	CALICE Electronics short meeting : minutes **USB module designed by Clement (LLR), Julie (LAPP) and Christophe (IPNL)**	Draft 0a

Revision status	**Comments**	**Date**
Draft 0	First version by RC. Distributed to participants for comments	18/06/08

CALICE Electronics short meeting: minutes

USB module

Held on Wednesday June 18

The purpose of this meeting was to present the USB module running for DHCAL and discuss about its functioning to determine if it could be suitable to others detectors.

The meeting started at 14:05 CEST.

List of the participants: Julie Prast (LAPP), Guillaume Vouters (LAPP), Christophe Combaret (IPNL), Mathias Reineke (DESY), Bart Hommels , Matt Waren, Franck Gastaldi (LLR), Clement Jauffret (LLR), Remi Cornat (LLR), Vincent Boudry (LLR)

Agenda on Indico : http://ilcagenda.linearcollider.org/conferenceDisplay.py?confId=2798 including FTDI chip data sheet, slides from Julie and Remi

1) General considerations about the USB module (Rémi)

See slides on the indico agenda.

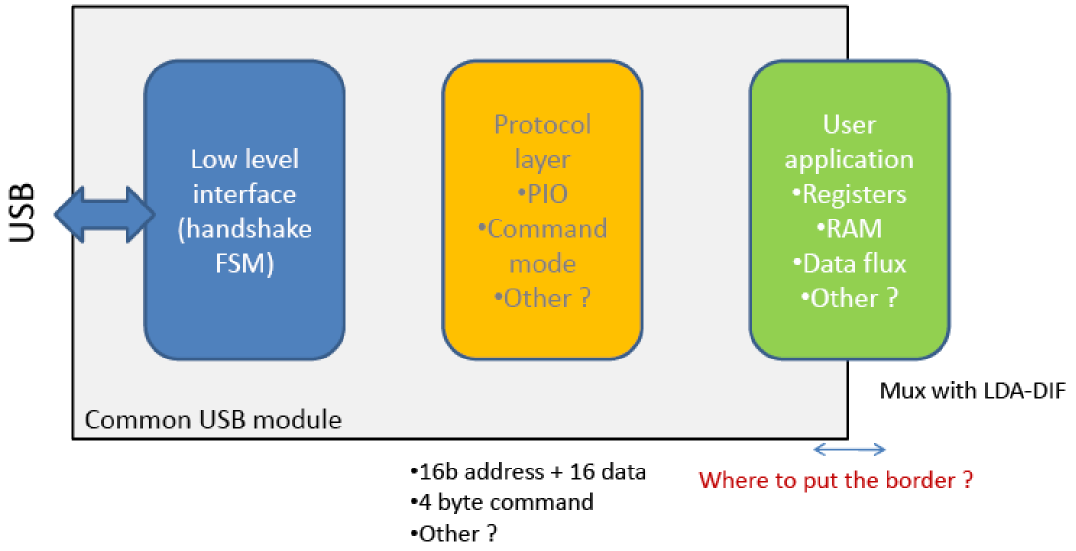
The USB link is based on commercial FTDI245BM chip which is a FIFO (buffers data between PC and FPGA), documentation available at http://www.ftdichip.com (HW and drivers)

The chip provides a simple interface to the FPGA (double handshake + 8b data bus) but with specific timing (depends on FPGA clock!).

Open questions about:

- User's needs: at least R/W registers (address+data) and commands (no address). What about a block transfer mode?
- Border of a common module (amount of application specific code inside the module)
- Link multiplexing priority (USB vs LDA-DIF and DIF-DIF)

Basic architecture of the module :



Notice: data transport is secured thanks to USB layers (assuming R/W in FIFO from FPGA is ok) but no higher level checking (packet ID, acknowledge,…)

2) Presentation of the existing FW (Julie)
   See Julie's slides on the agenda

The FW was first written by Clement Jauffret (LLR) and has been modified to keep uniform the byte order when building 16b words (MSB always first). The FW is at a "first release" state and should not be modified soon. It is being used and tested with SW (see point #3). An example of HW is given in Julie's slides.

Firmware available on
https://lyosvn.in2p3.fr/ilc/wiki/VHDL
Or
http://lappweb.in2p3.fr/~prast/GrosFichiers/VHDL_ILC
/VHDL_ILC.htm

It provides a parallel bus (16b address + 16b data) and an interface for commands (pulses). List is available on the slides.

The module includes 3 layers :

- R/W low level FSMs to manage transfer to/from the chip
- Transfer mode (bus or command) management bloc
- User specific (commands decoding and register bank)

User's specific components can be flavored to other detectors (code written for that).

Bloc transfer can be initiated by sending the appropriate command (e.g. write SC or Start Acquisition). External part (SC module, DAQ buffer controller?) takes the control of the USB R/W interface during the transfer. The appropriate number of words is send/received.

Code is running on ALTERA but does not contain specific instructions: it can be used on FPGAs from another manufacturer.

3) Presentation of existing SW (Christophe)

The code is working with the described firmware. The GUI and some high level features are made using XDAQ environment. A low level library is made to be generic enough; it can be used with other applications.

Code will be made available on Lyon Wiki server.

Mathias asked for a labview VI : examples are available on the FTDI web site. Julie and Remi proposed to give some examples (very application specific).

The following is related to the general discussion:

4) Need of a block transfer mode ?
Remi asked for a block transfer mode. It did not appear to be essential as the USB link will be used for debugging purposes. In addition, Julie highlighted that the transfer of large amount of bytes can be initiated by the SW by sending the appropriate command.
This is not a generic block transfer mode but it seems to be sufficient (SC and DAQ).

5) Interaction with LDA-DIF and DIF-DIF link
Bart gave some details about LDA-DIF-DIF links : a loop-back feature is used to determine if the link is active. If not, the DIF automatically switch to the DIF-DIF link which is supposed to provide the same features as the LDA-DIF link (and especially the clock).
When using USB, the priority is managed by the DIF at the level of the LDA-DIF module.

The USB link does not provide the clock. If the DIF is not connected to any LDA or another DIF, the CCC signals (at least the clock as control signals can be emulated with commands) must be provided thanks to some other way (embedded oscillator, LEMO connector …).

6) Use for DAQ (with synchronous signals)
Following a question from Mathias: The USB link cannot provides isochronous signals as it is a software driven point to point link between a PC and a single DIF. It therefore cannot be used to trigger actions at the same time on many DIFs.

To be used for DAQ purpose (instead of debugging) a dedicated CCC link should be used.

7) Next meetings
The following topics have been suggested: DAQ&CCC (running modes, signaling, synchronization…), DIF-LDA link.
In addition it was suggested (by Mathias in a private mail) to address some specific questions about HW (line drivers, PCB design, B field, clock speed etc…) during a dedicated meeting.
Remi asked whether these topics should be discussed with people physically present (as it should trigger long exchanges not easy to manage with EVO or equivalent). Julie suggested having a standard electronics meeting before the summer break. Open question!

The meeting ended at 15:20.

Conclusion on existing USB module: a first release of the code is made available by Julie and Christophe. The code is tested and used for the DHCAL. The existing module is flavored to the DHCAL but can be modified for other detectors. All required features seem to be implemented for data I/O (does not provide CCC).