
openDAQ Documentation

Release 1.4.0

Jose R. Blanco, Juan Menendez

May 12, 2017

Contents

1	Serial protocol	3
1.1	Regular format	3
1.2	Stream format	4
2	Serial commands	5
2.1	List of commands	5
2.2	AIN	6
2.3	AINALL	6
2.4	AINCFG	7
2.5	BURSTCREATE	7
2.6	CAPTUREINIT	7
2.7	CAPTURESTOP	8
2.8	CHANNELCFG	8
2.9	CHANNELDESTROY	9
2.10	CHANNELFLUSH	9
2.11	CHANNELSETUP	9
2.12	COUNTERINIT	9
2.13	EEPROMREAD	10
2.14	EEPROMWRITE	10
2.15	ENABLECRC	10
2.16	ENCODERINIT	11
2.17	ENCODERSTOP	11
2.18	EXTERNALCREATE	11
2.19	GETCALIB	12
2.20	GETCAPTURE	12
2.21	GETCOUNTER	12
2.22	GETENCODER	13
2.23	IDCONFIG	13
2.24	LEDW	14
2.25	NAK	14
2.26	PIO	14
2.27	PIODIR	15
2.28	PORT	15
2.29	PORTDIR	16
2.30	PWMDUTY	16
2.31	PWMINIT	17

2.32	PWMSTOP	17
2.33	RESET	17
2.34	RESETCALIB	17
2.35	SETCALIB	18
2.36	SETDAC	18
2.37	SIGNALLOAD	18
2.38	SPISWCONFIG	19
2.39	SPISWSETUP	19
2.40	SPISWTRANSFER	20
2.41	STREAMCREATE	20
2.42	STREAMDATA	20
2.43	STREAMSTART	21
2.44	STREAMSTOP	21
2.45	TRIGGERSETUP	21

Contents:

CHAPTER 1

Serial protocol

For reasons of compatibility with Arduino platform, USB handler uses a VCP (Virtual COM Port) configuration. Thus, when the device is connected to a computer, a new serial port is created, and it can be accessed as any other serial port. In order to communicate with openDAQ, the following settings must be configured in the terminal program that you are using:

- Rate: 115200 baud
- Data Bits: 8
- Parity: None
- Stop Bits: 1
- Flow Control: None

Following is a description of openDAQ's low level protocol. Device communicates with host computer by means of a binary serial protocol. There are two kinds of commands used by openDAQ:

- *Regular format*, used for most of the commands.
- *Stream format*, used by the *STREAMDATA* and *STREAMSTOP* packets when the device is running in stream mode.

Regular format

Regular openDAQ commands (command-response mode) use a minimum of 4 bytes to transmit a message to the device. Maximum allowed length for these packets is 64 bytes.

First two bytes consist on a 16 bit checksum or CRC. This checksum is calculated doing the sum of all other bytes in the packet into a 16 bit integer. Then, the result is complemented.

Be aware that some commands transmit 16 bit integers, and they do it in **big-endian** format (higher bytes first), while Arduino stores variables in its memory in **little-endian** format (lower bytes first).

Device will react to all these commands with a response with the same command number and different fields of data depending on the command.

See the *List of commands* for a complete description of each one.

Stream format

Stream packets differ from regular commands because the device sends them without being asked to do it, and no further response from the host computer is expected.

This format is only used by the *STREAMDATA* and *STREAMSTOP* packets.

When openDAQ is performing one or more simultaneous experiments, recorded data should be transmitted as fast as possible in order not to saturate limited internal memory buffers. The device can't wait for the computer to ask for data, but instead it will send that data in packets as fast as possible. In order to keep synchronization between openDAQ and the host computer, this packets use a special byte (0x7E) for marking the start of a packet. No other byte in the packet must have this same value, so we have to *escape* this value. When another 0x7E must be transmitted inside the packet, the openDAQ will substitute it by 0x7D 0x5E. In the same way, the byte 0x7D will be transmitted as 0x7D 0x5D.

CHAPTER 2

Serial commands

List of commands

Name	Number	Description
<i>AIN</i>	1	Read ADC with current settings
<i>AINALL</i>	4	Read all analog inputs
<i>AINCFG</i>	2	Read ADC after configuring analog settings
<i>BURSTCREATE</i>	21	Create a burst experiment
<i>CAPTUREINIT</i>	14	Initialize the capture mode
<i>CAPTURESTOP</i>	15	Disable the the capture mode
<i>CHANNELCFG</i>	22	Configure an experiment
<i>CHANNELDESTROY</i>	57	Delete an experiment
<i>CHANNELFLUSH</i>	45	Reset the data buffer of an experiment
<i>CHANNELSETUP</i>	32	Setup an experiment
<i>COUNTERINIT</i>	41	Initialize the edge counter
<i>EEPROMREAD</i>	31	Read a byte from the EEPROM at a given position
<i>EEPROMWRITE</i>	30	Write a byte in the EEPROM at a given position
<i>ENABLECRC</i>	55	Enable/disable cyclic redundancy check
<i>ENCODERINIT</i>	50	Initialize the encoder mode
<i>ENCODERSTOP</i>	51	Disable the the encoder mode
<i>EXTERNALCREATE</i>	20	Create an external experiment
<i>GETCALIB</i>	36	Read a calibration register
<i>GETCAPTURE</i>	16	Get current period length
<i>GETCOUNTER</i>	42	Get current counter value
<i>GETENCODER</i>	52	Get current encoder position
<i>IDCONFIG</i>	39	Read device information
<i>LEDW</i>	18	Set LED color
<i>NAK</i>	160	Invalid command (response only)
<i>PIO</i>	3	Read/write PIO value
<i>PIODIR</i>	5	Read/write PIO direction
Continued on next page		

Table 2.1 – continued from previous page

Name	Number	Description
<i>PORT</i>	7	Read/write all PIOs
<i>PORTDIR</i>	9	Read/write all PIO directions
<i>PWMDUTY</i>	12	Configure PWM duty cycle
<i>PWMINIT</i>	10	Initialize PWM
<i>PWMSTOP</i>	11	Disable the PWM
<i>RESET</i>	27	Restart the device
<i>RESETCALIB</i>	38	Reset a calibration register
<i>SETCALIB</i>	37	Write a calibration register
<i>SETDAC</i>	13	Set DAC output voltage
<i>SIGNALLOAD</i>	23	Load an array of values to preload DAC output
<i>SPISWCONFIG</i>	26	Configure bit-bang SPI (clock properties)
<i>SPISWSETUP</i>	28	Configure bit-bang SPI (clock properties)
<i>SPISWTRANSFER</i>	29	Transfer SPI data (send and receive)
<i>STREAMCREATE</i>	19	Create a stream experiment
<i>STREAMDATA</i>	25	This packet transmits the data captured by a running experiment (stream mode only)
<i>STREAMSTART</i>	64	Start an automated experiment
<i>STREAMSTOP</i>	80	This packet marks the end of a running experiment (stream mode only)
<i>TRIGGERSETUP</i>	33	Setup a trigger experiment

AIN

Read ADC with current settings

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	1	
3	1	Packet size	0	Number of bytes, excluding the header

Response:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	1	
3	1	Packet size	16	Number of bytes, excluding the header
4	16	Value		Raw value of analog input

AINALL

Read all analog inputs

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	4	
3	1	Packet size	2	Number of bytes, excluding the header
4	1	Samples	1-255	Number of averaged samples per measure
5	1	Gain index	0-4	

Response:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	4	
3	1	Packet size	16	Number of bytes, excluding the header
4	16	Values		Raw value of all analog inputs

AINCFG

Read ADC after configuring analog settings

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	2	
3	1	Packet size	4	Number of bytes, excluding the header
4	1	Positive input		Positive/Single-ended analog input
5	1	Negative input		Negative analog input
6	1	Gain index	0-4	
7	1	Samples	1-255	Number of averaged samples per measure

Response:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	2	
3	1	Packet size	2	Number of bytes, excluding the header
4	2	Read value		ADC response

BURSTCREATE

Create a burst experiment

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	21	
3	1	Packet size	2	Number of bytes, excluding the header
4	2	Period	100-65535	Period in microseconds

Response:

Same as command

CAPTUREINIT

Initialize the capture mode

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	14	
3	1	Packet size	4	Number of bytes, excluding the header
4	4	Period		Approximate period of the wave, in microseconds (int32)

Response:

Same as command

CAPTURESTOP

Disable the the capture mode

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	15	
3	1	Packet size	0	Number of bytes, excluding the header

Response:

Same as command

CHANNELCFG

Configure an experiment

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	22	
3	1	Packet size	6	Number of bytes, excluding the header
4	1	Number	1-4	DataChannel number
5	1	Mode	0-5	0=ANALOG_INPUT, 1=ANALOG_OUTPUT, 2=DIGITAL_INPUT, 3=DIGITAL_OUTPUT, 4=COUNTER_INPUT, 5=CAPTURE_INPUT
6	1	Positive input		Positive/Single-ended analog input
7	1	Negative input		Negative analog input
8	1	Gain index	0-4	
9	1	Samples	1-255	Number of averaged samples per measure

Response:

Same as command

CHANNELDESTROY

Delete an experiment

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	57	
3	1	Packet size	1	Number of bytes, excluding the header
4	1	Number	0-4	DataChannel number. 0: delete all experiments

Response:

Same as command

CHANNELFLUSH

Reset the data buffer of an experiment

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	45	
3	1	Packet size	1	Number of bytes, excluding the header
4	1	Number	0-4	DataChannel number. 0: flush all data buffers

Response:

Same as command

CHANNELSETUP

Setup an experiment

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	32	
3	1	Packet size	4	Number of bytes, excluding the header
4	1	Number	1-4	DataChannel number
5	2	Points	0-65535	Number of points. 0: continuous acquisition
7	1	Repetition mode	0, 1	0: continuous, 1: run once

Response:

Same as command

COUNTERINIT

Initialize the edge counter

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	41	
3	1	Packet size	1	Number of bytes, excluding the header
4	1	Edge	0, 1	Which edge increments the counter: 0=H->L, 1=L->H

Response:

Same as command

EEPROMREAD

Read a byte from the EEPROM at a given position

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	31	
3	1	Packet size	1	Number of bytes, excluding the header
4	1	Address	16-2000	Memory address

Response:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	31	
3	1	Packet size	2	Number of bytes, excluding the header
4	1	Address		Memory address
5	1	Value		Data value

EEPROMWRITE

Write a byte in the EEPROM at a given position

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	30	
3	1	Packet size	2	Number of bytes, excluding the header
4	1	Address		Memory address
5	1	Value		Data value

Response:

Same as command

ENABLECRC

Enable/disable cyclic redundancy check

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	55	
3	1	Packet size	1	Number of bytes, excluding the header
4	1	Number	0-4	0: disabled, 1: enabled

Response:

Same as command

ENCODERINIT

Initialize the encoder mode

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	50	
3	1	Packet size	4	Number of bytes, excluding the header
4	4	Resolution		Maximum number of encoder edges (int32)

Response:

Same as command

ENCODERSTOP

Disable the the encoder mode

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	51	
3	1	Packet size	0	Number of bytes, excluding the header

Response:

Same as command

EXTERNALCREATE

Create an external experiment

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	20	
3	1	Packet size	2	Number of bytes, excluding the header
4	1	Number	1-4	DataChannel number
5	1	Edge	0, 1	

Response:

Same as command

GETCALIB

Read a calibration register

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	36	
3	1	Packet size	1	Number of bytes, excluding the header
4	1	Address		Calibration register address

Response:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	36	
3	1	Packet size	5	Number of bytes, excluding the header
4	1	Address		Calibration register address
5	2	Gain	-32768 to 32767	Gain calibration
7	2	Offset	-32768 to 32767	Offset calibration

GETCAPTURE

Get current period length

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	16	
3	1	Packet size	1	Number of bytes, excluding the header
4	1	Edge	0, 1, 2	0: low cycle, 1: high cycle, 2: full period

Response:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	16	
3	1	Packet size	5	Number of bytes, excluding the header
4	1	Edge	0, 1, 2	0: low cycle, 1: high cycle, 2: full period
5	4	Value		Period of the wave, in microseconds (int32)

GETCOUNTER

Get current counter value

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	42	
3	1	Packet size	1	Number of bytes, excluding the header
4	1	Reset count	0, 1	Reset the counter after measuring

Response:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	42	
3	1	Packet size	4	Number of bytes, excluding the header
4	4	Count		Number of counted edges (int32)

GETENCODER

Get current encoder position

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	52	
3	1	Packet size	0	Number of bytes, excluding the header

Response:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	52	
3	1	Packet size	4	Number of bytes, excluding the header
4	4	Position		Current encoder value (int32)

IDCONFIG

Read device information

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	39	
3	1	Packet size	0	Number of bytes, excluding the header

Response:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	39	
3	1	Packet size	4	Number of bytes, excluding the header
4	1	Hardware version		Hardware version
5	1	Firmware version		Firmware version
6	2	Serial number	0-65535	Serial number

LEDW

Set LED color

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	18	
3	1	Packet size	2	Number of bytes, excluding the header
4	1	LED color	0-3	0: off, 1: green, 2: red, 3: orange
5	1	LED number	0	LED number (not used)

Response:

Same as command

NAK

Invalid command (response only)

Response:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	160	
3	1	Packet size	0	Number of bytes, excluding the header

PIO

Read/write PIO value

Command (read):

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	3	
3	1	Packet size	1	Number of bytes, excluding the header
4	1	PIO number	1-6	

Command (write):

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	3	
3	1	Packet size	2	Number of bytes, excluding the header
4	1	PIO number	1-6	
5	1	Value	0, 1	PIO value

Response:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	3	
3	1	Packet size	2	Number of bytes, excluding the header
4	1	PIO number	1-6	
5	1	Value	0, 1	PIO value

PIODIR

Read/write PIO direction

Command (read):

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	5	
3	1	Packet size	1	Number of bytes, excluding the header
4	1	PIO number	1-6	

Command (write):

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	5	
3	1	Packet size	2	Number of bytes, excluding the header
4	1	PIO number	1-6	
5	1	Direction	0, 1	PIO direction: 0=input, 1=output

Response:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	5	
3	1	Packet size	2	Number of bytes, excluding the header
4	1	PIO number	1-6	
5	1	Direction	0, 1	PIO direction: 0=input, 1=output

PORT

Read/write all PIOs

Command (read):

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	7	
3	1	Packet size	0	Number of bytes, excluding the header

Command (write):

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	7	
3	1	Packet size	1	Number of bytes, excluding the header
4	1	Value	0, 1	Value of all PIOs

Response:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	7	
3	1	Packet size	1	Number of bytes, excluding the header
4	1	Value	0, 1	Value of all PIOs

PORTDIR

Read/write all PIO directions

Command (read):

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	9	
3	1	Packet size	1	Number of bytes, excluding the header
4	1	PIO number	1-6	

Command (write):

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	9	
3	1	Packet size	2	Number of bytes, excluding the header
4	1	PIO number	1-6	
5	1	Directions	0, 1	Directions of all PIOs: 0=input 1=output

Response:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	9	
3	1	Packet size	2	Number of bytes, excluding the header
4	1	PIO number	1-6	
5	1	Directions	0, 1	Directions of all PIOs: 0=input 1=output

PWMDUTY

Configure PWM duty cycle

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	12	
3	1	Packet size	2	Number of bytes, excluding the header
4	2	Duty cycle	0-1023	High time of the signal: 0=always low, 1023=always high

Response:

Same as command

PWMINIT

Initialize PWM

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	10	
3	1	Packet size	4	Number of bytes, excluding the header
4	2	Frequency	0-65535	Frequency of the signal, in microseconds
6	2	Duty cycle	0-1023	High time of the signal: 0=always low, 1023=always high

Response:

Same as command

PWMSTOP

Disable the PWM

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	11	
3	1	Packet size	0	Number of bytes, excluding the header

Response:

Same as command

RESET

Restart the device

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	27	
3	1	Packet size	0	Number of bytes, excluding the header

Response:

Same as command

RESETCALIB

Reset a calibration register

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	38	
3	1	Packet size	1	Number of bytes, excluding the header
4	1	Address		Calibration register address

Response:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	38	
3	1	Packet size	5	Number of bytes, excluding the header
4	1	Address		Calibration register address
5	2	Gain	-32768 to 32767	Gain calibration
7	2	Offset	-32768 to 32767	Offset calibration

SETCALIB

Write a calibration register

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	37	
3	1	Packet size	5	Number of bytes, excluding the header
4	1	Address		Calibration register address
5	2	Gain	-32768 to 32767	Gain calibration
7	2	Offset	-32768 to 32767	Offset calibration

Response:

Same as command

SETDAC

Set DAC output voltage

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	13	
3	1	Packet size	2	Number of bytes, excluding the header
4	2	Value		Raw DAC value. 16 bits, signed

Response:

Same as command

SIGNALLOAD

Load an array of values to preload DAC output

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	23	
3	1	Packet size	4	Number of bytes, excluding the header
4	2	Number of samples	1-400	The packet size depends of this number of samples
6	2	Sample list		List of samples. Each one is a signed int16

Response:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	23	
3	1	Packet size	2	Number of bytes, excluding the header
4	2	Number of samples	1-400	Number of uploaded samples

SPISWCONFIG

Configure bit-bang SPI (clock properties)

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	26	
3	1	Packet size	2	Number of bytes, excluding the header
4	1	CPOL	0, 1	Clock polarity: clock pin state when inactive
5	1	CPHA	0, 1	Clock phase (reading edge): 0=leading, 1=trailing

Response:

Same as command

SPISWSETUP

Configure bit-bang SPI (clock properties)

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	28	
3	1	Packet size	3	Number of bytes, excluding the header
4	1	BBCK pin	1-6	Clock pin for bit-bang SPI transfer
5	1	BBMOSI pin	1-6	MOSI pin
6	1	BBMISO pin	1-6	MISO pin

Response:

Same as command

SPISWTRANSFER

Transfer SPI data (send and receive)

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	29	
3	1	Packet size	1	Number of bytes, excluding the header
4	1	Data to send		Array of bytes to transfer

Response:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	29	
3	1	Packet size	1	Number of bytes, excluding the header
4	1	Received data		Array of received bytes

STREAMCREATE

Create a stream experiment

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	19	
3	1	Packet size	3	Number of bytes, excluding the header
4	1	Number	1-4	DataChannel number
5	2	Period	1-65535	Reading period in microseconds

Response:

Same as command

STREAMDATA

This packet transmits the data captured by a running experiment (stream mode only)

Command:

This packet is always sent by the openDAQ

Response:

Position	Size	Name	Value	Description
0	1	Init byte	0x7e	
1	2	Unused		
3	1	Command number	25	
4	1	Packet size	6	Number of bytes, excluding the header
5	1	Number	1-4	DataChannel number
6	1	Positive input		Positive/Single-ended analog input
7	1	Negative input		Negative analog input
8	1	Gain index	0-4	
9	2	Sample list		List of captured samples. Each one is a signed int16

STREAMSTART

Start an automated experiment

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	64	
3	1	Packet size	0	Number of bytes, excluding the header

Response:

Same as command

STREAMSTOP

This packet marks the end of a running experiment (stream mode only)

Command:

This packet is always sent by the openDAQ

Response:

Position	Size	Name	Value	Description
0	1	Init byte	0x7e	
1	2	Unused		
3	1	Command number	80	
4	1	Packet size	1	Number of bytes, excluding the header
5	1	Number	1-4	DataChannel number

TRIGGERSETUP

Setup a trigger experiment

Command:

Position	Size	Name	Value	Description
0	2	CRC		Cyclic redundancy check
2	1	Command number	33	
3	1	Packet size	4	Number of bytes, excluding the header
4	1	Number	1-4	DataChannel number
5	1	Trigger mode	1-5	SW_TRG 0->Software trigger (run on start) DIN1_TRG 1->Digital triggers, DIN2_TRG 2, DIN3_TRG 3, DIN4_TRG 4, DIN5_TRG 5, DIN6_TRG 6, ABIG_TRG 10->Analog triggers (use current channel configuration: chp, chm, gain), ASML_TRG 20
6	2	Trigger value	1-65535	

Response:

Same as command