

# An External Trigger Alert System for the LIGO Control Rooms

Andrew Williamson

External triggers (ExtTrigs) such as gamma-ray bursts (GRBs) and supernovae represent valuable opportunities for combined gravitational and electromagnetic observations in the advanced detector era. In order for externally triggered gravitational wave searches to reap maximum rewards during observation runs it will be wise to do whatever possible to provide periods of stable data around the times of these astrophysical events. Specifically, planned hardware injections and planned downtime might be cancelled or postponed for a period of time after an ExtTrig event. It will therefore be useful to have a system in place in the Hanford and Livingston control rooms that will alert the on-duty controllers and interface with hardware injection software whenever such a promising event occurs.

## Querying GraceDB

The control room systems run EPICS software [1] with a graphical interface provided via MEDM screens [2]. An MEDM screen dedicated to ExtTrigs and hardware injections was modified to display information on the latest ExtTrig event of interest (Fig. 1). Four EPICS channels were created per interferometer to contain this information.

- `$(IFO):CAL-INJ_EXTTRIG_ALERT_ID` – GraceDB ID
- `$(IFO):CAL-INJ_EXTTRIG_ALERT_TIME` – event GPS time
- `$(IFO):CAL-INJ_EXTTRIG_ALERT_TYPE` – event type {GRB, Supernova}
- `$(IFO):CAL-INJ_EXTTRIG_ALERT_SOURCE` – original source of event information {Fermi, Swift, SNEWS}

The above channels may be easily read and written to from a running python process by making use of the python package PyEpics [3]. All changes are visible in real time on the MEDM screen and software may easily monitor these channels and react accordingly when they are updated.

ExtTrig events are collated in GraceDB, the Gravitational Wave Candidate Event Database [4]. Queries may be made to this database from python code by utilising the `ligo.gracedb` mod-

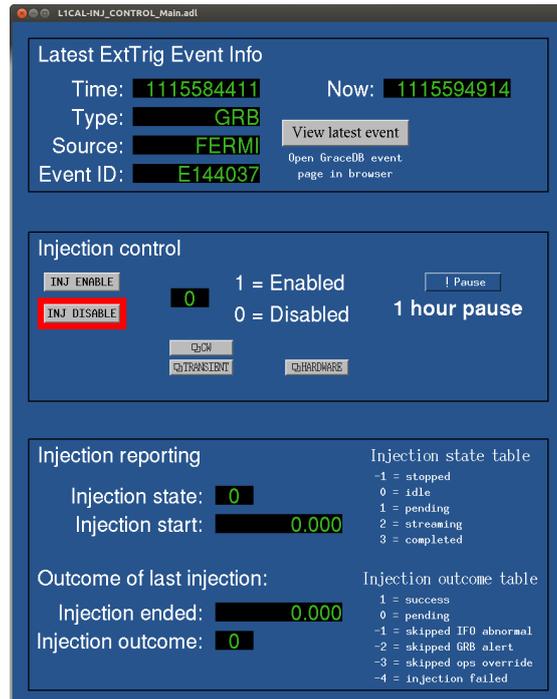


Figure 1: Screenshot of the injection control MEDM screen, featuring information on the latest ExtTrig event as processed by the python script. If the event occurred within a pre-defined period of time before the present – say 2 minutes – the top rectangular box will flash red to alert the controller. The ‘View latest event’ button opens the relevant GraceDB event page in the default web browser.

ule, which requires an X509 user certificate. For our purposes we request the information on all ‘External’ group events between the present time and  $T$  seconds before present, which are compiled into a list. This query is repeated every  $z$  seconds, where  $z < \frac{T}{N_f}$  to allow for  $N_f$  failed/skipped queries. If no event has occurred in the time period  $T$  the code sleeps before repeating the query. If there have been one or more events within  $T$  the first record in this list is conveniently always the most recent event, and therefore generally our event of interest. If this is a new event – ie. the GraceDB entry was created after the last successful query – the event’s unique GraceDB ID, GPS time, type, and source are parsed by the code. If the type and source identify this as an event of interest, the channels are updated with the new event details and. The latency for this whole process is only a few seconds for reasonable lookback times  $T$  (of order tens of minutes). To mitigate against untimely code/query failures, a value of  $T$  of order one hour may be used without increasing the latency. A standard value for  $z$  is 5 seconds.

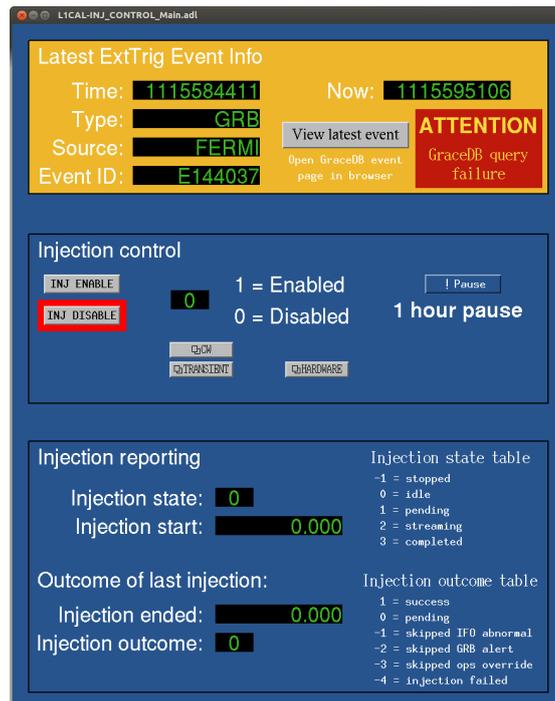


Figure 2: Screenshot of the injection control MEDM screen when no GraceDB query has been executed in the past minute. If an ExtTrig event were to have occurred in the last five minutes – and the code were still running – the top box would be red, not yellow, and the text box on the right would inform the user that a new event had occurred.

## Monitoring

In order to monitor the running of this python code, a fifth new EPICS channel was added,  $\$(IFO):CAL-INJ\_EXTTRIG\_ALERT\_QUERY\_TIME$ . This channel is populated with the GPS time whenever a successful query is performed. The MEDM screen monitors this channel and will display an ‘ATTENTION’ message if it has been more than one minute since the last successful query (Fig. 2).

## Hardware Injections

Another related feature of this code is the disabling of hardware injections upon receipt of a new event alert. This is achieved by first decreasing the hardware injection gain to 0 over the space of one second, achieved by setting the time period of the change in  $\$(IFO):CAL-INJ\_TINJ\_TRAMP$  to 1, then the final gain value in  $\$(IFO):CAL-INJ\_TINJ\_GAIN$  to 0. This is followed by a complete disable by setting  $\$(IFO):CAL-INJ\_TINJ\_ENABLE$  to 0. The code then notes the time at which to re-enable injections, after which time the process above is reversed (set  $\*_ENABLE$  to 1 then ramp the gain up to 1 over a second). The current default time period for the disabling of

injections for all GRB and supernova alerts is 3 hours.

## References

- [1] Experimental physics and industrial control system. <http://www.aps.anl.gov/epics/>. Accessed April 24, 2015.
- [2] Medm: Motif editor and display manager. <http://www.aps.anl.gov/epics/extensions/medm/>. Accessed April 24, 2015.
- [3] Matthew Newville. Pyepics: Epics channel access for python. <http://cars9.uchicago.edu/software/python/pyepics3/>. Accessed April 24, 2015.
- [4] Gracedb — gravitational wave candidate event database. <https://gracedb.ligo.org/>. Accessed April 24, 2015.