Meeting Minutes

Craig C. Drennan

Principle Engineer

AD / Instrumentation

P.O. Box 500, MS 308

Kirk Road and Pine Street  
Batavia, Illinois 60510-50­11 USA

Office: 630.840.2160

[cdrennan@fnal.gov](mailto:cdrennan@fnal.gov)

**Date:** May 8, 2020

**Re:** Meeting Minutes, MTCA.4 Reference Design 3rd Meeting

**Meeting Time: 9:00 am to 10:00 am**

**Meeting Location: ZOOM Meeting**

**Attendees:**

Craig Drennan AD/Instrumentation (Secretary)

Duane Voy AD/Instrumentation

John Diamond AD/Instrumentation

Robert Santucci AD/Instrumentation

Peter Prieto AD/Instrumentation

Dennis Nicklaus AD/Controls

Ryan Rivera SCD/DAQ & Electronics

Andres Quintero-Parra SCD/DAQ & Electronics

**Request for Feedback:**

Please send any additions, corrections, rephrasing and/or comments to Craig Drennan. There has been liberal use of paraphrasing in recalling what others have said in the meeting. There may also be some editorializing and introduction of new thoughts that could use some review. Thank You.

**Minutes:**

## Introduction

This was the third meeting to discuss the MTCA.4 demonstration and reference design project. We are hoping to receive the Vadatech crate, modules, and other components in June.

Krzysztof Szczurek was expected to come to work for the Scientific Computing Division and work on this project, but the COVID-19 pandemic has complicated his VISA approval process. Krzysztof has decided to not take the job at Fermilab and stay in Europe and work towards a PhD. Andres Quintero-Parra is a new hire into SCD and will take Krzysztof’s place in the project.

## Background

As review, the project is to demonstrate the use of MTCA.4 equipment and provide a reference design for the firmware and software. The equipment has been ordered from Vadatech, Inc. (PO 665382) and is expected to arrive in June 2020. Vadatech supplies firmware IP and other documentation through a private website portal that Fermilab is given access. We can freely use the firmware IP from Vadatech on Vadatech equipment.

The FPGA on the Vadatech AMC523 module is a Xilinx Kintex-7 (XC7K410T-2FFG900). SCD has experience with both the Xilinx FPGAs and the uTCA systems. SCD has been asked to work with the front-end software support programmers in AD to produce this demonstration and reference design in the short amount of time between receiving the Vadatech equipment sometime in June and the end of September 2020.

The goals are to develop firmware for reading the analog to digital converters, perform memory management, and implement the DDCP ethernet protocol being used by AD/Instrumentation to communicate the data to a host computer external to the MTCA.4 crate. Software will be developed to interface with the AMC523 module and the MTCA.4 crate, MCH control hub. There are several MCH functions that are necessary to manage power to the AMC523 module and other features of the crate.

SCD has already developed a similar UDP ethernet protocol and there have been discussions as to which features of this protocol might be implemented as additions or revisions to the DDCP protocol. In the 2nd MTCA.4 Meeting held in April 2020, Ryan Rivera presented slides outlining and comparing DDCP protocol as currently written to the OEI protocol that they have developed (Beams-doc-8290).

One important feature being considered is the ability to remotely set the IP address of a MTCA.4 module. Up to this point, trial implementations of the DDCP protocol has been done with AD/Instrumentation VME digitizers that set their IP with DIP switches on the boards. Mark Austin from AD/Controls also sets IP addresses on the MFTU, Multi-Function Timing Unit with DIP/Hex switches. However, the MTCA module we get from Vadatech or other vendors may not have such switches and we will need to find another method of setting the IP addresses. We will be looking at the functions of the MCH crate controller for methods to use with the MTCA modules.

## Updating and Editing the DDCP Protocol Specification Documents

Aside from what features may or may not be added to the DDCP protocol, there was a discussion of how the DDCP protocol specification documents could be reviewed, updated, and edited as a group. It was agreed to send the main document out as a shared file on Microsoft OneDrive. The final document versions and the supporting documents would be stored in the Accelerator Division Beam Docs database.

There are documents and code used by the software programmers. Doxygen is a tool that is used for generating user documentation using tags and fields embedded in the software code files. It explains how to use the DDCP API library that Duane Voy developed for client nodes. The client node in this case is the host computers that collect data from the digitizers (servers) and manages the clock and timing units.

The main DDCP protocol specification document needs to describe the protocol and the reader will determine how the protocol is implemented, whether in the software of a processor or the firmware, RTL of an FPGA, or a combination of the two. In determining what goes into the protocol we want to keep in mind that a designer may want to use only RTL or a small processor and try to keep the complexity of the protocol lower. We have SCD’s OEI protocol as an example of an effective protocol that is implemented completely in VHDL/RTL. At this time, the digitizer (server) side of the DDCP protocol is mostly RTL which uses a small, NIOS soft processor.

## Erlang Version of a DDCP Client

Dennis Nicklaus has been writing DDCP client using Erlang code. This code is considered mostly complete, but only supports a subset of the functions. Dennis felt that a client only needs to support the features it wants to handle and does not need to support everything the servers have to offer.

In an email that Dennis had sent the group prior to the meeting he noted that the current DDCP protocol specification document says that “All incoming requests must ultimately be acknowledged with a reply of the correct format”, but the document currently does not define the expected response for each message (Operation) type.

Duane Voy said that he does have a spreadsheet of the rules and responses and says that this information should be in the protocol document. Duane had called this spreadsheet the DDCP Rulebook. Dennis said that it would probably be enough to add comments to the existing spreadsheet and include this with the protocol document.

## Mark Austin and the Multi-Function Timing Unit (MFTU)

Mark is interested in implementing the DDCP protocol in the Multi-Function Timing Unit (MFTU). The MFTU is going to be an essential device on our private DAQ network, providing TCLK, LCLK, Beam sync, triggers for the digitizers, etc. Mark is currently planning to use the Arria 10 FPGA in this unit. This Arria 10 does have the ARM processor in it which could benefit from the support given to the Low-Level RF efforts with the Arria 10. Also, Ryan spoke to Mark to say that they should stay in touch with one another in developing the IP for the DDCP protocol. Ryan and Andres’ approach will be to code all of the DDCP logic in VHDL/RTL and not rely on additional software in a hard or soft processor.

Dennis and Duane had asked Mark several months ago to consider using the DDCP protocol in place of the CEC protocol that he had started with. Mark said that his implementation of the CEC protocol used a NIOS processor in a Cyclone 5 Intel FPGA.

## Review of Currently Proposed Edits to the DDCP Document

Discussion of the DDCP specification included some edits provided by Ryan Rivera, some discussion in the meeting and several email exchanges after the meeting.

Mark and Ryan discussed how IP addresses and even MAC addresses could be changed. It was mentioned that changing the MAC addresses would not be allowed on the lab network but would be allowable as long as the network remained private and isolated. Ryan said that computing had, for the test beam DAQ systems, given them a set of MAC addresses that they could use. In this case the network people could track down a node that may have accidentally gotten on to the lab network.

Ryan also explained that they became familiar with the ARP protocol and supported ARP requests with their OEI, otsdaq network. Some of us need to become more familiar with this. Ryan recommended Wiki articles on the ARP protocol.

Ryan also emphasized the differences between a completely VHDL/RTL implementation and one that uses a hard or soft processor. There are complex things like DHCP that fall into the too complex category but working at a low level allows more flexibility with changing IP and MAC addresses.

There was also more discussion on whether we would want to be able to setup the network servers (digitizers) to exchange data and control packets with more than one destination. Folks could imagine some use for this, but it is not necessary for this demonstration and reference design project. If this function is implemented in either or both the client and server, it need not be employed in all applications.

It was expressed that having a private DAQ network with one host front-end computer that sends control functions to and receives data from multiple servers (digitizers, timing units) is enough for now. We also include the ability for the servers to send unsolicited Interrupts to the host front-end computer.

Additionally, in order to finish the MTCA.4 demonstration by the end of September, fixed IP and MAC addresses could be used. Configuration management functions to set IP and MAC addresses and other features are eventually needed but can be neglected for the demonstration. Finishing the demonstration is seen as a necessary step in establishing the DAQ infrastructure for PIP-II and many other machines in the Accelerator Division. Funding and manpower decisions are waiting on this.

## Post Meeting Email Exchange Summary

In email following the meeting John Diamond proposed that our network be defined with the following documents.

1. **The DDCP Protocol Description:**

This is the top-level protocol requirements, specifications and operating modes. This top-level, what we call DDCP, is an application layer protocol that makes no assumptions about the underlying communications mechanism

1. **Common Field Hardware Description:** (*maybe Configuration Management*)

This document would prescribe methods particular to the FPGA modules and other devices that have a limited operating system or none at all. Here we can describe the methods of configuration management and assignment of IP and MAC addresses. A possible method could be using the functions in the MTCA crate to probe and interrogate the modules in each slot and assign IP addresses according to slot or other information in a boards non-volatile memory such as board type or model number

1. **DDCP API Description:**

This document is that generated from the source code comments with Doxygen. This document describes the DDCP library and only concerns the software client-side API.

Duane Voy and others recommended placing limits on the scope of the DAQ network we are proposing. Duane stated that originally Nathan Eddy, John Diamond, Alexei Semenov and he set a goal to produce a proof of principal for getting large amounts of data from a 'server' into a 'client' efficiently, reliably and with high throughput. The solution should be capable of reading and setting general server data as well. The project assumed a star network topology on an ethernet network using the UDP protocol. Essentially, we wanted to provide feature parity with the VME data and address busses but extend it outside the physical crate -- essentially a fieldbus.

One extension to the current star network topology is to make provisions for the servers (digitizers, timing units) to send data to more than a single destination. This is something that was setup by SCD’s OEI/otsdaq network. Dennis Nicklaus commented that he has also thought about the "use DDCP to tell the server to start sending data somewhere else" issue. There is no reason we cannot use DDCP settings to one Feature as a way to enable/disable other actions of the server h/w. Basically, the client sends the DDCP command to enable streaming to {IP x, Port n}. The server replies to the client like it would for any other DDCP message, but has, through callbacks or whatever is connected to that Feature, also kicked off some additional computing task (streaming to a different client, with a possible completely different protocol) in this case. So, the desire for other actions like this does not mean that we have to necessarily change DDCP as a protocol definition.

John Diamonds comment on this was, “I think this is another example of a consideration that could use DDCP but should be outside of the scope of the protocol and should go into the Common Field Hardware Description.

To the topic of setting IP and MAC addresses Craig Drennan noted a few things.

1. The uTCA MCH crate control hub likely has slot control features besides turning the power on. We should look in the MCH documents to see what may be available.
2. I really think that we will have non-uTCA nodes like the Timing module from Controls. DIP switches are probably OK to use in these situations and we should include the provision in one of our Network specifications. Note I do not necessarily mean the DDCP protocol document.
3. Other non-UTCA nodes that do not have DIP switches, perhaps, should/can be avoided.
4. For the demonstration we could skip addressing the configuration management and the dynamic setting of IP and MAC addresses and hard code these in the one or two modules use for the demo. That said, if we have time to put in something fancy we should.

If you remember something you found important that we should include in these minutes, let me know and I will add it.