DM482e

Programming Manual

Version 1.2, 08-2013

Table of Contents

DM482e_DPINOpen

ViStatus DM482e_DPINOpen (resourceName, dpingroup_sel, init_options, optionString, vi)

ViRsrc resourceName (in)

> Specifies the resourceName assigned by PXI. For example, "PXI36::0::INSTR" is the resourceName of an instrument. resourceName can also be a logical IVI name.

ViInt32 dpingroup_sel (in)

> Specifies the pin group to be turned on or activated.
> 1 = DPin group 0 (PIN0 to PIN5) and trigout0
> 2 = DPin group 1 (PIN6 to PIN11) and trigout1
> 3 = DPin group 0 and 1(PIN0 to PIN11), trigout0 and trigout1

ViInt32 init_options (in)

> To open a session and leave the device in its existing configuration, set this argument to 0.
> To place the device in a known start-up state when creating a new session set the appropriate bit value:
> bit[0] = Specifies whether to reset the device during the initialization procedure (enabled) or maintain current configuration in the hardware (disabled).
> bit[1] = Specifies whether to reset the cmd fifo during the initialization procedure.
> bit[2] = Specifies whether to reset the result fifo during the initialization procedure.
> bit[3] = Specifies whether to reset the lock status to unlock during the initialization procedure.
> bit[5] = To bypass calibration data loading process.

ViConstString optionString (in)

> You can optionally set the initial state of the following session attributes, separated by semicolons. Otherwise pass in an empty string ("").

a. "Simulate = 0" to open session as online mode OR "Simulate = 1" to open as offline mode. This option string can place the instrument in simulation mode.

b. "DriverSetup=Model:<model number>"

Example: "Simulate = 0, DriverSetup=Model:DM428e"

ViSession* vi (out)

Returns an instrument handle that you can use to identify the instrument in all subsequent function calls.

Descriptions

DPINOpen creates a new VISA session to the instrument specified in the resourceName, which provides methods to control and interact with the instrument, and returns a session handle you use to identify the session in all subsequent function calls.

This VI also allows you to configure the instrument into known states upon initialization via the init_options:

1. Set bit[0] to "1" to reset the instrument during instrument initialization.

2. Bits[1-2] are meant to clear the command FIFO[1] and result FIFO during initialization. Command FIFO is used to store any data written from the PXI/PXIe bus from the backplane, keeping the instructions in proper order for execution. Result FIFO is used to keep the data to be written to the PXI/PXIe bus back to the host computer. The status of Result FIFO will be read before the data is retrieved back to the host.

This function returns zero if successful and non-zero if otherwise.

---

[1] FIFO is an acronym for **First In, First Out,** which is an abstraction related to ways of organizing and manipulation of data relative to time and prioritization. This expression describes the principle of a queue processing technique or servicing conflicting demands by ordering process by first-come, first-served (FCFS) behavior: where the persons leave the queue in the order they arrive, or waiting one's turn at a traffic control signal. FIFO blocks are designed in the processor of the instrument to manage data write and read to avoid traffic jam.

DM482e_DPINClose

ViStatus DM482e_DPINClose (vi)

ViSession vi (in)
>   Specifies the instrument handle.

DM482e_DPINClose closes the session specified in instrument handle. The instrument will maintain its last running state.

This function returns zero if successful and non-zero if otherwise.

DM482e_DPINForce

ViStatus DM482e_DPINForce (vi, pinNo, state)

ViSession vi (in)
>   Specifies the instrument handle.

ViInt32 pinNo (in)
>   Specify the pin number.
>   If dpingroup_sel = 1 during DM482e_DPINOpen, pinNo=0-5 (PIN0 to PIN5).
>   If dpingroup_sel = 2 during DM482e_DPINOpen, pinNo=6-11 (PIN6 to PIN11).
>   If dpingroup_sel = 3 during DM482e_DPINOpen, pinNo=0-11 (PIN0 to PIN11).

ViInt32 state (in)
>   Specifies the operation mode:

State	Description
0	CONST_FORCE_STATE_VECTOR
1	CONST_FORCE_STATE_PMU
2	CONST_FORCE_STATE_DIO
5	CONST_FORCE_STATE_CLOCK
6	CONST_FORCE_STATE_INVERTED_CLOCK

Table 1: Mode of Operation

DM482e_DPINForce specifies the operation mode. Definition of each mode is explained below.

Mode	Description
CONST_FORCE_STATE_VECTOR	Run a series of digital patterns
CONST_FORCE_STATE_PMU	Perform parametric measurement unit functions

CONST_FORCE_STATE_DIO	Perform general digital I/O functions such as Drive Pin, Driver Port, Read Pin and Read Port.
CONST_FORCE_STATE_CLOCK	Configure the selected pin to drive continuous clock signal
CONST_FORCE_STATE_INVERTED_CLOCK	Configure the selected pin to drive continuous clock signal, whose state is inversed of the pin configured in CONST_FORCE_STATE_CLOCK mode.

Table 2: Description of Mode

This function returns zero if successful and non-zero if otherwise.

DM482e_ReadRevision

ViStatus DM482e_ReadRevision (vi, instrumentDriverRevision, firmwareRevision)

Arguments

ViSession vi (in)

> Specifies the instrument handle.

ViChar* instrumentDriverRevision (out)

> Returns the driver revision information of DM482e.dll. Space allocated by the caller, must be at least 32 bytes.

ViChar* firmwareRevision (out)

> Space allocated by the caller, must be at least 32 bytes, returns firmware revision information for the device you are using. This argument returns both firmware version of master and daughter card separated by a hyphen.

Descriptions

DM482e_ReadRevision returns the driver and firmware revision of the DM482e.

This function returns zero if successful and non-zero if otherwise.

DM482e_ReadChannelTemperature

ViStatus DM482e_ReadChannelTemperature (vi, pinNo, temperature)

ViSession vi (in)

>   Specifies the instrument handle.

ViInt32 pinNo (in)

>   Specify the pin number.
>   If dpingroup_sel = 1 during DM482e_DPINOpen, pinNo=0-5.
>   If dpingroup_sel = 2 during DM482e_DPINOpen, pinNo=6-11.
>   If dpingroup_sel = 3 during DM482e_DPINOpen, pinNo=0-11.

ViReal64* temperature (out)

>   Read the temperature of the on-board Pin Electronics IC. Temperature is in degree Celsius.

DM482e_ReadChannelTemperature reads the channel temperature of the on-board Pin Electronics IC.

This function returns zero if successful and non-zero if otherwise.

DM482e_ReadAmbientTemperature

ViStatus DM482e_ReadAmbientTemperature (vi, temperature)

ViSession vi (in)

>Specifies the instrument handle.

ViReal64* temperature (out)

>Return the current temperature on the board, in degrees Celsius. This temperature sensor is placed on bottom plane of master card.

DM482e_ReadAmbientTemperature returns the current temperature of the board. Refer to DM482e_ReadChannelTemperature for more precise temperature of each pin.

This function returns zero if successful and non-zero if otherwise.

DM482e_ReadSerialNumber

ViStatus DM482e_ReadSerialNumber (vi, sn)

Arguments

ViSession vi (in)

Specifies the instrument handle.

ViChar* sn (out)

Serial number length should at least 64 bytes. It returns both master and daughter card serial numbers separated by a hyphen.

Descriptions

DM482e_ReadSerialNumber returns the serial number of the device. Space allocated by the caller, and must be at least 64 characters.

This function returns zero if successful and non-zero if otherwise.

DM482e_Reset

ViStatus DM482e_Reset (vi)

ViSession vi (in)

> Specifies the instrument handle.

DM482e_Reset resets all session attributes to default value on hardware and turn off all pins.

This function returns zero if successful and non-zero if otherwise.

DM482e_ResetGroup

Synopsis

ViStatus DM482e_ResetGroup (vi, group)

Arguments

ViSession vi (in)
>   Specifies the instrument handle.

ViInt32 group (in)
>   Specifies the pin group to be reset.

Descriptions

DM482e_ResetGroup resets only selected pin group to default value on hardware and turn off all pins in the group.

This function returns zero if successful and non-zero if otherwise.

DM482e_ConfigureMultiSiteMode

Synopsis

ViStatus DM482e_ConfigureMultiSiteMode (vi, mode)

Arguments

ViSession vi (in)

Specifies the instrument handle.

ViInt32 mode (in)

Mode = 0 for single site operation. Lock/unlock operation is not performed. This is the default mode.

Mode = 1 for multi-site operation. Lock/unlock operation is performed.

Descriptions

DM482e_ConfigureMultiSiteMode configures DM482e to operate in multi-site or single-site mode.

This function returns zero if successful and non-zero if otherwise.

ReadSlotAddress

ViStatus ReadSlotAddress (vi, slotAddress)

ViSession vi (in)
>   Specifies the instrument handle.

Vint32* slotAddress (out)
>   Return the slot address of the DM482e.

ReadSlotAddress returns the slot number of the module where it is slotted.

This function returns zero if successful and non-zero if otherwise.

Section 2: DIO

DM482e_DrivePort

Synopsis

ViStatus DM482e_DrivePort (vi, value)

Arguments

ViSession vi (in)
> Specifies the instrument handle.

Vint32 value (in)
> Specifies the port value to be driven.
> If dpingroup_sel = 1 during DM482e_DPINOpen, port covers PIN0-5 only. LSB refers to pin 0.
> If dpingroup_sel = 1 during DM482e_DPINOpen, port covers PIN6-11 only. LSB refers to pin 6.
> If dpingroup_sel = 3 during DM482e_DPINOpen, port covers PIN0-11. LSB refers to pin 0.

Descriptions

DM482e_DrivePort drives port value to the specified pin group.

This function returns zero if successful and non-zero if otherwise.

DM482e_DrivePin

Synopsis

ViStatus DM482e_DrivePin (vi, pinNo, value)

Arguments

ViSession vi (in)
        Specifies the instrument handle.

Vint32 pinNo (in)
        If dpingroup_sel = 1 during DM482e_DPINOpen, pinNo refers to pin 0 to pin 5.
        If dpingroup_sel = 2 during DM482e_DPINOpen, pinNo refers to pin 6 to pin 11.
        If dpingroup_sel = 3 during DM482e_DPINOpen, pinNo refers to pin 0 to pin 11.

Vint32 value (in)
        Specifies the drive value of the selected pin.

Descriptions

DM482e_DrivePin drives logic 0 or 1 to the specified pin number.

This function returns zero if successful and non-zero if otherwise.

DM482e_SetPortDirection

ViStatus DM482e_SetPortDirection (vi, value)

ViSession vi (in)

Specifies the instrument handle.

Vint32 value (in)

Specifies the direction of the targeted port where each bit value represents one port.
If dpingroup_sel = 1 during DM482e_DPINOpen, port covers pinNo 0-5 only. LSB refers to pin 0 value.
If dpingroup_sel = 2 during DM482e_DPINOpen, port covers pinNo 6-11 only. LSB refers to pin 6 value.
If dpingroup_sel = 3 during DM482e_DPINOpen, port covers pinNo 0-11. LSB refers to pin 0 value.
Bit 0 means input, bit 1 means output.

DM482e_SetPortDirection sets the direction of the selected pin group.

This function returns zero if successful and non-zero if otherwise.

DM482e_SetPinDirection

ViStatus DM482e_SetPinDirection (vi, pinNo, value)

Arguments

ViSession vi (in)
    Specifies the instrument handle.

Vint32 pinNo (in)
    Specifies the pin to be acted upon.
    If dpingroup_sel = 1 during DM482e_DPINOpen, pinNo is 0-5 only.
    If dpingroup_sel = 2 during DM482e_DPINOpen, pinNo is 6-11 only.
    If dpingroup_sel = 3 during DM482e_DPINOpen, pinNo is 0-11.

Vint32 value (in)
    Specifies the direction of the targeted pin.

Descriptions

DM482e_SetPinDirection sets the direction of the selected pin.

This function returns zero if successful and non-zero if otherwise.

DM482e_ReadPort

ViStatus DM482e_ReadPort (vi, value)

ViSession vi (in)

>Specifies the instrument handle.

Vint32* value (out)

>Returns the read back port value.
>
>If dpingroup_sel = 1 during DM482e_DPINOpen, port covers pinNo is 0-5 only. LSB refers to pin 0 value.
>
>If dpingroup_sel = 2 during DM482e_DPINOpen, port covers pinNo is 6-11 only. LSB refers to pin 6 value.
>
>If dpingroup_sel = 3 during DM482e_DPINOpen, port covers pinNo is 0-11. LSB refers to pin 0 value.

DM482e_ReadPort reads back the value present at the specified port. 2 binary bits are returned for each pin, which represents:

Binary Bits	Description
00	Logic low (L)
01	Logic high (H)
10	Undefined state (Z)

Table 3: Binary Representation of Logic State

For example, if dpingroup_sel = 3, and the returned value is 0x5505, the binary representation is "10 10 01 01 01 01 00 00 01 01". Every 2 binary bits represent the logic state of a pin. In this example, the state of each pin is "LLZZHHHHLLHH".

This function returns zero if successful and non-zero if otherwise.

DM482e_ReadPin

ViStatus DM482e_ReadPin (vi, pinNo, value)

ViSession vi (in)
>   Specifies the instrument handle.

Vint32 pinNo (in)
>   Specifies the pin to be acted upon.
>   If dpingroup_sel = 1 during DM482e_DPINOpen, pinNo is 0-5 only.
>   If dpingroup_sel = 2 during DM482e_DPINOpen, pinNo is 6-11 only.
>   If dpingroup_sel = 3 during DM482e_DPINOpen, pinNo is 0-11.

Vint32* value (out)
>   Returns the read back pin value.

DM482e_ReadPin reads back the value present at the specified pin. 2 binary bits are returned, which represents:

Binary Bits	Description
00	Logic low (L)
01	Logic high (H)
10	Undefined state (Z)

Table 4: Binary Representation of Logic State

This function returns zero if successful and non-zero if otherwise.

Section 3: Pin Electronics

DM482e_DPINVectorResourceAllocation

ViStatus DM482e_DPINVectorResourceAllocation (vi, vecSetCount, resourceArray)

Arguments

ViSession vi (in)
> Specifies the instrument handle.

ViInt32 vecSetCount (in)
> The amount of vector set to be used.

ViInt32 * resourceArray (in)
> Specifies the required memory depth of each vector set. This has to be multiple of 512k.
> For example, if resourceArray[0]=6, resourceArray[1]=2, resourceArray[2]=3, it means
> that vector set 0 requires 6*512k of memory, vector set 1 requires 2*512k of memory,
> and vector set 2 requires 3*512k of memory.
> By default, vecSetCount is 32 and resourceArray is 1 for each vector set.

Descriptions

DM482e_DPINVectorResourceAllocation specifies the resource required to be allocated in the hardware. This function has to be called before vectors are loaded. DM482e_Reset will not reset vecSetCount and resourceArray on the hardware.

This function returns zero if successful and non-zero if otherwise.

DM482e_DPINLevel

ViStatus DM482e_DPINLevel (vi, pinNo, VIH, VIL, VOH, VOL, IOH, IOL, VCH, VCL, VTERM)

Arguments

ViSession vi (in)
> Specifies the instrument handle.

ViInt32 pinNo (in)
> If dpingroup_sel = 1 during DM482e_DPINOpen, pinNo refers to pin 0 to pin 5.
> If dpingroup_sel = 2 during DM482e_DPINOpen, pinNo refers to pin 6 to pin 11.
> If dpingroup_sel = 3 during DM482e_DPINOpen, pinNo refers to pin 0 to pin 11.

ViReal64 VIH (in)
> Specifies the output driver high voltage level from the pin.

ViReal64 VIL (in)
> Specifies the output driver low voltage level from the pin.

ViReal64 VOH (in)
> Specifies the input comparator high voltage level into the pin.

ViReal64 VOL (in)
> Specifies the input comparator low voltage level into the pin.

ViReal64 IOH (in)
> Specifies the sink current when active load is enabled.

ViReal64 IOL (in)
> Specifies the source current when active load is enabled.

ViReal64 VCH (in)
> Specifies the reflection voltage clamp level for high range.

ViReal64 VCL (in)

> Specifies the reflection voltage clamp level for low range.

ViReal64 VTERM (in)

> Specifies the termination voltage when input termination is enabled. If input termination is disabled, driver is in high impedance.

Descriptions

DM482e_DPINLevel programs the level setting for the specified pin.

This function returns zero if successful and non-zero if otherwise.

DM482e_DPINVecLoad

ViStatus DM482e_DPINVecLoad (vi, option, vecSetNo, vecFileName)

ViSession vi (in)

Specifies the instrument handle.

ViInt32 option (in)

Specifies the clock frequency rate based for following options:

0 : Supprots bidirectional IO but clock frequency up to 200 Mbps.

1 : Supports dedicated IO but clock frequency up to 300 Mbps.

ViInt32 vecSetNo (in)

Specifies the vector set to be loaded.

char* vecFileName (in)

Specifies file name of the vector file to be loaded.

DM482e_DPINVecLoad loads the vectors to the device specified in the vector file.

This function returns zero if successful and non-zero if otherwise.

DM482e_DPINPeriod

Synopsis

ViStatus DM482e_DPINPeriod (vi, timingSetNo, period_s)

Arguments

ViSession vi (in)
Specifies the instrument handle.

ViInt32 timingSetNo
Specifies the timing set to be used.

ViReal64 period_s (in)
Specifies the period of each bit of data as pointed by the specified timing set.

Descriptions

DM482e_DPINPeriod specifies the timing set to be used.

This function returns zero if successful and non-zero if otherwise.

DM482e_DPINOn

Synopsis

ViStatus DM482e_DPINOn (vi, pinNo)

Arguments

ViSession vi (in)
>Specifies the instrument handle.

ViInt32 pinNo (in)
>Specifies the pin to be acted upon.
>If dpingroup_sel = 1 during DM482e_DPINOpen, pinNo is 0-5 only.
>If dpingroup_sel = 2 during DM482e_DPINOpen, pinNo is 6-11 only.
>If dpingroup_sel = 3 during DM482e_DPINOpen, pinNo is 0-11.

Descriptions

DM482e_DPINOn enables the specified pin. This function needs to be called before mode of operation (DM482e_DPINForce) can be effective.

This function returns zero if successful and non-zero if otherwise.

DM482e_DPINOff

ViStatus DM482e_DPINOff (vi, pinNo)

ViSession vi (in)

Specifies the instrument handle.

ViInt32 pinNo (in)

Specifies the pin to be acted upon.
If dpingroup_sel = 1 during DM482e_DPINOpen, pinNo is 0-5 only.
If dpingroup_sel = 2 during DM482e_DPINOpen, pinNo is 6-11 only.
If dpingroup_sel = 3 during DM482e_DPINOpen, pinNo is 0-11.

DM482e_DPINOff disables the specified pin.

This function returns zero if successful and non-zero if otherwise.

DM482e_DPINHVOn

ViStatus DM482e_DPINHVOn (vi, pinNo)

ViSession vi (in)
> Specifies the instrument handle.

ViInt32 pinNo (in)
> Specifies the pin to be acted upon.
> If dpingroup_sel = 1 during DM482e_DPINOpen, pinNo is 0-5 only.
> If dpingroup_sel = 2 during DM482e_DPINOpen, pinNo is 6-11 only.
> If dpingroup_sel = 3 during DM482e_DPINOpen, pinNo is 0-11.

DM482e_DPINHVOn enables the high voltage (HV) output relay of the specified pin. Only pins 0, 2, 4, 6, 8, and 10 contain high voltage capabilities.

This function returns zero if successful and non-zero if otherwise.

DM482e_DPINHVOff

ViStatus DM482e_DPINHVOff (vi, pinNo)

ViSession vi (in)
> Specifies the instrument handle.

ViInt32 pinNo (in)
> Specifies the pin to be acted upon.
> If dpingroup_sel = 1 during DM482e_DPINOpen, pinNo is 0-5 only.
> If dpingroup_sel = 2 during DM482e_DPINOpen, pinNo is 6-11 only.
> If dpingroup_sel = 3 during DM482e_DPINOpen, pinNo is 0-11.

DM482e_DPINHVOff disables the high voltage (HV) output relay of the specified pin. Only pins 0, 2, 4, 6, 8, and 10 contain high voltage capabilities.

This function returns zero if successful and non-zero if otherwise.

DM482e_RunVector

Synopsis

ViStatus DM482e_RunVector (vi, vecSetNo)

Arguments

ViSession vi (in)
> Specifies the instrument handle.

ViInt32 vecSetNo (in)
> Specifies the vector set to be executed.

Descriptions

DM482e_RunVector execute the specified vector set.

This function returns zero if successful and non-zero if otherwise.

DM482e_AcquireVecEngineStatus

ViStatus DM482e_AcquireVecEngineStatus (vi, status)

ViSession vi (in)
> Specifies the instrument handle.

ViInt32* status (out)
> Return the status of the vector execution.
> 0: Completed.
> 1: Busy.

DM482e_AcquireVecEngineStatus return the status of the vector execution.

This function returns zero if successful and non-zero if otherwise.

DM482e_ReadHistoryRam

Synopsis

ViStatus DM482e_ReadHistoryRam (vi, vectorCount, startVectorLocation, vecSetNo, history_ram_data)

Arguments

ViSession vi (in)
> Specifies the instrument handle.

ViInt32 vectorCount (in)
> Specifies the number of vectors to be returned from the device.

ViInt32 startVectorLocation (in)
> Specifies the starting location of the vectors to be returned from the device.

ViInt32 ViInt32 vecSetNo (in)
> Specifies the vector set of interest.

ViUInt32* history_ram_data(out)
> Returns an array of measurements. User has to allocate enough memory for history_ram_data.

Descriptions

DM482e_ReadHistoryRam reads back every stored bit in the device. One memory space of history_ram_data contains information of one vector for all the pins. The format of the returned history_ram_data is as follows:

Binary Bits	Description
00	Logic low (L)
01	Logic high (H)
10	Undefined state (Z)

Table 5: Binary Representation of Logic State

For example, if dpingroup_sel = 3, and history_ram_data returns 0x5505, the binary representation is "10 10 01 01 01 01 00 00 01 01". Every 2 binary bits represent the logic state of a pin. In this example, the state of each pin is "LLZZHHHHLLHH".

This function returns zero if successful and non-zero if otherwise.

DM482e_ConfigurePEAttribute

ViStatus DM482e_ConfigurePEAttribute (vi, pinNo, inputTerminationEnable, HVEnable, activeLoadEnable, differentialComparatorEnable)

Arguments

ViSession vi (in)
    Specifies the instrument handle.

ViInt32 pinNo (in)
    Specifies the pin to be acted upon.
    If dpingroup_sel = 1 during DM482e_DPINOpen, pinNo is 0-5 only.
    If dpingroup_sel = 2 during DM482e_DPINOpen, pinNo is 6-11 only.
    If dpingroup_sel = 3 during DM482e_DPINOpen, pinNo is 0-11.

ViBoolean inputTerminationEnable (in)
    Specifies whether to enable input termination.
    If input termination is disabled, the specified pin is in high-Z.
    If input termination is enabled, the specified pin drives VT voltage level.

ViBoolean HVEnable (in)
    Specifies whether to enable high voltage capability.
    When HVEnable is enabled, VIH is twice of VT level while VIL maintains.

ViBoolean activeLoadEnable (in)
    Specifies whether to enable active load capability.

ViBoolean differentialComparatorEnable (in)
    Specifies whether to enable differential comparator capability.
    When differential comparator is enabled, odd pins (1, 3, 5, 7, 9 and 11) will be disabled.
    The voltage difference between pin 0 and pin 1 will be feed to the comparator of pin 0 only. Same goes to other pair of differential channels.

DM482e_ConfigurePEAttribute programs the attribute of pin electronics of the specified pin.

This function returns zero if successful and non-zero if otherwise.

DM482e_ConfigureVectorEngineAttribute

ViStatus DM482e_ConfigureVectorEngineAttribute (vi, triggerEnable, continuousEnable)

ViSession vi (in)

> Specifies the instrument handle.

ViBoolean triggerEnable (in)

> Vector Engine only starts to drive vector when an external trigger is received. External trigger can be configured using triggers API.

ViBoolean continuousEnable(in)

> Vector Engine continuously repeats the vector until reset is called or argument continuousEnable is disabled. Vector file has to contain at least 5k bits of vector to avoid discontinuity. In addition, the vector file has to be in the multiple of 16 bits to avoid discontinuity.

DM482e_ConfigureVectorEngineAttribute configures the attribute of vector engine.

This function returns zero if successful and non-zero if otherwise.

DM482e_ConfigureClockFrequency

Synopsis

ViStatus DM482e_ConfigureClockFrequency (vi, frequency)

Arguments

ViSession vi (in)
> Specifies the instrument handle.

ViReal64 frequency (in)
> Specify the frequency, in Hz.

Descriptions

DM482e_ConfigureClockFrequency programs the clock frequency for states DM482E_CONST_FORCE_STATE_CLOCK and DM482E_CONST_FORCE_STATE_INVERTED_CLOCK as specified in DM482e_DPINForce function.

This function returns zero if successful and non-zero if otherwise.

DM482e_AcquireVectorFailCount

Synopsis

ViStatus DM482e_AcquireVectorFailCount (vi, failCount)

Arguments

ViSession vi (in)
> Specifies the instrument handle.

ViInt32* failCount (out)
> Return the total fail count.

Descriptions

DM482e_AcquireVectorFailCount returns the total fail count during the vector execution.

This function returns zero if successful and non-zero if otherwise.

DM482e_AcquireChannelVectorFailCount

ViStatus DM482e_AcquireChannelVectorFailCount (vi, pinNo, failCount)

Arguments

ViSession vi (in)

> Specifies the instrument handle.

ViInt32 pinNo(in)

> Specifies the pin to be acted upon.
> If dpingroup_sel = 1 during DM482e_DPINOpen, pinNo is 0-5 only.
> If dpingroup_sel = 2 during DM482e_DPINOpen, pinNo is 6-11 only.
> If dpingroup_sel = 3 during DM482e_DPINOpen, pinNo is 0-11.

ViInt32* failCount (out)

> Return the fail count.

Descriptions

DM482e_AcquireChannelVectorFailCount returns the fail count of the specified pin during the vector execution.

This function returns zero if successful and non-zero if otherwise.

DM482e_AcquireChannelFirstFailVectorCount

Synopsis

ViStatus DM482e_AcquireChannelFirstFailVectorCount (vi, pinNo, failCount)

Arguments

ViSession vi (in)

Specifies the instrument handle.

ViInt32 pinNo(in)

Specifies the pin to be acted upon.
If dpingroup_sel = 1 during DM482e_DPINOpen, pinNo is 0-5 only.
If dpingroup_sel = 2 during DM482e_DPINOpen, pinNo is 6-11 only.
If dpingroup_sel = 3 during DM482e_DPINOpen, pinNo is 0-11.

ViInt32* failCount (out)

Return the first-failed vector location count.

Descriptions

DM482e_AcquireChannelFirstFailVectorCount returns the first location where the vector starts to fail. This function will return -1 if no vector fails.

This function returns zero if successful and non-zero if otherwise.

DM482e_ConfigureInputChannelDelay

ViStatus DM482e_ConfigureInputChannelDelay (vi, pinNo, delay)

ViSession vi (in)
> Specifies the instrument handle.

ViInt32 pinNo(in)
> Specifies the pin to be acted upon.
> If dpingroup_sel = 1 during DM482e_DPINOpen, pinNo is 0-5 only.
> If dpingroup_sel = 2 during DM482e_DPINOpen, pinNo is 6-11 only.
> If dpingroup_sel = 3 during DM482e_DPINOpen, pinNo is 0-11.

ViReal64 delay (in)
> Specifies the delay, in seconds. Delay range is from -10 of clock cycle to 100 of clock
> cycles.
> The delay resolution is 11 ps for delay from 0 ns to 5.621 ns.
> For delay more than 5.621 ns, the delay resolution is half of the clock cycle.

DM482e_ConfigureInputChannelDelay configure the delay to be inserted to the specified pin during DM482e_RunVector for pin-to-pin skew adjustment purpose.

Below is an example of how to use DM482e_ConfigureInputChannelDelay to adjust for the input skew, when sending "0010111" to the receiver (pin 0).

1. Reset delay to 0 via DM482e_ConfigureInputChannelDelay.
2. Transmit data "0010111" to receiver.
3. Read back data via DM482e_ReadHistoryRam, and compare with expected results.
4. Acquire the pin fail count via DM482e_AcquireChannelVectorFailCount.
5. For every fail bit, tune the delay via DM482e_ConfigureInputChannelDelay, and repeat steps 3-4, until all results are passing.

This function returns zero if successful and non-zero if otherwise.

Section 4: Pin Measurement Unit

DM482e_ConfigurePMUVoltageLevel

ViStatus DM482e_ConfigurePMUVoltageLevel (vi, pinNo, level)

ViSession vi (in)
> Specifies the instrument handle.

ViInt32 pinNo(in)
> Specifies the pin to be acted upon.
> If dpingroup_sel = 1 during DM482e_DPINOpen, pinNo is 0-5 only.
> If dpingroup_sel = 2 during DM482e_DPINOpen, pinNo is 6-11 only.
> If dpingroup_sel = 3 during DM482e_DPINOpen, pinNo is 0-11.

ViReal64 level (in)
> Specifies the voltage level, in volts.

DM482e_ConfigurePMUVoltageLevel configures the voltage level for the specified pin.

This function returns zero if successful and non-zero if otherwise.

DM482e_ConfigurePMUVoltageLimit

ViStatus DM482e_ConfigurePMUVoltageLimit (vi, pinNo, high_limit, low_limit)

ViSession vi (in)

Specifies the instrument handle.

ViInt32 pinNo(in)

Specifies the pin to be acted upon.

If dpingroup_sel = 1 during DM482e_DPINOpen, pinNo is 0-5 only.

If dpingroup_sel = 2 during DM482e_DPINOpen, pinNo is 6-11 only.

If dpingroup_sel = 3 during DM482e_DPINOpen, pinNo is 0-11.

ViReal64 high_limit (in)

Specifies the voltage clamp level for High Clamp Range (VCH), in volts.

ViReal64 low_limit (in)

Specifies the voltage clamp level for Low Clamp Range (VCL), in volts.

DM482e_ConfigurePMUVoltageLimit configures the voltage compliance/clamp limits for the specified pin.

This function returns zero if successful and non-zero if otherwise.

DM482e_ConfigurePMUCurrentLevel

ViStatus DM482e_ConfigurePMUCurrentLevel (vi, pinNo, level)

ViSession vi (in)
> Specifies the instrument handle.

ViInt32 pinNo(in)
> Specifies the pin to be acted upon.
> If dpingroup_sel = 1 during DM482e_DPINOpen, pinNo is 0-5 only.
> If dpingroup_sel = 2 during DM482e_DPINOpen, pinNo is 6-11 only.
> If dpingroup_sel = 3 during DM482e_DPINOpen, pinNo is 0-11.

ViReal64 level (in)
> Specifies the current level, in amps.

DM482e_ConfigurePMUCurrentLevel configures the current level for the specified pin.

This function returns zero if successful and non-zero if otherwise.

DM482e_ConfigurePMUCurrentLevelAndRange

ViStatus DM482e_ConfigurePMUCurrentLevelAndRange (vi, pinNo, level, range)

ViSession vi (in)

Specifies the instrument handle.

ViInt32 pinNo(in)

Specifies the pin to be acted upon.

If dpingroup_sel = 1 during DM482e_DPINOpen, pinNo is 0-5 only.

If dpingroup_sel = 2 during DM482e_DPINOpen, pinNo is 6-11 only.

If dpingroup_sel = 3 during DM482e_DPINOpen, pinNo is 0-11.

ViReal64 level (in)

Specifies the current level, in amps.

ViReal64 range (in)

Specifies the current range, in amps.

DM482e_ConfigurePMUCurrentLevelAndRange configures the current level and its range for the specified pin.

This function returns zero if successful and non-zero if otherwise.

DM482e_ConfigurePMUCurrentLimitRange

ViStatus DM482e_ConfigurePMUCurrentLimitRange (vi, pinNo, range)

ViSession vi (in)

> Specifies the instrument handle.

ViInt32 pinNo(in)

> Specifies the pin to be acted upon.
> If dpingroup_sel = 1 during DM482e_DPINOpen, pinNo is 0-5 only.
> If dpingroup_sel = 2 during DM482e_DPINOpen, pinNo is 6-11 only.
> If dpingroup_sel = 3 during DM482e_DPINOpen, pinNo is 0-11.

ViReal64 range (in)

> Specifies the current limit range, in amps.

DM482e_ConfigurePMUCurrentLimitRange configures the current limit (for source/sink operations) range for the specified pin.

This function returns zero if successful and non-zero if otherwise.

DM482e_GetPMUVoltageLevelRange

ViStatus DM482e_GetPMUVoltageLevelRange (vi, pinNo, range)

ViSession vi (in)
> Specifies the instrument handle.

ViInt32 pinNo(in)
> Specifies the pin to be acted upon.
> If dpingroup_sel = 1 during DM482e_DPINOpen, pinNo is 0-5 only.
> If dpingroup_sel = 2 during DM482e_DPINOpen, pinNo is 6-11 only.
> If dpingroup_sel = 3 during DM482e_DPINOpen, pinNo is 0-11.

ViReal64* range (out)
> Returns the range of the voltage level, in volts.

DM482e_GetPMUVoltageLevelRange returns the range of the voltage level, in volts, of the specified pin.

This function returns zero if successful and non-zero if otherwise.

DM482e_GetPMUVoltageLimitRange

ViStatus DM482e_GetPMUVoltageLimitRange (vi, pinNo, range)

ViSession vi (in)
> Specifies the instrument handle.

ViInt32 pinNo(in)
> Specifies the pin to be acted upon.
> If dpingroup_sel = 1 during DM482e_DPINOpen, pinNo is 0-5 only.
> If dpingroup_sel = 2 during DM482e_DPINOpen, pinNo is 6-11 only.
> If dpingroup_sel = 3 during DM482e_DPINOpen, pinNo is 0-11.

ViReal64* range (out)
> Returns the range of the voltage limit, in volts.

DM482e_GetPMUVoltageLimitRange returns the range of the voltage limit, in volts, of the specified pin.

This function returns zero if successful and non-zero if otherwise.

DM482e_GetPMUCurrentLevelRange

ViStatus DM482e_GetPMUCurrentLevelRange (vi, pinNo, range)

Arguments

ViSession vi (in)
> Specifies the instrument handle.

ViInt32 pinNo(in)
> Specifies the pin to be acted upon.
> If dpingroup_sel = 1 during DM482e_DPINOpen, pinNo is 0-5 only.
> If dpingroup_sel = 2 during DM482e_DPINOpen, pinNo is 6-11 only.
> If dpingroup_sel = 3 during DM482e_DPINOpen, pinNo is 0-11.

ViReal64* range (out)
> Returns the range of the current level, in amps.

Descriptions

DM482e_GetPMUCurrentLevelRange returns the range of the current level, in amps, of the specified pin.

This function returns zero if successful and non-zero if otherwise.

DM482e_GetPMUCurrentLimitRange

ViStatus DM482e_GetPMUCurrentLimitRange (vi, pinNo, range)

ViSession vi (in)
> Specifies the instrument handle.

ViInt32 pinNo(in)
> Specifies the pin to be acted upon.
> If dpingroup_sel = 1 during DM482e_DPINOpen, pinNo is 0-5 only.
> If dpingroup_sel = 2 during DM482e_DPINOpen, pinNo is 6-11 only.
> If dpingroup_sel = 3 during DM482e_DPINOpen, pinNo is 0-11.

ViReal64* range (out)
> Returns the range of the current limit, in amps.

DM482e_GetPMUCurrentLimitRange returns the range of the current limit, in amps, of the specified pin.

This function returns zero if successful and non-zero if otherwise.

DM482e_ConfigurePMUOutputFunction

ViStatus DM482e_ConfigurePMUOutputFunction (vi, pinNo, function)

Arguments

ViSession vi (in)
> Specifies the instrument handle.

ViInt32 pinNo(in)
> Specifies the pin to be acted upon.
> If dpingroup_sel = 1 during DM482e_DPINOpen, pinNo is 0-5 only.
> If dpingroup_sel = 2 during DM482e_DPINOpen, pinNo is 6-11 only.
> If dpingroup_sel = 3 during DM482e_DPINOpen, pinNo is 0-11.

ViInt32 function (out)
> Select DVCI or DICV mode.

Descriptions

DM482e_ConfigurePMUOutputFunction configures the pin to operate either in DVCI (Drive-Voltage Clamp Current) or DICV (Drive-Current Clamp Voltage) mode:

function	Setting
DVCI	0 (Default)
DICV	1

Table 6: Output Function

This function returns zero if successful and non-zero if otherwise.

DM482e_ConfigurePMUSamplingTime

Synopsis

ViStatus DM482e_ConfigurePMUSamplingTime (vi, pinNo, samplingTime, units)

Arguments

ViSession vi (in)

       Specifies the instrument handle.

ViInt32 pinNo(in)

       Specifies the pin to be acted upon.

       If dpingroup_sel = 1 during DM482e_DPINOpen, pinNo is 0-5 only.

       If dpingroup_sel = 2 during DM482e_DPINOpen, pinNo is 6-11 only.

       If dpingroup_sel = 3 during DM482e_DPINOpen, pinNo is 0-11.

ViReal64 samplingTime (in)

       Specifies the sampling time.

ViInt32 units (in)

       Specifies the unit of the sampling time:

units	Description
0	Seconds
1	PLC

Table 7: Sampling Time Unit

Descriptions

DM482e_ConfigurePMUSamplingTime configures the sampling time or NPLC (number of power line cycle) on the specified channel(s). The supported values depend on the units.

The measurement result is the average value of all samples taken for nth number of power line cycle (PLC). For example, if the line frequency is set to 50Hz (period is 20ms), then 1 PLC equals to 20ms. The measured data is a resultant of averaged of raw samples captured by ADC for 20ms.

The NPLC setting of an instrument allows adjustment of the tradeoff between speed and accuracy. The greater the number of power line cycles, the greater noise rejection and better resolution the signal value will be.

The following table lists some examples of sampling times for the instrument, in PLC format:

PLC	60Hz Power Line Frequency	50Hz Power Line Frequency
8	133.3 ms	160 ms
4	66.66 ms	80 ms
2	33.33 ms	40 ms
1	16.66 ms	20 ms
1/2	8.33 ms	10 ms
1/4	4.16 ms	5 ms
1/8	2.08 ms	2.5 ms
1/16	1.04 ms	1.25 ms
1/32	520 us	625 us
1/64	260 us	312 us

Table 8: PLC Setting

This function returns zero if successful and non-zero if otherwise.

DM482e_ConfigurePowerLineFrequency

Synopsis

ViStatus DM482e_ConfigurePowerLineFrequency (vi, powerLineFrequency)

Arguments

ViSession vi (in)
    Specifies the instrument handle.

ViReal64 powerLineFrequency (in)
    Specifies the power line frequency.

Descriptions

DM482e_ConfigurePowerLineFrequency specifies the frequency of the AC power line:

plf	Description
50.0	50Hz power line frequency (Default)
60.0	60Hz power line frequency

Table 9: Power Line Frequency

This function returns zero if successful and non-zero if otherwise.

DM482e_ConfigurePMUSense

ViStatus DM482e_ConfigurePMUSense (vi, pinNo, sense)

ViSession vi (in)
> Specifies the instrument handle.

ViInt32 pinNo(in)
> Specifies the pin to be acted upon.
> If dpingroup_sel = 1 during DM482e_DPINOpen, pinNo is 0-5 only.
> If dpingroup_sel = 2 during DM482e_DPINOpen, pinNo is 6-11 only.
> If dpingroup_sel = 3 during DM482e_DPINOpen, pinNo is 0-11.

ViInt32 sense (in)
> Specifies local or remote sense operation.

DM482e_ConfigurePMUSense selects either local or remote sense operation:

sense	Description
0	Local sense (Default)
1	Remote sense

Table 10: Local/Remote Sense Operation

This function returns zero if successful and non-zero if otherwise.

DM482e_GetPMUSense

ViStatus DM482e_GetPMUSense (vi, pinNo, sense)

ViSession vi (in)

Specifies the instrument handle.

ViInt32 pinNo(in)

Specifies the pin to be acted upon.
If dpingroup_sel = 1 during DM482e_DPINOpen, pinNo is 0-5 only.
If dpingroup_sel = 2 during DM482e_DPINOpen, pinNo is 6-11 only.
If dpingroup_sel = 3 during DM482e_DPINOpen, pinNo is 0-11.

ViInt32* sense (out)

Returns the sense (local/remote) operation.

DM482e_GetPMUSense returns the current sense status of the specified pin.

This function returns zero if successful and non-zero if otherwise.

DM482e_PMUMeasure

Synopsis

ViStatus DM482e_PMUMeasure (vi, pinNo, measurementType, measurement)

Arguments

ViSession vi (in)

Specifies the instrument handle.

ViInt32 pinNo(in)

Specifies the pin to be acted upon.

If dpingroup_sel = 1 during DM482e_DPINOpen, pinNo is 0-5 only.

If dpingroup_sel = 2 during DM482e_DPINOpen, pinNo is 6-11 only.

If dpingroup_sel = 3 during DM482e_DPINOpen, pinNo is 0-11.

ViInt32 measurementType (in)

Specifies voltage or current measurement.

ViReal64* measurement (out)

Returns the measured result.

Descriptions

DM482e_PMUMeasure returns the measured data based on measurementType:

measurementType	Description
0	Measure Current
1	Measure Voltage

Table 11: Voltage/Current Measurement Selection

This function returns zero if successful and non-zero if otherwise.

Section 5: Trigger

DM482e_ConfigureTriggerEdgeLevel

Synopsis

ViStatus DM482e_ConfigureTriggerEdgeLevel (vi, triggerEnum, option)

Arguments

ViSession vi (in)

Specifies the instrument handle.

ViInt32 triggerEnum (in)

Specifies the triggering source.

ViInt32 option (in)

Specifies the triggering mode.

Descriptions

DM482e_ConfigureTriggerEdgeLevel configures how device can be triggered.

triggerEnum	Description
0	PXI backplane trigger line 0
1	PXI backplane trigger line 1
2	PXI backplane trigger line 2
3	PXI backplane trigger line 3
4	PXI backplane trigger line 4
5	PXI backplane trigger line 5
6	PXI backplane trigger line 6
7	PXI backplane trigger line 7
8	PXI backplane PXI_LBL6 signal
9	PXI backplane PXI_LBR6 signal
10	PXI backplane star trigger line
11	PXI backplane differential start trigger PXIE_DSTARA

12	PXI backplane differential start trigger PXIE_DSTARB
13	PXI backplane differential start trigger PXIE_DSTARC
14	External input trigger port 0
15	External input trigger port 1
16	Software trigger signal 0
17	External output trigger port 0
18	External output trigger port 1
19	Software trigger signal 1
20	No trigger

Table 12: Trigger Source

option	Description
0	Pin will be triggered when rising edge is detected. This is the default mode
1	Pin will be triggered when falling edge is detected
2	Pin will be triggered when trigger signal is below a TTL logic level
3	Pin will be triggered when trigger signal exceeds logic level high

Table 13: Trigger Mode

Note that DM482e_ConfigureTriggerEdgeLevel configures has no effects on software trigger lines.

This function returns zero if successful and non-zero if otherwise.

DM482e_MapTriggerInToTriggerOut

ViStatus DM482e_MapTriggerInToTriggerOut (vi, inputTerminal, outputTerminal)

Arguments

ViSession vi (in)
    Specifies the instrument handle.

ViInt32 inputTerminal (in)
    Specifies the triggering line to be connected to outputTerminal.

ViInt32 outputTerminal (in)
    Specifies the triggering line to be connected to inputTerminal.

Descriptions

DM482e_MapTriggerInToTriggerOut connects 2 trigger lines together.

inputTerminal / outputTerminal	Description
0	PXI backplane trigger line 0
1	PXI backplane trigger line 1
2	PXI backplane trigger line 2
3	PXI backplane trigger line 3
4	PXI backplane trigger line 4
5	PXI backplane trigger line 5
6	PXI backplane trigger line 6
7	PXI backplane trigger line 7
8	PXI backplane PXI_LBL6 signal
9	PXI backplane PXI_LBR6 signal
10	PXI backplane star trigger line
11	PXI backplane differential start trigger PXIE_DSTARA
12	PXI backplane differential start trigger PXIE_DSTARB
13	PXI backplane differential start trigger PXIE_DSTARC

14	External input trigger port 0
15	External input trigger port 1
16	Software trigger signal 0
17	External output trigger port 0
18	External output trigger port 1
19	Software trigger signal 1
20	No trigger

Table 14: Available Trigger Lines

This function returns zero if successful and non-zero if otherwise.

DM482e_ConfigureInputTriggerSelect

Synopsis

ViStatus DM482e_ConfigureInputTriggerSelect (vi, triggerInput)

Arguments

ViSession vi (in)

　　　　Specifies the instrument handle.

ViInt32 triggerInput (in)

　　　　Specifies the triggering source.

Descriptions

DM482e_ConfigureInputTriggerSelect selects the trigger input to be used.

triggerEnum	Description
0	PXI backplane trigger line 0
1	PXI backplane trigger line 1
2	PXI backplane trigger line 2
3	PXI backplane trigger line 3
4	PXI backplane trigger line 4
5	PXI backplane trigger line 5
6	PXI backplane trigger line 6
7	PXI backplane trigger line 7
8	PXI backplane PXI_LBL6 signal
9	PXI backplane PXI_LBR6 signal
10	PXI backplane star trigger line
11	PXI backplane differential start trigger PXIE_DSTARA
12	PXI backplane differential start trigger PXIE_DSTARB
14	External input trigger port 0
15	External input trigger port 1
16	Software trigger signal 0
19	Software trigger signal 1

20	No trigger

Table 15: Available Trigger Inputs

This function returns zero if successful and non-zero if otherwise.

DM482e_ConfigureOutputTriggerSelect

Synopsis

ViStatus DM482e_ConfigureOutputTriggerSelect (vi, triggerOutput0, triggerOutput1)

Arguments

ViSession vi (in)
       Specifies the instrument handle.

ViInt32 triggerOutput0 (in)
       Specifies the triggering output signal.

ViInt32 triggerOutput1 (in)
       Specifies the triggering output signal.

Descriptions

DM482e_ConfigureOutputTriggerSelect selects the trigger output to be used.

triggerEnum	Description
0	PXI backplane trigger line 0
1	PXI backplane trigger line 1
2	PXI backplane trigger line 2
3	PXI backplane trigger line 3
4	PXI backplane trigger line 4
5	PXI backplane trigger line 5
6	PXI backplane trigger line 6
7	PXI backplane trigger line 7
8	PXI backplane PXI_LBL6 signal
9	PXI backplane PXI_LBR6 signal
10	PXI backplane star trigger line

13	PXI backplane differential start trigger PXIE_DSTARC
17	External output trigger port 0
18	External output trigger port 1
20	No trigger

Table 16: Available Trigger Outputs

This function returns zero if successful and non-zero if otherwise.

DM482e_DriveSoftwareTrigger

ViStatus DM482e_DriveSoftwareTrigger (vi, select, pulseWidth)

Arguments

ViSession vi (in)
>   Specifies the instrument handle.

ViInt32 select (in)
>   Specifies the software trigger line.

ViReal64 pulseWidth (in)
>   Specifies the pulse width of the trigger signal, in seconds.

Descriptions

DM482e_DriveSoftwareTrigger drives software trigger immediately. Software trigger lines are digital signals coming from the on-board processor.

select	Description
16	Software trigger signal 0
19	Software trigger signal 1

Table 17: Available Software Trigger Lines

This function returns zero if successful and non-zero if otherwise.

Section 6: MIPI

DM482e_MIPI_Configure_Clock

ViStatus DM482e_MIPI_Configure_Clock (vi, mipi_pair, freq_Hz)

ViSession vi (in)
> Specifies the instrument handle.

ViInt32 mipi_pair (in)
> Specifies the MIPI controller set number.

mipi_pair	Remark	Pin Number	Function
0	Available for dpingroup 1 & 3 only	0	SCLK0
		1	SDATA0
		2	VIO0
1	Available for dpingroup 1 & 3 only	3	VIO1
		4	SCLK1
		5	SDATA1
2	Available for dpingroup 2 & 3 only	6	SCLK2
		7	SDATA2
		8	VIO2
3	Available for dpingroup 2 & 3 only	9	VIO3
		10	SCLK3
		11	SDATA3

Table 18: MIPI Pair

ViInt32 freq_Hz (in)
> Specifies the frequency of the full-speed clock.

DM482e_MIPI_Configure_Clock configures the full-clock frequency for MIPI operations.

Refer to Specifications and User Manual for the available number sets of MIPI controller per DM482e, as well as the supported full-clock frequency.

This function returns zero if successful and non-zero if otherwise.

DM482e_MIPI_Connect

Synopsis

ViStatus DM482e_MIPI_Connect (vi, mipi_pair, setting)

Arguments

ViSession vi (in)

Specifies the instrument handle.

ViInt32 mipi_pair (in)

Specifies the MIPI controller set number.

mipi_pair	Remark	Pin Number	Function
0	Available for dpingroup 1 & 3 only	0	SCLK0
		1	SDATA0
		2	VIO0
1	Available for dpingroup 1 & 3 only	3	VIO1
		4	SCLK1
		5	SDATA1
2	Available for dpingroup 2 & 3 only	6	SCLK2
		7	SDATA2
		8	VIO2
3	Available for dpingroup 2 & 3 only	9	VIO3
		10	SCLK3
		11	SDATA3

Table 19 :MIPI Pair

ViInt32 setting (in)

Enable or disable MIPI operations.

Descriptions

DM482e_MIPI_Connect enables or disables MIPI operations for the selected mipi_pair. Once MIPI is enabled, all other operation modes (VECTOR, PMU, DIO, CLOCK, and INVERTED_CLOCK) will be disabled automatically.

setting	Description
0	Disable MIPI
1	Enable MIPI

Table 20: MIPI setting

Refer to Specifications and User Manual for the available number sets of MIPI controller per DM482e, as well as the supported full-clock frequency.

This function returns zero if successful and non-zero if otherwise.

DM482e_MIPI_RFFE_WR

Synopsis

ViStatus DM482e_MIPI_RFFE_WR (vi, mipi_pair, Command, Data)

Arguments

ViSession vi (in)

Specifies the instrument handle.

ViInt32 mipi_pair (in)

Specifies the MIPI controller set number.

mipi_pair	Remark	Pin Number	Function
0	Available for dpingroup 1 & 3 only	0	SCLK0
		1	SDATA0
		2	VIO0
1	Available for dpingroup 1 & 3 only	3	VIO1
		4	SCLK1
		5	SDATA1
2	Available for dpingroup 2 & 3 only	6	SCLK2
		7	SDATA2
		8	VIO2
3	Available for dpingroup 2 & 3 only	9	VIO3
		10	SCLK3
		11	SDATA3

Table 21: MIPI Pair

ViInt32 Command (in)

Specifies the command for the operation.

Command_frame [11:0] = { Slave_Address[2], Command_frame_lower }		
Register 0 Write	Command_frame_lower[7:0]	8'b{ 1, Data[6:0] }
Register Write	Command_frame_lower[7:0]	8'b{ 0,1,0, Address[4:0] }
Extended Register Write	Command_frame_lower[7:0]	8'b{ 0,0,0,0, BC[3:0] }

Extended Register Write Long	Command_frame_lower[7:0]	8'b{ 0,0,1,1,0, BC[2:0] }

<p align="center">Table 22: Write Command Frame</p>

1. Command frame excludes parity bit, which is handled by firmware.
2. Slave address is 4 bits and common for all operations.
3. BC => byte_count for data frame, ie. If BC is zero, then M = 1, refer to data frame table

ViInt32 * Data (in)

Specifies a pointer to an array corresponding to each 8 bit data frame that will be written to the pin.

Data_frame[7:0]		
Register 0 Write	-	-
Register Write	One Data_frame	-
Extended Register Write	One Data_frame for Address	Data_frame[0]= address [7:0], followed by M^2 number of Data Frames containing up to 16 bytes of data. Example: If M = 3, Data_frame[1] = data0, Data_frame[2] = data1, Data_frame[3] = data2.
Extended Register Write Long	Two Data_frame for Address	Data_frame[0]= address [15:8] and Data_frame[1] = address [7:0], followed by M^2 number of Data Frames containing up to 8 bytes of data. Example: If M = 3, Data_frame[2] = data0, Data_frame[3] = data1, Data_frame[4] = data2

<p align="center">Table 23: Write Data Frame</p>

1. Data frame excludes parity bit, which is handled by firmware.
2. BC => byte_count for data frame. If BC is zero, then M = 1, refer to data frame table.

Descriptions

DM482e_MIPI_RFFE_WR writes the data into the pin according to the operation selected.

This function returns zero if successful and non-zero if otherwise.

Example Code

This example performs the operation of Extended Register Write Long.

Command = (0xF<<8)|(0x6<<3)|(0x2); //Extended register write long, 3 data bytes
Data[0] = 0x1; //Address [15:8]
Data[1] = 0x23; //Address [7:0]
Data[2] = 0x31; //Byte 1 data
Data[3] = 0x31; //Byte 2 data
Data[4] = 0x31; //Byte 3 data
DM482e_MIPI_RFFE_WR(vi, 0, Command, Data); //Select MIPI controller set 0

For the Command, (0xF<<8)is the slave address, (0x6<<3)is the command frame and (0x2) is data_frame with the number of byte to be written into the channel.

Note: Refer to appendix for more details.

DM482e_MIPI_RFFE_RD

ViStatus DM482e_MIPI_RFFE_RD (vi, mipi_pair, speed, Command, Data)

ViSession vi (in)

> Specifies the instrument handle.

ViInt32 mipi_pair (in)

> Specifies the MIPI controller set number.

mipi_pair	Remark	Pin Number	Function
0	Available for dpingroup 1 & 3 only	0	SCLK0
		1	SDATA0
		2	VIO0
1	Available for dpingroup 1 & 3 only	3	VIO1
		4	SCLK1
		5	SDATA1
2	Available for dpingroup 2 & 3 only	6	SCLK2
		7	SDATA2
		8	VIO2
3	Available for dpingroup 2 & 3 only	9	VIO3
		10	SCLK3
		11	SDATA3

Table 24: MIPI Pair

ViInt32 speed (in)

> Specifies the speed of reading.
>
> 1 = full speed, 0 = half speed.

ViInt32 Command (in)

> Specifies the command of read operation.

Command_frame [11:0] = { Slave_Address[2], Command_frame_lower }		
Register Read	Command_frame_lower[7:0]	8'b{ 0,1,1, Address[4:0] }
Extended Register Read	Command_frame_lower[7:0]	8'b{ 0,0,1,0, BC[3:0] }
Extended Register Read Long	Command_frame_lower[7:0]	8'b{ 0,0,1,1,1, BC[2:0] }

Table 25: Read Command Frame

1. Command frame excludes parity bit, which is handled by firmware.
2. Slave address is 4 bits and common for all operations.

ViInt32* Data (in)
    The array of addresses that will be reading from the channel.

Data_frame[7:0]		
Register Read	-	-
Extended Register Read	One Data frame for Address	Data_frame[0]= address [7:0]
Extended Register Read Long	Two Data frame for Address	Data_frame[0]= address [15:8]; Data_frame[1]= address [7:0]

Table 26: Read Data Frame

1. Data frame excludes parity bit, which is handled by firmware.

Descriptions

DM482e_MIPI_RFFE_RD reads the data from the corresponding addresses.

This function returns zero if successful and non-zero if otherwise.

Example Code

This example performs Extended Register Read operation.

Command = (0xF<<8)|(0x2<<4)|(0x2); //Specifies command for Register Read
DataRead[0] = 0x1; //Specifies the address to read from
DM482e_MIPI_RFFE_RD(vi,0,1,Command,DataRead) //Select MIPI controller set 0. Use full-speed clock.

For the Command, (0xF<<8)is the slave address, (0x2<<4)is the command frame and (0x2) is the number of byte to be read from the channel.

Note: Refer to appendix for more details.

DM482e_MIPI_RFFE_RETRIEVE

Synopsis

ViStatus   DM482e_MIPI_RFFE_RETRIEVE   (vi,   mipi_pair,   rd_byte_data_count,   rd_data, parity_check)

Arguments

ViSession vi (in)

Specifies the instrument handle.

ViInt32 mipi_pair (in)

Specifies the MIPI controller set number.

mipi_pair	Remark	Pin Number	Function
0	Available for dpingroup 1 & 3 only	0	SCLK0
		1	SDATA0
		2	VIO0
1	Available for dpingroup 1 & 3 only	3	VIO1
		4	SCLK1
		5	SDATA1
2	Available for dpingroup 2 & 3 only	6	SCLK2
		7	SDATA2
		8	VIO2
3	Available for dpingroup 2 & 3 only	9	VIO3
		10	SCLK3
		11	SDATA3

Table 27: MIPI Pair

ViInt32* rd_byte_data_count (out)

Specifies the number of bytes of data retrieved from the pin.

ViInt32* rd_data (out)

Returns the array of data that retrieved from pin.

rd_data format is "ABCDEFGHIJKLMNOP" in binary, where

"OP" refers to LSB, "MN" refers to bit 1, "KL" refers to bit 2, "AB" refers to MSB, and so on. 2 binary bit represent the logic state, as shown below.

Binary Bits	Description
00	Logic low (L)
01	Logic high (H)
10	Undefined state (Z)

Table 28: Binary Representation of Logic State

ViInt32* parity_check (out)

Returns the array of parity_check corresponding to the array of rd_data that retrieved from the pin.

Value	Description
0	Parity check passes
1	Parity check fails, or undefined read back data found

Table 29: Binary Representation of Logic State

Descriptions

DM482e_MIPI_RFFE_RETRIEVE retrieves and returns an array of data specified in operation DM482e_MIPI_RFFE_RD.

This function returns zero if successful and non-zero if otherwise.

DM482e_MIPI_ConfigureInputDelay

Synopsis

ViStatus DM482e_MIPI_ConfigureInputDelay (vi, mipi_pair, value)

Arguments

ViSession vi (in)
Specifies the instrument handle.

ViInt32 mipi_pair (in)
Specifies the MIPI controller set number.

mipi_pair	Remark	Pin Number	Function
0	Available for dpingroup 1 & 3 only	0	SCLK0
		1	SDATA0
		2	VIO0
1	Available for dpingroup 1 & 3 only	3	VIO1
		4	SCLK1
		5	SDATA1
2	Available for dpingroup 2 & 3 only	6	SCLK2
		7	SDATA2
		8	VIO2
3	Available for dpingroup 2 & 3 only	9	VIO3
		10	SCLK3
		11	SDATA3

Table 30: MIPI Pair

ViInt32 value (in)
Specifies the delay inserted before data sampling during read operation starts.
$0 \leqslant value \leqslant 7$.
Value is 0.5*MIPI full speed clock period.
If value is 1, then, input delay is 0.5*MIPI full speed clock period.
If value is 2, then, input delay is 1*MIPI full speed clock period.

Descriptions

DM482e_MIPI_ConfigureInputDelay is needed when the cable distance between DM482e and the DUT is too long. This API tells the MIPI IP to sample the incoming SDATA slower than usual due to long cable delay.

This function returns zero if successful and non-zero if otherwise.

Section 7: Appendix – RFFE Supported Command Sequences

Figure 1: RFFE Supported Command Sequences

Section 8: Error Message

Error Code (Hex)	Description
0xAE050006	AEMHW_E_WRONG_MSG_SIZE
0xAE050007	AEMHW_E_API_NOT_IMPLEMENTED
0xAE050009	AEMHW_E_FIRMWARE_FILE_CHECKSUM_ERROR
0xAE05000A	AEMHW_E_FIRMWARE_FILESIZE_ERROR
0xAE05000B	AEMHW_E_WAIT4UNLOCK_TIMEOUT
0xAE090001	AEMPXIE_E_OPEN_FAIL
0xAE090002	AEMPXIE_E_DLL_NOT_FOUND
0xAE090003	AEMPXIE_E_API_NOT_FOUND
0xAE090004	AEMPXIE_E_INVALID_MULTI_SITE_MODE
0xAE040002	AEMVECTOR_SYNTAX_ERROR
0xAE040003	AEMVECTOR_FILE_CORRUPTED
0xAE040008	AEMVECTOR_NOT_FOUND
0xAE04000C	AEMVECTOR_INVALID_ARGUMENT
0xAE04000D	AEMVECTOR_INVALID_CH
0xAE04000E	AEMVECTOR_UNDEFINED_PIN
0xAE0B0001	AEMDM_E_INVALID_ARGUMENTS
0xAE0B0002	AEMDM_E_INVALID_CALIBRATION_ARGUMENTS
0xAE0B0003	AEMDM_E_UNDEFINED_VECTOR_SET
0xAE0B0004	AEMDM_E_CHANNEL_NOT_IN_PMU_STATE
0xAE0B0005	AEMDM_E_ADATE_TIMEOUT
0xAE0B0006	AEMDM_E_ADC_TIMEOUT
0xAE0B0007	AEMDM_E_DDS_TIMEOUT
0xAE0B0008	AEMDM_E_PATTERN_MEMORY_TIMEOUT
0xAE0B0009	AEMDM_E_HISTORY_RAM_TIMEOUT
0xAE0B000A	AEMDM_E_D2R_FAIL_CALIBRATION_STAGE1
0xAE0B000B	AEMDM_E_CAL_HEADER_NOT_FOUND
0xAE0B000C	AEMDM_E_CAL_SIZE_NOT_MATCH
0xAE0B000D	AEMDM_E_INVALID_CAL_ACCUM_COUNT
0xAE0B000E	AEMDM_E_INVALID_CAL_DATE
0xAE0B000F	AEMDM_E_VECTOR_ENGINE_BUSY
0xAE0B0010	AEMDM_E_EXCEED_VECTOR_MEMORY_LIMIT
0xAE0B0011	AEMDM_E_EXCEED_TOTAL_VECTOR_MEMORY_LIMIT
0xAE0B0012	AEMDM_E_DM482E_NOT_FOUND
0xAE0B0013	AEMDM_E_D2R_FAIL_CALIBRATION_STAGE2

0xAE0B0014	AEMDM_E_INPUT_DELAY_OUT_OF_RANGE
0xAE0B0015	AEMDM_E_EXCEED_PERIOD_LIMIT
0xAE0B0016	AEMDM_E_EXCEED_READ_VECTOR_LIMIT
0xAE0B0017	AEMDM_E_CHANNEL_NOT_IN_DIO_STATE
0xAE0B0018	AEMDM_E_INVALID_DIO_VALUE
0xAE0B0041	AEMDM_E_INVALID_CLK_FREQ
0xAE0B0042	AEMDM_E_INVALID_CH
0xAE0B0043	AEMDM_E_INVALID_MIPI_COMMAND
0xAE0B0044	AEMDM_E_INVALID_MIPI_DATA
0xAE0B0045	AEMDM_E_INVALID_MIPI_CONNECT
0xAE0B0046	AEMDM_E_INVALID_MIPI_SPEED
0xAE0B0047	AEMDM_E_MIPI_STATUS_TIMEOUT
0xAE0B0048	AEMDM_E_INVALID_MIPI_RDDATA_COUNT
0xAE0B0049	AEMDM_E_INVALID_MIPI_RDDATA

Section 9: Revision History

1.0	DEC 2012	INITIAL RELEASE
1.1	JUNE 2013	ADDED MIPI FUNCTIONS
1.2	AUG 2013	ADDED MIPI_PAIR DESCRIPTION ADDED DM482E_MIPI_CONFIGUREINPUTDELAY

Section 10: Contact Us

To obtain service, warranty or technical assistance, please contact Aemulus.

Aemulus Corporation Berhad
Krystal Point, B-2-04, B-2-05, B-2-06 & B-2-07
303, Jalan Sultan Azlan Shah,
11900 Penang, Malaysia
Tel: +604 6446399
Fax: +604 6466799

Web: www.aemulus.com
Email: enquiry@aemulus.com

Product specifications and descriptions in this document are subject to change without prior notice.