ACCES I/O PRODUCTS, INC.

March 12, 2014

STATEMENT OF LETTER OF VOLATILITY

The following statement applies to the USB/104 Data Acquisition product line, as follows:

USB-DIO-32	USB-CTR-15	USB-AI12-32	USB-AIO16-64MA	USB-AIO12-128A
USB-DIO-48	USB-IIRO4-2SM	USB-AI12-32E	USB-AIO16-64ME	USB-AIO12-128
USB-DIO-96	USBP-DIO16RO8	USB-AI16-64A	USB-AIO12-64MA	USB-AIO12-128E
USB-DIO-32I	PICO-DIO16RO8	USB-AI16-64E	USB-AIO12-64M	USB-AO16-16A
USB-DIO24	USBP-II8IDO4A	USB-AI12-64A	USB-AIO12-64ME	USB-AO16-16E
USB-DIO24-CTR6	PICO-II8IDO4A	USB-AI12-64	USB-AIO16-32A	USB-AO16-12A
USB-DIO-16H	USB-DIO48DO24	USB-AI12-64E	USB-AIO16-32E	USB-AO16-12E
USB-DI16A	USB-DIO24DO12	USB-AI16-96A	USB-AIO12-32A	USB-AO16-8A
USB-DO16A	USB-DO24	USB-AI16-96E	USB-AIO12-32	USB-AO16-8E
USB-DIO-16A	USB-DIO24	USB-AI12-96A	USB-AIO12-32E	USB-AO16-4A
USB-IIRO-16	USB-AI16-16A	USB-AI12-96	USB-AIO16-64A	USB-AO16-4E
USB-II-16	USB-AI16-16E	USB-AI12-96E	USB-AIO16-64E	USB-AO12-16A
USB-RO-16	USB-AI12-16A	USB-AI16-128A	USB-AIO12-64A	USB-AO12-16E
USB-IIRO-8	USB-AI12-16	USB-AI16-128E	USB-AIO12-64	USB-AO12-12A
USB-II-8	USB-AI12-16E	USB-AI12-128A	USB-AIO12-64E	USB-AO12-12E
USB-IIRO-4	USB-AI16-64MA	USB-AI12-128	USB-AIO16-96A	USB-AO12-8A
USB-IDIO-16	USB-AI16-64ME	USB-AI12-128E	USB-AIO16-96E	USB-AO12-8E
USB-IDO-16	USB-AI12-64MA	USB-AIO16-16A	USB-AIO12-96A	USB-AO12-4A
USB-IDIO-8	USB-AI12-64M	USB-AIO16-16E	USB-AIO12-96	USB-AO12-4E
USB-IDIO-4	USB-AI12-64ME	USB-AIO12-16A	USB-AIO12-96E	USB-DA12-8A
USB-II16-OEM	USB-AI16-32A	USB-AIO12-16	USB-AIO16-128A	USB-DA12-8E
USB-II8-OEM	USB-AI16-32E	USB-AIO12-16E	USB-AIO16-128E	USB-DO16-16
USB-II4-OEM	USB-AI12-32A			

Questions and Answers

1. Are all memory components within the hardware device volatile, meaning that any data stored on these components is lost when power to the unit is removed?

No.

2. If non-volatile components exist, are any of them designed to be modified by the user or by the devices during normal operations? Please briefly describe the type of data stored in these components.

Yes.

These devices contain a plug-and-play (PnP) configuration EEPROM 64-kbits in length. The last 512 bytes are designated for customer use, with the remaining bytes reserved for factory use including the required PnP data, as well as serial number and model number verification codes. Models designated "-S25" have had the firmware burned into the onboard EEPROM, in the "factory reserved" space.

The user-designated 512 bytes can be used for any data the customer desires through our AIOUSB.DLL's "CustomEEPROMWrite()" and "CustomEEPROMRead()" APIs. The remaining factory reserved bytes are accessible only through IOCTL calls, or use of the undocumented "GenericVendorWrite()" "GenericVendorRead()" functions; however, these devices are based on Cypress' EzUSB (FX2) chips, which are commonplace in the industry, and procedures to access them are well known.

3. Are any RAM components battery-backed? If so, please briefly describe the nature being retained and the location of the memory.

No.

4. Where is the BIOS located? Can it be locked out with a password? If yes, please provide the sequence to do so.

> N/A. In general the devices run "firmware" loaded by drivers from the host computer into onboard volatile RAM at each device reset; no firmware is located in non-volatile memory. None of these devices have "BIOS" as such. The firmware loaded by the drivers is part of the software driver package (AIOUSB).

5. Does this equipment contain any devices, such as RF transmitters and dial out capabilities via either telephone landline or cellular transmission?

> No.

Clearing Procedures

There are many procedures available to clear the contents of the onboard nonvolatile (EEPROM) memory.

At the lowest level:

> Make USB control transfers to the vendor-specific request number 0xA2 specifying the address in the EEPROM you wish to write and passing a buffer containing the pattern you wish to use to overwrite the onboard EEPROM. Because the Cypress FX2 is restricted to 4096 bytes per control transfer two requests will be needed. For example:

```
allocate and clear buffer=4096 byte array of zeroes
call GenericVendorWrite(0xA2, 0, 0, sizeof(buffer), buffer)[1]
call GenericVendorWrite(0xA2, 0, 4096, sizeof(buffer), buffer)
```

At the highest level:

> Download http://accesio.com/files/forever/USB104LoVClear.exe and click the "Erase now" button. Make sure no errors result.

A wide variety of methods for clearing the EEPROM exist between these extremes. For example, ACCES offers debugging utilities (FirmLoader.exe, KeyMaster.exe, etc.) which can be used to implement clearing schemes.

Note [1]: The first five bytes of EEPROM will be erased with this statement as shown. Erasing these five bytes will inhibit the proper plug-and-play operation of the device, effectively breaking it.

The statement to erase the non-volatile data area without affecting the first five bytes is as follows:

```
allocate and clear buffer=4096 byte array of zeroes
call GenericVendorWrite(0xA2, 5, 0, 4091, buffer)
call GenericVendorWrite(0xA2, 0, 4096, 4096, buffer)
```