PROPRIETARY NOTICE
THIS DOCUMENT AND THE TECHNICAL DATA DISCLOSED ARE PROPRIETARY TO VXI TECHNOLOGY, INC. AND SHALL NOT, WITHOUT EXPRESS WRITTEN PERMISSION OF VXI TECHNOLOGY, INC., BE USED, RELEASED, OR DISCLOSED IN WHOLE OR PART OR USED TO SOLICIT QUOTATIONS FROM A COMPETITIVE SOURCE OR BE USED FOR MANUFACTURE BY ANYONE OTHER THAN VXI TECHNOLOGY, INC. THE INFORMATION HERE ON DISCLOSED HAS BEEN DEVELOPED AT PRIVATE EXPENSE AND MAY ONLY BE USED FOR PURPOSES OF ENGINEERING EVALUATION AND FOR INCORPORATION INTO TECHNICAL DOCUMENTS WHICH SPECIFY PROCUREMENT FROM VXI TECHNOLOGY, INC.

REVISION HISTORY

REV	ECO	DESCRIPTION	DATE	APPROVED
A	6263	Initial Release	12/23/08	F.G.

1. Purpose

1.1 To document the hardware and software interfaces to the VT3625.

2. Scope

2.1 The information in this document applies to the following:

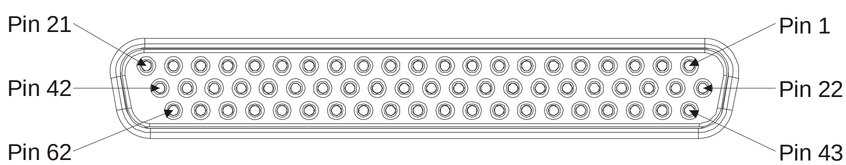
VTI P/N 70-0360-100, FINAL ASSY, VT3625, 16-CHANNEL DAC, 0 to 35 V peak 400 Hz OUTPUT MODULE

(VTI P/N 52-0772-000, ASSY, PCB, VT3625, 16-CHANNEL DAC, 0 to 35 V peak OUTPUT MODULE)

FRONT PANEL CONNECTOR / SIGNALS

The following table details the signals associated with each pin of the VT3625 front panel. Mating connector information and pin location are also provided.

Pin#	Signal	Pin#	Signal	Pin#	Signal
1	GND	**22**	CH1_OP	**43**	GND
2	GND	**23**	CH2_OP	**44**	GND
3	GND	**24**	CH3_OP	**45**	GND
4	GND	**25**	CH4_OP	**46**	GND
5	GND	**26**	CH5_OP	**47**	GND
6	GND	**27**	CH6_OP	**48**	GND
7	GND	**28**	CH7_OP	**49**	GND
8	GND	**29**	CH8_OP	**50**	GND
9	GND	**30**	CH9_OP	**51**	GND
10	GND	**31**	CH10_OP	**52**	GND
11	GND	**32**	CH11_OP	**53**	GND
12	GND	**33**	CH12_OP	**54**	GND
13	GND	**34**	CH13_OP	**55**	GND
14	GND	**35**	CH14_OP	**56**	GND
15	GND	**36**	CH15_OP	**57**	GND
16	Not Used	**37**	CH16_OP	**58**	GND
17	GND	**38**	Not Used	**59**	GND
18	Not Used	**39**	Not Used	**60**	GND
19	GND	**40**	Not Used	**61**	GND
20	Not Used	**41**	Not Used	**62**	GND
21	GND	**42**	Not Used		



(Front view, 90° Counterclockwise Rotation)

P/N: 27-0160-062 (Positronics P/N: ODD62F500T2X)

Mating connector: Positronics P/N: ODD62M210HE0 (62-pin, male, solderable, cable-side connector)

OVERVIEW

The VT3625 is capable of generating sine waves between 0 and 35 V peak in 1.068 mV_{PEAK} increments. To generate a signal from the VT3625, set a channel's output amplitude using the vtvmSmip_setChannelData() function call. Once an output is defined, the channel is enabled using the vtvmSmip_setChannelEnable() call.

Standard Function Calls

vtvmSmip_close

Function Prototype

ViStatus vtvmSmip_close (ViSession **instHndl**);

Function Parameters

instHndl = the instrument handle issued by the resource manager. The driver uses the handle to indicate the target SMIP in calls to the VISA Library.

Description

This API call terminates the session with the specified instrument and de-allocates any system resources that were allocated.

vtvmSmip_GetModelInfo

Function Prototype

vtvmSmip_GetModelInfo (ViSession **instHndl**, ViInt16 **smodNmbr**, ViPInt16 **populated**, ViPInt16 **modelNumber**, ViPInt16 **maxRelays**);

Function Parameters

instHndl = the instrument handle issued by the resource manager. The driver uses the handle to indicate the target VT3625 in calls to the VISA Library.
smodNmbr = a "zero-based" integer that indicates the SMIP carrier slot the VT3625 occupies which is being commanded. Valid input values for the VT3625 = 0.
populated = outputs a Boolean value indicating whether the specified switch module is installed on the SMIP associated with the **instHndl** parameter. Valid return values: 0 (unpopulated) or 1 (populated).
modelNumber = points to storage for the 16-bit model number of the specified switch module.
maxRelays = points to storage for the number of relays installed on the specified switch module. Always returns 16 for the VT3625.

Description

This API outputs a Boolean indicating whether the specified switch module is installed in the SMIP. The function also outputs the model number, and the number of relays installed. If the board is not populated, then the **modelNumber** and **maxRelays** parameters are irrelevant. For the VT3625, the **maxRelays** parameter will return a value of "16".

vtvmSmip_IdQuery

Function Prototype

ViStatus vtvmSmip_IdQuery (ViSession **instHndl**, ViInt16 **smodNmbr**, ViChar_VI_FAR **idStr[]**, ViPInt16 **length**);

Function Parameters

instHndl = the instrument handle issued by the resource manager. The driver uses the handle to indicate the target VT3625 in calls to the VISA Library.
smodNmbr = a "zero-based" integer that indicates the SMIP carrier slot the VT3625 occupies which is being commanded. Valid input values for the VT3625 = 0.
idStr[] = on return, if successful, contains the id string from the specific switch module's Non-Volatile Memory. The **idStr[]** parameter must be at least **vtvmSmip_ID_STRING_SIZE** long.
length = returns the length of the id string. Valid return values: 0 to **vtvmSmip_ID_STRING_SIZE**.

Description

This API outputs the ID string for the module specified in the **smodNmbr** parameter.

vtvmSmip_init

FUNCTION PROTOTYPE

ViStatus vtvmSmip_init (ViRsrc **instrDesc**, ViBoolean **idQuery**, ViBoolean **resetFlag**, ViPSession **instHndl**);

FUNCTION PARAMETERS

instrDesc = must contain a valid descriptor string for the SMIP to be initialized. The string is of the form: VXI::22::INSTR, where “22” is the logical address of the target SMIP.
idQuery = a Boolean indicating whether or not an ID Query is to be performed. Valid input values: 0 (do not perform query) or 1 (perform ID Query).
resetFlag = a Boolean indicating whether or not a Device Reset is to be performed. Valid input values: 0 (do not perform device reset) or 1 (perform device reset).
instHndl = On exit, this 32 bit output parameter contains the session handle to the instrument. The session handle is issued by the resource manager. The driver uses the handle to indicate the target SMIP in calls to the VISA Library.

DESCRIPTION

This API initializes the VT3625 specified in the **instDesc** parameter. It can optionally send a vtvmSmip_IdQuery and a device reset.

vtvmSmip_reset

FUNCTION PROTOTYPE

ViStatus vtvmSmip_reset (ViSession **instHndl**);

FUNCTION PARAMETERS

instHndl = the instrument handle issued by the resource manager. The driver uses the handle to indicate the target VT3625 in calls to the VISA Library.

DESCRIPTION

This API commands the selected VT3625 to place it into a known state. All channels are disabled and set to 0 V and, if defined, the user calibration constants are loaded.

ADDITIONAL FUNCTION CALLS

In addition to the standard SMIP *II* function calls, the VT3625 utilizes seven additional functions. These functions are described in detail below.

vtvmSmip_setChannelEnable

FUNCTION PROTOTYPE

ViStatus vtvmSmip_setChannelEnable(ViSession **instHndl**, ViInt16 **smodNmbr**, ViInt32 **channel**, ViBoolean **enable**);

FUNCTION PARAMETERS

instHndl = the instrument handle issued by the resource manager. The driver uses the handle to indicate the target VT3625 in calls to the VISA Library.
smodNmbr = a "zero-based" integer that indicates the SMIP carrier slot the VT3625 occupies which is being commanded. Valid input values for the VT3625 = 0.
channel = an integer indicating the channel number which will be affected by this function. Valid input values: 1 through 16.
enable = a Boolean value indicating whether the channel indicated by the **channel** parameter will be enabled. Valid input values: 0 (disabled) or 1 (enabled).

DESCRIPTION

This API call enables or disables the channel indicated by the **channel** parameter. When the enable parameter is set to "1", the selected channel will be enabled by this function. If set to "0", the channel will be disabled.

vtvmSmip_setChannelData

FUNCTION PROTOTYPE

ViStatus vtvmSmip_setChannelData(ViSession **instHndl**, ViInt16 **smodNmbr**, ViInt32 **channel**, ViReal64 **data**);

FUNCTION PARAMETERS

instHndl = the instrument handle issued by the resource manager. The driver uses the handle to indicate the target VT3625 in calls to the VISA Library.
smodNmbr = a "zero-based" integer that indicates the SMIP carrier slot the VT3625 occupies which is being commanded. Valid input values for the VT3625 = 0.
channel = an integer indicating the channel number which will be affected by this function. Valid input values: 1 through 16.
data = a value indicating the output to be generated on the **channel**. Valid input values: 0 through 35.

DESCRIPTION

This API call sets the voltage level for the channel indicated by the **channel** parameter. The **data** parameter indicates the channels peak output voltage. For the VT3625, the **data** value must be between 0 and 35, with a resolution of 2 mV, and corresponds to a 0 V peak to 35 V peak output voltage.

vtvmSmip_getChannelEnable

FUNCTION PROTOTYPE

ViStatus vtvmSmip_getChannelEnable(ViSession **instHndl**, ViInt16 **smodNmbr**, ViInt32 **channel**, ViPBoolean **enable**);

FUNCTION PARAMETERS

instHndl = the instrument handle issued by the resource manager. The driver uses the handle to indicate the target VT3625 in calls to the VISA Library.
smodNmbr = a "zero-based" integer that indicates the SMIP carrier slot the VT3625 occupies which is being commanded. Valid input values for the VT3625 = 0.
channel = an integer indicating the channel number which will be affected by this function. Valid input values: 1 through 16.
enable = a Boolean value indicating whether the **channel** is enabled. Valid input values: 0 (disabled) or 1 (enabled).

DESCRIPTION

This API call indicates whether the channel indicated by the **channel** parameter is enabled.

vtvmSmip_getChannelData

FUNCTION PROTOTYPE

ViStatus vtvmSmip_getChannelData(ViSession **instHndl**, ViInt16 **smodNmbr**, ViInt32 **channel**, ViReal64 **data**, ViPBoolean **overload**);

FUNCTION PARAMETERS

instHndl = the instrument handle issued by the resource manager. The driver uses the handle to indicate the target VT3625 in calls to the VISA Library.
smodNmbr = a "zero-based" integer that indicates the SMIP carrier slot the VT3625 occupies which is being commanded. Valid input values for the VT3625 = 0.
channel = an integer indicating the channel number which will be affected by this function. Valid input values: 1 through 16.
data = a value indicating the output currently generated on by the **channel** specified. Valid return values: 0 through 35.
overload = a Boolean value indicating whether the amplifier on the **channel** has been exceeded. Valid input values: 0 (disabled) or 1 (enabled).

DESCRIPTION

This API call returns the data on a specified channel and indicates whether the amplifier limit has been exceeded. The **data** parameter indicates the channels peak output voltage. For the VT3625, the **data** value must be within 0 and 35, with a resolution of 2 mV, and corresponds to a 0 V peak to 35 V peak output voltage. If a "1" is returned for the **overload** parameter, this indicates that the amplifier limit has been exceeded. A "0" returned for this parameter indicates that the operation has occurred within the amplifier's limits.

vtvmSmip_setChannelCalibration

FUNCTION PROTOTYPE

ViStatus vtvmSmip_setChannelCalibration(ViSession **instHndl**, ViInt16 **smodNmbr**, ViInt32 **channel**, ViReal64 **gain**, ViReal64 **offset**);

FUNCTION PARAMETERS

instHndl = the instrument handle issued by the resource manager. The driver uses the handle to indicate the target VT3625 in calls to the VISA Library.
smodNmbr = a "zero-based" integer that indicates the SMIP carrier slot the VT3625 occupies which is being commanded. Valid input values for the VT3625 = 0.
channel = an integer indicating the channel number which will be affected by this function. Valid input values: 1 through 16.
gain = a value that indicates the gain calibration coefficient. Typical value: 0.96.
offset = an value that indicates the offset calibration coefficient. Valid values are from 0 through 65,535.

DESCRIPTION

This API call sets the calibration gain and offset values for the channel indicated by the **channel** parameter. For more information, please refer to the *Calibration Procedure* section.

vtvmSmip_getChannelCalibration

FUNCTION PROTOTYPE

ViStatus vtvmSmip_getChannelCalibration(ViSession **instHndl**, ViInt16 **smodNmbr**, ViInt32 **channel**, ViReal64 **gain**, ViReal64 **offset**);

FUNCTION PARAMETERS

instHndl = the instrument handle issued by the resource manager. The driver uses the handle to indicate the target VT3625 in calls to the VISA Library.
smodNmbr = a "zero-based" integer that indicates the SMIP carrier slot the VT3625 occupies which is being commanded. Valid input values for the VT3625 = 0.
channel = an integer indicating the channel number which will be affected by this function. Valid input values: 1 through 16.
gain = a value that indicates the gain calibration coefficient. Typical value: 0.96.
offset = an value that indicates the offset calibration coefficient. Valid values are from 0 through 65,535.

DESCRIPTION

This API call returns the calibration gain and offset values for the channel indicated by the **channel** parameter. For more information, please refer to the *Calibration Procedure* section.

vtvmSmip_saveCalibration

FUNCTION PROTOTYPE

ViStatus vtvmSmip_saveCalibration(ViSession **instHndl**, ViInt16 **smodNmbr**, ViInt32 **type**, ViInt32 **password**);

FUNCTION PARAMETERS

instHndl = the instrument handle issued by the resource manager. The driver uses the handle to indicate the target VT3625 in calls to the VISA Library.
smodNmbr = a "zero-based" integer that indicates the SMIP carrier slot the VT3625 occupies which is being commanded. Valid input values for the VT3625 = 0.
type = an integer indicating the type of calibration which will be performed. Valid input values: factory or user.
password = an integer which defines the password required to initiate calibration. Valid input values: 0x0 through 0xFFFFFFFF.

DESCRIPTION

This API call saves to memory the calibration coefficients for all channels. The **type** parameter has the following acceptable values:

Decimal Value	Hex Value	#define	type Description
1	0x01	vtvmSmip_FACTORY	Factory calibration
2	0x02	vtvmSmip_USER	User calibration

The only function of the **password** is to protect factory calibration (**type** = 0x01). Although a valid value must be provided when user calibration is performed, this value is ignored and does not password protect the calibration data. Similarly, if a password is used for user calibration, the same password does not have to be provided when performing a subsequent user calibration. The new calibration constants will simply overwrite the existing constants. Note that when factory calibration is both factory and user calibration constants are set.

vtvmSmip_loadCalibration

FUNCTION PROTOTYPE

ViStatus vtvmSmip_loadCalibration(ViSession **instHndl**, ViInt16 **smodNmbr**, ViInt32 **type**);

FUNCTION PARAMETERS

instHndl = the instrument handle issued by the resource manager. The driver uses the handle to indicate the target VT3625 in calls to the VISA Library.
smodNmbr = a "zero-based" integer that indicates the SMIP carrier slot the VT3625 occupies which is being commanded. Valid input values for the VT3625 = 0.
type = an integer indicating the type of calibration which will be performed. Valid input values: factory or user.

DESCRIPTION

This API call loads from memory the calibration coefficients for all channels. The **type** parameter has the following acceptable values:

Decimal Value	Hex Value	#define	type Description
0	0x00	vtvmSmip_DEFAULT	Default calibration
1	0x01	vtvmSmip_FACTORY	Factory calibration
2	0x02	vtvmSmip_USER	User calibration

For more information, please refer to the *Calibration Procedure* section.

CALIBRATION PROCEDURE

3. Introduction

3.1 Overview

The VT3625 is a 16-channel, 16-bit digital-to-analog converter (DAC) configured as a sine wave generator with a fixed output frequency of 400 Hz. The output amplitude is adjustable from 0 to 35V peak.

The VT3625 is a 'register based' VXI instrument which is interfaced to the VXI backplane through a VXI Technology, Inc. (VTI) Switch Module Interface Platform (SMIP). The VT3625 is supported by the SMIP VXI *plug&play* driver (VTI P/N: 72-0029-012, revision 4.30 or later). The driver supports setting or getting each channel's output amplitude, output enable state, and calibration factors. The driver also supports writing the calibration factors to non-volatile memory on the VT3625 and loading the calibration factors from non-volatile memory to the calibration factor registers. In normal operation, the calibration factors are loaded from non-volatile memory to the calibration factor registers following a power on event.

At the hardware level, the VT3625's output amplitude is set by a 16-bit value. This yields a resolution of 1.174 mV per count (77 V / 65,535 counts). To ensure accuracy, each channel is provided with a 32-bit floating point gain control register and a 16-bit integer offset control register. The calibration process determines the values of the gain and offset factors.

Automating the calibration of the VT3625 is accomplished with a 16-channel, 2-wire multiplexer and a precision DMM that supports computer control.

3.2 Audience

This document is intended for technical staff who need to understand the process by which the VT3625 is calibrated.

NOTE Calibration should only be performed by qualified personnel. Performing calibration improperly can result in incorrect readings and may place the instrument in an unknown state.

4. Terminology

4.1 Factory Calibration

Factory calibration refers to the full calibration process that is conducted prior to unit shipment. In the factory calibration process, each channel's gain and offset factors are determined and stored in nonvolatile memory.

4.2 User Calibration

User calibration refers to a second set of calibration factors that are user programmable. The process of determining the value of the user calibration factors is the same as for factory calibration.

4.3 Default Calibration

Default calibration factors are used by the VT3625 when no other calibration data has been programmed. The default gain and offset values can be read from the instrument by first loading the default calibration factors using the vtvmSmip_loadCalibration() function, then querying the values using the vtvmSmip_getChannelCalibration() function.

5. Required Resources

5.1 Automated calibration of the VT3625 requires the following equipment and software:

PC Compatible Computer
VXI Chassis and Slot 0 Controller
16 channel multiplexer (VM8016 or equivalent)
Precision DMM (VM2710A or equivalent)
Test Cable to connect DUT through multiplexer to DMM
SMIP *plug&play* Driver (VTI P/N 72-0029-012)

6. Setup

6.1 Install the SMIP *plug&play* driver on the PC.

6.2 Install the Slot 0 controller, DUT, and any other VXI Instruments used (e.g. VM2710A, VM8016, etc) in the VXI chassis. Connect the VXI chassis to the PC using the appropriate interface cable (Ethernet, FireWire, etc.).

6.3 Connect the DUT through the multiplexer to the DMM.

7. Calibration

7.1 Overview

7.1.1 The VT3625 is calibrated using a two-point process. That is, the DUT is commanded to output a voltage near the bottom of its range and the actual output amplitude is measured and recorded. Then the DUT is commanded to output a voltage near the top of its range and, again, the actual output amplitude is measured and recorded. From these two measurements, the gain factor is computed and sent to the DUT. Finally, the DUT is commanded to output 0 V and the dc voltage is measured. From this measurement and the gain factor, the offset factor is computed.

7.2 Procedure

7.2.1 Power up the PC and the test equipment.

7.2.2 Connect the DUT channel to be calibrated (test channel) to the DMM.

7.2.3 Calibrate the test channel:

7.2.3.1 Command the DUT to enable the test channel.
vtvmSmip_setChannelEnable()

7.2.3.2 Set the gain factor to 1.0 and the offset factor to 32768.
vtvmSmip_setChannelCalibration()

7.2.3.3 Command the DUT to output 1.0 V_{PEAK}.
vtvmSmip_setChannelData()

7.2.3.4 Read the computed actual output setting from the DUT and check the test channel for an overload condition. Record the output setting as V_{SET1}.
vtvmSmip_getChannelData()

7.2.3.5 Measure the actual output using the DMM. The reading will most likely be in volts root-mean-squared (rms). Convert the V_{RMS} reading to V_{PEAK} using the following formula:

$$V_{PEAK} = V_{RMS} * \sqrt{2}$$

Record the reading as V_{RDG1}.

7.2.3.6 Command the DUT to output 28.0 V_{PEAK}
vtvmSmip_setChannelData()

7.2.3.7 Read the computed actual output setting from the DUT and check the test channel for an overload condition. Record the output setting as V_{SET2}.
vtvmSmip_getChannelData()

7.2.3.8 Measure the actual output using the DMM. Convert the V_{RMS} reading to V_{PEAK}. Record the reading as V_{RDG2}.

7.2.3.9 Compute the gain factor from the following equation:

$$gain = \frac{(V_{SET2} - V_{SET1})}{(V_{RDG2} - V_{RDG1})}$$

7.2.3.10 Update the DUT's calibration factors using the gain calculated in the previous step and 32768 for the offset factor.
vtvmSmip_setChannelCalibration()

7.2.3.11 Command the DUT to output 0.0 V_{PEAK}
vtvmSmip_setChannelData()

7.2.3.12 Read the computed actual output setting from the DUT and check the test channel for an overload condition. Record the output setting as V_{SET3}.
vtvmSmip_getChannelData()

7.2.3.13 Measure the actual DCV output using the DMM. Record the reading as V_{RDG3}.

7.2.3.14 Compute the offset factor from the following equation:

$$offset = 32768 - (gain * V_{RDG3} / resolution)$$

where $resolution = 77 / 65535$

7.2.3.15 Update the DUT's calibration factors using the gain calculated above and the offset calculated in the previous step.
vtvmSmip_setChannelCalibration()

7.2.3.16 Save the new user calibration data to non-volatile memory.
vtvmSmip_saveCalibration()
For the calibration type parameter, pass vtvmSmip_DAC_USER_CALIBRATION.
For the password parameter, pass 0 (zero).

7.2.4 After all channels have been adjusted and the new calibration factors have been saved, it is a good idea to verify that the outputs all meet the specifications by testing the accuracy of two or more set points per channel.

VT3625 SPECIFICATIONS

GENERAL SPECIFICATIONS	
NUMBER OF CHANNELS	
	16
VOLTAGE MODES	
Output range	0 to 35 V peak
Resolution	1.174 mV
Output current	< 25 mA maximum per channel short circuit
Output frequency	400 Hz
Short circuit	Continuous duration
Resolution	16 bits
SETTLING TIME	
	< 15 µs to 0.1% of specified value
ACCURACY SPECIFICATIONS	
Output	±(1.0% of setting + 2 mV)
Temperature coefficient	±(0.005% of setting + 400 µV) / ºC
Total harmonic distortion	< 0.5%
Frequency	< ±0.1 Hz
POWER REQUIREMENTS	
	+5.0 V @ 1 A, +12 V @ 0.20 A, -12 V @ 0.20 A, +24.0 V @ 1.0 A,-24 V @ 1.0 A