Drive Electronics Manual

ACTIVE OPTICAL SYSTEMS
2021 Girard Blvd. Suite 150
Albuquerque, NM 87106
(505) 245-9970 x184
www.aos-llc.com

Version 1.4 – 10/2/2018

Table of Contents

1 Quick-Start Guide

1.1 Warnings

HIGH VOLTAGE: The AOS Drive Electronics produces high voltage! High voltage can cause injury or death. Please use the AOS Drive Electronics only with an AOS deformable mirror properly attached to minimize the risk of injury.

USING HYPERTERM OR PRINTF: The Microsoft Windows Hyperterminal program can be used to interface to the AOS Drive Electronics, but it cannot send the character 0x00, even using the Alt+000 on the numeric keypad. The same is true of the printf() functions in C since the string terminator is the null character.

FUSE: This device contains a fuse to prevent damage to the internal components. If the device is not responding, check the fuse.

1.2 Quick-Start Instructions – USB Drive Electronics

Details for each of the following steps are shown in the following document.

1. Install the AOS software.
2. Attach the Drive Electronics to the deformable mirror, the USB port of the control computer, and then to the power.
3. Turn on the Drive Electronics. The computer should recognize the drive electronics upon power-up. Follow the driver installation instructions.
4. Run the DM Controller software.
    a. Make sure that the software has properly connected to the drive electronics.
    b. Load the *.DM file associated with your deformable mirror.
5. At this point, the deformable mirror should be responding to computer commands.

2 Introduction

This manual describes the operation of AOS Drive Electronics systems. The manual is divided into a section on each type of drive electronics. Each section describes the general theory of operation, driver installation, and any external interfaces. The DM Controller software is described in the AOS Software Manual.

2.1 External Software Interface

The DMController software has an external interface that can be accessed via C or C# in Windows. This software interface works for all of our drive electronics. It is documented in more detail in the software manual. This is the best way to interface to our electronics.

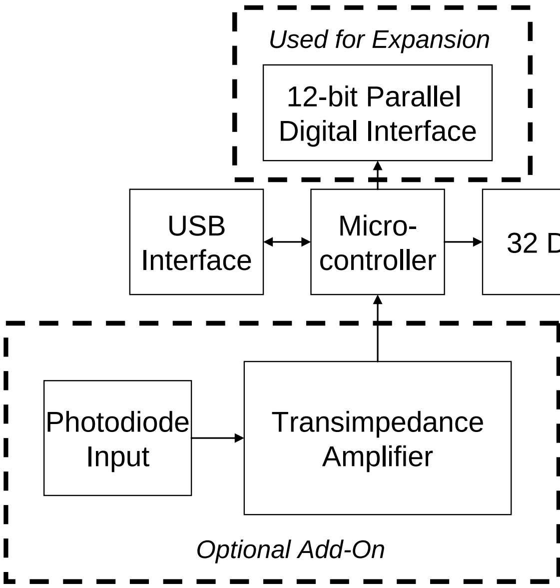
3 USB Drive Electronics

This section of the manual covers model number DE1-32-1. It begins with a brief introduction to the Drive Electronics including a summary of device operation. The introduction concludes with some important warnings about the usage of the AOS Drive Electronics. Then a description of the software installation is provided. Finally, a description of the DMController software and its features are included. This manual should be read in its entirety before using the AOS Drive Electronics.

Figure 1 - USB Drive Electronics without external photodiode interface.

3.1 USB Drive Electronics Theory of Device Operation

The AOS USB Drive Electronics (DE) is designed to be used to drive the AOS membrane deformable mirrors (MDMs). An operational schematic is shown below. The USB input from the computer sends a command to the firmware inside the DE. Some



Schematic View of the AOS USB Drive Electronics

commands are documented below. This command is then interpreted by the on-board microcontroller and sent to the 32 channels of low-voltage DACs on the board. The low-voltage signals are then amplified by a factor of 72 up to a peak of 295V. There is a 12-bit parallel digital interface in the drive electronics that is used for expanding the capability of the single board. For example, additional drive electronics boards can be commanded using the digital interface so that a user can command additional drive electronics boards to get more channels. The drive electronics board is also optionally equipped with a photodiode input to enable the board obtain feedback and do metric based control. The USB electronics can run up to 800 Hz update rate.

The AOS Drive Electronics firmware is setup for an automatic command reset if a command has not been completed in 1 second. This was setup to avoid the firmware locking-up if a command is interrupted. This feature can be toggled on and off using the "T" command.

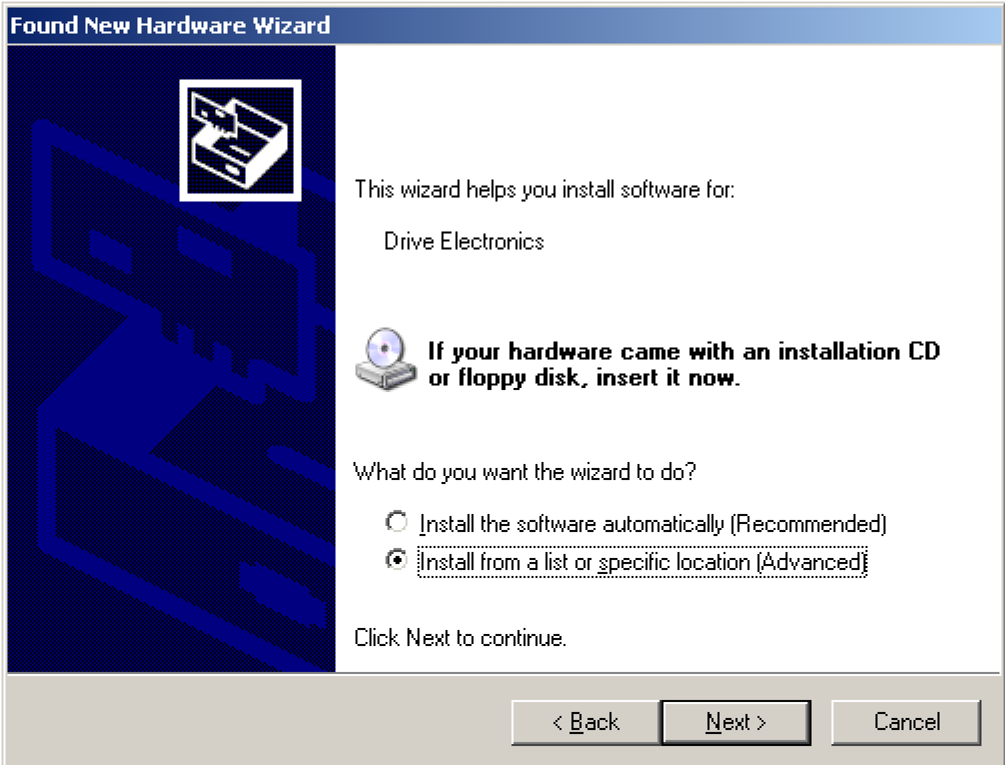
3.2 USB Drive Electronics Driver Installation

- To install the software provided by AOS, insert the provided disk and run the installer.
- Plug in the Drive Electronics to the wall socket (120 VAC).
- Connect the USB port of the drive electronics to the computer using the cable provided.

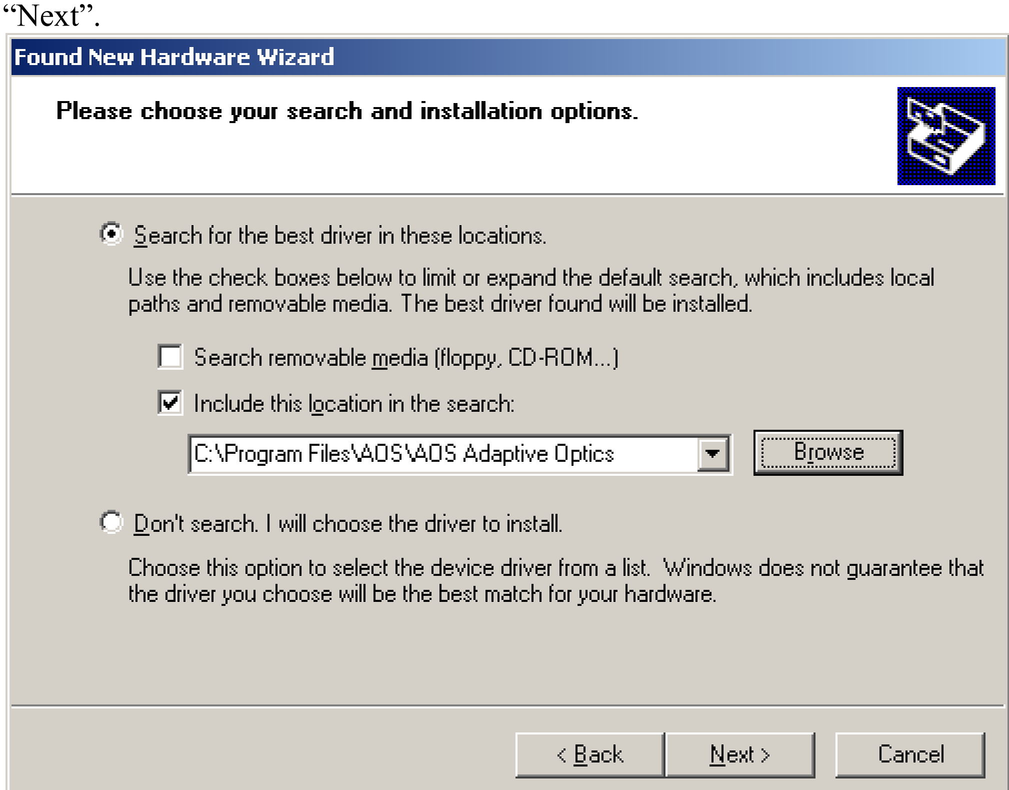
- Turn on the Drive Electronics.
- The following message box should appear off of the task bar:

- Then the following dialog box may appear. Choose “No, not this time”.

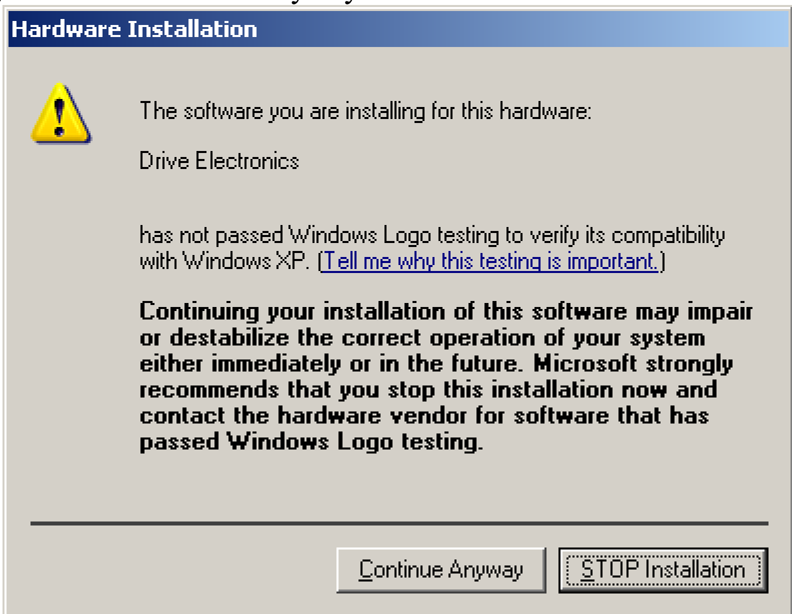
- Then the following dialog box will appear. Choose to install from a specific location.



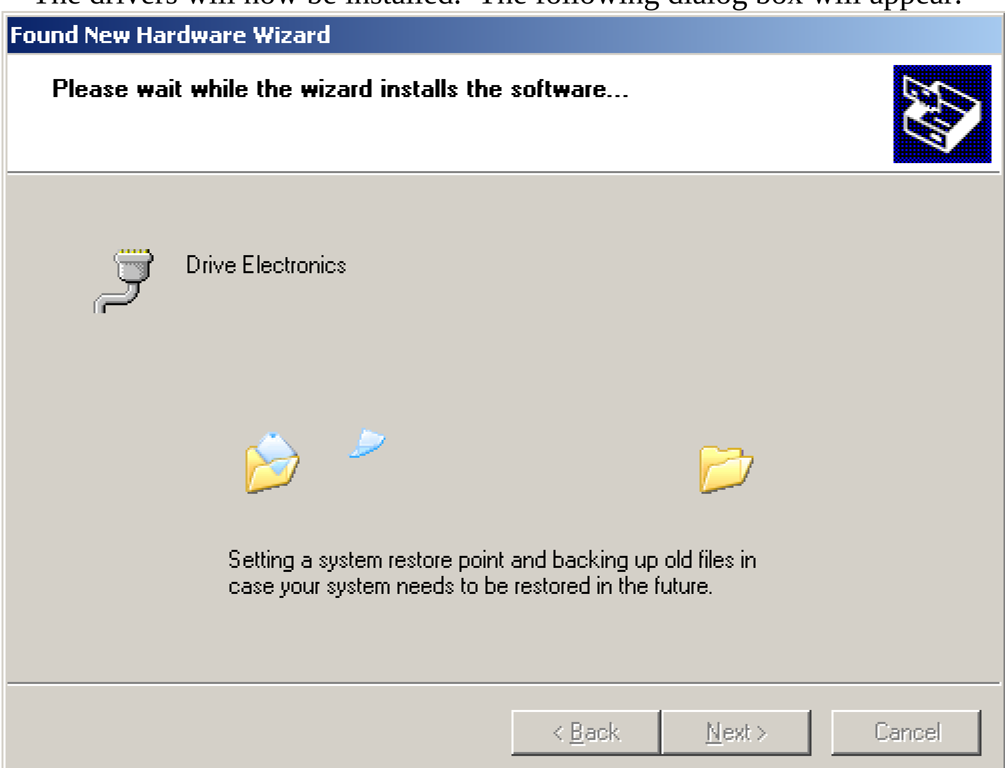
- When the following dialog box appears, do not search removable media. Instead, specify the location "C:\Program Files\AOS\AOS Adaptive Optics" and select "Next".



- Windows will then warn you that this driver has not passed Windows Logo testing. Select "Continue Anyway".

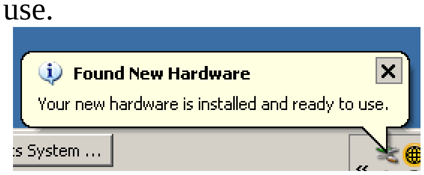


- The drivers will now be installed. The following dialog box will appear.



- When the installation is complete, the following dialog box will appear. Select "Finish".

- A pop-up from the task bar (shown below) will appear to inform you that the hardware is ready to use.



3.3 USB Virtual COM Port Command Interface

The following commands are supported by the AOS Drive Electronics over the Virtual COM Port (VCP) interface:

3.3.1 Command Format

Commands are sent to the AOS Drive Electronics as a command byte followed by parameters if necessary. Some of the commands have return values that can be ignored or used for diagnostics. The table below describes all the commands implemented in the AOS Drive Electronics firmware. Commands should not be sent to the Drive Electronics that are not on the table below.

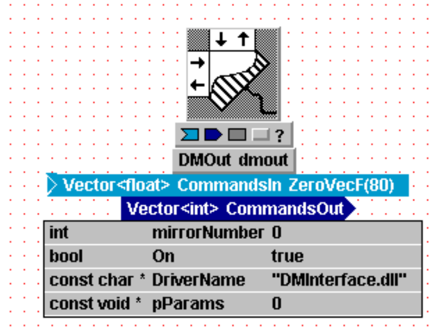
Commands

Description	Command	Parameters	Return Value
Set all channels to the same	A	Byte 1: DAC level	None

value.			
Zero all channels	R	None	None
Zero a single channel	Z	Byte 1: Channel	None
Set a single channel to a value	S	Byte 1: Channel Byte 2: DAC level	None
Multiple channel write	M	Byte 1: Number of channels Following Bytes: 8-bit values	None
Turn the command timer off	T	None	"TIMER ON" if the timer has been turned on. "TIMER OFF" if the timer has been turned off. If a command is interrupted the timer will return "RESET" after ~1 second.
Request the type of device and version	I	None	"DE1.1\r\n" if the device is the drive electronics with version 1.1 of the firmware. The \r\n is a carriage return and a line feed.
Request the value of the photodiode input (if installed)	P	None	Text value of the measured photodiode input.
Optimization of the channels based on photodiode input (if installed)	O	Byte 1: Type of optimization Byte 2: Number of milliseconds between sending the command and sampling the photodiode. Bytes 3 and up: To be determined in later versions.	None *NOTE: Optimization is still under development and should not be attempted at this time.*
Enter into firmware loading mode.	B	None	*NOTE: This is an advanced feature that should not be used at this time. Using this may cause the device to stop working until a double reboot is executed.*

3.3.2 USB WaveTrain Component

The figure to the right shows the DM control component in WaveTrain. The tempus system description (TSD) for this component is shown below. To operate this component in WaveTrain, a special interface DLL is required. This DLL can be obtained from MZA Associates Corporation (www.mza.com).



DMOut Component in WaveTrain

3.3.2.1 Tempus System Description (.TSD) File Contents

```
NAME = DMOut    [Interface: 15 Jun 2005 16:43:02 GMT]
[Implementation: 15 Jun 2005 16:43:02 GMT]   (*)

DESCRIPTION
Writes an input vector of commands out to a physical DM via a driver
DLL. The driver DLL may perform some limiting of the commands; the
output vector will contain the commands actually output.

C++ CODE
PARAMETERS
      int          mirrorNumber = 0 // Deformable mirror selection
      bool     On = true         // If false, DLL is not even loaded
      const char * DriverName = "HVDD.dll" // name of driver dll (see
dmdriver.h for spec)
      const void * pParams = 0 // optional parameter block for driver
init data

INPUTS
   Vector<float>  CommandsIn = ZeroVecF(80) // Vector of DM Commands

OUTPUTS
   Vector<int>  CommandsOut =  // Commands that were output (with
limiting)

SUBSYSTEMS

CONNECTIONS

PROPERTIES
#Sat Dec 15 16:45:09 MST 2007
CommandsIn.position=(46, 71)
CommandsOut.position=(49, 100)
Version=2007A\n
```

3.3.3 USB Matlab Example

Matlab has a serial port interface that can be used to command the AOS Drive Electronics using the commands outlined in Section **Error! Reference source not found.** above. Below is an example script on how to search for the drive electronics by opening a range of COM ports and looking for the response from the "I" command. Please remember that the null (0x00) character cannot be sent with fprintf(…). Use fwrite(…) instead.

```
function [] = TestDM
% TestDM
% Demonstrate communication with DM via Matlab serial port interface
  ports = instrfind; % close all ports
  fclose(ports);
  for ii = 1:length(ports) % try to find DM on some port
      try
          port = ports(ii);
          set(port,'BAUD', 9600);
          % could also do port = serial('COM12', 'Baud', 9600);
          fopen(port); % open the port
          fprintf(port, 'I'); % send query character
          idn = fscanf(port); % read response
          if (strncmp(idn,'DE',2)) % response should be 'DE1.1' or similar
              disp(['Found DM on port ' get(port, 'Port')]); % report success
              break;
          end
      catch
          %disp(['No DM on port ' port]);
      end
      fclose(port); % close port and continue
  end
  fclose(port); % ensure port is closed on exit
return
```

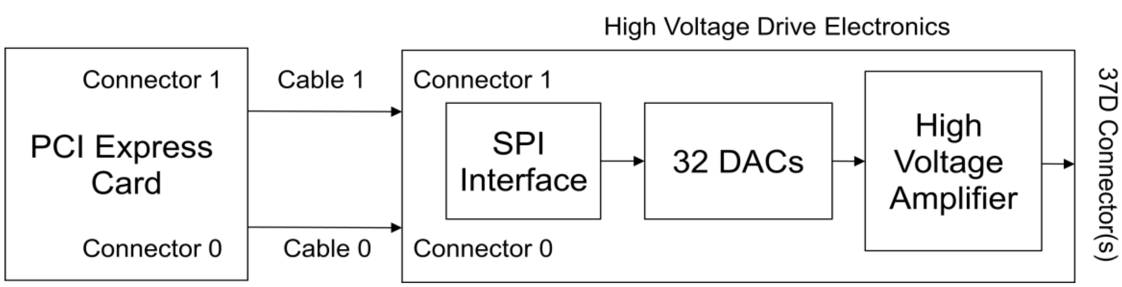
4 SPI / NI Drive Electronics

4.1 SPI Drive Electronics Operation and Setup

The SPI drive electronics interface uses a National Instruments card to create a serial peripheral interface (SPI) to the DAC chips inside the drive electronics package. This interface to the drive electronics was designed to have an update rate of ~4 kHz. We provide this electronics with a PCI Express 6259 National Instruments card for addressing the electronics.

Figure 2 – High-Power SPI Drive Electronics with Optional Analog Output and Triggering BNC Connections

Below is a schematic of the AOS SPI Drive Electronics and the hardware setup. We recommend installing the National Instruments software before installing the NI card. Once the PCI express card is installed in the computer, cables need to be run between the card and the AOS SPI Drive Electronics. The card has two connectors labeled with a 1 and a 0. These connectors need to be attached to the corresponding connectors on the AOS SPI Drive Electronics. Once all the connections have been made and the AOS and NI software is installed, then power can be applied to the AOS SPI Drive Electronics.



SPI Drive Electronics Interface Schematic and Hardware Setup

4.2 SPI Drive Electronics Software Installation

Here are the steps for installing the necessary software and drivers for the AOS SPI Drive Electronics:

- Install the NI Software Interface (NI-DAQmx).
- Shutdown the computer and install the NI PCI Express card.
- Connect the cables from the NI card to the AOS SPI Drive Electronics including the power cable, but leave the power switch off.
- Power-up the computer and complete the driver software installation.
- Install the AOS Software.
- Turn on the AOS SPI Drive Electronics
- Run the AOS Software or any other software interface.

4.3 Optional Additional Interfaces & Scalability

With the NI card interface, we have the ability to add some additional functionality to the user. We can customize our SPI Drive Electronics boxes to include external connections for analog inputs, analog outputs, and digital interfaces. The SPI Drive Electronics can be scaled to up more than 200 channels without any degradation in speed.

4.3.1 High-Power PDM SPI Drive Electronics

We have adapted our SPI Drive Electronics to provide higher power for driving PZT actuators for our plate deformble mirrors (PDMs). The high-power drive electronics takes the high-voltage output and adds a current amplifier stage to enable higher-speed operation of the PZT actuators. In the high-power drive electronics, power is provided via an external circular cable shown in Figure 2. The external power connection should never be severed without powering down the electronics since this can result in a very specific kind of damage to the electronics.

4.4 External Software Interfaces

We offer an external interface to our SPI Drive Electronics via C or C# DLLs. This is discussed more in the software manual.

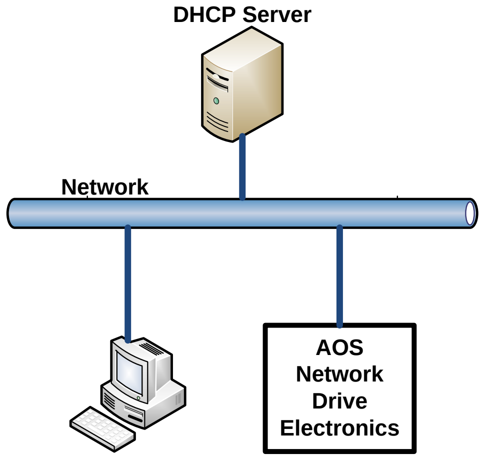
5 Ethernet Drive Electronics

5.1 Connecting the Drive Electronics to the Network

The AOS Ethernet drive electronics support IP configuration through Dynamic Host Configuration Protocol (DHCP) or assignment of a fixed IP address. If you are unsure what type of network you are using, please contact your network administrator.

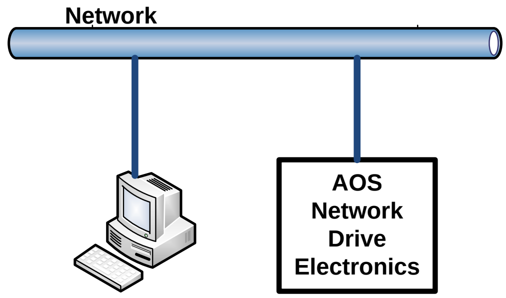
5.2 Connection to a DHCP Network

Initially, the drive electronics will be configured to enable automatic IP configuration via DHCP. When connected to a network supporting DHCP, the drive electronics will automatically obtain an IP address from the DHCP server. By default, the Network drive electronics is configured to automatically obtain an IP address from a DHCP server. Most corporate and home networks are configured to support DHCP. Connect the Network drive electronics to the DHCP enabled network and proceed to Section 5.5.



5.3 Connection to a Fixed IP Network

The Network drive electronics may be configured with a fixed IP address to support operation in a network without a DHCP server. Connect the Network drive electronics to the network and proceed to Section 5.5.



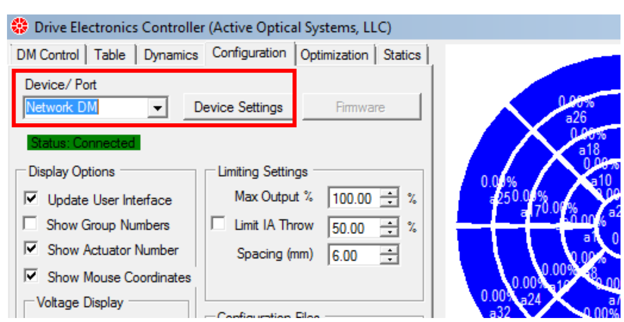
5.4 Direct Connection with a Crossover Cable

The Network drive electronics may communicate directly with a PC via a crossover cable. In this configuration, the computer and the drive electronics must be configured with fixed IP address from the same subnet. Connect the Network drive electronics directly to your computer using a crossover Ethernet cable and proceed to Section 5.5.

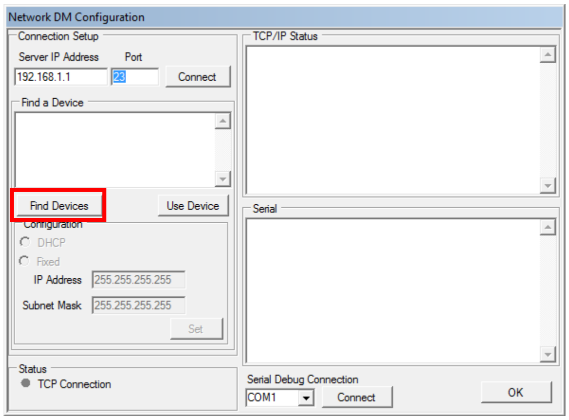
5.5 Connecting to the Drive Electronics with DM Controller

1. Launch the DM Controller application either through the start menu program group or from the Windows menu item of AOS Adaptive Optics.

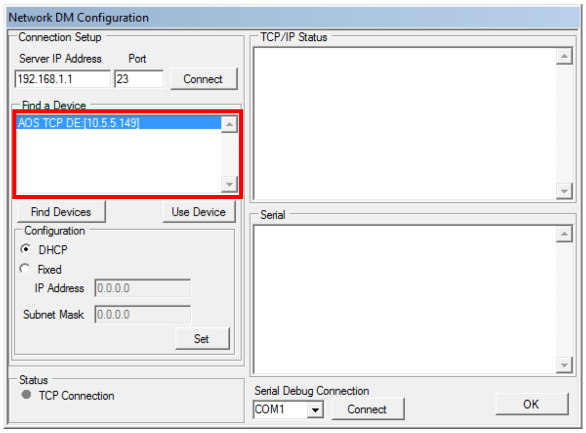
2. On the Configuration tab, select Network DM and click the "Device Settings" button to launch the Network configuration form:



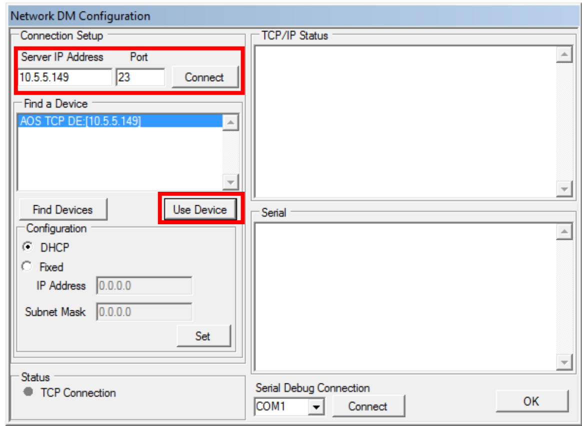
3. Click the "Find Devices" button:



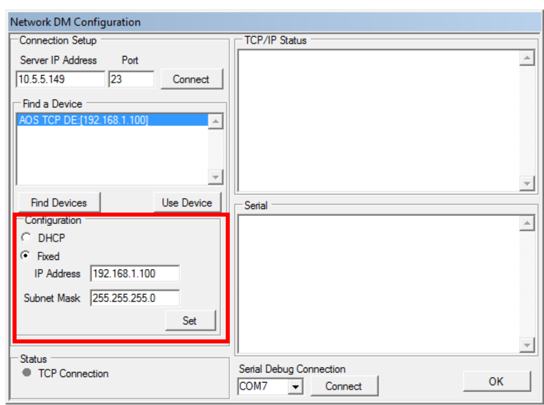
4. Select the device from the list that you wish to connect to or configure. The device shown below is configured for DHCP operation and has been assigned the IP address 10.5.5.149.



5. To connect to the device, click "Use Device" to transfer the IP address to the Server IP Address box and click "Connect." If the assigned IP address is 0.0.0.0, then the device was unable to obtain an IP address from a DHCP server. To connect using a fixed IP, proceed to the next step.



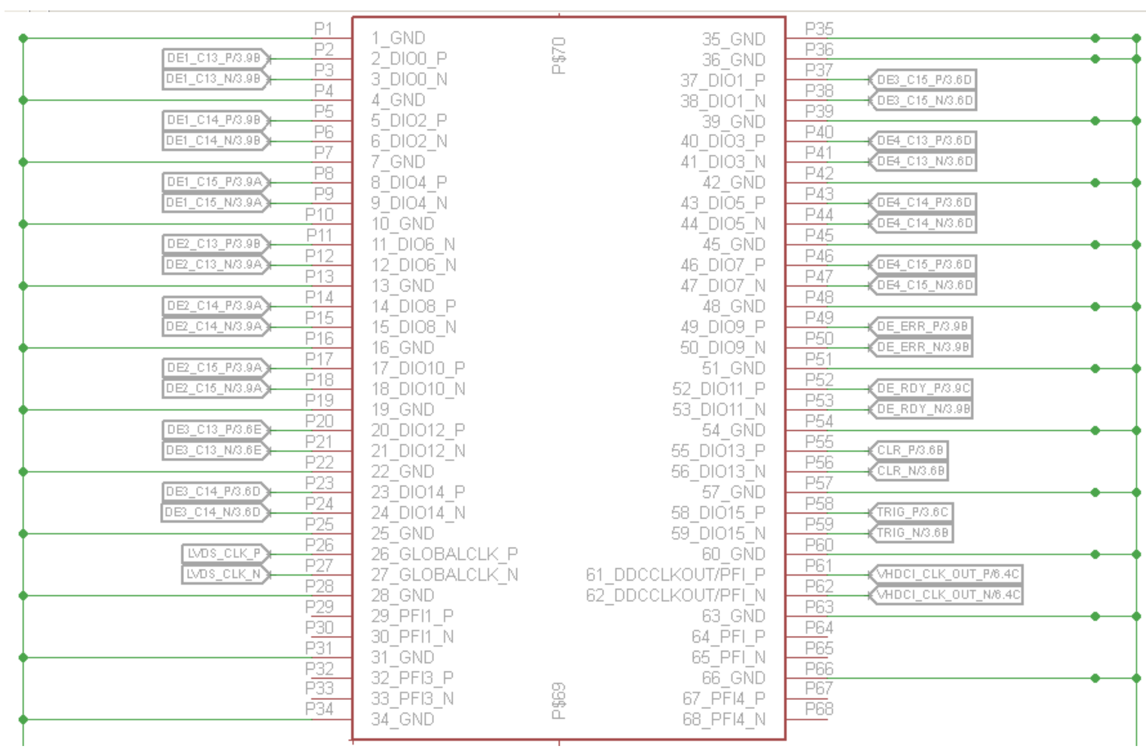
6. To configure the drive electronics to use a fixed IP, select the device you wish to configure from the list and select the “Fixed” radio button. Enter a valid IP address and subnet mask for your network and click the “Set” button. Reselect the device from the list and it should appear with the new IP address assigned. To revert to DHCP mode, select the “DHCP” radio button and click “Set.”



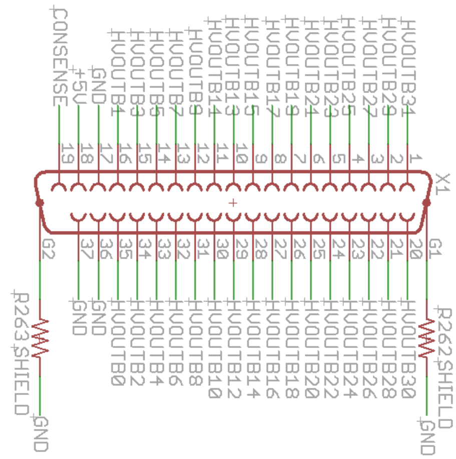
6 LVDS Drive Electronics-PDE6

The PZT high power drive electronics has either an Ethernet or an LVDS high speed interface. The Ethernet interface is the same as that described above. The LVDS interface is described in this section. The LVDS interface is addressable via a VHDCI connector to enable usage from an NI FPGA board. The pin out of the VHDCI connector is shown below. The VHDCI connector provides capability of addressing up to four 32-channel boards. Additional information is summarized in Chapter 12: Appendix: LVDS Protocol Summary for PDE6 .

Label	Pin Usage
C13	EN (enable)
C14	ADDR (data1)
C15	DATA (data2)



The output from each 32-channel board in the electronics is shown below.

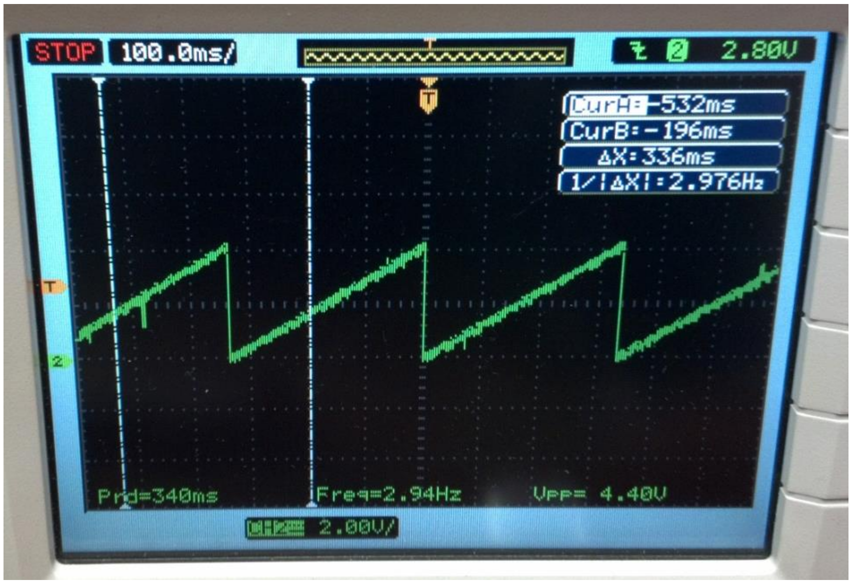


6.1 Data Command Format

The LVDS signal into the VHDCI port is framed by the enable signal. The LVDS interface has been tested up to 100 MHz. The DATA1 (D1) and DATA2 (D2) signals are both used to communicate data to each of up to 4 drive electronics (DE) boards. The command to the drive electronics begins with 8 bits of zero on both data lines. Then the 16-bit DAC commands are sent to the 32 channels via D1 and D2 simultaneously so that there are exactly 16 write cycles of 16 bits to transmit the data. The data is first written to the first two 8-channel DACs then to the second two DACs over a 50 MHz SPI communication protocol. The data is written to the DACs, but is not written until the TRIG signal goes high. The command and the data takes 16*17 = 272 clock cycles or 2.72 micro-seconds. Since the DACs have a limited bandwidth and a more complex command cycle, writing to all 32 DACs takes as much as 8 micro-seconds, but happens in parallel with the read-out of the LVDS signals, so the digital write time takes less than 10 micro-seconds.

6.2 Typical Performance

The following chart shows performance results from writing out a 16-bit ramp. The ramp reset every 336 ms corresponding to a write time from the DE board FPGA to the DACs of 5.1 micro-seconds. This corresponds well with a 25 clock-cycle write to the DAC at 50 MHz (5 micro-seconds). (NOTE: The glitching on the signal was due to a poor probe connection with the low-voltage DAC output.)

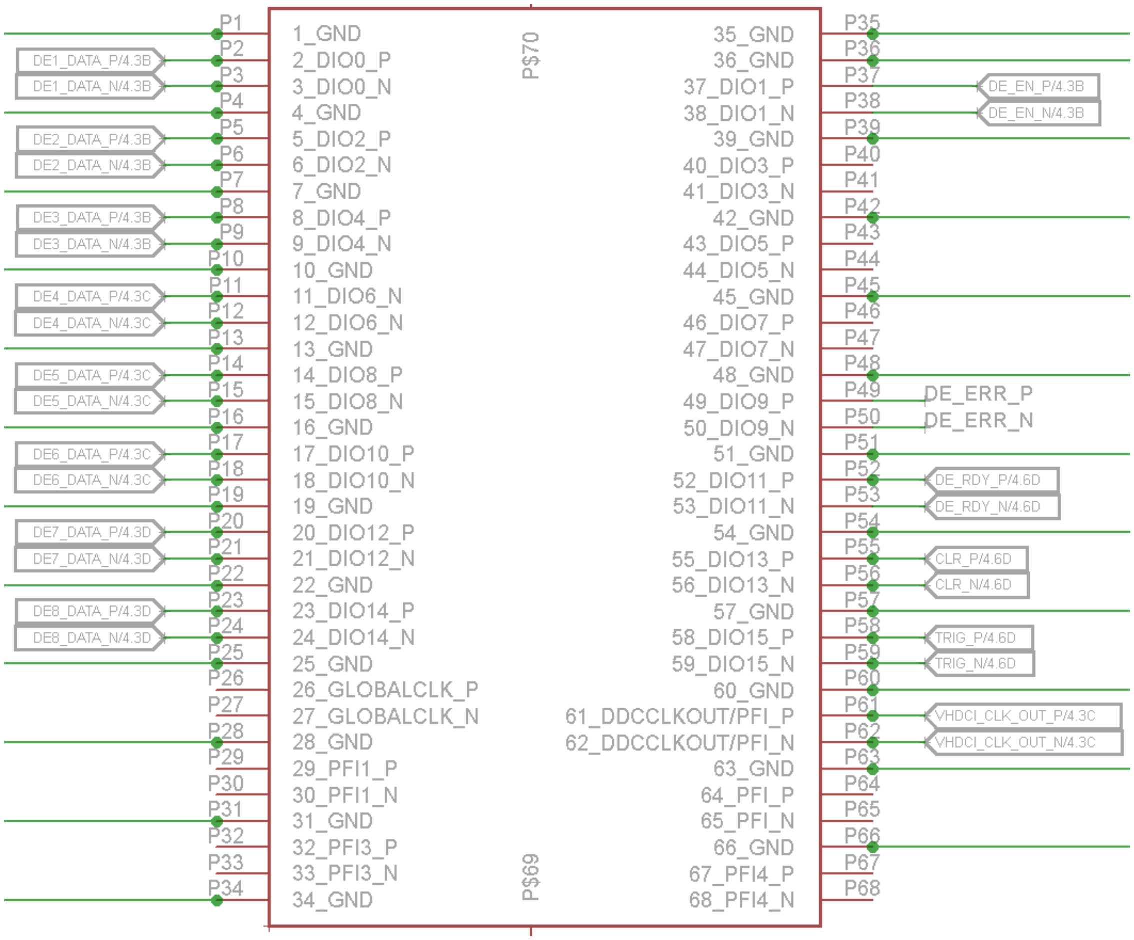


More information is available from AOS on the LVDS drive electronics signal. Please contact AOS for additional information on interfacing with the LVDS port.

7 LVDS Drive Electronics-PDE7

The PDE7 DE uses a system on module MicroZed board with a Xilinx Zynq FPGA core to provide more functionality inside the PDE7 box, and thus has introduced important changes that will impact the user. These include changes to the send protocol, data rate, maximum number of high voltage channels, and physical locations on the VHDCI connector, as well as built in inter-actuator limiting and flat command capabilities. The interface is addressable via the AOS software for Ethernet operation or through a VHDCI connector to enable usage from an NI FPGA board or other controller. The VHDCI connector provides capability of addressing up to eight 32-channel boards (256 total high voltage channels). Shown below is the new pinout for the VHDCI connector in the AOS drive electronics PDE7 box. Important information is summarized in Chapter 13: Appendix PDE7 Operation Information. This includes information about protocol changes, the location of the micro-sd card mounted on the MicroZed, and about the inter-actuator limiting and flat command functionalities.

Important: In the new PDE7 DE protocol, simultaneous DE_RDY and DE_ERR signals indicates an inter-actuator violation in the previous command. See Chapter 12: PDE7 Operation Information.



7.1 Data Command Format and General Operation

The LVDS signal into the VHDCI port is framed by the enable signal, which is now shared between all of the DE boards. The LVDS interface has been tested up to 200 MHz. The single data line (DE[n]_DATA) is used to communicate data to each of up to 8 drive electronics (DE) boards. The command to the drive electronics begins with 32 bits of zero on the 8 data lines. Then the 16-bit DAC commands are sent to the 32 channels via the corresponding DE data line so that there are exactly 32 write cycles of 16 bits to transmit the data. The data is first written to the first two 8-channel DACs then to the second two DACs over a 50 MHz SPI communication protocol. The data is written to the DACs, but is not written to the high voltage channels until the TRIG signal goes high.

7.1.1 Ethernet Operation

Ethernet operation is performed through the AOS software; see chapter 5 for more information. Control through the AOS software has not changed, but the following provides a guide to setup and connect to your PDE7 DE box, as well as some troubleshooting tips in case the DE is not operating as expected.

The user can connect the PDE7 box directly to a computer, through a router, or a gigabit switch. On boot, the DE searches for an IP via DHCP (this is the default configuration and will be assigned an IP address by your router). If it cannot get an IP address (e.g. you are connected directly to your computer) the box will default to an IP of 169.254.254.254 unless an IP configuration file already exists on the micro-sd card in which case it will use the fixed IP provided. Setting a fixed IP address can be accomplished through the AOS software. This will create an IP configuration file on the micro-sd card and will default to that IP in future use. The box can be set back to DHCP configuration through the AOS software, or by deleting the ipconfig file from the micro-sd. Connections to the PDE7 box are always on port 23.

Troubleshooting Tips:

- Firewall – You may need to disable windows firewall or make an exception, especially when using the default IP of 169.254.254.254

- IP Conflicts – You may need to set the IP address on the PC to 169.254.254.xxx where xxx is anything between 0 and 253. Make sure the accompanying subnet mask is 255.255.255.0

- Not finding DE in DHCP- Some network adapters do not properly support the UDP-based discovery that the DE boxes use. In this configuration, we recommend using a router that will show the DHCP table of assigned IP addresses. Once you have obtained the IP address, the AOS software can be used to connect to the DE even if the UDP find cannot locate it.

7.2 Typical Performance

The operation of this electronics will vary based on the input clock. For a 200MHz clock, the DE has a measured digital latency of <8μs.

8 DM Controller Software Installation

The software package that is provided for sending commands to the AOS Drive Electronics is called DM Controller. It is documented in the AOS Software Manual. More details on the software installation can be found in the AOS Software Manual.

9 Appendix: DM File Format

The DM file format provides the DMController with information about the deformable mirror and how to adjust the user interface. The first character if each line specifies the type of information that is on that line. The type specifers are as follows:

- A = Actuator Definition
- V = Voltage Information
- G = Grouping Information
- C = Communication Specification

Each line is comma delimited and terminated with a newline character.

"A" = Actuator Definitions

Following the A in this order is:

1. the number of points in the shape definition,
2. the channel number of the electronics that controls this actuator, and
3. a series of x and y coordinates for the shape representing that actuator.

The index number in the AOS DM Controller software is given by the order in which these actuators are read into the program.

"V" = Voltage Information

Following the V is a comma delimited list of the voltage on each of the actuators in counts on the DAC.

"G"=Grouping Information

Following the G is a comma delimited list of each actuator's group.

"C" = Communication Specification

Communication with the USB AOS Drive Electronics is done using a virtual COM port. The specifications for this communication are stored in the "C" section. The first line specifies the baud rate. Subsequent lines specify the parity, data-bits, and stop bits. In earlier versions these lines were used by the software, but the current version no longer uses information specified in this section.

Example DM File

Here is an example DM file for a 31 actuator hex-grid DM:

```
A,7,1,3.034615,1.559956,2.965385,1.559956,2.930769,1.500000,2.965385,1.440044,3.034615,1.
440044,3.069231,1.500000,3.034615,1.559956,
A,7,2,3.034615,1.688527,2.965385,1.688527,2.930769,1.628571,2.965385,1.568616,3.034615,1.
568616,3.069231,1.628571,3.034615,1.688527,
A,7,3,3.145961,1.624241,3.076731,1.624241,3.042115,1.564286,3.076731,1.504330,3.145961,1.
504330,3.180577,1.564286,3.145961,1.624241,
A,7,4,3.145961,1.495670,3.076731,1.495670,3.042115,1.435714,3.076731,1.375759,3.145961,1.
375759,3.180577,1.435714,3.145961,1.495670,
A,7,5,3.034615,1.431384,2.965385,1.431384,2.930769,1.371429,2.965385,1.311473,3.034615,1.
311473,3.069231,1.371429,3.034615,1.431384,
A,7,6,2.923269,1.495670,2.854039,1.495670,2.819423,1.435714,2.854039,1.375759,2.923269,1.
375759,2.957885,1.435714,2.923269,1.495670,
A,7,7,2.923269,1.624241,2.854039,1.624241,2.819423,1.564286,2.854039,1.504330,2.923269,1.
504330,2.957885,1.564286,2.923269,1.624241,
```

```
A,7,8,3.034615,1.817098,2.965385,1.817098,2.930769,1.757143,2.965385,1.697187,3.034615,1.
697187,3.069231,1.757143,3.034615,1.817098,
A,7,9,3.145961,1.752813,3.076731,1.752813,3.042115,1.692857,3.076731,1.632902,3.145961,1.
632902,3.180577,1.692857,3.145961,1.752813,
A,7,10,3.257308,1.688527,3.188077,1.688527,3.153461,1.628571,3.188077,1.568616,3.257308,1
.568616,3.291923,1.628571,3.257308,1.688527,
A,7,11,3.257308,1.559956,3.188077,1.559956,3.153461,1.500000,3.188077,1.440044,3.257308,1
.440044,3.291923,1.500000,3.257308,1.559956,
A,7,12,3.257308,1.431384,3.188077,1.431384,3.153461,1.371429,3.188077,1.311473,3.257308,1
.311473,3.291923,1.371429,3.257308,1.431384,
A,7,13,3.145961,1.367098,3.076731,1.367098,3.042115,1.307143,3.076731,1.247187,3.145961,1
.247187,3.180577,1.307143,3.145961,1.367098,
A,7,14,3.034615,1.302813,2.965385,1.302813,2.930769,1.242857,2.965385,1.182902,3.034615,1
.182902,3.069231,1.242857,3.034615,1.302813,
A,7,15,2.923269,1.367098,2.854039,1.367098,2.819423,1.307143,2.854039,1.247187,2.923269,1
.247187,2.957885,1.307143,2.923269,1.367098,
A,7,16,2.811923,1.431384,2.742692,1.431384,2.708077,1.371429,2.742692,1.311473,2.811923,1
.311473,2.846539,1.371429,2.811923,1.431384,
A,7,17,2.811923,1.559956,2.742692,1.559956,2.708077,1.500000,2.742692,1.440044,2.811923,1
.440044,2.846539,1.500000,2.811923,1.559956,
A,7,18,2.811923,1.688527,2.742692,1.688527,2.708077,1.628571,2.742692,1.568616,2.811923,1
.568616,2.846539,1.628571,2.811923,1.688527,
A,7,19,2.923269,1.752813,2.854039,1.752813,2.819423,1.692857,2.854039,1.632902,2.923269,1
.632902,2.957885,1.692857,2.923269,1.752813,
A,7,20,3.145961,1.881384,3.076731,1.881384,3.042115,1.821429,3.076731,1.761473,3.145961,1
.761473,3.180577,1.821429,3.145961,1.881384,
A,7,21,3.257308,1.817098,3.188077,1.817098,3.153461,1.757143,3.188077,1.697187,3.257308,1
.697187,3.291923,1.757143,3.257308,1.817098,
A,7,22,3.368654,1.624241,3.299423,1.624241,3.264808,1.564286,3.299423,1.504330,3.368654,1
.504330,3.403269,1.564286,3.368654,1.624241,
A,7,23,3.368654,1.495670,3.299423,1.495670,3.264808,1.435714,3.299423,1.375759,3.368654,1
.375759,3.403269,1.435714,3.368654,1.495670,
A,7,24,3.257308,1.302813,3.188077,1.302813,3.153461,1.242857,3.188077,1.182902,3.257308,1
.182902,3.291923,1.242857,3.257308,1.302813,
A,7,25,3.145961,1.238527,3.076731,1.238527,3.042115,1.178571,3.076731,1.118616,3.145961,1
.118616,3.180577,1.178571,3.145961,1.238527,
A,7,26,2.923269,1.238527,2.854039,1.238527,2.819423,1.178571,2.854039,1.118616,2.923269,1
.118616,2.957885,1.178571,2.923269,1.238527,
A,7,27,2.811923,1.302813,2.742692,1.302813,2.708077,1.242857,2.742692,1.182902,2.811923,1
.182902,2.846539,1.242857,2.811923,1.302813,
A,7,28,2.700577,1.495670,2.631346,1.495670,2.596731,1.435714,2.631346,1.375759,2.700577,1
.375759,2.735192,1.435714,2.700577,1.495670,
A,7,29,2.700577,1.624241,2.631346,1.624241,2.596731,1.564286,2.631346,1.504330,2.700577,1
.504330,2.735192,1.564286,2.700577,1.624241,
A,7,30,2.811923,1.817098,2.742692,1.817098,2.708077,1.757143,2.742692,1.697187,2.811923,1
.697187,2.846539,1.757143,2.811923,1.817098,
A,7,31,2.923269,1.881384,2.854039,1.881384,2.819423,1.821429,2.854039,1.761473,2.923269,1
.761473,2.957885,1.821429,2.923269,1.881384,
V,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
G,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
C,9600,
C,None,
C,8,
C,1,
C,1,
C,255,
```

10 Appendix: Software Channel to 37-D Pin-Out Mapping

10.1 Software Channel to 37-D Pin Mapping

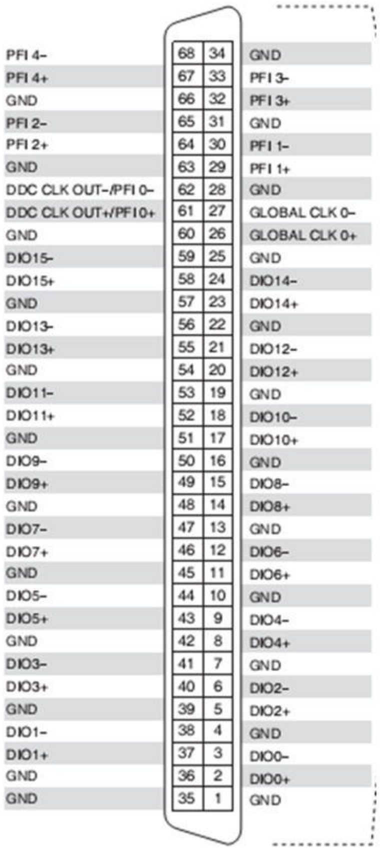
Software Channel	37D Pin
0	34
1	15
2	35
3	16
4	32
5	13
6	33
7	14
8	30
9	11
10	31
11	12
12	28
13	9
14	29
15	10
16	26
17	7
18	27
19	8
20	24
21	5
22	25
23	6
24	22
25	3
26	23
27	4
28	20
29	1
30	21
31	2
GND	17,36

11 Appendix: Troubleshooting

Problem	Potential Resolution
The AOS Drive Electronics is not being recognized by my computer.	• Check that the USB cable is attached between the computer and the Drive Electronics. • Check that power is connected to the Drive Electronics and that the switch is turned on. • Check that Windows is responding. • Check to see if the Device Manager (Start-Control Panel-System-Hardware Tab-Device Manager Button) changes when the Drive Electronics is turned on. A COM port should appear under the Ports section. • Try connecting the hardware to another computer. • Remove the power connection. Check the fuse in the power entry module.
The DM does not appear to be moving in response to voltage commands.	• Check the USB and power connections to the Drive Electronics. Make sure the device is turned on. • Check that the computer is connected to the Drive Electronics in the Device Manager (Start-Control Panel-System-Hardware Tab-Device Manager Button) under the Ports section. • Start the DMController software. A message box should appear indicating that the software has found the drive electronics. If this does not appear, the device is not properly connected. • Apply up to 255 counts to each of the actuators and look either at any diagnostic like a focused spot reflected from the deformable mirror, an interferometer, or a Hartmann sensor.
Unsure about the USB Drive Electronics Connectivity to the Computer	• To check if the drive electronics is properly connected to the PC first check the computer's Device Manager. This can be accessed in the control panel. The "Drive Electronics" is under the Ports section.

12 Appendix: LVDS Protocol Summary for PDE6

VHDCI Connector Definition

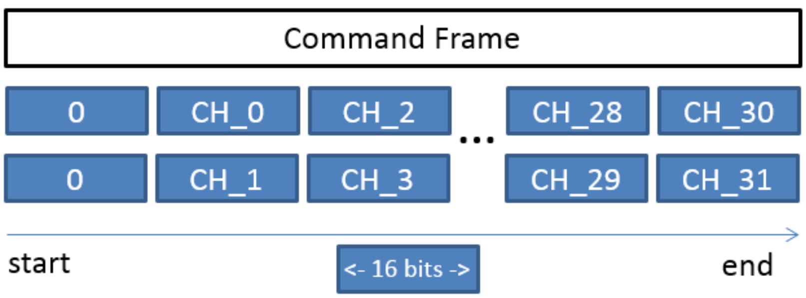


Pin #	Pin Name	Signal	Direction	Pin #	Pin Name	Signal	Direction
1	GND	GND	-	35	GND	GND	-
2	D0+	DE1_EN	IN	36	GND	GND	-
3	D0-			37	D1+	DE3_D2	IN
4	GND	GND	-	38	D1-		
5	D2+	DE1_D1	IN	39	GND	GND	-
6	D2-			40	D3+	DE4_EN	IN
7	GND	GND	-	41	D3-		
8	D4+	DE1_D2	IN	42	GND	GND	-
9	D4-			43	D5+	DE4_D1	IN
10	GND	GND	-	44	D5-		
11	D6+	DE2_EN	IN	45	GND	GND	-
12	D6-			46	D7+	DE4_D2	IN
13	GND	GND	-	47	D7-		
14	D8+	DE2_D1	IN	48	GND	GND	-
15	D8-			49	D9+	DE_ERR	OUT
16	GND	GND	-	50	D9-		
17	D10+	DE2_D2	IN	51	GND	GND	-
18	D10-			52	D11+	DE_RDY	OUT
19	GND	GND	-	53	D11-		
20	D12+	DE3_EN	IN	54	GND	GND	-
21	D12-			55	D13+	DE_CLR	IN
22	GND	GND	-	56	D13-		
23	D14+	DE3_D1	IN	57	GND	GND	-
24	D14-			58	D15+	DE_TRIG	IN
25	GND	GND	-	59	D15-		
26	GLB_CLK+	NC	-	60	GND	GND	-
27	GLB_CLK-	NC	-	61	CLK+	CLK	IN
28	GND	GND	-	62	CLK-		
29	PFI_1+	NC	-	63	GND	GND	-
30	PFI_1-	NC	-	64	PFI_2+	NC	-
31	GND	GND	-	65	PFI_2-	NC	-
32	PFI_3+	NC	-	66	GND	GND	-
33	PFI_3-	NC	-	67	PFI_4+	NC	-
34	GND	GND	-	68	PFI_4-	NC	-

Protocol

- Each DE board has two data lines (D1 and D2) and an enable signal.
- CLK, DE_ERR, DE_RDY, DE_CLR, and DE_TRIG are shared for all of the DE boards.
- Commands are interpreted as 16-bit unsigned integers.
- Commands to the DE boards are serialized MSB first on D1 and D2.
- Inputs are read on the rising edge of CLK when EN is high.

Signal	Description
DE[n]_EN	D1 and D2 for the n'th DE board are read when this signal is high
DE[n]_D1	First serialized data channel for DE board n
DE[n]_D2	Second serialized data channel for DE board n
DE_ERR	Goes high when one or more DE boards is in the error state.
DE_RDY	Goes high when all DE boards are ready to receive a new command or trigger.
DE_CLR	Sets all output DACs to zero on all DE boards and resets the drive electronics. Active Low
DE_TRIG	Set high to all of the DE boards to assert the last command on the DACs. Wait for DE_RDY to go low before de-asserting DE_TRIG.



First 16 bits = constant 0 for current version
Following 16-Bit Pieces = DE voltage command data

12.1 Electrical Characteristics

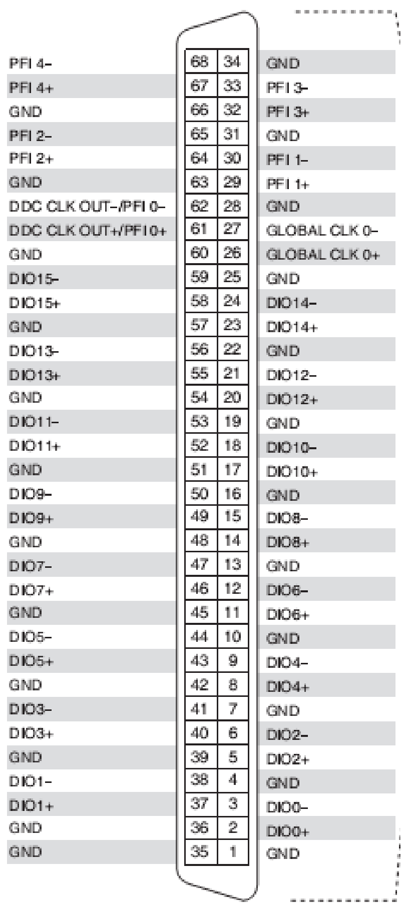
- Signals are 3.3V LVDS
- Input signals are 100Ω terminated
- Input signals are read on the rising edge of CLK.

12.2 Example Command Sequence

1. Wait for DE_RDY to go high indicating that the electronics is ready for a command.
2. Send serialized command frame to electronics.
3. Wait for DE_RDY to go high indicating that the electronics is ready for a trigger.
4. Set DE_TRIG high assert the last command on the DACs.
5. Wait for DE_RDY to go low before lowering DE_TRIG.
6. GOTO 1

13 Appendix: PDE7 Operating Information

VHDCI Connector Definition-PDE7

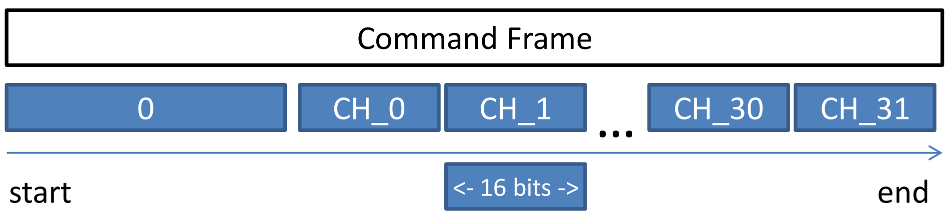


Pin #	Pin Name	Signal	Direction	Pin #	Pin Name	Signal	Direction
1	GND	GND	-	35	GND	GND	-
2	D0+	DE1_DATA	IN	36	GND	GND	-
3	D0-			37	D1+	DE_EN	IN
4	GND	GND	-	38	D1-		
5	D2+	DE2_DATA	IN	39	GND	GND	-
6	D2-			40	D3+	NC	-
7	GND	GND	-	41	D3-		
8	D4+	DE3_DATA	IN	42	GND	GND	-
9	D4-			43	D5+	NC	-
10	GND	GND	-	44	D5-		
11	D6+	DE4_DATA	IN	45	GND	GND	-
12	D6-			46	D7+	NC	-
13	GND	GND	-	47	D7-		
14	D8+	DE5_DATA	IN	48	GND	GND	-
15	D8-			49	D9+	DE_ERR	OUT
16	GND	GND	-	50	D9-		
17	D10+	DE6_DATA	IN	51	GND	GND	-
18	D10-			52	D11+	DE_RDY	OUT
19	GND	GND	-	53	D11-		
20	D12+	DE7_DATA	IN	54	GND	GND	-
21	D12-			55	D13+	DE_CLR	IN
22	GND	GND	-	56	D13-		
23	D14+	DE8_DATA	IN	57	GND	GND	-
24	D14-			58	D15+	DE_TRIG	IN
25	GND	GND	-	59	D15-		
26	GLB_CLK+	NC	-	60	GND	GND	-
27	GLB_CLK-	NC	-	61	CLK+	CLK	IN
28	GND	GND	-	62	CLK-		
29	PFI_1+	NC	-	63	GND	GND	-
30	PFI_1-	NC	-	64	PFI_2+	NC	-
31	GND	GND	-	65	PFI_2-	NC	-
32	PFI_3+	NC	-	66	GND	GND	-
33	PFI_3-	NC	-	67	PFI_4+	NC	-
34	GND	GND	-	68	PFI_4-	NC	-

Protocol

- Each DE board has a single data line and all share an enable signal.
- CLK, DE_ERR, DE_RDY, DE_CLR, and DE_TRIG are also shared for all of the DE boards.
- Commands are interpreted as 16-bit unsigned integers.
- Commands to the DE boards are serialized MSB first on their respective data line.
- Inputs are read on the rising edge of CLK when EN is high.

Signal	Description
DE_EN	All DE board data signals are read when this signal is high
DE[n]_DATA	Serialized data channel for DE board n
DE_ERR	Goes high when one or more DE boards is in the error state. This indicates a bad header if DE_RDY is not also high. DE_ERR and DE_RDY will both be high in the case of an inter-actuator limit violation.
DE_RDY	Goes high when all DE boards are ready to receive a new command or trigger.
DE_CLR	Sets all output DACs to zero on all DE boards and resets the drive electronics. Active Low
DE_TRIG	Set high to all of the DE boards to assert the last command on the DACs. Wait for DE_RDY to go low before de-asserting DE_TRIG.



First 32 bits = constant 0
Following 16-Bit Pieces = DE voltage command data

13.1 Electrical Characteristics

- Signals are 3.3V and 2.5V LVDS Compatible
- Input signals are 100Ω terminated
- Input signals are read on the rising edge of CLK.

13.2 Example Command Sequence

1. Wait for DE_RDY to go high indicating that the electronics is ready for a command and monitor the DE_ERR line to determine if the previous command returned an inter-actuator limit violation.
2. Send serialized command frame to electronics.
3. Wait for DE_RDY to go high indicating that the electronics is ready for a trigger.
4. Set DE_TRIG high to assert the last command on the DACs.
5. Wait for DE_RDY to go low before lowering DE_TRIG.

6. GOTO 1

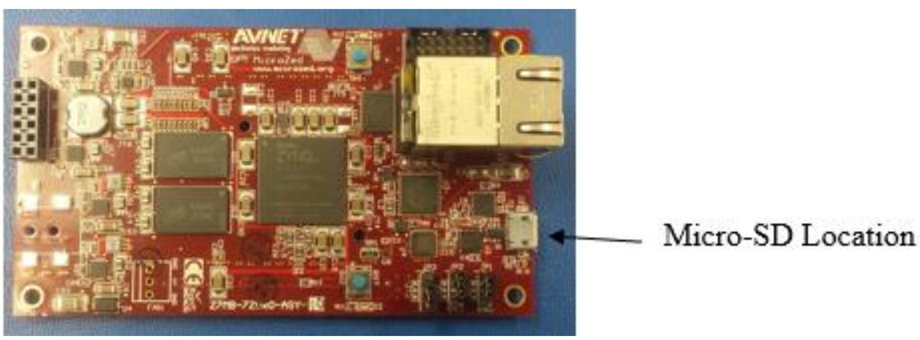
13.3 Fixed IP Operation

Fixed IP Operation functions the same as controlled through the AOS software. The only difference is that a PDE7 operating in fixed IP mode will generate an ipconfig file on the mounted SD card which should never be modified. If the box is switched back to DHCP mode the file will automatically be deleted.

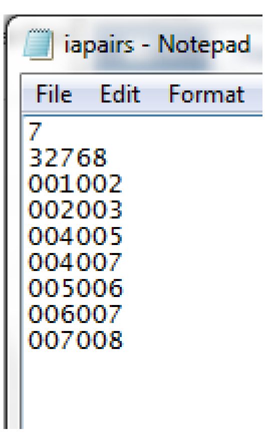
13.4 Inter-Actuator Limiting

This version of the drive electronics allows for inter-actuator limiting to prevent potential damage to a deformable mirror. This limiting is determined by a text file loaded on startup. AOS can provide these files and it is highly discouraged that the customer modify the file provided to them, especially anything other than the limit threshold value. These instructions give the user the ability to modify their threshold or paired actuators at their own risk.

In the drive electronics box there is a system on module card called a microzed. This is a removable, red card located on the interface board. Mounted on the back of this is a micro SD card that will need to be removed and the contents modified to change inter-actuator or flat command behavior. Below is an image showing the location of the micro-sd card mounted on the back of the MicroZed.



The iapairs.txt file is what determines the number of actuator pairs as well as the maximum difference between these paired actuators and is structured as follows. The first line is the number of desired pairs, followed by the limit value in counts from 0(0% no differential throw) to 65535(100% no limiting). These lines are followed by software channel pairs, which are different than actuator numbers and must be determined before attempting to modify this file. These pairs must be 3 characters in length (actuator 1 is written as 001 and actuator 10 is written as 010) and the two channels to be paired are put on a line. The total number of pairs must match the number given in the first line of the file. Below is an example of the file format with 7 inter-actuator pairs and a threshold limit value of 50%.


iapairs - Notepad

File Edit Format

```
7
32768
001002
002003
004005
004007
005006
006007
007008
```

Due to pipelining in the drive electronics, the minimum number of inter-actuator pairs is 7. If no limiting is desired, the user can set the threshold to 65535 or set the pairs to compare channels to themselves down to a minimum of 7 pairs (e.g. 001001 cannot give a violation).

13.5 Flatfile

PDE7 drive electronics also contain the option for a flatfile.txt file which will be sent to the HV channels upon power-up. The flatfile.txt provided is a flat determined by AOS and should not be modified. If this isn't a desired action, the file can be deleted or renamed for later use as the command in the file will only be applied if the file is named "flatfile.txt".

14 Ethernet Command Format

WARNING: It is highly recommended that you use the AOS software when interfacing to the drive electronics over the Ethernet port because it enables limiting and inter-actuator limiting. If your application needs to write directly to the drive electronics over the Ethernet port, you are doing so at your own risk.

To command the drive electronics (DE) over the Ethernet port:

1. Connect to the DE using a TCP connection. Unless otherwise indicated the port is typically 23. You can use the AOS software or your router or the "ipconfig" command to identify the IP address.
2. Send a command with the following format:

mwrite [2-bytes specifying the unsigned integer number of commands][2-bytes per command in DAC counts (typically 16-bit)][\r][\n]

3. If successful, the TCP connection will return the prompt, ">>".

Here is example C# code for writing to the drive electronics over the Ethernet port:

```
private int WriteVoltagesToDM(UInt16[] voltageCounts, NetworkStream m_stream)
{
            byte[] cmd = new byte[voltageCounts.Length * 2 + 11];
            cmd[0] = (byte)'m';
            cmd[1] = (byte)'w';
            cmd[2] = (byte)'r';
            cmd[3] = (byte)'i';
            cmd[4] = (byte)'t';
            cmd[5] = (byte)'e';
            cmd[6] = (byte)' ';

            UInt16 length = (UInt16)(voltageCounts.Length * 2);
            byte[] lbytes = BitConverter.GetBytes(length);
            cmd[7] = lbytes[0];
            cmd[8] = lbytes[1];


            int cnt = 9;
            byte[] valBytes;
            for (int i = 0; i < voltageCounts.Length; i++)
            {
                valBytes = BitConverter.GetBytes(voltageCounts[i]);
                cmd[cnt++] = valBytes[0];
                cmd[cnt++] = valBytes[1];
            }

            cmd[cnt++] = (byte)'\r';
            cmd[cnt++] = (byte)'\n';
            byte[] retData = new byte[4];

            if (m_stream == null || !m_stream.CanWrite) return(-1);
            else
            {
                lock (locker)
                {
                    byte[] cmd_b = cmd;
                    byte[] crnl = new byte[2];
                    m_stream.Write(cmd_b, 0, cmd_b.Length);
                }
            }
            return 0;
}
```

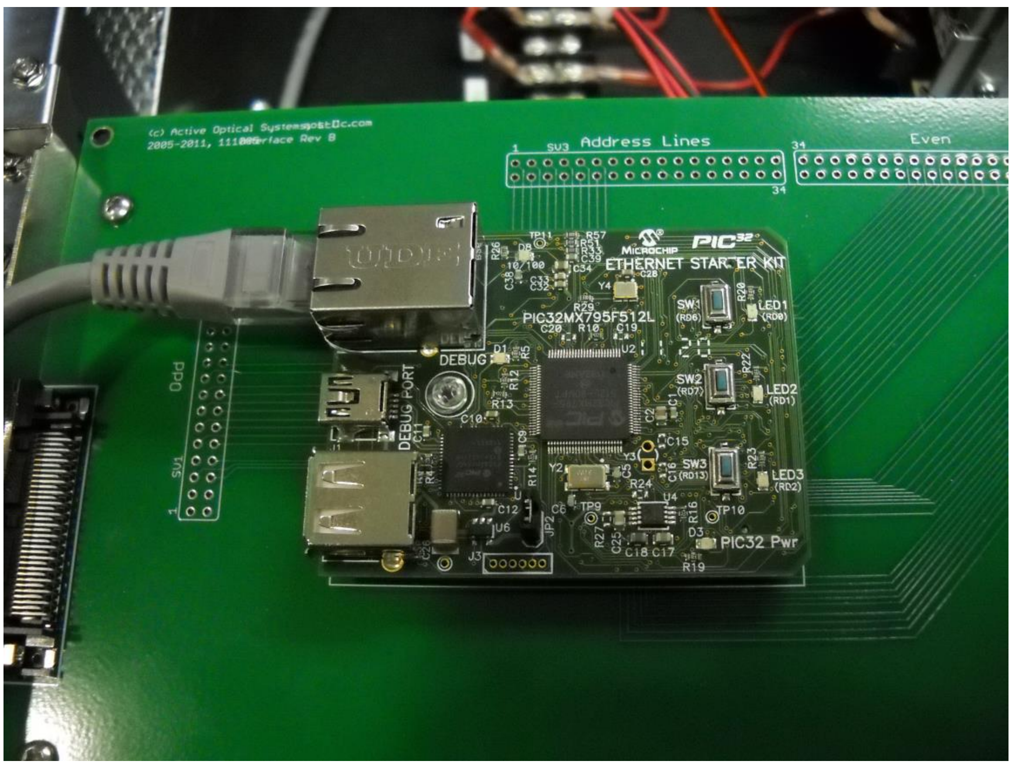
15 Appendix: Instruction on Switching VHDCI LVDS to Ethernet Interface with the PIC 32

NOTE: Only qualified electronics technicians should attempt this transition. AOS can perform this switch for you. Contact AOS to arrange for this operation to be performed by our qualified personnel.

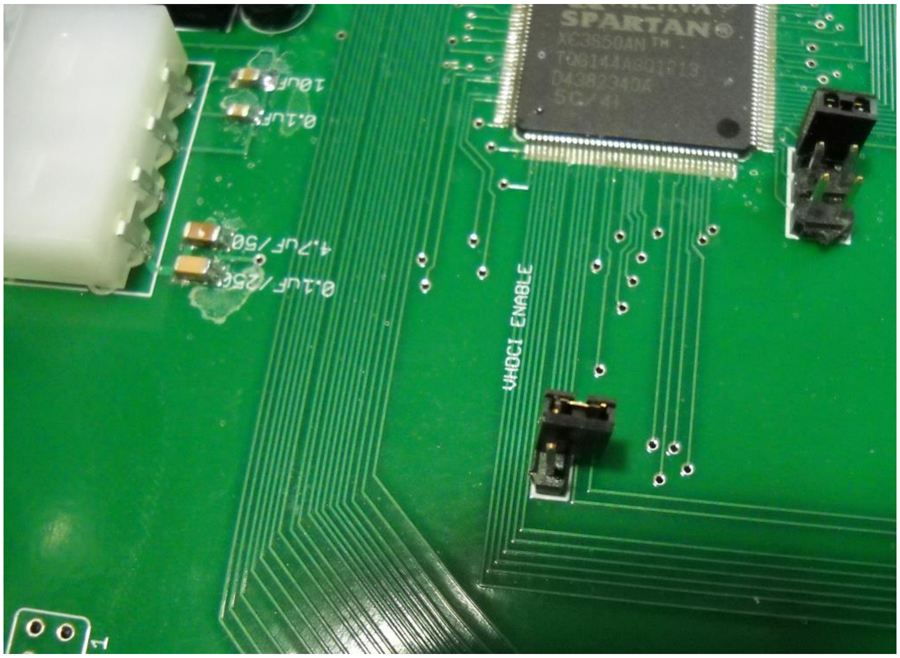
The drive electronics comes with a PIC32 Ethernet PCB that can be used to switch the electronics into an Ethernet interface mode. Attempting this switch should only be done by qualified electronics personnel. It is essential that the electronics be powered off for at least 5 minutes to allow for the power supplies to drain the capacitors prior to making this switch. It is also essential that a grounding strap be used during this operation. Grounding of the box can be maintained by keeping the box plugged into the wall, but this does expose the operator to 120VAC from the wall inside the box, so care must be taken to avoid electric shock.

Here is the switching procedure:

1. Open the box top using the screws on the perimeter.
2. Remove the interface board (the board on the far side of the box with the VHDCI connector) from the backplane.
3. Remove the PIC32 from the static bag under grounding precautions and plug it into the interface board as shown here. Connect the internal short Ethernet cable from the external interface to the PIC32.



4. Disconnect the jumper labeled “VHDCI Enable” adjacent to the Spartan FPGA chip as shown below.



5. Replace the Interface Board into the backplane and reassemble the box. Now the board will be able to be commanded via Ethernet from the AOS software.

16 Appendix: Switching from LVDS to Ethernet on the 3U Microzed interface boards

To switch from Ethernet to LVDS, create a file called “mode.txt” with only the text number “0” as the first character in the file on the Micro SD card.