ACQ2106_TIGA User Guide

Prepared By: Peter Milne

Date: 18 August 2021

Rev	***Date***	***Description***
1	20 July	Init
2	24 July	More pictures.
3	4 July 21	PG5 flavor
4	18 August 21	pg_test.py

Table of Contents

Copyright and Attribution.

Document created using OpenOffice.Org www.openoffice.org.

1 Introduction

User Guide for **ACQ2106_TIGA : Timing Generator Appliance.**

1.1 New For Rev 3:

New model. PG5 programmable output.

Currently 2 versions are supported:

- **TIGA TYPE A: ACQ2106-4xACQ423-2xPG4-TIGA**
  - 128AI, 200kSPS
  - 2 x 4 channel PG, CLKOUT, TRGIN
  - 4 LEMO adapter above site 6 providesCLK, TRG, AUX1, AUX2 to ACQ2106 motherboard.

- **TIGA TYPE B: ACQ2106-2xACQ482-2xPG5-TIGA**
  - 32AI, 80MSPS
  - 2 x 5 channel PG, module-local TRG input NOT connected.
  - No additional Site 6 adapter, so the TRG inputs are CLK and TRG on the acq2106 motherboard, respectively.

1.2 Intended Audience

TIGA users.

1.3 Scope

Programming and basic signal definitions.

1.4 Glossary

- TAI : Atomic Time.
- PG : Pulse Generator
  - Timed series of pulses, user programmed by STL
- WR : White Rabbit
  - Precision network timing, clock shared with fixed phase on network
  - Every UUT shares TAI with a time step size of 100nS with an accuracy of<1nS
- TAI Stamp
  - External TRGIN timestamped against TAI
- WRTD
  - White Rabbit Trigger Distribution: method for sharing triggers definitions on network.
- WRTT
  - White Rabbit Time Trigger: trigger edge generated at programmed TAI.
- Derived CLKOUT
  - Output clock, divided from input clock.
- Direct DO
  - Logic output value, direct control from networked software knob.

1.5 References

1. 4GUG : 4G User Guide.
2. HAPI : Host API
3. ACQ400CSS : GUI using EPICS and cs-studio
4. ACQ2106 Installation Guide.

1.6 Notation

- **command** : indicates name of a program (command)
- preformatted text : literal input or output from terminal session.
- *Defined Term* : some term or acronym specific to this domain (perhaps referenced in the glossary)

2 System Description

2.1 ACQ2106_TIGA : Timing Generator Appliance

- ACQ2106 Carrier with White Rabbit
- 1..4 DIO482_TD modules in Sites 2..5
  - "TD" is a hardware adaptation with
    - 6 x LEMO-OO "SPL" on a "Top Deck" mount
      - 4 x DO4
      - 1 x CLKOUT
      - 1 x TRGIN
    - 6 x LED indicators.
- 1..2 ACQ423ELF-32 ADC modules in Sites 1,6
- Standard CLK, TRG, WR 10MHz, WR 1PPS functionality above site 6.

2.2 ACQ423 32 simultaneous AI

- Absolutely standard 32 or 64 channel digitizer.
- Sample clock choices: use any of CLK.d[0-7], with local CLKDIV.
- Trigger choices: use any of TRG.d[0-7]

2.3 DIO482-PG Logic

FPGA logic configures the DIO482 module as Pulse Generator:

2.3.1 DO4 PG

- 4 Digital Outputs
  - Either Direct from software knob or
  - Driven from Pulse Generator PG
- PG Loaded with STL at port 450{site}1
  - STL identical to GPG, but fixed clock from WR 10MHz
  - TRG is user selectable site.trg=EN,Dx,EDGE.
    - options to trigger from local WRTT, local TRGIN
    - global WRTT0, WRTT1 available on TRG.d0, TRG.d1
  - Output bits 0..3 output to front panel DO4.
  - Output bit 4 optionally output to TRIGGER bus (TRG.d{site+2})

- a PG in one site can trigger another PG in another site.

2.3.2 CLKOUT

- Front panel SPL CLKOUT driven from divided local clock
- Master Clock, CLK.d{0..7}
  - CLK.d0 : WR 250MHz
  - CLK.d1 : WR 10MHz
  - CLK.d2 + other sites CLKOUT (if enabled).
- CLKOUT = CLK.dX / CLKDIV 1..65525.

2.3.3 TRGIN

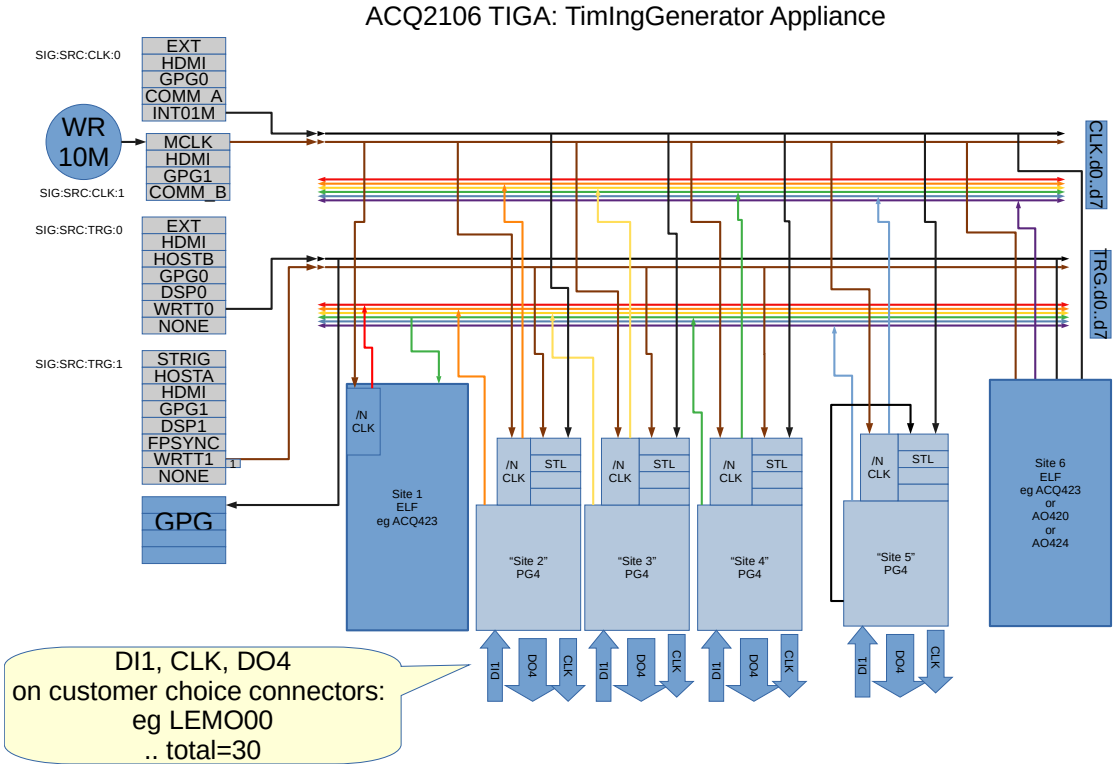
- Input to Front panel SPL TRGIN
- Unique TAI latch per site. Can configure to source WRTD message.
- TRGIN can optionally drive TRG.d{site+2} to trigger other sites.
- TRGIN can trigger local PG.

2.3.4 WRTD

- Dedicated site-specific White Rabbit trigger latch and trigger output.

2.4 Block Diagram

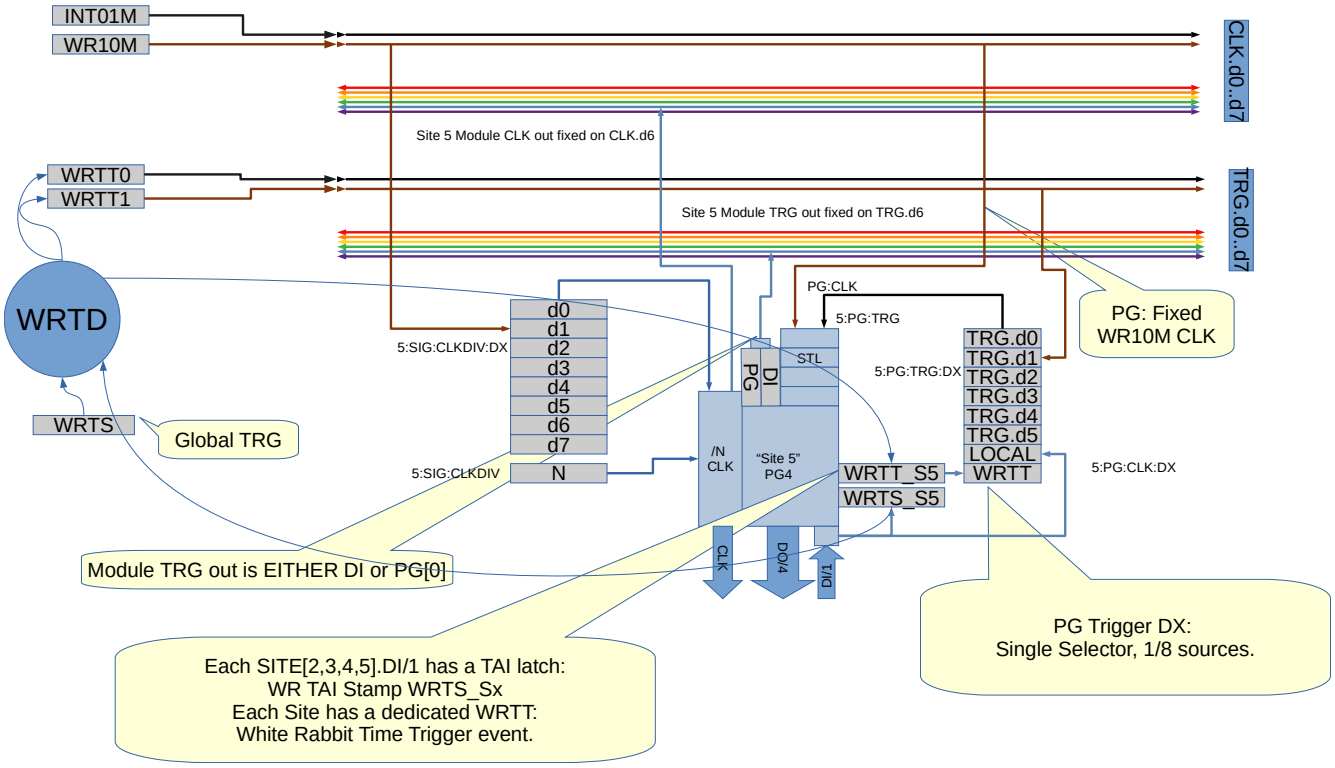
2.4.1 System With Six Sites



- Each PG Site has a dedicated TRGIN, this can ..
  - Be sampled against WR TAI to support a WRTD trigger message
  - Can trigger the local PG
  - Can pass to the Trigger bus.
- Each PG Site has a 5 bit PG function
  - Can be triggered by the local TRGIN or
  - Can be triggered by a local site-specific WRTT or
  - Trigger off the Trigger bus from Motherboard or other Site.
  - 4 Bits of PG output to the front panel.
  - Optionally 1 bit “PGIDX” outputs the to the Trigger bus
    - TRGOUT : choice of either
      - TRGIN or
      - PGIDX
- Each PG site has a CLKOUT function

2.4.2 Single PG Site Detail

PulseGeneratorModule : v6 Detail .. using Site 5 as example



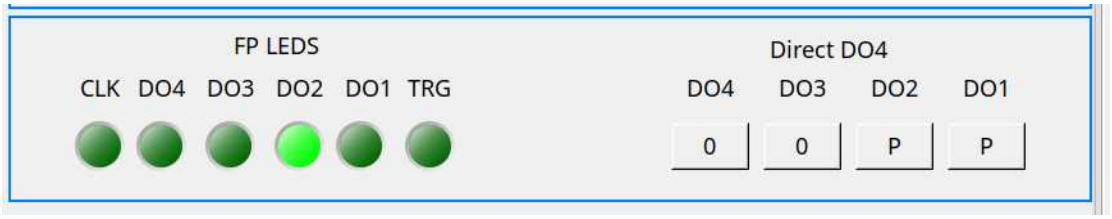
3 Indicator Lamps

3.1 DIO482-TD

The DIO482 hardware has 6 indicator lamps:

CLK [DO4 DO3 DO2 DO1] TRG

3.2 OPI



3.2.1 FP LEDS

Mimics the front panel LEDS.

- PV: ${UUT}:${SITE}:LED:{CLK ,DO4,DO3,DO2,DO1,TRG
- HAPI: uut.sX.LED_CLK

3.2.2 Direct DO4

Controls whether the Digital Output LEMO DOx driven from the PG or direct from SOFTWARE

- PV: ${UUT}:${SITE}:DO:{1,2,3,4}
- HAPI uut.sS.DO_x
- Values:
  - P : driven from PG
  - 0 : direct output LO
  - 1 : direct output HI

3.2.3 LED Flash Patterns

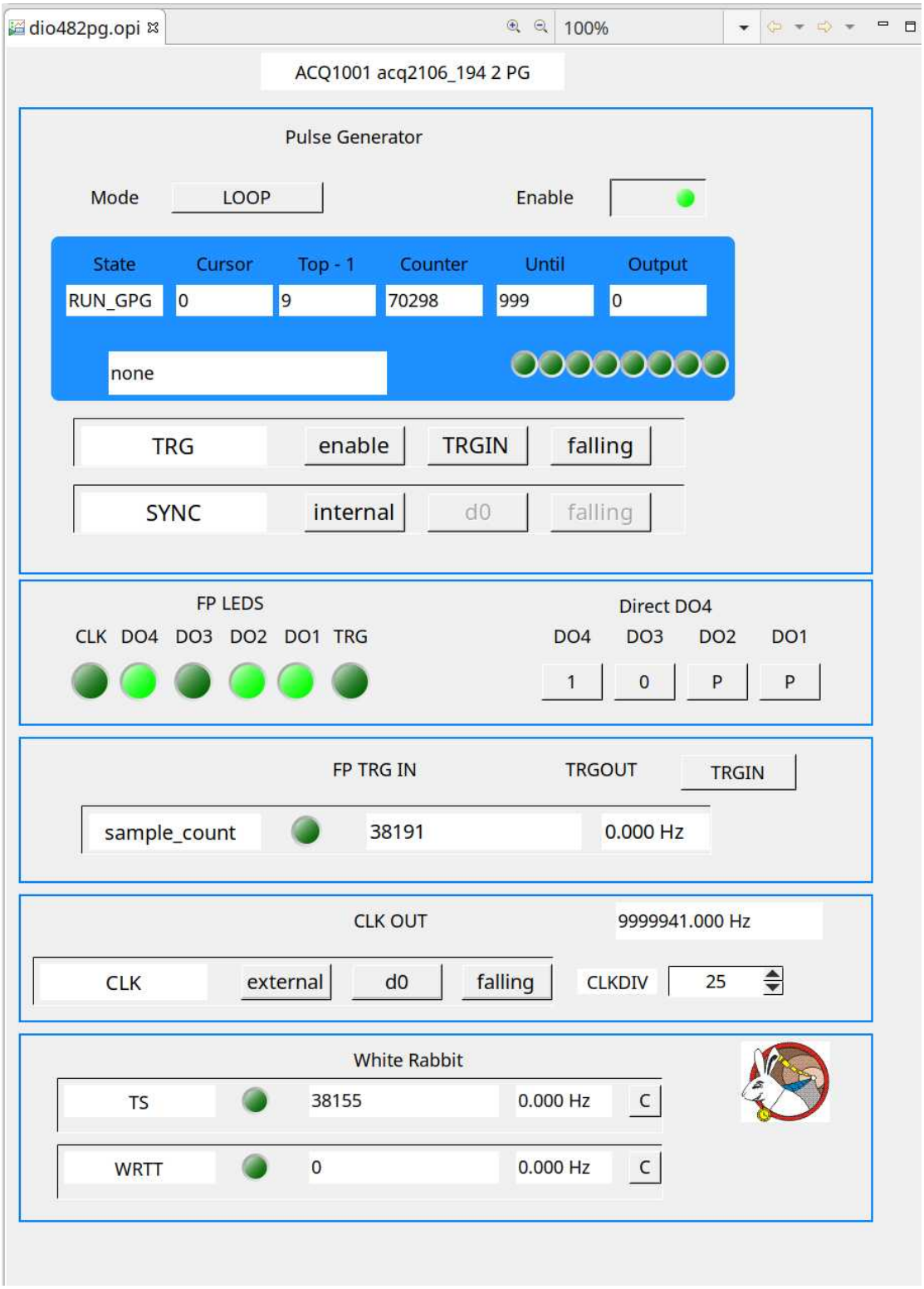
- CLK, TRG:
  - FREQ > 1Hz : Decade flash, count flashes in 2s period,
    - < 10Hz : 1 flash
    - < 100Hz : 2 flashes
    - < 1kHz : 3 flashes
    - < 10kHz : 4 flashes
    - < 100kHz : 5 flashes
    - < 1MHz : 6 flashes
    - < 10MHz : 7 flashes
    - < 100MHz : 8 flashes
  - FREQ < 1Hz: LED toggles every time an edge is counted.

- DOX:
  - ON:OFF %
  - 0:100 OFF: Direct OUTPUT LOW or PG disabled.
  - 50:50 Long (“dashes”) : PG WAIT_TRG
  - 50:50 Short (“dots”) : PG RUN
  - 20:80 Completed, channel fired, final output LO
  - 80:20 Completed, channel fited, final output HI
  - 100:0 ON : Direct OUTPUT HI

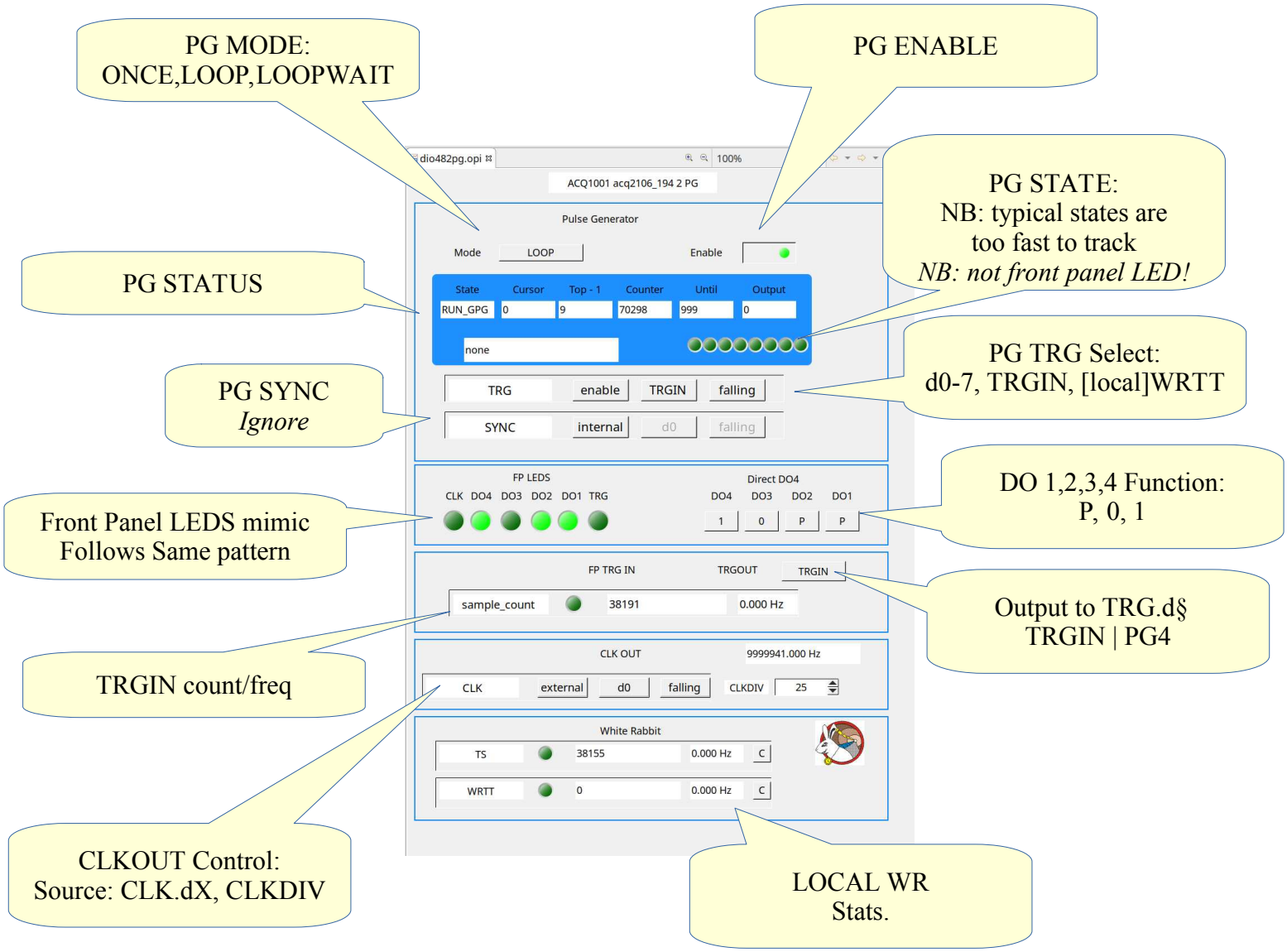
- EPICS OPI flash pattern matches front panel.
- Front panel LED’s are close together. Best viewed by using a pinhole mask to isolate individual LED’s to enable accurate counting.

4 Software Control

4.1 DIO482PG OPI



4.1.1 DIO482 OPI Key



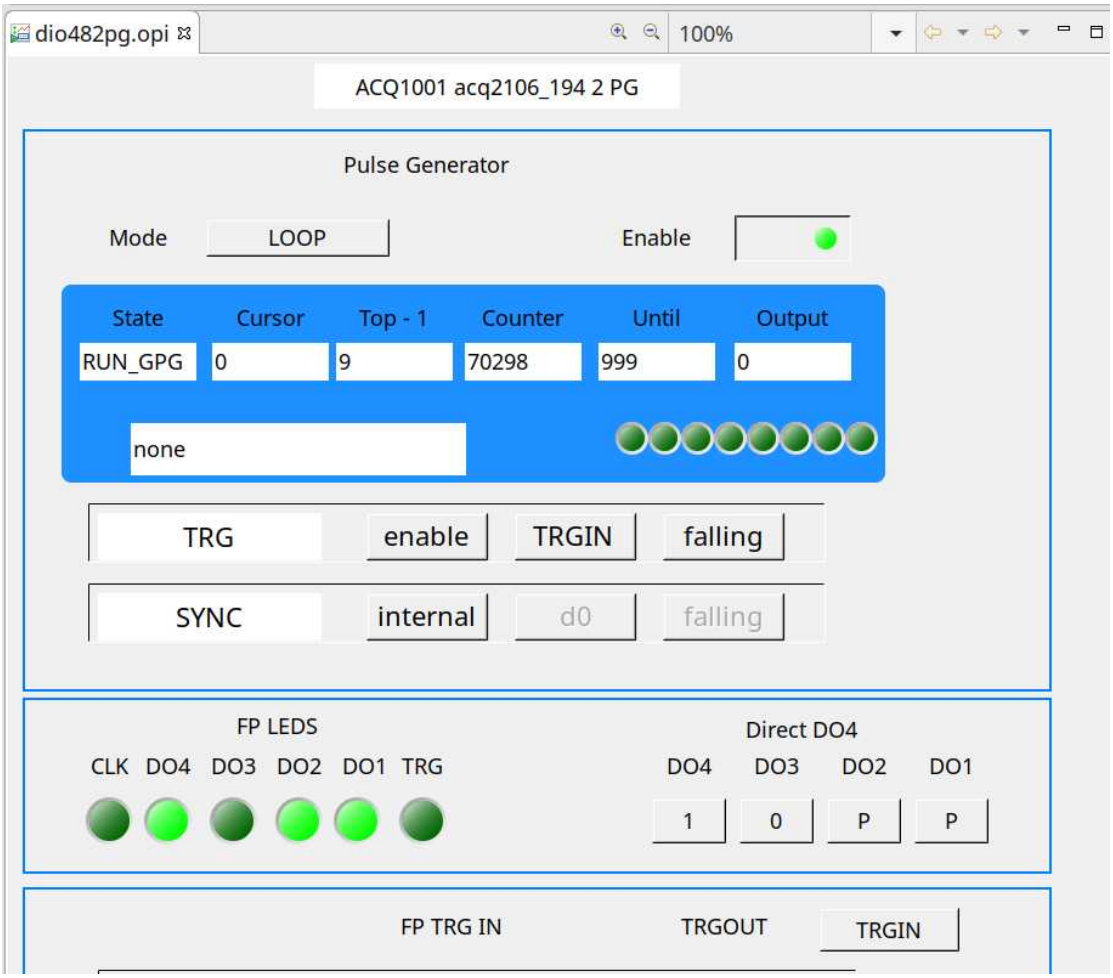
4.2 HAPI

```
class Acq2106_TIGA(Acq2106):

    def __init__(self, uut, monitor=True):
        print("acq400_hapi.Acq2106_TIGA %s" % (uut))
        Acq2106.__init__(self, uut, monitor, has_wr=True)

    def load_dio482pg(self, site, stl, trace = False):
        self.load_stl(stl, AcqPorts.DIO482_PG_STL+site*10, trace)

    def set_DO(self, site, dox, value = 'P'):
        self.svc["s{}".format(site)].set_knob("DO_{}".format(dox),
                                              value)
```

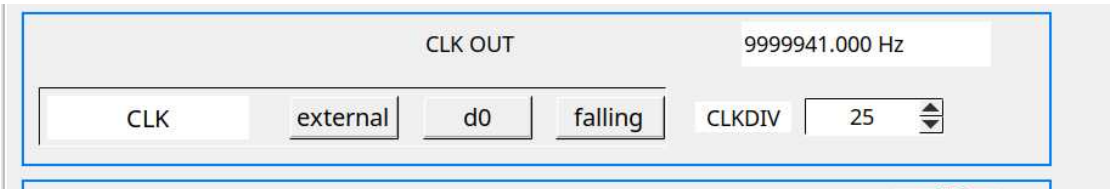
4.3 PG4 Knobs

- set.site SITE gpg_enable 0
- load STL to Port 405{S}1 where S={2,3,4,5}
  - The PG has a 4+1 bit state:
    - d0-d3 go to front panel DO1 .. DO4,
    - when selected d7 “Index” routes to TRG.d§ # S§ = SITE+1
    - select d7 with set.site $SITE TRGOUT PGIDX # alt TRGIN
- PG clock is
  - WR 10MHz on TIGA TYPEA
  - WR 40MHz on TIGA TYPE B
- set.site $SITE gpg_mode $MODE # 0:ONCE, 2:LOOP, 3:LOOPWAIT
- set.site $SITE TRG dX EDGE # dX: d0,d1,d1 .. TRGIN, WRTT
  - TRGIN is the local FP TRGIN signal
  - WRTT is the local WRTTx, site specific.
- OUTPUT Enables:
  - set.site DO:x VALUE # P: source from PG, 1:set HI, 0: set LO
    - set.site $SITE TRGOUT PG4 # routes d4 to TRG.d§
      - eg for triggering another unit.

4.3.1 PG4 OPI


dio482pg.opi
100%
ACQ1001 acq2106_194 2 PG
Pulse Generator
Mode
LOOP
Enable
State
Cursor
Top - 1
Counter
Until
Output
RUN_GPG
0
9
70298
999
0
none
TRG
enable
TRGIN
falling
SYNC
internal
d0
falling
FP LEDS
CLK
DO4
DO3
DO2
DO1
TRG
Direct DO4
DO4
DO3
DO2
DO1
1
0
P
P
FP TRG IN
TRGOUT
TRGIN

4.4 CLKOUT



4.4.1 Configure the Clock

- set.site $SITE CLK ENABLE
- set.site $SITE CLK:DX
  - CLK.d0 : WR 250M
  - CLK.d1 : WR 10M
  - CLK.d2-d7 : other CLK lines. DO NOT use CLK.d§
- set.site $SITE CLKDIV
  - Clock divider 1..65535

4.4.2 Connect Output to main CLK bus

- Always connects to CLK.d§
  - CLK.d§ : accessible by other sites as a source clock
  - May be routed to SYNC_OUT (HDMI)
  - Counter/Freq Monitor (COUNTERS PAGE).

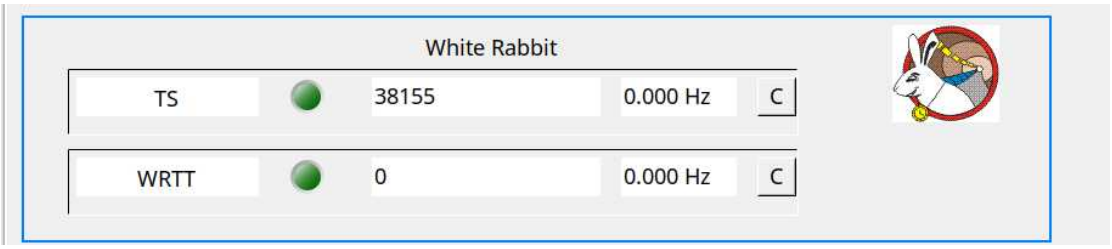
4.5 TRGIN

- Always latched by wr_tai_sample_sX
- May be used as local PG trigger.
- May be connected to TRG.d§
  - set.site $SITE TRGOUT TRGIN

4.6 White Rabbit Triggers

A TIGA box supports 8 WR Triggers, two Global (mainboard) and one Local per site trigger. At this time, only sites with DIO482-PG logic support the Local trigger, so in practise, it's 2+4 rather than 8.

4.6.1 WR Trigger OPI



4.6.2 WR_TAI_STAMP

- Each SITE TRGIN is has a unique TAI STAMP latch, and this may be used to drive a WRTD TX packet.

4.6.3 WRTTX

- Each SITE can generate WRTTX, specific to the site, and consumed locally within the site when PG trigger is set to WRTT

4.6.4 WRTD Packet Enhancement.

- The WRTD Packet has a 16 char IDENT. In the enhanced mode:
  - IDENT[15] is an 8 bit binary mask, addressing one or more triggers:
    - 0x01 : WRTT0 (Global0, routes to TRG.d0)
    - 0x02 : WRTT1 (Global1, routes to TRG.d1)
    - 0x03 : n/a
    - 0x04 : WRTT3 (Local Site 2, routes to Site 2 PG TRG.WRTT)
    - 0x10 : WRTT4 (Local Site 3, routes to Site 3 PG TRG.WRTT)
    - 0x20 : WRTT3 (Local Site 4, routes to Site 4 PG TRG.WRTT)
    - 0x40 : WRTT3 (Local Site 5, routes to Site 5 PG TRG.WRTT)
  - The WRTD_TX_MASK parameter in a site controls the transmitted mask for any wrtd_tx action from that site. WRTD_TX_MASK=0 reverts to original behaviour
- WRTD control for each site is the SAME command as the SITE0 WRTD control but accessed from SITE 2,3,4,5

4.6.5 Global Triggers

WRTT0 and WRTT1 are controlled from the “site 11” service as before.

When enabled (WRTD_TX=1), the Global TIMESTAMP transmit service will send a WRTD message with ID from the site 11 WRTD_ID value. This could select either WRTT0 or WRTT1 depending on mask values in WRTD_RX_MATCHES or WRTD_RX_MATCHES1 for WRTT0, WRTT1 respectively.

The Global triggers may be propagated on the Trigger Bus at TRG.d0, TRG.d1 respectively, and any site can be set to use that trigger.

4.6.6 Local Triggers

Each PG Site has a Local Trigger controlled from the normal “site N” service.

When enabled (WRTD_TX=1), the Local TIMESTAMP transimit service will send a WRTD message with ID from the same site WRTD_ID value.

The Site Local Trigger is by default LOCAL to the site, but may be propagated to the global Trigger Bus at TRG.d§ when the following are set:

- Site PG Trigger source is WRTT
  - set.site N TRG:DX=WRTT
- Site TRGOUT is set to TRGIN
  - set.site N TRGOUT TRGIN

4.6.7 WRTD RX

A single receiver handles all incoming events for all sites.

- WRTD_TX_MASK == 0:
  - Packets matched by the WRTD_RX_MATCHES filter will result in a Global WRTT0 trigger
  - Packets matched by the WRTD_RX_MATCHES1 filter will result in a Global WRTT1 trigger.
- WRTD_TX_MASK != 0:
  - Packets matched by the WRTD_RX_MATCHES filter will result in one or more triggers as determined by WRTD_TX_MASK.
  - It is possible to activate any and all of the available triggers using this mechanism.

4.6.8 Immediate Triggers

An “Immediate” trigger can be invoked on any WRTD site service with the following commands:

- set.site N wrtd_txi 1 # send single default message
- set.site N wrtd_txi ID # send single message ID

“Immediate” means : send message with current timestamp + WRTD_DELTA_NS.

“Default” means message defined by : WRTD_ID, WRTD_TX_MASK and the timestamp.

4.6.9 TX Quick

A TX Quick packet can be sent at any time from software command.

The packet contains a “Timestamp” code TAI_QUICK (0xFFFFFFFF) with special meaning:

When a WRTT trigger register is loaded with the TAI_QUICK it will trigger at once.

TX Quick is seen as a kind of multicast Emergency Stop command.

- set.site N wrtd_txq 1 # send single default message
- set.site N wrtd_txq ID # send single message ID.

“Default” means message defined by : WRTD_ID, WRTD_TX_MASK and the timestamp.

4.7 STL Notes:

4.7.1 STL Format:

AT_COUNT, STATE:

AT_COUNT::

clock count from trigger when state transition occurs

STATE::

hex code for the state, a mask of

- 0x01 : PG1
- 0x02 : PG2
- 0x04 : PG3
- 0x08 : PG4
- 0x10 : PG5 (if enabled)
- 0x80 : TRGOUT, output to TRG.dX if TRGOUT=PGIDX
  - dX :: X = site+1

CONVENTIONS:

- Initial STATE is ZERO, final declared state should end at ZERO to avoid glitch on restart
- STL should set PGIDX as second final state, this gives a useful “Finished” indicator on TRG.dX
- CSCALE env var may be applied to scale up the counts dynamically.
  - each AT_COUNT is multiplied by CSCALE on load.
- Hardware TSCALE control is deprecated, since it doesn’t doesn’t scale well.

5 PG5

PG5 uses identical hardware to PG4, but a boot-time switch selects the CLK output as PG5.

5.1 Enable

cat /mnt/local/sysconfig/acq400.sh

TIGA TYPE B (essential):

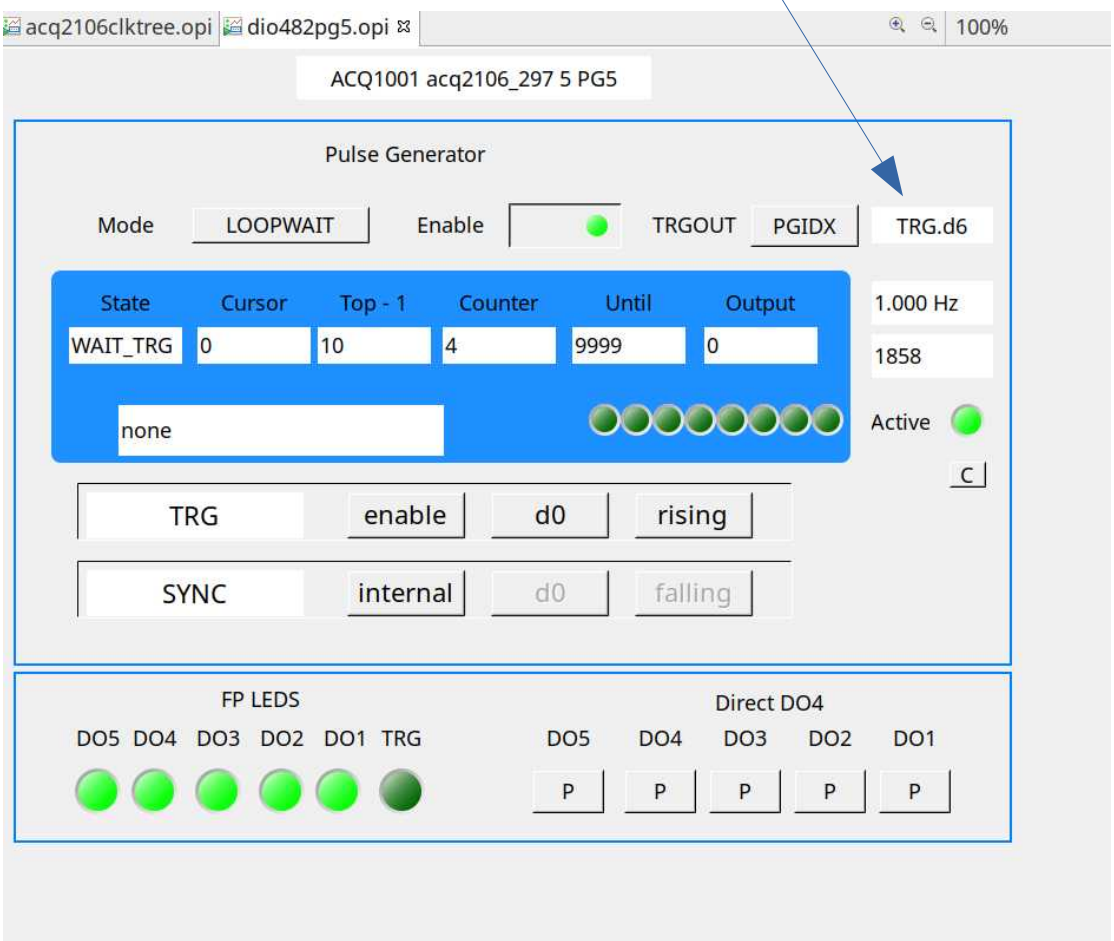
```
export DIO482TD_PG5=notrg
```

TIGA TYPE A (option), module-local trigger input is still available:

```
export DIO482TD_PG5=y
```

5.2 OPI

Simplified OPI as follows. The site trigger counter TRG.d6 is a count of Index pulses from the PG, it's strongly recommended to include the Index pulse in the STL, then there's a clear indication at end of cycle (a single count for a one-shot, a frequency for loop functions).



5.3 PG5 Knobs

- set.site SITE gpg_enable 0
- load STL to Port 405{S}1 where S={2,3,4,5}
  - The PG has a 5+1 bit state:
    - d0-d4 go to front panel DO1 .. DO5,
    - when selected d7 “Index” routes to TRG.d§ # S§ = SITE+1
    - select d7 with set.site $SITE TRGOUT PGIDX # alt TRGIN
- PG clock is
  - WR 10MHz on TIGA TYPEA
  - WR 40MHz on TIGA TYPE B
- set.site $SITE gpg_mode $MODE # 0:ONCE, 2:LOOP, 3:LOOPWAIT
- set.site $SITE TRG dX EDGE # dX: d0,d1,d1 .. TRGIN, WRTT
  - TRGIN is the local FP TRGIN signal if connected.
  - WRTT is the local WRTTx, site specific.
- OUTPUT Enables:
  - set.site DO:x VALUE # P: source from PG, 1:set HI, 0: set LO
    - set.site $SITE TRGOUT PGIDX # routes d4 to TRG.d§
      - eg for triggering another unit.
    - set.site $SITE TRGOUT TRGIN # routes PG4 trigger to TRG.d§
      - also for triggering another unit.

5.4 Programming Example

- BEST option: pg_test.py :: Python version of scrip below, has more options and works remotely.

```
acq2106_297> cat /usr/local/CAREpg_test
#!/bin/sh

export SITE=${SITE:-2}
TSCALE=${TSCALE:-1}
MODE=${1:-2}

[ $TSCALE -ne 1 ] && echo Loading with timescaler=$TSCALE

set.site $SITE gpg_enable 0
set.site $SITE gpg_timescaler $TSCALE
/usr/local/epics/scripts/gpg_monitor 1
# index pulse appears as site TRG counter.
set.site $SITE TRGOUT PGIDX

stl2gpg  /dev/acq400.$SITE.gpg <<EOF
10,1
20,0
30,2
40,0
50,4
60,0
70,1f
# 8x : index pulse
80,81
90,0
EOF

# Set GPG to LOOP and Enable GPG
set.site $SITE gpg_mode $MODE
set.site $SITE gpg_enable 1
```

- MODE: 1: ONECE, 2: LOOP, 3: LOOP-WAIT (for repeating trigger)
- TSCALE is now a software scaler, all time entries are multiplied by TSCALE .. so the same STL can be used at different rates .. useful eg so that we can slow things down for easier observation.
- former CSCALE is deprecated.

```
set.site 6 gpg_timescaler 40
SITE=6 /mnt/local/pg_test 3
set.site 4 gpg_timescaler 400
SITE=5 /mnt/local/pg_test 3
```