Instruments That Advance The Art

microDXP

RS-232 Communications Specification

Version 3.40
August 27, 2024

microDXP Hardware Revision: H (8) & J (10)

XIA LLC
2744 East 11th Street STE H2
Oakland, CA 94601 USA
Email: support@xia.com
Tel: (510) 401-5760; Fax: (510) 401-5761
http://www.xia.com/

Command Structure

The general structure for commands and responses is as follows:

`[Esc][Command][Ndata (2 bytes)][data1]…[dataN][XOR CS]`

Where:

`[Esc]`	Escape (ASCII 0x1B) as a command start byte
`[Command]`	Single byte for command number, allowing up to 255 commands. See the tables below for command definitions.
`[Ndata]`	Number of data bytes to follow. Two bytes, low byte first.
`[XOR CS]`	Exclusive-or checksum (bitwise XOR of all bytes except for the initial `[Esc]`). If the checksum is incorrect, an error response is returned.

The format of responses echo the format of commands; that is, they start with the `[Esc]` character and pass back the command # to which it is responding, followed by appropriate data and checksum. The first data byte of all responses is the return status, which is zero for a successful command. In case of an error, only the error byte is returned – no other data bytes are sent. Please note that it is necessary to receive the entire response from one command before sending the next command, as there is no queuing of commands in the microDXP communications processor.

Commands Supported

This specification pertains to the current microDXP hardware offerings: the Xilinx-based RevH first released in 2015, and the Efinix-based RevJ first released in 2022. For older RevG hardware, please refer to version 3.2 of this manual.

Run Control and Readout

Command	***Meaning***	***Response data***
0x00: Start Run	Enable data taking. Ndata = 1 [data1] = Clear MCA; 1: new run, 0: resume run	Returns the run number. Ndata = 3 [data1] = Return status (0: OK) [data2,data3] = Run number (low byte first)
0x01: End Run	Stop data taking. Ndata = 0	Ndata = 1 [data1] = Return status (0: OK) 0: OK, 1: error
0x02: Read MCA	Read MCA spectrum. Ndata = 5 [data1] = First MCA bin (low byte) [data2] = First MCA bin (high byte) [data3] = # of bins (low byte) [data4] = # of bins (high byte) [data5] = bytes per bin (1, 2 or 3)	MCA data, with the specified number of bytes per MCA bin. Ndata =1+ (Number of bins)*(bytes per bin) [data1] = Return status (0: OK) [data2] = low byte of first MCA bin … [dataN] = highest byte of highest MCA bin

Command	***Meaning***	***Response data***
0x04: Read MultiSCA	Read number of events in defined SCA regions. Ndata = 0.	MultiSCA data; 4 bytes per SCA Ndata = 2 + 4*NUMSCA (max 66) [data1] = Return status (0: OK) [data2] = NUMSCA [data3] = low byte of SCA0 … [dataN] = highest byte of last SCA
0x06: Read Run Statistics	Read real time, livetime, input events, and output events. Ndata = 0 or 1 Return data depends on Statistics readout mode: short readout mode omits the under- and overflows, while long readout mode includes everything. [data1]: Readout mode 0: short, 1: long. If omitted, the short readout mode is used.	Returns run statistics – Livetime, realtime, input events (fastpeaks), output events for short readout mode; long readout mode adds under- and overflows. Ndata = 21 for short mode, 29 for long mode. [data1] = Return status (0: OK) [data2 – data7] = Livetime, low byte first [data8 – data13] = Realtime (as above) [data14 – data17] = Fastpeaks [data18 – data21] = Output events For long readout mode only: [data22 – data25] = Underflows [data26 – data29] = Overflows (For DSP versions earlier than 1.08, long readout mode is not supported.)
0x07: Set/Get Run Preset	Set or read the preset run length. Default is no preset run length (indefinite run); values are persistent once set. Ndata = 6 or 8 (set) or 1 (get) [data1] = 0: set, 1: get [data2] = Preset type: 0 = no preset (indefinite run), 1 = fixed realtime, 2 = fixed livetime, 3 = fixed output counts, 4 = fixed input counts [data3–4] = preset length low word, low byte first. [data5–6] = preset length mid word, low byte first. [data7–8] = preset length high word, low byte first (only necessary for large realtime or livetime presets, interpreted as zero if missing). Times are specified in 500 ns units.	Ndata = 6 or 8 [data1] = Return status (0: OK) [data2] = Preset type [data3–4] = preset length low word, low byte first. [data5–6] = preset length mid word, low byte first. [data7–8] = preset length high word, low byte first (only necessary for large realtime or livetime presets, interpreted as zero if missing). Note that the response to a 'set' command is matched to the length of the command, for legacy support. Note, however, that the response to a 'get' command is now always 8. If Handel uses the 'get' command, it should accept either 6 or 8 bytes in the response to support both older and newer DSP code.

Command	***Meaning***	***Response data***
0x08: Take MCA Snapshot	Take snapshot of MCA contents. Allows readout of stable MCA spectrum during data acquisition. Only supported if MCA length is 4096 bins or less. Ndata = 0 or 1 [data1] = 0: no action, 1: clear spectrum and statistics after taking snapshot. If omitted, no action is taken.	Ndata = 1 [data1] = Return status (0: Ok, 1: MCALEN>4096 2: Timeout)
0x09: Read Snapshot MCA	Read snapshot of MCA spectrum. Ndata = 5 [data1] = First MCA bin (low byte) [data2] = First MCA bin (high byte) [data3] = # of bins (low byte) [data4] = # of bins (high byte) [data5] = bytes per bin (1, 2 or 3)	MCA data, with the specified number of bytes per MCA bin. Ndata =1+ (Number of bins)*(bytes per bin) [data1] = Return status (0: OK, 1: MCALEN>4096) [data2] = low byte of first MCA bin … [dataN] = highest byte of highest MCA bin
0x0A: Read Snapshot Run Statistics	Read real time, livetime, input events, and output events from snapshot of MCA data. Ndata = 0	Returns snapshot run statistics – Livetime, realtime, input events (fastpeaks), and output events. Ndata = 29 [data1] = Return status (0: OK, 1: MCALEN>4096) [data2 – data7] = Livetime, low byte first [data8 – data13] = Realtime (as above) [data14 – data17] = Fastpeaks [data18 – data21] = Output events [data22 – data25] = Underflows [data26 – data29] = Overflows

Command	***Meaning***	***Response data***
0x0B: Set Run Mode* *Not supported by standard firmware	Change the data taking method. Ndata = 1 or 6 [data1] = RUNMODE (0 - normal, 1 - auto-readout/resume run, 3 - auto-readout/new run) [data2] = First MCA bin (low byte) [data3] = First MCA bin (high byte) [data4] = # of bins (low byte) [data5] = # of bins (high byte) [data6] = # bytes per bin (1, 2 or 3)	Returns with specified run mode setting. Ndata = 2 [data1] = Return status (0: OK) [data2] = Run Mode
0x0C: Read Snapshot MultiSCA	Read snapshot of number of events in defined SCA regions. Ndata = 0.	MultiSCA data; 4 bytes per SCA Ndata = 2 + 4*NUMSCA (max 66) [data1] = Return status (0: OK) [data2] = NUMSCA [data3] = low byte of SCA0 … [dataN] = highest byte of last SCA
0x0C-0x0F: Reserved		

Diagnostic Tools

Command	*Meaning*	*Response data*
0x10: Read Diagnostic Histogram	Read 1 kword histogram of diagnostic samples. Ndata = 0 or 1 [data1] = Data type 0 = Baseline 1 = ADC 2 = Fast-filter output 3 = Raw baseline 3 = Subtracted raw baseline 5 = Slow-filter baseline Note that changing the data type will reset the histogram. If data type is unmodified or omitted, the histogram is just read back.	Returns diagnostic histogram data – 1024 16-bit words. Ndata = 2049 [data1] = Return status (0: OK) [data2] = first bin (low byte) [data3] = first bin (high byte) … [data 2048] = last bin (low byte) [data 2049] = last bin (high byte)

Command	***Meaning***	***Response data***
0x11: Read Diagnostic Trace	Read diagnostic trace. Sampling time is specified in digitizing clock periods. The minimum period is dependent on the clock rate (setting sampling time to zero will give the minimum period). Ndata = 2 or 6 [data1] = Sampling interval (low byte) [data2] = Sampling interval (high byte) [data3] = Direct Readout via USB* [data4] = Trigger Position (0-255) 0 → No pre-trigger data 128 → Trigger occurs at 50% 255 → All pre-trigger data [data5] = Trigger Type 0 = No trigger (free run) 1 = Fast-filter event 2 = Intermediate or slow event 4 = Good event 8 = Overflow 16 = Underflow 32 = Fast pileup 64 = Slow pileup 128 = ADC out-of-range [data6] = Trace Type 0 = ADC 1 = ADC Average 2 = Fast Filter 3 = Raw Intermediate Filter 4 = Baseline Samples 5 = Baseline Average 6 = Scaled Intermediate Filter 7 = Raw Slow Filter 8 = Scaled Slow Filter	The 8000-deep HISTORY buffer is filled with 16-bit data samples of the selected type, each separated by the specified number of clock ticks. **If Direct Readout via USB is NOT selected ([data3]=0):** Ndata = 16001 [data1] = Return status (0: OK) [data1] = first sample, low byte [data2] = first sample, high byte … [dataN] = last sample, high byte **If Direct Readout via USB is selected ([data3]=1):** Ndata = 1 [data1] = Return status (0: OK) To read the data, command 0x11 should be followed with a direct (DMA) memory read… Note that the HISTORY buffer start address and length are stored in DSP parameters HSTSTART and HSTLEN, respectively.
0x13-0x1F: Reserved		

General Communications and Control

Command	*Meaning*	*Response data*
0x40: I2C Read or Write	Read from or write to an I2C device. Currently, only I2C devices with 7-bit addresses are supported. The maximum number of bytes to read is 64; the sum of the number of command bytes (e.g., target address for an EEPROM) and data bytes to write may not exceed 80. Ndata = 4 + NCMD (for a read) or 4 + NCMD + NI2CD (for a write) [data1] = 0: read, 1: write [data2] = 7-bit I2C address (in bits 1-7; bit 0 indicates a read or a write, and is set internally) [data3] = NCMD. The number of command bytes [data4] = NI2CD = number of I2C data bytes to write/read [data5 …] = Command bytes then data bytes (for a write)	Ndata = 1 + number of bytes read (if any) [data1] = Return status (0: OK) [data2…] = return data
0x41: Read Temperature	Read the temperature measured by the onboard digital temperature sensor. Ndata = 0	Ndata = 3 [data1] = Return status (0: OK) [data2] = signed integer temperature (2's complement) in degrees Celsius. [data3] = fractional portion of temperature. Bit 7 corresponds to 2^(-1), bit 6 to 2^(-2), etc. Bits 0-3 always read 0 (temperature is reported to a precision of 1/16 degree). The fractional portion of the temperature is always positively added to the integer temperature.

Command	*Meaning*	*Response data*
0x42: Read DSP Parameter Names	Read the ordered list of DSP parameters names, which allows referencing the parameters by name. Ndata = 1 [data1] = Readout option 0: Return parameter name string and length parameters 1: Return only length parameters (allows pre-allocating memory for name string)	Ndata = 5 or 5 + StringLen depending on readout option. StringLen is the total number of characters (including null separators) used for the list of public DSP parameters. [data1] = Return status (0: OK) [data2, data3] = Number of parameters [data4, data5] = StringLen [data6…] List of DSP parameter names (in ASCII), null separated.
0x43: Read/Write DSP Parameter *Changes to DSP parameters via command 0x43 should typically be followed by command 0x9F: Apply Settings. NOTE: Use of this command is not recommended, unless advised by XIA support staff.*	Read or write a single DSP parameter. Ndata = 2 (read) or 4 (write) [data1] = 0 (read) or 1 (write) [data2] = parameter number, defined by the position in the ordered parameter list (starting at 0) For write only: [data3,data4] = parameter value, low byte first	Ndata = 3 [data1] = Return status (0: OK) [data2,data3] = parameter value, low byte first
0x44: Read/Write DSP Program Memory *NOTE: Use of this command is not recommended, unless advised by XIA support staff.*	Read directly from or write directly to DSP program memory. Program memory is organized in 24-bit (3 byte) words. Note the unusual byte ordering (dictated by the DSP host interface) Ndata = 4 (read) or 4+3*Nwords (write) [data1] = 0 (read) or 1 (write) [data2] = Nwords (max = 24 for 72 bytes) [data3, data4] = target address, relative to start of PM (low byte first) Write only: [data5…] Data to write (3 bytes per word, middle byte first, then high byte then low byte last)	Ndata = 1 (write) or 1+3*Nwords (read) [data1] = Return status (0: OK) Read only: [data2…] = Data read from PM (3 bytes per word, middle byte then high byte then low byte last)

Command	***Meaning***	***Response data***
0x45: Read/Write DSP Data Memory *NOTE: Use of this command is not recommended, unless advised by XIA support staff.*	Read directly from or write directly to DSP data memory. Data memory is organized in 16-bit (2 byte) words. Ndata = 4 (read) or 4 + 2*Nwords (write) [data1] = 0 (read) or 1 (write) [data2] = Nwords (max = 32 for 64 bytes) [data3, data4] = target address, relative to start of DM (low byte first) Write only: [data5…] Data to write (2 bytes per word, low byte first)	Ndata = 1 (write) or 1+2*Nwords (read) [data1] = Return status (0: OK) Read only: [data2…] = Data read from DM (2 bytes per word, low byte first)
0x48: Read Serial Number	Read serial number. Ndata = 0.	Ndata = 17 [data1] = Return status (0: OK) [data2 – data17] = serial number, ASCII, null terminated (16 character maximum, including null)

Command	***Meaning***	***Response data***
0x49: Get Board Information	Read board information – versions, variants, clock information, FiPPI information. Ndata = 0	[Ndata] = 21 [data1] = Return status (0: OK) [data2] = PIC code variant [data3] = PIC code major version [data4] = PIC code minor version [data5] = DSP code variant [data6] = DSP code major version [data7] = DSP code minor version [data8] = DSP clock speed (MHz) [data9] = Clock Enable register [data10] = NFiPPI (number of FPGA configurations) [data11] = Gain mode (0: fixed, 3: Switched-Gain + Digital, 4: High/Low + Digital) [data12-13] = Nominal gain (gain with variable gain=1) 1.15 mantissa [data14] = Nominal gain exponent Nominal gain = (mantissa/32768)*2^(exponent) [data15] = Nyquist filter (0: 2 MHz, 1: 4 MHz, 2: > 4 MHz) [data16] = ADC speed grade (0: 20 MHz, 1: 40 MHz, 2: 65 MHz) [data17] = FPGA speed (0: normal, 1: fast) [data18] = Analog power supply (0: local regulators, 1: no local regulators) [data19] = FiPPI decimation [data20] = FiPPI version [data21] = FiPPI variant
0x4A: Echo	Echoes command	Ndata same as input command. Checksum is recalculated.
0x4B: Status	Returns board status	Ndata = 6 [data1] = Return status [data2] = PIC status (0: OK) [data3] = DSP boot status (0: OK) [data4] = Run state (0: idle, 1: running) [data5] = DSP BUSY value [data6] = DSP RUNERROR value
0x4C: Set/Get Input Enable	Set or read MicroDXP input enable state (input signal can be disabled with CMOS switch). Ndata = 2 (set) or 1 (get) [data1] = 0: set, 1: get [data2] = input enable (0: disable input, 1: enable input)	Ndata = 2 [data1] = Return status (0: OK) [data2] = Input Enable state (0: disabled, 1: enabled)

Command	*Meaning*	*Response data*
0x4D: Flash Write-Protect Control *NOTE: Use of this command is not recommended, unless advised by XIA support staff.*	The Flash memory write-protection feature prevents writing to the first 4 sectors. It is enabled by default. Ndata = 1 [data1] = write-protect mode 0: protected mode (default) 1: unprotected mode	Ndata = 2 [data1] = Return status (0: OK) [data2] = write-protect mode 0: protected mode (default) 1: unprotected mode
0x4E: Reset FPGA	Reset the FPGA Ndata = 4 [data1] = 0xAA [data2] = 0x55 [data3] = 0xAA [data4] = 0x55	Ndata = 1 [data1] = Return status (0: OK)
0x4F: Reset DSP	Reset the processor – the DSP is rebooted and the FPGA is reprogrammed. The four data bytes must match the values specified below to enable the reset. Ndata = 4 [data1] = 0xAA [data2] = 0x55 [data3] = 0xAA [data4] = 0x55	Ndata = 1 [data1] = Return status (0:OK)

Spectrometer Control

Command	***Meaning***	***Response data***
0x82: Set/Get Parameter Set	Select one of twenty-four peaking times, or read current setting. Ndata = 2 (set) or 1 (get) [data1] = 0: set, 1: get [data2] = Selected parameter set	Ndata = 2 [data1] = return status 0: OK, 1: invalid setting [data2] = Current parameter set
0x83: Set/Get General Set	Select one of five general settings, or read current setting. The GENSET includes parameters related to the MCA and SCA format, and gain. Ndata = 2 (set) or 1 (get) [data1] = 0: set, 1: get [data2] = Selected general set	Ndata = 2 [data1] = return status 0: OK, 1: invalid setting [data2] = Current parameter set
0x84: Set/Get MCA Bin Width	Set the granularity of the MCA bins, or read the current setting. Ndata = 3 (set) or 1 (get) [data1] = 0: set, 1: get [data2] = Granularity 0: Very fine (e.g. 5 eV/bin) 1: Fine (e.g. 10 eV/bin) 2: Medium (e.g. 20 eV/bin) 3: Coarse (e.g. 40 eV/bin) 4: Custom For the custom setting (otherwise ignored): [data3] = Width in terms of the minimum (Very fine) bin width (e.g. a setting of 7 would give 35 eV/bin)	Ndata = 3 [data1] = return status 0: OK, 1: invalid setting [data2] = Current bin granularity [data3] = Custom bin scaling factor
0x85: Set/Get Number of MCA Bins	Set or read the number of MCA bins (max 8192) and the offset (normally 0). Ndata = 5 (set) or 1 (get) [data1] = 0: set, 1: get [data2, data3] = Number of MCA bins (low byte first) [data4,data5] = Offset (the first bin of the spectrum; can be nonzero)	Ndata = 5 [data1] = return status 0: OK, 1: invalid setting [data2, data3] = Current number of MCA bins (low byte first) [data4,data5] = Current MCA offset

Command	*Meaning*	*Response data*
0x86: Set/Get Threshold	Set threshold to apply to one of the filters (up to three), or read the current settings. All thresholds range from 0 to 4095 Ndata = 4 (set) or 1 (get) [data1] = 0: set, 1: get [data2]: Filter choice (0 = fast, 1 = intermediate, 2 = energy) [data3] = Threshold value (low byte) [data4] = Threshold value (high byte)	Ndata = 7 [data1] = return status 0: OK, 1: invalid setting [data2] Fast threshold (low byte) [data3] Fast threshold (high byte) [data4] Intermediate threshold (low byte) [data5] Intermediate threshold (high byte) [data6] Slow threshold (low byte) [data7] Slow threshold (high byte)
0x87: Set/Get Detector Polarity	Select detector preamplifier polarity or read current value. Ndata = 2 (set) or 1 (get) [data1] = 0: set, 1: get [data2] : Polarity: 0 = negative-going steps, 1 = positive	Ndata = 2 [data1] = return status 0: OK, 1: invalid setting [data2] = Current polarity setting
0x89: Set/Get RC Time Constant (Tau)	Set or read the preamplifier RC decay time; applicable to RC feedback preamplifiers only. The time is specified in terms of the digitization clock period. For example, using a preamplifier with a 50 microsecond decay time at a digitization rate of 40 MHz, the decay time setting is 40*50 = 2000. Ndata = 3 (set) or 1 (get) [data1] = 0: set, 1: get [data2] = low byte of decay time [data3] = high byte of decay time	Ndata = 3 [data1] = return status 0: OK, 1: invalid setting [data2] = low byte of Tau [data3] = high byte of Tau
0x8A: Set/Get Preamplifier Reset Time	Set or read the preamplifier reset time; applicable to reset type preamplifiers only. The time is specified in microseconds. Ndata = 2 (set) or 1 (get) [data1] = 0: set, 1: get [data2] reset time in microseconds	Ndata = 2 [data1] = return status 0: OK, 1: invalid setting [data2] = reset time in microseconds

Command	***Meaning***	***Response data***
0x8B: Set/Get Filter Parameter Value	Ndata = 4 (set) or 2 (get) [data1] = 0: set, 1: get [data2] = Parameter number [data3] = Parameter value (low byte) [data4] = Parameter value (high byte) Parameter numbers are as follows: 0-7: Same as before 8: BFACTOR INTLEN = SLOWLEN/2^(BFACTOR+1) 9: PEAKMODE 0 – finding 1 – sampling	Ndata = 4 [data1] = return status 0: OK, 1: invalid [data2] = Parameter number [data3] = Parameter value (low byte) [data4] = Parameter value (high byte)
0x8C: Read Parameter Set Values	Read values in a parameter set (PARSET). The parameter set starts with the parameter NUMPARSET and contains the NUMPARSET parameters immediately following NUMPARSET in DSP parameter memory. The names of the parameters can be obtained by reading the full list of DSP parameter names (command 0x42) and identifying the block starting with NUMPARSET. Ndata = 1 or 3 [data1] = 0: only return NUMPARSET, 1: return NUMPARSET and all parameter values from current set, 2: return all parameters from selected PARSET [data2] = FiPPI number [data3] = PARSET number (0 through 23)	Ndata = 2 or 4 + 2*NUMPARSET [data1] = Return status (0: OK) [data2] = NUMPARSET The following bytes are only returned when the full PARSET data are requested: [data3 – data4] = PARSET version [data5…] = 16-bit parameter values, low byte first
0x8D: Save Parameter Set	Save the current parameter set. The two tag bytes must be correct to enable this command. Ndata = 3 [data1] = PARSET number (0 through 23) [data2] = 0x55 [data3] = 0xAA	Ndata = 2 [data1] = Return status [data2] = Parameter set number (0 through 23)

Command	*Meaning*	*Response data*
0x8E: Read Current General Set Values	Read values in the currently selected general set. The general set starts with the parameter NUMGENSET and contains the NUMGENSET parameters immediately following NUMGENSET in DSP parameter memory. The names of the parameters can be obtained by reading the full list of DSP parameter names (command 0x42) and identifying the block starting with NUMGENSET. Ndata = 0 (only return NUMGENSET) or 1 [data1] = 0: only return NUMGENSET, 1: return NUMGENSET and all parameter values	Ndata = 2 or 4 + 2*NUMGENSET [data1] = Return status (0: OK) [data2] = NUMGENSET The following bytes are only returned when the full GENSET data are requested: [data3 – data4] = GENSET version [data5…] = 16-bit parameter values, low byte first
0x8F: Save Current General Set	Save the current general set. The four tag bytes must be correct to enable this command. Ndata = 3 [data1] = General set number (0 through 4) [data2] = 0x55 [data3] = 0xAA	Ndata = 2 [data1] = Return status [data2] = General set number
0x90: Read SLOWLEN Values	Read out SLOWLEN values from all parameter sets (in order to build a list of available peaking times). The peaking time for a given PARSET can be calculated as: Peaking Time = (1/DSPSPEED) *2^(CLKSET+Decimation)*SLOWLEN In microseconds, where - 1/DSPSPEED is the period of the DSP clock, in microseconds - The digitizing clock is obtained by dividing down by 2^CLKSET - 2^Decimation samples are combined into one entry into the digital filter - SLOWLEN entries are combined to form the trapezoidal filter Ndata = 0	Ndata = 52 [data1] = Return status [data2] = CLKSET [data3] = NFiPPI==1 [data4] = Decimation [data5] = SLOWLEN, PARSET 0 (low byte) [data6] = SLOWLEN, PARSET 0 (high byte) … [data51] = SLOWLEN, PARSET 23 (low byte) [data52] = SLOWLEN, PARSET 23 (high byte)
0x91: Set/Get Gain Trim	Set or get GAINTWEAK value corresponding to current GENSET. This setting fine tunes the digital gain on a per-PARSET (peaking time) basis Ndata = 3 (set) or 1 (get) [data1] = 0: set, 1: get [data2] = GAINTWEAK (low byte) [data3] = GAINTWEAK (high byte)	Ndata = 3 [data1] = Return status [data2] = GAINTWEAK (low byte) [data3] = GAINTWEAK (high byte)

Command	***Meaning***	***Response data***
0x92: Set/Get BLFILTER	Set or get the baseline averaging length via BLFILTER, where: BLFILTER = 65536 / 2^(averaging length) Ndata = 3 (set) or 1 (get) [data1] = 0: set, 1: get [data2] = BLFILTER (low byte) [data3] = BLFILTER (high byte)	Ndata = 3 [data1] = Return status [data2] = BLFILTER (low byte) [data3] = BLFILTER (high byte)
0x93: Set/Get RUNTASKS *NOTE: Use of this command is not recommended, unless advised by XIA support staff.*	Set or get RUNTASKS. Ndata = 3 (set) or 1 (get) [data1] = 0: set, 1: get [data2] = RUNTASKS (low byte) [data3] = RUNTASKS (high byte)	Ndata = 3 [data1] = Return status [data2] = RUNTASKS (low byte) [data3] = RUNTASKS (high byte)
0x94: Set/Get FIPCONTROL *NOTE: Use of this command is not recommended, unless advised by XIA support staff.*	Set or get FIPCONTROL. Ndata = 3 (set) or 1 (get) [data1] = 0: set, 1: get [data2] = FIPCONTROL (low byte) [data3] = FIPCONTROL (high byte)	Ndata = 3 [data1] = Return status [data2] = FIPCONTROL (low byte) [data3] = FIPCONTROL (high byte)
0x96: Set/Get SCA Pulse Periods	Set or get the pulse periods for the SCA pulse output in 25ns units. Ndata = 3 (set) or 1 (get) [data1] = 0: set, 1: get [data2] = SCATIMEON (pulse assert time) *Note: Setting SCATIMEON=0 disables the AUXIO output buffers in the FiPPI!* [data3] = SCATIMEOFF (minimum wait time between pulses)	Ndata = 3 [data1] = Return status [data2] = SCATIMEON [data3] = SCATIMEOFF
0x97: Set/Get Limits for Multiple SCA Regions	Set or get limits for NUMSCA (up to 16) regions. Ndata = 2+4*NUMSCA (set) or 1 (get) [data1] = 0: set, 1: get [data2] = NUMSCA [data3] = SCA0LIMLO (low byte) [data4] = SCA0LIMLO (high byte) [data5] = SCA0LIMHI (low byte) [data6] = SCA0LIMHI (high byte) [data7] = SCA1LIMLO (low byte) etc.	Ndata = 2+4*NUMSCA (max 66) [data1] = Return status [data2] = NUMSCA [data3] = SCA0LIMLO (low byte) [data4] = SCA0LIMLO (high byte) [data5] = SCA0LIMHI (low byte) [data6] = SCA0LIMHI (high byte) [data7] = SCA1LIMLO (low byte) etc.

Command	***Meaning***	***Response data***
0x98: Set/Get Autostart (note: only valid for PIC version 1.3 or later)	Set or get Auto start setting (if set, a run is automatically started after initial boot). Ndata = 1 (get) or 2 (set) [data1] = 0: set, 1: get [data2] = Autostart (0: disabled, 1: enabled)	Ndata = 2 [data1] = Return status (0: OK) [data2] = Autostart setting
0x99: Set/Get SlopeDAC/OffsetDAC Value	Set or read the 16-bit SLOPEDAC (for reset preamplifier) or OFFSETDAC (for RC preamplifier) value. Ndata = 3 (set) or 1 (get) [data1] = 0: set, 1: get [data2] = Slope/ Offset DAC (low byte) [data3] = Slope/ Offset DAC (high byte)	Ndata = 3 [data1] = return status 0: OK [data2] = Slope/ Offset DAC (low byte) [data3] = Slope/ Offset DAC (high byte)
0x9B: Set/Get Switched-Gain	Set or read the 4-bit SWGAIN, which determines the discrete switched-gain. Ndata = 2 (set) or 1 (get) [data1] = 0: set, 1: get [data2] = SWGAIN[3:0]	Ndata = 2 [data1] = return status 0: OK [data2] = SWGAIN[3:0]
0x9C: Set/Get Digital Base Gain	Set or read the 16-bit DGAINBASE (0.16 mantissa) and signed 8-bit DGAINBASEEXP (exponent), which determine the Digital Base Gain. Ndata = 4 (set) or 1 (get) [data1] = 0: set, 1: get [data2] = DGAINBASE[7:0] [data3] = DGAINBASE[15:8] [data4] = DGAINBASEEXP[7:0]	Ndata = 4 [data1] = return status 0: OK [data2] = DGAINBASE[7:0] [data3] = DGAINBASE[15:8] [data4] = DGAINBASEEXP[3:0]
0x9D: Set/Get ADC Offset	**RC variants only:** Set or read the 14-bit ADC Offset value (to SETOFFADC). The DSP immediately adjusts the Offset DAC iteratively to match the baseline ADC value to the user setting. If ADC Offset = 0, DAC is just set to mid-range value (0x8000). Ndata = 3 or 4 (set) or 1 (get) [data1] = 0: set, 1: get [data2] = ADC Offset (low byte) [data3] = ADC Offset (high byte) [data4] = enable auto-adjust zero: DAC will remain static until next command 0x9A (or power cycle) non-zero: DAC will be re-adjusted whenever ApplySettings routine is called, e.g. when analog gain or other settings are changed. IF Ndata =4 GLOBSET IS SAVED! If Ndata=3, current auto-adjust setting is maintained.	Ndata = 4 [data1] = return status 0: OK [data2] = ADC Offset (low byte) [data3] = ADC Offset (high byte) [data4] = enable auto-adjust

Command	***Meaning***	***Response data***
0x9E: Get Reset/RC Switch Position	Set or read SWITCHPOS Ndata = 0	Ndata = 2 [data1] = return status 0: OK [data2] = Switch Position: Reset (0) or RC (1)
0x9F: Apply Settings	Apply settings defined by DSP parameter values. **This command is only needed after DSP parameters have been modified via command 0x43: Read/Write DSP Parameter.** Ndata = 0	Ndata = 1 [data1] = Return status
0xA0: Set/Get ADC Max	**Reset variants only:** Set or read the 14-bit ADC Maximum value (to ADCMAX), which sets the overhead reserve (from ADCMAX to 16383) in order to improve the representation of high-energy events in the spectrum. Signal is considered ‘out of range’ when ADC is above ADCMAX, however, if the ADC is still below 16383 the preceding event can still be processed. Ndata = 3 or 4 (set) or 1 (get) [data1] = 0: set, 1: get [data2] = ADC Maximum (low byte) [data3] = ADC Maximum (high byte) [data4] = enable ADC Maximum 0: Use absolute maximum value 16383, i.e. ignore ADCMAX setting 1: Use ADCMAX value	Ndata = 4 [data1] = return status 0: OK [data2] = ADC Max (low byte) [data3] = ADC Max (high byte) [data4] = enable