HDMC Development (1)

Murrough Landon – 15 March 2000

Wishlist (unordered)

- Handling of bus errors; those and other errors via exceptions
- Use only STL; support only STL capable platforms (except for busserver)
- Agree how to support Module subclasses for DAQ
- Agree how to provide unified GUI for Module and its Parts
- Support for testing Parts and Part hierarchies
- Initialisation of Part hierarchy (also) from DAQ -1 OO database
- Agree how to implement convenient access to bit fields
- (Someday!) Pretty view of the configuration

Module Subclasses

- Can add Module data members for known components of specific Modules
- Can initialise these from PartManager. . .
- . . . but not sure about Part hierarchy and dependent Parts which arent truly components of a Module, eg IOFrame depending on a Module Register
- Useful methods for DAQ activities can use specific data members to initialise, download, read out specific module subclasses.

HDMC Development (2)

Module GUI

- Original idea: be very general, using PartCollection...
- ...but Module and PartCollection are different branches of the Part class model...and PartCollection now removed?
- Also needs compact Register display
- Who is going to do the work?

Bitfield Access

- Want convenient access to bitfields from C++
- One way: use a compiler to create many separate Register subclasses from the configuration files. Each bitfield becomes two functions to set and get the values, implementing appropriate masking etc.
- Alternative: Assembler type class to do it all on the fly.
- Former requires a compiler, but code would look simpler and more comprehensible. Latter requires no relinking to change behaviour. (Is this good or bad in DAQ environment?!)

Test and Verify

- Some scripting capability has been added (and documented?)
- Need something like Part::Verify() for recursive simple checks
- Scheme for running test vectors with sets of inputs and known outputs