Light Injection – DAQ Interface

This is a summary of the discussions held on Oct 2 on the DAQ – Light Injection interface. This is not a design document. Those present were: Tass Belias, Lisa Falk-Harris, Philip Harris, Saeed Madani, Tim Nicholls and Geoff Pearce.

Control

1. Running Mode

This section updated on Oct 16 to reflect a modification to the control route

Run Control is in overall control of data taking. Control of the detector is executed by the DCS with instructions that are abstracted from the details of the detector itself. The Light Injection (LI) system will be controlled by Run Control. Control will be instructions on the LI running mode required with configuration details left to the Flasher System. It is not anticipated that Run Control will require control at a more detailed level than this (with the exception of 'pause' instructions, see below). There are three basic operating modes for the LI system:

- Dedicated light injection run
- Free running, i.e. flasher data injected during data taking
- Idle, i.e. light injected disabled

Run Control selects the L.I. mode at run start and instructs the L.I. system to enter into this mode as a component of the overall run prepare configuration. Note that the LI system will require details of the DAQ configuration in order to correctly configure itself. For example it will need to know which crates are offline in order to correctly sequence light injection (to avoid flashing detector regions where the corresponding PIN diodes are offline, to avoid flashing at low light levels if the trigger PMT is offline and so on).

2. LI Mode Parameters

The LI running modes will have a default configuration, which the shift leader may wish to modify. For example, there may be a need to perform a full LI calibration of the detector at a particular time rather than waiting for the scheduling to complete one on its default time scale. Control of the light injection at this level will be performed via the DCS or flasher PC and not via Run Control. DCS may be used rather than the flasher PC in order to keep user interfaces to a minimum, though a decision on this should be the outcome of discussions with the DCS group. When in free running mode such parameters can, in principle, be changed 'on the fly'; there is no need to stop data taking in order to make a change to the sequencing.

3. Veto

Run Control needs to be able to pause or stop light injection automatically as a tool for rate control should the data rate become unexpectedly large. This could occur due to detector faults, increasing noise levels etc and not necessarily be caused by the LI system. Since an increase in data rate sufficient to warrant this action would come from monitoring over a period of seconds or more so there is no requirement for instantaneous control. Pause instructions will be issued via the DCS over the LAN.

Operating Rates

In fact this was discussed following the meeting but is included here as it has a bearing on the other issues. The light injection data under the current calibration plans is a very substantial data source. The salient points are:

1. The detector is not guaranteed to be stable for longer than ~ 1 hour so the capability for one complete detector light injection calibration every hour is required.
2. There are 320 LEDs at the Far Detector and 33 LEDs at the Near Detector so this is achieved by completing one full calibration point on one LED every 11sec (Far) or 109sec (Near). Each calibration point requires 1000 light injections to achieve the required accuracy.
3. A single flash is estimated to produce up to 20KB (FD) or 100KB (ND) of data. A single calibration point therefore requires 20MB (FD) or 100MB (ND) of data. This assumes that cross talk to nearest neighbour pixels, with the current pixel flashing pattern, generates readout and a 5 (3) time slice occupancy per flashed (cross talk) pixel at the Near Detector (including reflection). If a more lenient worst case can be guaranteed these limits would reduce. These data are spread over 2 crates (FD) and 2-3 crates (ND).
4. The offline preference would be that all the data for a single calibration point be contained within a single Time Frame (i.e. it all goes in one massive chunk to a single trigger processor). For a 1sec Time Frame this would place up to 10MB/s (FD) or 40MB/s (ND) into a crate. Given that each readout processor (ROP) has two time frame buffers, this would require 20MB (FD) or 80MB (ND) of on-board memory in each crate for the light injection data alone. However, splitting the data into chunks (e.g.20 chunks of 50 flashes) and spreading them over a number of time frames is a perfectly acceptable scenario for the flasher sequencing and analysis and offers substantial rate reductions.
5. Splitting the data from a single calibration point across n time frames reduces the data rates by a factor n. Choosing n=10 for the FD and n=40 for the ND (for example only) places a total of ~2MB into a single time frame. Note that this implies a 10 (40) fold increase in the light injection summary data written out. Every other time frame could be used for the Near Detector but to complete in 11sec every time frame would need to be used at the Far Detector unless a scheme involving simultaneous LED operation were invoked. The size of a Time Frame is programmable in the DAQ and will be set to a convenient size when the on-site rates are known but is expected to be in the range 1/10 to 1 sec. A smaller Time Frame changes the preceding numbers accordingly.
6. The data volume required for the light injection calibration is potentially very high and may even require essentially continuous flashing (i.e. every second). There is flexibility in the flasher sequencing that can be used to spread the data over crates and time to average out the rate and this will be used. The impact of cross talk is substantial and needs to be better understood in terms of readout generated, particularly for time slices from the M64s. This is as true for 'physics' data as it is for LI data.
7. The detector calibration requirements will be examined with the aim of reducing the amount of LI data required. For example, if it is the PMT gain that requires hourly calibration perhaps only a subset of fibres need to be flashed rather than all of them. Phil Harris will construct and document a sequencing model for the Light Injection that will satisfy the calibration requirements with the minimum of LI data.

Communication with the Flasher PC.

A need has been expressed for information to flow in real time between the flasher PC and the trigger farm in order that the flasher analysis output can be tagged with data describing the light injection conditions at the time of the event. Since these communications are required only by the light injection software it was agreed to encapsulate them within the flasher process in the trigger farm. The procedure will be as follows:

1. The Light Injection PC will be on both the DAQ and the DCS LAN.
2. When the flasher analysis in the trigger farm is passed a Time Frame of flasher events (see below for a description of this interface) it will issue a request to the flasher PC over the DAQ LAN for the required information. Request and Information flow will be over a socket and an acknowledgement will be issued when the information has been received. The data supplied by the flasher PC will cover (at least) the full period of the Time Frame. If bookkeeping complications allow, a longer time period could be covered to minimise the number of requests over the LAN. The quantity of information required is estimated at ~ 100 bytes

which, at the frequency of once per Time Frame or less, will have no significant impact on the LAN traffic.

3. When the flasher analysis on a Time Frame has been completed the flasher PC will be notified over the LAN. This will allow the flasher system bookkeeping to monitor successful flashing and reschedule if required. There will be a timeout built in to the flasher PC bookkeeping software.
4. This is all considered part of the light injection software system for the farm and will be provided by Lisa. Close co-operation with the DAQ group will be maintained.

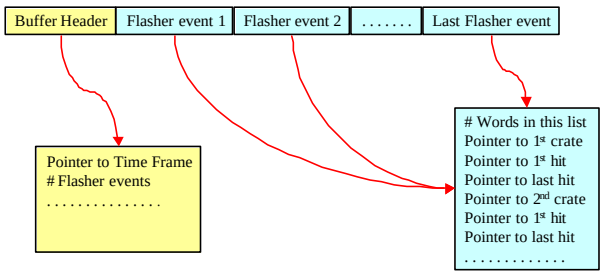
Interface to the flasher analysis.

The interface is bi-directional: passing light injection hits from the Time Frames to the analysis software and passing the output from the analysis back for output.

Passing hits to the analysis software

Since the quantity of data in a Time Frame from flasher hits is potentially large, a pointer system will be used for passing data to the LI process. The whole Time frame will be made available to the flasher software along with a pointer table identifying the location of the flasher events.

1. Two buffers will be reserved in physical memory to hold pointers to the flasher events within Time Frames. A single pointer buffer will be used to describe an entire Time Frame. Two such buffers should be sufficient. If more are required it means that the flasher analysis is taking longer to complete than all other Time Frame processing – a situation that is extremely unlikely but which would need to be remedied. The same buffer management package used elsewhere in the DAQ will be used to manage these buffers.
2. The trigger farm software will identify the flasher events within a Time Frame (using the PIN diode and/or the trigger PMT readout) and load pointers (offsets) to the flasher events into the TF pointer table. The number of non-flasher hits within the ~100ns span of a flasher event are negligible and will be included in the definition of the flasher event.
3. When the Time Frame processing has been completed by the triggering and monitoring processes, both the pointer table and the Time Frame are made available to the flasher process. Flasher processing proceeds. Mapping from physical to process memory for buffer referencing will be provided.
4. The farm will use multiple Time Frame buffers (as a tool for averaging out fluctuations in instantaneous rate). This allows the flasher analysis to proceed asynchronously and independently of the next Time Frame.
5. The format of the pointer buffer will be a header followed by a list of pointers relative to the start of the Time Frame buffer:



Passing analysis results back for O/P.

1. A buffer will be reserved in physical memory of the TP for the o/p from the flasher analysis. This buffer will be used to hold the results of the analysis of a complete Time Frame.
2. The format of the flasher O/P will follow the simple data block structure used for DAQ output, i.e. a header followed by the data. The data can contain sub-blocks if required. Lisa will define the content and format of this block in consultation with the DAQ group and others in the near future. It is estimated that each time frame containing flasher data will produce up to 15KB (25KB when timing summaries are included) of summary for output if the cross talk channels can be ignored. Writing summaries for the cross talk channels as well would increase this accordingly.
3. Completed summaries will be passed by the farm software to the DAQ DDS process for output to the ROOT file by the same route as all other data from the farm.