

# Digital Gammasphere Software Development at ANL

Timothy Madden

# Driver Development

- asynDriver used for hardware drivers.
  - See <http://www.aps.anl.gov/epics/modules/soft/asyn/>
- asyn eliminates need for custom driver code and custom EPICS DTYP's.
- All PVs represented as “parameters” in source code, similar to fields in an object.
- C++ based code.
- Simplifies development and support.
- Created Python scripts to convert Gretina EPICS databases to be compatible with asyn.
- Allows easy exposure of raw VME registers as PVs.

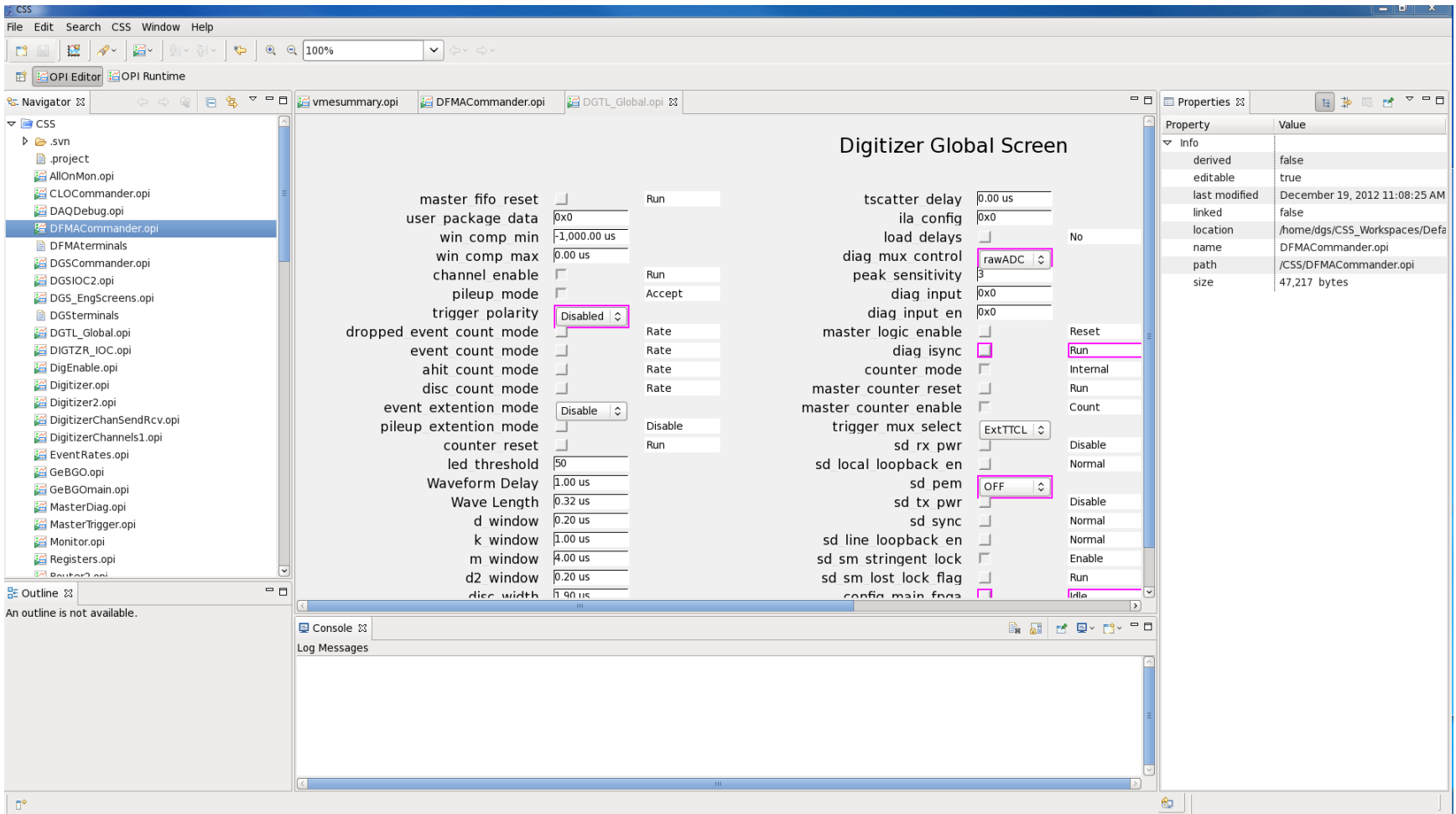
# Changes to Sender/Receiver

- Changed Send/Receive to accommodate new Digitizer header structure.
- Refactored code to put header-related operations into functions rather than inline code.
- Added switches and PVs to control timing and CPU usage of sender, sorter.
- Various bug fixes, esp. reading digitizer FIFO into “throw-away” buffer on low memory condition, rather than crashing.
- Added PVs to allow sending a random sampling of data, and setting percent of data to send, for UDP monitor.

# Changes to GUI

- Migrate from EDM to Control System Studio.
  - See <http://sourceforge.net/apps/trac/cs-studio/wiki/BOY>
- Created python scripts to convert EDM screens to CSS screens.

# Some screens



Global PVs, Edit Mode

# Screens

Sender Summary

	TCP	Bufs	SndBufs	AAA Err	Mon %	Sendto	Stat	UDP
1	0.0	400	0	0	0.000000		DNS ERROR	0
2	0.0	400	0	0	0.000000		DNS ERROR	0
3	0.0	400	0	0	0.000000		DNS ERROR	0
4	0.0	400	0	0	0.000000		DNS ERROR	0
5	0.0	400	0	0	0.000000	192.168.1.4	Attached	0
6	0.0	400	0	0	0.000000	192.168.1.4	Attached	0
7	0.0	400	0	0	0.000000		DNS ERROR	0
8	0.0	400	0	0	0.000000		DNS ERROR	0
10	0.0	400	0	0	0.000000		DNS ERROR	0

Set Sender:  Mon %:

Router Configuration Screens

Router 1: DSSD Master Trigger, DSSD Router 1, DSSD Router 2, DSSD Router 3

Router 4: DSSD Router 4, DSSD Router 5

Each router screen displays: Misc Status, Misc Control, and Input Link Mask (U R L H G F E D C B A).

DFMA Control Panel

09:29:39

Start/Stop: Stop

Save/NoSave: No Save

Copy/Sort: Sort

Buttons: Trigger, GlobalControl, Terminals, Digitizers, VME Status, Digitizer Ena, Misc Control, Live TimeStamp, Scripts, Debug Screens

# Debug Screens

VME Space VME01: REBOOT

Write Read

Value

Address

Slot

Raw VME Register Space  
Reboot Button

DbgReg Triggers

VME-ClkIntMode

REBOOT ALL CRATES

Digitizer VME01 Digitizer VME06

Digitizer VME02 Digitizer VME07

Digitizer VME03 Digitizer VME08

Digitizer VME04 Digitizer VME09

Digitizer VME05 Digitizer VME10

Digitizer VME11

Main Debug Screen  
Global Reboot Button

DGS Digitizer Board Registers VME02:DIG:

board id	0x10030		Channels	version	0x30420C32
programming_done	0x300000	0x300000			
hardware_status	0x11110100				
user_package_data	0xCD	0xCC			
win_comp_min	0x7960	0x7960			
win_comp_max	0x0	0x0			
dac	0x0	0x0			
tscatter_delay	0xDEADADD0	0xDEADADD0			
ila_config	0x0	0x0			
channel_pulsed_control	0x0				
diag_mux_control	0x0	0x0			
peak_sensitivity	0x3	0x3			
baseline_start	0xA2000	0xA2000			
diag_channel_input	0x0	0x0			
fbus_status	0x0				
live_timestamp_lsb	0xD8588782				
live_timestamp_msb	0x130E				
master_logic_status	0x00050	0x00050			
trigger_config	0x2	0x2			
fbus_command	0xDEADADD0				
code_revision	0x4C98				
code_date	0x20121211				
aux_io_read	0xDEADADD0				
aux_io_write	0xDEADADD0				
aux_io_config	0xDEADADD0				
sd_config	0x80	0x180			
adc_config	0xDEADADD0				
fpga_ctrl_reg	0x0	0x0			
vme_status	0x400041				
vme_aux_status	0x0				
vme_qp_ctrl	0x2	0x0			
vme_timeout_val	0x27100				

Raw Board Registers  
Auto-generated screen.

# Auto-generation of Screens and EPICS Databases

- Register specification of VME boards are documented in a spreadsheet.
- Created Python scripts to
  - Read spreadsheet.
  - Generate C++ code for digitizer and trigger driver.
  - Generate EPICS databases
  - Generate CSS GUI screens.
  - Generate st.cmd files for all crates.
- Allows for support of
  - Multiple system configurations such as DFMA, DGS and Clover.
  - Configuration of system for multiple firmware versions and firmware upgrades.



# Auto-generation of Screens and EPICS Databases *cont.*

- New PV names:
  - VME01:DIG4:xxx rather than Cry4xxx
- Python code to parse EPICS databases
  - EPICS PV class, for setting/getting PV fields.
  - Simplifies reading/writing, find/replace, auto-generation of databases.
  - EPICS database represented as Python list of `epics_pv` objects.

# Auto-generation of Screens and EPICS Databases *cont.*

- Python object-oriented code for GUI generation.
  - CSS screen appears as `cssScreen` object.
  - Widgets appear as `cssWidget` object.
  - Read/Write, Search/Replace.
  - Allows auto-generation of screens based on EPICS database.
  - Allows the generation of EPICS database from a screen.
  - Can read EDM screens.

# VME Address Space Exposed as PVs and Flashing FPGA via EPICS

- Created driver called `asynDebugDriver()` to represent raw VME address space in Channel Access.
- EPICS PVs represent
  - Board Slot
  - Board Address (offset)
  - Data to read/write
  - PV to perform VME read/write.
  - PV to store FPGA Firmware as EPICS array.
- Allows for hardware development/debugging in-system.
- Allows for flashing FPGA over EPICS.
  - Post FPGA firmware as a large EPICS array.
  - PVs for erasing, writing, verifying FPGA Flash.
  - Linux Command Line to Flash FPGAs.