

Model P580
Relay Multiplexer
User's Manual

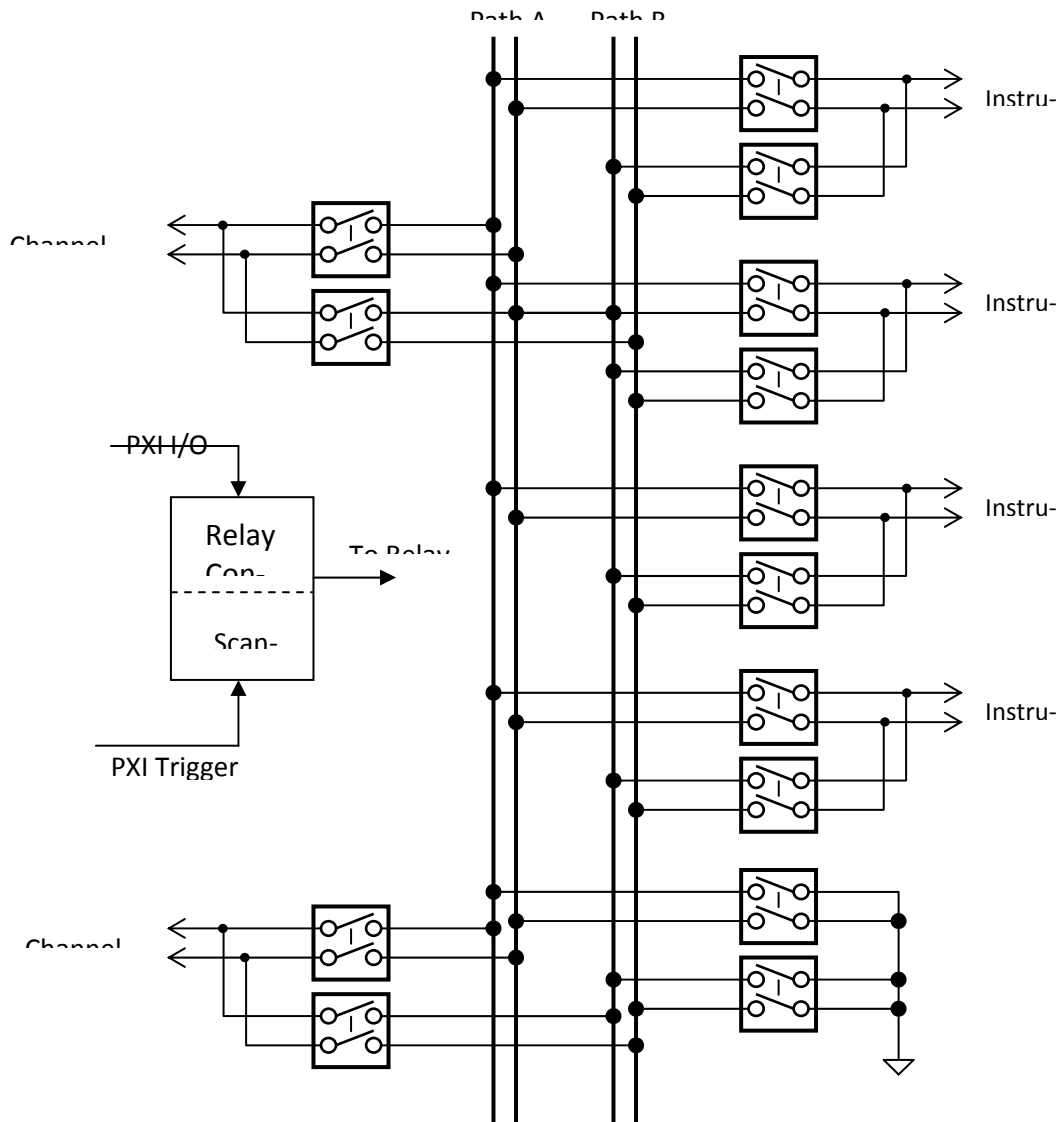
Copyright by
KineticSystems Company, LLC
Lockport, Illinois
All rights reserved

Chapter 1: Understanding the P580

Overview

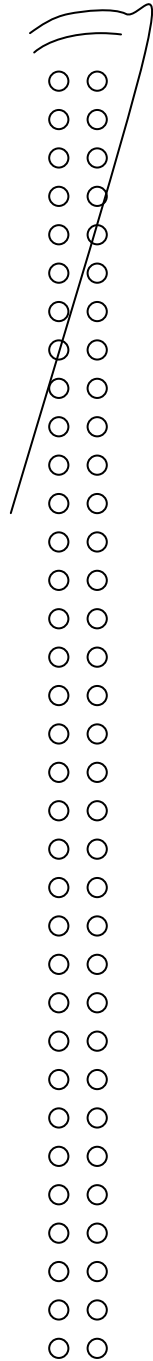
The P580 is a double width 3U PXI module that functions as a multiplexer. The P580 incorporates two independent differential analog paths that can be connected to any of 34 differential signal paths through the 68 position front panel mounted SCSI connector. Also, each path can be routed to any of the four SMB front panel mounted connectors.

The following diagram shows the internal architecture of the P580.



Basic Circuit Operation

Front-panel Connector Pinout



Chapter 2: Configuration and Operational Registers

Required Configuration Registers Operational Registers

The following operational registers are used to configure the P580. The following table shows the registers and their offsets relative to the assigned base address in memory.

Offset	Register	Access
00h	Instrument Connection Register	Write/Read
04h	Trigger Connection Register	Write/Read
08h	Path A Connection Register Low	Write/Read
0Ch	Path A Connection Register High	Write/Read
10h	Path B Connection Register Low	Write/Read
14h	Path B Connection Register High	Write/Read
18h	Scanner Configuration Register	Write/Read
1Ch	Step Scanner	Write-Only
1Ch-FFCh	Reserved	-
1000h	Scanner Memory Location 1	Write/Read
1004h	Scanner Memory Location 2	Write/Read
1008h	Scanner Memory Location 3	Write/Read
100Ch	Scanner Memory Location 4	Write/Read
1FF8h	Scanner Memory Location 1023	Write/Read
1FFCh	Scanner Memory Location 1024	Write/Read

Instrument Connection Register

00h

This read/write register selects path to which the external instruments are connected through the front panel mounted SMB connectors. This register is also used to control the grounding of each analog path.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read/write	0	0	0	0	0	0	0	0	0	0	0	0	0	0	GND PTHB	GND PTHA
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	INST SL4B	INST SL4A	0	0	INST SL3B	INST SL3A	0	0	INST SL2B	INST SL2A	0	0	INST SL1B	INST SL1A

GND PTHB

Bit 17

The Ground Path B bit is used to ground the differential pair on analog Path B. Both the positive and negative side of the path are grounded. Setting this bit to a “1” grounds the path and a “0” removes the path from ground.

GND PTHA

Bit 16

The Ground Path A bit is used to ground the differential pair on analog Path A. Both the positive and negative side of the path are grounded. Setting this bit to a “1” grounds the path and a “0” removes the path from ground.

INST SL4B and INST SL4A

Bits 13 and 12

The Instrument Select 4B and 4A bits are used to control the connection of the front panel mounted INST 4 SMB. This SMB connector can remained unconnected to any path, connected to Path A, or connected to Path B. The binary combination of these bits determine the connection path as shown in the following table.

INST SL4B	INST SL4A	Connection
0	0	Not Connected
0	1	Connected to Path A
1	0	Connected to Path B
1	1	Connected to Path A and Path B

INST SL3B and INST SL3A

Bits 9 and 8

The Instrument Select 3B and 3A bits are used to control the connection of the front panel mounted INST 3 SMB. This SMB connector can remained unconnected to any path, connected to Path A, or connected to Path B. The binary combination of these bits determine the connection path as shown in the following table.

INST SL3B	INST SL3A	Connection
0	0	Not Connected
0	1	Connected to Path A

1	0	Connected to Path B
1	1	Connected to Path A and Path B

INST SL2B and INST SL2A **Bits 5 and 4**

The Instrument Select 2B and 2A bits are used to control the connection of the front panel mounted INST 2 SMB. This SMB connector can remain unconnected to any path, connected to Path A, or connected to Path B. The binary combination of these bits determine the connection path as shown in the following table.

INST SL2B	INST SL2A	Connection
0	0	Not Connected
0	1	Connected to Path A
1	0	Connected to Path B
1	1	Connected to Path A and Path B

INST SL1B and INST SL1A **Bits 1 and 0**

The Instrument Select 1B and 1A bits are used to control the connection of the front panel mounted INST 1 SMB. This SMB connector can remain unconnected to any path, connected to Path A, or connected to Path B. The binary combination of these bits determine the connection path as shown in the following table.

INST SL1B	INST SL1A	Connection
0	0	Not Connected
0	1	Connected to Path A
1	0	Connected to Path B
1	1	Connected to Path A and Path B

Trigger Connection Register **04h**

This read/write register is used to configure the connection of the PXI Trigger signals to either Path A or B. This feature is included in this module so that testing of other KSC product that uses the PXI trigger lines can be accommodated. When configuring the trigger lines, ensure that the direction of the trigger signal is observed. Input and Output references to the trigger lines are referred to the P580. Inputs are received by the P580 and outputs are generated by the P580.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read/write	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	TRGB DIR	TRG SL4B	TRG SL3B	TRG SL2B	TRG SL1B	0	0	0	TRGA DIR	TRG SL4A	TRG SL3A	TRG SL2A	TRG SL1A

TRGB DIR**Bit 12**

The Trigger B Direction bit is used to specify the direction of the PXI trigger line connected to Path B. Setting this bit to a “1” specifies that the trigger line is an output from the P580 and a “0” specifies the trigger line as an input.

TRG SL4B through TRG SL1B**Bits 11-8**

The TRG SL4B through TRG SL1B bits are used to select the trigger line which connects up to Path B of the P580. The binary combination of these bits determine the connection path as shown in the following table.

TRG SL4B – TRG SL1B Value	Trigger Connected to Path B
0	Disabled
1	PXI Trigger Line 0
2	PXI Trigger Line 1
3	PXI Trigger Line 2
4	PXI Trigger Line 3
5	PXI Trigger Line 4
6	PXI Trigger Line 5
7	PXI Trigger Line 6
8	PXI Trigger Line 7
9	PXI Star Trigger

TRGA DIR**Bit 4**

The Trigger A Direction bit is used to specify the direction of the PXI trigger line connected to Path A. Setting this bit to a “1” specifies that the trigger line is an output from the P580 and a “0” specifies the trigger line as an input.

TRG SL4A through TRG SL1A**Bits 3-0**

The TRG SL4A through TRG SL1A bits are used to select the trigger line which connects up to Path A of the P580. The binary combination of these bits determine the connection path as shown in the following table.

TRG SL4A – TRG SL1A Value	Trigger Connected to Path A
0	Disabled
1	PXI Trigger Line 0
2	PXI Trigger Line 1
3	PXI Trigger Line 2
4	PXI Trigger Line 3
5	PXI Trigger Line 4
6	PXI Trigger Line 5
7	PXI Trigger Line 6
8	PXI Trigger Line 7
9	PXI Star Trigger

Path A Connection Register Low

08h

This read/write register configures which of the differential front-panel positions that are to be connected to analog Path A. Each bit in this register refers to a differential pair of analog signals.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read/write	CH32	CH31	CH30	CH29	CH28	CH27	CH26	CH25	CH24	CH23	CH22	CH21	CH20	CH19	CH18	CH17
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH16	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1

CH32 through CH1

Bits 31-0

These bits are used to select which channels of the front panel 68-position SCSI connector are connected to Path A. This register contains the control for channels 1 through 32. The other 2 channels are configured in the *Path A Connection Register High*. Each bit that is set to a “1” connects the corresponding channel differential pair to Path A, and each bit that is set to a “0” disconnects the differential pair from Path A.

Path A Connection Register High

0Ch

This read/write register configures which of the differential front-panel positions that are to be connected to analog Path A. Each bit in this register refers to a differential pair of analog signals.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read/write	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CH34	CH33

CH34 through CH33

Bits 1-0

These bits are used to select which channels of the front panel 68-position SCSI connector are connected to Path A. Each bit that is set to a “1” connects the corresponding channel differential pair to Path A, and each bit that is set to a “0” disconnects the differential pair from Path A.

Path B Connection Register Low

10h

This read/write register configures which of the differential front-panel positions that are to be connected to analog Path B. Each bit in this register refers to a differential pair of analog signals.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
--	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Read/write	CH32	CH31	CH30	CH29	CH28	CH27	CH26	CH25	CH24	CH23	CH22	CH21	CH20	CH19	CH18	CH17
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH16	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1

CH32 through CH1

Bits 31-0

These bits are used to select which channels of the front panel 68-position SCSI connector are connected to Path B. This register contains the control for channels 1 through 32. The other 2 channels are configured in the *Path B Connection Register High*. Each bit that is set to a “1” connects the corresponding channel differential pair to Path B, and each bit that is set to a “0” disconnects the differential pair from Path A.

Path B Connection Register High

14h

This read/write register configures which of the differential front-panel positions that are to be connected to analog Path B. Each bit in this register refers to a differential pair of analog signals.

Read/write	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CH34	CH33

CH34 through CH33

Bits 1-0

These bits are used to select which channels of the front panel 68-position SCSI connector are connected to Path B. Each bit that is set to a “1” connects the corresponding channel differential pair to Path B, and each bit that is set to a “0” disconnects the differential pair from Path B.

Scanner Configuration Register

18h

This read/write register controls the multiplexer scanning feature of the P580. The P580 contains a 1024 word RAM that is used to hold a list of switch connections to be made as each scanner step is taken. Each element of the RAM indicates what instrument is connected to each analog path as well as which front panel channel is connected to a path. The sequencing of the scanner is controlled by a PXI trigger line or an external signal. Once the module receives a trigger to step to the next switch configuration, it can acknowledge the request by asserting another trigger line.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Scan Ena

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	Trig Out Pol	Scanner Trigger Out					0	0	Trig In Pol	Scanner Trigger In				

Scanner Enable**Bit 16**

The Scanner Enable bit is used to enable and disable the ability of the P580 to automatically scan through a predetermined list of switch connections. Setting this bit to a “1” enables the scanner and a “0” disables the scanner. When the scanner is disabled, all internal addresses for the Scanner Memory are reset to zero.

Trigger Out Polarity**Bit 13**

The Trigger Output Polarity bit is used to specify the polarity of the trigger output. Setting this bit to a “1” sets the trigger polarity to high true and a “0” selects low true polarity.

Scanner Trigger Out**Bits 12-8**

These 5 bits are used to specify the trigger output signal path. A pulse can be generated on either PXI trigger lines or an external connection to signify that the scanner step is complete. This signal can be used by other devices as an acknowledgement that the multiplexer is ready. Therefore, the P580 must take into account the settle time of the longest settling delay before generating this pulse. The pulse has a duration of 200 nanoseconds.

The following table shows the Scanner Trigger Out value that selects the destination of the scanner increment acknowledgement pulse

Scanner Trigger Out Value	Scanner Trigger Out Destination
0	Disabled
1	PXI Trigger Line 0
2	PXI Trigger Line 1
3	PXI Trigger Line 2
4	PXI Trigger Line 3
5	PXI Trigger Line 4
6	PXI Trigger Line 5
7	PXI Trigger Line 6
8	PXI Trigger Line 7
9	PXI Star Trigger
10	External Scanner Trigger Out

Trigger In Polarity**Bit 5**

The Trigger Input Polarity bit is used to specify the polarity of the trigger input. Setting this bit to a “1” sets the trigger polarity to high true and a “0” selects low true polarity.

Scanner Trigger In**Bits 4-0**

These 5 bits are used to specify the source of the signal which causes the multiplexer scanner to increment to the next switch selection. The source can be a PXI Trigger line, the PXI Star Trigger, or an exter-

nal source. The following table shows the Scanner Trigger In value that selects the source of the scanner increment pulse.

Scanner Trigger In Value	Scanner Trigger In Source
0	Disabled
1	PXI Trigger Line 0
2	PXI Trigger Line 1
3	PXI Trigger Line 2
4	PXI Trigger Line 3
5	PXI Trigger Line 4
6	PXI Trigger Line 5
7	PXI Trigger Line 6
8	PXI Trigger Line 7
9	PXI Star Trigger
10	External Scanner Trigger In

Step Scanner

1Ch

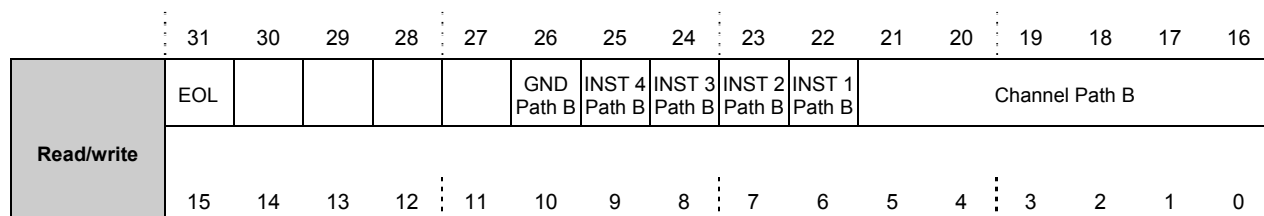
This write-only address is used to step the scanner memory to the next location. Once the scanner is enabled, the step scanner write operation can be used to move to the next configuration.

Scanner Memory

1000h – 1FFCh

The Scanner Memory is used to define the switching sequence used once the scanner is enabled. Once the scanner is enabled, the first Scanner Memory location is read by the P580 and the appropriate switch configuration is set. Once set, a trigger acknowledgement is sent through the Scanner Trigger Output configured in the Scanner Configuration Register. The switch selection remains until the scanner is signalled to increment to the next selection. The P580 can receive the increment signal from a PXI trigger line or an external trigger. Once the increment is received, the next switch selection is made and an acknowledge sent. This continues until the End Of List bit is encountered in the Scanner Memory. The Scanner Memory holds the various switch configuration steps in a RAM. The contents of the memory inform the P580 as to which switch closures to make for the given step. The memory contains information regarding which front panel channel to connect to Path A or Path B ; and which front panel instrument to connect to Path A and Path B. Also, an End Of List (EOL) bit is used to indicate when the last element of the scanner memory has been accessed. Once the last memory element has been accessed with the EOL bit set, the next increment causes the cycle to repeat from the beginning.

The P580 scanner memory can hold 1024 steps. The memory configuration is 1024 by 23 bits. The following diagram shows the bit layout for each scanner memory word.



						GND Path A	INST 4 Path A	INST 3 Path A	INST 2 Path A	INST 1 Path A	Channel Path A
--	--	--	--	--	--	---------------	------------------	------------------	------------------	------------------	----------------

End Of List**Bit 31**

This bit is used to signify the end of the scanner memory list. This bit is set to a “1” to indicate that the current memory location is the last element in the list ; and that the scanner should jump to the beginning of the scanner memory when the next increment indication is received. If the bit is set to a “0”, the element is not the last element in the list and the next sequential memory location should be accessed when an increment indication is received.

Ground Path B**Bit 26**

This bit is used to connect analog Path B to ground. Setting this bit to a “1” causes the path to be grounded and a “0” causes the path to be removed from ground.

Instrument 4 Path B**Bit 25**

This bit is used to connect the Instrument 4 front-panel mounted SMB to Path B. Setting this bit to a “1” causes the instrument to be connected to Path B and a “0” to be removed from Path B.

Instrument 3 Path B**Bit 24**

This bit is used to connect the Instrument 3 front-panel mounted SMB to Path B. Setting this bit to a “1” causes the instrument to be connected to Path B and a “0” to be removed from Path B.

Instrument 2 Path B**Bit 23**

This bit is used to connect the Instrument 2 front-panel mounted SMB to Path B. Setting this bit to a “1” causes the instrument to be connected to Path B and a “0” to be removed from Path B.

Instrument 1 Path B**Bit 23**

This bit is used to connect the Instrument 1 front-panel mounted SMB to Path B. Setting this bit to a “1” causes the instrument to be connected to Path B and a “0” to be removed from Path B.

Channel Path B**Bits 21 –16**

This field is used to control which differential pair of the front-panel 68-position SCSI connector is to be connected to Path B. Valid values for this field range from 0 to 34. Values of 1 through 34 refer to channels 1 through 34 on the connector. A value of 0 in this field removes all channels from Path B.

Ground Path A**Bit 10**

This bit is used to connect analog Path A to ground. Setting this bit to a “1” causes the path to be grounded and a “0” causes the path to be removed from ground.

Instrument 4 Path A**Bit 9**

This bit is used to connect the Instrument 4 front-panel mounted SMB to Path A. Setting this bit to a “1” causes the instrument to be connected to Path B and a “0” to be removed from Path A.

Instrument 3 Path A**Bit 8**

This bit is used to connect the Instrument 3 front-panel mounted SMB to Path A. Setting this bit to a “1” causes the instrument to be connected to Path A and a “0” to be removed from Path A.

Instrument 2 Path A**Bit 7**

This bit is used to connect the Instrument 2 front-panel mounted SMB to Path A. Setting this bit to a “1” causes the instrument to be connected to Path A and a “0” to be removed from Path A.

Instrument 1 Path A**Bit 6**

This bit is used to connect the Instrument 1 front-panel mounted SMB to Path A. Setting this bit to a “1” causes the instrument to be connected to Path A and a “0” to be removed from Path A.

Channel Path A**Bits 5-0**

This field is used to specify which differential pair of the front-panel 68-position SCSI connector is to be connected to Path A. Values for this field range from 0 to 34. Values of 1 through 34 refer to channels 1 through 34 on the connector. A value of 0 in this field removes all channels from Path A.

The Scanner Memory start at offset 1000h from the base address of the P580. Each subsequent Scanner Memory location is accessed by adding 4 to the previous address location. The following shows the arrangement of the memory offsets.

Address Offset from Base Address	Scanner Memory Location
1000h	Location 1
1004h	Location 2
1008h	Location 3
1FF8h	Location 1023
1FFCh	Location 1024

Chapter 3: Programming Information

The software that accompanies the P580 must support all specified frameworks in PXI, as well as appropriate software to operate the unit on a CPCI platform. The software for the PXI environment includes a Plug and Play (PNP) driver as well as application examples. The PNP driver utilizes VISA calls to communicate with the P580 hardware. The VISA layer is responsible for translating the PNP driver calls into register I/O operations to communicate with the actual hardware.

For the PXI platform, the PNP driver and associated software must support the following programming languages:

- 'C'
- Visual Basic
- Labview
- ATEasy

In each of these languages, an appropriate number of application examples must be generated to show usage of the PNP driver. These examples also be a vehicle for demonstrating the functionality of the device. One should not have one application example that shows all the devices' functionality, but rather split them up into multiple examples. The examples should range from basic operation to more complex operations.

For the CPCI platform, there is no requirement for the controller of the CPCI chassis to support the VISA standard. For the CPCI platform, we will write a generic VISA layer for mimicking the operation of the VISA layer on the PXI platform. This VISA layer does not need to be a full implementation of the VISA specification. The VISA commands included in this layer need only fulfill the requirements of the PNP drivers that sit on top of them. Also, a Resource Manager (RESMAN) must be developed for identifying CPCI devices on the bus that can be controlled with the PNP drivers.

It is anticipated, and must be a design goal, to have only one PNP driver that can function in both the PXI and CPCI environments without modification. The only difference in between the 2 platforms is the addition of a reduced functionality VISA layer in the CPCI platform.

Initialize Function	
Initialize	ksp580_init

Configuration Functions	
Set Instrument Connection	ksp580_setInstrumentConnection
Get Instrument Connection	ksp580_getInstrumentConnection
Set Path To Ground	ksp580_setPathGround
Get Path To Ground	ksp580_getPathGround
Set Channel Connection	ksp580_setChannelConnection
Get Channel Connection	ksp580_getChannelConnection
Set Scanner Trigger Input Configuration	ksp580_setScannerTriggerInputConfiguration
Get Scanner Trigger Input Configuration	ksp580_getScannerTriggerInputConfiguration
Set Scanner Trigger Output Configuration	ksp580_setScannerTriggerOutputConfiguration
Get Scanner Trigger Output Configuration	ksp580_getScannerTriggerOutputConfiguration
Set Scanner Memory Path A	ksp580_setScannerMemoryPathA
Get Scanner Memory Path A	ksp580_getScannerMemoryPathA
Set Scanner Memory Path B	ksp580_setScannerMemoryPathB
Get Scanner Memory Path B	ksp580_getScannerMemoryPathB
Set Scanner Memory End Of List	ksp580_setScannerMemoryEndOfList
Get Scanner Memory End Of List	ksp580_getScannerMemoryEndOfList
Set Scanner Enable	ksp580_setScannerEnable
Get Scanner Enable	ksp580_getScannerEnable

Action and Status Functions	
Clear Path Connections	ksp580_ClearPathConnections

Utility Functions	
Get Device List	ksp580_getDeviceList
Get Module Suffix	ksp580_getModuleSuffix
Error Message	ksp580_error_message
Query Revision	ksp580_revision_query
Reset	ksp580_reset

Close Function	
Close	ksp580_close

Initialization Functions

ksp580_init

Syntax

C:

```
ViStatus _VI_FUNC ksv635_init (ViChar rsrcName[], ViBoolean bld_query,
                              ViBoolean bReset, ViPSession vi);
```

Visual Basic:

```
Function ksv635_init As ViStatus (rsrcName As ViChar, ByVal bld_query As ViBoolean,
                                ByVal bReset As ViBoolean, vi As ViSession)
```

Remarks:

ksp580_init establishes communication with a P580 and, if enabled, resets and queries the device. Once these operations are complete, an instrument handle is returned for subsequent communication with the device. The initialize function call queries the instrument specified in the session and makes sure that the unit is a P580. For each instrument handle used to call the initialize function, a corresponding **ksp580_close** should be executed prior to termination of the application program.

The **rsrcName** parameter is passed into this function and specifies the resource string to be initialized. The format of the resource name string is as follows:

TBD

bld_query is set to either KSP580_QUERY_ENABLE (VI_TRUE) or KSP580_QUERY_DISABLE (VI_FALSE). When set to VI_TRUE, a query of the hardware is executed to verify that the instrument handle is accessing a P580. When set to VI_FALSE, the query is not performed.

bReset is set to either KSP580_RESET_ENABLE (VI_TRUE) or KSP580_RESET_DISABLE (VI_FALSE). When set to enabled, a hardware reset operation is done to the P580 which restores the device to its default configuration. If set to disabled, a reset operation is not executed.

vi is the returned instrument handle. This handle is used for subsequent communication with the device.

Return Values

Return Value	Description
VI_SUCCESS	Successful
VI_ERROR_INV_OBJECT	Invalid Instrument Handle
VI_ERROR_INV_SETUP	Invalid Setup Information
VI_ERROR_RSRC_NFOUND	Resource Not Found
VI_ERROR_FAIL_ID_QUERY	Device Query Failed
VI_ERROR_ALLOC	Error Allocating Memory

ksp580_setInstrumentConnection

Syntax

C:

```
ViStatus _VI_FUNC ksp580_setInstrumentConnection (ViSession vi, ViInt16 ilInstrument1,
                                                ViInt16 ilInstrument2, ViInt16 ilInstrument3,
                                                ViInt16 ilInstrument4);
```

Visual Basic:

```
Function ksp580_setInstrumentConnection As ViStatus (ByVal vi As ViSession,
                                                    ByVal ilInstrument1 As ViInt16,
                                                    ByVal ilInstrument2 As ViInt16,
                                                    ByVal ilInstrument3 As ViInt16,
                                                    ByVal ilInstrument4 As ViInt16)
```

Remarks:

ksp580_setInstrumentConnection controls the connection of the four front-panel mounted SMB's to the internal analog paths. Each of the four instrument inputs can be connected to Path A, Path B, both Path A and Path B, or disconnected from both.

The **vi** parameter is the instrument handle. This handle is used to uniquely identify a communication channel between the driver and the instrument. For every instrument, a unique handle exists.

The **ilInstrument1 – ilInstrument4** parameters represent the four instrument connections on the front-panel of the P580. For each parameter, the connection path must be specified. Each instrument can independently be connected to Path A, Path B, Path A and Path B, or disconnected.

The following table shows the constants defined for the connection paths.

Connection Path	Constant
No Connection to Either Path	KSP580_NO_CONNECTION
Connection To Path A Only	KSP580_PATHA
Connection To Path B Only	KSP580_PATHB
Connection To Both Path A and Path B	KSP580_PATHAB

Return Values

Return Value	Description
VI_SUCCESS	Successful
VI_ERROR_INV_OBJECT	Invalid Instrument Handle
VI_ERROR_INV_SETUP	Invalid Setup Information
KSP580_INVALID_CONNECTION	Invalid Connection Path

ksp580_getInstrumentConnection

Syntax

C:

```
ViStatus _VI_FUNC ksp580_getInstrumentConnection (ViSession vi, ViInt16 *iInstrument1,
                                                ViInt16 *iInstrument2, ViInt16 *iInstrument3,
                                                ViInt16 *iInstrument4);
```

Visual Basic:

```
Function ksp580_getInstrumentConnection As ViStatus (ByVal vi As ViSession,
                                                    iInstrument1 As ViInt16,
                                                    iInstrument2 As ViInt16,
                                                    iInstrument3 As ViInt16,
                                                    iInstrument4 As ViInt16)
```

Remarks:

ksp580_getInstrumentConnection returns the current connection of the four front-panel mounted SMB's to the internal analog paths. Each of the four instrument inputs may be connected to Path A, Path B, both Path A and Path B, or disconnected from both.

The **vi** parameter is the instrument handle. This handle is used to uniquely identify a communication channel between the driver and the instrument. For every instrument, a unique handle exists.

The **iInstrument1 – iInstrument4** parameters represent the four instrument connections on the front-panel of the P580. Each instrument parameter returns the current internal analog path connection. Each instrument can independently be connected to Path A, Path B, Path A and Path B, or disconnected.

The following table shows the values returned for the connection paths.

Connection Path	Constant
No Connection to Either Path	KSP580_NO_CONNECTION
Connection To Path A Only	KSP580_PATHA
Connection To Path B Only	KSP580_PATHB
Connection To Both Path A and Path B	KSP580_PATHAB

Return Values

Return Value	Description
VI_SUCCESS	Successful
VI_ERROR_INV_OBJECT	Invalid Instrument Handle
VI_ERROR_INV_SETUP	Invalid Setup Information

ksp580_setPathGround

Syntax

C:

```
ViStatus _VI_FUNC ksp580_setPathGround (ViSession vi, ViInt16 iPath, ViBoolean iGnd);
```

Visual Basic:

```
Function ksp580_setPathGround As ViStatus (ByVal vi As ViSession,  
                                           ByVal iPath As ViInt16,  
                                           ByVal iGnd As ViBoolean )
```

Remarks:

ksp580_setPathGround connects or removes ground connections to a specified path. This function can connect or disconnect ground from either Path A, Path B, or both.

The **vi** parameter is the instrument handle. This handle is used to uniquely identify a communication channel between the driver and the instrument. For every instrument, a unique handle exists.

The **iPath** parameter specifies the path(s) to be acted upon. The path information can be Path A, Path B, or both paths.

The following table shows the constants defined for the connection paths.

Connection Path	Constant
Ground Path A Only	KSP580_PATHA
Ground Path B Only	KSP580_PATHB
Ground Both Path A and Path B	KSP580_PATHAB

The **iGnd** parameter specifies whether the path(s) specified in the **iPath** parameter is to be connected to ground or removed from ground. A value of **VI_TRUE** connects the specified path(s) to ground and a value of **VI_FALSE** disconnects the specified path(s) from ground.

Return Values

Return Value	Description
VI_SUCCESS	Successful
VI_ERROR_INV_OBJECT	Invalid Instrument Handle
VI_ERROR_INV_SETUP	Invalid Setup Information
KSP580_INVALID_CONNECTION	Invalid Connection Path

ksp580_getPathGround

Syntax

C:

```
ViStatus _VI_FUNC ksp580_getPathGround (ViSession vi, ViInt16 iPath, ViBoolean *iGnd);
```

Visual Basic:

```
Function ksp580_setPathGround As ViStatus (ByVal vi As ViSession,  
                                           ByVal iPath As ViInt16,  
                                           iGnd As ViBoolean )
```

Remarks:

ksp580_getPathGround returns information regarding the connection of a path to ground. This function can only be used to access one path at a time.

The **vi** parameter is the instrument handle. This handle is used to uniquely identify a communication channel between the driver and the instrument. For every instrument, a unique handle exists.

The **iPath** parameter specifies the path that is to be queried. The path information can be Path A or Path B.

The following table shows the constants defined for the connection paths.

Connection Path	Constant
Path A is Grounded	KSP580_PATHA
Path B is Grounded	KSP580_PATHB

The **iGnd** parameter returned reflects whether the path specified in the **iPath** parameter is connected to ground or removed from ground. A value of VI_TRUE indicates that the specified path is connected to ground and a value of VI_FALSE indicates that the specified path is not grounded.

Return Values

Return Value	Description
VI_SUCCESS	Successful
VI_ERROR_INV_OBJECT	Invalid Instrument Handle
VI_ERROR_INV_SETUP	Invalid Setup Information
KSP580_INVALID_CONNECTION	Invalid Connection Path

ksp580_setChannelConnection

Syntax

C:

```
ViStatus _VI_FUNC ksp580_setChannelConnection (ViSession vi, ViUInt32 IPathAChannelLo,
                                              ViUInt32 IPathAChannelHi,
                                              ViUInt32 IPathBChannelLo,
                                              ViUInt32 IPathBChannelHi);
```

Visual Basic:

```
Function ksp580_setChannelConnection As ViStatus (ByVal vi As ViSession,
                                                  ByVal IPathAChannelLo As ViUInt32,
                                                  ByVal IPathAChannelHi As ViUInt32,
                                                  ByVal IPathBChannelLo As ViUInt32,
                                                  ByVal IPathBChannelHi As ViUInt32)
```

Remarks:

ksp580_setChannelConnection. connects the specified channels to differential analog Path A or Path B.

The Channel Low and Channel High parameters specify the channels to be connected to the indicated path. The Channel Low parameters specifies channel numbers 1 through 32 and Channel High specifies channel numbers 33 and 34. Each parameter is a bit field with each bit assigned to a channel.

The **vi** parameter is the instrument handle. This handle is used to uniquely identify a communication channel between the driver and the instrument. For every instrument, a unique handle exists.

The **iPath** parameter specifies the path(s) to be acted upon. The path information can be Path A, Path B, or both paths.

The following table shows the constants defined for the connection paths.

Connection Path	Constant
Ground Path A Only	KSP580_PATHA
Ground Path B Only	KSP580_PATHB
Ground Both Path A and Path B	KSP580_PATHAB

The **iGnd** parameter specifies whether the path(s) specified in the **iPath** parameter is to be connected to ground or removed from ground. A value of VI_TRUE connects the specified path(s) to ground and a value of VI_FALSE disconnects the specified path(s) from ground.

Return Values

Return Value	Description
VI_SUCCESS	Successful
VI_ERROR_INV_OBJECT	Invalid Instrument Handle
VI_ERROR_INV_SETUP	Invalid Setup Information
KSP580_INVALID_CONNECTION	Invalid Connection Path

ksp580_getChannelConnection

Syntax

C:

```
ViStatus _VI_FUNC ksp580_setChannelConnection (ViSession vi, ViUInt32 IPathAChannelLo,
                                              ViUInt32 IPathAChannelHi,
                                              ViUInt32 IPathBChannelLo,
                                              ViUInt32 IPathBChannelHi);
```

Visual Basic:

```
Function ksp580_setChannelConnection As ViStatus (ByVal vi As ViSession,
                                                  ByVal IPathAChannelLo As ViUInt32,
                                                  ByVal IPathAChannelHi As ViUInt32,
                                                  ByVal IPathBChannelLo As ViUInt32,
                                                  ByVal IPathBChannelHi As ViUInt32)
```

Remarks:

ksp580_setChannelConnection. connects the specified channels to differential analog Path A or Path B. The Channel Low and Channel High parameters specify the channels to be connected to the indicated path. The Channel Low parameters specifies channel numbers 1 through 32 and Channel High specifies channel numbers 33 and 34. Each parameter is a bit field with each bit assigned to a channel. The **vi** parameter is the instrument handle. This handle is used to uniquely identify a communication channel between the driver and the instrument. For every instrument, a unique handle exists. The **iPath** parameter specifies the path(s) to be acted upon. The path information can be Path A, Path B, or both paths.

The following table shows the constants defined for the connection paths.

Connection Path	Constant
Ground Path A Only	KSP580_PATHA
Ground Path B Only	KSP580_PATHB
Ground Both Path A and Path B	KSP580_PATHAB

The **iGnd** parameter specifies whether the path(s) specified in the **iPath** parameter is to be connected to ground or removed from ground. A value of VI_TRUE connects the specified path(s) to ground and a value of VI_FALSE disconnects the specified path(s) from ground.

Return Values

Return Value	Description
VI_SUCCESS	Successful
VI_ERROR_INV_OBJECT	Invalid Instrument Handle
VI_ERROR_INV_SETUP	Invalid Setup Information
KSP580_INVALID_CONNECTION	Invalid Connection Path

