Message-based Firmware Specification

The message-based firmware architecture uses a String Message Based Interface (**SMBI**) to communicate with supported devices using a minimal number of device transactions.

For those interested in developing a driver to interface directly with the device, the source code for the DAQFlex software API may be used as a guide.

Help revision: 5.0

February, 2011

SMBI data types

Special formatting characters are returned with RAW data (except streaming data from bulk transfers) to indicate the following data types:

- char[64]: String data in a character array
- uint8: unsigned 8-bit value
- uint16: unsigned 16-bit value
- uint32: unsigned 32-bit value
- float32: float 32-bit value

Communication mechanism

The device enumerates as a USB device using an MCC driver.

Control transfers

- Control transfers to endpoint 0 are used to communicate with the device.
- The control transfer maximum buffer size is 64 bytes.
- The control transfers are vendor transfers.
- The command code is in the bRequest field.
    - SMBI uses the 0x80 data type for strings, and the 0x81 data type for RAW data, when supported.
- Parameters passed for OUT control transfers are sent during the data stage.
- The SMBI data stage returns RAW data, with LSB first.

  The DEV component message "DEV:DATATYPE=ENABLED" specifies whether raw data is enabled or disabled for returned data. When using "DEV:DATATYPE=ENABLED" the first byte indicates the data type returned. Refer to the RawData Notes for more information.
- The amount of data being transferred must be specified in the wLength field.
- If a data overrun occurs, the device stalls the BULK IN endpoint (EP_1) if the AISCAN STALL option is enabled.
- Bulk transfers to endpoint 2 OUT are used for analog output scans. Data is transmitted in two bytes per channel, with the 12-bit representation packed into the lower 16-bits and transmitted in RAW format, resulting in an output of 0-4.096 V. The device is preloaded with up to 1024 bytes. The device will ACK when it is filling. When the four internal 256-byte buffers are filled, the device will NAK until an analog output scan is started.

  If a data overrun occurs, the device will stall the BULK OUT endpoint (EP_2) if the AOSCAN STALL option is enabled.

See also

Message-based control transfers

Message-based control transfers

Command	Description	bRequest	Direction
Digital I/O Commands			
StringMessage	Send string messages to the device	0x80	In/Out
RawData	Return raw data from the device	0x81	In

StringMessage — SMBI string message

This command reads/writes the string message components of a device. The device stores the various components and parameters to provide read back.

Output arguments

string	char[64]; 64 character string including a NULL terminator.

Input arguments

None

Input response

string	char[64]; 64 character response string including a NULL terminator.

Notes

- Invalid output strings result in a STALL to the USB Control Transfer, and the input response returns "INVALID".

RawData — Raw data with SMBI

This command reads the RAW data from the last StringMessage sent to the device. Data is only returned if the StringMessage has matching RAW data; otherwise a 0 length packet is returned.

Output arguments

N/A – output is currently unsupported.

Input arguments

None

Input response

DATATYPE	0x03 – uint8 0x07 – uin16 0x09 – uint32 0x0A – float32 0xFF – Invalid
value	This depends upon the StringMessage sent, but will have the form of DATATYPE. Invalid StringMessage result in a 0 length packet, or if DATATYPE is set to ENABLE, one byte reading 0xFF.

Notes

- When using "DEV:DATATYPE=ENABLE" the input response will have the first byte set to the data type used. If "DISABLE", the data will not include the DATATYPE byte.

  If DATATYPE is ENABLE, a Valid SendMessage without a RAW return will only return a 0 length packet. Only invalid StringMessages will result in the return of 0xFF.

Analog input data format

When performing analog input operations with the "AI" component, the return data is LSB (Least Significant Bit) justified with a value of 0 as Min Scale.

Example: The bit table below shows 12-bits of data acquired using a typical DAQFlex-supported device installed with firmware version 2.03 or later.

A/D Converter Values

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x

where x the actual conversion value.

Note

Some devices return data as 12-bit MSB (Most Significant Bit) justified, and require conversion to the LSB justified data format shown above. Refer to the device-specific information for details.

Updating device firmware

Note: To determine the firmware version currently installed on a device, run the DAQFlexFWLoader.exe provided with the DAQFlex for Windows software. The firmware version installed on a device displays in the **Device firmware version** field.

Perform the following procedure to update the firmware that is installed on a message-based DAQ device:

1. Go to the Measurement Computing Firmware Updates page at www.mccdaq.com/fwupdate.
2. Select the firmware version to download and click the **Download** button. A download dialog automatically launches.
3. Save the firmware *.hex file to the FirmwareImages subdirectory of the DAQFlex installation directory ("`C:\Program Files\Measurement Computing\DAQFlex For Windows\FirmwareImages`" by default).
4. Connect the message-based DAQ device to your computer.
5. Run the DAQFlex Firmware Loader utility from `C:\Program Files\Measurement Computing\DAQFlex For Windows`, and run **DAQFlexFWLoader.exe**.
6. From the **Device** drop down list select the device in which to update the firmware.
7. Click the **Load Firmware** button to start installing the firmware.

The progress bar updates as the firmware is installed on the device.

Once the firmware is installed, the **Status** field displays "Firmware update completed", and the **Device firmware version** field displays the current firmware version installed on the device. Click the **X** in the upper right corner of the dialog to exit the DAQFlex Firmware Loader utility.

Programming and developing applications

The firmware architecture uses a String Message Based Interface (SMBI) to communicate with DAQFlex series hardware. A minimal number of device transactions is required. Communication with the device is accomplished through the USB driver (WinUsb, libusb, or other custom driver).

Refer to the Windows driver (WinUsb) or the Linux driver (libusb) for an example of how to communicate with DAQFlex series hardware. You can also use the DAQFlex software API source code as a guide for developing a driver to interface directly with the hardware.

Notes

- Experience with CONTROL transfers for asynchronous communications, and BULK transfers for synchronous communications is recommended.
- Only experienced USB programmers should attempt to write a driver for use with the DAQFlex firmware.

See also

SMBI data types

Communication mechanism

Message-based control transfers

Hardware reference

Select your DAQFlex-supported device below for the components and programming messages supported by the firmware.

- USB-2001-TC
- USB-7202
- USB-7204

USB-2001-TC

Component support

Component	Supported Property/Command	Set/Get	Supported Values
AI	CJC/format	Get	CJC/DEGC, CJC/DEGF, CJC/KELVIN
	OFFSET	Get	Floating point numeric
	SENSOR	Set/Get	TC/char
	SLOPE	Get	Floating point numeric
	STATUS	Get	BUSY, ERROR, READY
	VALUE	Get	Unsigned integer numeric
AI{ch}	RANGE	Set/Get	BIP73.125E-3V (±0.073125 volts) BIP146.25E-3V (±0.14625 volts)
DEV	FLASHLED	Set	0 - 255
	FWV	Get	Firmware version
	ID	Set/Get	Up to 57 characters
	MFGCAL	Get	yyyy-mm-dd HH:MM:SS
	MFGCAL{YEAR}	Get	Year as yyyy; 20xx
	MFGCAL{MONTH}	Get	Month as mm; 01 to 12
	MFGCAL{DAY}	Get	Day as dd; 01 to 31
	MFGCAL{HOUR}	Get	Hour as HH; 0 - 23
	MFGCAL{MINUTE}	Get	Minute as MM; 1 - 59
	MFGCAL{SECOND}	Get	Second as SS; 1 - 59
	MFGSER	Get	Up to 8 numeric characters
	RESET		DEFAULT, SYSTEM

Hardware features

- One analog input channel, numbered 0
- Supports thermocouple types B, E, J, K, N, R, S, and T
- Possible gain ranges:
  - ±146.25 mV (not calibrated)
  - ±73.125 mV
- 1024 bytes of nonvolatile FLASH program memory; used for storing configuration information, calibration data, and user data.

USB-7202

Component support

Component	Supported Property/Command	Set/Get	Supported Values
AI{ch}	OFFSET	Get	4-byte floating point numeric
	RANGE	Set/Get	BIP10V, BIP5V, BIP2V, BIP1V
	SLOPE	Get	4-byte floating point numeric
	VALUE	Get	Counts
AISCAN	DEBUG	Set	ENABLE, DISABLE
	EXTPACER	Set/Get	ENABLE/MASTER, ENABLE/SLAVE, DISABLE
	HIGHCHAN	Set/Get	0 - 7
	LOWCHAN	Set/Get	0 - 7
	RANGE	Set/Get	BIP10V, BIP5V, BIP2V, BIP1V
	SAMPLES	Set/Get	0 to N (0 = continuous scan; N = 32-bit)
	START		
	STATUS	Get	IDLE, RUNNING, OVERRUN, INTERRUPTED
	STOP		
	TRIG	Set/Get	ENABLE, DISABLE
	XFRMODE	Set/Get	BLOCKIO, SINGLEIO, BURSTIO
AISCAN{ch}	RANGE	Set	Sets all channels to a specified range
	RATE	Set/Get	0.596 to 50,000 Hz (1 channel)
AITRIG	TYPE	Set/Get	EDGE/RISING, EDGE/FALLING
CTR	START		
	STOP		
	VALUE	Set	0
		Get	0 – 4,294,967,295
DEV	DATATYPE	Set/Get	ENABLE, DISABLE
	FLASHLED	Set	0 - 255
	FWV	Get	MM.mm (M = major; m = minor)
	ID	Set/Get	Up to 57 characters
	MFGCAL	Get	yyyy-mm-dd HH:MM:SS
	MFGCAL{YEAR}	Get	Year as yyyy; 20xx
	MFGCAL{MONTH}	Get	Month as mm; 01 to 12
	MFGCAL{DAY}	Get	Day as dd; 01 to 31
	MFGCAL{HOUR}	Get	Hour as HH; 0 - 23
	MFGCAL{MINUTE}	Get	Minute as MM; 1 - 59
	MFGCAL{SECOND}	Get	Second as SS; 1 - 59
	MFGSER	Get	Up to 8 numeric characters
	RESET		DEFAULT, SYSTEM
DIO	DIR	Set	IN, OUT
		Get	0 – 255 (bit field: 0 = all output, 255 = all input)
	VALUE	Get	0 – 255 (port) 0 – 1 (bit)

Hardware features

- 8 analog input channels, numbered 0-7.
  - Gain ranges:

    ±10V

    ±5V

    ±2V

    ±1V
- 1 digital port.

  All bits are individually configurable as input or output.
- External trigger input
- External pacer input/output

  This feature allows multiple devices on a single USB to acquire synchronized samples. One master device is used to drive the signal. Additional devices must be configured as slave devices using the "AISCAN:EXTPACER=*value*" message. Value may be "ENABLE[/MASTER]", "ENABLE[/SLAVE]" or "DISABLE".
- 1024 bytes of nonvolatile EEPROM memory; used for storing configuration information, calibration data, and user data.
- *RATE* takes a float value.

  If the input scan rate requested is less than the slowest rate supported by the device, the device is set to the slowest rate supported by the device. If the input scan rate requested is greater than the fastest rate supported by the device, an error is generated.

USB-7204

Component support

Component	Supported Property/Command	Set/Get	Supported Values
AI	CHMODE	Set	SE, DIFF
	OFFSET	Get	4-byte floating point numeric
	SLOPE	Get	4-byte floating point numeric
	VALUE	Get	Counts
AI{ch}	RANGE	Set/Get	BIP20V, BIP10V, BIP5V, BIP4V, BIP2PT5V, BIP2V, BIP1PT25V, BIP1V
AISCAN	EXTPACER	Set/Get	ENABLE/MASTER, ENABLE/SLAVE, ENABLE/GSLAVE
	HIGHCHAN	Set/Get	0 - 7
	LOWCHAN	Set/Get	0 - 7
	QUEUE	Set/Get	ENABLE, DISABLE, RESET
	RATE	Set/Get	0.596 to 50,000 Hz (1 channel)
	RANGE	Get	0 - 15 (the number of elements in the queue)
		Set	BIP20V, BIP10V, BIP5V, BIP4V, BIP2PT5V, BIP2V, BIP1PT25V, BIP1V
	SAMPLES	Set/Get	0 to N (0 = continuous scan; N = 32-bit)
	START		
	STATUS	Get	IDLE, RUNNING, OVERRUN
	STOP		
	TRIG	Set/Get	ENABLE, DISABLE
	XFRMODE	Set/Get	BLOCKIO, SINGLEIO
AISCAN{ch}	RANGE	Set/Get	See the range values above.
AISCAN {element/ch}	RANGE	Set	Element: 0 - 15 Channel: 0-7 single-ended, 0-3 differential Range: See the range values above.
AITRIG	REARM	Set/Get	ENABLE, DISABLE
	TYPE	Set/Get	EDGE/RISING, EDGE/FALLING
AO	RANGE	Get	UNI4.096V
	VALUE	Set	0 to 4095
AOSCAN	HIGHCHAN	Set/Get	0 - 1
	LOWCHAN	Set/Get	0 - 1
	RATE	Set/Get	one channel: 1 to 10 kHz two channels: 1 to 5 kHz
	SAMPLES	Set/Get	0 to N (0 = continuous scan; N = 32-bit)
	START		
	STATUS	Get	IDLE, RUNNING, UNDERRUN
	STOP		
	TRIG	Set/Get	ENABLE, DISABLE
AOSCAN{ch}	RANGE	Get	UNI4.096V
CTR	START		
	STOP		
	VALUE	Set	0

		Get	0 – 4,294,967,295
DEV	DATATYPE	Set/Get	ENABLE, DISABLE
	FLASHLED	Set	0 - 255
	FWV	Get	MM.mm (M = major; m = minor)
	ID	Set/Get	Up to 57 characters
	MFGCAL	Get	yyyy-mm-dd HH:MM:SS
	MFGCAL{YEAR}	Get	Year as yyyy; 20xx
	MFGCAL{MONTH}	Get	Month as mm; 01 to 12
	MFGCAL{DAY}	Get	Day as dd; 01 to 31
	MFGCAL{HOUR}	Get	Hour as HH; 0 - 23
	MFGCAL{MINUTE}	Get	Minute as MM; 1 - 59
	MFGCAL{SECOND}	Get	Second as SS; 1 - 59
	MFGSER	Get	Up to 8 numeric characters
	RESET		DEFAULT, SYSTEM
DIO	DIR	Set/Get	IN, OUT (port-configurable)
	VALUE	Get	0 – 255 (port) 0 – 1 (bit)

Hardware features

- 8 analog input channels, numbered 0-7.

  Analog input mode is configurable for single-ended (8 channels) or differential (4 channels).

  - Gain ranges:

    ±20V (differential mode)

    ±10V (differential or single–ended mode)

    ±5V (differential mode)

    ±4V (differential mode)

    ±2.5V (differential mode)

    ±2V (differential mode)

    ±1.25V (differential mode)

    ±1V (differential mode)

- 2 analog output channels, numbered 0-1.

  Analog output range is 0-4.096 V, 1 mV per LSB

- 2 digital ports. Each port is individually configurable as input or output.
- External trigger input
- External pacer input/output

  This feature allows multiple devices to acquire synchronized samples. One master device is used to drive the signal. Additional devices must be configured as slave devices using the "AISCAN:EXTPACER=*value*" message. Value may be "ENABLE/MASTER," "ENABLE/SLAVE," or "ENABLE/GSLAVE."

  - When set to *ENABLE/SLAVE*, the first clock pulse after setting up the scan is ignored to ensure adequate setup time for the first conversion. Use this mode when the device is paced from a continuous clock source.
  - When set to *ENABLE/GSLAVE*, the first clock pulse after setting up the scan is held off to ensure adequate setup time for the first conversion. No pulses are ignored. Use this mode when the device is paced from another USB-7204.

- 1024 bytes of nonvolatile EEPROM memory; used for storing configuration information, calibration data, and user data.
- *RATE* takes a float value.

  If the input scan rate requested is less than the slowest rate supported by the device, the device is set to the slowest rate supported by the device. If the input scan rate requested is greater than the fastest rate supported by the device, the device is set to the maximum rate supported by the device.

Analog input data format

Note: This section applies to data acquired using firmware versions ≥2.03. For data acquired using firmware versions ≤2.02, refer to the data format shown below in Table 3.

When performing analog input operations with the "AI" component, the return data is LSB (Least Significant Bit) justified with a value of 0 as Min Scale.

When performing analog input operations with the "AISCAN" component, the differential return data is 12-bit MSB (Most Significant Bit) justified. Additional steps are required to convert single-ended data to 11- or 12-bit representation (LSB justified).

The bit tables below show the data for each operating mode, and lists the steps to convert the data to LSB justified.

Differential mode

The following bit table applies to data acquired using firmware ≥2.03.

Table 1. A/D Converter Values (differential mode)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	OVR

where:

x The actual conversion value.

OVR 0 = Valid conversion
1 = An analog over-range problem has occurred in the PGA; the conversion value may be invalid. **Note**: this bit can be ignored.

Converting differential data to 12-bit LSB justified

To convert differential data to 12-bit LSB justified, shift the upper 12-bits of data to the right by 4.

The adjusted data format is shown in the Analog input data format topic.

Single-ended mode

Table 2. A/D Converter Values (single-ended mode)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data	c	x	x	x	x	x	x	x	x	x	x	x	0	0	0	OVR

where:

c A checkbit for valid A/D transfer functions. When this bit is not set the value is 0.

x The actual conversion value.

OVR 0 = Valid conversion
1 = An analog over-range problem has occurred in the PGA; the conversion value may be invalid. **Note**: this bit can be ignored.

When bit 15 (c) is set or a value greater than 0x7FFF (32,767), you need to convert the data to LSB justified to obtain an 11-bit representation of the 12-bit data.

Converting single-ended data to 12-bit LSB justified

Converting data to LSB justified is required to obtain a 12-bit representation of 11-bit data. Perform the following steps to convert single-ended data to 12-bit LSB justified:

1. Check the value of bit 15 (c) to see if the value is greater than 0x7FFF.

   If the value is greater than 0x7FFF continue with step 2.

   If the value is not greater than 0x7FFF then the value is 0 and the conversion procedure is not required.
2. Mask the bits with 0x7FF0.
3. Shift the bits to the right by 3 for 12-bit data, or shift the bits to the right by 4 for 11-bit data.

The bit table below shows the 11-bit data represented as 12-bit data.

Table 3. 11-bit data represented as 12-bit LSB justified (bit 15 is set)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	0

where x is the actual conversion value.

Known issues

Firmware version	Description
≤2.02	Data overruns may occur or data integrity may be compromised when using AINSCAN to acquire data at rates above 37 kS/s. To resolve this issue, update the device firmware version to 2.03 or later. Refer to the Updating device firmware topic for instructions.