

Mini-X2 Programming Guide

(A subset of 'Amptek Digital Products Programmer's Guide')

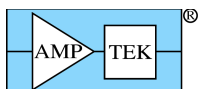
Amptek Inc.

14 Deangelo Drive, Bedford, MA 01730
PH: +1 781 275 2242 FAX: +1 781 275 3470
amptek.sales@ametek.com www.amptek.com

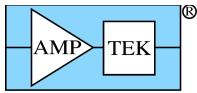
Note: This document applies to Amptek Mini-X2. DP5 references may also apply to the Mini-X2.

Table of Contents

1	Change List.....	3
1.1	Mini-X2 Programming Guide Rev B2 (this document).....	3
2	Software.....	3
2.1	Uploading new firmware.....	3
2.2	Software Resources.....	3
2.2.1	DP5 VB Demo.....	3
2.2.2	SDK.....	3
3	Interfaces.....	4
3.1	USB.....	4
3.1.1	Installing the WinUSB Driver.....	4
3.1.2	WinUSB GUIDs.....	5
3.2	RS232.....	5
3.3	Netfinder (Mini-X2).....	5
4	Packet Format.....	6
4.1	Mini-X2 Request Packets (Host to Mini-X2).....	8
4.1.1	Request packet: "Request Status Packet".....	9
4.1.2	Request packet: "Request Misc Data".....	9
4.1.3	Request packet: "Request Netfinder Packet".....	10
4.1.4	Request packet: "I ² C Transfer".....	11
4.1.5	Request packet: "Request Mini-X Tube & Interlock table" [Mini-X2 only].....	12
4.1.6	Request packet: "Request Mini-X Warmup table" [Mini-X2 only].....	12
4.1.7	Request packet: "Request Mini-X Timestamp record" [Mini-X2 only].....	12
4.1.8	Request packet: "Request Mini-X Fault record" [Mini-X2 only].....	13
4.1.9	Request packet: "Text Configuration".....	14
4.1.10	Request packet: "Text Configuration Readback".....	15
4.1.11	Request packet: "Write 512-byte Misc Data".....	16
4.1.12	Request packet: "Select RS232 Baud Rate".....	16
4.1.13	Request packet: "Write Mini-X Tube & Interlock Table" [Mini-X2 only].....	17
4.1.14	Request packet: "Write Mini-X Timestamp" [Mini-X2 only].....	18
4.1.15	Request packet: "Write Mini-X Warmup Table" [Mini-X2 only].....	19
4.1.16	Request packet: "Initiate Mini-X Warmup Sequence" [Mini-X2 only].....	21
4.1.17	Request packet: "Comm test - Request ACK packet".....	21
4.1.18	Request packet: "Comm test - Echo packet".....	22



4.2	Mini-X Response Packets.....	22
4.2.1	Response packet: “Mini-X Status Packet” [Mini-X only].....	23
4.2.2	Response packet: “512-byte Misc Data Packet”.....	25
4.2.3	Response packet: “Configuration Readback”.....	25
4.2.4	Response packet: “Netfinder Packet”.....	26
4.2.5	Response packet: “I ² C Read Data”.....	26
4.2.6	Response packet: “Mini-X Tube & Interlock Table” [Mini-X2 only].....	27
4.2.7	Response packet: “Mini-X Warmup Table” [Mini-X2 only].....	29
4.2.8	Response packet: “Mini-X Timestamp Record” [Mini-X2 only].....	31
4.2.9	Response packet: “Mini-X Fault Record” [Mini-X2 only].....	32
4.2.10	Response packet: “Comm test - Echo packet”.....	32
4.3	Acknowledge Packets.....	33
4.3.1	Acknowledge packet: “OK”.....	33
4.3.2	Acknowledge packet: “OK, with Interface Sharing Request”.....	33
4.3.3	Acknowledge packet: “Sync Error”.....	34
4.3.4	Acknowledge packet: “PID Error”.....	34
4.3.5	Acknowledge packet: “LEN Error”.....	34
4.3.6	Acknowledge packet: “Checksum Error”.....	34
4.3.7	Acknowledge packet: “Bad Parameter”.....	35
4.3.8	Acknowledge packet: “Unrecognized Command”.....	35
4.3.9	Acknowledge packet: “PC5 Not Present”.....	35
4.3.10	Acknowledge packet: “Bad Hex Record”.....	35
4.3.11	Acknowledge packet: “FPGA Error”.....	36
4.3.12	Acknowledge packet: “CP2201 Not Found”.....	36
4.3.13	Acknowledge packet: “Scope Data Not Available”.....	36
4.3.14	Acknowledge packet: “I ² C Error”.....	36
4.3.15	Acknowledge packet: “Feature not supported by this FPGA version”.....	37
4.3.16	Acknowledge packet: “Calibration data not present”.....	37
5	ASCII Commands.....	38
	Table 2 – ASCII Command Summary [Mini-X2].....	38
5.1.1	CUSE [Mini-X2] – Set Tube Current.....	39
5.1.2	FAOR [Mini-X2]- Set Fault Override.....	40
5.1.3	HVSE [Mini-X2] – Set Tube High Voltage.....	41
5.1.4	LIOR [Mini-X2]- Set Limit Override.....	42
5.1.5	VOLU [Mini-X2] – Mini-X2 Speaker Control.....	43



1 Change List

Mini-X2 Programming Guide B2 is released with firmware version FW6.09.09, FPGA version FP7.03, and Visual Basic Demo v2.87.

1.1 Mini-X2 Programming Guide Rev B2 (this document)

Support for Mini-X2 X-ray tube controller

2 Software

2.1 Uploading new firmware

The Firmware Manager application can be downloaded from the Amptek website for use in loading new firmware, changing various calibration settings, or changing the TCP/IP configuration of the device. Firmware Manager can be downloaded from here:

<http://www.amptek.com/firmware/firmwaremanager.zip>

Firmware Manager works best if it is allowed to access the Internet, in which case it can install the very latest firmware. However, the ZIP file download includes all current firmware files, so they can be manually installed as needed.

Firmware Manager replaces the Firmware Loader and DP5 Loader applications. Firmware Manager can upgrade FW5-based DP5s via RS232, and all other devices via USB, Ethernet or RS232.

2.2 Software Resources

There are several resources available to assist in developing an application:

2.2.1 DP5 VB Demo

The DP5 VB Demo (includes Mini-X2 demonstration) can be downloaded from the Amptek website. It is written in Visual Basic 5, and shows how to use WinUSB, Winsock, and the VB Comm control to communicate with a DP5-family device. It handles nearly all packet types; it can upload new firmware and set various calibration values; it supports List Mode, though not at the highest count rates.

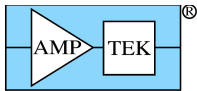
http://www.amptek.com/zip/dppsoft_3.zip

2.2.2 SDK

A Software Development Kit (SDK) is available from the Amptek website. It has C++ and VB examples. LabVIEW and other language examples may also be found on the website. Contact Amptek for more information.

<http://www.amptek.com/products/dp5-digital-pulse-processor-software/>

Amptek Inc.



3 Interfaces

The Mini-X2 supports two communication interfaces: USB and RS232. Details about the specific interfaces are in the following sections.

Note: Attempting to use multiple interfaces simultaneously can result in unexpected behavior.

3.1 USB

The DP5 (includes Mini-X2) family supports ‘full-speed’ (12Mbps) USB 2.0. Three endpoints are used: the ‘control’ endpoint (EP0); EP1 IN (for response packets from the DP5 to host PC); and EP2 OUT (for request packets from the host PC to DP5.)

Transfers on EP1 and EP2 are terminated by a ‘short packet’ (a packet smaller than the max packet size of 64 bytes), or a zero-length packet (‘ZLP’).

For the Mini-X2, the USB Vendor ID (VID) and Product ID (PID) are: VID: 0x10C4 PID: 0x842A

The DPPMCA application (and DP5 Visual Basic demo) use the WinUSB driver. This is a Microsoft product that is native to Vista and Windows 7 (32- and 64-bit), and installable on WinXP. The Visual Basic demo includes source code, and VC++ classes are available for those wishing to write their own communication software.

3.1.1 Installing the WinUSB Driver

The WinUSB driver has been tested by WinQual and signed for WinXP 32/64, Vista 32/64, and Win7 32/64. It is available via Windows Update, which means that a PC with an Internet connection should be able to install the driver automatically. Directions for manually installing the driver are in the ‘WinUSB Driver Installation’ folder in the ZIP file.

The ZIP file includes the Microsoft document ‘WinUsb_HowTo.docx’, which describes the complex process of using Windows’ SetupAPI to find the device path, etc. [The VB demo demonstrates this technique, as well as setting pipe policies, opening pipes, sending and receiving data, etc.]

USB.h and USB100.h contain many structures, enums, etc. used by Windows’ USB implementation – these files (from DDK6001) are in the ZIP file.

Microsoft’s website gives information on how to call the specific WinUSB functions:

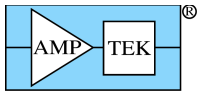
[http://msdn.microsoft.com/en-us/library/ff540046\(v=VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ff540046(v=VS.85).aspx)

As of FW6.07.04, Windows 8 USB descriptors have been added, which may allow Win8 to automatically load WinUSB without having to install it. Microsoft’s documentation is a bit sketchy on this; it hasn’t been tested yet at Amptek.

Win8 descriptors added:

1. An ‘OS String Descriptor’, to tell Windows that extended feature descriptors are present;
2. An ‘Extended Compat ID OS feature descriptor’, to tell Windows to use WinUSB; and

Amptek Inc.



3. An 'Extended properties OS feature descriptor', to specify the device's interface GUID.

3.1.2 WinUSB GUIDs

The WinUSB driver for the DP5 family utilizes two GUIDs:

FW6 GUIDs:

Setup Class GUID: {6A4E9A2D-9368-4f01-8E60-B3F9CDBAB5E8}

Interface Class GUID: {5A8ED6A1-7FC3-4b6a-A536-95DF35D03448}

3.2 RS232

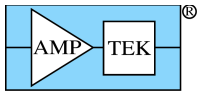
The RS232 port on the DP5 family operates at 115,200 baud, with 1 start bit, 8 data bits, no parity bit, and 1 stop bit. Neither hardware nor software handshaking is used. 57,600 or 19,200 baud operation can be selected; however, selecting these rates will disable the Ethernet port.

The RS232 interface employs a 'gap timer' on its receiver, to aid in synchronization – if the time between consecutive received bytes exceeds 100mS, all buffered bytes are discarded, and the DP5 will resume searching for the sync characters that signal the start of a new packet. No acknowledge packet will be sent to indicate this has occurred.

3.3 Netfinder (Mini-X2)

While the Mini-X2 does not support Ethernet operation, it does support the Netfinder Identity Packet via USB (and RS232) for ease of device identification. The Netfinder Request Packet is specified in section 4.1.3 and the resulting Response Packet data is given here:

Data Offset	Value
0	0x01
1-3	0
4-5	Event 1 Days
6	Event 1 Hours
7	Event 1 Minutes
8-11	0
12	Event 1 Seconds
13	0
14-31	0
variable	Null-terminated string: 'Amptek Mini-X2 – S/N ' + serial number
variable	Null-terminated string: 'Tube P/N ' + part number (up to 19 characters)
variable	Null-terminated string – Event 1 description: 'Time Powered' [Note: refers to the time the Mini-X2 has currently been powered on, not the x-ray tube]
variable	Null-terminated string – Event 2 description: "none"



4 Packet Format

All communications with the DP5 follow a request/response format: the host sends a Request Packet to the DP5, and the DP5 returns a Response Packet. The Response Packet sent by the DP5 to the host will be:

- a packet containing the requested data;
- an acknowledge (ACK) packet, indicating the request packet was received and interpreted without error (if the request packet is not one for which data is returned);
- an ACK packet, indicating the request packet contains an error in structure or content;
- an ACK packet, indicating why the request can't be executed;
- an ACK packet, indicating that an ASCII command had a unrecognized command field or an invalid parameter field – the command in question will be echoed in the data field of the ACK packet.
- an ACK packet, indicating the request packet was received and interpreted without error, and a sharing request was received from another computer via Ethernet;

All packets sent to and from the DP5 use the same basic packet format: a 6-byte header, which defines the type and length of the packet, an optional data field, and a 16-bit checksum. For packets sent to the DP5, the optional data field can be up to 512 bytes, for a maximum packet size of 520 bytes. For packets sent by the DP5 to the host PC, the maximum data field size is 32767 for a maximum packet size of 32775 bytes.

Packet Format

Offset	0	1	2	3	4	5	6...5+LEN	6+LEN	7+LEN
Value	0xF5	0xFA	PID1	PID2	LEN_MSB	LEN_LSB	Data (optional, 0-512 bytes)	CHKSUM_MSB	CHKSUM_LSB

Fields:

Offset 0 & 1: Sync bytes – these fields have fixed values of 0xF5 and 0xFA.

Offset 2 & 3: Packet ID fields 1 & 2 (PID1 & PID2) – these define the meaning of the packet, as summarized in Error: Reference source not found, Error: Reference source not found and Table 1.

Offset 4 & 5: 16-bit length field (LEN) of the optional data field. If data field is not present, LEN=0.

If LEN =0:

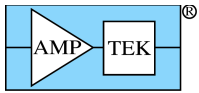
Offset 6 & 7: 16-bit checksum: MSB, then LSB. This is a two's-complement of the 16-bit sum of all bytes prior to the checksum. (i.e. the 16-bit sum of the checksum and all other bytes in the packet is 0.)

If LEN > 0:

Offset 6: The start of the data field, whose length is given by the 16-bit LEN field. The last byte is offset 5+LEN.

Offset 6+LEN, Checksum, as described above.

7+LEN:



Checksum Visual Basic 5 example code:

‘ Note that a VB Long (32 bit) integer is used for the sum, as VB integers are signed,
‘ and a 16-bit signed INT wouldn’t work
‘ P() is the raw packet buffer

```
Dim CSum As Long, X as Integer, LEN as Integer
```

```
CSum = 0
```

```
LEN = P(4) * 256 + P(5) ‘ get the LEN field
```

```
For X = 0 To LEN + 5 ‘ add up all the bytes except checksum
```

```
    CSum = CSum + P(X)
```

```
Next X
```

```
CSum = CSum + 256 * CLng(P(LEN + 6)) + CLng(P(LEN + 7)) ‘ add the 16-bit checksum
```

```
If (CSum And &HFFFF) = 0 Then
```

```
    ‘ packet is OK
```

```
    ‘ process the packet
```

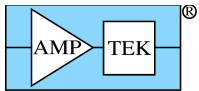
```
Else
```

```
    ‘ checksum error
```

```
    ‘ handle the error
```

```
End If
```

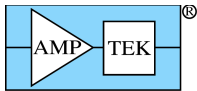
Note: Prior to Programmer’s Guide Rev B1, the line ‘CSum = CSum + 256...’ was missing!



4.1 Mini-X2 Request Packets (Host to Mini-X2)

Description	Support*	PID1	PID2	LEN MSB	LEN LSB	Data (optional)	Response
Request status packet	ALL	1	1	0	0		Data packet
Request 512-byte misc data	ALL	3	2	0	0		Data packet
Request Netfinder packet	ALL	3	7	0	0		Data packet
Perform I2C transfer	ALL	3	8	varies		I2C transfer description	ACK or Data
Request Mini-X Tube & Interlock table	ALL	3	11	0	0		
Request Mini-X Warmup table	ALL	3	12	0	0		Data packet
Request Mini-X Timestamp record	ALL	3	13	0	0		Data packet
Request Mini-X Fault record	ALL	3	14	0	0		Data packet
Text configuration (to DP5)	ALL	0x20	2	varies		ASCII configuration	ACK packet
Text configuration Readback (from DP5)	ALL	0x20	3	varies		ASCII configuration	Data packet
Write 512-byte Misc Data	ALL	0xF0	0x09	2	0		ACK packet
Select RS232 baud rate	---	0xF0	0x13	0	1	Baud rate	ACK packet
Write Mini-X Tube & Interlock table	ALL	0xF0	0x30	0	0x5E		ACK packet
Write Mini-X Timestamp	ALL	0xF0	0x31	0	6	Binary DMYHMS	ACK packet
Write Mini-X Warmup Sequence	ALL	0xF0	0x32	0	0x30		ACK packet
Initiate Mini-X Warmup Sequence	ALL	0xF0	0x33	0	1	0=daily, 1=monthly	ACK packet
Comm test - Request ACK packet	ALL	0xF1	0...15	0	0		ACK packet
Comm test - Echo packet	ALL	0xF1	0x7F	varies		Data to be echoed	Data packet

* Initial firmware release which supports this packet type;



4.1.1 Request packet: “Request Status Packet”

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	1	1	0	0	0xFE	0x0F

Purpose: This request directs the device to construct a status packet and send it to the host PC. The status packet can be requested with this request packet, or with one of the ‘Request Spectrum plus Status’ variants, in which case the status packet is included in the packet data field along with the spectrum data.

The status packet contains information about the current state of the spectrum, and of the device. Its format is listed with the “Status Packet” Response Packet.

Response: If no errors are detected, a “Status Packet” Response Packet will be returned. For a Mini-X2, the ‘Mini-X2 Status Packet’ Response Packet will be returned.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of the requested data.

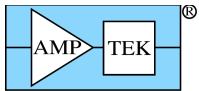
4.1.2 Request packet: “Request Misc Data”

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	3	2	0	0	0xFE	0x0C

Purpose: This requests that the DP5 send the 512-bytes of ‘miscellaneous data’ that is stored in the DP5’s non-volatile memory. (See also the “Write Misc Data” Request Packet for information on to write this data to the DP5.)

Response: If no errors are detected, the DP5 will respond with the “512-byte Misc Data” Response Packet (PID1=0x82, PID2=2).

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of the requested data.



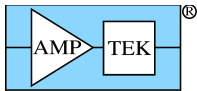
4.1.3 Request packet: “Request Netfinder Packet”

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	3	7	0	0	0xFE	0x07

Purpose: This requests a Netfinder packet, but through the active communications port. [The ‘Netfinder packet’ is generally obtained via a broadcast on the TCP/IP Netfinder socket. See section Error: Reference source not found for details.]

Response: If no errors are detected, the DP5 will respond with the “Netfinder Packet” Response Packet (PID1=0x82, PID2=8).

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of the requested data.



4.1.4 Request packet: “I²C Transfer”

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6...5+LEN	6+LEN	7+LEN
Value	0xF5	0xFA	3	8	varies		varies	varies	

Purpose: The data field of this Request Packet contains the description for an I²C transfer to be performed by the DP5. Read, Write and Combined I²C transfers are supported.

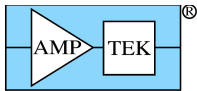
The format of the data field is listed below. The first 3 bytes are mandatory; following those are the bytes to be written, which can number from 0 to a maximum of 32. Therefore, the size of the data field (LEN) has a minimum of 3 and a maximum of 35.

Data Offset	Allowed Values	Value
0	1-127	7-bit I ² C slave address (‘SA’)
1	0-32	Number of bytes to read (‘NOBR’)
2	0-32	Number of bytes to write (‘NOBW’)
3-34	0-255	Bytes to be written (optional)

Note: The DP5 uses Slave Addresses of 0x20 and 0x4C. The optional PC5 uses Slave Addresses of 0x1F, 0x34, 0x50 and 0x6D. Transfers should not be attempted using these Slave Addresses. Also, the value in offset 0 is the 7-bit address – this value will be shifted left one bit, and the appropriate R/W bit merged in, before it’s transmitted on the I²C bus.

Response: If no errors are detected, the DP5 will respond either with an ACK OK packet (if NOBR=0), or with an I²C Response Packet with the bytes read from the I²C slave (if NOBR > 0).

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of the requested data or the ACK OK packet. If the I²C master didn’t detect the I²C ACK at the proper time, an ‘I²C ERROR’ ACK packet will be returned. A BAD PARAMETER ACK packet will be returned if there are structure problems with the request. (NOBR=NOBW=0; LEN not equal to NOBW+3; NOBR > 32; NOBW > 32; or SA = 0).



4.1.5 Request packet: “Request Mini-X Tube & Interlock table” [Mini-X2 only]

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	3	0x0B	0	0	0xFE	0x03

Purpose: This requests the Mini-X tube & interlock table.

Response: The ‘Mini-X Tube & Interlock Table’ Response Packet will be returned.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of the requested data.

4.1.6 Request packet: “Request Mini-X Warmup table” [Mini-X2 only]

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	3	0x0C	0	0	0xFE	0x02

Purpose: This requests the Mini-X warmup table.

Response: The ‘Mini-X Warmup Table’ Response Packet will be returned.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of the requested data.

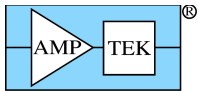
4.1.7 Request packet: “Request Mini-X Timestamp record” [Mini-X2 only]

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	3	0x0D	0	0	0xFE	0x01

Purpose: This requests the Mini-X Timestamp record. This is used to determine when the x-ray tube was last powered on, so that a warmup cycle can be recommended if necessary. This is left to the application software to determine.

Response: The ‘Mini-X Timestamp record’ Response Packet will be returned.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of the requested data.



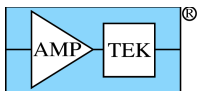
4.1.8 Request packet: “Request Mini-X Fault record” [Mini-X2 only]

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	3	0x0E	0	0	0xFE	0x00

Purpose: This requests the Mini-X Fault record. If the Mini-X detects any of a number of faults, it will shut the x-ray tube off and generate the Fault Record, which completely describes the conditions that led to the fault.

Response: The ‘Mini-X Fault record’ Response Packet will be returned.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of the requested data.



4.1.9 Request packet: “Text Configuration”

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6...5+LEN	6+LEN	7+LEN
Value	0xF5	0xFA	0x20	2	varies		varies		varies

Purpose: This sends a Text Configuration packet to the DP5, which the DP5 interprets and executes. (See the section on ASCII Commands for information on individual commands.) The commands are packed into the Data field, with the following rules:

1. Alphabetic characters must be uppercase.
2. Each command consists of a 4-character command, followed by “=”, followed by the parameter (max of 10 characters), and terminated by a semicolon (Ex: “TPEA=10;”)
3. Units may be included with the parameter, but they are ignored, and count towards the 10-character maximum length for the parameter. (Ex: “TPEA=10US;”)
4. The commands are packed together with no whitespace of any kind (tabs, spaces, CR/LF, etc.)
5. **For some commands, the ordering is important.** See the ‘ASCII Commands’ section for more discussion of this.
6. If all commands fit in the max 512 byte data field, then the configuration can be sent as a single packet. If they don’t fit, then additional Text Configuration packets can be sent. Generally, the ASCII Command “RESC=Y;” would be the first command of the first packet, and it would not appear in later packets. If additional packets are needed, ensure that commands are not chopped between packets – only complete command/parameter strings are allowed.

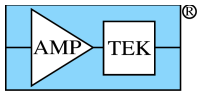
Here’s an example of the data field:

```
RESC=YES;CLCK=AUTO;TPEA=10US;TFLA=0US;CUSP=50%;
RESL=3000US;PDMD=NORM;THSL=1.00%;THFA=40;
```

Note: After the corresponding ACK Packet is sent, the configuration is written into Flash memory. This process takes anywhere from 80-400mS, during which time further packet processing is stalled. So, if a Request Packet is received immediately after the Text Configuration Request Packet is acknowledged, the corresponding ACK packet may be delayed by up to 400mS.

Response: If no errors are detected, the DP5 will respond with an ACK OK packet. This indicates that all commands were recognized and executed without error.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of the requested data. If one or more errors are detected while parsing the Text Configuration, an ‘Unknown/Bad Parameter’, ‘Unrecognized Command’, or ‘PC5 Not Present’ ACK packet will be returned, and the data field will contain the ASCII command and parameter which caused the error. If more than one error was detected, only the last error & command will be returned.



4.1.10 Request packet: “Text Configuration Readback”

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6...5+LEN	6+LEN	7+LEN
Value	0xF5	0xFA	0x20	3	varies		varies		varies

Purpose: This sends a Text Configuration Readback Request Packet to the DP5, which the DP5 uses as a ‘template’ to read back the current configuration. The data field is the same format as for the Text Configuration Request Packet – a list of ASCII command and parameters, separated by semicolons. The DP5 strips off any parameters and replaces them with the current settings for each command specified. The result is returned in the Configuration Readback Response Packet. It is possible to send a text configuration packet via the Text Configuration Request Packet (PID1=0x30 / PID2=2), and then reuse the same packet data field for the Text Configuration Readback Request Packet (PID1=0x30 / PID2=3) to read back the actual settings that the DP5 derived from the Text Configuration packet.

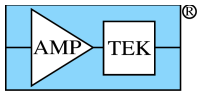
The “=” and parameter are optional for each command listed; the minimum required is the 4-character commands, separated by semicolons. There is one exception to this – the SCAI (SCA Index) command. As this is a directive to the command processor rather than a true command, the ‘=’ and parameter are required. Here is an example data field, to read back the low and high thresholds for SCA1 & SCA2: “SCAI=1;SCAL;SCAH;SCAI=2;SCAL;SCAH;”

Also note that the ‘RESC’ (Reset Configuration) has no meaning in the context of Configuration Readback – if included, it will return ‘RESC=?;’.

Any number of commands may be read back, as long as they fit within the maximum data field of 512 bytes. As of FW6.01, there is no error handling for the condition where the generated Response Packet exceeds the maximum data field size, so limiting the size of the Request Packet data field is a good idea to avoid this issue.

Response: If no errors are detected, the DP5 will respond with a Configuration Readback Response Packet (PID1=0x82 / PID2 = 7). Any command which is not recognized in the data field will return “CMDA=?;”, where ‘CMDA’ is the unknown command. [This is returned in the Response Packet with the rest of the commands.]

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of the requested data. If one or more errors are detected while parsing the Text Configuration, an ‘Unknown/Bad Parameter’, ‘Unrecognized Command’, or ‘PC5 Not Present’ ACK packet will be returned, and the data field will contain the ASCII command and parameter which caused the error. If more than one error was detected, only the last error & command will be returned.



4.1.11 Request packet: “Write 512-byte Misc Data”

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-517	518	519
Value	0xF5	0xFA	0xF0	9	2	0	Misc Data	varies	

Purpose: This commands the DP5 to store the 512-byte data field in non-volatile memory. The data can be binary, ASCII, etc. (It could contain the description or location of the unit as an ASCII string, for example.) It is read back with the “Request Misc Data” Request Packet.

Response: If no errors are detected, the DP5 will respond with ACK OK packet. Note this command requires a FLASH memory erase operation – the ACK packet could be delayed as much as 400mS from the request packet.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of ACK OK.

4.1.12 Request packet: “Select RS232 Baud Rate”

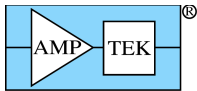
	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7	8
Value	0xF5	0xFA	0xF0	0x13	0	1	Baud Rate	varies	

Purpose: This selects the RS232 baud rate. The default is 115,200 baud. 57,600 baud or 19,200 baud can be selected; however, these setting are incompatible with the Ethernet stack. Therefore, **selecting 57,600 baud or 19,200 baud will disable the Ethernet interface. When this value is changed, the new setting will not take effect until the next power cycle.** The interpretation of the baud rate value (‘setting’) is:

0 => 115,200 baud;
1 => 57,600 baud;
2 => 19,200 baud;

Response: If no errors are detected, the DP5 will respond with ACK OK packet. If the data byte is outside the range of 0-2, then a BAD PARAMETER error is returned.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of ACK OK.

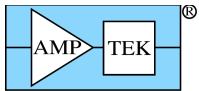


4.1.13 Request packet: “Write Mini-X Tube & Interlock Table” [Mini-X2 only]

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-99	100	101
Value	0xF5	0xFA	0xF0	0x30	0	0x5E	Tube & Interlock Table	varies	

Purpose: This request packet sends the Mini-X Tube & Interlock table, which will be stored in the non-volatile memory of the Mini-X. The Tube portion of the table defines the characteristics of the tube: acceptable HV, current and power ranges, scale factors, etc. The Interlock portion of the table specifies the voltage with which the external interlock circuit will be biased, and the acceptable resulting current range.

Data Offset	Value
0-19	Tube Part Number, ASCII, unused characters = CHR(0) for nul-terminated string
20-31	Tube Serial Number, ASCII, unused characters = CHR(0) for nul-terminated string
32	HVMIN: Tube minimum HV, in kV
33	HVMAX: Tube maximum HV, in kV
34	IMIN: Tube minimum current, in uA
35-36	IMAX (MSB, LSB): Tube maximum current, in uA
37	PMAX: Tube maximum power, in W 6.2 fixed point, i.e. floating point W x 4
38-43	TBD
44-45	HVSCALE (MSB, LSB): Scale factor for control and monitor HV analogs; kV/V, 8.8 fixed point, i.e. floating point SCALE x 256
46-47	ISCALE (MSB, LSB): Scale factor for control and monitor current analogs; uA/V, 8.8 fixed point, i.e. floating point SCALE x 256
48	Interlock set voltage, 20mV/LSB
49-50	Interlock minimum acceptable current (MSB, LSB), 12.44uA/LSB, max 50.94uA
51-52	Interlock maximum acceptable current (MSB, LSB), 12.44uA/LSB, max 50.94uA



53	VIN_MIN; Minimum acceptable tube supply voltage, in V 4.4 fixed point, i.e. floating point VIN_MIN * 16
54	VIN_MAX; Maximum acceptable tube supply voltage, in V 4.4 fixed point, i.e. floating point VIN_MAX * 16
55-61	TBD
62-93	Tube Description, ASCII, unused characters = CHR(0) for nul-terminated string

Response: If no errors are detected, the device will respond with ACK OK packet.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of ACK OK.

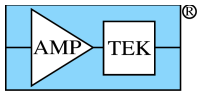
4.1.14 Request packet: “Write Mini-X Timestamp” [Mini-X2 only]

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-11	12	13
Value	0xF5	0xFA	0xF0	0x31	0	6	Timestamp	varies	

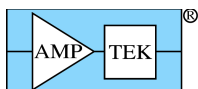
Purpose: This request packet sends the current time to the Mini-X, for the purpose of internally tagging when the tube is switched off. The Mini-X stores both the timestamp, and the number of seconds between the timestamp being received and the tube being switched off. Combined (by the application software), these provide the time and date that the tube was last on, for the purpose of determining whether a warmup cycle would be beneficial.

Data Offset	Value
0	Current Month, 1-12
1	Current Day, 1-31
2	Current Year – 2000, i.e. 2018 => 18
3	Current Hour, 0-23
4	Current Minute, 0-59
5	Current Second, 0-59

Response: No limit checking is done on the individual bytes, so if the request packet is the proper length, the device will respond with ACK OK packet.



Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of ACK OK.

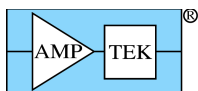


4.1.15 Request packet: “Write Mini-X Warmup Table” [Mini-X2 only]

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-53	54	55
Value	0xF5	0xFA	0xF0	0x32	0	0x30	Warmup Table	varies	

Purpose: This request packet sends two warmup tables to the Mini-X, to be stored in non-volatile memory. Each table consists of up to 6 steps, with voltage (kV), current (uA) and time (s) for each step. The first table is intended for use if the tube has been off for more than 24 hours. The 2nd table is intended for use on initial turn-on of the tube, or if the tube has been off for 30 days or more.

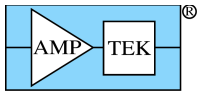
Data Offset	Value
0	Daily warmup, step 1, voltage (kV)
1	Daily warmup, step 1, current (uA)
2-3	Daily warmup, step 1, time (s), [MSB, LSB]
4	Daily warmup, step 2, voltage (kV)
5	Daily warmup, step 2, current (uA)
6-7	Daily warmup, step 2, time (s), [MSB, LSB]
8	Daily warmup, step 3, voltage (kV)
9	Daily warmup, step 3, current (uA)
10-11	Daily warmup, step 3, time (s), [MSB, LSB]
12	Daily warmup, step 4, voltage (kV)
13	Daily warmup, step 4, current (uA)
14-15	Daily warmup, step 4, time (s), [MSB, LSB]
16	Daily warmup, step 5, voltage (kV)
17	Daily warmup, step 5, current (uA)
18-19	Daily warmup, step 5, time (s), [MSB, LSB]
20	Daily warmup, step 6, voltage (kV)
21	Daily warmup, step 6, current (uA)
22-23	Daily warmup, step 6, time (s), [MSB, LSB]
24	Monthly warmup, step 1, voltage (kV)
25	Monthly warmup, step 1, current (uA)



26-27	Monthly warmup, step 1, time (s), [MSB, LSB]
28	Monthly warmup, step 2, voltage (kV)
29	Monthly warmup, step 2, current (uA)
30-31	Monthly warmup, step 2, time (s), [MSB, LSB]
32	Monthly warmup, step 3, voltage (kV)
33	Monthly warmup, step 3, current (uA)
34-35	Monthly warmup, step 3, time (s), [MSB, LSB]
36	Monthly warmup, step 4, voltage (kV)
37	Monthly warmup, step 4, current (uA)
38-39	Monthly warmup, step 4, time (s), [MSB, LSB]
40	Monthly warmup, step 5, voltage (kV)
41	Monthly warmup, step 5, current (uA)
42-43	Monthly warmup, step 5, time (s), [MSB, LSB]
44	Monthly warmup, step 6, voltage (kV)
45	Monthly warmup, step 6, current (uA)
46-47	Monthly warmup, step 6, time (s), [MSB, LSB]

Response: No limit checking is done on the individual bytes, so if the request packet is the proper length, the device will respond with ACK OK packet.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of ACK OK.



4.1.16 Request packet: “Initiate Mini-X Warmup Sequence” [Mini-X2 only]

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7	8
Value	0xF5	0xFA	0xF0	0x33	0	1	Daily/Monthly	varies	

Purpose: This commands the Mini-X to initiate a warmup sequence. The default daily warmup sequence has a total duration of 30 seconds, and the default monthly warmup sequence has a total duration of 30 minutes. The warmup sequence can be cancelled by sending HVSE/CUSE ascii commands.

- 0 => Initiate daily warmup sequence;
- 1 => Initiate monthly warmup sequence;

Response: If no errors are detected, the DP5 will respond with ACK OK packet. If the data byte is outside the range of 0-1, then a BAD PARAMETER error is returned.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of ACK OK.

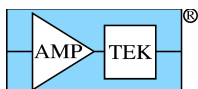
4.1.17 Request packet: “Comm test - Request ACK packet”

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	0xF1	0..15	0	0	varies	

Purpose: This Request Packet is handy for software developers – it requests that a specific ACK packet be returned in response to the request. (It’s useful for testing application error handling – this is much easier than trying to force the various error conditions.) The value of PID2 is used as PID2 in the ACK packet; i.e. sending the Request Packet PID1=0xF1/**PID2=4** will yield a ‘Checksum Error” ACK packet (PID1=0xFF/**PID2=4**).

Response: An ACK packet, with PID2 equal to the PID2 of the Request Packet. This allows all types of ACK packets to be returned.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of the requested ACK type.



4.1.18 Request packet: “Comm test - Echo packet”

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	DATA FIELD	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6...5+LEN	6+LEN	7+LEN
Value	0xF5	0xFA	0xF1	0x7F	varies		Data to be echoed	varies	

Purpose: This Request Packet is handy for software developers – it requests that the DP5 return a Response Packet, whose length and data field are the same as those of this Request Packet – essentially, it echoes the data field in a Response Packet.

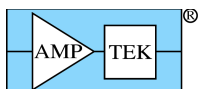
Response: If no errors are detected, the DP5 will respond with “Comm test - Echo packet” Response Packet (PID1=0xF1/PID2=0x7F).

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of the Response Packet.

4.2 Mini-X Response Packets

Description	Support*	PID1	PID2	LEN MSB	LEN LSB	Data (optional)
Mini-X Status Packet	ALL	0x80	2	0	0x40	See description
512-byte misc data packet	ALL	0x82	2	2	0	See description
Configuration readback packet	ALL	0x82	7	varies		
Netfinder packet	ALL	0x82	8	varies		See Section 3.3
I ² C Read Data	ALL	0x82	9	varies		See description
Mini-X Tube & Interlock Table	ALL	0x82	13	0	0x5E	See description
Mini-X Warmup Table	ALL	0x82	14	0	0x30	See description
Mini-X Timestamp Record	ALL	0x82	15	0	0x0F	See description
Mini-X Fault Record	ALL	0x82	16	0	0xE8	See description
Comm test - Echo packet	ALL	0x8F	0x7F	varies		Echo data

* Initial firmware release which supports this packet type



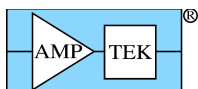
4.2.1 Response packet: “Mini-X Status Packet” [Mini-X only]

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data Field	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-69	70	71
Value	0xF5	0xFA	0x80	2	0	0x40	Status data	varies	

Purpose: This response packet is returned in response to a ‘Request Status Packet’ Request Packet. Internally, the analog monitors are updated at a 10Hz rate (i.e., every 100mS.)

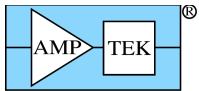
Data Format: The data field contains 64 bytes of status data. The format is given below.

Offset	Data bits	Allowed value	Description	Notes
0	D7-D0	0-255	Mini-X Serial Number LSB	
1	D7-D0	0-255	Mini-X Serial Number byte 2	
2	D7-D0	0-255	Mini-X Serial Number byte 3	
3	D7-D0	0-255	Mini-X Serial Number MSB	
4	D7-D4 D3-D0	0-15 0-15	Firmware Major Version Firmware Minor Version	Ex: 0x69 = v6.09.xx
5	D7-D4 D3-D0	0-15 0-15	TBD Firmware Build Number	Ex: 0x0B = vxx.xx.11
6	D7-D0	0-255	Tube HV Monitor LSB	Tube HV (kV) = ((MSB * 256) + LSB) / 1000 * HVSCALE [HVSCALE is from Tube Table]
7	D7-D4 D3-D0	0 0-15	Tube HV Monitor MSB	
8	D7-D0	0-255	Tube current Monitor LSB	Tube Current (uA) = ((MSB * 256) + LSB) / 1000 * ISCALE [ISCALE is from Tube Table]
9	D7-D4 D3-D0	0 0-15	Tube Current Monitor MSB	
10	D7-D0	0-255	Interlock Current Monitor LSB	Interlock Current (mA) = ((MSB * 256) + LSB) / 80.39
11	D7-D4 D3-D0	0 0-15	Interlock Current Monitor MSB	
12	D7-D0	0-99	Tube Power Supply Monitor LSB	Tube Power Supply (V) = ((MSB * 256) + LSB) / 306.7
13	D7-D4 D3-D0	0 0-15	Tube Power Supply Monitor MSB	
14	D7-D0	0-255	Mini-X Power Supply Monitor LSB	Mini-X Power Supply (V) = ((MSB * 256) + LSB) / 306.7
15	D7-D4 D3-D0	0 0-15	Mini-X Power Supply Monitor MSB	
16	D7 D6 D5 D4 D3-D0	0 1 0 0 1 0 1 0-15	Tube HV is disabled Tube HV is enabled TBD Tube power is off Tube power is on Accessory is off Accessory is on Interlock/fault status =====>	Interlock/fault status: 0 = Interlock closed (normal) 1 = Interlock open 2 = Interlock shorted 3 = VIN undervoltage 4 = VIN overvoltage 5 = HV_MON below limit 6 = HV_MON over limit 7 = I_MON below limit 8 = I_MON over limit 9 = USB/RS232 disconnected



Mini-X2 Programming Guide Rev B2 Preliminary

				10 = no communication 11 = warmup seq complete 12-15 = TBD
17	D7-D0	0-255	Mini-X temperature	8-bit signed value, in C
18	D7	0	Speaker is enabled	'Previous fault' encoding is same as 'Interlock/Fault status' in offset 16
		1	Speaker is disabled	
	D6	0	Fault checks are enabled	
		1	Fault checks are disabled	
	D5	0	Limit checks are enabled	
		1	Limit checks are disabled	
	D4	0	Tube analog control	
		1	Tube I2C control	
	D3-D0	0-15	Previous Fault	
19	D7	0	Warmup sequence not running	
		1	Warmup sequence running	
	D6-D4	TBD	TBD	
	D3-D0	0-5	Daily warmup step 1-6	
		6-11	Monthly warmup step 1-6	
20	D7-D0	0-255	Warmup step time remaining LSB	Remaining time (s) = MSB*256 + LSB
21	D7-D0	0-255	Warmup step time remaining MSB	
22	D7-D0	0-255	Total tube runtime, LSB	Runtime (s) = (MSB * 2^24) + (Byte2 * 2^16) + (Byte3 * 2^8) + LSB
23	D7-D0	0-255	Total tube runtime, byte 2	
24	D7-D0	0-255	Total tube runtime, byte 3	
25	D7-D0	0-255	Total tube runtime, MSB	
26	D7-D0	0-255	HVSCALE, MSB	8.8 fixed point; HVSCALE = (MSB * 256 + LSB) / 256
27	D7-D0	0-255	HVSCALE, LSB	
28	D7-D0	0-255	ISCALE, MSB	8.8 fixed point; ISCALE = (MSB * 256 + LSB) / 256
29	D7-D0	0-255	ISCALE, LSB	
30-63			TBD	



4.2.2 Response packet: “512-byte Misc Data Packet”

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data Field	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-517	518	519
Value	0xF5	0xFA	0x82	2	2	0	Misc data	Varies	

Purpose: This response packet is returned in response to a ‘Request Misc Data’ packet (PID1=3, PID2=2). [See the ‘Write Misc Data’ Request Packet for information on writing the Misc Data.]

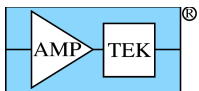
Data Format: The data field contains 512 bytes, which is the Misc Data programmed via a ‘Write Misc Data’ request packet. The data can be binary, ASCII, etc. This is stored in non-volatile memory, so it remains after the power is cycled.

4.2.3 Response packet: “Configuration Readback”

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data Field	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6...5+LEN	6+LEN	7+LEN
Value	0xF5	0xFA	0x82	7	varies		Configuration readback (ASCII)	varies	

Purpose: This response packet is returned in response to a ‘Configuration Readback’ Request Packet (PID1=0x20, PID2=3). The data field contains ASCII commands and parameters, separated by semicolons.

Data Format: The data field will contain the current ASCII configuration – it will have all the commands that were sent with the “Configuration Readback” Request Packet, except the parameters will have been replaced with the actual settings in use by the DP5. See the “Text Configuration” Request Packet, “Configuration Readback” Request packet, and the section on ASCII Commands for more details.



4.2.4 Response packet: “Netfinder Packet”

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	DATA FIELD	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6...5+LEN	6+LEN	7+LEN
Value	0xF5	0xFA	0x82	8	varies		Netfinder data	varies	

Purpose: This is used to return the ‘Netfinder’ packet via the normal communications link. [Normally, the Netfinder data is returned by a dedicated TCP/IP socket.]

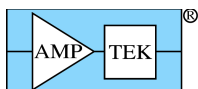
Data Format: The data field is formatted the same as an independent Netfinder packet, as documented in section 3.3 NetFinder (Mini-X2).

4.2.5 Response packet: “I²C Read Data”

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	DATA FIELD	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6...5+LEN	6+LEN	7+LEN
Value	0xF5	0xFA	0x82	9	varies		I²C read data	varies	

Purpose: This is used to return the data read by the ‘I²C Transfer’ Request Packet. If the Number of Bytes to Read (‘NOBR’) in the ‘I²C Transfer’ Request Packet is nonzero, and the I²C transfer was successful, then the ‘I²C Read Data’ Response Packet is returned, with the data field containing the bytes read by the transfer. [Note that if NOBR = 0, the DP5 will return an ACK OK packet rather than an ‘I²C Read Data’ packet with LEN=0.]

Data Format: The LEN value will be the same as NOBR, and the data field will contain the bytes read, in the order that they were read.



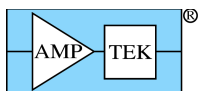
4.2.6 Response packet: “Mini-X Tube & Interlock Table” [Mini-X2 only]

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-99	100	101
Value	0xF5	0xFA	0x82	0x0D	0	0x5E	Tube & Interlock Table	varies	

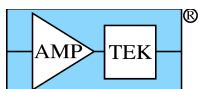
Purpose: This packet is sent by the Mini-X in response to a ‘Request Mini-X Tube & Interlock Table’ Request Packet (PID1/2 = 0x03/0x0B). The Tube portion of the table defines the characteristics of the tube: acceptable HV, current and power ranges, scale factors, etc. The Interlock portion of the table specifies the voltage with which the external interlock circuit will be biased, and the acceptable resulting current range. The data field for this packet has the same format as the ‘Write Mini-X Tube & Interlock’ Request Packet (PID1/2 = 0xF0/0x30)

Data Format: The data field contains 94 bytes. The format for the data field is given below:

Data Offset	Value
0-19	Tube Part Number, ASCII, unused characters = CHR(0) for nul-terminated string
20-31	Tube Serial Number, ASCII, unused characters = CHR(0) for nul-terminated string
32	HVMIN: Tube minimum HV, in kV
33	HVMAX: Tube maximum HV, in kV
34	IMIN: Tube minimum current, in uA
35-36	IMAX (MSB, LSB): Tube maximum current, in uA
37	PMAX: Tube maximum power, in W 6.2 fixed point, i.e. floating point W x 4
38-43	TBD
44-45	HVSCALE (MSB, LSB): Scale factor for control and monitor HV analogs; kV/V, 8.8 fixed point, i.e. floating point SCALE x 256
46-47	ISCALE (MSB, LSB): Scale factor for control and monitor current analogs; uA/V, 8.8 fixed point, i.e. floating point SCALE x 256
48	Interlock set voltage, 20mV/LSB
49-50	Interlock minimum acceptable current (MSB, LSB), 12.44uA/LSB, max 50.94uA



51-52	Interlock maximum acceptable current (MSB, LSB), 12.44uA/LSB, max 50.94uA
53	VIN_MIN; Minimum acceptable tube supply voltage, in V 4.4 fixed point, i.e. floating point VIN_MIN * 16
54	VIN_MAX; Maximum acceptable tube supply voltage, in V 4.4 fixed point, i.e. floating point VIN_MAX * 16
55-61	TBD
62-93	Tube Description, ASCII, unused characters = CHR(0) for nul-terminated string



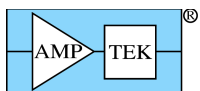
4.2.7 Response packet: “Mini-X Warmup Table” [Mini-X2 only]

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-53	54	55
Value	0xF5	0xFA	0x82	0x0E	0	0x30	Warmup Table	varies	

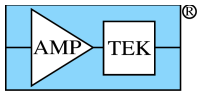
Purpose: This request packet is sent in response to the ‘Request Mini-X Warmup Table’ Request Packet (PID1/2 = 0x03/0x0C). This response packet contains both warmup tables from the Mini-X. Each table consists of up to 6 steps, with voltage (kV), current (uA) and time (s) for each step. The first table is intended for use if the tube has been off for more than 24 hours. The second table is intended for use on initial turn-on of the tube, or if the tube has been off for 30 days or more.

Data Format: The data field contains 48 bytes. The format for the data field is given below:

Data Offset	Value
0	Daily warmup, step 1, voltage (kV)
1	Daily warmup, step 1, current (uA)
2-3	Daily warmup, step 1, time (s), [MSB, LSB]
4	Daily warmup, step 2, voltage (kV)
5	Daily warmup, step 2, current (uA)
6-7	Daily warmup, step 2, time (s), [MSB, LSB]
8	Daily warmup, step 3, voltage (kV)
9	Daily warmup, step 3, current (uA)
10-11	Daily warmup, step 3, time (s), [MSB, LSB]
12	Daily warmup, step 4, voltage (kV)
13	Daily warmup, step 4, current (uA)
14-15	Daily warmup, step 4, time (s), [MSB, LSB]
16	Daily warmup, step 5, voltage (kV)
17	Daily warmup, step 5, current (uA)
18-19	Daily warmup, step 5, time (s), [MSB, LSB]
20	Daily warmup, step 6, voltage (kV)
21	Daily warmup, step 6, current (uA)
22-23	Daily warmup, step 6, time (s), [MSB, LSB]
24	Monthly warmup, step 1, voltage (kV)



25	Monthly warmup, step 1, current (uA)
26-27	Monthly warmup, step 1, time (s), [MSB, LSB]
28	Monthly warmup, step 2, voltage (kV)
29	Monthly warmup, step 2, current (uA)
30-31	Monthly warmup, step 2, time (s), [MSB, LSB]
32	Monthly warmup, step 3, voltage (kV)
33	Monthly warmup, step 3, current (uA)
34-35	Monthly warmup, step 3, time (s), [MSB, LSB]
36	Monthly warmup, step 4, voltage (kV)
37	Monthly warmup, step 4, current (uA)
38-39	Monthly warmup, step 4, time (s), [MSB, LSB]
40	Monthly warmup, step 5, voltage (kV)
41	Monthly warmup, step 5, current (uA)
42-43	Monthly warmup, step 5, time (s), [MSB, LSB]
44	Monthly warmup, step 6, voltage (kV)
45	Monthly warmup, step 6, current (uA)
46-47	Monthly warmup, step 6, time (s), [MSB, LSB]



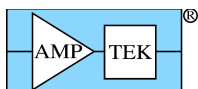
4.2.8 Response packet: “Mini-X Timestamp Record” [Mini-X2 only]

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-20	21	22
Value	0xF5	0xFA	0x82	0x0F	0	0x0F	Timestamp Record	varies	

Purpose: This response packet is sent in response to a ‘Request Mini-X Timestamp Record’ Request Packet (PID1/2 = 0x03/0x0D). By adding the lag (in seconds) to the timestamp, the application software can determine the time and date that the tube was last on, for the purpose of determining whether a warmup cycle would be beneficial. The total runtime of the tube is also present in the timestamp record.

This packet should be read when the application first connects to the Mini-X, before a new timestamp overwrites it.

Data Offset	Value	Notes
0	Flags	
1	Current Month, 1-12	
2	Current Day, 1-31	
3	Current Year – 2000, i.e. 0x12 => 2018	
4	Current Hour, 0-23	
5	Current Minute, 0-59	
6	Current Second, 0-59	
7	Timetag Lag, MSB	Timetag Lag (s) = (MSB*2 ²⁴) + (Byte2 * 2 ¹⁶) + (Byte3 * 2 ⁸) + LSB
8	Timetag Lag, Byte 2	
9	Timetag Lag, Byte 3	
10	Timetag Lag, LSB	
11	Total Runtime, MSB	Total Runtime (s) = (MSB*2 ²⁴) + (Byte2 * 2 ¹⁶) + (Byte3 * 2 ⁸) + LSB
12	Total Runtime, Byte 2	
13	Total Runtime, Byte 3	
14	Total Runtime, LSB	



4.2.9 Response packet: “Mini-X Fault Record” [Mini-X2 only]

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-237	238	239
Value	0xF5	0xFA	0x82	0x10	0	0xE8	Fault Record	varies	

Purpose: This response packet is sent in response to a ‘Request Mini-X Fault Record’ Request Packet (PID1/2 = 0x03/0x0E.) Embedded in this packet are a Timestamp record, a status packet, a tube table, and the HV and current settings that were in effect when the fault occurred.

This packet can be read by the application software for diagnostic purposes, to learn about why the x-ray tube was unexpectedly shut off.

Data Offset	Value	Notes
0	Flags	TBD
1-15	Timestamp record	See section 4.2.8 for encoding.
16-17	Timestamp record checksum (MSB, LSB)	16-bit checksum; can be ignored
18-81	Status packet	See section 4.2.1 for encoding
82-177	Tube table	See section 4.2.6 for encoding
178-225	Warmup table	See section 4.2.7 for encoding
224-225	Tube table + Warmup table checksum (MSB, LSB)	16-bit checksum; can be ignored
226-227	Tube HV setting	
228-229	Tube Current setting	
230-231	Fault record checksum (MSB, LSB)	16-bit checksum for entire fault record; can be ignored

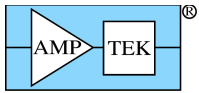
4.2.10 Response packet: “Comm test - Echo packet”

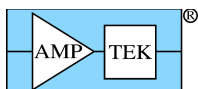
	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	DATA FIELD	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6...5+LEN	6+LEN	7+LEN
Value	0xF5	0xFA	0x8F	0x7F	varies		Echo data	varies	

Purpose: This Response Packet is sent in response to the Request Packet of the same name. The LEN & DATA fields are the same as in the Request Packet.

Data Format: Whatever data is in the data field of the Request Packet is copied, byte-for-byte, into the data field of the Response packet.

Amptek Inc.





4.3 Acknowledge Packets

Table 1– Text highlighted in Blue is new/modified since Rev A6 of the Programmer’s Guide

Description	Support*	PID1	PID2	LEN MSB	LEN LSB	Data (optional)
OK	ALL	0xFF	0	0	0	
Sync error	ALL	0xFF	1	0	0	
PID error	ALL	0xFF	2	0	0	
LEN error	ALL	0xFF	3	0	0	
Checksum error	ALL	0xFF	4	0	0	
Bad parameter	ALL	0xFF	5	varies		ASCII command
Bad hex record (structure/chksum)	ALL	0xFF	6	0	0	
Unrecognized command	ALL	0xFF	7	varies		ASCII command
FPGA error (not initialized)	ALL	0xFF	8	0	0	
CP2201 not found	ALL	0xFF	9	0	0	
Scope data not available (not triggered)	ALL	0xFF	0x0A	0	0	
PC5 not present	ALL	0xFF	0x0B	varies		ASCII command
OK + Interface sharing request	ALL	0xFF	0x0C	0	0	
Busy - another interface is in use		0xFF	0x0D	0	0	
I2C error	6.01	0xFF	0x0E	0	0	
OK + FPGA upload address	6.06.04	0xFF	0x0F	0	3	Intel hex record: ADDR MSB, LSB, record type
Feature not supported by this FPGA version	6.06.05	0xFF	0x10	0	0	
Calibration data not present	6.07.02	0xFF	0x11	0	0	

* Initial firmware release which supports this packet type

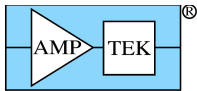
4.3.1 Acknowledge packet: “OK”

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	0xFF	0	0	0	0xFD	0x12

Purpose: This ACK packet indicates that the Request Packet was received without error, that its contents were recognized, and that the requested action was carried out. This ACK packet is only returned by Request Packets which do not request data.

4.3.2 Acknowledge packet: “OK, with Interface Sharing Request”

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	0xFF	0x0C	0	0	0xFD	0x06



Purpose: This ACK packet has the same meaning as the “OK” ACK packet. In addition, another computer or interface is requesting that activity be paused so that it can gain access to the DP5. To allow it access, the application should stop sending traffic to the DP5. To maintain control (and deny it access), continue sending Request Packets. (Send ‘Interface keep-alive’ packets periodically, if no other periodic traffic is occurring. See the Interface section for more details.)

4.3.3 Acknowledge packet: “Sync Error”

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	0xFF	1	0	0	0xFD	0x11

Purpose: This ACK packet indicates that the sync bytes in Request Packet were not correct, and therefore, the Request Packet was rejected.

Note: This error condition can only be generated via USB or Ethernet; RS232 uses the sync bytes to locate a packet in the RS232 data stream. In the absence of the sync byte pattern, no Request Packet is detected and thus no ACK packet can be returned.

4.3.4 Acknowledge packet: “PID Error”

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	0xFF	2	0	0	0xFD	0x10

Purpose: This ACK packet indicates that PID1 & PID2 combination is not recognized as a valid Request Packet, and therefore, the Request Packet was rejected.

4.3.5 Acknowledge packet: “LEN Error”

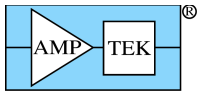
	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	0xFF	3	0	0	0xFD	0x0F

Purpose: This ACK packet indicates that the LEN field of the Request Packet was not consistent with Request Packet type defined by the PID1 & PID2 combination. It is not recognized as a valid Request Packet, and therefore, the Request Packet was rejected.

4.3.6 Acknowledge packet: “Checksum Error”

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	0xFF	4	0	0	0xFD	0x0E

Purpose: This ACK packet indicates that the checksum of the Request Packet was incorrect, and therefore, the Request Packet was rejected.



4.3.7 Acknowledge packet: “Bad Parameter”

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	DATA	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6...5+LEN	6+LEN	7+LEN
Value	0xF5	0xFA	0xFF	5	0	varies	ASCII Command Echo	varies	

4.3.8 Acknowledge packet: “Unrecognized Command”

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	DATA	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6...5+LEN	6+LEN	7+LEN
Value	0xF5	0xFA	0xFF	7	0	varies	ASCII Command Echo	varies	

4.3.9 Acknowledge packet: “PC5 Not Present”

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	DATA	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6...5+LEN	6+LEN	7+LEN
Value	0xF5	0xFA	0xFF	0x0B	0	varies	ASCII Command Echo	varies	

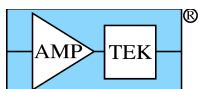
Purpose: These ACK packets are in response to ASCII command errors – the data field will contain the ASCII command and parameter which caused the error. “Bad Parameter” means that the parameter isn’t recognized or exceeds the range of the command. “Unrecognized Command” means that the 4-character command isn’t recognized. “PC5 Not Present” is returned if a PC5 is not mated to the DP5, and a command requiring a PC5 is sent. (i.e. “HVSE”, Set High Voltage.) [A ‘Bad Parameter’ ACK packet may also be returned for a malformed I2C Request Packet, in which case LEN=0.]

If an incomplete or garbled command is returned in the data field, it may mean that the ASCII Configuration Packet has structural issues. (Disallowed whitespace, missing semicolon, etc.)

4.3.10 Acknowledge packet: “Bad Hex Record”

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	0xFF	6	0	0	0xFD	0x0C

Purpose: This ACK packet applies to microcontroller or FPGA upload packets (PID1=0x30, PID2=2 or 7.) It means that the hex record contained in the data field of the Request Packet had a checksum or other structural error.



4.3.11 Acknowledge packet: “FPGA Error”

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	0xFF	8	0	0	0xFD	0x0A

Purpose: This ACK packet indicates that the FPGA failed initialization, and is returned in response to “Request Spectrum” (PID1=2) packets.

4.3.12 Acknowledge packet: “CP2201 Not Found”

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	0xFF	9	0	0	0xFD	0x09

Purpose: This ACK packet indicates that the “Set Ethernet Settings” (PID1=0xF0, PID2=0x11) was received, but an Ethernet controller was not detected on the DP5. [Some early DP5s were manufactured without an Ethernet controller.]

4.3.13 Acknowledge packet: “Scope Data Not Available”

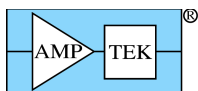
	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	0xFF	0x0A	0	0	0xFD	0x08

Purpose: This ACK packet indicates that a “Send Scope Data” (PID1=3, PID2=1 or 3) was received, but the digital oscilloscope hasn’t triggered, so no data is available. [The digital oscilloscope must be armed, and then a trigger must occur for data to be available.]

4.3.14 Acknowledge packet: “I²C Error”

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	0xFF	0x0E	0	0	0xFD	0x04

Purpose: This ACK packet indicates that the I²C Bus Master attempted a transfer as described by the ‘I²C Transfer’ Request Packet, but no I²C ACK was detected from an I²C Slave.



4.3.15 Acknowledge packet: “Feature not supported by this FPGA version”

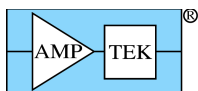
	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	0xFF	0x10	0	0	0xFD	0x02

Purpose: This ACK packet indicates that a Request Packet has been recognized as valid by the firmware, but it is not supported by the installed FPGA version. Update the FPGA to the latest FP version.

4.3.16 Acknowledge packet: “Calibration data not present”

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	0xFF	0x11	0	0	0xFD	0x01

Purpose: This ACK packet indicates that a Request Packet has been recognized as valid by the firmware, but that the requested calibration data has not been previously programmed into the device.



5 ASCII Commands

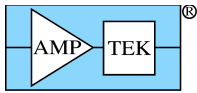
In the following section, the format is specified for all the ASCII commands recognized by the DP5. These commands are sent to the DP5 in the ‘Text Configuration’ Request Packet (PID1=0x20, PID2=2). The commands can be sent singly, but most often, they are grouped together into a single Text Configuration packet so that the entire configuration can be sent at once.

Generally, when the entire configuration is sent at once, the first command should be “RESC=Y”. This resets the configuration to a known state before applying the configuration commands. This is not automatically done because it may be possible that the required configuration is larger than can be sent in a single ‘Text Configuration’ Request Packet. If this is the case, then “RESC=Y” and as many commands as will fit should be sent in the first Text Configuration packet, with the remaining commands in later packets. Also, it may be desirable to send individual commands to fine-tune the configuration.

This is discussed in the ‘Text Configuration’ Request Packet definition, but it’s worth mentioning again: all ASCII characters must be upper-case; no whitespace is allowed (i.e. no spaces, tabs, CR-LF, etc.); and the commands are separated by semicolons.

Table 2 – ASCII Command Summary [Mini-X2]

Description	Support*	Command	Parameter	Units	Default	ORDER	Limits
Set Mini-X2 Tube Current	ALL	CUSE	[###.###]	uA	0	N/A	Set by Mini-X2 Tube Table
Mini-X2 Fault Override	ALL	FAOR	[ON OF{F}]			N/A	
Set Mini-X2 Tube High Voltage	ALL	HVSE	[##.###]	kV	0	N/A	Set by Mini-X2 Tube Table
Mini-X2 Limit Override	ALL	LIOR	[ON OF{F}]			N/A	
Mini-X2 Speaker Control	ALL	VOLU	[ON OF{F}]			N/A	



5.1.1 CUSE [Mini-X2] – Set Tube Current

Command: CUSE
Parameter: [###.###]
Units: uA
Default: 0
Range: Range is determined by contents of Mini-X2 Tube Table (see Section)
If Limit checking is overridden ('LIOR=OFF', see section xx), then any value is accepted. The maximum is 255.996.

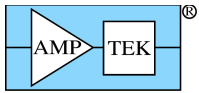
Dependencies: none

Supported: All

Description: CUSE sets the x-ray tube current. The minimum and maximum values are specified in the Mini-X2 Tube Table. Note that if either HVSE or CUSE (Current Set) are set to 0, the Mini-X2 firmware will set both to 0. If the x-ray tube was not on, then the tube will be powered on and enabled when valid HVSE and CUSE commands are received, assuming all interlock conditions are satisfied.

Errors: An INVALID PARAMETER error will be returned if the specified current is outside the range listed in the Mini-X2 tube table, unless limit checking is overridden (LIOR=OFF').

Example: CUSE=50; // Set the x-ray tube current to 50uA



5.1.2 FAOR [Mini-X2]- Set Fault Override

Command: FAOR
Parameter: [ON|OF{F}]
Units: none
Default: OFF

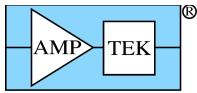
Dependencies: none
Supported: ALL
Related: LIOR

Description: FAOR is used to turn fault detection on/off. With FAOR=OFF (i.e. fault monitoring is not overridden), if either the x-ray tube HV or current monitor is not within 2% of the nominal value, then the x-ray tube will be turned off, a fault record will be generated, and the speaker will produce a warning warble. With FAOR=ON (i.e. fault monitoring is overridden), the monitors are not checked for validity.

The state of this command is 'remembered' while the Mini-X2 controller is powered on, i.e. the FAOR command state will be in effect until another FAOR command is received, or the Mini-X2 controller is powered off.

Errors: An INVALID PARAMETER error will be returned if the parameter is not 'ON' or 'OFF'.

Example: FAOR=ON; // Ignore faults



5.1.3 HVSE [Mini-X2] – Set Tube High Voltage

Command: HVSE

Parameter: [###.###]

Units: kV

Default: 0

Range: Range is determined by contents of Mini-X2 Tube Table (see Section)
If Limit checking is overridden ('LIOR=OFF', see section xx), then any value is accepted. The maximum is 255.996.

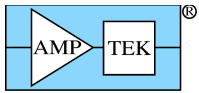
Dependencies: none

Supported: All

Description: HVSE sets the x-ray tube high voltage. The minimum and maximum values are specified in the Mini-X2 Tube Table. Note that if either HVSE or CUSE (Current Set) are set to 0, the Mini-X2 firmware will set both to 0. If the x-ray tube was not on, then the tube will be powered on and enabled when valid HVSE and CUSE commands are received, assuming all interlock conditions are satisfied.

Errors: An INVALID PARAMETER error will be returned if the specified HV is outside the range listed in the Mini-X2 tube table, unless limit checking is overridden (LIOR=OFF').

Example: HVSE=50; // Set the x-ray tube to 50kV



5.1.4 LIOR [Mini-X2]- Set Limit Override

[DP5]

Command: LIOR
Parameter: [ON|OF{F}]
Units: none
Default: OFF

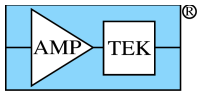
Dependencies: none
Supported: ALL
Related: FAOR

Description: LIOR is used to turn limit detection on/off. With LIOR=OFF (i.e. command limit checking is not overridden), the parameters for HVSE and CUSE are checked to ensure they are within the limits set in the Mini-X2 Tube Table. (If not, the parameter is set to 0, and an error is returned.) With LIOR=ON (i.e. command limit checking is overridden), the command parameters are not checked for validity. **WARNING: TUBE DAMAGE COULD RESULT FROM OPERATING THE TUBE OUTSIDE OF RECOMMENDED LIMITS. THIS IS AN ADVANCED FEATURE, AND SHOULD ONLY BE USED BY KNOWLEDGEABLE PERSONEL.**

The state of this command is ‘remembered’ while the Mini-X2 controller is powered on, i.e. the LIOR command state will be in effect until another LIOR command is received, or the Mini-X2 controller is powered off.

Errors: An INVALID PARAMETER error will be returned if the parameter is not ‘ON’ or ‘OFF’.

Example: LIOR=ON; // Override limit checking



5.1.5 VOLU [Mini-X2] – Mini-X2 Speaker Control

Command: VOLU

Parameter: [ON|OF{F}]

Range: ON or OF{F}

Default: OFF

Dependencies: none

Supported: ALL

Description: This enables or disables the Mini-X2 warning speaker. With VOLU=ON, the Mini-X2 will make a once-per-second beep anytime the tube is producing x-rays. Regardless of the state of this command, the Mini-X2 speaker will warble when a fault occurs (interlock failure while tube is on, monitor out of limit, etc.)

Errors: An INVALID PARAMETER error will be returned if the parameter is not ON or OF{F}.

Example: VOLU=ON; // Enable the Mini-X2 speaker