VDSP31

HIGH PERFORMANCE
EIGHT CHANNEL
DIGITAL SERVO CONTROLLER

FIRMWARE REFERENCE MANUAL

Revision 1.2
February 27, 1996

ALPHI TECHNOLOGY CORPORATION
6202 S. Maple Avenue #128
Tempe, AZ 85283 USA
Tel : (602) 838-2428
Fax: (602) 838-4477

NOTICE

The information in this document has been carefully checked and is believed to be entirely reliable. While all reasonable efforts to ensure accuracy have been taken in the preparation of this manual, ALPHI TECHNOLOGY assumes no responsibility resulting from omissions or errors in this manual, or from the use of information contain herein.

ALPHI TECHNOLOGY reserves the right to make any changes, without notice, to this or any of ALPHI TECHNOLOGY's products to improve reliability, performance, function or design.

ALPHI TECHNOLOGY does not assume any liability arising out of the application or use of any product or circuit described herein; nor does ALPHI TECHNOLOGY convey any license under its patent rights or the rights of others.

INTRODUCTION

The VDSP31 is a VXI based, digital servo control module. The VDSP31 offers a complete solution for high performance closed loop applications. The VDSP31 has been designed to solve specific problems related to the control of hydraulic actuators. Signal conditioning modules can be combined on a per channel basis to configure the VDSP31 for load or displacement feedback. The VDSP31 contains application firmware that will control up to eight hydraulic actuators. The board can be programmed using SCPI commands or by directly accessing the VDSP31 registers through the dual-access RAM. Shown below in **Figure 1** is a functional block diagram of the system. Host commands are processed by the VXIbus command processor and routed to the various subsystems. The Sine and Endpoint blocks are used to generate setpoints for the servo. The composite setpoint is fed to a control law that calculates a control output based on the actuator feedback. The dither generator applies a small amplitude waveform to the valve to eliminate valve "stiction".

VDSP31 SOFTWARE COMPONENTS

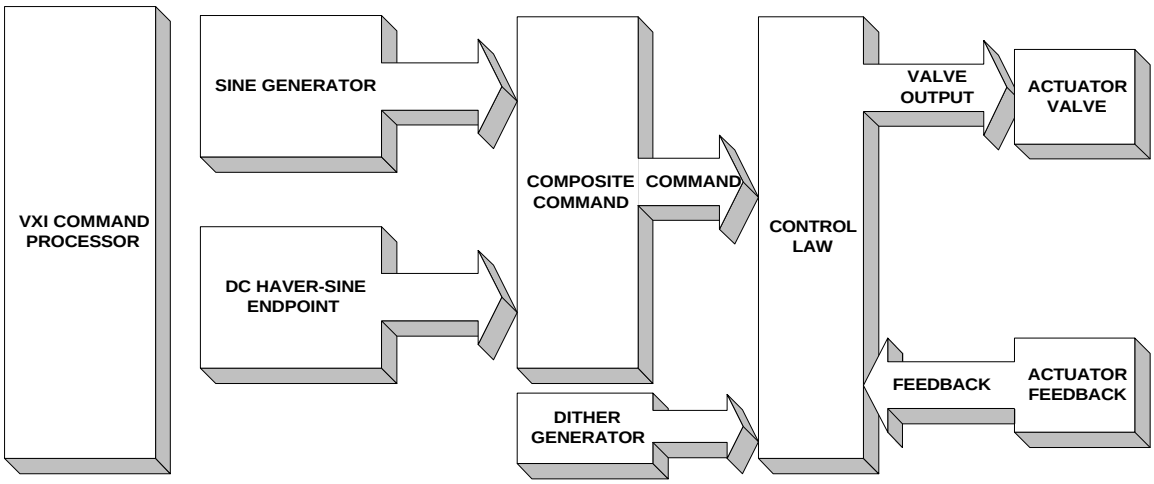

Figure 1

PROGRAMMING THE VDSP31

VXI TO VDSP COMMUNICATION

The VXI host communicates with the VDSP31 firmware through a shared RAM interface. The VXI host can download setup and configuration data for each servo channel. Once configured, the VDSP31 firmware will wait for commands to be sent by the VXI host. The VDSP31 can be programmed to interrupt to the VXI host upon command completion or if an error occurs. The VDSP31 is designed to operate on its own without VXI host intervention. The VXI host need only communicate with the VDSP31 to send new commands or to check status. The VDSP31 maintains time history information and status for each channel such that the VXI host can monitor servo performance.

VXI A16 REGISTER MAP

The VDSP incorporates all the configuration registers and message-based device registers required by the VXIbus specification. All registers accessible by the VXIbus conform to the definitions and rules given in the VXIbus specification. The register map for the VXIbus is shown below :

OFFSET	FUNCTION
00	ID/LOGICAL ADDRESS
02	DEVICE TYPE
04	STATUS/CONTROL
06	OFFSET
08	PROTOCOL/SIGNAL
0A	RESPONSE
0C	UNUSED
0E	DATA LOW
10	A24 POINTER HIGH
12	A24 POINTER LOW
14	A32 POINTER HIGH
16	A24 POINTER LOW
18 - 1E	VXI RESERVED

Table 1

A16 ADDRESS

The VDSP31 is setup at the factory for dynamic configuration. The power-up logical address is 255. The VXI resource manager will assign the VDSP31 a logical address during system configuration. To VDSP31 can be located at a fixed address by installing jumpers as described in the VDSP31 User's Manual.

A32/A24 ADDRESS SPACE CONFIGURATION

The VDSP shared RAM can be located in either A24 or A32 space. The A32/A24 address space selection can be changed by running the VDSP setup utility via the local serial port. Connect a PC or terminal to the VDSP system console. The serial communication settings should be 19200 baud, 8 bits, no parity. The VDSP console program uses ANSI / VT100 escape sequences so set your terminal emulation to match as required. Next, press the "Reset" button on the VDSP31 front panel. After a brief delay, the main screen should appear displaying the VDSP31 firmware revision and Flash memory status. Press the "Menu" button and a menu should appear as shown below in **Figure 2**:

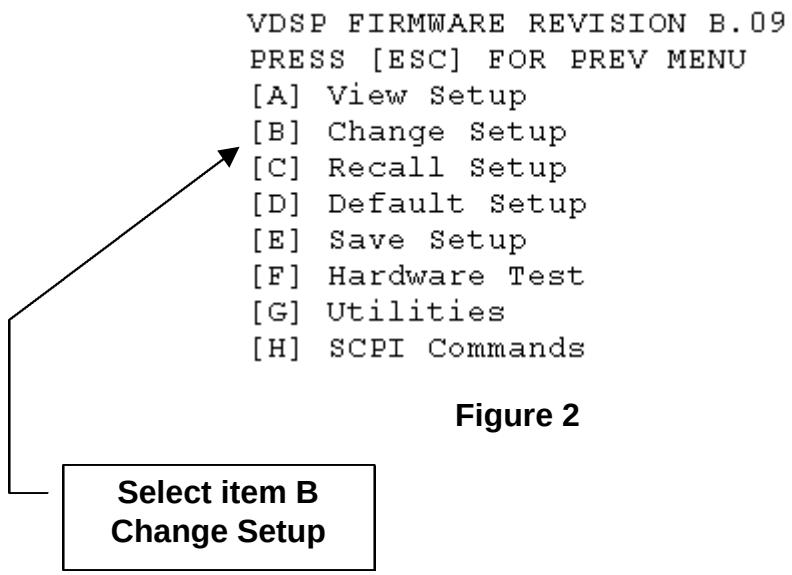

Figure 2

Now, answer the questions as shown below in **Figure 3**. Press <Enter> to accept the default responses.

```
Number of active channels..... (0 - 8), default = 0 : 8

SCPI commands from console..... [N] ?
32 channel A/D board installed. [N] ?
8 channel board installed..... [N] ? Yes
Mezzanine DSP board installed.. [N] ?
VXI A32/D32 access............ [N] ? Yes

Channel 00 Configuration:
Channel disabled.............. [N] ?
```

Figure 3

To enable VXIbus A32/D32 access, press 'Y'. Press 'N' for A24/D16 access.

When the configuration is complete, return to the main menu to save your changes to the Flash memory. The VDSP31 will use the new settings after the next reset.

```
VDSP FIRMWARE REVISION B.09
PRESS [ESC] FOR PREV MENU
[A] View Setup
[B] Change Setup
[C] Recall Setup
[D] Default Setup
[E] Save Setup
[F] Hardware Test
[G] Utilities
[H] SCPI Commands
```

Figure 4

Select item E Save Setup

PROGRAMMING THE SETPOINT GENERATOR

Please refer to **Figure 5** during this discussion. The shaded boxes represent user programmable parameters. The Setpoint Generator is a digital function generator that drives the PID controller command on each channel. The generator contains an AC section and a DC section. The AC section contains a 256 point +/- 1 volt sine wave lookup table. The VDSP maintains a read pointer that increments at a rate determined by the **AC Period** parameter. The AC period value applies to all channels. Phase for each channel is calculated by adding a fixed offset to the table read pointer. The **AC Phase** parameter is used to change the phase for each channel. The AC amplitude for each channel is calculated by multiplying a span constant with the current table value. The **AC Span** parameter is used to vary the amplitude. The DC section contains a 256 point +/- 1 volt haver-sine lookup table which spans -90.0 to +90.0 degrees of a sinusoid. The VDSP maintains a read pointer unique to each channel for use as an index into this table. The DC index increments at a rate determined by the **DC Period** parameter. The DC index will begin to change when a channel has received a new **DC Offset** command. The outputs of the AC and DC sections are combined to form a composite command that is sent to the PID summing junction.

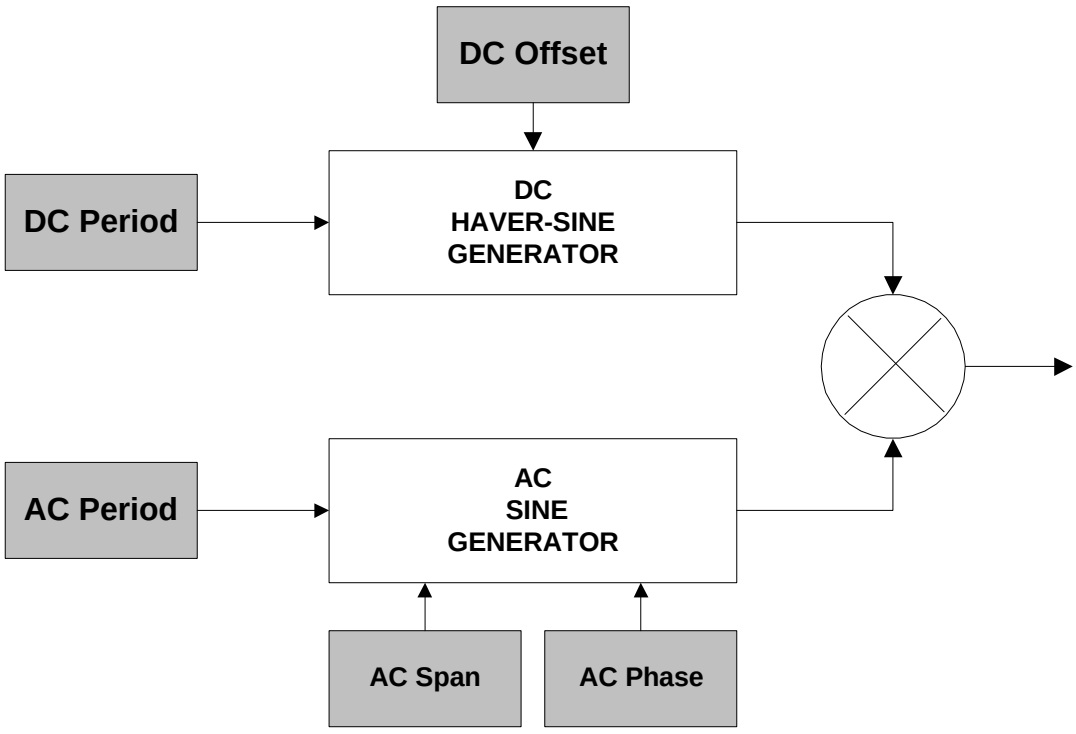

Figure 5

As shown below in **Figure 6**, the setpoint generator can be used generate complex DC transitions and AC waveforms.

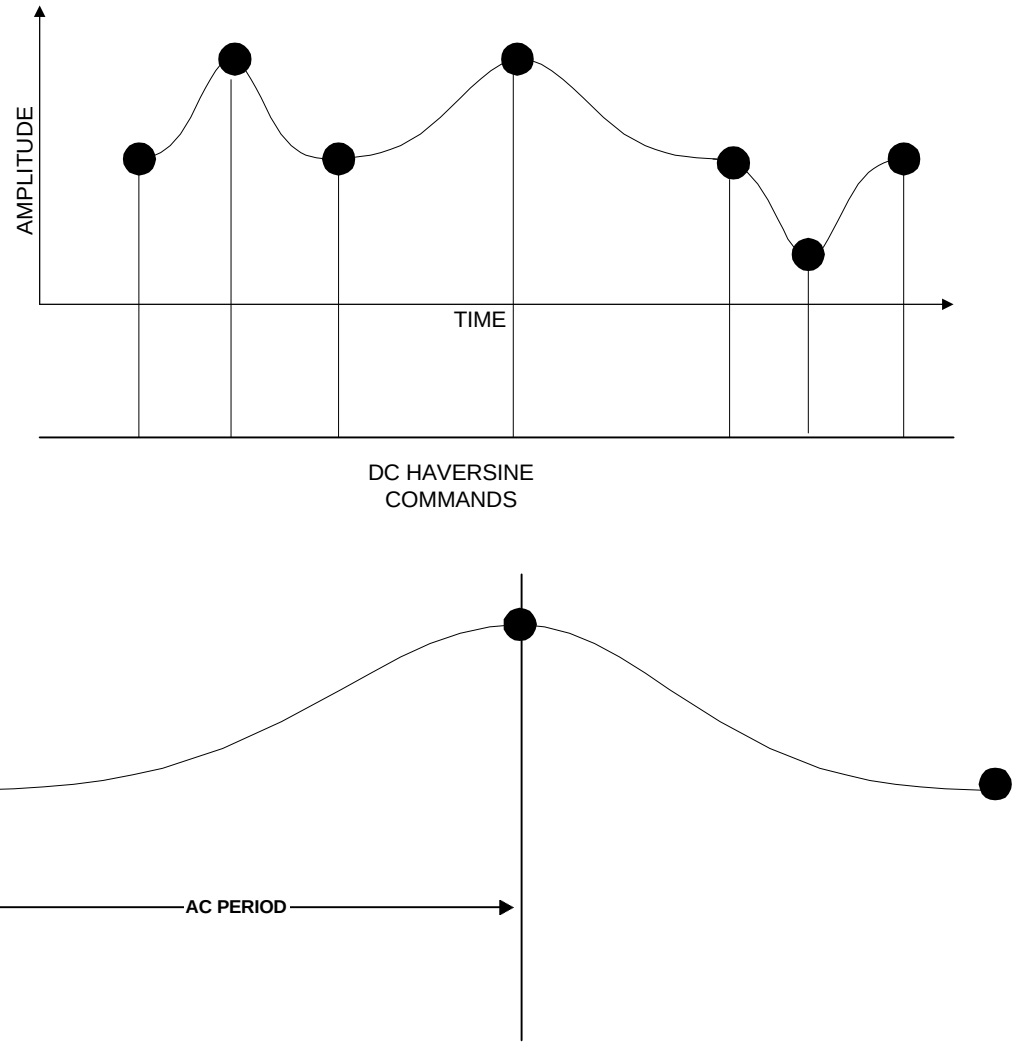

Figure 6

PROGRAMMING THE FEEDBACK SUBSYSTEM

Please refer to **Figure 7** for this discussion. The shaded boxes represent user programmable parameters. External low-level bridge input signals from the front panel connectors are fed to a programmable instrumentation amplifier. The amplifier gain is programmable in fixed ranges of 1,2,4,8,100,200,400 and 800. The **PGA Range** parameter is used to select the gain range. The output of this amplifier is summed with a D/A converter that offsets or nulls the amplifier output from -10.0 to + 10.0 volts DC. This helps preserve the dynamic range of the input signal. The **PGA Offset** parameter is used to provide the DC offset. The VDSP is able to perform software polarity switching for the high level feedback signal. The **Polarity** parameter is used to select normal or reverse polarity. A second D/A provides bridge excitation from 0.0 to 10.0 volts DC. The **DC Excitation** parameter is used to change the excitation output. A calibration relay can apply a shunt calibration resistor to one leg of a bridge transducer. The VDSP is supplied with 100 KΩ precision resistors.

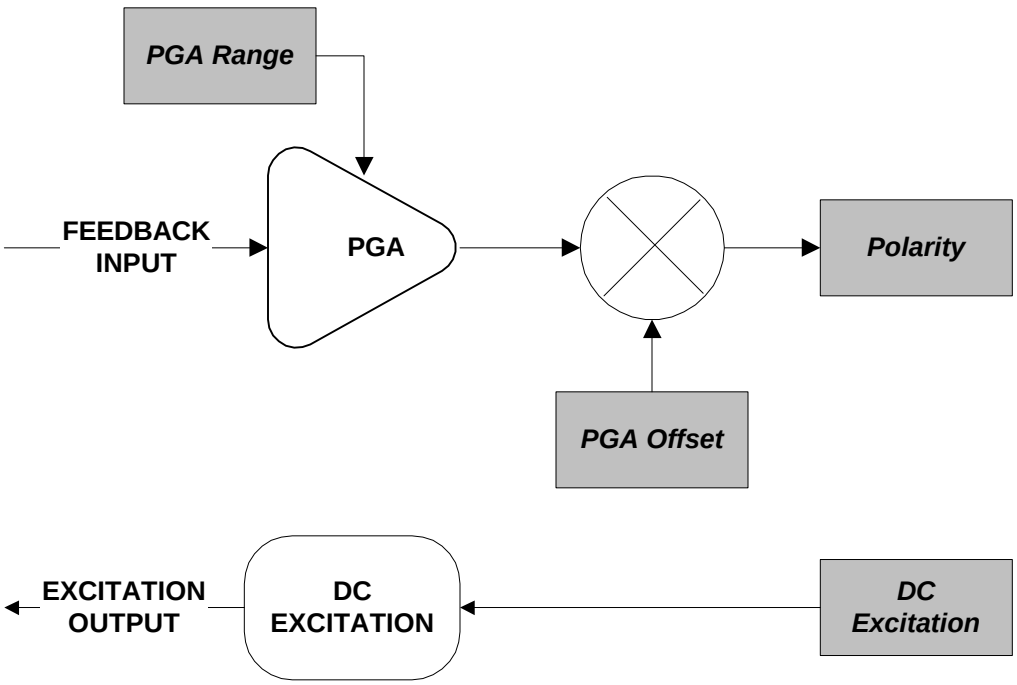

Figure 7

PROGRAMMING THE CONTROL AND VALVE SUBSYSTEMS

Please refer to **Figure 8** for this discussion. The shaded boxes represent user programmable parameters. Each VDSP channel contains a valve output D/A with 16 bit resolution. The output of the valve D/A is converted into a current and made available to the front panel connectors. The VDSP performs PID calculations on each channel 1000 times per second. The result of this calculation is a voltage that is supplied to the valve D/A. The VDSP sums a 500 Hz square wave dither signal with the final valve output. The **DITHER AMPLITUDE** parameter is used to set the dither level. The VDSP is able to reverse the valve voltage polarity. The **POLARITY** parameter is used to select the valve polarity. The **VALVE OFFSET** parameter can be used to provide an electronic offset for mechanical valves that are not balanced. The **ILIMIT** parameter is used to bound the absolute value for the integrator at a fixed voltage. The **DSPER** parameter is used to change the sampling interval for the derivative function. The **P GAIN** parameter is used to set the Proportional gain. The **I GAIN** parameter is used to set the Integral gain. The **D GAIN** parameter is used to set the Derivative gain.

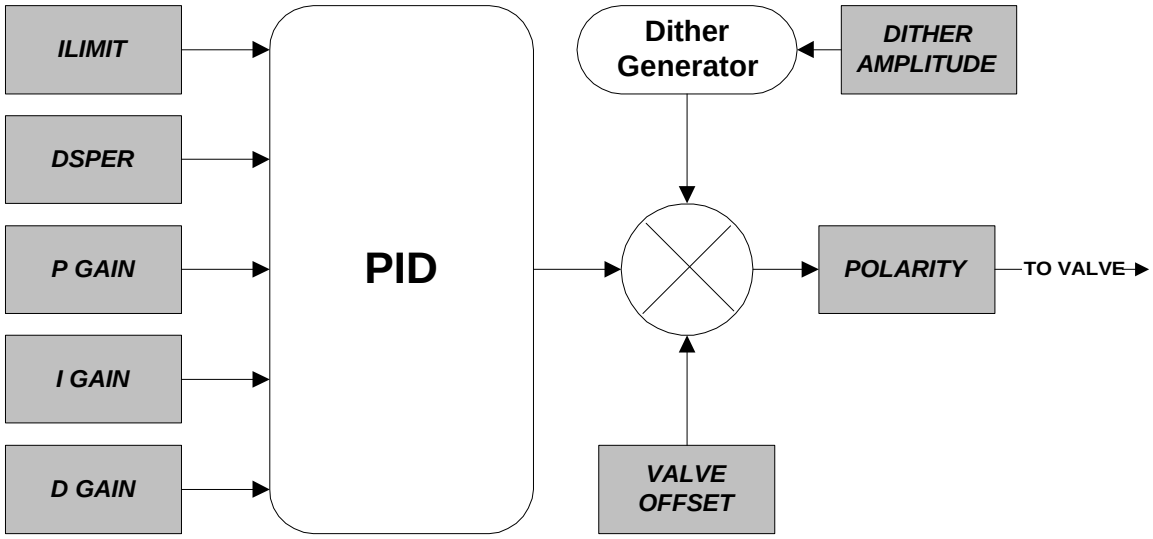

Figure 8

MONITORING INTERNAL VARIABLES

The VDSP firmware converts selected internal variables into voltage signals. These voltage signals are made available at the VDSP front panel through four miniature BNC connectors. The VDSP composite setpoint command, feedback, error and valve outputs for a single channel can be viewed by connecting a scope or other monitoring device to these connectors. **Figure 9** below depicts the association of VDSP variables to front panel connectors. The firmware command **FP_CHANNEL** can be used to change which VDSP channel is sent to the front panel monitor.

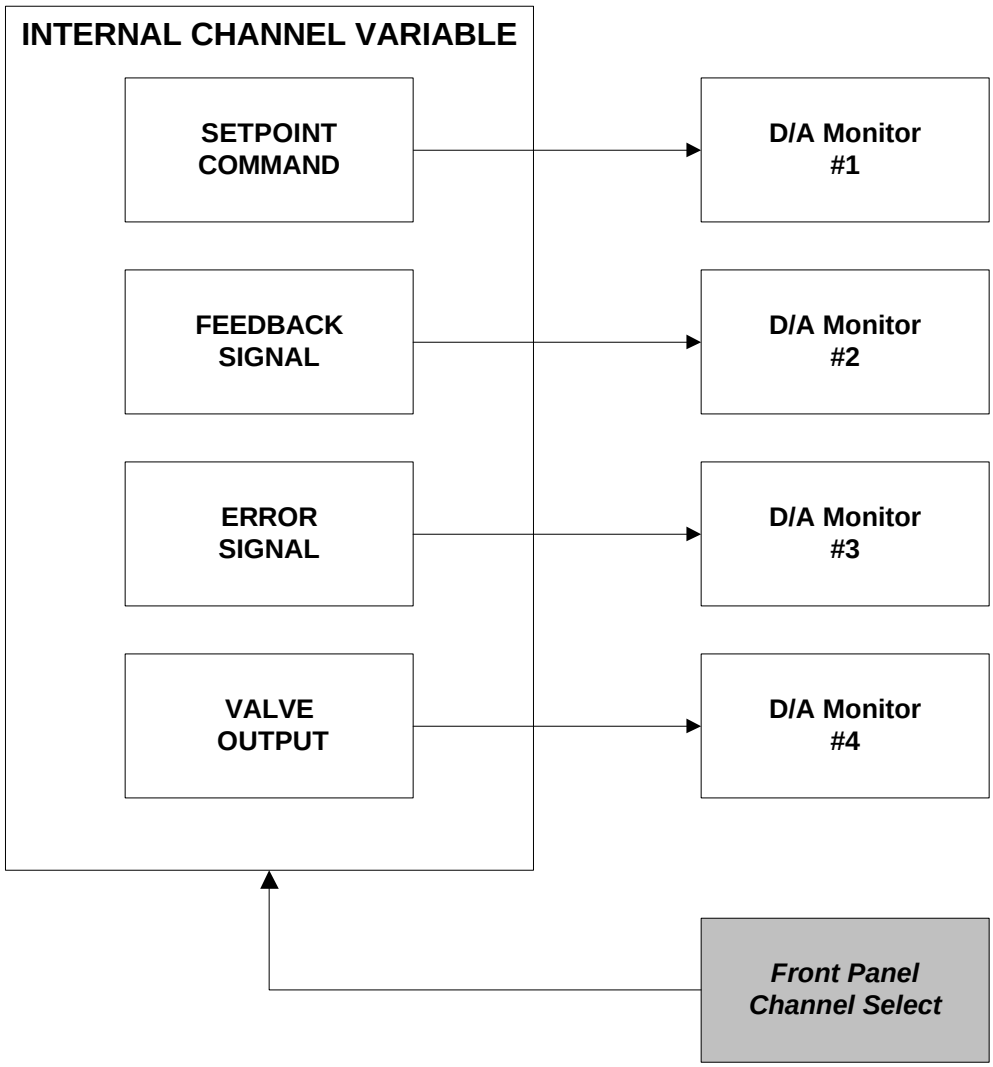

Figure 9

DISCRETE FAULT OUTPUT

The VDSP31 contains a watch dog safety circuit. The watch dog circuitry must be accessed by the DSP at regular intervals. If the DSP fails to perform the required access for any reason, the watch dog will pull an output low to inform the outside world that the VDSP31 hardware or software has failed. The watch dog fail output is also used to hold valve driver module D/A hardware in reset (0v). This signal can be connected to external relays, indicators other hardware. The output will sink up to 30 mA at 28 VDC. Watch dog outputs from multiple VDSP31 boards can be connected in parallel if required.

SENDING COMMANDS TO THE VDSP31

The VDSP contains a dual access memory that is shared between the VDSP CPU and the VXIbus. This memory space is referred to as Host Variable Space (HVS). The first HVS section is shown below in **Table 2** :

HOST COMMAND VARIABLES

OFFSET	REGISTER NAME	DATA TYPE	R/W	PURPOSE
0000	CMD_CODE	UINT32	W	Command Code to execute
0004	CMD_STAT	UINT32	R	Command status
0008	ERR_CODE	UINT32	R	Command error code
000C	FIRST_CHAN	UINT32	R/W	First channel affected by command
0010	LAST_CHAN	UINT32	R/W	Last channel affected by command
0014	IRQ_ENABLE	UINT32	R/W	VXI Interrupt enable mask
0018	IRQ_STAT	UINT32	R	Interrupt status
001C	IRQ_LEVEL	UINT32	R/W	Interrupt Level
0020	IRQ_VECTOR	UINT32	R/W	Interrupt Vector

Table 2

CMD_CODE

Writing a command code to this register will instruct the VDSP to execute a command. The VDSP polls this register at regular intervals, checking for a non-zero value. The VDSP will clear this register when command processing begins. The data parameters associated with a command code must be written by the VXIbus host before writing to the CMD_CODE register. Some command codes can affect more than one VDSP channel. The range of channels to which the command code should be applied are determined by the FIRST_CHAN and LAST_CHAN registers. Valid command codes are 1 - 21. Currently supported command code are described below in **Table 3**:

VDSP31 COMMAND CODE TABLE

CODE	FUNCTION
1	CAL_RELAY
2	DITHER_AMPLITUDE
3	VALVE_OFFSET
4	VALVE_POLARITY
5	FBK_OFFSET
6	FBK_POLARITY
7	PGA_GAIN
8	DC_EXCITATION
9	PID_GAINS
10	PID_ENABLE
11	FP_CHANNEL
12	DC_OFFSET
13	AC_AMPLITUDE
14	AC_PHASE
15	MASTER_SPAN
16	AC_PERIOD
17	DC_PERIOD
18	TRIGGER_STATE
19	DIGITAL_OUTPUTS
20	*BOOT_STRAP RESERVED*
21	RAMP_TYPE

Table 3

CMD_STAT

This register indicates the status of the last command written to the CMD_CODE register. This register will remain zero until a command completes.

BIT	FUNCTION
0	**1 = Ready for Command / OK, 0 = BUSY**
1	**1 = Command Error**

ERR_CODE

This register will contain an error code for commands that did not complete properly. The following codes are currently defined:

Code		Meaning
0	**ERR_NONE**	**No Error, OK**
1	**ERR_BAD_CHAN**	**The specified channel range was invalid**
2	**ERR_BAD_CMD**	**The specified command code was invalid**
3	**ERR_BAD_VALUE**	**The data value(s) are out of range**

FIRST_CHAN

This register specifies the first channel to be used for commands that operate on a range of channels. This register, in combination with the LAST_CHAN register, can be used to specify a range of channels to be processed. For example, FIRST_CHAN = 0 and LAST_CHAN = 0 specifies only channel 0. Setting FIRST_CHAN = 0 and LAST_CHAN = 7 specifies all 8 channels.

LAST_CHAN

This register specifies the last channel to be used for commands that operate on a range of channels. This register, in combination with the FIRST_CHAN register, can be used to specify a range of channels to be processed. For example, FIRST_CHAN = 0 and LAST_CHAN = 0 specifies only channel 0. Setting FIRST_CHAN = 0 and LAST_CHAN = 7 specifies all 8 channels.

IRQ_ENABLE

This register enables or disables VXIbus interrupts from the VDSP. Currently, two interrupt enable bits are defined:

BIT	FUNCTION
0	1 = Enable MERR (Module Error) Interrupt
1	1 = Enable, CC (Command Complete) Interrupt

IRQ_STAT

This register indicates the VDSP interrupt status. Currently, two interrupt status bits are defined:

BIT	FUNCTION
0	1 = MERR (Module Error) IRQ
1	1 = CC (Command Complete) IRQ

MODULE VARIABLES

OFFSET	REGISTER NAME	DATA TYPE	R/W	PURPOSE
0024	AVAIL_CHAN	UINT32	R	Number of available channels
0028	MASTER_SPAN	IEEE-754	R/W	Master amplitude control for AC waveforms
002C	AC_PERIOD	IEEE-754	R/W	Period in seconds for one AC waveform cycle
0030	DC_PERIOD	IEEE-754	R/W	Period in seconds to complete a DC transition
0034	MSE	UINT32	R/W	Master Servo Enable
0038	FP_CHAN	UINT32	R/W	Front panel monitor channel selection
003C	DIG_IN	UINT32	R	32 bit digital input status
0040	DIG_OUT	UINT32	R/W	32 bit digital output image
0044	RAMP_TYPE	UINT32	R/W	Selects Ramp Type for DC transtions

Table 4

AVAIL_CHAN

This register indicates the number of active VDSP channels.

MASTER_SPAN

This register controls the output amplitude of all sinewave generators. The output from each sinewave generator is multiplied by the value in this register forming a master span or amplitude control for all channels. This register is useful for soft starts to ramp up the AC level and for soft stops to ramp down the AC level. The range for MASTER_SPAN is 0.0 to 1.0

AC_PERIOD

This register determines the period, in seconds, for one AC waveform cycle. The range of AC_PERIOD is .020 to 20.0 seconds. This range allows an AC frequency range of .05 to 50 Hz. The AC_PERIOD register is common to all VDSP channels.

DC_PERIOD

This register determines the period in seconds to complete a DC transition. The range of DC_PERIOD is .020 to 20.0 seconds. The DC_PERIOD register is common to all VDSP channels.

MSE

This register enables or disables the PID control loop for all VDSP channels. When MSE = 1 all channels can be enabled. Each VDSP channel has a PID_ENABLE register that can be used to enable or disable that channel. Therefore, a VDSP channel will only be enabled when its PID_ENABLE = 1 and MSE = 1. The VDSP firmware will force the composite command to equal the value of the current feedback for those channels that are disabled. This action forces the error at the summing junction to zero, resulting in zero valve output. NOTE: Valve offset and dither amplitude are not affected by MSE or PID_ENABLE.

FP_CHAN

This register determines which VDSP channel will be sent to the front panel channel monitor. The range of FP_CHAN is 0 to 7. By default, VDSP channel 0 is sent to the front panel monitor.

DIG_IN

This register indicates the current state of the 32 digital inputs.

DIG_OUT

Writing to this register will change the state of the 32 digital outputs.

RAMP_TYPE

This register determines what type of ramp will be used when the VDSP transitions from the current DC setpoint to a new DC setpoint. If RAMP_TYPE is a 1, Linear Ramp is selected. If RAMP_TYPE is a 0, Haver-Sine Ramp is selected.

CHANNEL VARIABLES

OFFSET	REGISTER NAME	DATA TYPE	R/W	PURPOSE
0048	FBK	IEEE-754	R	Feedback value in volts for channels 1 - 8
0068	CMD	IEEE-754	R	Composite command in volts for channels 1 - 8
0088	ERR	IEEE-754	R	Error value in volts for channels 1 - 8
00A8	VOUT	IEEE-754	R	Valve output value in volts for channels 1 - 8
00C8	DC_OFF	IEEE-754	R/W	DC Setpoint value in volts for channels 1 - 8
00E8	AC_AMP	IEEE-754	R/W	AC Setpoint value in volts for channels 1 - 8
0108	AC_PHS	IEEE-754	R/W	AC Phase value in degress for channels 1 - 8
0128	KP	IEEE-754	R/W	Proportional Gain for channels 1 - 8
0148	KI	IEEE-754	R/W	Integral Gain for channels 1 - 8

0168	KD	IEEE-754	R/W	Derivative Gain for channels 1 - 8
0188	IL	IEEE-754	R/W	Integration Limit in volts for channels 1 - 8
01A8	DS	UINT32	R/W	Derivative sample period for channels 1 - 8
01C8	CAL_RELAY	UINT32	R/W	Calibration Relay Control for channels 1 - 8
01E8	EXCITATION	IEEE-754	R/W	Bridge excitation output in volts for channels 1 - 8
0208	PGA_GAIN	UINT32	R/W	PGA Gain range for channels 1 - 8
0228	PGA_OFFSET	IEEE-754	R/W	PGA Offset in volts for channels 1 - 8
0248	DITHER_AMP	IEEE-754	R/W	Valve dither amplitude for channels 1 - 8
0268	FBK_POL	UINT32	R/W	Feedback polarity control for channels 1 - 8
0288	VLV_POL	UINT32	R/W	Valve polarity control for channels 1 - 8
02A8	PID_ENABLE	UINT32	R/W	PID control enable for channels 1 - 8
02C8	VLV_OFFSET	IEEE-754	R/W	Valve offset in volts for channels 1 - 8

Table 5

FBK

This is an array of IEEE-754 registers that contain the current feedback value in volts for servo channels 1 - 8. The PID control loop will update this array at regular intervals. The range for FBK is +/- 10.0 volts.

CMD

This is an array of IEEE-754 registers that contain the current composite command value in volts for servo channels 1 - 8. The composite command is a summation of the DC and AC setpoint generators. The PID control loop will update this array at regular intervals. The range for CMD is +/- 10.0 volts.

ERR

This is an array of IEEE-754 registers that contain the current error signal for servo channels 1 - 8. The error signal represents the difference between the FBK and CMD values. The PID control loop will update this array at regular intervals. The range for ERR is +/- 20.0 volts.

VOUT

This is an array of IEEE-754 registers that contain the current valve output voltage for channels 1 - 8. The valve output is a function of the PID calculation, valve offset and dither amplitude. The PID control loop will update this array at regular intervals.

DC_OFF

This is an array of IEEE-754 registers that determine the DC setpoint or offset for servo channels 1 - 8. These values are used to command the DC setpoint generator to ramp to a new DC target level. The move to the new DC level will be paced by the value specified in the DC_PERIOD register. The range for DC_OFF is +/- 10.0 volts.

AC_AMP

This is an array of IEEE-754 registers that determine the AC amplitude or span for servo channels 1 - 8. These values are used to command the AC setpoint generator to output a new AC amplitude. The period of the AC waveform is determined by the value written to the AC_PERIOD register. The range for AC_AMP is 0.0 to 10.0 volts. The MASTER_SPAN control can be used to scale the amplitude of all channels.

AC_PHS

This is an array of IEEE-754 registers that determine the AC phase for servo channels 1 - 8. These values are used to command the AC setpoint generator to output a new AC phase. The phase values are relative to channel 0. The range for AC_PHS is 0.0 to 360.0 degrees.

KP

This is an array of IEEE-754 registers that specify the Proportional Gain (Kp) factor used by the PID control loop for channels 1 - 8. The range for KP is 0.0 - 100.0.

KI

This is an array of IEEE-754 registers that specify the Integral Gain (Ki) factor used by the PID control loop for channels 1 - 8. The range for KI is 0.0 -100.0.

KD

This is an array of IEEE-754 registers that specify the Derivative Gain (Kd) factor used by the PID control loop for channels 1 - 8. The range for KD is 0.0 -100.0.

IL

This is an array of IEEE-754 registers that specify the Integration Limit factor used by the PID control loop for channels 1 - 8. The range for IL is 0.0 -10.0.

DS

This is an array of integer registers that specify the Derivative Sampling interval used by the PID control loop for channels 1 - 8. The range for DS is 2 - 32.

CAL_RELAY

This is an array of integer registers that specify the state of the calibration relays for servo channels 1 - 8. 1 = Relay Closed, 0 = Relay Open.

EXCITATION

This is an array of IEEE-754 registers that specify the DC bridge excitation voltage output for servo channels 1 - 8. The range for EXCITATION is 0.0 to 10.0 volts.

PGA_GAIN

This is an array of integer registers that specify a numeric gain range for programmable instrumentation amplifiers on channels 1 - 8. The discrete PGA_GAIN range values are as follows:

```
0 = X1
1 = X2
2 = X3
3 = X4
4 = X100
5 = X200
6 = X400
7 = X800
```

PGA_OFFSET

This is an array of IEEE-754 registers that specify a PGA offset voltage to be applied to channels 1 - 8. The range for PGA_OFFSET is -10.0 to +10.0.

DITHER_AMP

This is an array of IEEE-754 registers that determine the level of valve dither applied to PID control channels 1 - 8. The range for DITHER_AMP is 0.0 to 10.0 volts. The dither frequency is 500 Hz.

FBK_POL

This is an array of integer registers that determine the feedback polarity for channels 1 - 8. When FBK_POL = 0, polarity = +1.0. When FBK_POL = 1, polarity = -1.0. The polarity value is multiplied by the incoming feedback voltage before use by the PID control. FBK_POL can be used to compensate for polarity reversal in field wiring.

VLV_POL

This is an array of integer registers that determine the feedback polarity for channels 1 - 8. When VLV_POL = 0, polarity = +1.0. When VLV_POL = 1, polarity = -1.0. The polarity value is multiplied by the output of the PID control block before being sent to the valve D/A. VLV_POL can be used to compensate for polarity reversal in field wiring.

PID_ENABLE

This is an array of integer registers that determine if a PID control channel is enabled on channels 1 - 8. When PID_ENABLE = 1, the PID control is enabled. When PID_ENABLE = 0, the PID control is disabled. PID_ENABLE is used in conjunction with the MSE parameter. PID channels that are disabled drive the servo error to zero by setting the composite command equal to the feedback signal. Valve dither and valve offset are still applied during this state.

VLV_OFFSET

This is an array of IEEE-754 registers that specify the valve offset in volts for PID control channels 1 - 8. The range for VLV_OFFSET is -10.0 to 10.0 volts. VLV_OFFSET can be used to electronically balance valves to zero flow.

TRACE BUFFER VARIABLES

OFFSET	REGISTER NAME	DATA TYPE	R/W	PURPOSE
02E8	TBRECORD	UINT32	R/W	Trace buffer record flag
02EC	TBWPTR	UINT32	R	Trace buffer write pointer
0300	CMD_TRACE	IEEE-754	R	Channel 1 Command trace buffer. 1024 points
1300	FBK_TRACE	IEEE-754	R	Channel 1 Feedback trace buffer. 1024 points
2300	ERR_TRACE	IEEE-754	R	Channel 1 Error trace buffer 1024 points
3300	VLV_TRACE	IEEE-754	R	Channel 1 Valve trace buffer 1024 points
4300	CMD_TRACE	IEEE-754	R	Channel 2 Command trace buffer. 1024 points
5300	FBK_TRACE	IEEE-754	R	Channel 2 Feedback trace buffer. 1024 points
6300	ERR_TRACE	IEEE-754	R	Channel 2 Error trace buffer 1024 points
7300	VLV_TRACE	IEEE-754	R	Channel 2 Valve trace buffer 1024 points
8300	CMD_TRACE	IEEE-754	R	Channel 3 Command trace buffer. 1024 points
9300	FBK_TRACE	IEEE-754	R	Channel 3 Feedback trace buffer. 1024 points
A300	ERR_TRACE	IEEE-754	R	Channel 3 Error trace buffer 1024 points
B300	VLV_TRACE	IEEE-754	R	Channel 3 Valve trace buffer 1024 points

C300	CMD_TRACE	IEEE-754	R	Channel 4 Command trace buffer. 1024 points
D300	FBK_TRACE	IEEE-754	R	Channel 4 Feedback trace buffer. 1024 points
E300	ERR_TRACE	IEEE-754	R	Channel 4 Error trace buffer 1024 points
F300	VLV_TRACE	IEEE-754	R	Channel 4 Valve trace buffer 1024 points
10300	CMD_TRACE	IEEE-754	R	Channel 5 Command trace buffer. 1024 points
11300	FBK_TRACE	IEEE-754	R	Channel 5 Feedback trace buffer. 1024 points
12300	ERR_TRACE	IEEE-754	R	Channel 5 Error trace buffer 1024 points
13300	VLV_TRACE	IEEE-754	R	Channel 5 Valve trace buffer 1024 points
14300	CMD_TRACE	IEEE-754	R	Channel 6 Command trace buffer. 1024 points
15300	FBK_TRACE	IEEE-754	R	Channel 6 Feedback trace buffer. 1024 points
16300	ERR_TRACE	IEEE-754	R	Channel 6 Error trace buffer 1024 points
17300	VLV_TRACE	IEEE-754	R	Channel 6 Valve trace buffer 1024 points
18300	CMD_TRACE	IEEE-754	R	Channel 7 Command trace buffer. 1024 points
19300	FBK_TRACE	IEEE-754	R	Channel 7 Feedback trace buffer. 1024 points
1A300	ERR_TRACE	IEEE-754	R	Channel 7 Error trace buffer 1024 points
1B300	VLV_TRACE	IEEE-754	R	Channel 7 Valve trace buffer 1024 points
1C300	CMD_TRACE	IEEE-754	R	Channel 8 Command trace buffer. 1024 points
1D300	FBK_TRACE	IEEE-754	R	Channel 8 Feedback trace buffer. 1024 points
1E300	ERR_TRACE	IEEE-754	R	Channel 8 Error trace buffer 1024 points
1F300	VLV_TRACE	IEEE-754	R	Channel 8 Valve trace buffer 1024 points

Table 6

TBRECORD

This 32 bit register controls the trace buffer write pointer (TBWPTR). When TBRECORD = 1, data recording starts at the last position of TBWPTR. TBWPTR will increment every .001 seconds. When TBRECORD = 0, TBWPTR does not increment and no new data is stored.

TBWPTR

This 32 bit register is incremented by the VDSP every .001 seconds. The VDSP control loop stores the current values for Command, Feedback, Error and Valve output in separate time history buffers. TBWPTR points to the next location in the trace buffer that will be written by the VDSP. Each trace buffer is 1024 elements in size. TBWPTR wraps around to zero when the end of buffer is reached.

CMD_TRACE

This is an IEEE-754 array of PID command time history. Up to 1024 samples can be stored in this buffer. There is a CMD_TRACE buffer for each channel.

FBK_TRACE

This is an IEEE-754 array of feedback time history. Up to 1024 samples can be stored in this buffer. There is a FBK_TRACE buffer for each channel.

ERR_TRACE

This is an IEEE-754 array of Error time history. Up to 1024 samples can be stored in this buffer.

VLV_TRACE

This is an IEEE-754 array of Valve time history. Up to 1024 samples can be stored in this buffer.

Part Number : 360-0-000-4002

SCPI COMMAND REFERENCE

```
:VALVe
   [:DITHer] . . . . . . . . . . <numeric_value>
    :TRIM . . . . . . . . . . .<numeric_value>
    :INVert . . . . . . . . . .<Boolean>
 :FEEDback
    :INVert . . . . . . . . . .<Boolean>
   [:OFFSet] . . . . . . . . . <numeric_value>
    :GAIN . . . . . . . . . . .G002 | G004 | G008 | G100 |
G200 | G400 | G800 | G001
    :EXCitation . . . . . . . .<numeric_value>
    :CALRelay . . . . . . . . .<Boolean>
 :SERVo
   [:GAIN] . . . . . . . . . .
<numeric_value>,<numeric_value>,<numeric_value>,<numeric_val
ue>,<numeric_value>
    :READ? . . . . . . . . . . CMD | ERR | VLV | FBK
    :ENABle . . . . . . . . . .<Boolean>
 :DIGital
    :INPuts?
    :OUTPuts . . . . . . . . . <numeric_value>
 :ANALog
    :TRACe . . . . . . . . . . <Boolean>
    :TRACe? . . . . . . . . . .<channel_list>,CMD | ERR |
VLV | FBK
 :FPMonitor
    :SOURce . . . . . . . . . .<numeric_value>
 :SETPoint
    :DC
       :LEVel . . . . . . . . .<numeric_value>
       :PERiod . . . . . . . . <numeric_value>
       :LINear . . . . . . . . <Boolean>
    :AC
       :PHASe . . . . . . . . .<numeric_value>
       :AMPLitude . . . . . . .<numeric_value>
       :MSPan . . . . . . . . .<numeric_value>
       :PERiod . . . . . . . . <numeric_value>
    :TRIGger
       :SOURce . . . . . . . . <Boolean>
 :SYSTem
    :ERRor?
    :VERSion?
```

```
:STATus
   :OPERation
     [:EVENt]?
      :CONDition?
      :ENABle . . . . . . . . <numeric_value>
   :QUEStionable
     [:EVENt]?
      :CONDition?
      :ENABle . . . . . . . . <numeric_value>
   :PRESet
```


VDSP DUAL-ACCESS RAM STRUCTURE

```
/* Host channel structure */

typedef struct host_chan {

        /* analog feedback variables */
IEEE_FLOAT  fbk[MAX_CHAN];    /* current value of feedback */
IEEE_FLOAT  cmd[MAX_CHAN];    /* current value composite cmd */
IEEE_FLOAT  err[MAX_CHAN];    /* current error */
IEEE_FLOAT  Vout[MAX_CHAN];   /* current valve output */

        /* Dc, Ac, Freq, and Phase reference controls */
IEEE_FLOAT  dc_off  [MAX_CHAN];    /* DC offset */
IEEE_FLOAT  ac_amp  [MAX_CHAN];    /* AC amplitude */
IEEE_FLOAT  ac_phs  [MAX_CHAN];    /* AC phase */

        /* control loop constants */
IEEE_FLOAT  Kp [MAX_CHAN];      /* Proportional gain constant */
IEEE_FLOAT  Ki [MAX_CHAN];      /* Integral gain constant */
IEEE_FLOAT  Kd [MAX_CHAN];      /* Derivative gain constant */
IEEE_FLOAT  Il [MAX_CHAN];      /* Integration limit */
UINT_32     Ds [MAX_CHAN];      /* Derivative sample period */

        /* calibration control */
UINT_32     cal_relay  [MAX_CHAN]; /* close/open cal relay */
IEEE_FLOAT  excitation [MAX_CHAN]; /* set bridge excitation */
UINT_32     pga_gain   [MAX_CHAN]; /* set PGA gain */
IEEE_FLOAT  pga_offset [MAX_CHAN]; /* set PGA offset */

        /* valve dither control */
IEEE_FLOAT  dither_amp  [MAX_CHAN];    /* sets dither amplitude */

        /* polarity controls */
UINT_32      fbk_pol [MAX_CHAN];        /* FeedBack Polarity */
UINT_32      vlv_pol [MAX_CHAN];        /* Valve Polarity */

       /* servo enable for channel */
UINT_32      pid_enable  [MAX_CHAN];        /* PID channel enable */

IEEE_FLOAT  vlv_offset [MAX_CHAN];  /* static valve offset */

} HCHAN;
```


Part Number : 360-0-000-4002

```
/* Host module structure */

typedef struct host_module {

    UINT_32     avail_chan;     /* Number of available channels */
    IEEE_FLOAT  master_span;    /* span control all channels */
    IEEE_FLOAT  ac_period;      /* period for AC waveforms */
    IEEE_FLOAT  dc_period;      /* period for DC transitions */
    UINT_32     mse;            /* Master SERVO enable */
    UINT_32     fp_chan;        /* front panel channel selection */
    UINT_32     dig_in;         /* Digital Input Image */
    UINT_32     dig_out;        /* Digital Output Image */
    UINT_32     ramp_type;      /* 0 = Haver-Sine, 1 = Linear */

} HMOD;



typedef struct _trace_buff {

    IEEE_FLOAT  cmd[TRACE_SIZE];
    IEEE_FLOAT  fbk[TRACE_SIZE];
    IEEE_FLOAT  err[TRACE_SIZE];
    IEEE_FLOAT  vlv[TRACE_SIZE];

} TRACE;



typedef struct _dport {

    UINT_32     cmd_code;      /* Command code to execute */
    UINT_32     cmd_stat;      /* Command status */
    UINT_32     err_code;      /* Command error code */
    UINT_32     first_chan;    /* First channel affected by command */
    UINT_32     last_chan;     /* Last channel affected by command */
    UINT_32     irq_enable;    /* Interrupt Enable */
    UINT_32     irq_stat;      /* Interrupt Status */
    UINT_32     irq_level;     /* VXI interrupt level to use */
    UINT_32     irq_vect;      /* VXI interrupt vector to use */
    HMOD        hmod;          /* Module variable list */
    HCHAN       hchan;         /* Channel variable list */
    UINT_32     TBRecord;      /* Trace Buffer Record Flag */
    UINT_32     TBWptr;        /* Trace Buffer Write pointer */
    UINT_32     Pad[4];        /* Padding to align TRACEs */
    TRACE       tbuf[MAX_CHAN];    /* 8 channel trace buffer */

} DPORT;
```


Part Number : 360-0-000-4002