SAMPIC 256-Ch Software & Library

J.Maalmi/D. Breton

august 2020

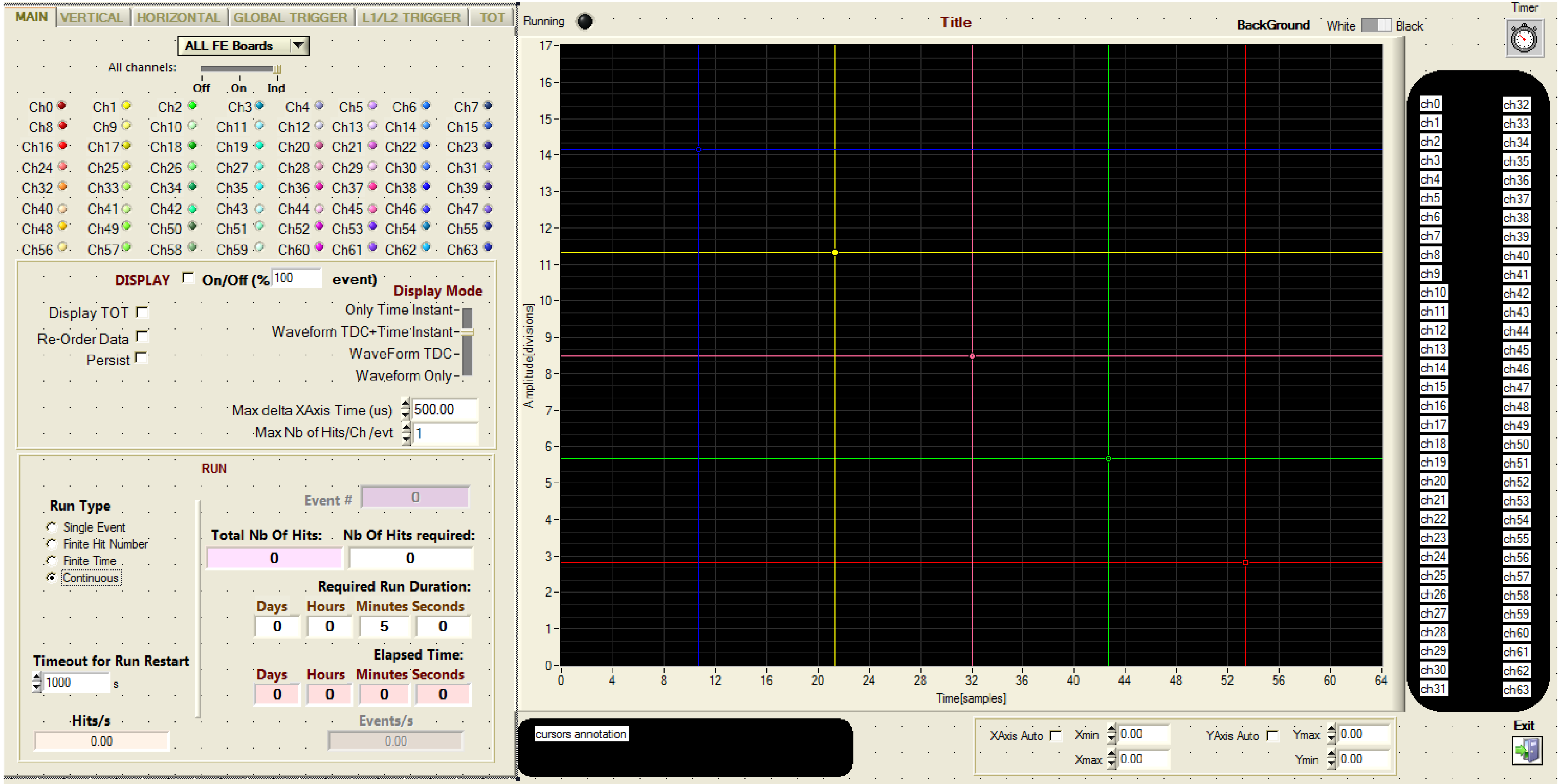
SAMPIC-256 Ch Software

- Developed under **Labwindows CVI**
- Based on the low level **LpDevC library** (handles USB and UDP, lal protocol)
- The UDP is 1-Gbit/s , secured LAL protocol (no frames lost because of a handshake)
  - ➢ UDP : **local** connexion with host, **no DHCP, no ping**.
  - ➢ IP address(es) and IP port(s) can be chosen and written to FLASH memory in the crate.
- The USB is USB-2.0 : (~ 15 MBytes/s) LAL protocol.
- based on the High-Level SAMPIC-256ch_Library (developement of library and Software in parallel …)

SAMPIC-256 Ch Software Features

- Software only for **Windows**
- Handles a **256-ch crate**
- 2 options for the **Control** and **Readout**
  - **single USB** or **UDP** connexion to the Controller Board for Control and Data Readout → option for T2K.
  - UDP connexion for Control Only (Controller Board) and 4 individual UDP connexions for data Readout on eac 64-ch board.
- Graphical Interface : displaying the waveforms, Trigger management, online Time measurements etc…
- Possibility to Save Data to files (Binary/ASCII)
- Data are **not Ordered in time**.
- Possibility to perform the needed calibrations (4 types of calibrations) and saving calibration files.

SAMPIC-256 Ch Software: GUI (1/3)



MAIN
VERTICAL
HORIZONTAL
GLOBAL TRIGGER
L1/L2 TRIGGER
TOT
Ch0 Ch1 Ch2 Ch3 Ch4 Ch5 Ch6 Ch7
Ch8 Ch9 Ch10 Ch11 Ch12 Ch13 Ch14 Ch15
Ch16 Ch17 Ch18 Ch19 Ch20 Ch21 Ch22 Ch23
Ch24 Ch25 Ch26 Ch27 Ch28 Ch29 Ch30 Ch31
Ch32 Ch33 Ch34 Ch35 Ch36 Ch37 Ch38 Ch39
Ch40 Ch41 Ch42 Ch43 Ch44 Ch45 Ch46 Ch47
Ch48 Ch49 Ch50 Ch51 Ch52 Ch53 Ch54 Ch55
Ch56 Ch57 Ch58 Ch59 Ch60 Ch61 Ch62 Ch63
Running
Title
Timer
Amplitude[divisions]
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
Time[samples]
0 4 8 12 16 20 24 28 32 36 40 44 48 52 56 60 64
ch0 ch1 ch2 ch3 ch4 ch5 ch6 ch7 ch8 ch9 ch10 ch11 ch12 ch13 ch14 ch15 ch16 ch17 ch18 ch19 ch20 ch21 ch22 ch23 ch24 ch25 ch26 ch27 ch28 ch29 ch30 ch31
ch32 ch33 ch34 ch35 ch36 ch37 ch38 ch39 ch40 ch41 ch42 ch43 ch44 ch45 ch46 ch47 ch48 ch49 ch50 ch51 ch52 ch53 ch54 ch55 ch56 ch57 ch58 ch59 ch60 ch61 ch62 ch63
cursors annotation
Exit

ALL FE Boards

All channels: Off On Ind

DISPLAY □ On/Off (% 100 event)

Display TOT □

Re-Order Data □

Persist □

Display Mode
- Only Time Instant
- Waveform TDC+Time Instant
- WaveForm TDC
- Waveform Only

Max delta XAxis Time (us) 500.00

Max Nb of Hits/Ch /evt 1

RUN

Run Type
- ○ Single Event
- ○ Finite Hit Number
- ○ Finite Time
- ● Continuous

Event # 0

Total Nb Of Hits: 0

Nb Of Hits required: 0

Required Run Duration:

Days	Hours	Minutes	Seconds
0	0	5	0

Elapsed Time:

Days	Hours	Minutes	Seconds
0	0	0	0

Timeout for Run Restart 1000 s

Hits/s 0.00

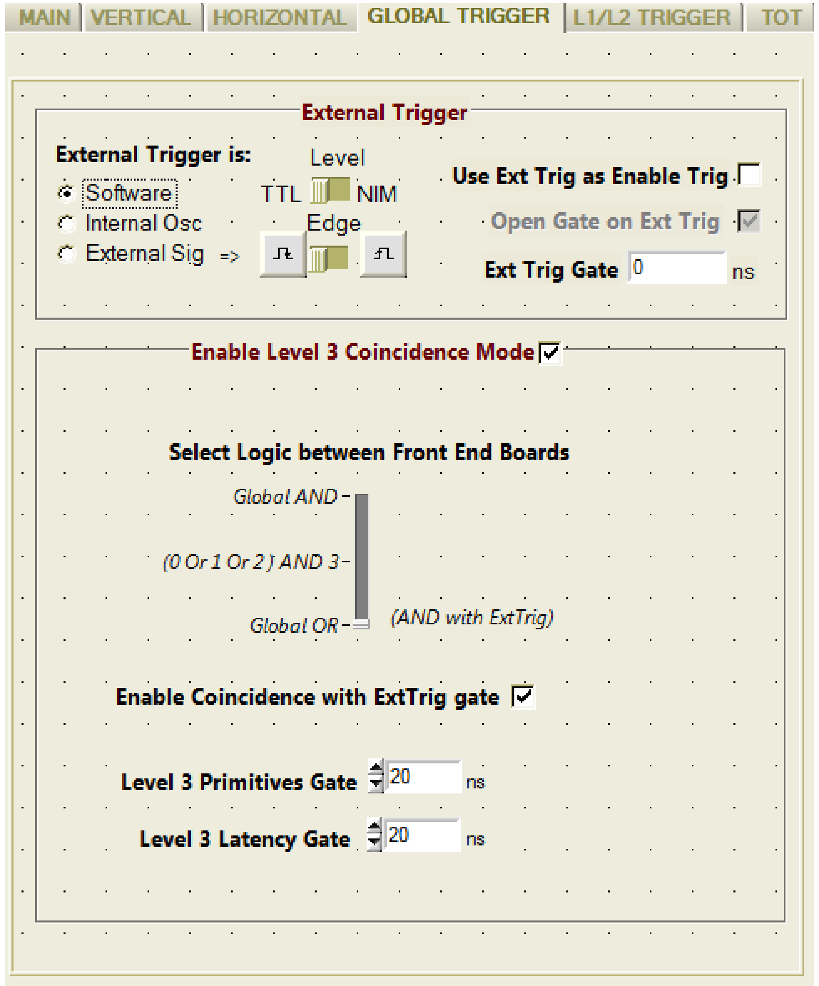
Events/s 0.00

BackGround White Black

XAxis Auto □ Xmin 0.00 Xmax 0.00

YAxis Auto □ Ymax 0.00 Ymin 0.00

SAMPIC-256 Ch Software: GUI (2/2)



External Trigger

External Trigger is: ● Software ○ Internal Osc ○ External Sig =>

Level: TTL / NIM

Edge

Use Ext Trig as Enable Trig □

Open Gate on Ext Trig ☑

Ext Trig Gate 0 ns

Enable Level 3 Coincidence Mode ☑

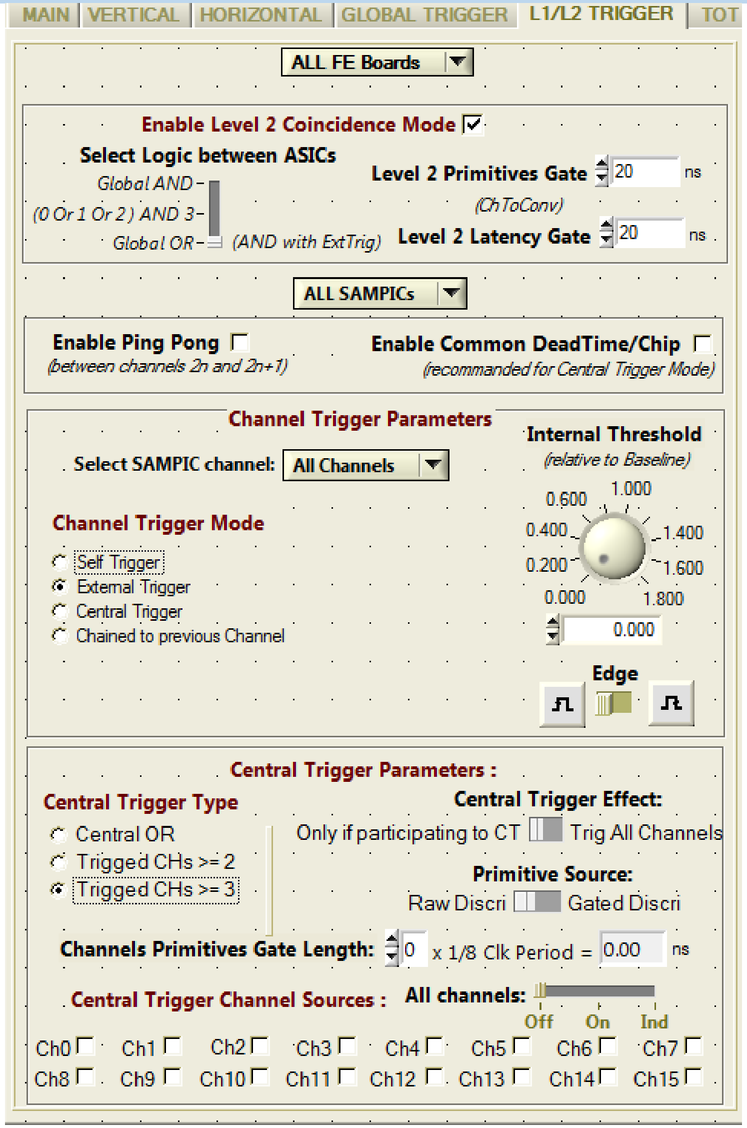
Select Logic between Front End Boards

Global AND –
(0 Or 1 Or 2) AND 3 –
Global OR – (AND with ExtTrig)

Enable Coincidence with ExtTrig gate ☑

Level 3 Primitives Gate 20 ns

Level 3 Latency Gate 20 ns



ALL FE Boards

Enable Level 2 Coincidence Mode ☑

Select Logic between ASICs

Global AND –
(0 Or 1 Or 2) AND 3 –
Global OR – (AND with ExtTrig)

Level 2 Primitives Gate 20 ns
(ChToConv)

Level 2 Latency Gate 20 ns

ALL SAMPICs

Enable Ping Pong □
(between channels 2n and 2n+1)

Enable Common DeadTime/Chip □
(recommanded for Central Trigger Mode)

Channel Trigger Parameters

Select SAMPIC channel: All Channels

Channel Trigger Mode

○ Self Trigger
● External Trigger
○ Central Trigger
○ Chained to previous Channel

Internal Threshold
(relative to Baseline)

0.000

Edge

Central Trigger Parameters :

Central Trigger Type

○ Central OR
○ Trigged CHs >= 2
● Trigged CHs >= 3

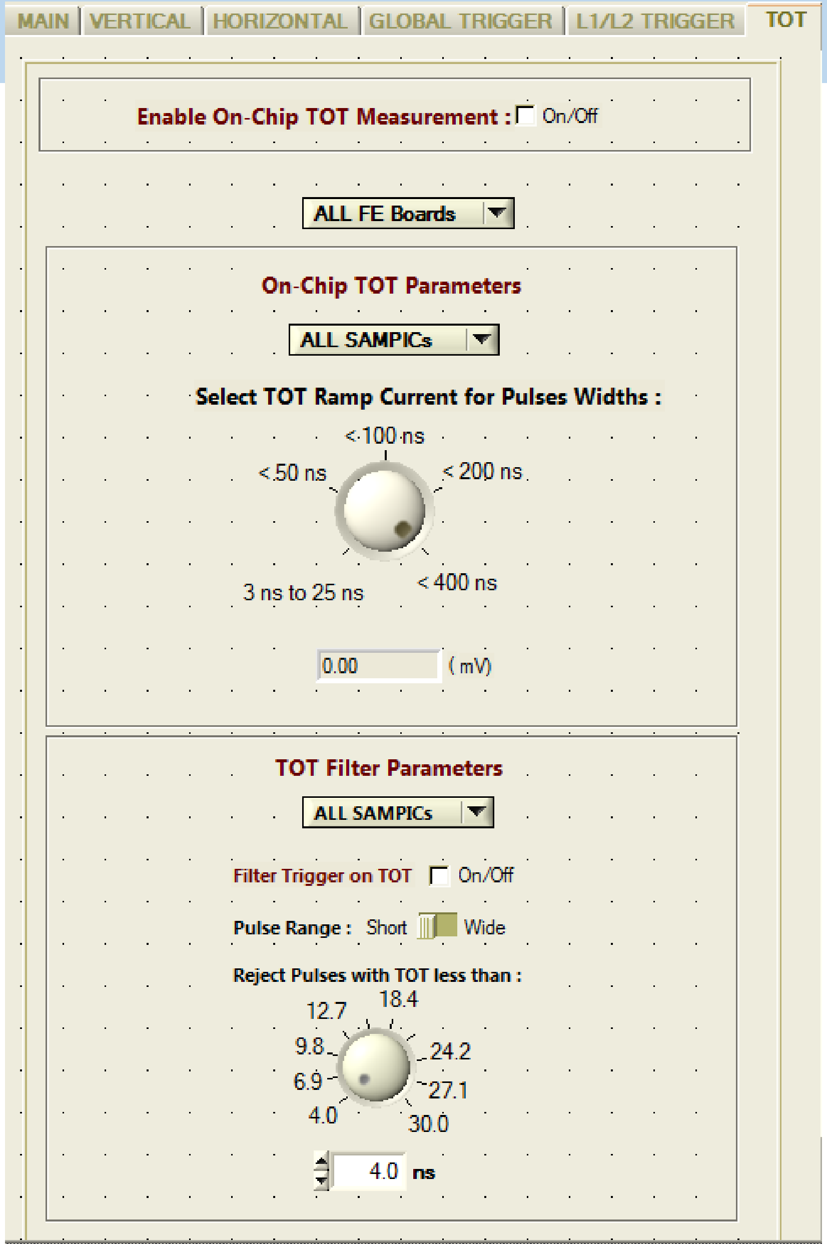
Central Trigger Effect:
Only if participating to CT / Trig All Channels

Primitive Source:
Raw Discri / Gated Discri

Channels Primitives Gate Length: 0 x 1/8 Clk Period = 0.00 ns

Central Trigger Channel Sources : All channels: Off On Ind

Ch0 □ Ch1 □ Ch2 □ Ch3 □ Ch4 □ Ch5 □ Ch6 □ Ch7 □
Ch8 □ Ch9 □ Ch10 □ Ch11 □ Ch12 □ Ch13 □ Ch14 □ Ch15 □



Enable On-Chip TOT Measurement : □ On/Off

ALL FE Boards

On-Chip TOT Parameters

ALL SAMPICs

Select TOT Ramp Current for Pulses Widths :

0.00 (mV)

TOT Filter Parameters

ALL SAMPICs

Filter Trigger on TOT □ On/Off

Pulse Range : Short / Wide

Reject Pulses with TOT less than :

4.0 ns

The Software developement

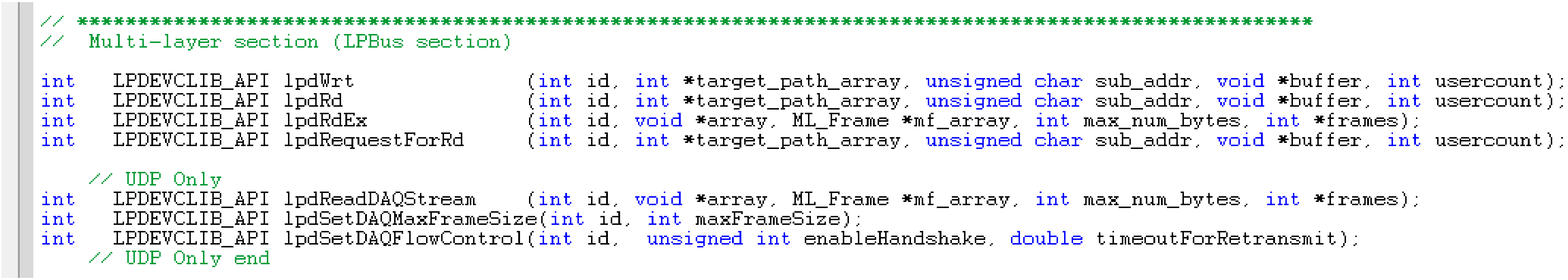
- Software (and Firmware) still under developement.

=> Possibility to add special features (for Triggering)

- Software can be used at first to start taking data before developing your own Software based on the library.

The low level LpDev library
(developed by Chafik Cheikali at IJClab)

- Low Level C/C++ Library that handles USB or UDP connexion, read/write etc…
- Version for **Linux and Windows**



```
// *****************************************************************************************************
//  Multi-layer section (LPBus section)

int   LPDEVCLIB_API lpdWrt              (int id, int *target_path_array, unsigned char sub_addr, void *buffer, int usercount);
int   LPDEVCLIB_API lpdRd               (int id, int *target_path_array, unsigned char sub_addr, void *buffer, int usercount);
int   LPDEVCLIB_API lpdRdEx             (int id, void *array, ML_Frame *mf_array, int max_num_bytes, int *frames);
int   LPDEVCLIB_API lpdRequestForRd     (int id, int *target_path_array, unsigned char sub_addr, void *buffer, int usercount);

    // UDP Only
int   LPDEVCLIB_API lpdReadDAQStream    (int id, void *array, ML_Frame *mf_array, int max_num_bytes, int *frames);
int   LPDEVCLIB_API lpdSetDAQMaxFrameSize(int id, int maxFrameSize);
int   LPDEVCLIB_API lpdSetDAQFlowControl(int id,  unsigned int enableHandshake, double timeoutForRetransmit);
    // UDP Only end
```

Documentation:

https://electronique.lal.in2p3.fr/echanges/lpDevLib/documentation/html/

Download:

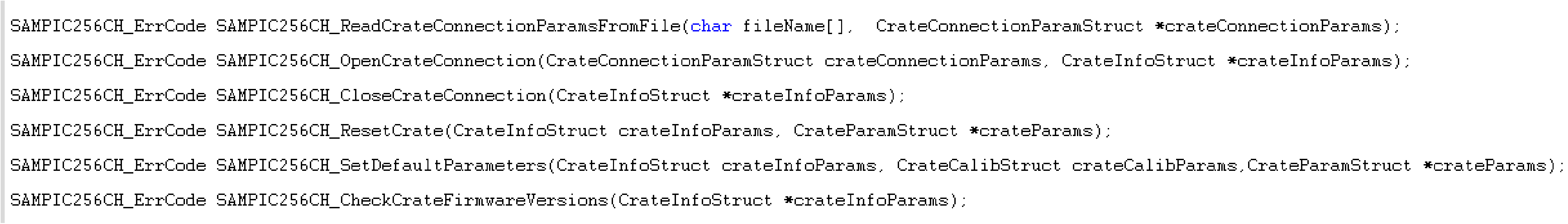
https://electronique.lal.in2p3.fr/echanges/lpDevLib/downloads/

The High level SAMPIC_256Ch library

- The Software is based on a this library (DLL under windows)
- Source Code will be furnished in order to build your own library under Linux.
- Library written in C.
- Source code from the SAMPIC-256Ch Software can be furnished as ‘sample’
- Files for the library
  - SAMPIC_256Ch_lib.c/ .h
  - SAMPIC_256Ch_hardware_core.c/.h
  - SAMPIC_256Ch_Type.h
  - lpDevC.h (and the static library for the Linux or .dll for Windows)

The High level SAMPIC_256Ch library

Example of C functions of the library:



```
SAMPIC256CH_ErrCode SAMPIC256CH_ReadCrateConnectionParamsFromFile(char fileName[],  CrateConnectionParamStruct *crateConnectionParams);

SAMPIC256CH_ErrCode SAMPIC256CH_OpenCrateConnection(CrateConnectionParamStruct crateConnectionParams, CrateInfoStruct *crateInfoParams);

SAMPIC256CH_ErrCode SAMPIC256CH_CloseCrateConnection(CrateInfoStruct *crateInfoParams);

SAMPIC256CH_ErrCode SAMPIC256CH_ResetCrate(CrateInfoStruct crateInfoParams, CrateParamStruct *crateParams);

SAMPIC256CH_ErrCode SAMPIC256CH_SetDefaultParameters(CrateInfoStruct crateInfoParams, CrateCalibStruct crateCalibParams,CrateParamStruct *crateParams);

SAMPIC256CH_ErrCode SAMPIC256CH_CheckCrateFirmwareVersions(CrateInfoStruct *crateInfoParams);
```

- High level functions to set Trigger options etc will be furnished. No need to use low level functions.
- Library will read Calibration files and perform data corrections.