Getting started with OCB **UT2** / ZTURN-V2 (**v1.0-455**)

Description

UT2 PS FW uses UT1PL FW.

UT2 PS FW similar to UT1 PS but now handles the complete data flow in the PL i.e. using a simulated MCB and a simulated FEB connected to a rolling buffer which produces data sent to the PS through DMA only if an event is received. The PS only received a interruption when the the DMA transfer initiated by the PS is completed at the end of each event, and push the event data to TCP.

The SIM_MCB produces event, gate and GTS signals. Gate and GTS signals are used by the SIM_FEB, event is used by the SIM_FEB and the Rolling buffer. MCB parameters are:

- *#Events per Gate*, driving a gate open, #N events (with prog. delay) per gate and a gate close and back to gate open etc… If parameter is set to 0 then the gate is open continuously
- *Event Delay* to set a delay between each event
- *Start* to start MCB (and FEB)

The SIM_FEB emulates FEB data with:

- Gate words (headers/time, trailers) on geta open/close from MCB
- GTS words (header/trailers) on GTS pulse from MCB
- RT/FT Data within GTS blocks in continuous mode (no event detected
- RT/FT/GH/LG Data within GTS block(s) in event mode (event detected)

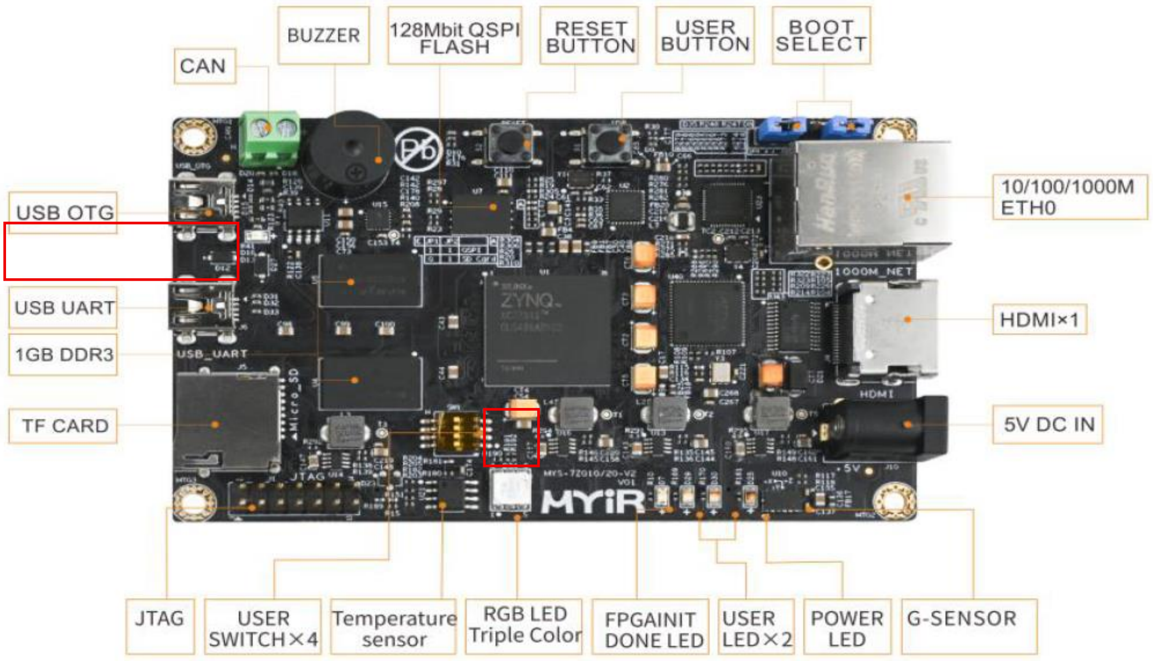
FEB parameters are:

- *T, HG, LG Enable* to enable RT/FT, HG and LG data for each channel. (HG/LG never sent in continuous mode)
- *#CH per GTS* to enable from 1 to 64-channels data per GTS
- *#GTS per Event* to enable data spread over #GTS for each event.

The Rolling Buffer can bin data in continuous mode if no event is detected but can also retrieve #N GTS blocks before the event detection. The parameter are:

- *#GTS before Event* to retrieve between 0 and 4 GTS blocks of data before the event detection

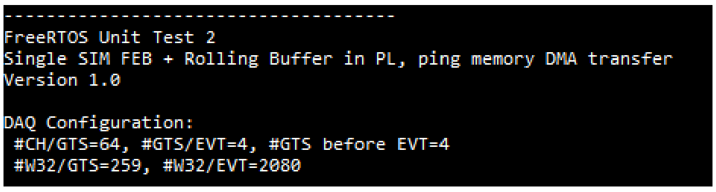
Quick start



1- Connect USB-UART to USB port on a PC. This will power on the board and you don't need to connect the external 5V PS.

2- Configure the USB port to be UART COM port with the following settings:
   - 115200 bit/s, 8-bits, no parity, 1 start, 1 stop

3- Connect a terminal on the configured COM port and reboot the board, the system will print the following and is listening for command interpreter client (CI) to connect on TCP port #50000 and DAQ client on TCP port #50001. UT2 terminal screen shown below:

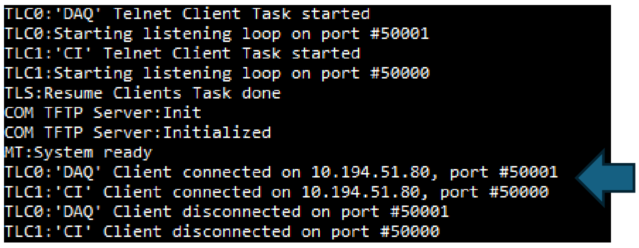


```
------------------------------------
FreeRTOS Unit Test 2
Single SIM FEB + Rolling Buffer in PL, ping memory DMA transfer
Version 1.0

DAQ Configuration:
 #CH/GTS=64, #GTS/EVT=4, #GTS before EVT=4
 #W32/GTS=259, #W32/EVT=2080
```

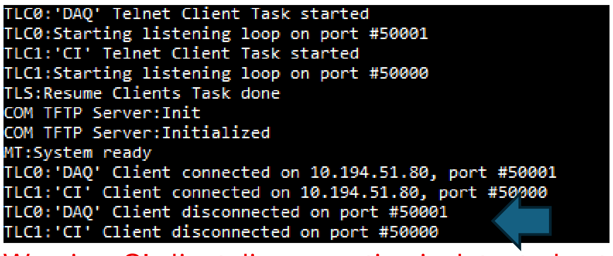
4- See 'UT1-Getting Started' for Mac & IP address setting. Inherited features:
   a. FAT partition
   b. TFTP Server
   c. UDP Logger
   d. `enablePrompt <En>` command

5- Connect the 2 clients, this will show:



```
TLC0:'DAQ' Telnet Client Task started
TLC0:Starting listening loop on port #50001
TLC1:'CI' Telnet Client Task started
TLC1:Starting listening loop on port #50000
TLS:Resume Clients Task done
COM TFTP Server:Init
COM TFTP Server:Initialized
MT:System ready
TLC0:'DAQ' Client connected on 10.194.51.80, port #50001
TLC1:'CI' Client connected on 10.194.51.80, port #50000
TLC0:'DAQ' Client disconnected on port #50001
TLC1:'CI' Client disconnected on port #50000
```

6- Disconnect the 2 clients, this will show:



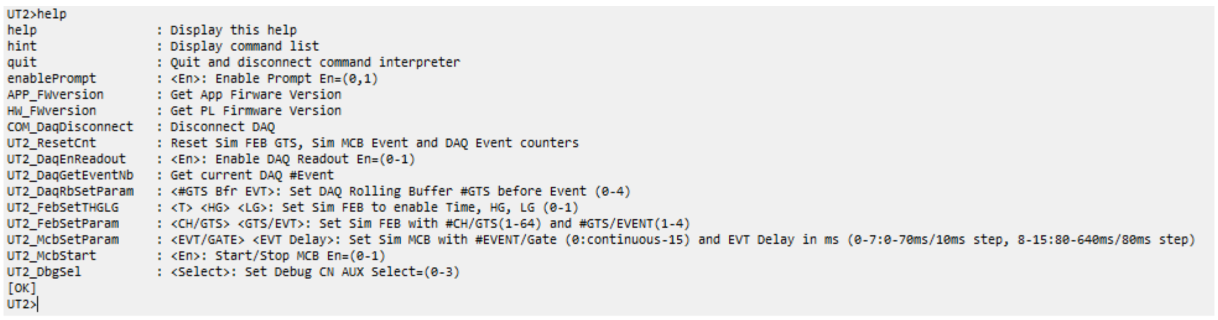
```
TLC0:'DAQ' Telnet Client Task started
TLC0:Starting listening loop on port #50001
TLC1:'CI' Telnet Client Task started
TLC1:Starting listening loop on port #50000
TLS:Resume Clients Task done
COM TFTP Server:Init
COM TFTP Server:Initialized
MT:System ready
TLC0:'DAQ' Client connected on 10.194.51.80, port #50001
TLC1:'CI' Client connected on 10.194.51.80, port #50000
TLC0:'DAQ' Client disconnected on port #50001
TLC1:'CI' Client disconnected on port #50000
```

Warning: CI client disconnection is detected automatically by the OCB but DAQ client disconnection is detected only during a DAQ transmission, meaning an abnormal situation since disconnection should be done when the readout is disabled.
To disconnect properly for DAQ and CI, one must:

a- Send the CI command `COM_DaqDisconnect` to force a disconnection
b- Disconnect the CI client

If not done, the DAQ client will remain connected and a new one couldn't be connected.

7- UT2 CI Commands with 'help':



```
UT2>help
help               : Display this help
hint               : Display command list
quit               : Quit and disconnect command interpreter
enablePrompt       : <En>: Enable Prompt En=(0,1)
APP_FWversion      : Get App Firware Version
HW_FWversion       : Get PL Firmware Version
COM_DaqDisconnect  : Disconnect DAQ
UT2_ResetCnt       : Reset Sim FEB GTS, Sim MCB Event and DAQ Event counters
UT2_DaqEnReadout   : <En>: Enable DAQ Readout En=(0-1)
UT2_DaqGetEventNb  : Get current DAQ #Event
UT2_DaqRbSetParam  : <#GTS Bfr EVT>: Set DAQ Rolling Buffer #GTS before Event (0-4)
UT2_FebSetTHGLG    : <T> <HG> <LG>: Set Sim FEB to enable Time, HG, LG (0-1)
UT2_FebSetParam    : <CH/GTS> <GTS/EVT>: Set Sim FEB with #CH/GTS(1-64) and #GTS/EVENT(1-4)
UT2_McbSetParam    : <EVT/GATE> <EVT Delay>: Set Sim MCB with #EVENT/Gate (0:continuous-15) and EVT Delay in ms (0-7:0-70ms/10ms step, 8-15:80-640ms/80ms step)
UT2_McbStart       : <En>: Start/Stop MCB En=(0-1)
UT2_DbgSel         : <Select>: Set Debug CN AUX Select=(0-3)
[OK]
UT2>
```

8- `EnablePromt <En>` can enable or disable prompt (enabled by default), see figure above

9- Readout quick start commands:

a. `UT2_FebSetTHGLG 1 1 1` to **configure SIM FEB data type sent** by enabling Timing, HG and LG on FEB simulator
b. `UT2_FebSetParam 5 2` to **configure SIM FEB** with an event having 5 channels enabled per GTS on 2 GTS (i.e. 10-ch sent over 2 GTS)
c. `UT2_McbSetParam 0 1` to **set a continuous gate open** and **a delay** of 10ms between each DAQ event sent over TCP
d. `UT2_ResetCnt` to **reset all counters**
e. `UT2_McbStart 1` to **start the MCB and the FEB**
f. `UT2_DaqEnReadout 1` to **start the readout** (DAQ client must be connected and listening to the data), OCB green led will be ON
g. `UT2_DaqEnReadout 0` to **stop the readout** (DAQ client must be connected and listening to the data), OCB green led will be OFF
h. `UT2_McbStart 0` to **stop the MCB and the FEB**
i. `UT2_DaqGetEventNb` to **retrieve the total number of events sent for this DAQ session**

Event data

For an event with 2 GTS with data & 1 GTS without any data, the correct sequence is now:

Event #0:

OCBDataHeader **//NB: Gate Tag is always 0 in this version**
GateHeaderA
GTSHeader
Data ...
GTSTrailer1
GTSTrailer2 (Data)
GTSHeader
Data ...
GTSTrailer1
GTSTrailer2 (Data)
GTSHeader
GTSTrailer1
GTSTrailer2 (No Data)
GTSHeader
EventDone
GTSTrailer1
GTSTrailer2 (No Data)
OCBFebDataTrailer
OCBDataTrailer **//NB: Gate Tag is always 0 in this version**

EventDone mistmatch with GATE:

- EventDone is sent by the FEB with the #words sent from GATE or EVENT message.
- In continuous GATE opening (i.e. like in FASER CAL), the gate message sent by the FEB is only sent once, basically at the start of the RUN, so if we have a coincidence right after the gate opening (unlikely), we got the correct #words in EventDone because the FEB will sent 3 consecutive words (Gate header 1 + Gate header 2 + Gate Time) : see FEB protocol
- However, if an EVENT message is received at least 1 GTS after the gate opening, the OCB artificially produce a GATE header 0 / FEB #x in order to notify a FEB data packet start for the FEB #x, Gate header 2 + Gate Time will never be sent in this case

Check to do on Gate word(s):

- only 1 word (GATE header 0 = artificial OCB):
  EventDone #Word = FEB data packet length - 1
- 3 words (Gate header 1 + Gate header 2 + Gate Time):
  EventDone #Word = FEB data packet length

T/HG/LG Word32 values:

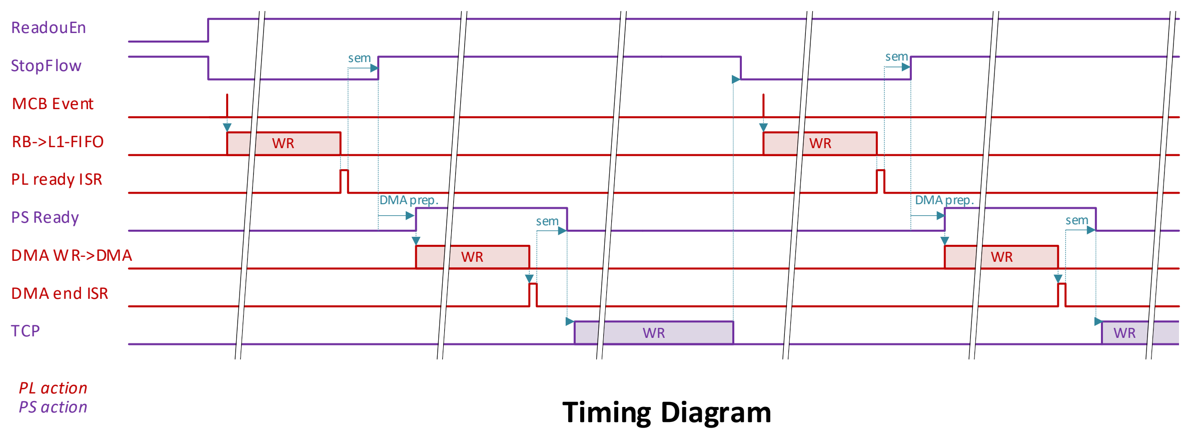
- Channel[7..0] = GTS/EVT[1..0] & CH/GTS[5..0]
- RT value[12..0] = 0 & GTS[4..0] & CH/GTS[5..0] & 0
- FT value[12..0] = 1 & GTS[4..0] & CH/GTS[5..0] & 1 = RT*2 +1
- HG value[11..0] = GTS[11..0]
- LG value[11..0] = 0 & GTS[10..0] = HG/2

NB: ***&*** *above is bit concatenation*

Continuous mode (before event):

Values seen in #GTS before event are only RT & FT, and number of channels seen is [#CH/GTS mod 16] + 1, e.g if #CH/GTS = 20 then 5 channels will be seen in GTS before event.

Timing



Timing Diagram