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: 4.0

September, 2010

SMBI data types

Special formatting characters are retuned 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.

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.

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.
- Refer to the topics in the Firmware Specification topic for information on the string-based messages, transfer methods, and commands that are supported by the DAQFlex firmware.

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
DEV	MFGSER	Get	Up to 8 numeric characters
	FWV	Get	Firmware version
	ID	Set/Get	Up to 56 characters
	MFGCAL	Get	yyyy-mm-dd HH:MM:SS
	MFGCAL{YEAR}	Get	Year as yyyy
	MFGCAL{MONTH}	Get	Month as mm
	MFGCAL{DAY}	Get	Day as dd
	MFGCAL{HOUR}	Get	0 - 59
	MFGCAL{MINUTE}	Get	0 - 59
	MFGCAL{SECOND}	Get	0
	FLASHLED		0 - 255
AI	RANGE{ch}	Set/Get	BIP73.125E-3V (±0.073125 volts) BIP146.25E-3V (±0.14625 volts)
	SENSOR	Set/Get	TC/B, TC/E, TC/J, TC/K, TC/N, TC/R, TC/S, TC/T
	CJC/format	Get	CJC/DEGC, CJC/DEGF, CJC/KELVIN
	STATUS	Get	BUSY, ERROR, READY
	OFFSET	Get	Floating point numeric
	SLOPE	Get	Floating point numeric
	VALUE	Get	Unsigned integer numeric

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
  - ±73.125 mV
- 512 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
DEV	MFGSER	Get	Up to 8 numeric characters
	FWV	Get	MM.mm (M = major; m = minor)
	ID	Set/Get	Up to 56 characters
	MFGCAL	Get	yyyy-mm-dd HH:MM:SS
	MFGCAL{YEAR}	Get	Year as yyyy
	MFGCAL{MONTH}	Get	Month as mm
	MFGCAL{DAY}	Get	Day as dd
	MFGCAL{HOUR}	Get	0 - 59
	MFGCAL{MINUTE}	Get	0 - 59
	MFGCAL{SECOND}	Get	0 - 59
	FLASHLED		0 - 255
	DATATYPE	Set/Get	ENABLE, DISABLE
	RESET		DEFAULT
AI	RANGE{ch}	Set/Get	BIP10V, BIP5V, BIP2V, BIP1V
	OFFSET	Get	4-byte floating point numeric
	SLOPE	Get	4-byte floating point numeric
	VALUE	Get	A/D range
	VALUE/format	Get	RAW, VOLTS
AISCAN	RANGE{ch}	Set/Get	BIP10V, BIP5V, BIP2V, BIP1V
	RANGE	Set	Sets all channels to specified range
	XFRMODE	Set/Get	BLOCKIO, SINGLEIO
	RATE	Set/Get	0.596 to 50,000 Hz (1 channel)
	SAMPLES	Set/Get	0 to N (0 = continuous scan; N = 32-bit)
	HIGHCHAN	Set/Get	0 - 7
	LOWCHAN	Set/Get	0 - 7 (must be ≤ HIGHCHAN)
	TRIG	Set/Get	ENABLE, DISABLE
	EXTPACER	Set/Get	ENABLE/MASTER, ENABLE/SLAVE, DISABLE
	DEBUG	Set/Get	ENABLE, DISABLE
	STATUS	Get	IDLE, RUNNING, OVERRUN
	START		
	STOP		
AITRIG	TYPE	Set/Get	EDGE/RISING, EDGE/FALLING
DIO	DIR	Set	IN, OUT
		Get	0 – 255 (bit field: 0 = all output, 255 = all input)
	VALUE	Get	0 – 255 (port) 0 – 1 (bit)
CTR	VALUE	Set/Get	0 (Set) 0 – 4,294,967,295 (Get)
	START		
	STOP		

Hardware features

- One digital port. All bits are individually configurable as input or output.
- Eight analog input channels, numbered 0 - 7.
- Possible gain ranges:
  - ±10V
  - ±5V
  - ±2V
  - ±1V
- 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, the device is set to the maximum rate supported by the device.

USB-7204

Component support

Component	Supported Property/Command	Set/Get	Supported Values
DEV	MFGSER	Get	Up to 8 numeric characters
	FWV	Get	MM.mm (M = major; m = minor)
	ID	Set/Get	Up to 56 characters
	MFGCAL	Get	yyyy-mm-dd HH:MM:SS
	MFGCAL{YEAR}	Get	Year as yyyy
	MFGCAL{MONTH}	Get	Month as mm
	MFGCAL{DAY}	Get	Day as dd
	MFGCAL{HOUR}	Get	0 - 59
	MFGCAL{MINUTE}	Get	0 - 59
	MFGCAL{SECOND}	Get	0 - 59
	FLASHLED		0 - 255
	DATATYPE	Set/Get	ENABLE, DISABLE
	RESET		DEFAULT
AI	RANGE{ch}	Set/Get	BIP20V, BIP10V, BIP5V, BIP4V, BIP2PT5V, BIP2V, BIP1PT25V, BIP1V
	CHMODE	Set	SE, DIFF
	OFFSET	Get	4-byte floating point numeric
	SLOPE	Get	4-byte floating point numeric
	VALUE	Get	A/D range
	VALUE/format	Get	RAW, VOLTS
AISCAN	RANGE	Get	0 - 15 (the number of elements in the queue)
	RANGE	Set	BIP20V, BIP10V, BIP5V, BIP4V, BIP2PT5V, BIP2V, BIP1PT25V, BIP1V
	RANGE{element/ch}	Set	Element: 0 - 15 Channel: 0-7 single-ended, 0-3 differential Range: See the range values above.
	RANGE{ch}	Set/Get	See the range values above.
	XFRMODE	Set/Get	BLOCKIO, SINGLEIO
	RATE	Set/Get	0.596 to 50,000 Hz (1 channel)
	SAMPLES	Set/Get	0 to N (0 = continuous scan; N = 32-bit)
	HIGHCHAN	Set/Get	0 - 7
	LOWCHAN	Set/Get	0 - 7 (must be ≤ HIGHCHAN)
	TRIG	Set/Get	ENABLE, DISABLE
	EXTPACER	Set/Get	ENABLE/MASTER, ENABLE/SLAVE, ENABLE/GSLAVE
	QUEUE	Set/Get	ENABLE, DISABLE, RESET
	DEBUG	Set/Get	ENABLE, DISABLE
	STATUS	Get	IDLE, RUNNING, OVERRUN
	START		
	STOP		
AITRIG	Type	Set/Get	EDGE/RISING, EDGE/FALLING
	REARM	Set/Get	ENABLE, DISABLE

AO	VALUE	Set	0 to 4095
	SCALE	Set/Get	ENABLE, DISABLE
DIO	DIR	Set/Get	IN, OUT (port-configurable)
	VALUE	Get	0 – 255 (port) 0 – 1 (bit)
CTR	VALUE	Set/Get	0 (Set) 0 – 4,294,967,295 (Get)
	START		
	STOP		

Hardware features

- Two digital ports. Each port is individually configurable as input or output.
- Eight analog input channels, numbered 0 - 7.
- Analog input mode is configurable for single-ended (eight channels) or differential (four channels).
- Possible 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)
- 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.