DI-188 Instrument

Command Protocol

Revised August 18, 2022

Table of Contents

Overview

Although DATAQ Instruments provides ready-to-run WinDaq software with DI-188 acquisition instrument, programmers will want the flexibility to integrate them in the context of their own application or add sensors to DI-188's data stream. To do so they want complete control over instruments hardware, which is accomplished by using the device at the protocol level. This white paper describes how protocol-level programming is implemented. We'll define the protocol command set and scan list architecture and finish with a description of the instrument ASCII and binary response formats.

DI-188 is an open-source firmware project. It allows the user to modify its functioning, such as to add various sensors to its data stream or change its features.

The project can be found in https://github.com/dataq-instruments/Arduino_WinDaq.

In repository, one can find examples to interface with 3-axis accelerometers, Temperature and Humidity, and Barometer sensors.

USB Interface Support

DI-188 model supports a USB interface called CDC mode (Communication Device Class) where it can appear as a COM port device to a computer, Windows or Linux. This approach is perhaps the most universal and easiest to manage. Since most instrument command and responses are ASCII, a standard terminal emulator (e.g. PuTTY, TeraTerm) can be used to experiment with the device.

Protocol Command Usage

The protocol employs an ASCII character command set that allows complete control of the instrument. All of the commands in the following table except legacy ones used by WinDaq must be terminated with a carriage return character (0xD) to be recognized by the instrument. Command arguments (if any) are also ASCII, and the command and each argument must be separated by a space character (0x20). All commands echo if the instrument is not scanning. Command arguments and responses are always in decimal unless an instrument is instructed to begin scanning in its binary output mode.

All commands echo if DI-188 is not scanning. Commands will not echo while scanning is active to prevent an interruption of the data stream. In this sense, the *start* command never echoes, and the *stop* command always echoes. In all the following descriptions and examples, references to command echoes assume that the instrument is not actively scanning.

Please note that DI-188's command buffer is small. To prevent command buffer overruns ensure that you do not send a new command without first receiving the previous command's echo.

Basic communication	
`info arg0`	Echoes the command and argument with additional information as defined by the argument
Measurement Configuration	
`encode arg0`	Defines device output coding
`eol arg0`	Defines EOL termination when outputting ASCII data
`slist arg0 arg1`	Defines scan list configuration
`rrate arg0`	Request or return a sample rate
`rgain`	Request available gain.
`ggrp`	Query the gain group configuration
`rchn arg0`	Request available channel info.
Scanning	
`start`	Start scanning
`stop`	Stop scanning
Calibration	
`readflash`	Returns system configuration stored in Flash
`Writeflash arg0`	Writes the modified version (scaling/offset) back to Flash
`offset arg0 arg1 arg2`	Change the offset for gain arg1 of channel arg0 to the value of arg2
`scale arg0 arg1 arg2`	Change the scale for gain arg1 of channel arg0 to the value of arg2
Windaq based legacy commands	
S commands	Instead of start and stop, WinDaq uses S1 and S0 to start and stop. Whenever S command is captured, the Windaq binary mode will be activated until scanning stop.
Other legacy commands Please refer to LegacyCommand in the source codes	To keep compatibility with Windaq, please leave LegacyCommand source unmodified

Info Commands

Basic Communication Commands	
ASCII Command	**Action**
`info 0`	Returns "info 0 DATAQ"
`info 1`	Returns device model number (e.g."info 1 188")
`info 2`	Returns firmware revision, 2 hex bytes (e.g. 65_{16} = 101_{10} for firmware revision 1.01)
`info 6`	Returns the instrument's unique serial number (left-most 8 digits only)

The protocol supports a number of basic command/response items that provide a simple means to ensure the integrity of the communication link between a program and DI-188. These commands elicit simple, yet useful responses from the instrument and should be employed as the programmer's first communication attempt. If these commands don't work with a functioning instrument then a problem exists in the communication chain and further programming efforts will be futile until resolved.
Responses to this set of commands include echoing the command, followed by a space (20_{16}), followed by the response, and ending with a carriage return (D_{16}). For example:

```
Command:    info 1            'what model is connected?
Response:   info 1 188        'command echo, plus connected model number
```

Measurement Configuration Commands

encode Command

The *encode* command defines the instrument's data stream output coding format:

encode Command*	
ASCII Command	**Action**
`encode 0`	Enables the binary output mode (DEFAULT)
`encode 1`	Enables the ASCII output mode.

*Refer to the binary and ASCII stream data format sections in this document for output formatting details

eol Command

The *eol* command defines a termination character the device uses for ASCII mode data output (see the *encode* command.) The *eol* command is valid only while in the ASCII data output mode (*encode* = 1.)

eol Command	
ASCII Command	**Action**
`eol 0`	Terminate ASCII data output with a carriage return (D_{16}) character (DEFAULT)
`eol 1`	Terminate ASCII mode output with a line feed character (A_{16}.)
`eol 2`	Terminate ASCII mode output with carriage return and line feed ($D_{16}A_{16}$) characters.

rchn Command

rchn Command (Required by WinDaq)	
ASCII Command	**Action**
`rchn`	Returns the number of channels available to this device.
`rchn arg0`	Returns the configuration of channel arg0. For example “Volt, -10, 10”

rgain Command

rgain Command (Required by WinDaq)	
ASCII	**Action**
`rgain`	Returns the available gains for each channel in a 32-bit integer, with each bit indicates an available gain, where all supported gains are listed as {1,2,4,5,8,10,16,20,32,40,50,64,80,100,128,200,256,400,500,512,800,1000,1024,2000,2048,4000,5000,10000,20000,50000,100000,1000000} For example, if gain 1,2,4 are available for channel 1 and 2, and gain 1,5,10 are available for the channel 3 and 4, the return will be 7,7, 41, 41 WinDaq can list up to 12 gains in the “Channel settings” dialog box for any channel. Gain index 0 will select the first gain in the list.

ggrp Command

ggrp Command (Required by WinDaq)	
ASCII Command	**Action**
`ggrp`	Returns gain group configuration in a 16-bit integer, with each bit represents a channel. When a higher bit repeats the value in lower bit position, the channel represented by the higher bit shares the same gain as the

	channel represented by the lower bit, e.g. when the gain of any channel within a gain group changes, all channels within the group will take the same gain. For example: If gains are independent for each channel, it will return 21845, or 0101010101010101_b If a device has four independent PGA channels and four tri-axis accelerometers that share the same gain, changing the gain in channel 4, 5, or 6 causes WinDaq to set the same gain index for all three of those channels, and adjust their full-scale ranges accordingly, whether or not they are enabled

slist Command

Instruments employ a scan list approach to data acquisition. A scan list is an internal schedule (or list) of channels to be sampled in a defined order. It is important to note that a scan list defines only the type and order in which data is to be sampled, not the sampled data itself. Since instruments may support different measurement characteristics, their scan list architectures will vary.

During general-purpose use each entry in the scan list is represented by a 16-bit number, which is defined in detail in the *Scan List Word Definitions* tables below. Writing any value to the first position of the scan list automatically resets the slist member count to 1. This count increases by 1 each time a new member is added to the list, which must be filled from lowest to highest positions.

The *slist* command along with two arguments separated by a space character is used to configure the scan list:

slist offset config

offset defines the index within the scan list and can range from 0 to 15 to address up to sixteen possible positions. *config* is the 16-bit configuration parameter as defined in table *Scan List Word Definitions tables*.

For example, the command *slist 0 1* sent to model DI-188 configures the first position of the scan list to specify data from the channel 2. Continuing with the DI-188 and assuming that we wish to sample analog channels 3 and 4, the following scan list configuration would work:

```
slist 0 1
slist 1 2
slist 2 3
```

Other considerations:

- Any analog, digital input, rate, or counter channel may appear in the scan list only once.
- *slist* positions must be defined sequentially beginning with position 0.
- To be consistent with general programming standards, analog channel numbers begin with 0 instead of 1 as indicated the product labels.

DI-188 Scan List Definition

DI-188 Default Scan List Word Definitions																
Function	Bit Position															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Analog In, Channel 0	Unused bits =0				Select one of the 12 gains if available				Unused bits =0				0	0	0	0
Analog In, Channel 1													0	0	0	1
Analog In, Channel 2													0	0	1	0
Analog In, Channel 3													0	0	1	1
Analog In, Channels	0	0	0	0					0	0	0	0	x	x	x	x

System Configuration Commands

DI-188 stores its system configuration in its flash memory, which can last about 10K write cycles, each time a new firmware is uploaded, all its contents will be erased. Please see the source codes for detail.

The most common parameters are offsets and scales for ADC channels.

When ADC raw data is retrieved, it goes through digital calibration process:

ADCOutput=(ADCRawReading+offset)*scale/16384

One can modify default/nominal offset and scale factors to better match his or her instruments.

offset Command

The offset of DI-188 digital calibration can be modified via offset command.

writeflash Command	
ASCII Command	**Action**
`offset arg0 arg1 arg2`	Assign offset (arg2) to gain (arg1) of channel (arg0). If default scale/offset needs to be modified, one should write 0 to offset and 16384 to scale first before calculating the real offset and scale ADCOutput=(ADCRawReading+offset)*scale/16384

scale Command

The scale of DI-188 digital calibration can be modified via scale commands.

writeflash Command	
ASCII Command	**Action**
`scale arg0 arg1 arg2`	Assign scale (arg2) to gain (arg1) of channel (arg0). If default scale/offset needs to be modified, one should write 0 to offset and 16384 to scale first before calculating the real offset and scale ADCOutput=(ADCRawReading+offset)*scale/16384

readflash Command

readflash Command	
ASCII Command	**Action**
`readflash`	Retrieve system configuration block from flash memory and store in RAM. Please refer to source code for detail

writeflash Command

writeflash Command	
ASCII Command	**Action**
`writeflash arg0`	When arg0 = -1, writes system configuration block store in RAM back to flash memory, which can last about 10K write cycles. Please refer to source code for detail

Scanning Commands

rrate Command

Command *rrate* requestes or returns sample rate per channel.

rrate Command	
ASCII Command	**Action**
`rrate arg0`	Request sample rate arg0 per channel
`rrate`	Returns the actual sample rate per channel under the current configuration

start Command

Since the *start* command immediately initiates scanning, the command is never echoed:

Start Command Modes	
ASCII Command	**Action**
`start`	Begin scanning: The instrument begins scanning the channels enabled in its scan list through the *slist* command at a rate defined by *rrate* commands depending upon output coding defined by the *encode* command.

```
Command:    start    'begin scanning
Response:            'never echoes
```

stop Command

The protocol's *stop* command terminates scanning. Since the *stop* command terminates scanning, it is always echoed.

```
Command:    stop     'stop scanning
Response:   stop     'always echoes
```

Binary Stream Output Formats

DI-188 outputs two different binary data formats, standard 16-bit binary format and Windaq-based binary format, both are designed for a fast sample rate. When the response mode is binary and the *start* command is received data begins issuing from the device in an order and rate defined by the previously configured scan list and sample rate. When the end of data defined by the last scan list element is reached

the instrument wraps back to data represented by the first scan list element and repeats. The result is an instrument that sends a continuous binary data stream that stops only after receiving the *stop* command.

Standard Binary Coding Notes

In this mode, DI-188 transmits every analog channel conversion in the form of a signed, 16-bit Two's complement value, without any sync bit. This mode is meant to be used in CDC mode only since USB communication is reliable.

WinDaq Binary Coding Notes

WinDaq/188 supports both true COM and CDC communication, thus sync bits are employed to guarantee data integrity, which is especially important for COM communication. Every analog reading is returned as 14-bit left justified (if ADC resolution is lower than 14-bit) Two's complement data, with two lowest bit serving as sync bits.

WinDaq Binary Data Stream Examples

Physical Channel	Word Count	Byte Count	B7	B6	B5	B4	B3	B2	B1	B0 (Sync)
1	1	1	AD6	AD5	AD4	AD3	AD2	AD1	AD0	0
		2	AD 13	AD 12	AD 11	AD 10	AD9	AD8	AD7	1
2	2	3	AD6	AD5	AD4	AD3	AD2	AD1	AD0	1
		4	AD 13	AD 12	AD 11	AD 10	AD9	AD8	AD7	1
3	3	5	AD6	AD5	AD4	AD3	AD2	AD1	AD0	1
		6	AD 13	AD 12	AD 11	AD 10	AD9	AD8	AD7	1
4	4	7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	1
		8	AD 13	AD 12	AD 11	AD 10	AD9	AD8	AD7	1

Channels 2 & 4 enabled:										
2	1	1	AD6	AD5	AD4	AD3	AD2	AD1	AD0	0
		2	AD 13	AD 12	AD 11	AD 10	AD9	AD8	AD7	1
4	2	3	AD6	AD5	AD4	AD3	AD2	AD1	AD0	1
		4	AD 13	AD 12	AD 11	AD 10	AD9	AD8	AD7	1

Only Channel 2 enabled:										
2	1	1	AD6	AD5	AD4	AD3	AD2	AD1	AD0	0
		2	AD 13	AD 12	AD 11	AD 10	AD9	AD8	AD7	1

ASCII Stream Output Format

When enabled with the *encode* commands, DI-188 generates a comma-separated value result with each line of the output terminated with character as defined by the *eol* command. Channel output order is defined by the scan list, and values are presented as floating point numbers scaled in volts. For example, if the following two assignments are made in the scan list using the *slist* command and *eol* command value = 1:

Position 0 = analog channel 1

Position 1 = analog channel 2

this CSV output results after issuing the *start* command:

```
(analog channel 1 value),(analog channel 2 value)<cr>
(analog channel 1 value),(analog channel 2 value)<cr>
```

```
(analog channel 1 value),(analog channel 2 value)<cr>
                         .
                         .
                         .
(analog channel 1 value),(analog channel 2 value)<cr>
```

Document Revisions

Revision	Date	Description
		Release revision, August 18, 2022