

Baby-Mind DAQ library

Yannick FAVRE
E. Noah
A. Blondel

University of Geneva

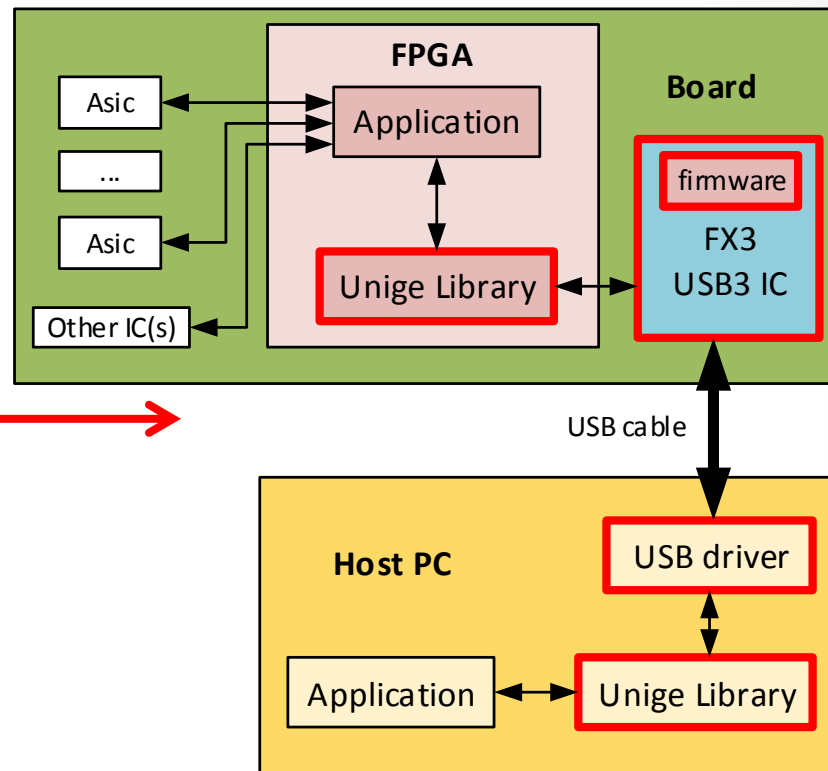
8-12-2017

Outline

- DAQ system overview 3
- Unige VHDL & UnigeFrontEnd C# libraries overview 4
- UnigeFrontEnd C# library architecture 5
- UnigeFrontEnd Configuration 6
- UnigeFrontEnd C# DLLs 9
- BabyMind 14

DAQ system overview

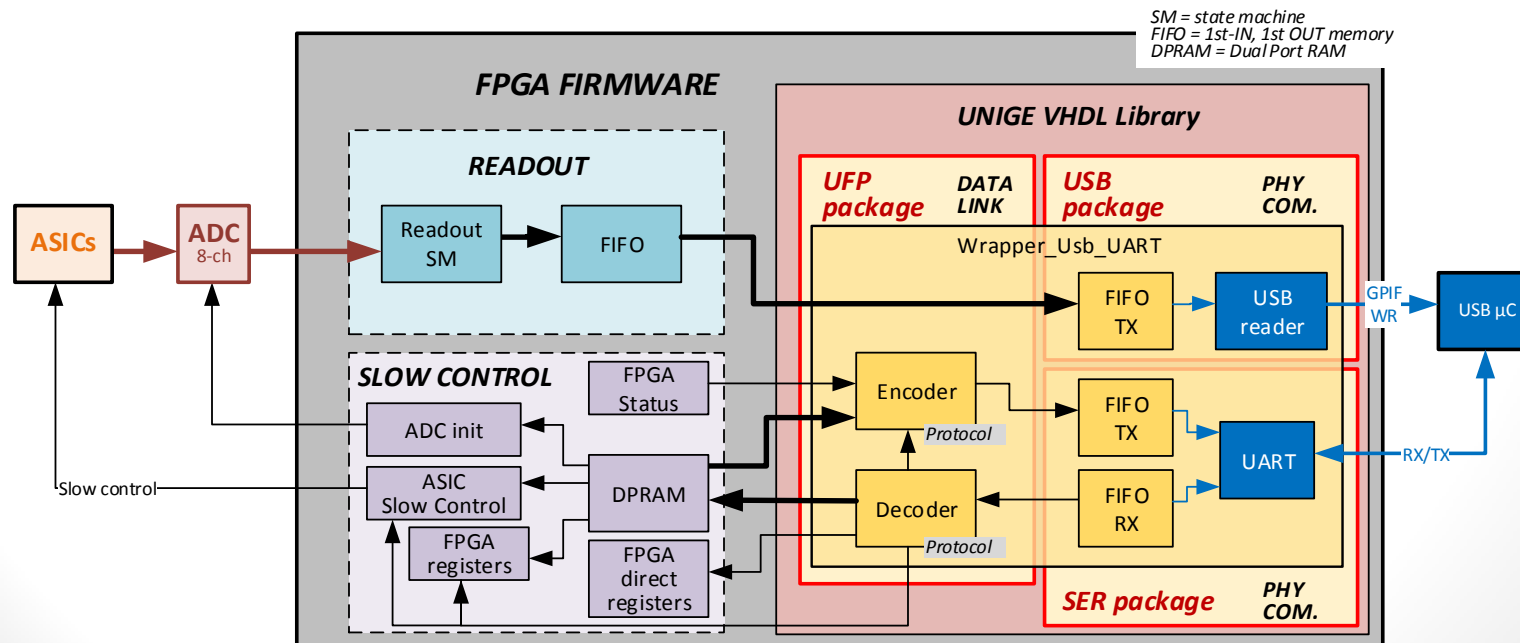
- B-MIND current DAQ is based on Unige Library and consists of:
 - VHDL application firmware code in FPGA
 - C# Host application code compiled for linux & windows
- Unige Library is based on:
 - VDHL firmware library in FPGA
 - USB3 Cypress FX3 μ C
 - Unige FX3 Firmware
 - Windows/Linux cypress USB driver for FX3
 - Unige Windows/Linux C# monodev library in host PC



Unige VHDL & UnigeFrontEnd C# Libraries overview

- Unige VHDL library is 'linked' with C# Host Library for:
 - Slow control protocol handling (UART link)
 - Status, read & write parameters...
 - Structured configuration objects, defined by a JSON file (hardware descriptor)
 - USB3 Readout (32-bit @100MHz => 3.2Gb/s theoretically, ~2.5Gb/s real)
 - Compatible with single or multiple boards in a chain

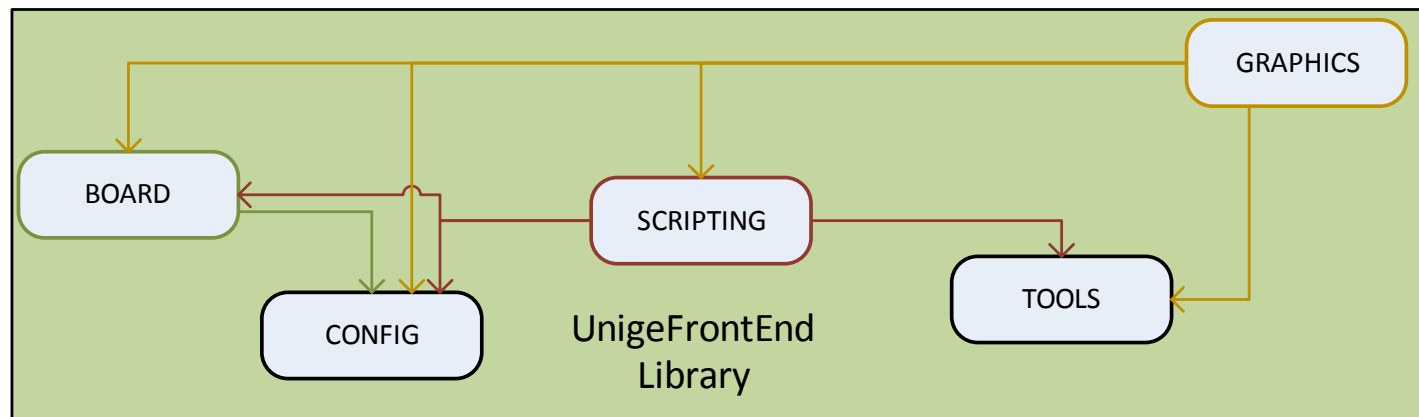
=> USB3 is transparent from both sides, providing 1 bi-directional Slow control channel and 1 fast readout FPGA to PC channel



UnigeFrontEnd C# Library architecture

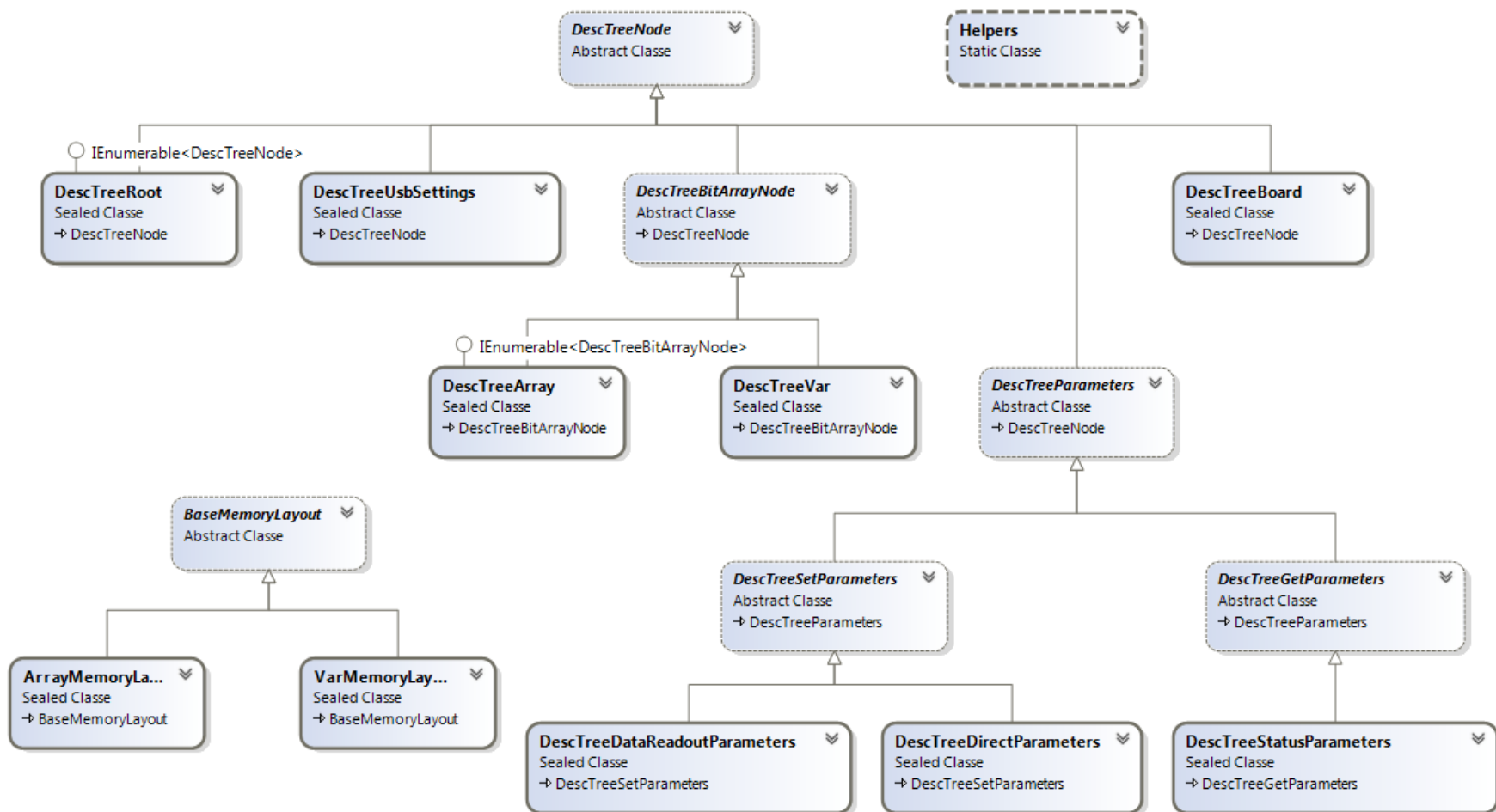
Based on C# DLLs compatible with mono-dev:

- **BOARD:** HW access for slow control & readout
- **CONFIG:** Configuration management (hardware descriptor, config. file manager)
- **GRAPHICS:** Graphical object management (linked with config.)
- **SCRIPTING:** Scripting, Socket & Command Interpreter management
- **TOOLS:** Plots/HistoPlots tools, App. settings, User Input, Utilities



UnigeFrontEnd Configuration: overview

Description of all hardware objects implemented into the FPGA from a single JSON conf. descriptor file (no code to be written)



UnigeFrontEnd Configuration: details

IEnumerable<DescTreeBitArrayNode>

Array

DescTreeArray
Sealed Class
→ DescTreeBitArrayNode

- Propriétés
 - BaseMemoryLayout
 - BitSize
 - Children
 - FootprintBitSize
 - Gui
 - GuiAutoLayout
 - MemoryLayout
 - Mode
 - NumInstances
- Méthodes
 - CheckAndClearBitArray
 - GetChildBitOffset
 - GetEnumerator
 - IEnumerator.GetEnumerator
 - OnDeserialized
 - SetRelations
- Types imbriqués

Variable

DescTreeVar
Sealed Class
→ DescTreeBitArrayNode

- Champs
 - _bitSize
 - EnumValues
- Propriétés
 - _enumValues
 - BaseMemoryLayout
 - Behavior
 - BitActiveLow
 - BitSize
 - Default
 - FootprintBitSize
 - GetMaxBitSize
 - Gui
 - Max
 - MemoryLayout
 - Min
 - ToolTip
 - Type
 - UnitName
- Méthodes
 - CheckAndClearBitArray
 - CheckDescTreeVarCoherence
 - DescTreeVar (+ 1 surcharge)
 - EnumToInt
 - GetBitOffset
 - GetType
 - IntToEnum
 - OnDeserialized
 - ParseValue
- Types imbriqués

Hardware link

ArrayMemoryLa...
Sealed Class
→ BaseMemoryLayout

- Champs
 - m_deviceIndexes
 - m_updateIndex...
- Propriétés
 - DeviceIndexes
 - GroupedByChild
 - IsDevice
 - IsUpdate
 - UpdateIndexes
- Méthodes
 - ArrayMemoryL...
 - OnDeserialized
- Types imbriqués

VarMemoryLay...
Sealed Class
→ BaseMemoryLayout

- Propriétés
 - Absolute
 - AbsoluteIndexes
 - Index
- Méthodes
 - OnDeserialized
 - VarMemoryLay...

Board

DescTreeBoard
Sealed Class
→ DescTreeNode

- Propriétés
 - Children
 - DataReadoutParameters
 - DirectParameters
 - Name
 - Parameters
 - StatusParameters
 - UsbSettings
- Méthodes
 - OnDeserialized
 - SanityCheck
 - SetRelations

All objects

DescTreeNode
Abstract Class

- Champs
 - m_versioning
- Propriétés
 - Name
 - Parent
 - Root
 - TreeName
 - Versioning
- Méthodes
 - InitializeVersionIdentifie...
 - OnDeserialized
 - SetRelations
 - ToString
- Types imbriqués

TOP Description

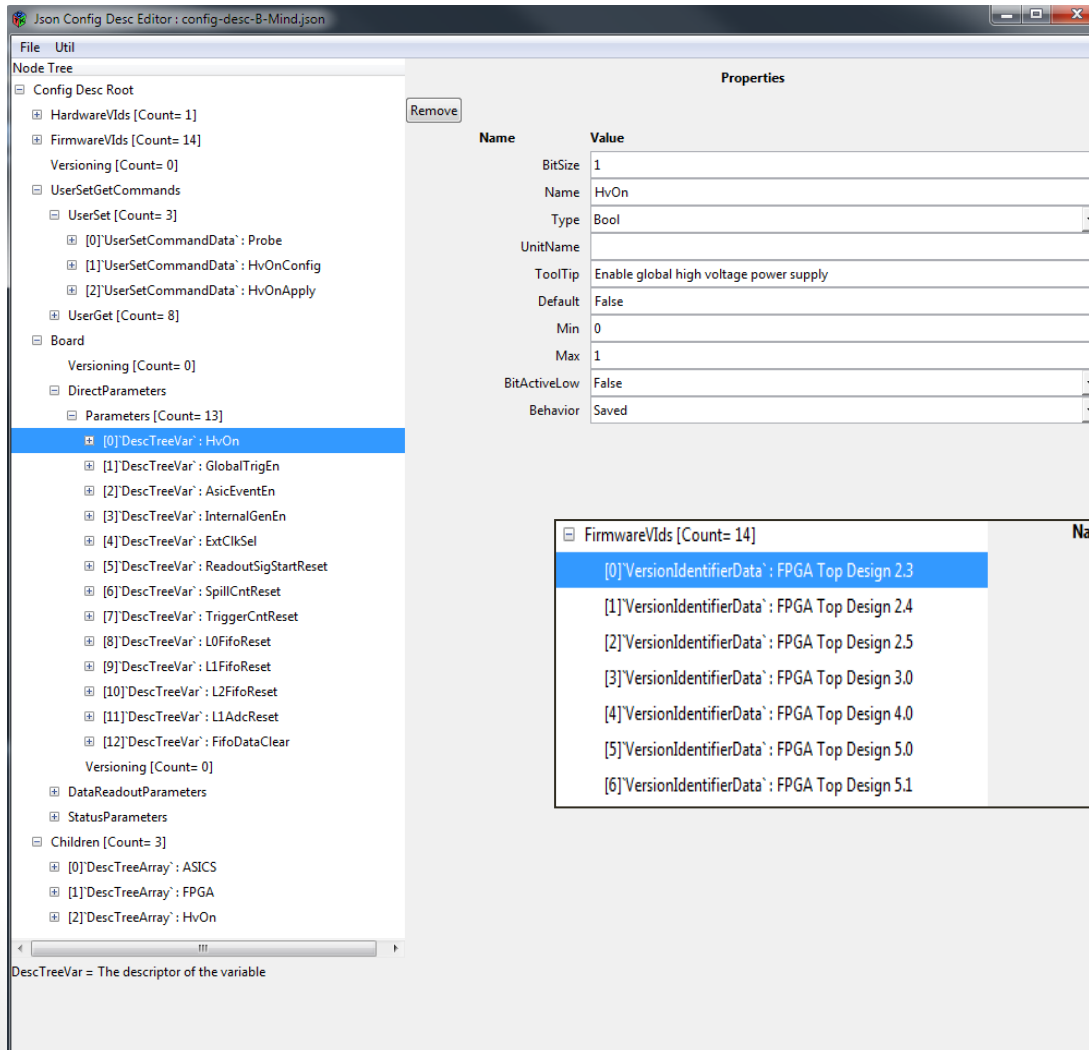
IEnumerable<DescTreeNode>

DescTreeRoot
Sealed Class
→ DescTreeNode

- Champs
 - m_firmwareVids
 - m_hardwareVids
 - m_minFpgaVersion
 - m_name
 - m_versionString
- Propriétés
 - ApplicationTitle
 - Board
 - Children
 - DeviceMemorySize
 - Devices
 - FirmwareVids
 - HardwareVids
 - MinFpgaVersion
 - Name
 - UserSetGetCommands
 - Version
- Méthodes
 - FindRootDevices
 - GetEnumerator
 - IEnumerator.GetEnumerator
 - OnDeserialized
 - SanityCheck
- Types imbriqués

UnigeFrontEnd Configuration: ConfigDescEditor

Allow to edit and modify JSON config. descriptor file with all user architecture/tree



Json Config Desc Editor : config-desc-B-Mind.json

File Util

Node Tree

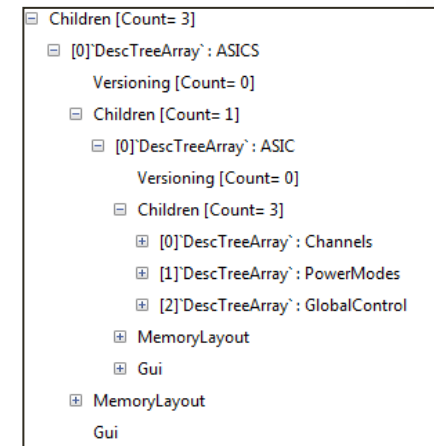
- Config Desc Root
 - HardwareVids [Count= 1]
 - FirmwareVids [Count= 14]
 - Versioning [Count= 0]
 - UserSetGetCommands
 - UserSet [Count= 3]
 - [0]'UserSetCommandData': Probe
 - [1]'UserSetCommandData': HvOnConfig
 - [2]'UserSetCommandData': HvOnApply
 - UserGet [Count= 8]
 - Board
 - Versioning [Count= 0]
 - DirectParameters
 - Parameters [Count= 13]
 - [0]'DescTreeVar': HvOn
 - [1]'DescTreeVar': GlobalTrigEn
 - [2]'DescTreeVar': AsicEventEn
 - [3]'DescTreeVar': InternalGenEn
 - [4]'DescTreeVar': ExtClkSel
 - [5]'DescTreeVar': ReadoutSigStartReset
 - [6]'DescTreeVar': SpillCntReset
 - [7]'DescTreeVar': TriggerCntReset
 - [8]'DescTreeVar': L0FifoReset
 - [9]'DescTreeVar': L1FifoReset
 - [10]'DescTreeVar': L2FifoReset
 - [11]'DescTreeVar': L1AdcReset
 - [12]'DescTreeVar': FifoDataClear
 - Versioning [Count= 0]
 - DataReadoutParameters
 - StatusParameters
 - Children [Count= 3]
 - [0]'DescTreeArray': ASICS
 - [1]'DescTreeArray': FPGA
 - [2]'DescTreeArray': HvOn

Remove

Properties

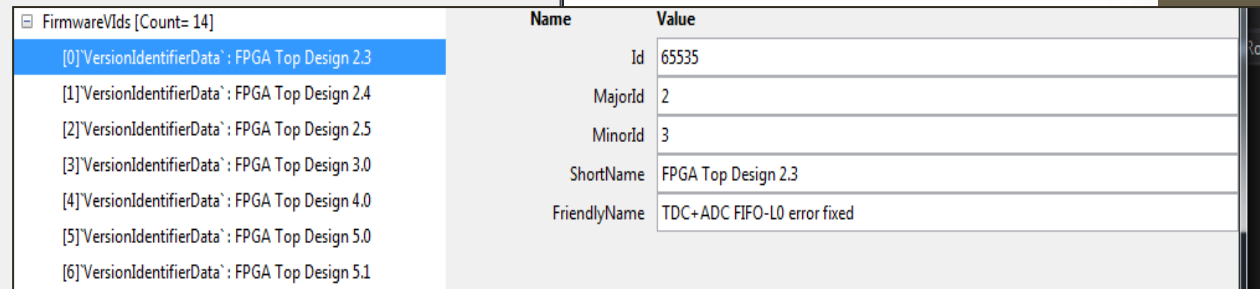
Name	Value
BitSize	1
Name	HvOn
Type	Bool
UnitName	
ToolTip	Enable global high voltage power supply
Default	False
Min	0
Max	1
BitActiveLow	False
Behavior	Saved

DescTreeVar = The descriptor of the variable



```

Children [Count= 3]
  [0]'DescTreeArray': ASICS
    Versioning [Count= 0]
  [1]'DescTreeArray': FPGA
    Versioning [Count= 0]
  [2]'DescTreeArray': HvOn
    Versioning [Count= 0]
    Children [Count= 3]
      [0]'DescTreeArray': Channels
      [1]'DescTreeArray': PowerModes
      [2]'DescTreeArray': GlobalControl
    MemoryLayout
    Gui
  MemoryLayout
  Gui
  
```



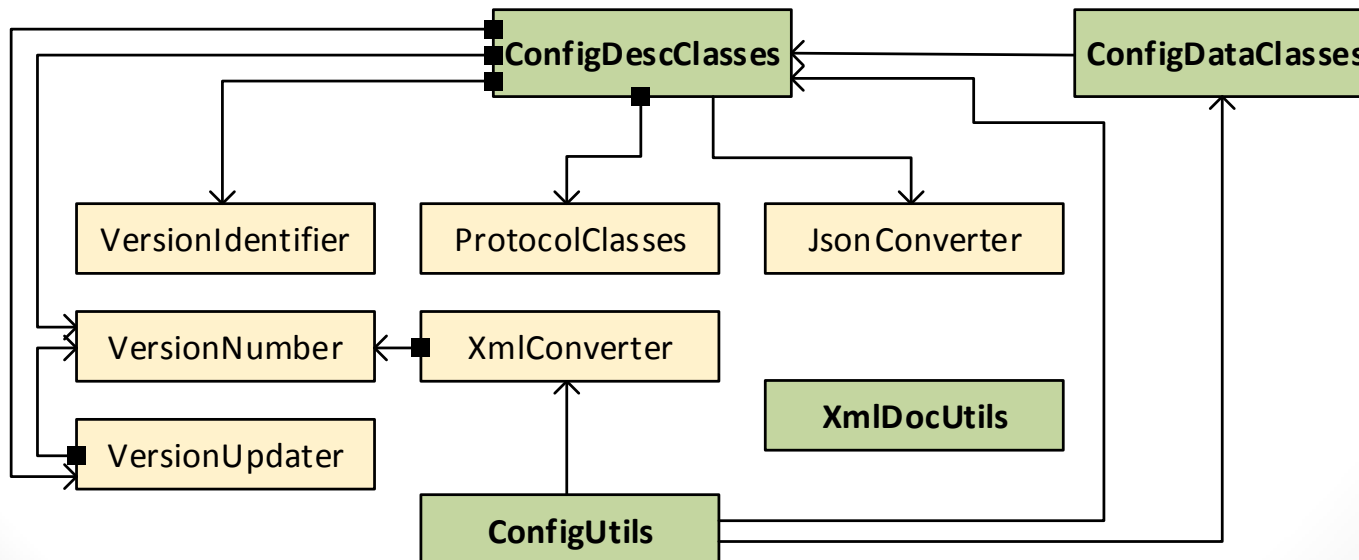
FirmwareVids [Count= 14]

Name	Value
[0]'VersionIdentifierData': FPGA Top Design 2.3	Id: 65535
[1]'VersionIdentifierData': FPGA Top Design 2.4	MajorId: 2
[2]'VersionIdentifierData': FPGA Top Design 2.5	MinorId: 3
[3]'VersionIdentifierData': FPGA Top Design 3.0	ShortName: FPGA Top Design 2.3
[4]'VersionIdentifierData': FPGA Top Design 4.0	FriendlyName: TDC+ADC FIFO-L0 error fixed
[5]'VersionIdentifierData': FPGA Top Design 5.0	
[6]'VersionIdentifierData': FPGA Top Design 5.1	

UnigeFrontEnd C# DLLs: Config

Configuration classes (hardware description & data handling):

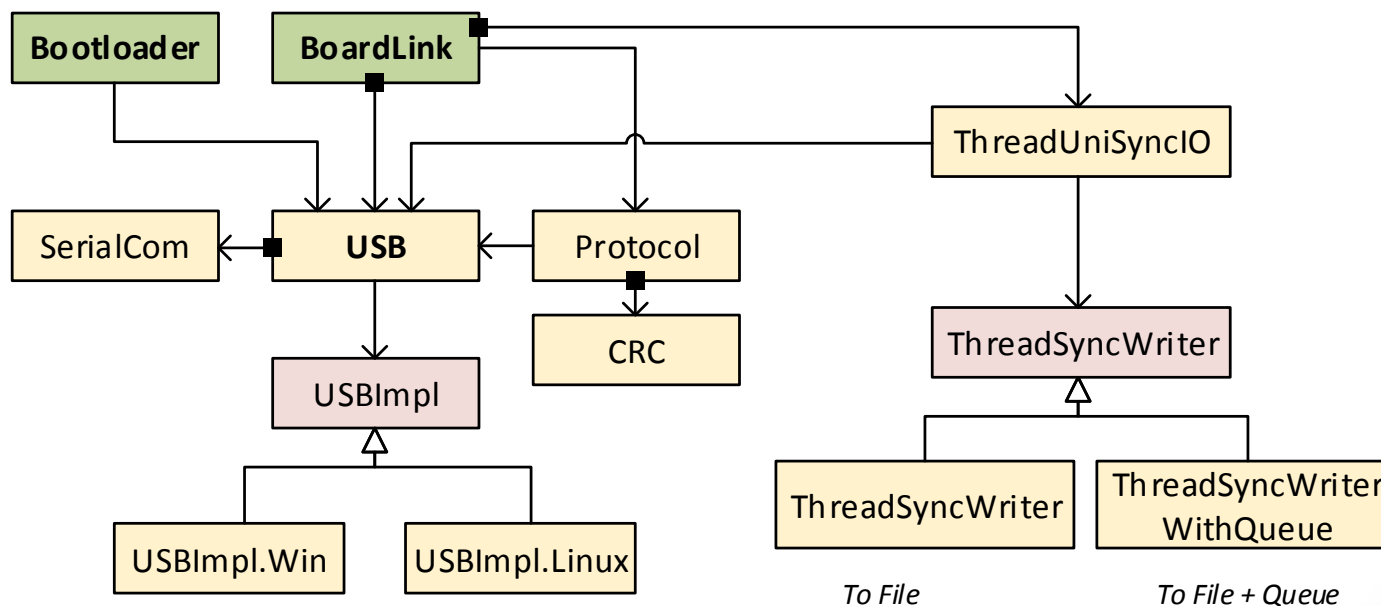
- **ConfigDescClasses**: All config. classes describing the hardware
- **ConfigDataClasses**: All config classes containing a user config. (File/Save,Open...)
- **JSONConverter** : static classes to access to Json config. description file
- **ProtocolClasses**: Classes to handle the image of FPGA objects (Devices, UserGet...)
- **VersionIdentifier**: Manage a list of objects having a version (FW, Lib...) from config. desc.
- **VersionNumber**: Manage an object having a version and utilities to compare them
- **VersionUpdater**: Manage all objects that can be upgraded in the config. desc.
- **XmlConverter**: static class to load/save config. files
- **XmlDocUtils**: static utilities to get member info. from XML document file
- **ConfigUtils**: static utilities related to config. (Help on memory layout, open/save file...)



UnigeFrontEnd C# DLLs: Board

HW access for slow control & readout:

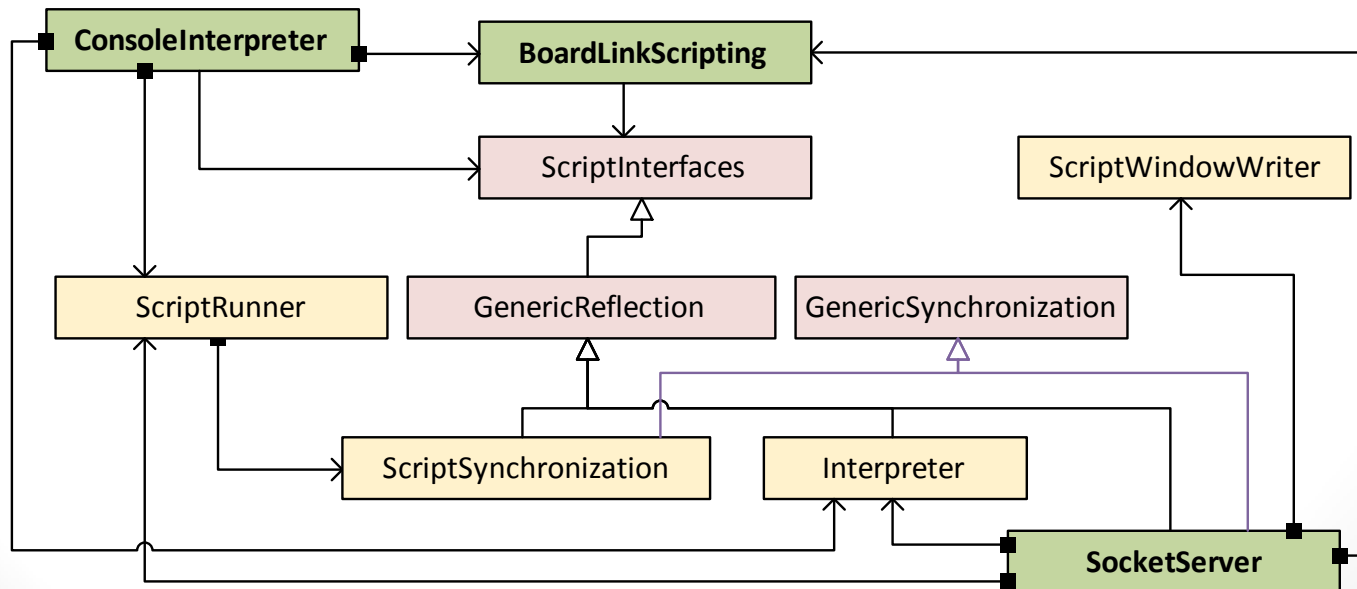
- **Boardlink:** User access to all hardware functions of the board (eq. to script Boardlink)
- USB, USBImpl, USBImpl.Linux, USBImpl.Win : low level USB classes
- CRC, Protocol, SerialCOM: low level slow control protocol handling
- ThreadUniSyncIO: Generic DAQ handling, to be connected to a 'ThreadSyncWriter'
- ThreadSyncWriter: Generic DAQ to File Writer
- ThreadSyncWriterWithQueue: Generic DAQ to File Writer with queue (monitoring)
- **Bootloader:** FX3 firmware upgrade tools



UnigeFrontEnd C# DLLs: Scripting

Scripting, interpreter & socket access:

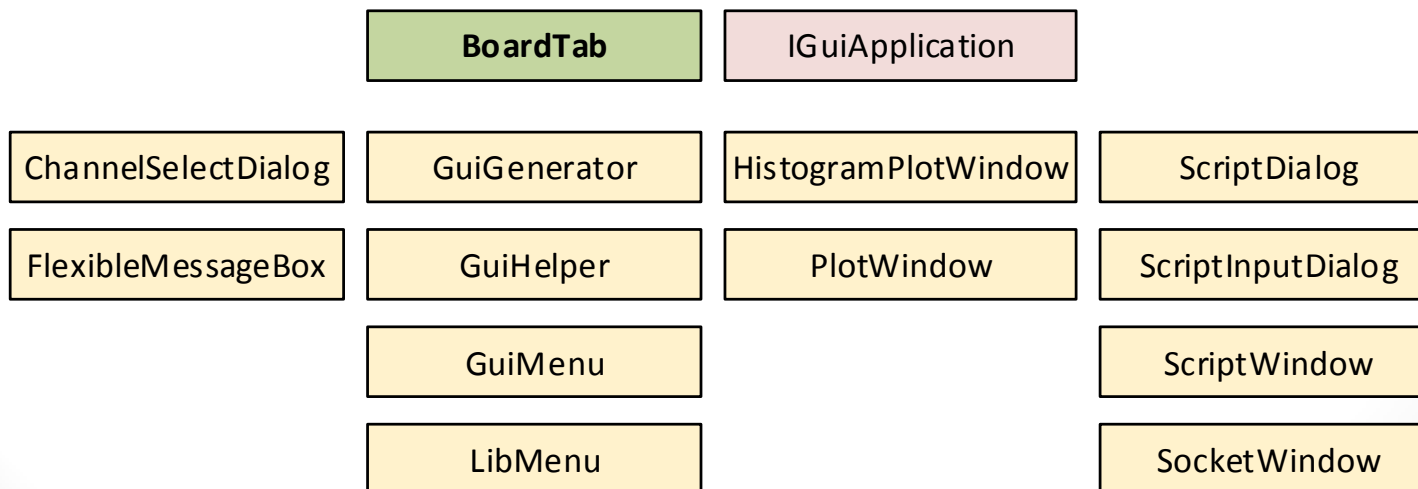
- **BoardLinkScripting**: All Board functions visible from script & socket (BoardLib help)
- **ConsoleInterpreter**: Console interpreter to be used with a console providing access to BoardLib functions & script run
- **SocketServer**: Connect a command interpreter to a TCP/IP socket
- **Interpreter**: basic functions for interpreting commands
- **GenericReflection**: A generic object in order to get dynamically methods properties (online help)
- **GenericSynchronization**: implements generic objects for script/socket synchronization
- **ScriptInterfaces** : generic interfaces for the script
- **ScriptSynchronization**: Script specific synchronization objects
- **ScriptRunner**: allows to run a script
- **ScriptWindowWriter**: allow to write messages to a console (remote socket or local window)



UnigeFrontEnd C# DLLs: Graphics

Graphical objects:

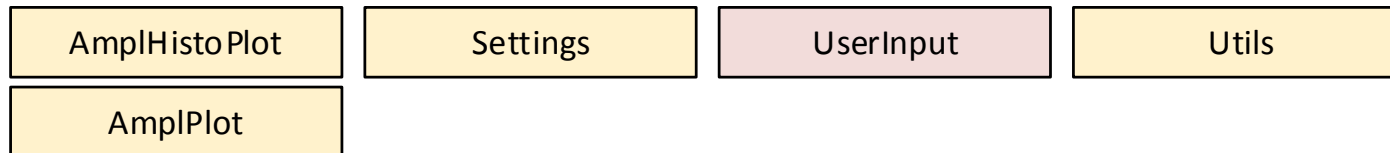
- **BoardTab: construct Application Board Tab**
- IGuiApplication: Interface that the application must implement
- ChannelSelectDialog, FlexibleMessageBox: User inputs
- GuiGenerator: Generates all objects from the config. descriptor and allows to open/save in a config file
- GuiHelper, GuiMenu: Utilities for Menu & File window handling
- LibMenu: Build Library standard menus
- HistogramPlotWindow, PlotWindow: Plot windows
- ScriptDialog, ScriptInputDialog, ScriptWindow, SocketWindow: Windows for scripting & socket
- StartWindow : Splash window



UnigeFrontEnd C# DLLs: Tools

Common/Basic utilities for all libraries:

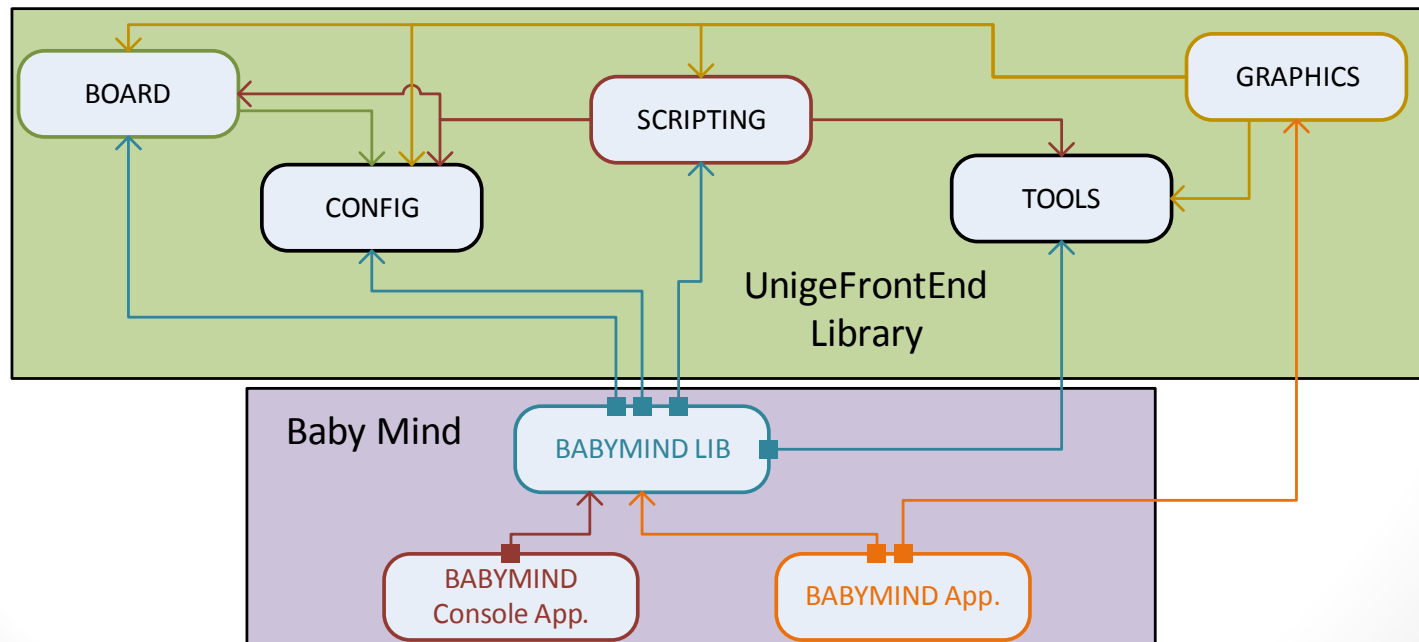
- AmplitudeHistoPlot: storage for amplitude histogram plots (used by GUI plots)
- AmplitudePlot: storage for amplitude plots (used by GUI plots)
- Settings: Handles library & application settings
- UserInput: Generic User Input (to be connected to a graphical object or cmd interpreter)
- Utils: miscellaneous fonctions



Baby-Mind: using UnigeFrontEnd C# Library

UnigeFontEnd libraries:

- **BOARD:** HW access for slow control & readout
 - **CONFIG:** Configuration management (hardware descriptor, config. file manager)
 - **GRAPHICS:** Graphical object management (linked with config.)
 - **SCRIPTING:** Scripting, Socket & Command Interpreter management
 - **TOOLS:** Plots/HistoPlots tools, App. settings, User Input, Utilities
- **B-Mind lib:** Additional B-Mind functions (e.g. TDM/FBW scripting fct, DaqTo...)



Baby-Mind: BabyMind.Lib

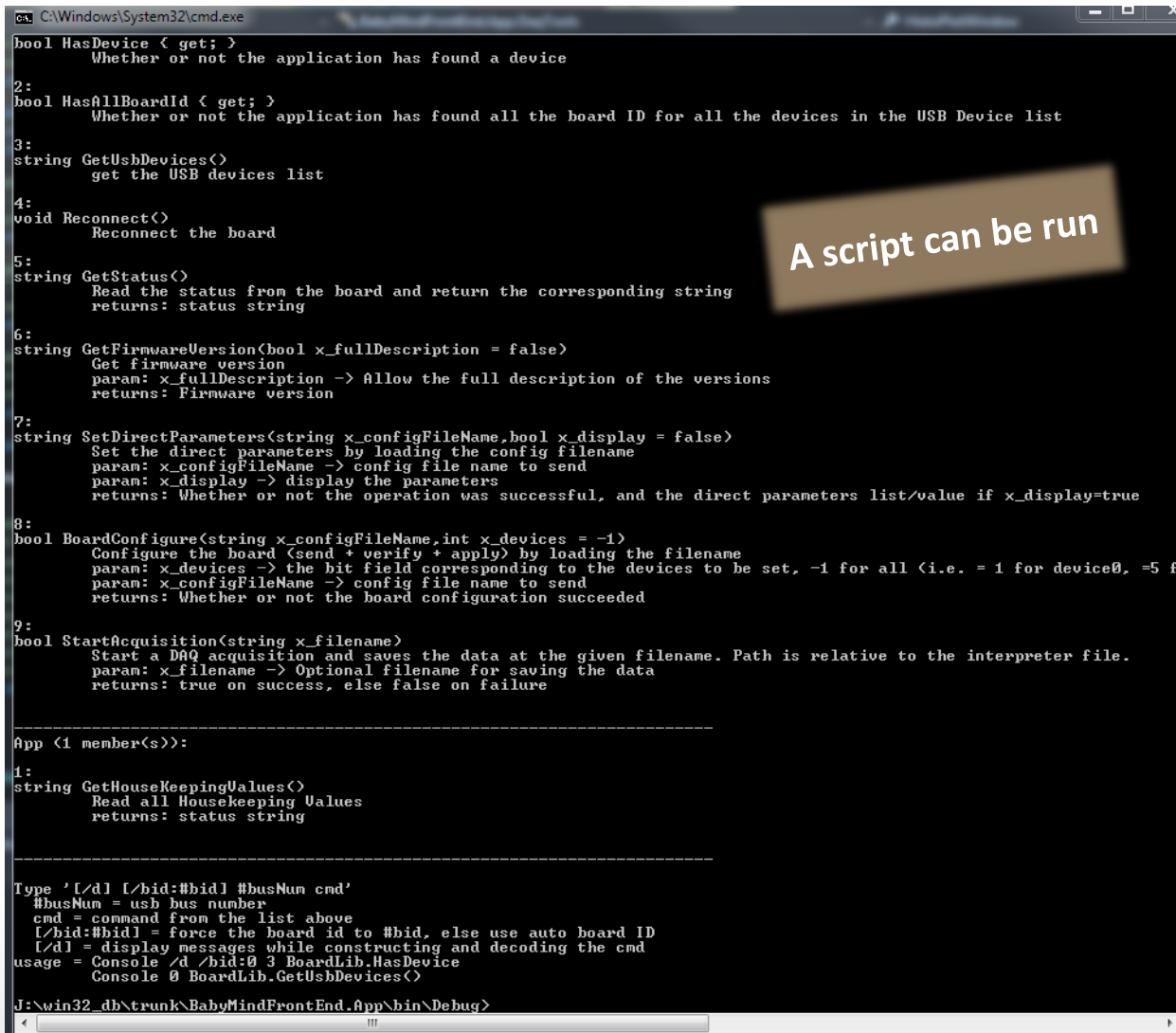
Added features used specifically for Baby-Mind:

- DaqProcess: handles all processes for DAQ conversion (ToCsv, ToHistoPlot ...)
- LibDaqTools: handles user interface and preparation before calling DaqProcess functions
- FpgaTools: debug class for FPGA automatic testing
- LibBoardLink: Additional HW features for B-MIND : FBW/TDM functions, Housekeeping...
- LibScripting: Additional scripting visible functions: FBW/TDM functions, Housekeeping...
- Protocol: Additional specific protocol functions for B-MIND (ASIC probing)



Baby-Mind: BabyMind.Console

Access to all scripting functions from a command interpreter (No user code to write)



```

C:\Windows\System32\cmd.exe
bool HasDevice < get; >
    Whether or not the application has found a device

2:
bool HasAllBoardId < get; >
    Whether or not the application has found all the board ID for all the devices in the USB Device list

3:
string GetUshDevices()
    get the USB devices list

4:
void Reconnect()
    Reconnect the board

5:
string GetStatus()
    Read the status from the board and return the corresponding string
    returns: status string

6:
string GetFirmwareVersion(bool x_fullDescription = false)
    Get firmware version
    param: x_fullDescription -> Allow the full description of the versions
    returns: Firmware version

7:
string SetDirectParameters(string x_configFileName, bool x_display = false)
    Set the direct parameters by loading the config filename
    param: x_configFileName -> config file name to send
    param: x_display -> display the parameters
    returns: Whether or not the operation was successful, and the direct parameters list/value if x_display=true

8:
bool BoardConfigure(string x_configFileName, int x_devices = -1)
    Configure the board (send + verify + apply) by loading the filename
    param: x_devices -> the bit field corresponding to the devices to be set, -1 for all (i.e. = 1 for device0, =5 f
    param: x_configFileName -> config file name to send
    returns: Whether or not the board configuration succeeded

9:
bool StartAcquisition(string x_filename)
    Start a D00 acquisition and saves the data at the given filename. Path is relative to the interpreter file.
    param: x_filename -> Optional filename for saving the data
    returns: true on success, else false on failure

-----
App <1 member(s)>:

1:
string GetHouseKeepingValues()
    Read all Housekeeping Values
    returns: status string

-----

Type '[/d] [/bid:#bid] #busNum cmd'
    #busNum = usb bus number
    cmd = command from the list above
    [/bid:#bid] = force the board id to #bid, else use auto board ID
    [/d] = display messages while constructing and decoding the cmd
usage = Console /A /bid:0 3 BoardLib.HasDevice
        Console 0 BoardLib.GetUshDevices()

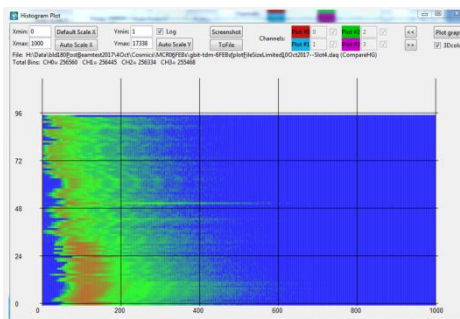
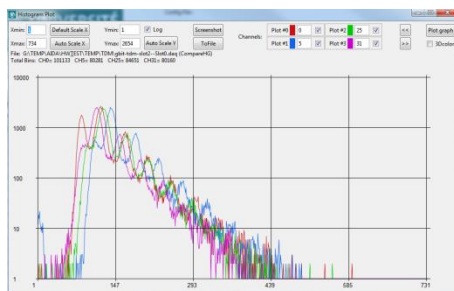
J:\win32_db\trunk\BabyMindFrontEnd.App\bin\Debug>

```

Baby-Mind: BabyMind.App

B-MIND additional GUI functions to B-Mind.Lib:

- AppBoardTab: Additional Board Tab GUI functions for B-Mind (Buttons...)
- AppMenu: Additional menu functions for B-Mind (DaqTo...)
- AppScripting: Additional scripting functions using GUI (Plots...)
- AppSettings: Additional settings for B-MIND
- DaqMonitoring: Connect the real-time monitoring plot window to DAQ thread with queue
- DaqTools: Connect Lib.DaqProcess to GUI for B-MIND (plot windows)
- HouseKeepingTab: addition housekeeping tab (T° , HR monitoring)



Application

PlotHG

Plot LG

Plot TDiff

Plot HKB

Reset Fifos before Start

Reset Counters before Start

Reset 'Readout Signal Start' before Start

HG Monitoring Plot

Board	ASIC0	ASIC1	ASIC2	FPGA	HvOn	HouseKeeping	Debug
Board ID Start:	0	Board ID Stop:		2	<input checked="" type="checkbox"/> Multi-board	<input type="button" value="Refresh"/>	
Channel	Board 0	Board 1	Board 2	Board 3	Board 4	Board 5	
ASIC0 temperature (°C)	33.9	34.8	34.8	?	?	?	
ASIC1 temperature (°C)	34.0	34.8	35.0	?	?	?	
ASIC2 temperature (°C)	33.9	34.8	34.7	?	?	?	
FPGA temperature (°C)	41.0	44.0	39.0	?	?	?	
BOARD temperature (°C)	37.2	38.6	36.6	?	?	?	
BOARD Humidity (%)	11.5	-23.8	-23.8	?	?	?	
High Voltage (V)	23.245	69.633	70.048	?	?	?	

App Help

- Data Processing
 - DaqToCsv
 - TDMTToDaq
 - GtrigSpillCheck
 - DaqToAmplHistoHG
 - DaqToAmplHistoLG
 - DaqToTimeHistoRising
 - DaqToTimeHistoFalling
 - DaqToTimeHistoDiff
 - PlotAmplHistoHG
 - PlotAmplHistoLG
 - PlotAmplHistoComputeHG
 - PlotAmplHistoComputeLG
 - PlotTimeHistoRising
 - PlotTimeHistoFalling
 - PlotTimeHistoDiff
- Version
- Save Config To Citiroc

Connected:

Get

HvOn HvOn



Thanks for you attention
Questions ...