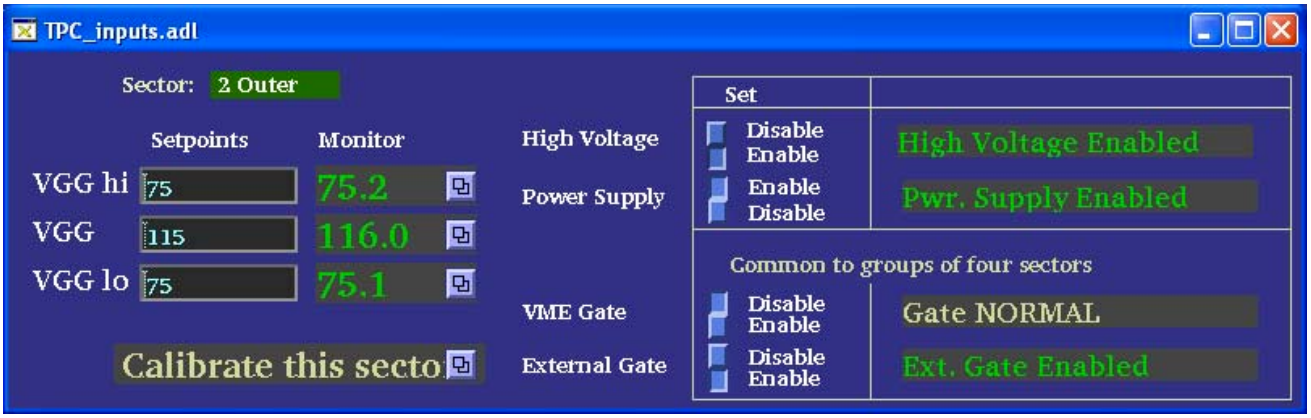
You should see the script input the voltage setpoints (VGG hi & VGG lo = 75, VGG = 115 and see the voltage ramp up in the monitor window. The High Voltage should change to enabled and the Ext. Gate should change to Enabled.

When all the sectors have been set, the main GG GUI should turn green:

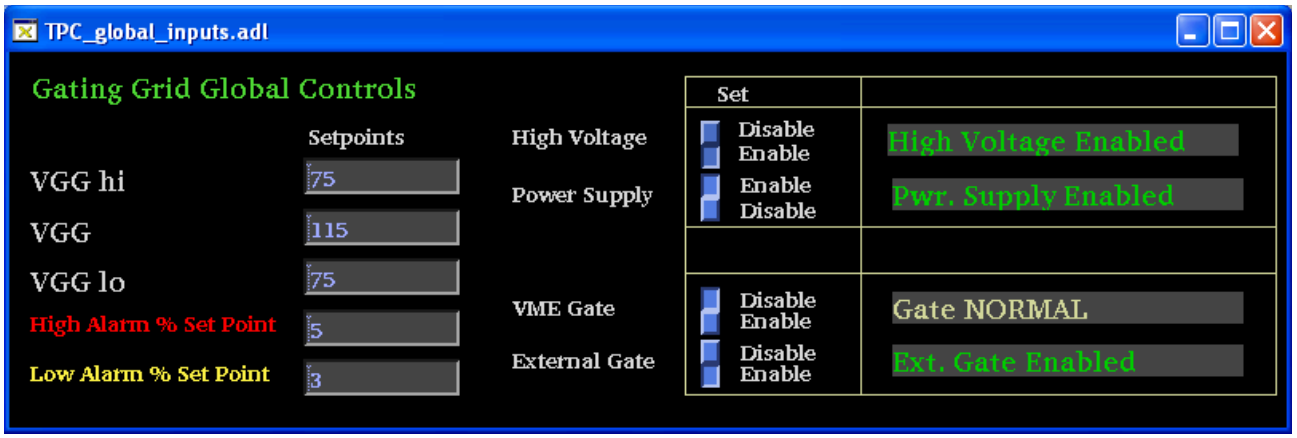


The GUIs for a typical sector and for the global control are shown below for normal operation:

Single sector control GUI:



Global control for all sectors:



For normal running, the gated grid voltages are left on all the time. It is not necessary to turn them off between fills. The GG voltages are monitored by slow controls and will alarm if they drift from nominal. The current criteria are a yellow alarm for a 3% excursion and a red alarm for 5%. Studies originally done by Geno Yamamoto determined that a 3% excursion would not cause a distortion that could be seen in the data, so we can keep running with a yellow alarm. A red alarm would cause noticeable problems with the data, so data taking should stop until the problem is fixed.

For normal data taking in Runs 7 and 8 the gated grid was driven at sustained rates of up to 300 Hz, with no problems. Note that rare triggers that issue a level 0 trigger, but then abort at level 2, cause the gate to be opened, even if the event was discarded. This is because the level 2 decision comes after the 40 microsec drift time of the TPC. Hence, the total GG rate is the TPC DAQ rate plus the level 2 abort rate. It's the total GG rate that determines the charge loading in the TPC and hence the lifetime of the wires.

At the end of Run 8 we did tests of the TPX electronics and the GG was driven at rates up to 2000 Hz. (Note that in a mixed system, with only some sectors having TPX electronics, the gate is opened for ALL sectors even for TPX only events.) For the high rate tests the charge loading scaled linearly with the trigger rate. See page 36 of my notebook #2.

EXPERT OPERATIONS:

Normally, the gated grid system needs no operator changes once it is turned on and the GUI is all green. However, for special calibration runs or to calibrate one of the voltages, there are some expert only controls. To access these controls, click on the “Expert Controls” button on the Gated Grid GUI. This brings up the GUI:

In order to make changes to the voltages (global or individual sector) one must first click on the “Write Disabled” button on the Expert GUI. Then:

One can now use the global control or a single sector control to input a new set of Vgg voltages (e.g. if runs will be taken at a different cathode voltage) or to open and close the gate during the calibration procedure (see below). When the changes have been made don’t forget to click on the “Write Enabled” button to change it back to “Disabled”.

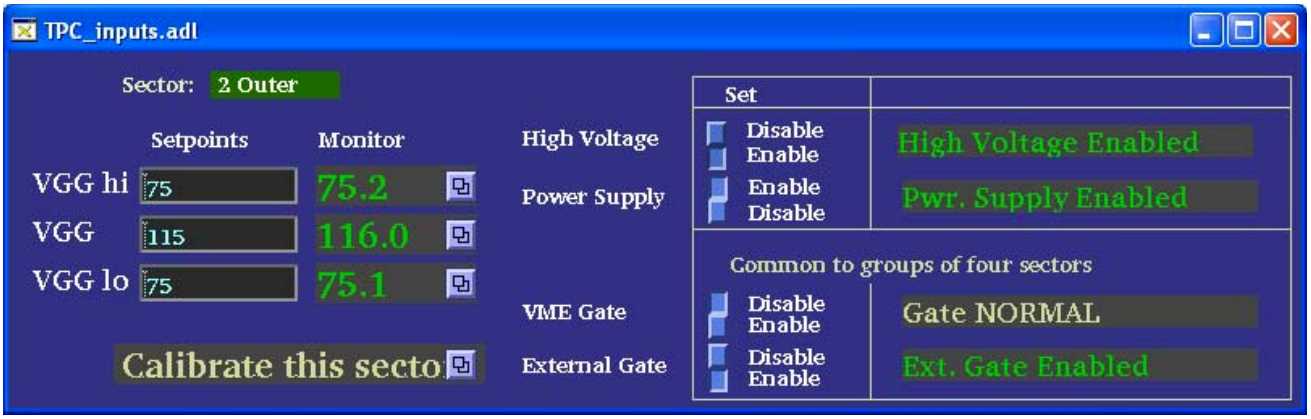
We used to have problems with the stability of some of the GG voltages, especially after long shutdowns. Most of these problems were eventually traced to the ribbon cables that connect the control modules to the drivers. These cables were replaced in 2006 and we have had virtually no problems with wandering voltages since. However, it may be necessary to occasionally recalibrate a drifting voltage.

As stated above a voltage that drifts 3% from nominal will cause a yellow alarm, but data taking can continue in this case. A 5% drift probably causes enough of a distortion in the data that it needs to be fixed as soon as possible.

To recalibrate a voltage that has drifted, follow this procedure:

1. Gain access to the platform.

2. Bring up the GUI for the sector that is drifting:



(For purposes of illustration we'll assume the readback (monitor) voltage for VGG hi reads 70 V and has turned red.

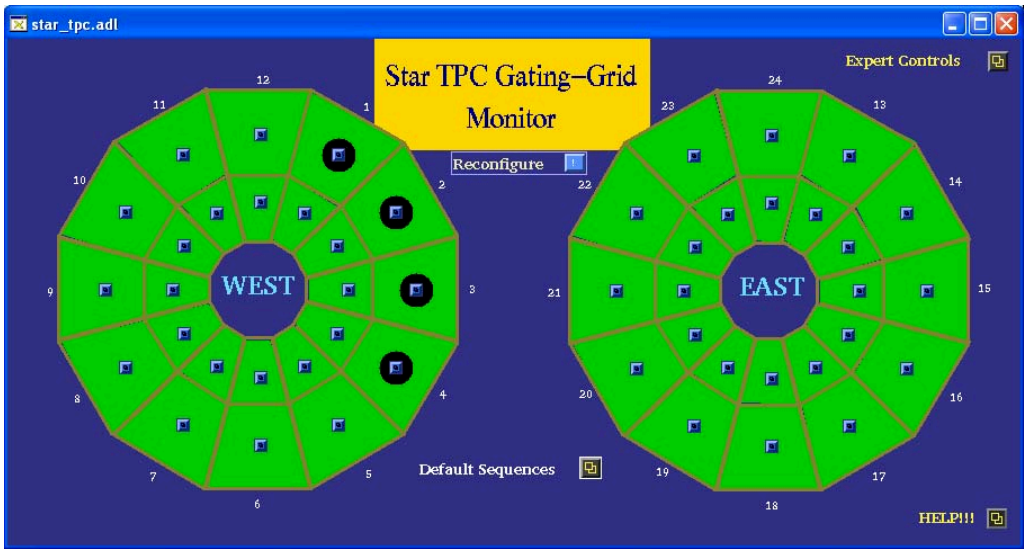
3. Make sure that the button in the Experts GUI says "Write Enable"

4. Record the readings of the three voltages shown in the Monitor window: i.e. VGG hi, VGG, and VGG lo).

5. In back of Rack 2A6 find the breakout box and the cable for the bad sector (here 2 Outer). Disconnect the driver pigtail from the 100' cable and plug the two ends into the breakout box.

6. Using a DVM measure the voltage between VGG hi and ground, and between VGG lo and ground – record these values.

7. Using the GUI for the bad sector click on the enable button for the VME Gate. It should change from "Gate Normal" to "Gate Open" – this opens the gate for all four sectors of that driver module. (Because of the modulo 4 nature of the controls, you can't open the gate for only one channel.) The GG GUI should now look like:



This shows Sectors 1-4 Outer have their gate in the open state.

8. Using the breakout box, again measure from VGG hi to ground and VGG lo to ground – since the gate is open they should both read the same (canonical = -115 V). Record this value.

9.Note that there are two possibilities for a drifting voltage:

The DAC which sets the voltage could be wrong, in which case the voltage you measure with the DVM will agree with the voltage measured on the GUI i.e. they both read something other than the nominal value.

The ADC which reads the voltage back could be wrong, in which case you will measure a correct voltage with the DVM (i.e. 75 volts), and the monitor readback on the GUI is wrong.

The only way to distinguish these two cases is to measure using the breakout box.

10.On the sector control GUI, click on the “Calibrate This Sector” button. This brings up the calibrate GUI:

11. Input the absolute value of the three MEASURED voltages that you recorded from the breakout box (make sure to hit enter after each entry) and then click on “Calibrate this sector”. The program should then recalibrate the drifting channel and save the new scale factors in the data base.

12. Check the sector control GUI again to confirm that all voltages now read nominal. If the calibration didn’t work, remeasure all voltages and try again.

13. When finished, make sure the gate is set back to “Normal” and the Expert GUI has been set back to “Write Disabled”.

It sometimes happens that a voltage has drifted far from nominal and the automatic calibration program is unable to bring it back. In such cases you can try replacing either the control or driver module or revert to the old calibration method which requires inserting new constants into the GG database by hand (see below for Geno’s original writeup.)

STARTUP PROCEDURE

As part of the startup check list for each new run we have developed a method to check if the GG cables are still attached to the TPC sectors. It is especially important to perform this check if cables have been installed or removed for other detectors. Note that this procedure was developed after we discovered that one sector was not cabled for a data run, necessitating a pole tip removal to fix it.

The method consists of measuring the capacitance between each twinex pin and ground and between the two pins, for all sectors. The check is done on the platform by breaking the connection between the pigtail and the long cable in the back of the driver crate. The blue TPC portable Fluke meter will measure capacitance to the needed accuracy. The canonical values are listed below. The measurement is made at the connector of the long cable – if the other end is not attached the reading will be different from that listed for wach sector.

Once this check is finished, it is usually sufficient to turn both crates on and turn on the voltages to check for uncalibrated channels.

TWO KNOWN PROBLEMS

1. It was discovered before run 7 that Sector 8 outer had an additional grid leak besides the known one between pad rows 13 and 14. It appeared that, in two places, some GG wires were possibly not connected to the buss. Experimentally we found that reversing the polarity for this sector helped to close this leak. We therefore installed a short polarity reversing cable for that sector between the driver pigtail and the long cable. This extra cable has an identifying tag. No other sectors were found that had this problem.

2. For some unknown reason, the VME processor for the GG seems to “go to sleep” after ~ some hours after a reboot. Various slow controls experts have looked at this but found no solution – we have even upgraded the processor once. It could be a slow controls problem or a memory leak or ? Fortunately, the processor continues to monitor the voltages, so alarms will still be generated for a drifting value. However the processor does NOT respond to any commands from the GUI, so one can’t change values, calibrate, or even turn the system off. The only way to recover control is to reboot the processor (which by definition turns the voltage off.) Since the GG system has been stable for the last 2 runs this “feature” has not caused too much grief.

SPARES & REPAIR

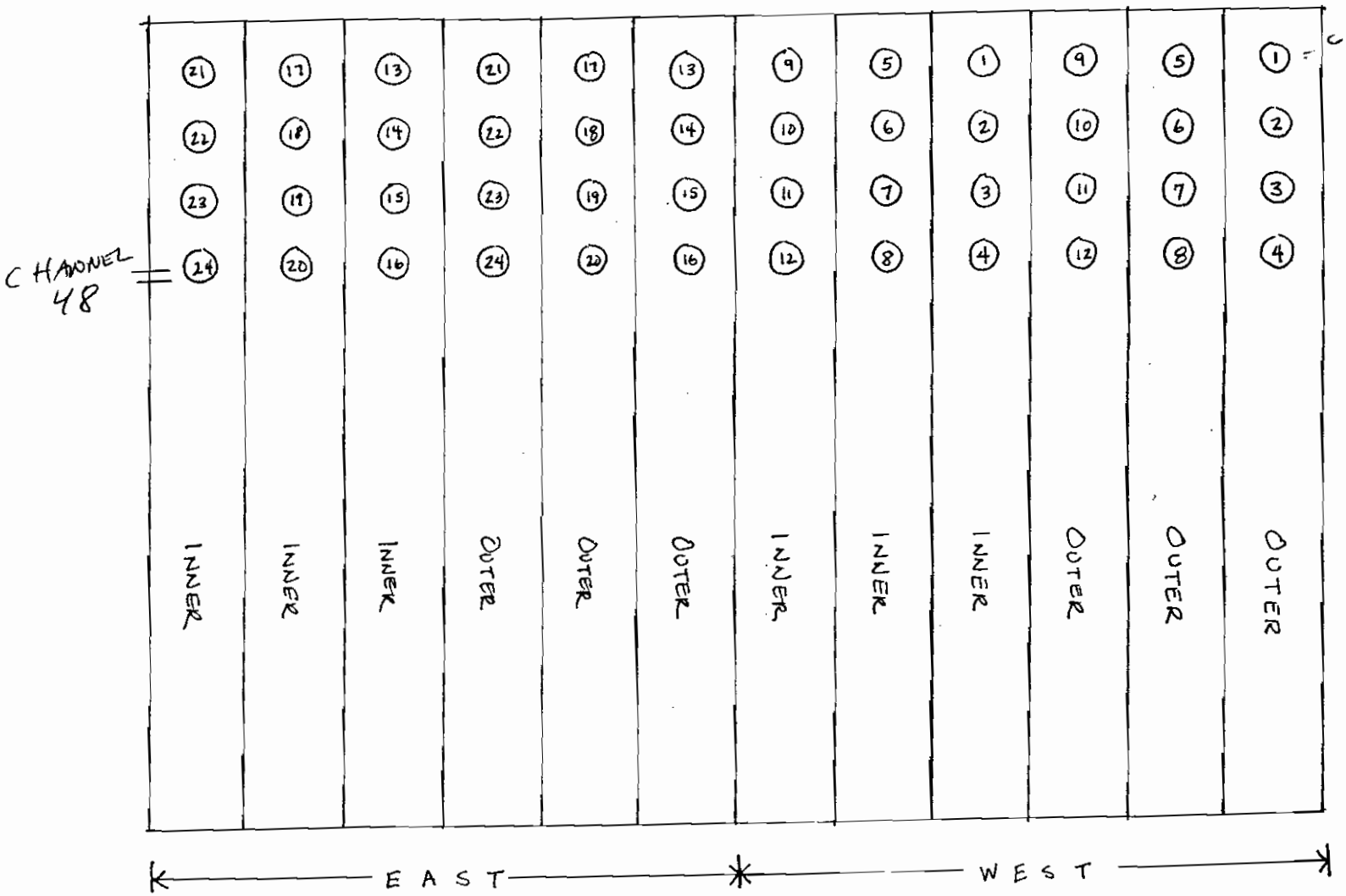
The GG system was originally built by Vahe Ghazikhanian from UCLA, and he also did repairs. He has now left UCLA, but the tech who did the repairs is still there, so there is some hope. Spare drivers and control modules (2 flavors) are in the TPC spares cabinet.

I have not replaced a module for 2 years, but the GG rate will increase for the DAQ1000 era.

HISTORICAL DOCUMENTS

Below are some documents for the GG, including the capacitance measurements, Geno’s original write-ups, Dennis Reichhold’s calibration code etc.

GATING GRID DRIVER: HV OUTPUT MAPPING (REAR VIEW)



LEGEND:

(N) ← HIGH-VOLTAGE CABLE FOR SECTOR N.

"OUTER" ← OUTER SECTOR

"INNER" ← INNER SECTOR

"WEST" ← WEST END OF TPC

"EAST" ← EAST END OF TPC

E. YAMAMOTO

Gating Grid Capacitance To Ground

Measured 11 July, 2000

Eugene T. Yamamoto
geno@physics.ucla.edu

Capacitance Measured between each pin and the ground sleeve. Pin 1 is pin closest to connector key.

Sector Number	Capacitance (nF) Outer Pin 1	Outer Pin 2	Inner Pin1	Inner Pin 2	
1	5.85	5.94	5.4	5.43	
2	5.87	5.87	5.32	5.42	
3	5.88	5.95	5.295.34	5.395.29	- 3/30/01?
4	5.89	5.9	5.34	5.48	
5	5.93	6.01	5.39	5.41	
6	5.89	5.97	5.4	5.46	
7	5.89	5.83	5.36	5.42	
8	5.89	5.97	5.4	5.46	
9	5.86	5.96	5.37	5.44	
10	5.9	6	5.32	5.48	
11	5.9	6	5.35	5.42	
12	5.9	6	5.4	5.45	
13	5.89	6	5.37	5.44	
14	5.84	5.95	5.29	5.47	
15	5.92	6.02	5.33	5.41	
16	5.87	5.98	5.27	5.44	
17	5.89	5.99	5.28	5.44	
18	6.14	6.26	5.3	5.46	Outer Sector. Long Ca
19	5.88	5.98	5.31	5.47	CABLE
20	5.84	5.96	5.33	5.42	
21	5.81	5.91	5.33	5.41	
22	5.83	5.94	5.12	5.2	Inner Sector. Short Ca
23	5.79	5.88	5.28	5.45	CABLE
24	5.8	5.93	5.33	5.42	
Mean	5.88125	5.966667	5.328333	5.42875	
sDev	0.065959	0.078777	0.06084	0.054479	

Measured 11 July, 200
Eugene T. Yamamoto
geno@physics.ucla.edu
Capacitance Measured between two pins of twinX cable
at the driver modules.

Sector Number	Capacitance (nF) Outer	Inner	
1	5.47	4.24	
2	5.5	4.21	
3	5.52	4.2	
4	5.46	4.24	
5	5.54	4.23	
6	5.53	4.26	
7	5.45	4.23	
8	5.52	4.26	
9	5.52	4.27	
10	5.53	4.25	
11	5.5	4.25	
12	5.53	4.26	
13	5.53	4.24	
14	5.52	4.23	
15	5.53	4.22	
16	5.53	4.23	
17	5.49	4.24	
18	5.66	4.24	Outer Sector. Long Cable
19	5.52	4.25	
20	5.48	4.23	
21	5.48	4.23	
22	5.53	4.11	Inner Sector. Short Cable
23	5.46	4.22	
24	5.48	4.24	
Mean	5.511667	4.2325	
sDev	0.041772	0.030822	

Replacing a Gating Grid (GG) module:

1) Make sure gating grid is off and the crates are unplugged from the wall socket. The crates are in rack 2A6.

2)Take cables off the front of the offending module and the module to the left and right of it. This is a necessary precaution to prevent pinching/cutting the cables when installing the replacement module.

3) Go to the back of the rack and unhook the HV cables from the TPC. Do the following: Slide the black insulation DOWN. Slide the connector sleeve as you are pulling the cable. This should disconnect the cable.

4) There are two screws holding the module in the crate. They reside on the top and bottom of the module. Unscrew these.

5) Slide module out.

6) To slide the replacement module in, grab the four HV cables in the back and place them as far back in the now open slot as possible. After doing that, slide the module partially in (1/3 of the way).

7) Go to the back of the crate, and grab the HV cables and pull them out. You can now pull the module GENTLY into place. Make sure the cables in front are out of the way and not getting pinched. This is a common problem. Often times there will not be quite enough space to get the module in place all the way. DO NOT FORCE the module into place by pulling the HV wires. Instead, unscrew all the modules on one side of the replacement module. Pull these out part way. Since there is extra space at the ends of the crate, this will allow you to put in the replacement module, assuming it is somewhere in the middle. After you have slide in the replacement module, slide modules in starting with the module that is farthest from the end of the crate.

7) Reattach cables and make sure that the black insulation covers the ground sleeve on the high voltage cables.

What to do if the Gating Grid Driver acts up:

This file lists some of the symptoms that I have encountered in the gating grid and what they have meant in the past. If a channel ever begins to act up:

please note the problem in the TPC run log.

Send e-mail to Vahe Ghazikhanian (vahe@physics.ucla.edu) and Eugene Yamamoto (geno@physics.ucla.edu), and include a copy of the log entry. Please be as descriptive as possible, including the setpoints for each channel in the offending sector, the status of the high voltage, power supply, gates. Also, information about the sector's behavior when the HV and/or power supply is cycled will help. (e.g. When I enable HV, Vgg shoots up to 297 volts. Vgg does not respond to my voltage requests.)

Some easy to solve problems with the gating grid driver.

I look at the GUI for the gating grid driver and all the sectors are white.

Problem: The gating grid control crate is not turned on.

Solution: Make sure both gating grid crates are on.

I look at the GUI for the gating grid driver and sector 1-outer is white.

Problem: mis-initialization of sector 1 power supply.

Solution: Manually enable power supply for that sector.

I did a startup sequence under global general controls. None of the sectors are on.

Problem: The gating grid driver crate my not be turned on.

Solution: Go to the VME status GUI on the lower right hand part of tpc_top.

Check the gating grid crates and make sure that the crates are on. If it isn't turn it on and repeat the startup procedure.

I try to use the default sequence button/global controls and I input stuff but nothing happens.

YES, both gating grid driver crates are on...

Problem: Some tasks need to be restarted on the VME cpu.

Solution: On sc.star.bnl.gov, and ONLY that machine, type 'telnet scserv 9002'

When you get the '->' prompt, type 'td "global"'. You need the quotation marks around global

You should get a message like 'value=0=0x0'.

Next, type 'seq &global'

You should see something like:

@(#)SEQ Version 1.9.1: Thu Nov 18 13:48:10 EST 1999

tShell 05/18/00 15:23:57: Spawning state program "global", task name = "global"

tShell 05/18/00 15:23:57: Task id = 12881028 = 0xc48c84

value = 12881028 = 0xc48c84

-> global 05/18/00 15:23:57: Spawning task 12870628: "global_1"

global 05/18/00 15:23:57: Spawning task 12860228: "global_2"

global 05/18/00 15:23:57: Spawning task 12849828: "global_3"

global 05/18/00 15:23:57: Spawning task 12839428: "global_4"

global 05/18/00 15:23:57: Spawning task 12829028: "global_5"

global 05/18/00 15:23:57: Spawning task 16285744: "global_6"

And some messages about setting global power, and setting power on sector X, where X is some number.

If this doesn't work, reboot the cpu. If it still doesn't work, try cycling the power on crate 54.

The power on the driver crate was cycled (interlocks or user power cycle). The gating grid crates are on and I just did a HV set using High voltage control. The GUI says the HV is on but when I input a setpoint, nothing happens.

Problem: The HV for the channel was enabled in the software but is not enabled in the gating grid module.

Solution: The best thing to do would be to execute the startup sequence under the default sequences. This will properly reinitiate the database values and ensure an edge for the gating grid modules to trigger on. If this doesn't work, try rebooting the cpu in the gating grid control crate. you can do this from the VME status screen by doing a system reset.

A group of four contiguous outer/inner sectors is not responding. They correspond to one module. THIS MODULE WAS JUST INSTALLED.

Problem: The pins to the power bus in the back are not making connections.

Solution: Pull the module that you just installed, check to make sure that the 6 pins in the back of the module are not pushed in. ** Please take a look at the next problem.

Symptoms which indicate you are in trouble (i.e. need an access):

A group of four contiguous outer/inner sectors is not responding. When I

look at the GG_Mapping.log file, I see that the four sectors are controlled

by the same gating grid module. THIS MODULE HAS BEEN WORKING UP 'TILL NOW.

Problem: Most likely a fuse is blown.

Solution: Please note this in the log and notify the above mentioned people. Call for access.

With HV enabled, Vgg/Vgghi/Vgglo seem to be set to the maximum allowable voltage (150 for Vgghi/Vgglo and 300 for Vgg). There is no way to control the voltage. When HV is disabled, the setting go to zero but as soon as the HV is enabled, the channel goes to its max.

Problem: It is most likely a shorted transistor that needs to be replaced.

Solution: Please note this in the log and notify the above mentioned people.

Vgglo seems to influence the Vgg setting. When Vgglo is set to 0 volts, Vgg

monitor value matches the requested value. In this case, Vgg goes from 130 to 140.6 when I set Vgglo to 75 volts.

Problem:

Solution: Please note this in the log and notify the above mentioned people.

(F)TPC Gating Grid Driver Calibration

Eugene T. Yamamoto *
University of California - Los Angeles
Los Angeles, CA 90095

June 11, 2000

Below are the procedures for calibration of the gating grid driver. There are 3 steps to calibrating the gating grid:

1. Measure the voltage out of the back of the driver modules.
2. Calibrate the ADC.
3. Calibrate the DAC.

You must have electrical training in order to perform the gating grid calibrations.
If you do not know what "driver module" means, STOP! you are not qualified to calibrate the gating grid.
If you do not know what ADC or DAC stands for, STOP! You are not qualified to calibrate the gating grid.
To measure the voltage out of the back of the driver modules you will need:

1. Voltmeter or scope
2. HV probe box
3. pen and paper

If you do not know what a voltmeter or scope is, STOP! You are not qualified to calibrate the gating grid.
If you do not know what the HV probe box is, STOP! You are not qualified to calibrate the gating grid.
If you do not know what pen or paper is, STOP! You are not qualified to calibrate the gating grid.
The procedure is as follows. If ANY of this does not make sense, you are not qualified to calibrate the gating grid.

- Make sure you are not getting any triggers if you are using a voltmeter.

*e-mail: geno@physics.ucla.edu

- Turn on gating grid driver modules to nominal operating voltage settings
- Use probe box to measure the voltage out of back of gating grid driver module (write these values down). Note that gui reads positive volt but output is negative.
  1. With Gate Normal, measure vgghi. This is vgg - vgghi.
  2. With Gate Normal, measure vgglo. This is vgg + vgglow
  3. With Gate OPEN, measure either vgghi or vgglo output. This is vgg
- at a computer, telnet scserv 9002
- In the sc.star.bnl.gov epics root directory, open

  `TPCggridApp/TPCggridDb/calibrations/analog_in_calibrations.txt.`

  This file contains all the ADC calibrations.
- In the serial connection, type: dbpr "channel",4. Channel is defined as:
  1. `GG_A_In_X_X`
  2. The first X is either A,B,C, where A is vgghi, B is vgglo, and C is vgg
  3. The second X is the channel number, from 1 to 48.
- Look for the field labeled 'EGUF'. This is the ADC calibration constant.
- In the serial connection, type: dbpf "channel.EGUF","(int) *measuredvalue/monitoredvalue* × *eguf*", where little eguf is the value of EGUF that printed out when you ran the dbpr command. (int) means find the closest integer to the result of *measuredvalue/monitoredvalue* × *eguf*. it should look something like:

  `dbpf ''GG_A_In_A_1.EGUF'',''262''`
- copy the dbpf line and paste it to the bottom of the

  `analog_in_calibrations.txt`

  file and allsectors.cal file.
- Once this is done, in the serial line, type:

  `seq &calibrate,"channel=X,sector=X"`

  where X is the appropriate value. This calibrates the DAC and creates a file in the calibrations directory with a self explanatory name. Replace the appropriate line in allsectors.cal with the new DAC calibration.

Note that you can do the DAC calibration globally using the file:

`TPCggridApp/TPCggridDb/scripts/calibrate.txt`

which is a vxWorks script. If you don't know how to run scripts, you should not do the global DAC calibration.
Of course, once you do the calibration, you don't have to cut and paste 144 values, you can cat or grep or whatever to get the values. If you don't know how to do this, you should not do the global DAC calibration.

User: sysuser
Host: sc3.starp.bnl.gov
Class: sc3.starp.bnl.gov
Job: newcalibrate.st

				AFTER CALIB	SHOULD BE
119.7	V_{cc}	HIGH	+48.9	HIGH = +77.1	+40
65.6		GG	-66.3	GG = -76.5	-75.0
115		LO	-180.9	LO = -146.5	-190

```
   alibrate.st:
  hor: Dennis Reichhold
his is based on calibrate.st
  hor: Eugene Yamamoto, UCLA
This sequencer handles the calibration of each channel.
This is to deal with drifting of voltages. This can also
be used in the calibration of replacement components.
*/

program newcalibrate

%%#include <stdio.h>
%%#include <stdlib.h>
%%#include <taskLib.h>
%%#include <time.h>

%%char calib_time[26];
%%time_t caltime;
%%struct tm timer;
%%struct tm *timerptr=&timer;
%%char bogus[2];
%%FILE *f;

int i;
float vOAvg;
float vIAvg;

string message;
f  at VGG_hi_meas;
  oat VGG_lo_meas;
float VGG_meas;
int sector;

assign message to "GG_Message";
assign VGG_hi_meas to "GG_Calibrate.A";
assign VGG_lo_meas to "GG_Calibrate.B";
assign VGG_meas to "GG_Calibrate.C";
assign sector to "GG_Calibrate.D";
monitor sector;

float VGG_hi_In;
float VGG_lo_In;
float VGG_In;
float VGG_hi_Out;
float VGG_lo_Out;
float VGG_Out;
int VGG_hi_In_EGUF;
int VGG_lo_In_EGUF;
int VGG_In_EGUF;
float VGG_hi_Out_EGUF;
float VGG_lo_Out_EGUF;
float VGG_Out_EGUF;
int VGG_hi_In_SEVR;
int VGG_lo_In_SEVR;
  VGG_In_SEVR;

assign VGG_hi_In to "";
assign VGG_lo_In to "";
assign VGG_In to "";
```

```
assign VGG_hi_Out to "";
assign VGG_lo_Out to "";
assign VGG_Out to "";
assign VGG_hi_In_EGUF to "";
assign VGG_lo_In_EGUF to "";
assign VGG_In_EGUF to "";
assign VGG_hi_Out_EGUF to "";
assign VGG_lo_Out_EGUF to "";
assign VGG_Out_EGUF to "";
assign VGG_hi_In_SEVR to "";
assign VGG_lo_In_SEVR to "";
assign VGG_In_SEVR to "";

float VGG_hi_comp;
float VGG_lo_comp;
float VGG_comp;
float VGG_hi_diff;
float VGG_lo_diff;
float VGG_diff;
int new_EGUF;

char filename[164];
char *sect;
char *chan;
char str[64];

ss calibrate {

  state wait {

    when(sector<0&&delay(0.1)) {
    } state wait

    when(sector==0&&delay(0.1)) {
      sprintf(message,"All dressed up and nowhere to go...");
      pvPut(message);
    } state wait

    when(sector>0) {
      pvGet(VGG_hi_meas);
      pvGet(VGG_lo_meas);
      pvGet(VGG_meas);
      VGG_hi_meas=abs(VGG_hi_meas);
      VGG_lo_meas=abs(VGG_lo_meas);
      VGG_comp=abs(VGG_meas);
      VGG_hi_comp=VGG_meas-VGG_hi_meas;
      VGG_lo_comp=VGG_lo_meas-VGG_meas;
    } state start_cal
  } /* end state wait */

  state start_cal {

    when(VGG_hi_meas<0.1||VGG_lo_meas<0.1||VGG_meas<0.1) {
      sprintf(message,"You must enter values for ALL channels");
      pvPut(message);
      sector=-1;
      pvPut(sector);
    } state wait

    when(VGG_hi_meas>0.1&&VGG_lo_meas>0.1&&VGG_meas>0.1) {
```

```
      sprintf(message,"Connecting channels for calibration...");
      pvPut(message);
      sprintf(str,"GG_A_Out_A_%d.VAL",sector);
      pvAssign(VGG_hi_Out,str);
      sprintf(str,"GG_A_Out_B_%d.VAL",sector);
      pvAssign(VGG_lo_Out,str);
      sprintf(str,"GG_A_Out_C_%d.VAL",sector);
      pvAssign(VGG_Out,str);
      sprintf(str,"GG_A_Out_A_%d.EGUF",sector);
      pvAssign(VGG_hi_Out_EGUF,str);
      sprintf(str,"GG_A_Out_B_%d.EGUF",sector);
      pvAssign(VGG_lo_Out_EGUF,str);
      sprintf(str,"GG_A_Out_C_%d.EGUF",sector);
      pvAssign(VGG_Out_EGUF,str);
      sprintf(str,"GG_A_In_A_%d.VAL",sector);
      pvAssign(VGG_hi_In,str);
      sprintf(str,"GG_A_In_B_%d.VAL",sector);
      pvAssign(VGG_lo_In,str);
      sprintf(str,"GG_A_In_C_%d.VAL",sector);
      pvAssign(VGG_In,str);
      sprintf(str,"GG_A_In_A_%d.EGUF",sector);
      pvAssign(VGG_hi_In_EGUF,str);
      sprintf(str,"GG_A_In_B_%d.EGUF",sector);
      pvAssign(VGG_lo_In_EGUF,str);
      sprintf(str,"GG_A_In_C_%d.EGUF",sector);
      pvAssign(VGG_In_EGUF,str);
      sprintf(str,"GG_A_In_A_%d.SEVR",sector);
      pvAssign(VGG_hi_In_SEVR,str);
      sprintf(str,"GG_A_In_B_%d.SEVR",sector);
      pvAssign(VGG_lo_In_SEVR,str);
      sprintf(str,"GG_A_In_C_%d.SEVR",sector);
      pvAssign(VGG_In_SEVR,str);

      sprintf(filename,
              "/star/sc/users/sysuser/epics/R3.12.2-LBL.4/TPCggridApp/TPCggridDb
      f = fopen(filename, "a");
%%    caltime = time(NULL);
%%    timerptr = localtime(&caltime);
%%    strcpy(calib_time,asctime(timerptr));
      fprintf(f,"# Auto-calibration on %s",calib_time);
    } state ADC_cal
  } /* end state start_cal */

  state ADC_cal {

    when(delay(1.0)) {
      pvGet(VGG_hi_In);
      pvGet(VGG_lo_In);
      pvGet(VGG_In);
      pvGet(VGG_hi_In_EGUF);
      pvGet(VGG_lo_In_EGUF);
      pvGet(VGG_In_EGUF);
      VGG_hi_diff=abs(VGG_hi_In-VGG_hi_comp);
      VGG_lo_diff=abs(VGG_lo_In-VGG_lo_comp);
      VGG_diff=abs(VGG_In-VGG_comp);
      if (VGG_hi_diff>1.5) {
        VGG_hi_In_EGUF*=VGG_hi_comp/VGG_hi_In;
        pvPut(VGG_hi_In_EGUF);
        fprintf(f,"dbpf \"GG_A_In_A_%i.EGUF\",\"%i\"\n",sector,VGG_hi_In_EGUF);
      }
```

```
    if (VGG_lo_diff>1.5) {
      VGG_lo_In_EGUF*=VGG_lo_comp/VGG_lo_In;
      pvPut(VGG_lo_In_EGUF);
      fprintf(f,"dbpf \"GG_A_In_B_%i.EGUF\",\"%i\"\n",sector,VGG_lo_In_EGUF);
    }
    if (VGG_diff>2.) {
      VGG_In_EGUF*=VGG_comp/VGG_In;
      pvPut(VGG_In_EGUF);
      fprintf(f,"dbpf \"GG_A_In_C_%i.EGUF\",\"%i\"\n",sector,VGG_In_EGUF);
    }
    sprintf(message,"Finished calibrating ADCs");
    pvPut(message);

  } state DAC_cal
} /* end ADC_cal */

state DAC_cal {
  when(delay(3.0)){
    pvGet(VGG_hi_In_SEVR);
    pvGet(VGG_lo_In_SEVR);
    pvGet(VGG_In_SEVR);

    if (VGG_hi_In_SEVR!=0) {
      vOAvg = 0;
      vIAvg = 0;
      for(i=0;i<10;i++) {
        pvGet(VGG_hi_Out);
        pvGet(VGG_hi_In);
        vOAvg += VGG_hi_Out;
        vIAvg += VGG_hi_In;
        %%taskDelay(sysClkRateGet()*2);
        sprintf(message,"Took %d voltage measurements\n", i);
        pvPut(message);
      }
      vOAvg *=.1;
      vIAvg *=.1;

      printf("Voltages: %f   %f\n",vOAvg, vIAvg);
      pvGet(VGG_hi_Out_EGUF);
      printf("EGUF  %f\n",VGG_hi_Out_EGUF);
      VGG_hi_Out_EGUF *= (vIAvg/vOAvg);
      printf("EGUF' %f\n",VGG_hi_Out_EGUF);

      pvPut(VGG_hi_Out_EGUF);
      fprintf(f,"dbpf \"GG_A_Out_A_%i.EGUF\",\"%f\"\n",sector,VGG_hi_Out_EGUF)
    }

    if (VGG_lo_In_SEVR!=0) {
      vOAvg = 0;
      vIAvg = 0;
      for(i=0;i<10;i++) {
        pvGet(VGG_lo_Out);
        pvGet(VGG_lo_In);
        vOAvg += VGG_lo_Out;
        vIAvg += VGG_lo_In;
        %%taskDelay(sysClkRateGet()*2);
        sprintf(message,"Took %d voltage measurements\n", i);
        pvPut(message);
      }
      vOAvg *=.1;
```

```
      vIAvg *=.1;

      printf("Voltages: %f   %f\n",vOAvg, vIAvg);
      pvGet(VGG_lo_Out_EGUF);
      printf("EGUF  %f\n",VGG_lo_Out_EGUF);
      VGG_lo_Out_EGUF *= (vIAvg/vOAvg);
      printf("EGUF' %f\n",VGG_lo_Out_EGUF);

      pvPut(VGG_lo_Out_EGUF);
      fprintf(f,"dbpf \"GG_A_Out_B_%i.EGUF\",\"%f\"\n",sector,VGG_lo_Out_EGUF)
    }

    if (VGG_In_SEVR!=0) {
      vOAvg = 0;
      vIAvg = 0;
      for(i=0;i<10;i++) {
        pvGet(VGG_Out);
        pvGet(VGG_In);
        vOAvg += VGG_Out;
        vIAvg += VGG_In;
        %%taskDelay(sysClkRateGet()*2);
        sprintf(message,"Took %d voltage measurements\n", i);
        pvPut(message);
      }
      vOAvg *=.1;
      vIAvg *=.1;

      printf("Voltages: %f   %f\n",vOAvg, vIAvg);
      pvGet(VGG_Out_EGUF);
      printf("EGUF  %f\n",VGG_Out_EGUF);
      VGG_Out_EGUF *= (vIAvg/vOAvg);
      printf("EGUF' %f\n",VGG_Out_EGUF);

      pvPut(VGG_Out_EGUF);
      fprintf(f,"dbpf \"GG_A_Out_C_%i.EGUF\",\"%f\"\n",sector,VGG_Out_EGUF);
    }

  } state clean_up
} /* ends state DAC_cal */

state clean_up {
  when(delay(3.0)) {
    fclose(f);
    VGG_hi_meas=0.0;
    VGG_lo_meas=0.0;
    VGG_meas=0.0;
    sector=0;
    pvPut(VGG_hi_meas);
    pvPut(VGG_lo_meas);
    pvPut(VGG_meas);
    pvPut(sector);
    sprintf(str,"");
    pvAssign(VGG_hi_Out,str);
    pvAssign(VGG_lo_Out,str);
    pvAssign(VGG_Out,str);
    pvAssign(VGG_hi_Out_EGUF,str);
    pvAssign(VGG_lo_Out_EGUF,str);
    pvAssign(VGG_Out_EGUF,str);
    pvAssign(VGG_hi_In,str);
    pvAssign(VGG_lo_In,str);
```

```
    pvAssign(VGG_In,str);
    pvAssign(VGG_hi_In_EGUF,str);
    pvAssign(VGG_lo_In_EGUF,str);
    pvAssign(VGG_In_EGUF,str);
    pvAssign(VGG_hi_In_SEVR,str);
    pvAssign(VGG_lo_In_SEVR,str);
    pvAssign(VGG_In_SEVR,str);
  } state wait
}
```