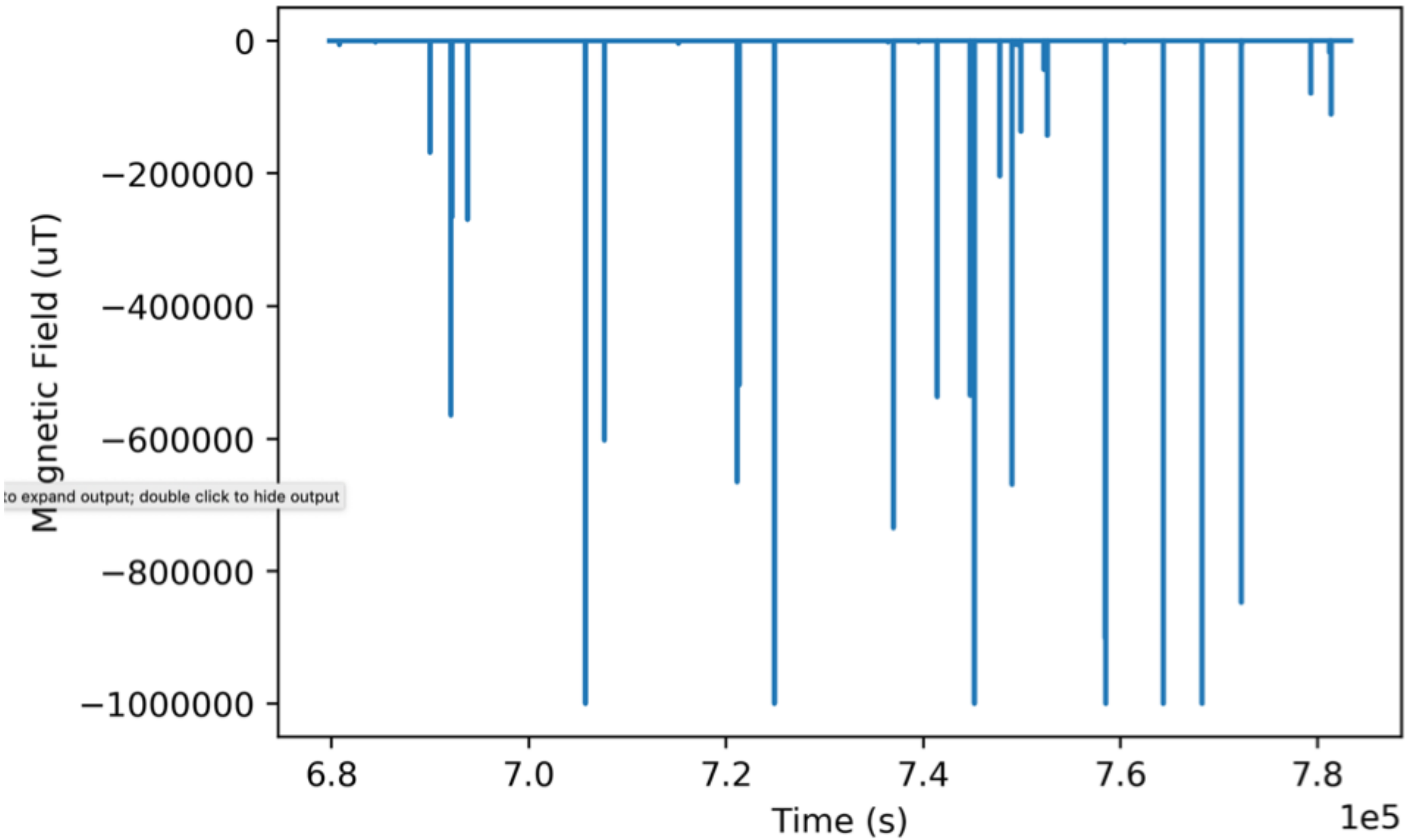
# Debugging fluxgates

17/06/2019

Takashi Higuchi
Xander Naumenko

# From the manual of T7 series

- Found here: https://ucn.triumf.ca/edm/magnetic-shielding/fluxgates-and-daq
- A few things of our interest:
  - 3.2 Stream Mode:
    - Stream mode is the way you get the highest input rate
    - *The T7 uses a feature called auto-recovery. If the buffer overflows, the T7 will continue streaming but discard data until the buffer is emptied, and then data will be stored in the buffer again. The T7 keeps track of how many scans are discarded and reports that value. Based on the number of scans discarded, the LJM driver adds the proper number of dummy samples (**-9999.0**) such that the correct timing is maintained. Auto-recover will only work if the first channel in the scan is an analog channel.*
    - # Another place where this number is found is 14.0 AIN about **AIN_ALL_RANGE**
      *A read will return the correct setting if all channels are set the same, but otherwise will return -9999.*
- Maximum data rate depends on the STREAM_RESOLUTION_INDEX (we use 0) and also on the # of channels used (15 chs not even in the table): see next page




- Scan rate is 500 scans/second
- Plotted here is the (averaged value of 500 buffer) * 100
- Sometimes, all the 500 in one buffer are -9999
- But even this happens, it is at maximum ~ 3 sec according to Xander

# From the manual of T7 series (Stream Data rate)

## Streaming Data Rates

The fastest data rates on the T7 occur when operating in stream mode. Much of the command response overhead is eliminated in stream mode because the T7 is responsible for initiating IO operations. Collected data is stored in the T7's stream buffer it is retrieved by the host application. The end result is a continuous data stream, sampled at regular intervals, collected with a minimum number of command response data sets [2.]. Table A1.4 and A1.5 provide typical stream-related performance results. The tabulated data is useful for determining what types of signals can be analyzed using a T7. The T7 is capable of streaming analog data at regular discrete intervals. As a result, various discrete time signal analysis tools can be utilized to interpret data.

The scan rates shown in table A1.4 are continuous over USB or Ethernet. When using **WiFi**, the device can acquire data at the fastest rates, but transfer of data to the host is limited to about 1 ksamples/second, so the fastest stream rates cannot be maintained continuously. In this case stream-burst can be used rather than continuous stream, where each stream is limited to a specified number of scans that fits in the device's stream buffer.

**Table A1.4.** Stream scan rates for stream mode over various gain, resolution index, channel count combinations.

| | Gain : Range | Maximum Scan Rate | | | | Maximum Sample Rate |
|---|---|---|---|---|---|---|
| | | 1 Channel [Hz] | 2 Channels [Hz] | 4 Channels [Hz] | 8 Channels [Hz] | >1 Channel [Hz] |
| Resolution Index = 1 | 1 : ±10V | 100k | 43.7k | 22.5k | 11.3k | 87.4k |
| | 10 : ±1V | 100k | 4.1k | 1.4k | 585 | 8.2k |
| | 100 : ±0.1V | 100k | 850 | 315 | N.S. | 1.7k |
| | 1000 : ±0.01V | N.S. | N.S. | N.S. | N.S. | N.S. |
| Resolution Index = 2 | 1 : ±10V | 48.0k | 19.8k | 9.0k | 4.0k | 39.6k |
| | 10 : ±1V | 48.0k | 3.6k | 1.3k | 550 | 7.2k |
| | 100 : ±0.1V | 48.0k | 400 | N.S. | N.S. | 800 |
| | 1000 : ±0.01V | N.S. | N.S. | N.S. | N.S. | N.S. |
| Resolution Index = 3 | 1 : ±10V | 22.0k | 9.9k | 4.5k | 2.4k | 19.8k |
| | 10 : ±1V | 22.0k | 1.4k | 500 | 225 | 2.8k |
| | 100 : ±0.1V | N.S. | N.S. | N.S. | N.S. | N.S. |
| | 1000 : ±0.01V | N.S. | N.S. | N.S. | N.S. | N.S. |
| Resolution Index = 4 | 1 : ±10V | 11.0k | 4.9k | 2.2k | 1.3k | 9.8k |
| | 10 : ±1V | 11.0k | 1.3k | 45 | N.S. | 2.6k |
| | 100 : ±0.1V | N.S. | N.S. | N.S. | N.S. | N.S. |
| | 1000 : ±0.01V | N.S. | N.S. | N.S. | N.S. | N.S. |
| Resolution Index = 5 | 1 : ±10V | 5.5k | 2.2k | 990 | 630 | 4.4k |
| | 10 : ±1V | 5.5k | 630 | 23 | N.S. | 1.3k |
| | 100 : ±0.1V | N.S. | N.S. | N.S. | N.S. | N.S. |
| | 1000 : ±0.01V | N.S. | N.S. | N.S. | N.S. | N.S. |
| Resolution Index = 6 | 1 : ±10V | 2.5k | 1.3k | 630 | 315 | 2.6k |
| | 10 : ±1V | 2.5k | 320 | N.S. | N.S. | 640 |
| | 100 : ±0.1V | N.S. | N.S. | N.S. | N.S. | N.S. |
| | 1000 : ±0.01V | N.S. | N.S. | N.S. | N.S. | N.S. |
| Resolution Index = 7 | 1 : ±10V | 1.2k | 650 | 315 | N.S. | 1.3k |
| | 10 : ±1V | 1.2k | 220 | N.S. | N.S. | 440 |
| | 100 : ±0.1V | N.S. | N.S. | N.S. | N.S. | N.S. |
| | 1000 : ±0.01V | N.S. | N.S. | N.S. | N.S. | N.S. |
| Resolution Index = 8 | 1 : ±10V | 600 | 315 | N.S. | N.S. | 630 |
| | 10 : ±1V | 600 | 200 | N.S. | N.S. | 400 |
| | 100 : ±0.1V | N.S. | N.S. | N.S. | N.S. | N.S. |
| | 1000 : ±0.01V | N.S. | N.S. | N.S. | N.S. | N.S. |

*N.S. indicates settings not supported in stream mode.

# From the manual of T7 series: maximum data rate of AIN

**Table A.3.1.1.** Effective resolution and sampling times for various gains and resolution index settings. Resolution index settings 9-12 apply to the T7-Pro only.

| Resolution Index | Effective Resolution [bits] | Effective Resolution [µV] | AIN Sample Time [ms/sample] |
|---|---|---|---|
| **Gain/Range: 1/±10V** | | | |
| 1 | 16.0 | 316 | 0.04 |
| 2 | 16.5 | 223 | 0.04 |
| 3 | 17.0 | 158 | 0.06 |
| 4 | 17.5 | 112 | 0.09 |
| 5 | 17.9 | 85 | 0.16 |
| 6 | 18.3 | 64 | 0.29 |
| 7 | 18.8 | 45 | 0.56 |
| 8 | 19.1 | 37 | 1.09 |
| 9 | 19.6 | 26 | 3.50 |
| 10 | 20.5 | 14 | 13.4 |
| 11 | 21.4 | 7.5 | 66.2 |
| 12 | 21.8 | 5.7 | 159 |
| **Gain/Range:10/±1V** | | | |
| 1 | 15.4 | 48 | 0.23 |
| 2 | 16.0 | 32 | 0.23 |
| 3 | 16.5 | 22 | 0.55 |
| 4 | 16.9 | 17 | 0.58 |
| 5 | 17.4 | 12 | 1.15 |
| 6 | 17.9 | 8.5 | 2.28 |
| 7 | 18.3 | 6.4 | 2.55 |
| 8 | 18.7 | 4.9 | 3.08 |
| 9 | 19.5 | 2.8 | 3.50 |
| 10 | 20.5 | 1.4 | 13.4 |
| 11 | 21.4 | 0.7 | 66.2 |
| 12 | 21.7 | 0.6 | 159 |
| **Gain/Range: 100/±0.1V** | | | |
| 1 | 13.3 | 21 | 1.03 |
| 2 | 14.2 | 11 | 2.03 |
| 3 | 14.7 | 7.8 | 5.05 |
| 4 | 15.2 | 5.5 | 5.08 |
| 5 | 15.7 | 3.9 | 5.15 |
| 6 | 16.3 | 2.6 | 10.28 |
| 7 | 16.7 | 1.9 | 10.55 |
| 8 | 17.2 | 1.4 | 11.08 |
| 9 | 18.3 | 0.6 | 3.50 |
| 10 | 19.1 | 0.4 | 13.4 |
| 11 | 19.6 | 0.3 | 66.2 |
| 12 | 19.7 | 0.2 | 159 |
| **Gain/Range: 1000/±0.01V** | | | |
| 1 | 10.9 | 11 | 5.03 |
| 2 | 12.3 | 4.1 | 10.0 |
| 3 | 12.7 | 3.1 | 10.1 |
| 4 | 13.3 | 2.1 | 10.1 |
| 5 | 13.8 | 1.5 | 10.2 |
| 6 | 14.4 | 1.0 | 10.3 |
| 7 | 14.7 | 0.8 | 10.6 |
| 8 | 15.0 | 0.6 | 11.1 |
| 9 | 15.4 | 0.5 | 3.50 |
| 10 | 16.1 | 0.3 | 13.4 |
| 11 | 16.4 | 0.2 | 66.2 |
| 12 | 16.4 | 0.2 | 159 |

- Depends on the resolution index and gain/range, 0.5 – 160 ms (200 ms <-> 5 Hz is safe)

# Discussions with Thomas

# He was not very sure about Labjack-related things, but gave us some advice from what he knows about MIDAS

- It is possible that sometimes network hacks make the dead time in connection
- But he was pretty sure that MIDAS is not causing the delay, he suspects the Labjack side
- He doesn't have any strong preference on which commands to be used, either of eReadName or eStreamRead
- The channel shift problem: He suspects that channels recorded in buffer might somehow be misidentified when eStreamRead is called
- Suggestions:
  - Record the time stamp before and after the eStreamRead in the code to check whether they are executed flawlessly
  - Perhaps make force initialization of the buffer when strange things happen

# Discussions with Thomas

- To use MIDAS from C:
- Create a frontend by a "equipment" class, you need all the headers such as

```
/* The frontend name (client name) as seen by other MIDAS clients */
  char *frontend_name = "feLabjack02";
  /* The frontend file name, don't change it */
  char *frontend_file_name = __FILE__;

  /* frontend_loop is called periodically if this variable is TRUE */
  BOOL frontend_call_loop = TRUE;
  …
```

- Then it appears as one of the equipment lists on MIDAS

| Equipment | | | | |
|---|---|---|---|---|
| **Equipment +** | **Status** | **Events** | **Events[/s]** | **Data[MB/s]** |
| scPico | Frontend stopped | 0 | 0.0 | 0.000 |
| SourceEpics | Ok | 53692 | 0.0 | 0.000 |
| BeamlineEpics | Ok | 101893 | 0.2 | 0.000 |
| Li6_Detector | Initialized | 0 | 0.0 | 0.000 |
| HE3_Detector | Frontend stopped | 730.259M | 598.5 | 0.031 |
| Labjack02 | Frontend stopped | 2466 | 1.0 | 0.000 |
| UCNSequencer2018 | Frontend stopped | 146 | 9.7 | 0.001 |
| MCC | Frontend stopped | 0 | 0.0 | 0.000 |
| SCMtemp | Frontend stopped | 0 | 0.0 | 0.000 |
| chronobox00 | Frontend stopped | 1.154M | 2.3 | 0.000 |
| Labjack01 | Frontend stopped | 19 | 1.0 | 0.000 |

# Discussions with Daryl

# He seems to be uncomfortable with the solution of positioning the power supply which Elspeth finally chose

- The power regulators and the fluxgate outputs could be all coupled to the power convertor through inductive coupling of the circuit. Could make them noisy
- Another problem of it is that a failure on the power line could damage all the components on the board
- Suggestion: try putting a test signal on the power line and see whether you see the effect on the analog read-out and fluxgates supply lines
- He gave us a SOLA SCP20T515-DN (EMERSON) power convertor and advised to replace it to the existing one, placing it far away the board in the box, or even outside the box (But he also said that it is not an easy work to replace parts from the board)

- Takashi's comment: we are in principle interested in a very low-frequency range. (<10Hz), We should first check how are the frequencies of this noise, and if we see the need, we should replace the power convertor. The test should be done with the power in the Meson Hall.

# Administrative things

- Budget code?
- Agresso account (or, shall we ask Florian every time?)

- The board layout was not found on the link shared by Beatrice (it will be much needed later when we do hardware debugging), Only found some Altimum files, I don't think this is the final version of the board (and I cannot open the Altimum files)
- Any caution in dealing with the board? (I heard that all the components burned out by one mis-operation previously)

# ToDo's

- One plan: Takashi would like to run a commercial probe (Hall or NMR) in parallel to check the system. It seems that Alex from electronics group is in charge of it. He will come back next week. Realizability of this may change what we would do next

- Software debugging:
  1. Continue tests with eReadName for a few more days
     - The last two days, we saw ~200 nT shifts from around 6 AM, good to confirm reproducibility
  2. Insert the time record in the online code as suggested by Thomas
  3. Make a simpler code of sStreamRead, too
  4. Check how much is the fastest rate to use eStreamRead/eReadName
  5. Purposely set blank of communication to see whether this causes -9999 to be returned
- Hardware check:
  1. Uninstall the board from the fluxgates, check how noise the signal becomes by the power convertor influences
  2. Probe each line as suggested by Daryl

# Except 1., the software debugging can be done in parallel to the hardware test. But disassembling of the system would be irreversible, so we should consider and prepare well beforehand.