X-ray

MXDPP-50 Digital Pulse Processor (DPP)

Programmers Reference Manual

DET-MAN-1002, Rev A

MOXTEK®

452 West 1260 North
Orem, UT., 84057 USA

Table of Contents

Introduction

This manual provides programming information for Moxtek Digital Pulse Processor.

- MXDPP-50 Box
- MXDPP-50 OEM Card Stack

Use this manual in junction with the MXDPP-50 Operation Manual and the included example code to write custom code for controlling and interfacing with the Moxtek MXDPP-50 Digital Pulse Processor.

Software Developers Kit (SDK)

SDK Contents

- MXDPP-50 Digital Pulse Processor (DPP) Programmers Reference Manual
- Example Code

Example Code

The Software Development Kit includes the following fully documented example programs:

- DPP50 VCP VB.NET (Written in Microsoft Visual Studio 2010)
- DPP50 D2XX VB.NET (Written in Microsoft Visual Studio 2010)
- LabVIEW VCP

The following example programs will be available in the future, contact Moxtek for an expected availability date.

- DPP50 VCB C#.NET
- DPP50 D2XX C#.NET
- LabVIEW D2XX

VCP vs. D2XX

The MXDPP-50 uses a FTDI FT232 USB Transceiver chip to provide USB communication. The drivers allow two different modes of communication, Virtual Com Port (VCP) and Direct Connect using FTDIs D2XX dll libraries.

Virtual Com Port (VCP)

The VCP driver creates a virtual com port on the user's computer that allows the user to treat the USB device as a simple serial port. The advantages are that it is simple to use and standard windows serial port controls can be used. The main disadvantage is that the DPP tuning factors cannot be read from the DPP using the Virtual Com Port. **Because of this Moxtek recommends using the Direct Connect method instead**.

Direct Connect (D2XX)

The Direct Connect uses FTDI's D2XX.dll to directly communicate with the FTDI chip. The main advantage of using the Direct Connect over the Virtual Com Port is that the DPP Tuning Factors can be read from the DPP through the Direct Connect and not through the Virtual Com Port.

Tuning Factors

The Tuning Factors are tuned at the factory and saved to each DPP. Due to different tolerances in electronic components each individual DPP must be tuned to compensate for electronic part tolerances. The Tuning Factors are used to calculate the Preprocessor Time Constant and the Equalization Factor. The Tuning Factors can only be read from the DPP using the Direct Connect (D2XX) method. The Tuning Factors are stored in the USB device description field in the following format:

MXDPP-50 (x.xxx,n.nnn)

Where:

x.xxx = EQ slope (See Tuning Factors)
n.nnn = EQ Offset Tuning

See the D2XX Commands section for a description of how to read the Tuning Factors from the DPP.

Command Structure

The general command structure is shown below:

[Command] [Data][CR]

Command

The Command consists of two ASCII characters.

Data

The Data to be sent to the DPP. The Data requirements are different for each command, and some commands don't require any data at all.

CR

Carriage Return Character. In ASCII the carriage return is defined as 13 or 0D in hexadecimal.

The DPP returns data in the same format without the command on the front.

[Data][CR]

Command Descriptions

Acquisition Commands

The acquisition commands are used to start and stop the collection of data into the spectrum buffer along with clearing the spectrum buffer.

GO – Start Acquisition

Starts or continues the DPP data collection process. To start a new acquisition, send the Clear Acquisition command before the start command.

PC → DPP	DPP → PC
GO[CR]	[CR]

SP – Stop Acquisition

Stops the acquisition currently in progress.

PC → DPP	DPP → PC
SP[CR]	[CR]

CL – Clear Acquisition

Clears any data currently in the acquisition buffer and clears the run statistics. If an acquisition is currently in process this will restart the acquisition without first stopping it.

PC → DPP	DPP → PC
CL[CR]	[CR]

AP – Acquisition in Progress

Requests the state of the current acquisition.

PC → DPP	DPP → PC	Parameter n
AP[CR]	n[CR]	0 = No Acquisition in Progress 1 = Acquisition in Progress

Parameter Commands

The Parameter commands control how the DPP processes the data. Some parameters are dependent upon other parameters. For example the peaking time is dependent on the clock divider setting.

CD – Clock Divider

The DPP clock runs at 100MHz but each peaking time settings doesn't need to run at the full clock speed. Instead the DPP divides the clock and runs it at a slower frequency for the selected peaking time.

Clock Divider	Clock Speed (MHz)	Peaking Time Range (µs)
4	25	0.08 – 20.00
8	12.5	20.08 – 40.00
16	6.25	40.16 – 80.00
32	3.125	80.32 – 160.00
64	1.5625	160.64 – 327.04

PC → DPP	DPP → PC	Parameter n
CDn[CR] CD[CR]	n[CR] n[CR]	4 to 64 (See Clock Divider Table)

The Sampling Speed is used in some parameter calculations. Here is the formula to determine the Sampling Speed.

$$t_S = \frac{1}{Clock\ Speed}$$

PT – Preprocessor Time Constant Index

The Preprocessor Index is used to calculate the decay time of the preprocessor. For slow channel peaking times < 2.56 see the table below, above 2.56 use *1.5 x Slow Peaking Time* to determine the setting for the Preprocessor Index.

Slow PT (µs)	P.P. Index (T)	Preprocessor Time Constant (τ)
0.08 – 0.12	0	0.036
0 .16 – 0.80	1	0.320
0.84 – 1.64	2	0.604
1.68 – 2.52	3	0.888
> 2.56	T Formula	τ Formula

$$T = \mathrm{PT} \times 1.5$$

Where:

PT = Slow Peaking Time

PC → DPP	DPP → PC	Parameter n (T)
PTn[CR] PT[CR]	n[CR] n[CR]	0 to 255 (See PT Formula & Table)

The Preprocessor Time Constant is used in some parameter calculations. Here is the formula to determine the Preprocessor Time Constant.

$$\tau = T \times m + b$$

Where:

T = *Preprocessor Index*
m = *EQ slope (See Tuning Factors)*
b = *EQ offset (See Tuning Factors)*

PG – Preprocessor Gain

The Preprocessor Gain is the analog gain before the signal is digitized. If the gain is too low the signal to noise ratio at the A/D converter is poor and can cause poor resolution. A good starting position is to set the Preprocessor Gain so that the digital gain can be set to 0.200 - 0.300. This provides 10eV per channel or 0.400 - 0.600 for 5eV per channel. Fine gain adjustments should be done using the digital gain. The PG command sets the analog gain of a variable gain amplifier. An 8 bit binary code that consists of a 7 bit fine gain code PG_{06} (lower bits) and a 1 bit coarse gain flag PG_7 (highest bit). The Actual gain is calculated with the following equation:

$$G = PG_{06} \times 0.055754 \times (1 + PG_7 \times 6.079458)$$

Where:

PG_{06} = *7 bit fine gain code*
PG_7 = *1 bit coarse gain flag*

The formula to calculate the parameter to send to DPP is as follows:

$$n = PG_7 \times 128 + PG_{06}$$

PC → DPP	DPP → PC	Parameter n
PGn[CR] PG[CR]	n[CR] n[CR]	0 to 255 (See PG Formula)

EF – Equalization Factor

The Equalization Factor is used to factor out the error caused from tolerances of the Resistor and Capacitors in the preprocessor. Variances in the preprocessor electronic components can cause undershoots and overshoots in the slow and fast channels. These errors can cause peak broadening and/or peak shifting in the spectrum. The EQ Factor is calculated using the Equalization Factor formula below.

$$E = \frac{t_S}{\tau} \times 2^{17}$$

Where:

t_S = *Sampling Speed (See Clock Divider)*
τ = *Preprocessor Time Constant*

PC → DPP	DPP → PC	Parameter n
EFn[CR] EF[CR]	n[CR] n[CR]	0 to 65,535 (See EF Formula)

ZF – Zero Factor

The Zero Factor sets the zero input level of the A/D converter input. The optimum value gives the same position of the peaks in cases of baseline restorer on and off. Automatic adjustment is done if the Zero Factor is -1.

PC → DPP	DPP → PC	Parameter n
ZFn[CR] ZF[CR]	n[CR] n[CR]	0 to 16,383

PS – Peaking Time Slow Index

The Slow Peaking Time is the main filter of the DPP. There are 511 possible peaking times per clock frequency. The index is calculated using this formula:

$$n = PS \times CLK$$

Where:

PS = *Slow Peaking Time (example 24µs)*
CLK = *Speed of the clock (example 12.5 MHz)*

PC → DPP	DPP → PC	Parameter n
PSn[CR] PS[CR]	n[CR] n[CR]	1 to 511

PX – Peaking Time Fast2 Index

The Fast2 Peaking Time is used in the pile up rejecter. There are 31 possible peaking times per clock frequency. The index is calculated using this formula:

$$n = PX \times CLK$$

Where:

PX = *Fast2 Peaking Time (example 0.32µs)*
CLK = *Speed of the clock (example 6.25 MHz)*

PC → DPP	DPP → PC	Parameter n
PXn[CR] PX[CR]	n[CR] n[CR]	1 to 31

PY – Peaking Time Fast3 Index

The Fast2 Peaking Time is used in the pile up rejecter. There are 31 possible peaking times per clock frequency. The index is calculated using this formula:

$$n = PY \times CLK$$

Where:

PY = *Fast3 Peaking Time (example 0.64µs)*
CLK = *Speed of the clock (example 6.25 MHz)*

PC → DPP	DPP → PC	Parameter n
PYn[CR] PY[CR]	n[CR] n[CR]	1 to 31

HS – Holding Time Slow

The Slow Holding Time is used in the main filter function to determine how long the flat top is of the triangular shaped pulse. There are 32 possible peaking times per clock frequency. The index is calculated using this formula:

$$n = HS \times CLK$$

Where:

PY = *Slow Holding Time (example 0.16µs)*
CLK = *Speed of the clock (example 6.25 MHz)*

PC → DPP	DPP → PC	Parameter n
HSn[CR] HS[CR]	n[CR] n[CR]	0 to 31

HX – Holding Time Fast2

The Fast2 Holding Time is used in the pile up rejecter function to determine how long the flat top is of the triangular shaped pulse. For the Fast channels this is usually set to zero. There are 32 possible peaking times per clock frequency. The index is calculated using this formula:

$$n = HX \times CLK$$

Where:

HX = *Fast2 Holding Time (example 0)*
CLK = *Speed of the clock (example 6.25 MHz)*

PC → DPP	DPP → PC	Parameter n
HXn[CR] HX[CR]	n[CR] n[CR]	0 to 31

HY – Holding Time Fast3

The Fast3 Holding Time is used in the pile up rejecter function to determine how long the flat top is of the triangular shaped pulse. For the Fast channels this is usually set to zero. There are 32 possible peaking times per clock frequency. The index is calculated using this formula:

$$n = HY \times CLK$$

Where:

HY = *Fast2 Holding Time (example 0)*
CLK = *Speed of the clock (example 6.25 MHz)*

PC → DPP	DPP → PC	Parameter n
HYn[CR] HY[CR]	n[CR] n[CR]	0 to 31

GS – Gain Slow

The digital gain is the gain after the signal has been digitized. This can be used to adjust the eV/Channel in the spectrum. Typical values for the digital gain are between 0.200 - 0.700 depending on the desired eV/Channel. The digital gain value needs to be converted to the gain command parameter using the following equation:

$$n = \frac{GS \times 2^{22}}{PS}$$

Where:

GS = *Gain Slow*
PS = *Peaking Time Slow*

PC → DPP	**DPP → PC**	**Parameter n**
GSn[CR] GS[CR]	n[CR] n[CR]	1 to 16,777,215 (See GS Formula)

GX – Gain Fast2

The Fast2 gain is typically set to the same value as the slow gain. The digital gain value needs to be converted to the gain command parameter using the following equation:

$$n = \frac{GX \times 2^{22}}{PX}$$

Where:

GX = *Gain Slow*
PX = *Peaking Time Fast2*

PC → DPP	**DPP → PC**	**Parameter n**
GXn[CR] GX[CR]	n[CR] n[CR]	1 to 16,777,215 (See GX Formula)

GY – Gain Fast3

The Fast2 gain is typically set to the same value as the slow gain. The digital gain value needs to be converted to the gain command parameter using the following equation:

$$n = \frac{GY \times 2^{22}}{PY}$$

Where:

GY = *Gain Fast3*
PY = *Peaking Time Fast3*

PC → DPP	DPP → PC	Parameter n
GYn[CR] GY[CR]	n[CR] n[CR]	1 to 16,777,215 (See GY Formula)

TS – Threshold Slow

The Threshold Slow sets the lowest energy of analysis. Any data below this channel is discarded.

PC → DPP	DPP → PC	Parameter n
TSn[CR] TS[CR]	n[CR] n[CR]	1 to 4095

TX – Threshold Fast2

The Threshold Fast2 sets the lowest energy of the pile up Fast2 filter. Setting the Fast2 threshold to 4095 turns off the pile up rejecter.

PC → DPP	DPP → PC	Parameter n
TXn[CR] TX[CR]	n[CR] n[CR]	1 to 4095

TY – Threshold Fast3

The Threshold Fast3 sets the lowest energy of the pile up Fast3 filter. Setting the Fast3 threshold to 4095 turns off the pile up rejecter.

PC → DPP	DPP → PC	Parameter n
TYn[CR] TY[CR]	n[CR] n[CR]	1 to 4095

IL – Reset Inhibit Length

The Reset Inhibit tells the DPP how long after the detector reset pulse to wait before starting to process the data.

$$n = IL \times CLK$$

Where:

IL = *Reset Inhibit Length (example 20µs)*
CLK = *Speed of the clock (example 6.25 MHz)*

PC → DPP	DPP → PC	Parameter n
ILn[CR] IL[CR]	n[CR] n[CR]	0 to 4095

BM – Base Line Mode

The Base Line Mode switches the baseline restorer ON and OFF. For normal operation leave it on (1). If the normal mode of the baseline restorer does not work well because of the heavy non-linearity of the preamplifier output ramp or frequent reset events, the time constant of the baseline restorer can be shortened using a parameter from 11 to 14. A parameter setting of 15 gives the same constant as 1.

PC → DPP	DPP → PC	Parameter *n*
BMn[CR] BM[CR]	n[CR] n[CR]	1 to 15

BW – Base Line Window

The Base Line Window changes the width of the amplitude for the baseline integration. If the number is unnecessarily large, larger peak shifts at high count rates occur.

PC → DPP	DPP → PC	Parameter n
BWn[CR] BW[CR]	n[CR] n[CR]	1 to 4095

DL – Dead Time Length

The Dead Time Length changes the length of the dead time for each event. If everything including detector, preamplifier and the signal processing is ideal, it can be one. However, in many cases after processing high energy events, undershoot or overshoot of the filter output produces ghost signals in the low energy region. So it is recommended to set it to 1 (DL = 1.5 x PT).

PC → DPP	DPP → PC	Parameter *n*
DLn[CR] DL[CR]	n[CR] n[CR]	0 = 1.0 x PT 1 = 1.5 x PT 2 = 2.5 x PT 3 = 3.0 x PT 4 = 3.5 x PT

CO – Channel Offset

The Channel Offset command can be used to adjust the small offset in the relationship between channel and energy. Each increment represents 1/16 of a channel. Zero equates to no adjustment.

PC → DPP	DPP → PC	Parameter n
COn[CR] CO[CR]	n[CR] n[CR]	-8191 to 8191

Preset Commands

The Presets are conditions that must be met before the DPP automatically stops an acquisition. There are four Preset conditions available in the MXDPP-50.

TM – Preset Timer Mode

The Preset Timer Mode sets the timer (Real or Live) that the PR Preset Timer uses.

PC → DPP	DPP → PC	Parameter *n*
TMn[CR] TM[CR]	n[CR] n[CR]	0 – Live Time 1 – Real Time

PR – Preset Timer

The Preset Timer stops the acquisition when either the Real Time or Live Time reaches the PR value. Use TM command to set the Timer mode. The Preset Time is set in milliseconds.

PC → DPP	DPP → PC	Parameter *n*
PRn[CR] PR[CR]	n[CR] n[CR]	1 to 4,294,967,295 (milliseconds)

TC – Preset Total Counts

The Preset Total Count command stops the acquisition when the total counts in the spectrum buffer reaches the TC value.

PC → DPP	DPP → PC	Parameter n
TCn[CR] TC[CR]	n[CR] n[CR]	1 to 4,294,967,295 (counts)

PC – Preset Peak Counts

The Preset Peak Counts command stops the acquisition when the highest peak counts in the spectrum buffer reaches the PC value.

PC → DPP	DPP → PC	Parameter n
PCn[CR] PC[CR]	n[CR] n[CR]	1 to 4,294,967,295 (counts)

Statistic Commands

The Statistic Commands retrieve the current acquisition statistic data from the DPP.

RT – Real Time

The Real Time command returns the current value for the Real Time Timer.

PC → DPP	DPP → PC	Parameter *n*
RT[CR]	*n*[CR]	0 to 4,294,967,295 (milliseconds)

LT – Live Time

The Live Time command returns the current value for the Live Time Timer.

PC → DPP	DPP → PC	Parameter n
LT[CR]	n[CR]	0 to 4,294,967,295 (milliseconds)

DR – Dead Time

The Dead Time command returns the current Dead Time in 0.001% units. The rate at which the Dead Time is calculated depends on the Rate Interval (RI) command.

PC → DPP	DPP → PC	Parameter n
DR[CR]	n[CR]	0 to 100,000 (0.001% units)

RS – Count Rate Slow

The Count Rate Slow command returns the current count rate in the Slow channel. The rate at which the count rate is calculated depends on the Rate Interval (RI) command.

PC → DPP	DPP → PC	Parameter n
RS[CR]	n[CR]	Slow Channel (CPS)

RX – Count Rate Fast2

The Count Rate Fast2 command returns the current count rate in the Fast2 channel. The rate at which the count rate is calculated depends on the Rate Interval (RI) command.

PC → DPP	DPP → PC	Parameter n
RX[CR]	n[CR]	Fast2 Channel (CPS)

RY – Count Rate Fast3

The Count Rate Fast3 command returns the current count rate in the Fast3 channel. The rate at which the count rate is calculated depends on the Rate Interval (RI) command.

PC → DPP	DPP → PC	Parameter *n*
RX[CR]	*n*[CR]	Fast3 Channel (CPS)

IR – Input Count Rate

The Input Count Rate command returns the current Input Count Rate (ICR). The rate at which the count rate is calculated depends on the Rate Interval (RI) command.

PC → DPP	DPP → PC	Parameter *n*
IR[CR]	*n*[CR]	Input Count Rate (CPS)

OR – Output Count Rate

The Output Count Rate command returns the current Output Count Rate (OCR). The rate at which the count rate is calculated depends on the Rate Interval (RI) command.

PC → DPP	DPP → PC	Parameter *n*
OR[CR]	*n*[CR]	Output Count Rate (CPS)

RM – Corrected Input Count Rate

The Corrected Input Count Rate command returns the Input Count Rate corrected for dead time. The rate at which the count rate is calculated depends on the Rate Interval (RI) command.

PC → DPP	DPP → PC	Parameter *n*
RM[CR]	*n*[CR]	Corrected Input Count Rate (CPS)

RI – Rate Interval

The Rate Interval command controls how often the DPP statistics are calculated. It also sets how often the Single Channel Analyzer rate output is updated.

PC → DPP	DPP → PC	Parameter *n*
IR*n*[CR] IR[CR]	*n*[CR] *n*[CR]	1 to 4,294,967,295 (milliseconds)

Detector Commands

High Voltage Polarity

The High Voltage Polarity command selects whether the high voltage supply outputs a Positive or Negative voltage. There is a relay on the output of the high voltage power supply that switches the output from positive to negative. Bit 0 of the Digital Output port controls the high voltage polarity.

PC → DPP	DPP → PC	Parameter *n*
PO*n*[CR] PO[CR]	*n*[CR] *n*[CR]	Bit 0 (nnnn nnnn nnnn nnnX) 0 = Positive, 1 = Negative

High Voltage Set Point

The High Voltage Set Point is controlled by a digital to analog converter which outputs a voltage to the high voltage supply. The equation for converting the desired high voltage to bits is as follows.

$$n = \frac{V}{50} \times 819$$

Where:

n = *Value to send to DPP*
V = *Desired high voltage bias setting*

PC → DPP	DPP → PC	Parameter *n*
&A*n*[CR] &A[CR]	*n*[CR] *n*[CR]	0 to 4095 See High Voltage Set Equation

High Voltage Monitor

The High Voltage is monitored through a voltage divider and is then fed into an ADC on the DPP. Use the following equation to convert the value read from the DPP to the actual high voltage value.

$$V = 2.5 \times \frac{n}{4095}$$

Where:

V = *High Voltage Value*
n = *Value read from DPP*

PC → DPP	**DPP → PC**	**Parameter *n***
#D[CR]	*n*[CR]	0 to 4095 See High Voltage Monitor Equation

Temperature Controller Mode

The Temperature Controller Mode selects whether the Temperature Controller is set to Detector or Box mode. There are a few relays on the DPP that switch the Temperature Controller Mode. Bit 2 of the Digital Output port controls the Temperature Controller Mode.

PC → DPP	**DPP → PC**	**Parameter *n***
PO*n*[CR] PO[CR]	*n*[CR] *n*[CR]	Bit 0 (nnnn nnnn nnnn nnnX) 0 = Detector, 1 = Box

Temperature Controller Set Point

The temperature set point tells the temperature controller what temperature to drive the detector to. This is only available when the temperature controller is set to BOX mode and the detector does not have a built in temperature controller.

The Temperature Controller Set Point is controlled by a digital to analog converter (DAC) which outputs a voltage which represents the desired temperature to the Temperature Controller. The equations for converting the desired set point temperature to bits are as follows.

First convert the temperature to the thermistor resistance using the inverse Steinhart-Hart Thermistor equation:

$$R_T = \exp\left(\sqrt[3]{\sqrt{\left(\frac{B}{3C}\right)^3 + \left(\frac{A - \frac{1}{T}}{2C}\right)^2} - \left(\frac{A - \frac{1}{T}}{2C}\right)} - \sqrt[3]{\sqrt{\left(\frac{B}{3C}\right)^3 + \left(\frac{A - \frac{1}{T}}{2C}\right)^2} + \left(\frac{A - \frac{1}{T}}{2C}\right)}\right)$$

Where:

R_T = *Thermistor Resistance*
T = *Desired Temperature (°C)*
A = *0.0018590668*
B = *0.0002367000*
C = *0.0000007811*

Next convert the thermistor resistance to the set point voltage using the voltage divider equation:

$$V_T = \frac{V_{REF} \times R_T}{R_T + R_P}$$

Where:

V_T = *Temperature Control Voltage*
R_T = *Thermistor Resistance Ω*
R_P = *Pull-Up Resistor (3320 Ω)*
V_{REF} = *Voltage Reference (5 V)*

Finally convert the Temperature Voltage to bits:

$$n = V_T \times 819$$

Where:

n = *Value to send to DPP*
V_T = *Temperature Control Voltage*

PC → DPP	**DPP → PC**	**Parameter *n***
&B*n*[CR]	*n*[CR]	0 to 4095
&B[CR]	*n*[CR]	See Temp CTRL Set Equations

Detector Temperature Monitor

The detector has an internal thermistor that is used to monitor the internal temperature. The thermistor is connected to a 3.32KΩ pull-up resistor connected to a +5V supply on the preamplifier which creates a voltage divider between the pull-up resistor and the thermistor. The output of the voltage divider is fed into an ADC on the DPP. Use the following equations to convert the value read from the DPP to the actual detector temperature.

First convert the value read from the DPP to volts:

$$V_M = 5 \times \frac{n}{4095}$$

Where:

V_M = Temperature Monitor Voltage
n = Value read from DPP

Next convert the monitor voltage to the thermistor resistance using a voltage divider equation:

$$R_T = \frac{V_T \times R_P}{V_{REF} - V_T}$$

Where:

R_T = Thermistor Resistance Ω
V_M = Temperature Monitor Voltage
R_P = Pull-Up Resistor (3320 Ω)
V_{REF} = Voltage Reference (5 V)

Finally we convert the thermistor resistance to temperature using the Steinhart-Hart Thermistor equation:

$$T = \frac{1}{A + B \times LN(R_T) + C \times LN(R_T)^3} - 273.15$$

Where:

T = Temperature °C
R_T = Thermistor Resistance Ω
A = 0.0018590668
B = 0.0002367000
C = 0.0000007811

PC → DPP	DPP → PC	Parameter *n*
#B[CR]	*n*[CR]	0 to 4095 See Detector Temp Mon Equations

Detector Ready Signal

The detector temperature controller outputs a TTL signal to the DPP to indicate that the detector is close to the temperature set point. Use the Digital Input command to read the status of the detector ready signal. The ready signal is bit 0 of the digital input port.

PC → DPP	DPP → PC	Parameter *n*
PI[CR]	*n*[CR]	Bit 0 (nnnn nnnX) 0 = Not Ready, 1 = Ready

Detector TEC Voltage

The detector uses a Thermoelectric Cooler (TEC) to keep the detector sensor cold. The voltage fed to the TEC is also fed to an analog to digital converter (ADC) so it can be monitored. Use the following equation to convert the value read from the DPP to the TEC voltage.

$$V_{TEC} = 5 \times \frac{n}{4095}$$

Where:

V_{TEC} = *Temperature Monitor Voltage*
n = *Value read from DPP*

PC → DPP	**DPP → PC**	**Parameter *n***
#C[CR]	n[CR]	0 to 4095 See TEC Monitor Equation

DPP Temperature Monitor

The DPP Temperature is monitored using a National Semiconductor LM62 temperature sensor. The output of the LM62 is fed into an ADC on the DPP. Use the following equations to convert the value read to temperature.

First, convert the value read from the DPP to voltage:

$$V = 2.5 \times \frac{n}{4095}$$

Where:

V = *Voltage from LM62*
n = *Value read from DPP*

Next, convert the Voltage to temperature using the LM62 equation:

$$T = \frac{V - 0.48}{0.0156}$$

Where:

T = *DPP Temperature (°C)*
V = *Voltage from LM62*

PC → DPP	DPP → PC	Parameter *n*
#A[CR]	*n*[CR]	0 to 4095 See DPP Temperature Equations

Ramp Polarity

The Ramp Polarity selects the detector input ramp polarity, Positive or Negative. There is a relay on the input of the DPP that switches the input from positive to negative. Bit 1 of the Digital Output port controls the Ramp Polarity.

PC → DPP	DPP → PC	Parameter *n*
PO*n*[CR] PO[CR]	*n*[CR] *n*[CR]	Bit 1 (nnnn nnnn nnnn nXnn) 0 = Negative, 1 = Positive

Auxiliary I/O Commands

PO – Digital Outputs

The Digital Output port controls various DPP and power supply functions. The output buffer is a 16 bit word with each bit controlling a different function in the DPP. See the following table for a description of each bit. For the SCA channels if the DPP Mode switch is in position 0 the outputs can be controlled using the PO command, for the SCA function the Mode switch should be in position 3.

Bit #	Port Description	Output State
0	High Voltage Polarity	0 = Positive, 1 = Negative
1	Detector Ramp Polarity	0 = Negative, 1 = Positive
2	Temp CTRL Mode	0 = Detector, 1 = Box
3	Not Used	N/A
4	Not Used	N/A
5	Not Used	N/A
6	Auxiliary Output 2	0 = Low, 1 = High

Bit #	Port Description	Output State
7	Auxiliary Output 1	0 = Low, 1 = High
8	SCA 8	0 = Low, 1 = High (Mode switch 0)
9	SCA 7	0 = Low, 1 = High (Mode switch 0)
10	SCA 6	0 = Low, 1 = High (Mode switch 0)
11	SCA 5	0 = Low, 1 = High (Mode switch 0)
12	SCA 4	0 = Low, 1 = High (Mode switch 0)
13	SCA 3	0 = Low, 1 = High (Mode switch 0)
14	SCA 2	0 = Low, 1 = High (Mode switch 0)
15	SCA 1	0 = Low, 1 = High (Mode switch 0)

PC → DPP	DPP → PC	Parameter *n*
PO*n*[CR] PO[CR]	*n*[CR] *n*[CR]	1 to 65,535 (See bit table)

PI – Digital Inputs

The Digital Input port can be used to monitor the Detector Ready Signal along with the Auxiliary inputs. The input buffer is an 8 bit word with each bit monitoring a different function in the DPP. See the following table for a description of each bit. Bits 1-5 are not used.

Bit #	Port Description	Output State
0	Detector Ready	0 = Not Ready, 1 = Ready
1	Not Used	N/A
2	Not Used	N/A
3	Not Used	N/A
4	Not Used	N/A
5	Not Used	N/A
6	Auxiliary Input 2	0 = Low, 1 = High
7	Auxiliary Input 1	0 = Low, 1 = High

PC → DPP	DPP → PC	Parameter *n*
PI[CR]	*n*[CR]	1 to 4095 (See bit table)

Single Channel Analyzer (SCA) Commands

There are eight Single Channel Analyzer (SCA) channels on the MXDPP-50. Each SCA channel has its own window with a low channel and a high channel defining the location and width of the window. When an X-Ray event happens that is inside the defined window it triggers an output (depending on SCA Mode selection) on the associated Auxiliary port. The DPP Mode Switch needs to be in position 3 for the SCA to work.

RD – Single Channel Analyzer Mode

The Single Channel Analyzer mode sets the output mode of the SCA. The two modes are Rate and Pulse. In Rate mode the spacing between the output pulses are based on the count rate within the SCA window. For example if the count rate was 2000cps then the output frequency on that SCA channel would be 2KHz. In Pulse mode whenever there is an X-Ray event processed in the SCA channel there is a corresponding pulse output on the channel.

PC → DPP	DPP → PC	Parameter n
RDn[CR] RD[CR]	n[CR] n[CR]	0 = Pulse Mode 1 = Rate Mode

RI – Rate Interval

The Rate Interval controls how often the DPP statistics are calculated. It also sets how often the Single Channel Analyzer rate output is updated.

PC → DPP	DPP → PC	Parameter n
IRn[CR] IR[CR]	n[CR] n[CR]	1 to 4,294,967,295 (milliseconds)

_A – SCA #1 Low Channel

The Low channel in the SCA window for channel #1.

PC → DPP	DPP → PC	Parameter n
_An[CR] _A[CR]	n[CR] n[CR]	1 to 4095 (channel)

_B – SCA #2 Low Channel

The Low channel in the SCA window for channel #2.

PC → DPP	DPP → PC	Parameter n
_Bn[CR] _B[CR]	n[CR] n[CR]	1 to 4095 (channel)

_C – SCA #3 Low Channel

The Low channel in the SCA window for channel #3.

PC → DPP	DPP → PC	Parameter n
_Cn[CR] _C[CR]	n[CR] n[CR]	1 to 4095 (channel)

_D – SCA #4 Low Channel

The Low channel in the SCA window for channel #4.

PC → DPP	DPP → PC	Parameter n
_Dn[CR] _D[CR]	n[CR] n[CR]	1 to 4095 (channel)

_E – SCA #5 Low Channel

The Low channel in the SCA window for channel #5.

PC → DPP	DPP → PC	Parameter n
_En[CR] _E[CR]	n[CR] n[CR]	1 to 4095 (channel)

_F – SCA #6 Low Channel

The Low channel in the SCA window for channel #6.

PC → DPP	DPP → PC	Parameter n
_Fn[CR] _F[CR]	n[CR] n[CR]	1 to 4095 (channel)

_G – SCA #7 Low Channel

The Low channel in the SCA window for channel #7.

PC → DPP	DPP → PC	Parameter n
_Gn[CR] _G[CR]	n[CR] n[CR]	1 to 4095 (channel)

_H – SCA #8 Low Channel

The Low channel in the SCA window for channel #8.

PC → DPP	DPP → PC	Parameter n
_Hn[CR] _H[CR]	n[CR] n[CR]	1 to 4095 (channel)

^A – SCA #1 High Channel

The High channel in the SCA window for channel #1.

PC → DPP	DPP → PC	Parameter n
^An[CR] ^A[CR]	n[CR] n[CR]	1 to 4095 (channel)

^B – SCA #2 High Channel

The High channel in the SCA window for channel #2.

PC → DPP	DPP → PC	Parameter n
^Bn[CR] ^B[CR]	n[CR] n[CR]	1 to 4095 (channel)

^C – SCA #3 High Channel

The High channel in the SCA window for channel #3.

PC → DPP	DPP → PC	Parameter n
^Cn[CR] ^C[CR]	n[CR] n[CR]	1 to 4095 (channel)

^D – SCA #4 High Channel

The High channel in the SCA window for channel #4.

PC → DPP	DPP → PC	Parameter n
^Dn[CR] ^D[CR]	n[CR] n[CR]	1 to 4095 (channel)

^E – SCA #5 High Channel

The High channel in the SCA window for channel #5.

PC → DPP	DPP → PC	Parameter n
^En[CR] ^E[CR]	n[CR] n[CR]	1 to 4095 (channel)

^F – SCA #6 High Channel

The High channel in the SCA window for channel #6.

PC → DPP	DPP → PC	Parameter n
^Fn[CR] ^F[CR]	n[CR] n[CR]	1 to 4095 (channel)

^G – SCA #7 High Channel

The High channel in the SCA window for channel #7.

PC → DPP	DPP → PC	Parameter n
^Gn[CR] ^G[CR]	n[CR] n[CR]	1 to 4095 (channel)

^H – SCA #8 High Channel

The High channel in the SCA window for channel #8.

PC → DPP	DPP → PC	Parameter n
^Hn[CR] ^H[CR]	n[CR] n[CR]	1 to 4095 (channel)

Memory Commands

The DPP contains a 1KBIT bank of EEPROM memory for storing DPP commands which are loaded during the power on/boot cycle. During the boot cycle the DPP reads commands in the EEPROM memory and executes them as if they were coming from the USB. Any of the commands used to control the DPP can be written to the memory and the DPP will execute them during the boot process. The commands must be in the same format as the commands that come from the USB.

EW – Write User Memory

Sending the "EW" command will redirect all following commands to the EEPROM until the DPP receives the end character " ` " (hexadecimal 60). Once the end character is received the DPP returns a count of all the characters written after the "EW" command including the end characters. The maximum number of characters that can be written to the EEPROM is 1024.

PC → DPP	DPP → PC	Send Parameter n	Return Parameter x
EWn[CR]	x[CR]	String of commands to store in EEPROM (Include [CR] between each command)	Number of characters written to EEPROM

ER – Read User Memory

Returns the contents of the EEPROM memory. The string of characters will end with the end character " ` " (hexadecimal 60).

PC → DPP	DPP → PC	Parameter n
ER[CR]	n[CR]	String of commands stored in EEPROM (Includes [CR] between each command)

D2XX Commands

The D2XX communication interface uses the FTD2XX.DLL. The FTD2XX.DLL can be found in the D2XX VB.NET example program or it can be downloaded on the FTDI's website. This manual does not provide extensive information about FTDI's FTD2XX.DLL. It only provides brief descriptions for the commands used in the example code. For more information on the D2XX communication interface download the *FTDI's D2XX Programmer's Guide* from the www.ftdi.com website.

FT_GetNumberOfDevices

Requests the number of FTDI devices attached to the computer.

FT_GetDeviceString

Requests a description of the device at the specified index. This command can be used to retrieve the DPP Tuning Factors. Once the device description has been retrieved the string will need to be parsed to extract the DPP Tuning Factors. See the D2XX VB.NET program for example of how this can be done.

FT_GetDeviceInfo

Requests the info about the device at the specified index. This command can be used to retrieve the DPP Tuning Factors. Once the device description has been retrieved the string will need to be parsed to extract the DPP Tuning Factors. See the D2XX VB.NET program for example of how this can be done.

FT_ResetDevice

Resets the device at the specified handler.

FT_OpenBySerialNumber

Opens a connection to the device by the specified serial number and retrieves the device handler.

FT_Close

Closes the connection to the device at the specified handler.

FT_SetBaudRate

Sets the communication speed, also known as (aka) Baud Rate.

FT_SetDataCharacteristics

Sets the data characteristics such as number of data bits, stop bits and the parity.

FT_SetFlowControl

Sets the communication flow control.

FT_SetRts

Asserts the request to send (RTS) line.

FT_SetDtr

Asserts the data terminal ready (DTR) line.

FT_Purge

Clears the receive and transmit buffers.

FT_Write_String

Writes a string of data to the device.

FT_Write_Bytes

Writes an array of bytes to the device.

FT_Read_String

Reads a string of data from the device.

FT_Read_Bytes

Reads an array of bytes from the device.

FT_GetQueueStatus

Retrieves the number of bytes or characters waiting to be read from the device.

Revision History

Re	DCN #	Author	Date	Description of Change
A	2012-2507	C Carter, T Zeal	09/06/2013	Initial release