

MicroBooNE Data structures/structs/ classes: an Introductory discussion

All pre-beamdata merge.

Eric, 10-Oct-2012

Right now, in fact, seb_trig passes on the data almost faithfully from the crates, tacking on header, trailer, then shipping to assembler.

```
typedef struct event_fragment_header
{
    uint16_t size; //bytes
    uint16_t crateNum; // # of Crates in run
    uint16_t event_number; // Event #
    uint16_t frame_number; // Frame #
    gps_t gps; // from GPS card! From crate 10.
    uint32_t place_holder; //
}event_fragment_header_t;
```

Then 1x:

```
uint32_t event_header;
```

Then crateNum x:

```
{
    typedef struct event_module_header
    {
        uint32_t id_and_module;
        uint32_t word_count;
        uint32_t event_number;
        uint32_t frame_number;
        uint32_t checksum;
    }event_module_header_t;
```

and N uint16_t data words, until I see end of event word

```
}
```

```
typedef struct event_fragment_trailer
{
    uint32_t place_holder; //
}event_fragment_header_t;
```

Assembler receives these and adds ...

From Assembler.

```
/* global_header. DAQ_version_number possible values.
   This is really a DATA-output version number */
#define VERSION 1

typedef struct {
    static const uint8_t DAQ_version_number = VERSION;
    uint8_t record_type; /* From event_types.h */
    uint8_t record_origin; /* DATA or MC */
    uint32_t run_number;
    uint32_t event_number;
    uint32_t event_number_crate; /* Crate's sense of the evt #. */

    uint32_t seconds; // GPS clock. Since Jan 1, 2012.
        // Do we need to worry about Leap seconds?
    uint16_t milli_seconds;
    uint16_t micro_seconds;
    uint16_t nano_seconds;
    uint8_t spare8_1;
    uint8_t spare8_2;
    uint8_t spare8_3;
    uint8_t spare8_4;
    uint16_t spare16_1;
    uint16_t spare16_2;
    uint16_t spare16_3;
    uint16_t spare16_4;
    uint32_t number_of_bytes_in_the_record;
    uint32_t spare32_1;
} global_header_t; /* bytes */
```

Requires opening and interrogating streams on individual seb_trig machines/processes. Should consider seb_SN streams too.

From seb_trig_10.

```
/* the trigger_data structure includes everything the
 * trigger is going to send to assembler. Will come faithfully
 * from trigger card. Hence, below is only a guess. Need to
 * get this from Chi/Bill.
 */
#define N_ACTIVITY_HIST 4
typedef struct trigger_data {
    uint32_t  trig_event_num; /* trigger_event_number */
    gps_t     gps_time;      /* gps_time of broadcast */
    uint16_t  trig_event_type; /* trigger event type e.g. beam, calib */
    uint16_t  frame; /* frame # where trigger happened*/
    uint64_t  clock; /* Master Crate Clock value where trigger happened*/
} trigger_data_t; /* xyz bytes */
```

From seb_trig_1-9.

```
typedef struct {
    crate_header tpc; /* This is actually the 1st crate header (from crate 1).
The header from each of the crates are checked for consistency in assembler
*/
    uint32_t  crate_event_num; /* crate's_event_number */
    uint16_t  number_of_hitwires;
    uint16_t  rate;
} tpc_header_t; /* __ bytes */
```

```
typedef struct {
    crate_header pmt; /* This is the only PMT crate header (from crate 10).
The header from each of the crates are checked for consistency in assembler
*/
    uint32_t  crate_event_num; /* crate's_event_number */
    uint16_t  number_of_hitpmts;
    uint16_t  rate;
} pmt_header_t; /* __ bytes */
```