microDXP

RS-232 Communications Specification

Version 3.19

January 12, 2016

microDXP Hardware Revision: H (8)

XIA LLC
31057 Genstar Rd
Hayward, CA 94544 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 not correct 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

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 … [lastdata] = highest byte of highest mca bin

Command	***Meaning***	***Response data***
0x03: Read SCA *Not implemented.*	Read number of events in defined SCA. Ndata = 0.	Single SCA data; 4 bytes. Ndata = 5) [data1] = Return status (0: ok) [data2] = lowest byte of SCA counts … [data3] = highest byte of SCA counts
0x04: Read MultiSCA *Not implemented.*	Read number of events in multiple SCA regions. Ndata = 0. Requires special firmware.	MultiSCA data; 4 bytes per SCA Ndata = 2 + 4*NUMSCA [data1] = Return status (0: ok) [data2] = NUMSCA [data3] = low byte of SCA0 … [lastdata] = highest byte of last SCA
0x05: Read MultiSCAMAP *Not implemented.*	Read MultiSCA map. Each point includes statistics. Ndata = 0. Requires special firmware.	MultiSCA map data. Each point contains statistics and MultiSCA data. Ndata = 4+NUMPOINTS*(12+3*NUMSCA) [data1] = Return status (0: ok) [data2] = NUMSCA [data3] = NUMPOINTS, low byte [data4] = NUMPOINTS, high byte [data5 – lastdata] = MultiSCA data For each point, the order of bytes is: Livetime: low, middle, high byte Realtime: low, middle, high byte Fastpeaks: low, middle, high byte Events: low, middle high byte SCA0: low, middle, high byte …

Command	*Meaning*	*Response data*
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. For PIC version 1.04 and later, an argument to this command is allowed: [data1]: Readout mode 0: short, 1: long. If omitted, the short readout mode is used.	Returns run statisitics – 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 (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 – data6] = preset length, low byte first. Times are specified in 500 ns units.	Ndata = 6 [data1] = Return status (0: ok) [data2] = Preset type [data3 – data6] = preset length, low byte first.
0x08: Take MCA Snapshot (Requires PIC version 1.04 and DSP version 1.08 or later)	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)

Command	*Meaning*	*Response data*
0x09: Read Snapshot MCA (Requires PIC version 1.04 and DSP version 1.08 or later)	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 … [lastdata] = highest byte of highest mca bin
0x0A: Read Snapshot Run Statistics (Requires PIC version 1.04 and DSP version 1.08 or later)	Read real time, livetime, input events, and output events from snapshot of MCA data. Ndata = 0	Returns snapshot run statisitics – 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
0x0B: Set Run Mode	Change the data taking method. Ndata = 1 [data1] = Mode 0: normal 1: auto read out	Returns with specified run mode setting. Ndata = 2 [data1] = Return status (0:ok) [data2] = Mode
0x0C-0x0F: Reserved		

Diagnostic Tools

Command	*Meaning*	*Response data*
0x10: Read Diagnostic Histogram	**Old microDXP:** Read 1 kword histogram of baseline samples. Ndata = 0 **SuperMicro:** 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.	**Old microDXP:** Returns baseline histogram data – 1024 16-bit words. **SuperMicro:** 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). **Old microDXP:** Always ADC trace with no trigger. Ndata = 0 (keep settings) or 2 [data1] = Sampling time (low byte) [data2] = Sampling time (high byte) **SuperMicro:** Ndata = 0 (keep settings), 2, 3 or 6 [data1] = Sampling time (low byte) [data2] = Sampling time (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 = Int or slow event 4 = Good event 8 = Overflow 16 = Underflow 32 = Fast pileup 64 = Slow pileup 128 = ADC out-of-range [data6] = Trace Type *If Ndata >= 3 AND [data3] = 1, trace data is omitted from the RS-232 response.	The HISTORY buffer is filled with HSTLEN samples of the data, separated by the specified number of clock ticks. **Old microDXP:** Ndata = 1 + 2 * HSTLEN [data1] = Return status (0: ok) [data1] = first sample, low byte [data2] = first sample, high byte … [lastdata] = last sample, high byte Note: HSTLEN varies with firmware, but is typically 8000. **SuperMicro:** Same as old microDXP, unless the new Direct Readout via USB mode has been selected, then: Ndata = 1 [data1] = Return status (0: ok) Note that the HISTORY buffer start address and length are stored in DSP parameters HSTSTART and HSTLEN, respectively.

Command	*Meaning*	*Response data*
0x12: Read Baseline History SuperMicro will return garbage – instead use command 0x11 with Trace Type = 3	Read a history of the most recent baseline average values. Ndata = 0.	The HISTORY buffer is filled with HSTLEN consecutive values of the baseline average. Ndata = 1 + 2 * HSTLEN [data1] = Return status (0: ok) [data2] = first point (low byte) [data3] = first point (high byte) … [lastdata] = last point, high byte
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.
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 preallocating 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.

Command	*Meaning*	*Response data*
0x43: Read/Write DSP Parameter	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	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)
0x45: Read/Write DSP Data Memory	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)

Command	*Meaning*	*Response data*
0x46: Set/Get Auto Sleep	Set or read auto sleep mode. If enabled, the uDXP will enter a low power state at the specified time after the end of a run (if another run is not started). Ndata = 5 (set) or 1 (get) [data1] = 0: set, 1: get [data2] = Auto sleep mode 0: disable automatic sleep *n*: enable automatic sleep into sleep mode *n* (see below – *n* ranges from 1 to 15) [data3,data4] = delay before entering sleep mode, in seconds (low byte first) [data5] = delay coming out of sleep (in 10 ms intervals – 2.55 second maximum)	Ndata = 5 [data1] = Return status (0: ok) [data2] = Auto sleep mode [data3,data4] = delay before entering low power mode (seconds, low byte first) [data5] = Wakeup delay (10 ms units)
0x47: Set/Get Sleep Mode	Set uDXP sleep mode. Several different subsystems can be powered down – the analog section (if using onboard regulators), the ADC (automatically powered down if analog voltages disabled), the FPGA and the DSP (slow idle state). Setting the bit corresponding to each subsection enables the power-down mode. Ndata = 3 (set) or 1 (get) [data1] = 0: set, 1: get [data2] = sleep mode (bit set to 1 means power-down) bit 0: Analog power-down bit 1: ADC power-down bit 2: FPGA power-down [data3] = Wakeup delay (in 10 ms intervals – 2.55 second maximum)	Ndata = 3 [data1] = Return status (0: ok) [data2] = Power-down state [data3] = Wakeup delay (10 ms units)
0x48: Read Serial Number	Read serial number. Ndata = 0.	Ndata = 1 + SerialLength [data1] = Return status (0: ok) [data2 – lastdata] = 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] = 18 + 3*NFiPPI [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) **Old microDXP:** [data11] = Gain mode (0: fixed, 1: VGA) **SuperMicro:** [data11] = Gain mode (0: fixed, 1: VGA, 2: VGA + Digital; 3: Switched-Gain + 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) Then, for each FiPPI: [dataN] = FiPPI decimation [dataN+1] = FiPPI version [dataN+2] = 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

Command	*Meaning*	*Response data*
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)
0x4D-0x4E: Reserved		
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*
0x80: Set/Get Digitizing Clock	Set or read current digitizing rate. Ndata = 2 (set) or 1 (get) [data1] = 0: set, 1: get [data2] = Clock selection **Old microDXP:** 0: DSPCLK, 1: DSPCLK/2, 2: DSPCLK/4. 3: DSPCLK/8 **SuperMicro:** -1: DSPCLK*2, 0: DSPCLK	Ndata = 2 [data1] = return status 0: OK, 1: invalid setting (selected clock speed not supported) [data2] = Current clock setting.
0x81: Set/Get FiPPI Configuration Could be used to select RC or Reset FiPPI for the SuperMicro…or we could create a new command.	Select or read fpga configuration. Up to three can be stored on the board. Ndata = 2 (set) or 1 (get) [data1] = 0: set, 1: get [data2] = FiPPI selection (0, 1 or 2) Number of configurations depends on board options.	Ndata = ~~3~~ 2 [data1] = return status 0: OK, 1: invalid setting [data2] = Current FiPPI configuration (0, 1 or 2) ~~[data3] = Decimation value~~ 150302 – Looks like Decimation isn't actually returned by the DSP or PIC.

Command	*Meaning*	*Response data*
0x82: Set/Get Parameter Set	**Old microDXP:** Select one of five peaking times available for a given Clock/FiPPI combination, or read current setting. The base peaking time is given by the selected clock and FiPPI: BaseTime = 6 * Clock Period * 2^Decimation Within a FiPPI, five peaking times are available: 1, 1.5, 2, 3 and 4 times the base peaking time value, corresponding to parameter sets 0, 1, 2, 3 and 4. **SuperMicro:** 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 General set includes parameters that do not depend on peaking time: gain, bin width, polarity, number of bins, preaplifier characteristics (tau rc or reset time). 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, 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/ch) 1: Fine (e.g. 10 eV/ch) 2: Medium (e.g. 20 eV/ch) 3: Coarse (e.g. 40 eV/ch) 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/ch)	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. **Old microDXP:** Ndata = 3 (set) or 1 (get) [data1] = 0: set, 1: get [data2] : Filter choice (0 = fast, 1 = intermediate, 2 = energy) [data3] : Threshold value (up to 255) Set threshold to apply to one of the filters (up to three), or read the current settings. **SuperMicro:** 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)	**Old microDXP:** Ndata = 4 [data1] = return status 0: OK, 1: invalid setting [data2] = Current fast threshold [data3] = Current intermediate threshold [data4] = Current slow threshold **SuperMicro:** 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
0x88: Set/Get Base Gain	**Only used if GAINMODE equals 1 or 2:** Set or read the 16-bit GAINBASE value. Changing this value modifies the gain for all choices of PARSET. Ndata = 3 (set) or 1 (get) [data1] = 0: set, 1: get [data2] = GAINBASE (low byte) [data3] = GAINBASE (high byte)	Ndata = 3 [data1] = return status 0: OK, 1: no GainDAC installed [data2] = GAINBASE (low byte) [data3] = GAINBASE (high byte)

Command	*Meaning*	*Response data*
0x89: Set/Get RC Time Constant (Tau)	Set or read tau, 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
0x8B: Set/Get Filter Parameter Value	Set or read the value of one of the parameters that controls the digital filtering. These parameters are part of the parameter set that is stored in flash memory, and only need to be changed for optimization purposes. **Old microDXP:** Ndata = 3 (set) or 2 (get) [data1] = 0: set, 1: get [data2] = Parameter number [data3] = Parameter value Parameter numbers are as follows: 0: SLOWLEN 1: SLOWGAP 2: PEAKINT 3: PEAKSAM 4: FASTLEN 5: FASTGAP 6: MINWIDTH 7: MAXWIDTH **SuperMicro:** 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	**Old microDXP:** Ndata = 3 [data1] = return status 0: OK, 1: invalid parameter [data2] = Parameter number [data3] = Parameter value **SuperMicro:** Ndata = 4 [data1] = return status 0: OK, 1: invalid [data2] = Parameter number [data3] = Parameter value (low byte) [data4] = Parameter value (high byte)

Command	*Meaning*	*Response data*
0x8C: Read Parameter Set Values	Read values in a parameter set. 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 **Old microDXP:** (0 through 4) **SuperMicro:** (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 **Old microDXP:** (0 through 4) **SuperMicro:** (0 through 23) [data2] = 0x55 [data3] = 0xAA	Ndata = 2 [data1] = Return status [data2] = Parameter set number **Old microDXP:** (0 through 4) **SuperMicro:** (0 through 23)
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

Command	*Meaning*	*Response data*
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: PeakingTime = (~~1/32~~ 1/DSPSPEED) *2^(CLKSET+Decimation)*SLOWLEN In microseconds, where ~~- 1/32 is the period of the 32 MHz clock~~ - 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	**Old microDXP:** Ndata = 3 + 6*NFiPPI [data1] = Return status [data2] = CLKSET [data3] = NFiPPI Then for each FiPPI: [dataN] = Decimation [dataN+1] = SLOWLEN, parset 0 [dataN+2] = SLOWLEN, parset 1 [dataN+3] = SLOWLEN, parset 2 [dataN+4] = SLOWLEN, parset 3 [dataN+5] = SLOWLEN, parset 4 **SuperMicro:** 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. Note that GAINTWEAK calculation now depends on GAINMODE (see SuperMicro_Gain_Specification.docx) 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)
0x92: Set/Get BLFILTER	Set or get BLFILTER value. 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)

Command	*Meaning*	*Response data*
0x93: Set/Get RUNTASKS	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	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)
~~**0x95: Set/Get TRACEWAIT**~~	~~Set or get TRACEWAIT.~~ ~~NData = 3~~ ~~[data1] = 0: set, 1: get~~ ~~[data2] = low byte of new TRACEWAIT value~~ ~~[data3] = high byte~~	~~Ndata = 3~~ ~~[data1] = Return status~~ ~~[data2] = TRACEWAIT, low byte~~ ~~[data3] = high byte~~
0x95: Set/Get AC-Coupling Controls	Set or read TAUCTRL, which controls for the AC-coupled circuit variant. Ndata = 2 (set) or 1 (get) [data1] = 0: set, 1: get [data2] = TAUCTRL[7:0] 0*: AC-coupled w/ 500ns decay time 1: AC-coupled w/ 1us decay time 2: AC-coupled w/ 4us decay time 3-254: AC-coupled w/ 10us decay time 255: DC-coupled *Note that 0 forces ASC_RESET*=0, overriding RUNTASKS bits 11 and 12.	Ndata = 2 [data1] = return status 0: OK [data2] = TAUCTRL[7:0]
0x96: Set/Get SCA Pulse Periods	Set or get time periods for the SCA pulse output: time = (SCATIMExx+1)*clock period 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

Command	*Meaning*	*Response data*
0x97: Set/Get Limits for Multiple SCA Regions	Set or get limits for NUMSCA (up to 4) 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 [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.
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 = 2 [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)
~~**0x9A: Set/Get Statistics Readout Mode**~~	~~Set or read the statistics readout mode (see command 0x6). A value of 0 specifies short readout mode (under- and overflows are not included); a nonzero value specifies long readout mode.~~ ~~Ndata = 2~~ ~~[data1] = 0: set, 1: get~~ ~~[data2] = Statistics readout mode:~~ ~~0: short, 1: long~~	~~Ndata = 2~~ ~~[data1] = return status~~ ~~0: OK~~ ~~[data2] = Statistics readout mode~~ ~~0: short, 1: long~~
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]

Command	*Meaning*	*Response data*
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	Set or read SETOFFADC Ndata = 3 (set) or 1 (get) [data1] = 0: set, 1: get [data2] = SETOFFADC [7:0] [data3] = SETOFFADC [15:8]	Ndata = 3 [data1] = return status 0: OK [data2] = SETOFFADC [7:0] [data3] = SETOFFADC [15:8]
0x9E: Get Reset/RC Switch Position	Set or read SETOFFADC 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. Ndata = 0	Ndata = 1 [data1] = Return status