McDonald Observatory
THE UNIVERSITY OF TEXAS AT AUSTIN

HET TCS: EPICS RELAY SERVICE

Background

McDonald Observatory
THE UNIVERSITY OF TEXAS AT AUSTIN

EPICS (Experimental Physics Industrial Control System) is an inter-process communication system used by HET to share parameters between the National Instruments I/O controller (cRIO) on the PFIP hardware and the higher level facility software (TCS, et al).

PFIP server, the TCS component that interacts directly with the EPICS system, currently handles the setting of EPICS parameters from the TCS side and the query of EPICS parameters from the cRIO at a ~1Hz sample rate. PFIP server architecture, and the manner in which EPICS communication is implemented, are limiting factors for increasing the sample rate substantially beyond 1Hz. Adding parameters to PFIP server does not scale sufficiently.

A new tool, **EPICS relay**, has been implemented to convert EPICS variables to TCS events. This enables capture of EPICS state, over time, in the TCS database.

The Tool

McDonald Observatory
THE UNIVERSITY OF TEXAS AT AUSTIN

- Read only, for now, access to EPICS variables
- Configuration provides the EPICS variable to event key mapping, i.e.

long-array-process-variables = guider1_position_actual:arm:carriage, \
guider2_position_actual:arm:carriage, \
wfs1_position_actual:arm:carriage, \
wfs2_position_actual:arm:carriage

Type specific configuration

EPICS array variable

Array element names

- Currently supported data types:

--double-process-variables <STRINGARRAY> []
--enum-process-variables <STRINGARRAY> []
--int-process-variables <STRINGARRAY> []
--long-process-variables <STRINGARRAY> []
--long-array-process-variables <STRINGARRAY> []
--string-process-variables <STRINGARRAY> []
--bitfield-process-variables <STRINGARRAY> []

- Native type array variables are given just as the long data type example above

Configuration

McDonald Observatory
THE UNIVERSITY OF TEXAS AT AUSTIN

$HET_SRC_ROOT/epics_relay/testing/test.conf

```
ioc-namespace = pfip
route = tcp://127.0.0.1:30123
event-route = tcp://127.0.0.1:30124


# XXX need to decide on bit order for the configuration, low to high, or high to low?
#                       0    1 2     3     4  5  6  7     8  9  10        11            12          13         14 15
bitfield-process-variables =
wfs1_status:ready:on:enabled:faulted:nil:nil:nil:warning:nil:nil:target_reached:internal_limit_active:setpoint_ack_xxx:following_erro
r:nil:nil,wfs2_status:ready:on:enabled:faulted:nil:nil:nil:warning:nil:nil:target_reached:internal_limit_active:setpoint_ack_xxx:followi
ng_error:nil:nil,guider1_status:ready:on:enabled:faulted:nil:nil:nil:warning:nil:nil:target_reached:internal_limit_active:setpoint_ack_
xxx:following_error:nil:nil,guider2_status:ready:on:enabled:faulted:nil:nil:nil:warning:nil:nil:target_reached:internal_limit_active:set
point_ack_xxx:following_error:nil:nil


#enum-process-variables = wfs1_power:enabled=6:disabled=0,wfs2_power:enabled=6:disabled=0
#enum-process-variables = wfs1_power:disabled:nil:nil:nil:nil:nil:enabled,wfs2_power:enabled=6:disabled=0,\
#enum-process-variables=cwfs_mirror_actual:deployed=1:retracted=2
#enum-process-variables=cwfs_mirror_actual:deployed=1:retracted=2,acqcam_mirror_actual:retracted=2:deployed=1


long-array-process-variables =
guider1_position_actual:arm:carriage,guider2_position_actual:arm:carriage,wfs1_position_actual:arm:carriage,wfs2_position_actua
l:arm:carriage
```

Configuration

McDonald Observatory
THE UNIVERSITY OF TEXAS AT AUSTIN

- Default behavior is to listen for EPICS beacons, but specific EPICS IOC URLs may be given
- A specific IOC name (i.e. “PFIP”) may optionally be given, we don’t have multiple IOCs at HET, yet
- EPICS variables are case insensitive
- What we know as PFIP:GUIDER1_STATUS may also be given as guider1_status
- Array, bitfield, and enum field names are up to the user but these determine the corresponding TCS event keying
- Need to be running a monitor to capture the resulting events
- Why is the configuration so hideous? LTL does not support line wrapping in the config, I worked on that briefly but likely won’t pick it back up until the holiday

Usage

McDonald Observatory
THE UNIVERSITY OF TEXAS AT AUSTIN

- Running the relay generates events and running the monitor captures those and puts them in a database:

  $ epics_relay –c /path/to/config 2>&1 | tee /path/to/epics-relay-console.log &

  $ monitor … (see Jim or Randy)

- I have only run the monitor with the localhost settings found in the epics_relay/testing/test.conf file

Result

McDonald Observatory
THE UNIVERSITY OF TEXAS AT AUSTIN

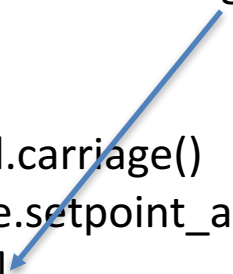
- Running like so, on htcs

```
$ ./epics_relay/epics_relay -c epics_relay/testing/test.conf --rate-hertz 10
```

- And a monitor in a separate terminal

```
$ ./monitor/tcs_monitor  --system-names tcp://127.0.0.1:30124 --db-file /tmp/epics-relay-test.db
```

- With a little python, assuming the carriage is moving while the setpoint ack is high

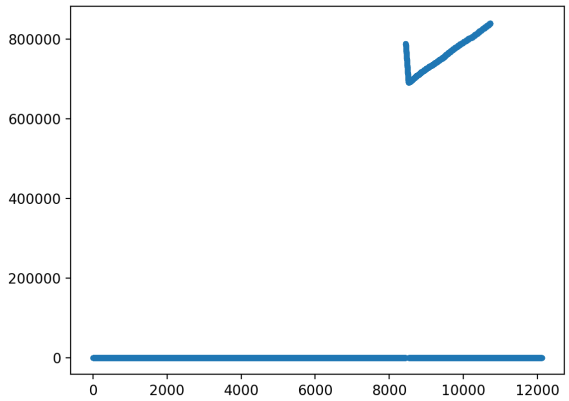


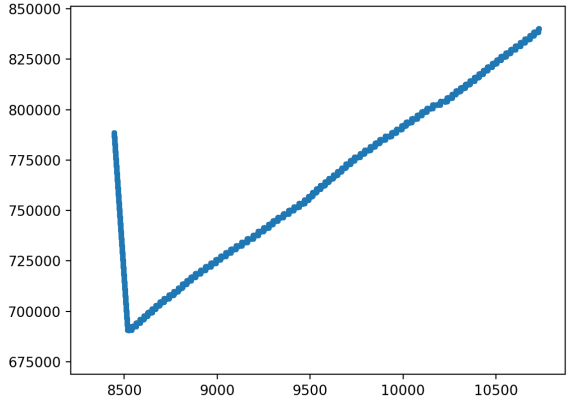
```
d = tcsdb('epics-relay-test.db')
[tg1,g1_car_enc] = d.epics.handle.pfip.guider1_position_actual.carriage()
[_,g1_car_sp_ack] = d.epics.handle.pfip.guider1_status.carriage.setpoint_ack()
move_waypoint = [1 if it == 'true' else 0 for it in g1_car_sp_ack]
plt.plot(tg1,np.array(move_waypoint)*np.array(g1_car_enc),'.')
```

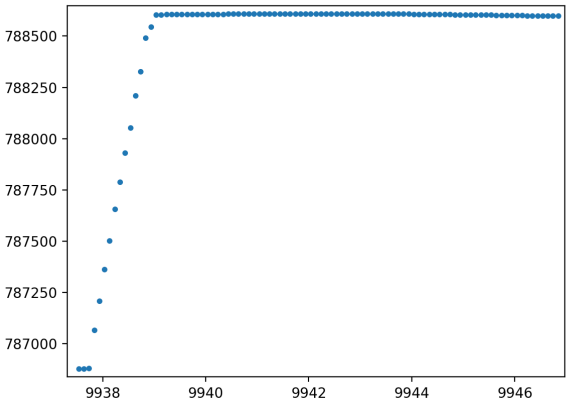
Result

McDonald Observatory
THE UNIVERSITY OF TEXAS AT AUSTIN

- We have a picture of a run of Randy's "small moves" script, for guider 1's carriage encoder counts (Y) over seconds of time (X)







- If data needs to get out of Python and into Excel, or such, then arrays of values are easily saved to text from Python

```
> np.savetxt('/path/to/g1_car_enc_file', g1_car_enc)
```

Next Steps

McDonald Observatory
THE UNIVERSITY OF TEXAS AT AUSTIN

- Finalize, for now, the configuration needed to capture all desired variables
  - George and Randy play with configuration and see what they can get from the data
  - Determine what we want to monitor automatically on a daily basis
- Implement an init script
  - Or multiple, depending on how we want to run the relay(s)
- Perhaps split the monitoring, put all tracker traffic in a separate db?
- Update default monitoring to include EPICS relay instance(s)
- Extend probe move information in daily report to higher fidelity, per move, move profiles
- Develop tools to analyze move data, over larger time scales, and catch anomalies
- Remove any redundant logging from PFIP server