



WESTERN AVIONICS

**STANAG 3910 VXI (Model 4701)
LabWindows/CVI Drivers**

P/N 1L01698H02 Rev 1.2

**User Manual
UM 01698 Rev C**

**© Western Avionics Ltd.
13/14 Shannon Free Zone
Co. Clare
Ireland**

16 April 1999

Index of Functions

wa4701_clearStatusRegister	6
wa4701_closeInstrSession	7
wa4701_configAndTransmitBCMsgs.....	8
wa4701_configCMAAndAcquireData	9
wa4701_configDevice.....	10
wa4701_configureAndEnableRT.....	12
wa4701_enableDisableRT	13
wa4701_errorMessage.....	15
wa4701_identificationQuery	16
wa4701_init.....	17
wa4701_loadBCMsgData	19
wa4701_loadRTData	21
wa4701_qryCMDelayAndPostTrigCount	24
wa4701_queryBCFrameParams.....	25
wa4701_queryBCMsgError.....	27
wa4701_queryBCMsgParams.....	29
wa4701_queryCMStack	33
wa4701_queryCMStartSeqArmTrig.....	35
wa4701_queryCMStopSeqArmTrig.....	37
wa4701_queryDeviceStatusReg.....	42
wa4701_queryInstrStatus	43
wa4701_queryRTHSError	44
wa4701_queryRTLSError	47
wa4701_queryRTPParams	50
wa4701_querySystemError	52
wa4701_readBCMsgData	54
wa4701_readCMStackData	57
wa4701_readRTData	61
wa4701_reset	64
wa4701_scanChassis	65
wa4701_selfTest	67
wa4701_setBCMsgOrFrameParamsToDefault.....	68
wa4701_setCMDelayAndPostTrigCount.....	69
wa4701_setOperatingMode	70
wa4701_setRTPParamsToDefaults.....	71
wa4701_setRTSrq	72
wa4701_setStatusMask.....	74
wa4701_setupBCCycleMsgSequence	77
wa4701_setupBCFrameParams.....	79
wa4701_setupBCMsgDTAG	81
wa4701_setupBCMsgErrorInj	83
wa4701_setupBCMsgParams	86
wa4701_setupBCMsgTiming	90
wa4701_setupBCParameters	93
wa4701_setupCMStartSeqArmTrig.....	96
wa4701_setupCMStopSeqArmTrig	98
wa4701_setupExternalTrigger.....	102
wa4701_setupRTHSErrInj	103
wa4701_setupRTLSErrInj.....	106
wa4701_setupRTPParams.....	109
wa4701_setupRTSubAddressParams	111
wa4701_setupRTTimingParams.....	114
wa4701_setupScapAndWindParams.....	117
wa4701_startStopBC	119
wa4701_startStopCMAcq	121

IIB-3910-VXI STANAG 3910 Bus Module

Introduction

The IIB-3910-VXI is programmed as a message based instrument using the VXI word serial protocol. IIB-3910-VXI tests and emulates systems utilising the STANAG 3910 digital data bus. It can emulate both the Bus Controller and Remote Terminals, or any number of Remote Terminals, or a Chronological Monitor of the bus traffic. IIB-3910-VXI can be programmed via high level, easy to use commands, via the VXI word serial protocol. Alternatively, it is possible to program IIB-3910-VXI by direct register read/writes to the 2 Mbytes of memory accessible via the A24 VXI bus. This driver programs the module via high level commands.

Instrument Driver

This instrument driver provides programming support for IIB-3910-VXI STANAG 3910 Bus Module. It contains functions for opening, configuring, taking measurement from, and closing the instrument.

Assumptions

To successfully use this module, the following conditions must be met:

For GPIB instruments drivers:

The instrument is connected to the GPIB.

The GPIB address supplied to the initialise function must match the GPIB address of the instrument.

For VXI instruments drivers:

The instrument is installed in the VXI mainframe and you are using one of the following controller options:

Embedded controller

AT-MXI

MC-MXI

GPIB-VXI

The logical address supplied to the initialise function must match the logical address of the instrument.

If you are using GPIB-VXI, there must be at least one unused device name in the handler.

For RS-232 instruments drivers:

The instrument is connected to the RS-232 interface.

The COM port, baud rate, parity, and time-out supplied to the initialise function must match the settings of the instrument.

Error Codes:

Error codes are returned as the return value of each instrument driver function. The `wa4701_errorMessage()` function should be used to retrieve the error message corresponding to the error code returned by this function. A program should examine this value after each call to an instrument driver function to determine if any error occurred. Possible error codes and their meanings are listed with the corresponding instrument driver function.

How To Use This Document:

This document is intended to be used as a programming reference manual.

It describes each function in the IIB-3910-VXI STANAG 3910 Bus Module instrument. The functions are listed in alphabetical order with a description of the function, C syntax of the function, and a description of each parameter.

Function Tree Layout

Initialise	wa4701_init
Configuration Functions	
Set Operating Mode	wa4701_setOperatingMode
Configure Device	wa4701_configDevice
Configure BC Functions	
Setup BC Parameters	wa4701_setupBCParameters
Setup BC Message Parameters	wa4701_setupBCMsgParams
Setup BC Frame Parameters	wa4701_setupBCFrameParams
Setup BC Error Injection Params	wa4701_setupBCMsgErrorInj
Setup BC Msg Timing Parameters	wa4701_setupBCMsgTiming
Setup BC Cycle Message Sequence	wa4701_setupBCCycleMsgSequenc
Set Msg/Frame Params to Default	wa4701_setBCMsgOrFrameParamsT
Setup BC Message Dynamic Tag	wa4701_setupBCMsgDTAG
Configure RT Functions	
Setup RT Parameters	wa4701_setupRTParams
Enable/Disable RT	wa4701_enableDisableRT
Setup RT Subaddress Parameters	wa4701_setupRTSubAddressParam
Setup RT Low Speed Err Inject	wa4701_setupRTLSErrInj
Setup RT High Speed Err Inject	wa4701_setupRTHSErrInj
Set RT Parameters to Default	wa4701_setRTParamsToDefaults
Set RT SRQ	wa4701_setRTSrq
Setup RT Timing Parameters	wa4701_setupRTTimingParams
Configure CM Functions	
Setup CM Start Sequence	wa4701_setupCMStartSeqArmTrig
Setup CM Stop Sequence	wa4701_setupCMStopSeqArmTrig
Setup Scapture/Window Params	wa4701_setupScapAndWindParams
Setup CM Delay/Post Trig Count	wa4701_setCMDelayAndPostTrigC
Setup CM External Trigger	wa4701_setupExternalTrigger
Action/Status Functions	
Query Instrument Status	wa4701_queryInstrStatus
BC Functions	
Start/Stop BC	wa4701_startStopBC
CM Functions	
Start/Stop CM Acquisition	wa4701_startStopCMAcq
Data Functions	
BC Functions	
Load BC Message Data	wa4701_loadBCMsgData
Read BC Message Data	wa4701_readBCMsgData
RT Functions	
Load RT Data	wa4701_loadRTData
Read RT Data	wa4701_readRTData
CM Functions	
Read CM Stack Data	wa4701_readCMStackData

Application Functions

Config And Transmit BC Messages	wa4701_configAndTransmitBCMsg
Config And Enable RT	wa4701_configureAndEnableRT
Config And Acquire CM Messages	wa4701_configCMAndAcquireData

Utility Functions

Reset	wa4701_reset
Self Test	wa4701_selfTest
Module Identification Query	wa4701_identificationQuery
Query System Error	wa4701_querySystemError
Scan Chassis for IIB-3910-VXIs	wa4701_scanChassis
Error Message	wa4701_errorMessage
Clear Status Register	wa4701_clearStatusRegister
Query Device Status Register	wa4701_queryDeviceStatusReg
Set Status Mask	wa4701_setStatusMask
Query BC Message Parameters	wa4701_queryBCMsgParams
Query BC Frame Parameters	wa4701_queryBCFrameParams
Query BC Message Error	wa4701_queryBCMsgError
Query RT Parameters	wa4701_queryRTParams
Query RT Low Speed Err Inject	wa4701_queryRTLSError
Query RT High Speed Err Inject	wa4701_queryRTHSError
Query CM Delay/Post Trig Count	wa4701_qryCMDelayAndPostTrigC
Query CM Start Sequence	wa4701_queryCMStartSeqArmTrig
Query CM Stop Sequence	wa4701_queryCMStopSeqArmTrig
Query CM Stack	wa4701_queryCMStack
Close	wa4701_closeInstrSession

wa4701_clearStatusRegister

Function:

ViStatus wa4701_clearStatusRegister (ViSession instrHndl);

Panel Name: Clear Status Register

Purpose:

This function clears the status reporting data structures. The effects of running this function are outlined below.

- (1) The standard event status register is cleared to zero
- (2) The operation event status register is cleared to zero
- (3) The error queue is cleared of all entries
- (4) The status byte (except for the MAV bit) is cleared to zero
- (5) The operation pending states are cleared to zero

Parameter List

instrHndl

Variable Type: ViSession

Control Name: instrHndl

Description: A unique session to the instrument.
This parameter gets its value from the Initialise function.

Variable Type: ViSession

Default Value -

Return Value

Control Name: status.

Description: Displays the results of the function call.

Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_closeInstrSession

Function:

ViStatus wa4701_closeInstrSession (ViSession instrHndl);

Panel Name: Close

Purpose:

This function closes the active session with the instrument and de-allocates the system resources.

Parameter List

instrHndl

Variable Type: ViSession

Control Name: instrHndl.

Description: A unique session handle to the instrument.
This parameter gets its value from the Initialise function.

Variable Type: ViSession

Valid Range: -

Default Value: -

Return Value

Control Name: status.

Description: Displays the results of the function call.

Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_configAndTransmitBCMsgs

Function:

void wa4701_configAndTransmitBCMsgs (void);

Panel Name: Config And Transmit BC Messages

Purpose:

This function has too many parameters to fit into the screen. This function is not intended to be called from an application. However, to run this function you may have to call it from an application code base. The purpose of this function is to demonstrate to the application developer how to group individual functions of this driver to achieve some application specific task. To understand more about this function please have a close look at the source code for this function.

wa4701_configCMAndAcquireData

Function:

void wa4701_configCMAndAcquireData (void);

Panel Name: Config And Acquire CM Messages

Purpose:

This function has too many parameters to fit into the screen. This function is not intended to be called from an application. However, to run this function you may have to call it from an application code base. The purpose of this function is to demonstrate to the application developer how to group individual functions of this driver to achieve some application specific task. To understand more about this function please have a close look at the source code for this function.

wa4701_configDevice

Function:

ViStatus wa4701_configDevice (ViSession instrHndl, ViReal64 voltage, ViInt16 typeOfCoupling, ViInt16 byteOrder);

Panel Name: Configure Device

Purpose:

This function sets the amplitude and coupling of the primary and secondary low speed buses. It also sets the byte order of the instrument as either Intel or Motorola byte order,

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument.
This parameter gets its value from the Initialise function.
Valid Range: -
Default Value: -

voltage

Variable Type: ViReal64
Control Name: voltage
Description: This parameter controls the amplitude of the low speed bus.
This amplitude is applied on both primary and secondary buses.
Variable Type: ViReal64.
Valid Range: wa4701_SET_VOLT_MIN (0.0Volts)
wa4701_SET_VOLT_MAX (20.0 Volts)
Default Value: wa4701_SET_VOLT_MIN

typeOfCoupling

Variable Type: ViInt16
Control Name: typeOfCoupling
Description: This parameter controls the coupling of the low speed bus.
Both the primary and the secondary low speed buses are switched together.
Variable Type: ViInt16.
Valid Range: wa4701_SET_DIR_COUP(Direct Coupling)
wa4701_SET_TRANS_COUP(Transformmer Coupling)
Default Value: wa4701_SET_DIR_COUP

byteOrder

Variable Type: ViInt16
Control Name: byteOrder
Description: This parameter specifies whether the byte order is to be set to Intel or Motorola byte order.
Variable Type: ViInt16.
Valid Range: wa4701_SET_ORDER_INT (Intel)
wa4701_SET_ORDER_MOT (Motorola)
Default Value: wa4701_SET_ORDER_MOT

Return Value

Control Name: status.

Description: Displays the results of the function call.

Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_configureAndEnableRT

Function:

void wa4701_configureAndEnableRT (void);

Panel Name: Config And Enable RT

Purpose:

This function has too many parameters to fit into the screen. This function is not intended to be called from an application. However, to run this function you may have to call it from an application code base. The purpose of this function is to demonstrate to the application developer how to group individual functions of this driver to achieve some application specific task. To understand more about this function please have a close look at the source code for this function.

wa4701_enableDisableRT

Function:

ViStatus wa4701_enableDisableRT (ViSession instrHndl, ViInt16 RTNumber, ViInt16 stateFlag, ViInt16 lsOrHsFlag, ViInt16 lsBus, ViInt16 hsBus);

Panel Name: Enable/Disable RT

Purpose:

This function is used to enable or disable the specified RT. It controls whether the RT is to be simulated by the IIB-3910-VXI or not. This function is effective in both the BC and MRT modes. In the MRT mode, an RT, which is 'OFF', will monitor all data to and from the external RT and store the data.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument.
Valid Range: -
Default Value: -

RTNumber

Variable Type: ViInt16
Control Name: RTNumber
Description: This parameter specifies the RT Number of the RT that is to be enabled or disabled depending on the value of the other input parameters.
Variable Type: ViInt16.
Valid Range: wa4701_RT_NUMBER_MIN (0)
wa4701_RT_NUMBER_MAX (31)
Default: -

stateFlag

Variable Type: ViInt16
Control Name: stateFlag
Description: : This parameter enables or disables RT simulation.
Variable Type: ViInt16.
Valid Range: wa4701_RT_SIMULATE(1055 / Simulate RT)
wa4701_RT_DISABLE (1056 / Don't Simulate RT)
Default: wa4701_RT_SIMULATE

lsOrHsFlag

Variable Type: ViInt16
Control Name: lsOrHsFlag
Description: : This parameter is used to specify enabling or disabling of the specified RT's low speed and high speed transmitters.
Variable Type: ViInt16.
Valid Range: wa4701_RT_LS (1074 / Low Speed Transmitter)
wa4701_RT_HS (1075 / High Speed Transmitter)
Default: wa4701_RT_LS

Note: When the input to this parameter is to enable the high speed transmitters, the low speed transmitters are also enabled according to the 'lsBus' parameter. However, when the input to this parameter is to enable the low speed transmitters, only the low speed transmitters are enabled depending on the 'lsBus' parameter.

lsBus

Variable Type: ViInt16
Control Name: lsBus
Description: This parameter selects the low speed bus whose transmitters are enabled or disabled.
Variable Type: ViInt16.
Valid Range: wa4701_RT_ENAB_NONE (1250 / None)
wa4701_RT_ENAB_PRI(1251 / Primary Low Speed Bus)
wa4701_RT_ENAB_SEC(1252 / Secondary Low Speed Bus)
wa4701_RT_ENAB_BOTH (1253 / Both Buses)
Default: wa4701_RT_ENAB_PRI.

hsBus

Variable Type: ViInt16
Control Name: hsBus
Description: This parameter selects the high speed bus whose transmitters are to be enabled or disabled.
Variable Type: ViInt16.
Valid Range: wa4701_RT_ENAB_NONE (1250 / None)
wa4701_RT_ENAB_PRI(1251 / Primary Low Speed Bus)
wa4701_RT_ENAB_SEC(1252 / Secondary Low Speed Bus)
wa4701_RT_ENAB_BOTH (1253 / Both Buses)
Default: wa4701_RT_ENAB_PRI.

Return Value

Control Name: status.
Description: Displays the results of the function call.
Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_errorMessage

Function:

ViStatus wa4701_errorMessage (ViSession instrHndl, ViStatus errorCode, ViChar errorMessage[]);

Panel Name: Error Message

Purpose:

This function takes in an error code as the input and returns the error message corresponding to it in a user readable format. All the driver functions return an error status regarding whether they have executed successfully or not. This errorCode is in Integer (Hex) format and often difficult to interpret. The same error status can be input to this function so that a corresponding user readable error string parameter gets its value from the Initialise function.

Parameter List

Variable Type: ViSession

Default Value: -

errorCode

Variable Type: ViStatus

Control Name: errorCode

Description: This parameter takes in the error code input returned by the driver functions for which the corresponding error string is to be output.

Variable Type: ViStatus

Default Value: -

errorMessage

Variable Type: ViChar []

Control Name: errorMessage

Description: This output parameter returns the error string corresponding to the input error code in a user readable format.

Variable Type: ViChar []

Default Value: -

Note: It must be ensured that this buffer is large enough to hold the output error string.

Return Value

Control Name: status.

Description: Displays the results of the function call.

Variable Type: ViStatus

Valid Range: VI_SUCCESS (0)

wa4701_identificationQuery

Function:

ViStatus wa4701_identificationQuery (ViSession instrHndl, ViChar instrID[]);

Panel Name: Module Identification Query

Purpose:

This function is used to query the instrument to obtain identification information as given below.

- (1) The Manufacturer's name and the Model Code.
- (2) Serial Number of IIB-3910-VXI
- (3) The Revision Levels of the Control Firmware and the Protocol Management Units' Firmware.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument.
This parameter gets its value from the Initialise function.
Variable Type: ViSession
Valid Range: -
Default Value: -

instrID

Variable Type: ViChar []
Control Name: instrID
Description: This output parameter returns identification information about the SURETEST 4701. The valid output response of this function should be as specified below.

Westinghouse Shannon Ireland, Model 4701,ssss,x.y

where ssss is the serial number of IIB-3910-VXI and x.y is the firmware revision level. The `x' is the revision level of the control firmware and the `y' is the revision level of the protocol management unit's firmware.

Variable Type: ViChar [].
Note: *This parameter should be at least 75 bytes long.*

Return Value

Control Name: status.
Description: Displays the results of the function call.
Variable Type: ViStatus

Note: *Use wa4701_errorMessage() function to retrieve the error message corresponding to the error code returned by this function.*

wa4701_init

Function:

ViStatus wa4701_init (ViChar *instrDesc, ViInt16 idQuery, ViInt16 resetFlag, ViSession *instrHndl);

Panel Name: Initialise

Purpose:

This function is used to initialise the IIB-3910-VXI instrument by opening the Instrument for communication and get a session handle to the same. This function provides options to query the instrument's Manufacturer ID and Model Code to ensure that it matches that of the IIB-3910-VXI. It also provides an option to reset the instrument to Power-On Default values.

Parameter List

instrDesc

Variable Type: ViChar *
Control Name: instrDesc.
Description: This variable selects either the GPIB or VXI communication interface.
Variable Type: ViChar *.
Valid Range: -
Default: VXI?:INSTR

where '?' is the logical address of the instrument whose handle is desired.

Note: *The instrument descriptor has to specify the logical address of the instrument. For example, if the instrument is currently set to a logical address of 32 and an AT-MXI or an embedded controller is used as the interface, then the instrument descriptor is set as follows. "VXI:32:INSTR"*

idQuery

Variable Type: ViInt16
Control Name: idQuery
Description: Specifies if an ID Query is sent to the instrument during the initialisation procedure. If an identification query is requested, then the Manufacturer's ID and Model Code of the instrument is compared against that of the IIB-3910-VXI to ensure that the instrument handle that is output by this function is a valid session handle to the IIB-3910-VXI.
Variable Type: ViInt16
Valid Range: wa4701_ID_QUERY(Perform ID Query)
wa4701_NO_ID_QUERY (Don't perform ID Query)
Default Value: wa4701_ID_QUERY

Note: *If the value of this field is specified such that no identification query be performed, then the resulting instrument handle could be an invalid session handle. If the user is sure that the other input parameters to this function are correct, then probably, the ID Query need not be performed.*

resetFlag

Variable Type: ViInt16
Control Name: resetFlag
Description: Specifies if the instrument is to be reset to its power-on settings during the initialisation procedure. If the user does not wish to reset the instrument, then the value of this field can be input as wa4701_NO_RESET
Variable Type: ViInt16.
Valid Range: wa4701_RESET (Perform Soft Reset)
wa4701_NO_RESET(Don't perform soft reset)
Default Value: wa4701_RESET

instrHndl

Variable Type: ViSession (passed by reference)
Control Name: instrHndl.
Description: Contains a unique session handle to a session with the instrument.
All other driver functions expect this input in order to be executed.
Variable Type: ViSession *
Valid Range: -
Default: -

Return Value

Control Name: status.
Description: Displays the results of the function call.
Variable Type: ViStatus.

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_loadBCMsgData

Function:

ViStatus wa4701_loadBCMsgData (ViSession instrHndl, ViInt16 msgNum, ViInt32 transmitData[], ViInt16 numOfTransmitWords, ViInt16 singleOrBlockMode, ViInt16 offset);

Panel Name: Load BC Message Data

Purpose:

This function loads the transmit data words into the transmit buffer of the specified message. The function supports both Single and Block Modes for loading of data.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument.
This parameter gets its value from the Initialise function.
Variable Type: ViSession
Valid Range: -
Default Value: -

msgNum

Variable Type: ViInt16
Control Name: msgNum
Description: This parameter specifies the message number of the message with which the transmit data is to be associated.
Variable Type: ViInt16.
Valid Range: wa4701_BC_MSG_NUM_MIN (1) to maxNumOfBCMsgs

where `maxNumOfBCMsgs' is obtained by executing `wa4701_getTransParamsNVRAM()' function
Default Value: -

transmitData

Variable Type: ViInt32 []
Control Name: transmitData
Description: This parameter specifies the array of data words which are to be associated with the specified message. In case of Block Mode, the words in this parameter are converted to the Arbitrary Block Format. In case of Single Mode, the elements are sent to the instrument in ASCII format.
Variable Type: ViInt32 []
Valid Range: The individual data words of the data array must not cross the range.
wa4701_MSG_WORD_VAL_MIN (0) to
wa4701_MSG_WORD_VAL_MAX (65535)
Default Value: -

numOfTransmitWords

Variable Type: ViInt16
Control Name: numOfTransmitWords
Description: This parameter specifies the number of words present in the input array `transmitData`.
Variable Type: ViInt16.
Valid Range: wa4701_BC_MSG_MIN_NUM_OF_WORDS (1)
wa4701_BC_MSG_MAX_NUM_OF_WORDS (4096)
Default Value: wa4701_BC_MSG_MIN_NUM_OF_WORDS

Note:

- (1) *The number of transmit words defined in a message cannot exceed 32 for a low speed message*
- (2) *The number of transmit words defined in a message cannot exceed 4096 for a high speed message.*
- (3) *The number of words that can be defined for any specific message depend on the predefined size of the associated data buffer as defined in the `Set BC Message Parameters` function.*

singleOrBlockMode

Variable Type: ViInt16
Control Name: singleOrBlockMode
Description: This parameter specifies whether the data words to be loaded to the transmit buffer of the specified message should be loaded in Block Mode (Block Data transfer) or Single Mode (transfer data word by word).
Variable Type: ViInt16.
Valid Range: wa4701_BLOCK_MODE(2220 / Block Mode) or
wa4701_SINGLE_MODE (2221 / Single Mode)
Default Value: wa4701_BLOCK_MODE

offset

Variable Type: ViInt16
Control Name: offset
Description: This parameter specifies the offset in the transmit buffer of the specified message where the data in to be loaded in Single Mode. If the Block Mode is selected then this parameter is ignored.
Variable Type: ViInt16.
Valid Range: wa4701_SINGLE_MODE_OFFSET_MIN (0) to
wa4701_SINGLE_MODE_OFFSET_MAX (4096).
Default Value: -

Return Value

Control Name: status.
Description: Displays the results of the function call.
Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_loadRTData

Function:

ViStatus wa4701_loadRTData (ViSession instrHndl, ViInt16 RTNumber, ViInt16 lsOrHsFlag, ViInt16 uniqueOrGlobalFlag, ViInt16 subAddressOrHSID, ViInt32 transmitData[], ViInt16 numOfTransmitWords, ViInt16 singleOrBlockMode, ViInt16 offset);

Panel Name: Load RT Data

Purpose:

This function is used to load data words into the transmit buffer of the specified RT's unique or global low speed subaddress or high speed identifier. This function allows the data to be loaded in either single or block mode. If the Wrap Around facility of the specified RT is ON, then the transmit buffer and receive buffer are one and the same and the transmit data is loaded into one common buffer.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument.
This parameter gets its value from the Initialise function.
Variable Type: ViSession
Valid Range: -
Default Value: -

RTNumber

Variable Type: ViInt16
Control Name: RTNumber
Description: This parameter specifies the RT Number into whose transmit buffer, the specified data words are to be loaded.
Variable Type: ViInt16.
Valid Range: wa4701_RT_NUMBER_MIN (0)
wa4701_RT_NUMBER_MAX (31)
Default: -

lsOrHsFlag

Variable Type: ViInt16
Control Name: lsOrHsFlag
Description: This parameter is used to indicate whether the specified data words are to be loaded into the transmit buffer of the unique/global low speed subaddress (or) the unique/global high speed identifier of the specified RT
Variable Type: ViInt16.
Valid Range: Value Interpretation
wa4701_RT_LS (1074) Unique/Global Low Speed Subaddress
wa4701_RT_HS (1075) Unique/Global High Speed Identifier
Default Value: wa4701_RT_LS

uniqueOrGlobalFlag

Variable Type: ViInt16
Control Name: uniqueOrGlobalFlag
Description: This parameter is used to specify whether the specified data words are to be loaded into the transmit buffer of the unique or global low speed subaddress/high speed identifier.
Variable Type: ViInt16.
Valid Range: wa4701_CONFIGURE_FOR_UNIQUE (2205 / Unique)
wa4701_CONFIGURE_FOR_GLOBAL (2206 / Global)
Default: wa4701_CONFIGURE_FOR_UNIQUE

subAddressOrHSID

Variable Type: ViInt16
Control Name: subAddressOrHSID
Description: This parameter specifies the low speed subaddress or high speed identifier into whose transmit buffer, the specified data words are to be loaded.
Variable Type: ViInt16
Valid Range: For Low Speed Subaddress
wa4701_RT_SA_MIN (1) to
wa4701_RT_SA_MAX (30).

For High Speed Identifier
wa4701_RT_HSID_MIN (1) to
wa4701_RT_HSID_MAX (127).
Default Value: -

Note : *This parameter is ignored by the function if the `uniqueOrGlobalFlag` is set for global low speed subaddress/high speed identifier.*

transmitData

Variable Type: ViInt32 []
Control Name: transmitData
Description: This parameter specifies the data words that are to be loaded into the transmit buffer of the specified unique/global low speed subaddress/high speed identifier.
Variable Type: ViInt32 []
Valid Range: Care must be taken to ensure that the individual elements of the data array are within the range specified below.
wa4701_MSG_WORD_VAL_MIN(0) to
wa4701_MSG_WORD_VAL_MAX(65535).
Default Value: -

Note: *If the Wrap Around facility of the particular subaddress/high speed identifier is wrapped on, then the transmit and receive buffers are one and the same and the data gets loaded into one common buffer. Therefore, the user should exercise caution while loading data into the RT subaddress/high speed identifier.*

numOfTransmitWords

Variable Type: ViInt16
Control Name: numOfTransmitWords
Description: This parameter specifies the number of data words present in the transmit array which is input to this function.
Variable Type: ViInt16.
Valid Range: For low speed subaddress
wa4701_RT_LS_DATA_SIZE_MIN (1) to
wa4701_RT_LS_DATA_SIZE_MAX (32)

For high speed identifier
wa4701_RT_HS_DATA_SIZE_MIN (1)
wa4701_RT_HS_DATA_SIZE_MAX (4096)
Default Value: -

singleOrBlockMode

Variable Type: ViInt16
Control Name: singleOrBlockMode
Description: This parameter specifies whether the data words to be loaded to the transmit buffer of the specified RT should be loaded in Block Mode or Single Mode.
Variable Type: ViInt16.
Valid Range: wa4701_BLOCK_MODE (2220 / Block Mode) or
wa4701_SINGLE_MODE(2221 / Single Mode)
Default Value: wa4701_SINGLE_MODE

offset

Variable Type: ViInt16
Control Name: offset
Description: This parameter specifies the offset into the transmit buffer from where the transmit data is to be loaded. This parameter is ignored if the `singleOrBlockMode` is set for Block Mode.
Variable Type: ViInt16
Valid Range: The range for this parameter depends on whether the transfer is a low speed or high speed type.

Low Speed Transfer
wa4701_SDATA_LS_OFFSET_MIN (0)
wa4701_SDATA_LS_OFFSET_MAX.(32)

High Speed Transfer
wa4701_SDATA_HS_OFFSET_MIN (0)
wa4701_SDATA_HS_OFFSET_MAX (4096)
Default Value: -

Return Value

Control Name: status.
Description: :Displays the results of the function call.
Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_qryCMDelayAndPostTrigCount

Function:

ViStatus wa4701_qryCMDelayAndPostTrigCount (ViSession instrHndl, ViReal64 *delayseconds, ViInt32 *postTriggerCount);

Panel Name: Query CM Delay/Post Trigger Count

Purpose:

This function is used to query configuration information regarding the following

- (1) The delay between the transition states after the TRIG:STOP condition has been satisfied
- (2) The Post Trigger Count i.e. the maximum number of messages which the CM will capture after the TRIG:STOP condition is satisfied.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique handle to the instrument.
Valid Range: -
Default: -

delayseconds

Variable Type: ViReal64 (passed by reference)
Control Name: delay
Description: This output parameter returns the currently configured delay value between the specified event detector being satisfied and the subsequent downward transition. In other words, it specifies a time in seconds for traversal between two states.

Variable Type: ViReal64 *

Possible Output: wa4701_CONF_CM_TIME_MIN (0 seconds) to
wa4701_CONF_CM_TIME_MIN (1.288497E7 seconds)

Default Value: -

postTriggerCount

Variable Type: ViInt32 (passed by reference)
Control Name: postTriggerCount
Description: This output parameter returns the currently configured value for the maximum number of messages that the chronological monitor will capture after the TRIG:STOP is exited.

Variable Type: ViInt32 *

Possible Output: wa4701_CONF_CM_CNT_MIN (0) to
wa4701_CONF_CM_CNT_MAX (32768)

Default Value: -

Return Value

Control Name: status.
Description: Displays the results of the function call.
Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_queryBCFrameParams

Function:

ViStatus wa4701_queryBCFrameParams (ViSession instrHndl, ViInt16 frameNum, ViInt16 *numberOfCycles, ViInt16 *syncAtEndOfCycle, ViReal64 *cyclePeriodseconds);

Panel Name: Query BC Frame Parameters

Purpose:

This function queries the current configuration information regarding the following BC Frame related parameters

- (1) Number Of Cycles within the frame (NCYC)
- (2) The Period of cycle in the frame(PER)
- (3) The Synchronisation at end of each cycle within the specified frame (SYNC)

Note : *The Cycle Period (PER) should be considered valid only if the Synchronisation Type (SYNC) employed is currently set to Internal*

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument.
This parameter gets its value from the Initialise function.
Variable Type: ViSession
Valid Range: -
Default Value: -

frameNum

Variable Type: ViInt16
Control Name: frameNum
Description: This parameter specifies the frame whose current configuration parameters are to be queried.
Variable Type: ViInt16.
Valid Range: wa4701_BC_FRAME_NUM_MIN (1) to numOfBCFrames.
where `numOfBCFrames' is the output obtained from `wa4701_getTransParamsNVRAM()'
Default Value: -

numberOfCycles

Variable Type: ViInt16 (passed by reference)
Control Name: numberOfCycles
Description: This output parameter returns the current configuration value for the number of cycles within the specified frame.
Variable Type: ViInt16 *.
Valid Range: -
Default Value: -

syncAtEndOfCycle

Variable Type: ViInt16 (passed by reference)
Control Name: syncAtEndOfCycle
Description: This output parameter returns the current configuration value for the synchronisation to be applied at the end of each cycle within the specified frame.
Variable Type: ViInt16 *.
Possible Output: Value Interpretation
wa4701_BC_FRAME_SYNC_NONE (1235) None
wa4701_BC_FRAME_SYNC_INT(1236) Internal
wa4701_BC_FRAME_SYNC_EXT(1237) External
Default Value: -

cyclePeriodseconds

Variable Type: ViReal64 (passed by reference)
Control Name: : cyclePeriod.
Description: This output parameter indicates last successfully configured value for the Cycle Period of each cycle within the specified frame. If this parameter is yet to be configured, then the default value of 20 microseconds is reported.
Variable Type: ViReal64 *
Valid Range: -
Default Value: -

Note: *The output value should be interpreted to be in seconds. If the parameter indicates a value of 0.02, then it is to be interpreted as 0.02 seconds i.e. 20 microseconds.*

Return Value

Control Name: status.
Description: Displays the results of the function call.
Variable Type: ViStatus

Note: *Use wa4701_errorMessage() function to retrieve the error message corresponding to the error code returned by this function.*

wa4701_queryBCMsgError

Function:

ViStatus wa4701_queryBCMsgError (ViSession instrHndl, ViInt16 msgNum, ViInt16 lsOrHsBus, ViInt32 *errorValue, ViInt32 *dataStatusReport);

Panel Name: Query BC Message Error

Purpose:

This function queries the Error Event Status of the Low Speed Bus and/or the High Speed Bus. It also returns the value of the Data Status Report from the Data Descriptor Block

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument.
This parameter gets its value from the Initialise function.
Variable Type: ViSession
Valid Range: -
Default Value: -

msgNum

Variable Type: ViInt16
Control Name: msgNum
Description: This parameter specifies the message number of the message for which the Error Event Status of the low speed bus and/or the high speed bus and the Data Status Report from the DDB is to be read.
Variable Type: ViInt16.
Valid Range: wa4701_BC_MSG_NUM_MIN (1) to maxNumOfBCMsgs
where 'maxNumOfBCMsgs' is obtained by executing
'wa4701_getTransParamsNVRAM()' function
Default Value: -

lsOrHsBus

Variable Type: ViInt16
Control Name: lsOrHsBus
Description: This parameter specifies whether the Error Event Status is to be read for the low speed bus or the high speed bus, which is associated with the specified message (or) whether, the Error Event Status is to be read for both the Low Speed and High Speed Buses are to be read.
Variable Type: ViInt16
Valid Value: wa4701_BC_MSG_ERR_BUS_LS (1242) (Low Speed Bus)
wa4701_BC_MSG_ERR_BUS_HS (1243) (High Speed Bus)
wa4701_BC_MSG_ERR_BUS_BBUS (1244) (Both Buses)
Default Value: wa4701_BC_MSG_ERR_BUS_LS

errorValue

Variable Type: ViInt32 (passed by reference)

Control Name: errorValue

Description: This output parameter contains the value of the Error Event Status of the low speed or high speed bus associated with the specified message. If a value of Both Buses is specified to the input parameter `lsOrHsBus`, then the return value of this parameter is the ORed value of the Low Speed & High Speed Error Event Status.

Variable Type: ViInt32 *

Valid Values: -

Note: If in case both the low speed and the high speed error event status are requested, then the final value of this parameter is the ORed value of both the errors. The meaning of each bit has been specified below.

Low Speed Error Event Status

Weight	Bit#	Event
32768	15	RT responded on either wrong bus or both buses
16384	14	RT did not respond
8192	13	Status Word had wrong RT Address
4096	12	Any of:- Manchester Error, too many bits, too few bits, Parity Error, Word Count Error, RT responded later than 12 microseconds
2048	11	Invalid Sync

High Speed Error Event Status

Weight	Bit#	Event
64	6	HS Data Overlap Occurred
32	5	Word Count Error
16	4	FCS Error
8	3	Invalid or No End Delimiter
4	2	Invalid or No Start Delimiter
2	1	No HS Frame within RIOUT time
1	0	Manchester Error

dataStatusReport

Variable Type: ViInt32 (passed by reference)

Control Name: dataStatusReport

Description: This output parameter contains the value of the Data Status Report from the DDB (Data Descriptor Block). This is a `live` value, valid at any time the function was executed.

Variable Type: ViInt32 *

Valid Values: -

Return Value

Control Name: status.

Description: Displays the results of the function call.

Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_queryBCMsgParams

Function:

ViStatus wa4701_queryBCMsgParams (ViSession instrHndl, ViInt16 msgNum, ViInt16 *sourceRTNum, ViInt16 *sourceSubAddress, ViInt16 *destRTNum, ViInt16 *destSubAddress, ViInt16 *numOfWords, ViInt16 *lsOrHsBus, ViInt16 *typeOfLSBus, ViInt16 *typeOfHSBus, ViInt16 *commandFlag, ViInt16 *modeCommand, ViInt32 *modeCommandData);

Panel Name: Query BC Message

Purpose:

This function queries the current configuration of the specified message. The various parameters reported are the message source and destination, the number of words to be transmitted, the bus to be used and whether the command is a mode command or a STANAG command.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument.
This parameter gets its value from the Initialise function.
Variable Type: ViSession
Valid Range: -
Default Value: -

msgNum

Variable Type: ViInt16
Control Name: msgNum
Description: This parameter specifies the message number of the message, which is to be queried for its configuration parameters.
Variable Type: ViInt16.
Valid Range: wa4701_BC_MSG_NUM_MIN (1) to maxNumOfBCMsgs
where `maxNumOfBCMsgs' is obtained by executing
wa4701_getTransParamsNVRAM()' function
Default Value: -

sourceRTNum

Variable Type: ViInt16 (passed by reference)
Control Name: sourceRTNum
Description: This output parameter indicates the source RT number of the input message number.
Variable Type: ViInt16 *.
Valid Range: BC is source wa4701_BC_MSG_SOURCE_RT_MIN (-1).
RT is source wa4701_BC_MSG_SOURCE_RT_MIN + 1 (0) to
wa4701_BC_MSG_SOURCE_RT_MAX (30).
Default Value: -

sourceSubAddress

Variable Type: ViInt16 (passed by reference)
Control Name: sourceSubAddress
Description: : This output parameter indicates the source RT subaddress of the specified message number.
Variable Type: ViInt16 *
Valid Range: BC is source
wa4701_BC_MSG_LS_SOURCE_SA_MIN (0) or
wa4701_BC_MSG_HS_SOURCE_SA_MIN (0)

RT is source in LS Transfer
wa4701_BC_MSG_LS_SOURCE_SA_MIN + 1 (1)
wa4701_BC_MSG_LS_SOURCE_SA_MAX (30)

RT is source in HS Transfer
wa4701_BC_MSG_HS_SOURCE_SA_MIN + 1 (1) to
wa4701_BC_MSG_HS_SOURCE_SA_MAX.(127)
Default Value: -

destRTNum

Variable Type: ViInt16 (passed by reference)
Control Name: destRTNum
Description: This parameter specifies the destination RT number of the message being set up.
Variable Type: ViInt16 *.
Valid Range: BC is Destination
wa4701_BC_MSG_DEST_RT_MIN (-1)

RT is Destination
wa4701_BC_MSG_DEST_RT_MIN + 1 (0)
wa4701_BC_MSG_DEST_RT_MAX (31)
Default Value: -

Note: *If the destination RT Number is 31, then the message is of the 'broadcast' type.*

destSubAddress

Variable Type: ViInt16 (passed by reference)
Control Name: destSubAddress
Description: : This output parameter indicates the destination RT's subaddress of the specified message number.
Variable Type: ViInt16 *
Valid Range : BC is destination
wa4701_BC_MSG_LS_DEST_SA_MIN (0) or
wa4701_BC_MSG_HS_DEST_SA_MIN (0)

RT is destination in Low Speed Transfer
wa4701_BC_MSG_LS_DEST_SA_MIN + 1 (1)
wa4701_BC_MSG_LS_DEST_SA_MAX (30)

RT is destination in High Speed Transfer
wa4701_BC_MSG_HS_DEST_SA_MIN + 1 (1)
wa4701_BC_MSG_HS_DEST_SA_MAX (127)
Default Value: -

numOfWords

Variable Type: ViInt16 (passed by reference)
Control Name: numOfWords
Description: This output parameter returns the number of words that the message has been configured for.
Variable Type: ViInt16 *
Valid Range: -
Default Value: -

IsOrHsBus

Variable Type: ViInt16 (passed by reference)
Control Name: IsOrHsBus
Description: This output parameter indicates whether the message has been set-up for transfer on the low speed or high speed bus.
Variable Type: ViInt16 *.
Valid Range: Value Interpretation
wa4701_BC_MSG_BUS_LS (1085) Low Speed Bus
wa4701_BC_MSG_BUS_HS (1086) High Speed Bus
Default Value: -

typeOfLSBus

Variable Type: ViInt16 (passed by reference)
Control Name: typeOfLSBus
Description: This output parameter indicates whether the instrument has been configured for Low Speed Primary Or Secondary Bus.
Variable Type: ViInt16 *.
Valid Range: Value Interpretation
wa4701_BC_MSG_LS_BUS_PRI (1089) Primary Bus
wa4701_BC_MSG_LS_BUS_SEC (1090) Secondary Bus
Default Value: -

typeOfHSBus

Variable Type: ViInt16 (passed by reference)
Control Name: typeOfHSBus
Description: This output parameter returns the type of High Speed Bus used in case of High Speed Transfer.
Variable Type: ViInt16 *.
Valid Range: Value Interpretation
wa4701_BC_MSG_HS_BUS_NONE (1091) None
wa4701_BC_MSG_HS_BUS_PRI(1092) Primary Bus
wa4701_BC_MSG_HS_BUS_SEC(1093) Secondary Bus
Default: -

commandFlag

Variable Type: ViInt16 (passed by reference)

Control Name: : commandFlag

Description: This output parameter indicates whether the command word associated with the specified message is a mode command or a STANAG command.

Variable Type: ViInt16 *.

Valid Range: Value Interpretation

wa4701_BC_MSG_MCOM_WITH_DATA(1095) Mode Command with data

wa4701_BC_MSG_MCOM_WITHOUT_DATA (1096) Mode Comm. without data

wa4701_BC_MSG_NOT_MCOM(1097) STANAG Command

Default Value: -

modeCommand

Variable Type: ViInt16 (passed by reference)

Control Name: : modeCommand

Description: This output parameter indicates the mode command that the specified message number has been configured for. This parameter contains a valid output only if the `modeCommFlag' parameter indicates a mode command.

Variable Type: ViInt16 *.

Valid Range: For low speed mode command

wa4701_BC_MSG_MCOM_LS_NUM_MIN (0) to

wa4701_BC_MSG_MCOM_LS_NUM_MAX (31).

For high speed mode command

wa4701_BC_MSG_MCOM_HS_NUM_MIN (0) to

wa4701_BC_MSG_MCOM_HS_NUM_MAX (127).

Default Value: -

modeCommandData

Variable Type: ViInt32 (passed by reference)

Control Name: modeCommandData

Description: This output parameter specifies the data word that has been associated with the mode command. This parameter contains a valid value only if the `modeCommFlag' flag specifies a value of `wa4701_BC_MSG_MCOM_WITH_DATA'.

Variable Type: ViInt32 *

Valid Range: wa4701_BC_MSG_MCOM_DATA_MIN (0) to

wa4701_BC_MSG_MCOM_DATA_MAX (65535).

Default Value: -

Return Value

Control Name: status.

Description: Displays the results of the function call.

Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_queryCMStack

Function:

ViStatus wa4701_queryCMStack (ViSession instrHndl, ViInt16 *trigStopMsgNum, ViReal64 *trigStopTimeTag, ViInt16 *stackOldMsgNum, ViInt16 *stackRecentMsgNum);

Panel Name: Query CM Stack

Purpose:

This function returns the following information from the stack,

- (1) The number of the message where TRIGger:STOP condition was satisfied
- (2) Time tag (in units of 0.5uS) where TRIGger:STOP condition was satisfied.
- (3) The number of the oldest message in the stack.
- (4) The number of the most recent message in the stack.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique handle to the instrument.
This parameter gets its value from the Initialise function.
Valid Range: -
Default: -

trigStopMsgNum

Variable Type: ViInt16 (passed by reference)
Control Name: trigStopMsgNum
Description: This indicator displays the number of the message where TRIGger:STOP condition was satisfied.
Variable Type: ViInt16 *
Valid Range: -
Default: -

trigStopTimeTag

Variable Type: ViReal64 (passed by reference)
Control Name: trigStopTimeTag
Description: This indicator displays the time tag in units of 0.5uS as a 32 bit integer where the TRIGger:STOP condition was satisfied.
Variable Type: ViReal64 *
Valid Range: -
Default: -

stackOldMsgNum

Variable Type: ViInt16 (passed by reference)
Control Name: stackOldMsgNum
Description: This indicator displays the oldest message number in the stack.
Variable Type: ViInt16 *
Valid Range: -
Default: -

stackRecentMsgNum

Variable Type: ViInt16 (passed by reference)

Control Name: stackRecentMsgNum

Description: This indicator displays the most recent message number in the stack.

Variable Type: ViInt16 *

Valid Range: Name: status: -

Return Value

Control Name: status.

Description: Displays the results of the function call.

Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_queryCMStartSeqArmTrig

Function:

ViStatus wa4701_queryCMStartSeqArmTrig (ViSession instrHndl,
ViInt16 armOrTriggerFlag, ViInt16 *source, ViInt16 *slope);

Panel Name: Query CM Start Sequence

Purpose:

This function queries current configuration parameters of the Arm and Trigger Event Detection Layer of the Start Sequence of the Chronological Monitor. It returns information regarding the source and slope.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique handle to the instrument.
This parameter gets its value from the Initialise function.
Variable Type: ViSession
Valid Range: -
Default: -

armOrTriggerFlag

Variable Type: ViInt16
Control Name: armOrTriggerFlag.
Description: This parameter specifies whether the configuration information to be obtained is for the Arm or Trigger Event Detection Layer of the Start Sequence.
Variable Type: ViInt16
Valid Range: wa4701_QUERY_CM_ARM (1050 / Arm) or
wa4701_QUERY_CM_TRIGGER (1051 / Trigger).
Default Value: wa4701_QUERY_CM_ARM.

source

Variable Type: ViInt16 (passed by reference)
Control Name: source
Description: This output parameter returns the currently configured Source used by the specified event detection layer for the specified sequence.
Variable Type: ViInt16 *.
Valid Range: For Arm
wa4701_CONF_CM_SOURCE_BUS (Bus / 1020) or
wa4701_CONF_CM_SOURCE_IMM.(Immediate / 1023)

For Trigger
wa4701_CONF_CM_SOURCE_EXT (External / 1022) or
wa4701_CONF_CM_SOURCE_IMM (Immediate / 1023)
Default Value: -

slope

Variable Type: ViInt16 (passed by reference)
Control Name: slope
Description: This output parameter returns the currently configured slope of the output pulse generated when TRIG:START condition is satisfied.
Variable Type: ViInt16 *
Valid Range: wa4701_CONF_CM_SLOPE_NEG (1050)
wa4701_CONF_CM_SLOPE_POS (1051)
Default Value: wa4701_CONF_CM_SLOPE_NEG.

Note: *This parameter returns a valid value **only** if the query is being performed for the Trigger and the Source has been configured as **External**.*

Return Value

Control Name: status.
Description: Displays the results of the function call.
Variable Type: ViStatus

Note: *Use **wa4701_errorMessage()** function to retrieve the error message corresponding to the error code returned by this function.*

wa4701_queryCMStopSeqArmTrig

Function:

ViStatus wa4701_queryCMStopSeqArmTrig (ViSession instrHndl, ViInt16 armOrTriggerFlag, ViInt16 layerNum, ViInt16 wordNum, ViInt16 lsOrHsFlag, ViInt16 *source, ViInt16 *maskFormat, ViChar digitalMask[], ViChar wordType[], ViChar bus[], ViInt16 *errorType, ViChar mode[]);

Panel Name: Query CM Stop Sequence

Purpose:

This function is used to query the configuration parameters of the Arm and Trigger Event Detection Layers of the Stop Sequence of the Chronological Monitor.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique handle to the instrument.
This parameter gets its value from the Initialise function.
Variable Type: ViSession
Valid Range: -
Default: -

armOrTriggerFlag

Variable Type: ViInt16
Control Name: armOrTriggerFlag.
Description: This parameter specifies whether the configuration information regarding the Arm or Trigger Event Detection Layer is to be queried.
Variable Type: ViInt16.
Valid Range: wa4701_QUERY_CM_ARM (1050 / Arm)
wa4701_QUERY_CM_TRIGGER (1051 / Trigger).
Default Value: wa4701_QUERY_CM_ARM.

layerNum

Variable Type: ViInt16
Control Name: layerNum.
Description: This parameter specifies the layer number of the Arm for which the configuration information is to be queried. In the case of Trigger, this parameter is ignored.
Variable Type: ViInt16.
Valid Range: wa4701_QUERY_CM_LAYER1 (1 / Layer 1)
wa4701_QUERY_CM_LAYER2 (2 / Layer 2)
wa4701_QUERY_CM_LAYER3 (3 / Layer 3)
Default Value: wa4701_QUERY_CM_LAYER1.

wordNum

Variable Type: ViInt16
Control Name: wordNum.
Description: This parameter specifies the digital word comparator number used by an event detector layer in the stop sequence.
Variable Type: ViInt16.
Valid Range: wa4701_QUERY_CM_WORD1 (1 / Word 1)
wa4701_QUERY_CM_WORD2 (2 / Word 2)
wa4701_QUERY_CM_WORD3 (3 / Word 3)
wa4701_QUERY_CM_WORD4 (4 / Word 4).
Default Value: wa4701_QUERY_CM_WORD1.

lsOrHsFlag

Variable Type: ViInt16
Control Name: lsOrHsFlag
Description: This parameter is a flag which specifies whether the configuration information to be queried is for a Low speed word or a High speed word.
Variable Type: ViInt16.
Valid Range: wa4701_QUERY_CM_LS(1076 / Low Speed)
wa4701_QUERY_CM_HS(1075 / High Speed)
Default Value: wa4701_QUERY_CM_LS

source

Variable Type: ViInt16 (passed by reference)
Control Name: source
Description: This output parameter returns the currently selected Signal Source used by the specified event detection layer for the specified sequence.
Variable Type: ViInt16 *
Valid Range: For Arm
wa4701_CONF_CM_SOURCE_INT (Internal / 1021) or
wa4701_CONF_CM_SOURCE_IMM.(Immediate / 1023)

For Trigger
wa4701_CONF_CM_SOURCE_INT (Internal / 1021) or
wa4701_CONF_CM_SOURCE_IMM (Immediate / 1023)
Default Value: -

maskFormat

Variable Type: ViInt16 (passed by reference)
Control Name: maskFormat
Description: This output parameter returns the Mask Format used to express the `digital mask' which is output by this function. The value of the digital mask has to be interpreted accordingly. For example, if this parameter returns a Binary Mask Format, then the value of the digital mask should be interpreted as a binary value.
Variable Type: ViInt16 *
Possible Output:

Value	Interpretation
wa4701_CONF_CM_MASK_DEC(1030)	Decimal Format
wa4701_CONF_CM_MASK_OCT(1031)	Octal Format
wa4701_CONF_CM_MASK_BIN(1032)	Binary Format
wa4701_CONF_CM_MASK_HEX(1033)	Hexadecimal Format
wa4701_CONF_CM_MASK_1553WORD (1034)	1553 Word Format

digitalMask

Variable Type: ViChar []
Control Name: digitalMask
Description: This output parameter returns the digital mask for the LS word. Depending on the format reported by the `maskFormat` field, the output can be interpreted to be in binary, octal, hexadecimal or the MIL-STD-1553 word.
Variable Type: ViChar [].
Valid Range: -
Default Value: -

wordType

Variable Type: ViChar []
Control Name: wordType
Description: This output parameter returns the type of the LS-word (Command, Status, RT-RT or Data) that has been defined for the specified event detector in the STOP sequence
Variable Type: ViChar []
Possible Output:

Value	Interpretation
X	Don't Care
COMMAND	Command Word
DATA	Data Word
RT-RT	RT-RT Word
STATUS	Status Word

Default Value: -

bus

Variable Type: ViChar []
Control Name: bus
Description: This output parameter returns information regarding which bus is to be searched for the specified digital comparator.
Variable Type: ViChar []
Valid Range:

Value	Interpretation
X	Don't Care
PRIMARY	Primary Bus
SECONDARY	Secondary Bus
BOTH	Both Buses

Default Value: -

errorType

Variable Type: ViInt16 (passed by reference)

Control Name: errorType.

Description: This output parameter returns the error type of the error(s) in the HS word or LS word depending on lsOrHsFlag. Each bit represents one error. A '1' means that error condition has been enabled. A '0' means it has is disabled.

For LS word the bit pattern is as follows.

BitError	Type
0	PARITY error
1	SYNC error
2	SWORD error
3	LWORD error
4	MANCHESTER error
5	NRESPONSE error
6	WCOUNT error
7	TADDRESS error.

For HS word the bit pattern is as follows.

BitError	Type
0	FCSEQUENCE error
1	EDELIMTER error
2	NRESPONSE error
3	WCOUNT error.

Variable Type: ViInt16.

Valid Range: For LS word :

wa4701_CONF_CM_LS_ERR_VAL_MIN (0)
wa4701_CONF_CM_LS_ERR_VAL_MAX (0xFF)

wa4701_LS_ERR_NONE (0)
wa4701_LS_ERR_PARITY (1)
wa4701_LS_ERR_SYNC (2)
wa4701_LS_ERR_SWORD(4)
wa4701_LS_ERR_LWORD(8)
wa4701_LS_ERR_MANC (16)
wa4701_LS_ERR_NRES (32)
wa4701_LS_ERR_WCO(64)
wa4701_LS_ERR_TADD (128)

For HS word :

wa4701_CONF_CM_HS_ERR_VAL_MIN (0)
wa4701_CONF_CM_HS_ERR_VAL_MAX (0xF)

wa4701_HS_ERR_NONE 0
wa4701_HS_ERR_FCS 1
wa4701_HS_ERR_EDEL 2
wa4701_HS_ERR_NRESP4
wa4701_HS_ERR_WCOUNT 8

Default Value: -

Note: *The above mentioned information should be used to interpret the bit-mapped error type to the corresponding error values. For example, in the LS Mode, if this parameter returned a value of 2, then the error condition is PARITY.*

mode

Variable Type: ViChar []

Control Name: mode

Description: This output parameter returns the mode in which the event detector condition is checked i.e. whether the normal or inverted logic is being used for checking event detection layers.

Variable Type: ViChar []

Possible Output: NORMAL (For Normal Mode)
INVERTED (For Inverted Mode)

Default Value: -

Return Value

Control Name: status.

Description: Displays the results of the function call.

Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_queryDeviceStatusReg

Function:

ViStatus wa4701_queryDeviceStatusReg (ViSession instrHndl, ViInt16 whichStatusReg, ViInt16 *valueStatusReg);

Panel Name: Query Device Status Register

Purpose:

This function reads any of the below mentioned registers within the IEEE488.2 & SCPI status model. and returns the current value of the same. This function supports reads to the following status registers

- (1) Status Byte Register
- (2) Event Status Register
- (3) Operation Condition Status Register
- (4) Operation Event Status Register

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique handle to the instrument.
This parameter gets its value from the Initialise function.
Variable Type: ViSession
Valid Range: -
Default: -

whichStatusReg

Variable Type: ViInt16
Control Name: whichStatusReg
Description: This parameter specifies the particular status register to be read.
Variable Type: ViInt16
Valid Values:

<i>Value</i>	<i>Register</i>
wa4701_STATUS_BYTE (1007)	Status Byte Register
wa4701_EVENT_REG (1008)	Event Status Register
wa4701_OPER_COND (1009)	Operation Condition Status Register
wa4701_OPER_EVENT(1010)	Operation Event Status Register

Default Range: wa4701_STATUS_BYTE

valueStatusReg

Variable Type: ViInt16 (passed by reference)
Control Name: valueStatusReg
Description: This output parameter returns the value of the selected register within the IEEE488.2 & SCPI status model.
Variable Type: ViInt16 *
Valid Range: 0 to 65535

Return Value

Control Name: status.
Description: Displays the results of the function call.
Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_queryInstrStatus

Function:

ViStatus wa4701_queryInstrStatus (ViSession instrHndl, ViInt16 *activeDevice);

Panel Name: Query Instrument Status

Purpose:

This function queries the IIB-3910-VXI and returns information regarding whether :-

- (1) The BC logical instrument is actively transmitting and receiving data over the data buses.
- (2) The MRT logical instrument is actively running, for example, responding to the bus
- (3) The CM logical instrument is actively monitoring and recording the bus traffic

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique handle to the instrument.
This parameter gets its value from the Initialise function.
Variable Type: ViSession
Valid Range: -
Default: -

activeDevice

Variable Type: ViInt16 (passed by reference)
Control Name: activeDevice
Description: This output parameter returns information regarding whether
(1) The BC logical instrument is actively transmitting and receiving data over the data buses.(or)
(2) The MRT logical instrument is actively running, for example, responding to the bus (or)
(3) The CM logical instrument is actively monitoring and recording the bus traffic.
Variable Type: ViInt16 *
Valid Range: wa4701_BC_ACTIVE(2000 / BC Active)
wa4701_MRT_ACTIVE (2001 / MRT Active)
wa4701_CM_ACTIVE(2002 / CM Active)

Return Value

Control Name: status.
Description: Displays the results of the function call.
Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_queryRTHSError

Function:

ViStatus wa4701_queryRTHSError (ViSession instrHndl, ViInt16 RTNumber, ViInt16 uniqueOrGlobalHSIDFlag, ViInt16 hsIdentifier, ViInt16 *wordCountError, ViInt16 *fcsError, ViInt16 *mancError, ViInt32 *mancErrBitPos, ViInt16 *mancErrLevel, ViChar mancErrField[]);

Panel Name: Query RT High Speed Err Inject

Purpose:

This function is used to query the errors currently injected into the HS Frame associated with the specified unique or global high speed identifier.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument.
This parameter gets its value from the Initialise function.
Variable Type: ViSession
Valid Range: -
Default Value: -

RTNumber

Variable Type: ViInt16
Control Name: RTNumber
Description: This parameter specifies the RT Number of the RT which has to be queried for the current configuration information regarding the High Speed Error Injection employed for the specified unique/global High Speed Identifier.
Variable Type: ViInt16.
Valid Range: wa4701_RT_NUMBER_MIN (0)
wa4701_RT_NUMBER_MAX (31)
Default: wa4701_RT_NUMBER_MIN (0)

uniqueOrGlobalHSIDFlag

Variable Type: ViInt16
Control Name: uniqueOrGlobalHSIDFlag
Description: This parameter is used to specify whether the high speed identifier for which errors are to be queried is a unique or global high speed identifier.
Variable Type: ViInt16.
Valid Range: wa4701_UNIQUE_HSID_ERR_QRY (2237 / Unique) or
wa4701_GLOBAL_HSID_ERR_QRY (2238 / Global).
Default: wa4701_UNIQUE_HSID_ERR_QRY

hsIdentifier

Variable Type: ViInt16
Control Name: hsIdentifier.
Description: This parameter specifies the High Speed Identifier of the RT whose associated HS Frame is to be queried for error injection.
Variable Type: ViInt16.
Valid Range: wa4701_RT_HSID_NUM_MIN (1)
wa4701_RT_HSID_NUM_MAX (127)
Default: wa4701_RT_HSID_NUM_MIN.

Note: *If the `uniqueOrGlobal' flag is set to query error injection for the global high speed identifier, then this parameter is ignored.*

wordCountError

Variable Type: ViInt16 (passed by reference)
Control Name: wordCountError
Description: This output parameter returns the Word Count Deviation from a legal transmission of the HS Frame associated with the unique/global high speed identifier.
Variable Type: ViInt16 *
Possible Output: wa4701_RT_HS_ERR_WCO_MIN (-1) to
wa4701_RT_HS_ERR_WCO_MAX (1)
Default Value: -

fcsError

Variable Type: ViInt16 (passed by reference)
Control Name: fcsError
Description: This output parameter returns the Frame Check Sequence Error associated with the HS Frame of the specified unique/global high speed identifier.
Variable Type: ViInt16 *
Possible Output: 0 (No FCS Error) or 1 (FCS Error)
Default Value: -

mancError

Variable Type: ViInt16 (passed by reference)
Control Name: mancError
Description: This output parameter returns information regarding whether the Manchester Error has been enabled or disabled in the HS Frame associated with the unique/global high speed identifier.
Variable Type: ViInt16 *
Possible Output: : 0 (Manchester Error Disabled) or 1 (Manchester Error Enabled)
Default Value: -

mancErrBitPos

Variable Type: ViInt32 (passed by reference)
Control Name: mancErrBitPos
Description: This output parameter returns information regarding the bit position of the bit in the HS Frame associated with the specified unique/global high speed identifier where the Manchester Encoding error has been injected.
Variable Type: ViInt32 *
Possible Output: 0 (Bit 0) to 65535 (Bit 65535)
Default Value: -

mancErrLevel

Variable Type: ViInt16 (passed by reference)
Control Name: mancErrLevel
Description: This output parameter returns information regarding the Level of the Manchester Error in the HS Frame associated with the specified unique/global high speed identifier.
Variable Type: ViInt16 *
Possible Output: 0 (Low Level) or 1 (High Level)
Default Value: -

mancErrField

Variable Type: ViChar []
Control Name: mancErrField
Description: This output parameter returns information regarding the field from which the returned bit position is offset.
Variable Type: ViChar []
Possible Output: PREAMBLE (Preamble)
 SDELIMITER (Start Delimiter)
 FCADDRESS (Frame Check/Physical Address)
 INFO (Info)
 FCSEQUENCE (Frame Check Sequence)
 EDELIMITER (End Delimiter)
Default Value: -

Return Value

Control Name: status.
Description: Displays the results of the function call.
Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_queryRTLSError

Function:

ViStatus wa4701_queryRTLSError (ViSession instrHndl, ViInt16 RTNumber, ViInt16 uniqueOrGlobalFlag, ViInt16 subAddress, ViChar errorType[], ViInt16 *errorParam1, ViInt16 *errorParam2);

Panel Name: Query RT Low Speed Err Inject

Purpose:

This function is used to query the current configuration information regarding the error injection employed for the specified unique/global low speed subaddress of the input RT number.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument.
This parameter gets its value from the Initialise function.
Valid Range: -
Default Value: -

RTNumber

Variable Type: ViInt16
Control Name: RTNumber
Description: This parameter specifies the RT Number of the RT which has to be queried for its current error injection configuration for the specified unique/global low speed subaddress.
Variable Type: ViInt16.
Valid Range: wa4701_RT_NUMBER_MIN (0)
wa4701_RT_NUMBER_MAX (31)
Default: wa4701_RT_NUMBER_MIN (0)

uniqueOrGlobalFlag

Variable Type: ViInt16
Control Name: uniqueOrGlobal
Description: This parameter is used to specify whether the subaddress for which errors are to be queried is a unique or global low speed subaddress.
Variable Type: ViInt16.
Valid Range: wa4701_UNIQUE_SA_LS_ERR_QRY (2235 / Unique) or
wa4701_GLOBAL_SA_LS_ERR_QRY (2236 / Global)
Default: wa4701_UNIQUE_SA_LS_ERR_QRY.

subAddress

Variable Type: ViInt16
Control Name: subAddress
Description: This parameter specifies the Low Speed Subaddress for which the current error injection configuration is to be queried.
Variable Type: ViInt16.
Valid Range: wa4701_RT_SA_MIN (1)
wa4701_RT_SA_MAX (30)
Default: wa4701_RT_SA_MIN (1)

Note: If the 'uniqueOrGlobal' flag specifies a Global Subaddress, then the value of this parameter is ignored by the function. The range check is done only for unique low speed subaddresses.

errorType

Variable Type: ViChar []
Control Name: errorType
Description: This output parameter returns the type of error that has been injected into the low speed words of the specified low speed subaddress.
Variable Type: ViChar []
Possible Output:

Value	Interpretation
NONE	None
PARITY	Parity
SYNCHRO	Synchro
MANCHESTER	Manchester
WCOUNT	Word Count
WLENGTH	Wrong Length
WBUS	Wrong Bus
BBUS	Both Buses

Default: -

errorParam1

Variable Type: ViInt16 (passed by reference)
Control Name: errorParam1
Description: This output parameter returns the additional parameter configuration information regarding the current low speed error injection. This parameter could contain different values depending on the type of error previously configured.
Variable Type: ViInt16 *
Valid Values:

<i>Error Type</i>	<i>Interpretation</i>
None	This parameter can be ignored.
Parity	wa4701_RT_ERR_WNUM_MIN (0) to wa4701_RT_ERR_WNUM_MAX (32).
Synchro	wa4701_RT_ERR_WNUM_MIN (0) to wa4701_RT_ERR_WNUM_MAX (32).
Manchester	wa4701_RT_ERR_WNUM_MIN (0) to wa4701_RT_ERR_WNUM_MAX (32).
Word Count	wa4701_RT_ERR_WRD_CNT_MIN (-63) to wa4701_RT_ERR_WRD_CNT_MAX (63).
Word Length	wa4701_RT_ERR_WNUM_MIN (0) to wa4701_RT_ERR_WNUM_MAX (32).
Wrong Bus	This parameter can be ignored.
Both Buses	This parameter can be ignored.

errorParam2

Variable Type: ViInt16 (passed by reference)
Control Name: errorParam2
Description: : This output parameter returns the additional parameter configuration information regarding the current low speed error injection. This parameter could contain different values depending on the type of error previously configured.

Variable Type: ViInt16 *

Valid Values:	Error Type	Interpretation
	None	This parameter can be ignored.
	Parity	This parameter can be ignored.
	Synchro	wa4701_RT_ERR_SYNC_PATTERN_MIN (1) to wa4701_RT_ERR_SYNC_PATTERN_MAX (63).
	Manchester	wa4701_RT_ERR_MANCH_BIT_POS_MIN (0) to wa4701_RT_ERR_MANCH_BIT_POS_MAX (31).
	Word Count	This parameter can be ignored.
	Word Length	wa4701_RT_ERR_WRD_LEN_MIN (-8) to wa4701_RT_ERR_WRD_LEN_MAX (8).
	Wrong Bus	This parameter can be ignored.
	Both Buses	This parameter can be ignored.

Return Value

Control Name: status.
Description: :Displays the results of the function call.
Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_queryRTParams

Function:

ViStatus wa4701_queryRTParams (ViSession instrHndl, ViInt16 RTNumber, ViInt32 *lsStatusWord, ViInt32 *hsStatusWord, ViInt32 *lsBitWord, ViInt32 *hsBitWord, ViInt32 *actionWord, ViInt32 *lsVectorWord);

Panel Name: Query RT Parameters

Purpose:

This function returns the below mentioned current configuration information regarding the specified RT.

- (1) Low Speed Status Word
- (2) High Speed Status Word
- (3) Low Speed BIT (Built In Test) Word
- (4) High Speed BIT (Built In Test) Word
- (5) Action Word
- (6) Vector Word

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: : A unique session handle to the instrument.
This parameter gets its value from the Initialise function.
Variable Type: ViSession
Valid Range: -
Default Value: -

RTNumber

Variable Type: ViInt16
Control Name: RTNumber
Description: : This parameter specifies the RT Number of the RT whose associated parameters like the Low Speed and High Speed Status Words, the Low Speed and High Speed BIT words, Action Word and the Vector Word are queried.
Variable Type: ViInt16.
Valid Range: wa4701_RT_NUMBER_MIN (0)
wa4701_RT_NUMBER_MAX (31)
Default: -

lsStatusWord

Variable Type: ViInt32 (passed by reference)
Control Name: lsStatusWord
Description: This output parameter returns the Low Speed Status Word associated with the specified RT.
Variable Type: ViInt32 *.
Valid Range: -
Default: -

hsStatusWord

Variable Type: ViInt32 (passed by reference)
Control Name: hsStatusWord
Description: : This output parameter returns the High Speed Status Word associated with the specified RT.
Variable Type: ViInt32 *.
Valid Range: -
Default: -

lsBitWord

Variable Type: ViInt32 (passed by reference)
Control Name: lsBitWord
Description: : This output parameter returns the Low Speed BIT (Built In Test) word associated with the specified RT.
Variable Type: ViInt32 *.
Valid Range: -
Default: -

hsBitWord

Variable Type: ViInt32 (passed by reference)
Control Name: hsBitWord
Description: : This output parameter returns the High Speed BIT (Built In Test) Word associated with the specified RT.
Variable Type: ViInt32 *.
Valid Range: -
Default: -

actionWord

Variable Type: ViInt32 (passed by reference)
Control Name: actionWord.
Description: : This output parameter returns the High Speed Action Word corresponding to the most recent HS message.
Variable Type: ViInt32 *.
Valid Range: -
Default: -

lsVectorWord

Variable Type: ViInt32 (passed by reference)
Control Name: lsVectorWord
Description: : This output parameter returns the most recent Vector Word transmitted by the simulated RT with the `Service Request' bit set in response to the `Transmit Vector' mode command by the BC
Variable Type: ViInt32 *
Valid Range: -
Default Value: -

Return Value

Control Name: status.
Description: : Displays the results of the function call.
Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_querySystemError

Function:

ViStatus wa4701_querySystemError (ViSession instrHndl, ViInt16 *errorNumber, ViChar errorString[]);

Panel Name : Query System Error

Purpose:

This function reads the first entry from the error queue. The error queue is a first in, first out queue where the errors are added as reported. The error message, which is read from this queue, is split up into its components i.e. the Error number and corresponding Error String.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument.
This parameter gets its value from the Initialise function.
Valid Range: -
Default Value: -

errorNumber

Variable Type: ViInt16 (passed by reference)
Control Name: errorNumber.
Description: This parameter returns the Error number component of the first entry read from the error queue.
Variable Type: ViInt16 *
Valid Range: -499 to 0

Possible Error Codes :

Value	Error		
0	No Error	-161	Invalid block data
-101	Invalid Character	-211	Trigger ignored
-102	Syntax Error	-213	Init ignored
-103	Invalid Separator	-215	Arm deadlock
-104	Data type error	-221	Settings conflict
-105	GET not allowed	-222	Data out of range
-108	Parameter not allowed	-223	Too much data
-109	Missing parameter	-224	Illegal parameter value
-110	Command header error	-225	Out of memory
-111	Header separator error	-310	System error
-113	Undefined header	-350	Too many errors
-114	Header suffix out of range	-410	Query INTERRUPTED
-121	Invalid character ni number	-420	Query UNTERMINATED
-131	Invalid suffix	-430	Query DEADLOCKED
-141	Invalid character data	-440	Query UNTERMINATED
-144	Character data too long		after indefinite response

errorString

Variable Type: ViChar []

Control Name: errorString.

Description: This output parameter returns the Error String corresponding to the error number.

Variable Type: ViChar []

Valid Range: -

Note: *It must be ensured that the character array, which is input to this function, is large enough to hold the error string returned by this function.*

Return Value

Control Name: status.

Description: Displays the results of the function call.

Variable Type: ViStatus

Note: *Use `wa4701_errorMessage()` function to retrieve the error message corresponding to the error code returned by this function.*

wa4701_readBCMsgData

Function:

ViStatus wa4701_readBCMsgData (ViSession instrHndl, ViInt16 msgNum, ViInt16 singleOrBlockMode, ViInt16 offset, ViInt32 receivedData[], ViInt16 *numOfRecvdWords, ViInt32 *statusWord1, ViInt32 *statusWord2, ViInt32 *timeTag);

Panel Name: Read BC Message Data

Purpose:

This function reads the data words from the receive buffer of the specified message. It returns the received data words array and the Status Words & Time Tag of the buffer.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument.
This parameter gets its value from the Initialise function.
Variable Type: ViSession
Valid Range: -
Default Value: -

msgNum

Variable Type: ViInt16
Control Name: msgNum
Description: This parameter specifies the message number from whose receive buffer the data words are to be read.
Variable Type: ViInt16.
Valid Range: wa4701_BC_MSG_NUM_MIN (1) to maxNumOfBCMsgs
where 'maxNumOfBCMsgs' is obtained by executing
'wa4701_getTransParamsNVRAM()' function
Default Value: -

singleOrBlockMode

Variable Type: ViInt16
Control Name: singleOrBlockMode
Description: This parameter specifies whether the data words are to be read in the Block Data or Single Data from the receive buffer of the specified message.
Variable Type: ViInt16.
Valid Range: wa4701_BLOCK_MODE (2220 / Block Mode) or
wa4701_SINGLE_MODE(2221 / Single Mode)
Default Value: wa4701_BLOCK_MODE

offset

Variable Type: ViInt16
Control Name: offset
Description: This parameter specifies the offset into the receive buffer of the specified message where the received data is to be read in Single Mode. If the Block Mode is used to read data, then this parameter is ignored.
Variable Type: ViInt16.
Valid Range: wa4701_SINGLE_MODE_OFFSET_MIN (0) to wa4701_SINGLE_MODE_OFFSET_MAX (4096).
Default Value: -

receivedData

Variable Type: ViInt32 []
Control Name: receivedData
Description: This output parameter is a pointer to an array, which will contain the data words read from the receive buffer of the specified message. It must be ensured that the array is large enough to hold the data words.
Variable Type: ViInt32 [].
Valid Range: -
Default Value: -

numOfRecvdWords

Variable Type: ViInt16 (passed by reference)
Control Name: numOfRecvdWords
Description: This output parameter specifies the number of data words in the received array. It would also contain the HS Overhead information in case of HS data.
Variable Type: ViInt16 *.
Valid Range: -
Default Value: -

statusWord1

Variable Type: ViInt32 (passed by reference)
Control Name: statusWord1
Description: This output parameter contains the status word received on the low speed bus from the RT(s) involved in the message transfer. Should an RT not have responded, this parameter will contain a value of 0xFFFF.
Variable Type: ViInt32 *.
Valid Range: -
Default Value: -

statusWord2

Variable Type: ViInt32 (passed by reference)
Control Name: statusWord2
Description: This output parameter contains the status word received from the second (receiving) RT in an RT to RT transfer. If the message does not define an RT to RT transfer, then this parameter will indicate a value of 0. Should the receiving RT not respond, then this parameter will indicate a value of 0xFFFF.
Variable Type: ViInt32 *.
Valid Range: -
Default Value: -

timeTag

Variable Type: ViInt32 (passed by reference)

Control Name: timeTag

Description: This output parameter contains the time tag value of the data buffer, which is associated with the specified message. The time tag is a 32 bit value, which indicates the time, in multiples of 10 microseconds, of the most recent transmission or reception of the data buffer.

Variable Type: ViInt32 *.

Valid Range: -

Default Value: -

Return Value

Control Name: status.

Description: Displays the results of the function call.

Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_readCMStackData

ViStatus wa4701_readCMStackData (ViSession instrHndl, ViInt16 msgNum, ViInt16 *numWordsLSBlock, ViInt16 *numWordsHSBlock, ViInt32 *timeTag, ViInt32 *respTimeRT1, ViInt32 *respTimeRT2, ViInt32 *lsMsgError, ViInt32 *hsMsgError, ViInt32 lsdataArray[], ViInt32 lsErrorWordArray[], ViInt32 hsHeaderArray[], ViInt32 hsdataArray[]);

Panel Name: Read CM Stack Data

Purpose:

This function reads the data of the specified message from the stack. Each message captured by the chronological monitor is identified by a number. The function reads the stack and retrieves the data corresponding to that message.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument.
This parameter gets its value from the Initialise function.
Variable Type: ViSession
Valid Range: -
Default Value: -

msgNum

Variable Type: ViInt16
Control Name: msgNum
Description: This parameter specifies the message number of the message in the stack whose data is to be read. Each message captured by the chronological monitor is identified by a number, zero is the oldest message captured, and the most recent message can be ascertained by executing the Query CM Stack function.
Variable Type: ViInt16.
Valid Range: minMsgNum to maxMsgNum
where `minMsgNum' is the oldest message captured in the stack and `maxMsgNum' is the most recent message captured in the stack. These values can be found out by executing the Query CM Stack function.
Default Value: -

numWordsLSBlock

Variable Type: ViInt16 (passed by reference)
Control Name: numWordsLSBlock.
Description: This output parameter indicates the number of words in the LS Block of the message whose data has been read from the stack.
Variable Type: ViInt16 *
Valid Range: -
Default: -

numWordsHSBlock

Variable Type: ViInt16 (passed by reference)
Control Name: numWordsHSBlock.
Description: This output parameter indicates the number of words in the HS Block of the message whose data has been read from the stack.
Variable Type: ViInt16 *
Valid Range: -
Default: -

timeTag

Variable Type: ViInt32 (passed by reference)
Control Name: timeTag.
Description: This output parameter indicates the Message Time Tag of the message whose data has been read from the stack. The Time Tag is the time in units of 0.5 microseconds since the chronological monitor was started to the point when the message was captured in the stack..
Variable Type: ViInt32 *
Valid Range: -
Default: -

respTimeRT1

Variable Type: ViInt32 (passed by reference)
Control Name: respTimeRT1.
Description: This output parameter specifies the RT1's response time in units of 0.5 microseconds. If the RT did not respond, then this parameter will indicate a value of FFFF Hex.
Variable Type: ViInt32 *
Valid Range: -
Default: -

respTimeRT2

Variable Type: ViInt32 (passed by reference)
Control Name: respTimeRT2.
Description: This output parameter specifies the RT2's response time in units of 0.5 microseconds. If the RT did not respond, then this parameter will indicate a value of FFFF Hex. This is normal for the second RT response time in all messages except RT to RT transfer.
Variable Type: ViInt32 *
Valid Range: -
Default: -

lsMsgError

Variable Type: ViInt32 (passed by reference)
Control Name: lsMsgError.
Description: This output parameter indicates the LS Message Error word obtained from the LS Block of the message whose data was read from the stack.
Variable Type: ViInt32 *
Valid Range: -
Default: -

hsMsgError

Variable Type: ViInt32 (passed by reference)
Control Name: hsMsgError.
Description: This output parameter indicates the HS Message Error word of the HS Block of the message whose data was read from the stack.
Variable Type: ViInt32 *
Valid Range: -
Default: -

IsdataArray

Variable Type: ViInt32 []
Control Name: IsdataArray.
Description: This output parameter is an array which contains the Low Speed Data words of the specified message read from the stack.
Variable Type: ViInt32 []
Valid Range: -
Default: -

IsErrorWordArray

Variable Type: ViInt32 []
Control Name: IsErrorWordArray.
Description: This output parameter is an array which contains the Low Speed Error Words in a one-to-one correspondence with the `IsData' Word Array.
Variable Type: ViInt32 []
Valid Range: -
Default: -

hsHeaderArray

Variable Type: ViInt32 []
Control Name: hsHeaderArray.
Description: This output parameter is an array which contains the following information from the HS Block of the message whose data has been read from the stack.
Element 0 Word Count Deviation
Element 1 Frame Control / Physical Address
Element 2 Frame Check Sequence
Variable Type: ViInt32 []
Valid Range: -
Default: -

hsdataArray

Variable Type: ViInt32 []
Control Name: hsdataArray.
Description: This output parameter is an array which contains the following information obtained from the HS Block of the message whose data has been read from the stack.
Element 0 - Destination Address
Element 1 - Word Count (If implemented, else the first two words are only data words)
Element 2 upto (Number Of Words in HS Block - 4)
 - contain the HS Data Words.
Variable Type: ViInt32 [].
Valid Range: -
Default: -

Return Value

Control Name: status.

Description: Displays the results of the function call.

Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_readRTData

Function:

ViStatus wa4701_readRTData (ViSession instrHndl, ViInt16 RTNumber, ViInt16 lsOrHsFlag, ViInt16 subAddressOrHSID, ViInt16 singleOrBlockMode, ViInt16 offset, ViInt16 uniqueOrGlobalFlag, ViInt32 *receiveTimeTag, ViInt32 *transmitTimeTag, ViInt32 receivedArray[], ViInt16 *numOfRecvdWords);

Panel Name: Read RT Data

Purpose:

This function reads from the receive data buffer of the specified RT/Subaddress. It also returns the Receive Time Tag and Transmit Time Tag of the last access made to the associated data buffer.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument.
This parameter gets its value from the Initialise function.
Variable Type: ViSession
Valid Range: -
Default Value: -

RTNumber

Variable Type: ViInt16
Control Name: RTNumber
Description: This parameter specifies the RT Number of the RT from whose receive buffer, the received data words are read.
Variable Type: ViInt16.
Valid Range: wa4701_RT_NUMBER_MIN (0)
wa4701_RT_NUMBER_MAX (31)
Default: -

Note: *If the 'Wrap Around' facility of the RT for the specified unique/global low speed subaddress/HDIS is set ON, then the transmit and receive buffers are one and the same and the data is read from one common buffer.*

lsOrHsFlag

Variable Type: ViInt16
Control Name: lsOrHsFlag
Description: This parameter is used to indicate whether the received data words are to be read from the receive buffer of the unique/global low speed subaddress (or) the unique/global high speed identifier of the specified RT
Variable Type: ViInt16.
Valid Range:

Value	Interpretation
wa4701_RT_LS (1074)	Unique/Global Low Speed Subaddress
wa4701_RT_HS (1075)	Unique/Global High Speed Identifier

Default Value: wa4701_RT_LS

subAddressOrHSID

Variable Type: ViInt16
Control Name: subAddressOrHSID
Description: This parameter specifies the low speed subaddress or high speed identifier from whose receive buffer the received data words are to be read.
Variable Type: ViInt16
Valid Range: For Low Speed Subaddress
w4701_RT_SA_MIN to w4701_RT_SA_MAX,
For High Speed Identifier
w4701_RT_HSID_MIN to w4701_RT_HSID_MAX.
Default Value: -

Note : *This parameter is ignored by the function if the `uniqueOrGlobalFlag' is set for global low speed subaddress/high speed identifier.*

singleOrBlockMode

Variable Type: ViInt16
Control Name: singleOrBlockMode
Description: This parameter specifies whether the data words to be read from the receive buffer of the specified RT should be read in Block Mode or Single Mode.
Variable Type: ViInt16.
Valid Range: wa4701_BLOCK_MODE or wa4701_SINGLE_MODE
Default Value: -

offset

Variable Type: ViInt16
Control Name: offset
Description: This parameter specifies the offset into the receive buffer from where the received data is to be read. This parameter is ignored if the `singleOrBlockMode' is set to Block mode.
Variable Type: ViInt16
Valid Range: The range for this parameter depends on whether the transfer was a low speed or high speed type.
Low Speed Transfer
wa4701_SDATA_LS_OFFSET_MIN (0)
wa4701_SDATA_LS_OFFSET_MAX.(32)

High Speed Transfer
wa4701_SDATA_HS_OFFSET_MIN (0)
wa4701_SDATA_HS_OFFSET_MAX (4096)
Default Value: -

uniqueOrGlobalFlag

Variable Type: ViInt16
Control Name: uniqueOrGlobal
Description: This parameter is used to specify whether the received data is to be read from the receive buffer of the unique or global low speed subaddress/high speed identifier.
Variable Type: ViInt16.
Valid Range: wa4701_CONFIGURE_FOR_UNIQUE,
wa4701_CONFIGURE_FOR_GLOBAL,
Default: wa4701_CONFIGURE_FOR_UNIQUE

receiveTimeTag**Variable Type:** ViInt32 (passed by reference)**Control Name:** receiveTimeTag**Description:** This output parameter contains the value of the time tag of the most recent access made to the receive data buffer. In the case that the RT is not simulated by the IIB-3910-VXI, the receive time tag and the transmit time tag will be identical, there being only one data buffer for monitoring all accesses to the external RT.**Variable Type:** ViInt32 ***Valid Range:** -**Default Value:** -**Note:** *The Time Tag Counter will be cleared to zero when the Multi Remote Terminal Logical Instrument is activated.***transmitTimeTag****Variable Type:** ViInt32 (passed by reference)**Control Name:** transmitTimeTag**Description:** This output parameter contains the value of the time tag of the most recent access made to the transmit data buffer. In the case that the RT is not simulated by the IIB-3910-VXI, the receive time tag and the transmit time tag will be identical, there being only one data buffer for monitoring all accesses to the external RT.**Variable Type:** ViInt32 ***Valid Range:** -**Default Value:** -**Note:** *The Time Tag Counter will be cleared to zero when the Multi Remote Terminal Logical Instrument is activated.***receivedArray****Variable Type:** ViInt32 []**Control Name:** receivedArray**Description:** This output parameter is an array which contains the data words read from the receive data buffer of the specified RT. If the Wrap Around of the RT for the specified low speed subaddress/HSID is ON, then the receive and transmit data buffers are one and the same and the data words will be read from one common buffer.**Variable Type:** ViInt32 []**Valid Range:** -**Default Value:** -**numOfRecvdWords****Variable Type:** ViInt16 (passed by reference)**Control Name:** numOfRecvdWords**Description:** This output parameter indicates the number of data words that have been read from the receive data buffer of the specified low speed subaddress/high speed identifier of the input RT number.**Variable Type:** ViInt16 ***Valid Range:** -**Default Value:** -**Return Value****Control Name:** status.**Description:** Displays the results of the function call.**Variable Type:** ViStatus**Note:** *Use wa4701_errorMessage() function to retrieve the error message corresponding to the error code returned by this function.*

wa4701_reset

Function:

ViStatus wa4701_reset (ViSession instrHndl);

Panel Name: Reset

Purpose:

This function performs a soft reset on the instrument.

Effects of *RST:

- (1) The currently active logical instrument (if any) is deactivated and the default logical instrument specified by the NVRAM is selected. Subsequent activation of a logical instrument will result in default parameters.
- (2) The Source Voltage is set to the default value specified by the NVRAM.
- (3) The output coupling is set to DIRECT.
- (4) The format for data and status register data is set to the default <NR1 NUMERIC RESPONSE DATA> format.
- (5) The byte order format is set to NORMAL.
- (6) The error queue is cleared.
- (7) Any pending *OPC and *OPC? commands are aborted.

Parameter List

instrHndl

Variable Type: ViSession

Control Name: instrHndl.

Description: A unique session handle to the instrument.
This parameter gets its value from the Initialise function.

Variable Type: ViSession

Valid Range: -

Default Value: -

Return Value

Control Name: status.

Description: Displays the results of the function call.

Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_scanChassis

Function:

ViStatus wa4701_scanChassis (ViInt16 laArray[], ViInt16 slotArray[],
ViInt16 *numOf4701sFound, ViInt16 *VXI_GPIBFlag);

Panel Name: Scan Chassis for IIB-3910-VXI

Purpose:

This function searches the chassis for all instances of the IIB-3910-VXI modules and returns information about the same. This function scans the entire chassis and returns an array of logical addresses of all the 4701 modules found and their respective slot numbers in the chassis. It also returns information regarding the number of 4701s found and the interface used to connect to them.

Parameter List

laArray

Variable Type: ViInt16 []

Control Name: laArray

Description: This output parameter returns an array of the Logical Addresses of all the IIB-3910-VXI modules found in the chassis.

Variable Type: ViInt16 []

Valid Range: -

Default Value: -

slotArray

Variable Type: ViInt16 []

Control Name: slotArray

Description: This output parameter returns an array of the Slot Numbers in the chassis where all the IIB-3910-VXI modules were found. There is a one - to - one correspondence between the 'laArray' and 'slotArray' output parameters. In other words, the slot number corresponding to the n'th valid element of the 'laArray' is the n'th element in the 'slotArray' where $0 \leq n < \text{numOf4701sFound}$

Variable Type: ViInt16[]

Valid Range: -

Default Value: -

numOf4701sFound

Variable Type: ViInt16 (passed by reference)

Control Name: numOf4701sFound

Description: This output parameter indicates the number of IIB-3910-VXI modules found in the chassis. This provides the user with the information as to how many elements in the 'laArray' and 'slotArray' arrays are valid.

Variable Type: ViInt16 *

Valid Range: -

Default Value: -

VXI_GPIBFlag

Variable Type: ViInt16 (passed by reference)
Control Name: VXI_GPIBFlag
Description: This output parameter returns whether the interface used to connect to the instruments is a GPIB-VXI interface or a VXI-MXI interface.
Variable Type: ViInt16 *
Valid Range: wa4701_VXI_CONNECTION (0) - VXI-MXI Interface
wa4701_GPIB_VXI_CONNECTION (1) - GPIB-VXI Interface
Default Value: -

Return Value

Control Name: status.
Description: Displays the results of the function call.
Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_selfTest

Function:

ViStatus wa4701_selfTest (ViSession instrHndl, ViInt16 *result);

Panel Name: Self Test

Purpose:

This function causes the IIB-3910-VXI to perform an internal self test. This activity could take up to 20 seconds to execute during which time the BC,MRT,CM & DDL LEDs' on the front panel are illuminated. A return value of 0 indicates a successful self test. Should the test fail, a non-zero result is bit-mapped. If it takes more than 25 seconds for self test to complete, then the function times out.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument.
This parameter gets its value from the Initialise function.
Variable Type: ViSession
Valid Range: -
Default Value: -

result

Variable Type: ViInt16 (passed by reference)
Control Name: result
Description: This output parameter returns the result of the self test performed on the IIB-3910-VXI. If the self test has been successful, then this parameter will contain a value of 0 else it will contain a bit-mapped value which can be interpreted as specified below.

Weight	Bit#	Test Failure
128	7	PMU's Internal Self Test
64	6	Interrupt Controller Logic
32	5	Multi-Function Peripheral
16	4	Serial I/O and Timer Logic
8	3	NVRAM
4	2	2 MByte Multi-Port SRAM
1	0	CPU EPROM

Variable Type: ViInt16 *

Return Value

Control Name: status.
Description: Displays the results of the function call.
Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_setBCMsgOrFrameParamsToDefault

Function:

ViStatus wa4701_setBCMsgOrFrameParamsToDefault (ViSession instrHndl,
ViInt16 msgOrFrameFlag, ViInt16 msgOrFrameNum);

Panel Name: Set Msg/Frame Params to Default

Purpose:

This function sets the parameters associated with either the specified frame or message back to their respective default values. This is a convenient method of restoring a known set of parameters of either the frame or message.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl
Description: A unique session handle to the instrument.
This parameter gets its value from the Initialise function.
Variable Type: ViSession
Default Value: -

msgOrFrameFlag

Variable Type: ViInt16
Control Name: msgOrFrameFlag
Description: This parameter specifies whether the input value to the parameter `msgOrFrameNum' is a message number or a frame number. Depending on the value of this parameter, the appropriate message or frame parameters are set to default values.
Variable Type: ViInt16
Default Value: -

msgOrFrameNum

Variable Type: ViInt16
Control Name: msgOrFrameNum
Description: This parameter specifies either the message number or frame number that is going to be set to default depending on what is specified in the `msgOrFrameFlag' flag. The maximum value for either frame number or message number depends on the last configured value in the NVRAM. The range has been specified below and would be applied based on what value is specified for the input parameter `msgOrFrameFlag' flag.
Variable Type: *For Message*
wa4701_BC_MSG_NUM_MIN to maxNumOfBCMsgs
where `maxNumOfBCMsgs' is obtained by executing
`wa4701_getTransParamsNVRAM()' function

For Frame
wa4701_BC_FRAME_NUM_MIN to numOfBCFrames.
where `maxNumOfBCMsgs' is obtained by executing
`wa4701_getTransParamsNVRAM()' function.

Return Value

Control Name: status.
Description: Displays the results of the function call.
Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_setCMDelayAndPostTrigCount

Function:

ViStatus wa4701_setCMDelayAndPostTrigCount (ViSession instrHndl, ViReal64 delayseconds, ViInt32 postTriggerCount);

Panel Name: Setup CM Delay/Post Trigger Count

Purpose:

This function is used to configure

- (1) The Delay (amount of time during which CM stores messages) after the TRIG:STOP condition has been satisfied
- (2) The Post Trigger Count i.e. the maximum number of messages which the CM will capture after the TRIG:STOP condition is satisfied.

Parameter List

instrHndl

Variable Type: ViSession

Control Name: instrHndl.

Description: A unique session handle to the instrument.
This parameter gets its value from the Initialise function.

Variable Type: ViSession

Valid Range: -

Default: -

delayseconds

Variable Type: ViReal64

Control Name: delay

Description: This parameter specifies the delay value to be set between the specified event detector being satisfied and the subsequent downward transition. In other words, it specifies a time in seconds for traversal between two states. This is valid only for the exit from the TRIG:STOP state.

Variable Type: ViReal64

Valid Range: wa4701_CONF_CM_TIME_MIN (0 seconds) to
wa4701_CONF_CM_TIME_MAX (1.288497E7 seconds)

Default Value: wa4701_CONF_CM_TIME_MIN

postTriggerCount

Variable Type: ViInt32

Control Name: postTriggerCount

Description: This parameter configures the maximum number of messages which the chronological monitor will capture after the TRIG:STOP is satisfied. If no other condition stops the capturing, then the number of messages specified by this parameter will be captured and stored in the stack.

Variable Type: ViInt32

Valid Range: wa4701_CONF_CM_CNT_MIN (0) to
wa4701_CONF_CM_CNT_MAX (32768)

Default Value: 100

Return Value

Control Name: status.

Description: Displays the results of the function call.

Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_setOperatingMode

Function:

ViStatus wa4701_setOperatingMode (ViSession instrHndl, ViInt16 operatingMode);

Panel Name: Set Operating Mode

Purpose:

This function sets the current operating mode of the instrument. Depending on the value specified to the function, the Bus Controller, the Multi Remote Terminal or the Chronological Monitor is set as the current logical instrument and activated.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument.
This parameter gets its value from the Initialise function.
Valid Range: -
Default: -

operatingMode

Variable Type: ViInt16
Control Name: operatingMode
Description: This selects the logical mode of the instrument that is to be activated.
Valid Range: wa4701_SET_MODE_BC(Bus Controller)
wa4701_SET_MODE_MRT (Multi Remote Terminal)
wa4701_SET_MODE_CM(Chronological Monitor)
Default: wa4701_SET_MODE_BC

Return Value

Control Name: status.
Description: Displays the results of the function call.
Variable Type: ViStatus.

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_setRTParamsToDefaults

Function:

ViStatus wa4701_setRTParamsToDefaults (ViSession instrHndl, ViInt16 RTNumber);

Panel Name: Set RT Parameters to Default

Purpose:

This function is used to set back the parameters associated with the specified RT back to default values.

Parameter List

instrHndl

Variable Type: ViSession

Control Name: instrHndl.

Description: A unique session handle to the instrument.
This parameter gets its value from the Initialise function.

Variable Type: ViSession

Valid Range: -

Default Value: -

RTNumber

Variable Type: ViInt16

Control Name: RTNumber.

Description: This parameter is used to specify the RT Number whose parameters are to be set back to default values.

Variable Type: ViInt16.

Valid Range: wa4701_RT_NUMBER_MIN (0)
wa4701_RT_NUMBER_MAX (31)

Default: -

Return Value

Control Name: status.

Description: Displays the results of the function call.

Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_setRTSrq

Function:

ViStatus wa4701_setRTSrq (ViSession instrHndl, ViInt16 RTNumber, ViInt16 srq, ViInt32 vector);

Panel Name: Set RT SRQ

Purpose:

This function is used to set the Vector Word of the specified RT. It sets up either the SRQ1 or the SRQ2.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument.
This parameter gets its value from the Initialise function.
Variable Type: ViSession
Valid Range: -
Default Value: -

RTNumber

Variable Type: ViInt16
Control Name: RTNumber
Description: This parameter specifies the RT Number of the RT for which the Vector Word is to be set. The specified vector word is then sent by the RT in response to the 'Transmit Vector' mode command sent by the BC.
Variable Type: ViInt16.
Valid Range: wa4701_RT_NUMBER_MIN (0)
wa4701_RT_NUMBER_MAX (31)

Default: -

srq

Variable Type: ViInt16
Control Name: srq.
Description: This parameter is used to specify whether the vector word is to be set using the SRQ1 or the SRQ2 command. If a sequence of SRQ1 and SRQ2 commands were to be sent to the IIB-3910-VXI, the SRQ2 vectors would be transmitted in response to the 'transmit vector' mode commands before the SRQ1 commands.
Variable Type: ViInt16.
Valid Range: wa4701_RT_SRQ1 (1)
wa4701_RT_SRQ2 (2)
Default: wa4701_RT_SRQ2

Note:

(1) *Setting up the vector word for the specified RT would instruct the IIB-3910-VXI to transmit all low speed status words from the simulated RT with the 'Service Request' bit set until the Bus Controller sends the 'Transmit Vector' mode command to that simulated RT. It is only after the receipt of this mode command and the subsequent transmission of the vector word that the IIB-3910-VXI clears the 'Service Request' Bit in the status words reported by the simulated RT.*

(2) *IIB-3910-VXI maintains a queue of up to 16 pending SRQ commands and vectors, such that the 'Service Request' bit may be set again immediately after the 'transmit vector' mode command.*

vector

Variable Type: ViInt32
Control Name: vector.
Description: This parameter is used to specify the Vector Word which is to be associated with the specified RT.
Variable Type: ViInt32.
Valid Range: wa4701_RT_SRQ_VECTOR_MIN(0)
wa4701_RT_SRQ_VECTOR_MAX(65535)
Default: -

Return Value

Control Name: status.
Description: : Displays the results of the function call.
Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_setStatusMask

Function:

ViStatus wa4701_setStatusMask (ViSession instrHndl, ViInt16 serviceReqEnableReg, ViInt16 eventStatusEnableReg, ViInt32 operationPosFilter, ViInt32 operationNegFilter, ViInt32 operationEventEnable);

Panel Name: Set Status Mask

Purpose:

This function enables or disables the mask bits which controls the status reporting registers on the instrument to control the following

- (1) which bits of the IEEE488.2 Status Byte Register cause the module to generate a 'request for service' to its VXI commander
- (2) which bits of the IEEE488.2 Standard Event Status register will cause the ESB bit in the Status Byte Register to be set.
- (3) which bits of the SCPI Operation Condition Register, when going from 0 to 1 set the corresponding bit in the Operation Event Register.
- (4) which bits of the SCPI Operation Condition Register, when going from 1 to 0 set the corresponding bit in the Operation Event Register.
- (5) which bits of the SCPI Operation Event Register will cause the OSS bit of the Status Byte Register to be set.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique handle to the instrument.
This parameter gets its value from the Initialise function.
Variable Type: ViSession
Valid Range: -
Default: -

serviceReqEnableReg

Variable Type: ViInt16
Control Name: serviceReqEnableReg
Description: Mask to control which bits of the IEEE488.2 Status Byte Register cause the module to generate a 'request for service' to its VXI commander. The bits are :

Bit #	Name
0	CM complete
1	BC complete
2	Error Queue not Empty
3	Always 0
4	MAV - Message Available
5	ESB - Event Status Bit
6	Master Summary Status
7	OSS - Operation Summary Status

Variable Type: ViInt16
Valid Range: wa4701_SERVICE_REQ_ENAB_MIN (0) to wa4701_SERVICE_REQ_ENAB_MAX (255)
Default Value: -

eventStatusEnableReg

Variable Type: ViInt16
Control Name: eventStatusEnableReg
Description: Mask to control which bits of the IEEE488.2 Standard Event Status register will cause the ESB bit in the Status Byte Register to be set. The bits are :

Bit #	Name
0	OPC - Operation Complete
1	RQC - Request Control. Always 0 on 4701.
2	QYE - Query Error
3	DDE - Device Dependent Error
4	EXE - Execution Error
5	CME - Command Error
6	URQ - User Request. Always 0 on 4701.
7	PON - Power On

Variable Type: ViInt16
Valid Range: wa4701_EVENT_STATUS_ENAB_MIN (0) to wa4701_EVENT_STATUS_ENAB_MAX (255)
Default Value: -

operationPosFilter

Variable Type: ViInt32
Control Name: operationPosFilter
Description: Mask to control which bits of the SCPI Operation Condition Register, when going from 0 to 1 set the corresponding bit in the Operation Event Register. The bits are :

Bit #	Name
5	TRG - This is set when CM logical instrument is active and waiting for trigger stop condition to be met
8	BC- BC Active
9	MRT - MRT Active
10	M- CM Active
11	DLT - DDL Source Active.
12	DLR - DDL Sink Active.

All other bits are ignored.

Variable Type: ViInt32
Valid Range: wa4701_OPER_PTR_MIN (0) wa4701_OPER_PTR_MAX (65535)
Default: -

operationNegFilter

Variable Type: ViInt32
Control Name: operationNegFilter
Description: Mask to control which bits of the SCPI Operation Condition Register, when going from 1 to 0 set the corresponding bit in the Operation Event Register. The bits are :

Bit #	Name
5	TRG - This is set when CM logical instrument is active and waiting for trigger stop condition to be met
8	BC- BC Active
9	MRT - MRT Active
10	CM- CM Active
11	DLT - DDL Source Active.
12	DLR - DDL Sink Active.

All other bits are ignored.

Variable Type: ViInt32
Valid Range: wa4701_OPER_NTR_MIN (0)
wa4701_OPER_NTR_MAX (65535)
Default Value: -

operationEventEnable

Variable Type: ViInt32
Control Name: : operationEventEnable
Description: Mask to control which bits of the SCPI Operation Event Register will cause the OSS bit of the Status Byte Register to be set. The bits are :

Bit #	Name
5	TRG - This is set when CM logical instrument is active and waiting for trigger stop condition to be met
8	BC- BC Active
9	MRT - MRT Active
10	CM- CM Active
11	DLT - DDL Source Active.
12	DLR - DDL Sink Active.

All other bits are ignored.

Variable Type: ViInt32
Valid Range: wa4701_OPER_EVENT_STAT_MIN (0)
wa4701_OPER_EVENT_STAT_MAX (65535)
Default Value: -

Return Value

Control Name: status.
Description: Displays the results of the function call.
Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_setupBCCycleMsgSequence

Function:

ViStatus wa4701_setupBCCycleMsgSequence (ViSession instrHndl, ViInt16 frameNum, ViInt16 cycleNum, ViInt16 sequenceOfMessages[], ViInt16 numOfMsgsInSequence, ViInt16 syncPulseFlag, ViInt16 typeOfSyncPulse);

Panel Name: Setup BC Cycle Message Sequence

Purpose:

This function is used to set up the sequence of messages to be transmitted within the referenced frame & cycle. It also sets the state of the synchronisation pulse within the referenced cycle.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument.
This parameter gets its value from the Initialise function.
Valid Range: -
Default Value: -

frameNum

Variable Type: ViInt16
Control Name: frameNum
Description: This parameter specifies the frame number of the frame to which the specified cycle belongs. The valid maximum range for this parameter depends on the last configured value in the NVRAM.
Variable Type: ViInt16.
Valid Range: wa4701_BC_FRAME_NUM_MIN to numOfBCFrames.
where `numOfBCFrames' is the output obtained from `wa4701_getTransParamsNVRAM()'
Default Value: -

cycleNum

Variable Type: ViInt16
Control Name: cycleNum
Description: This parameter specifies the cycle number of the input frame for which the sequence of messages and the synchronisation pulse is to be configured. The maximum valid range for this parameter depends on the last configured value in the NVRAM..
Variable Type: ViInt16
Valid Range: wa4701_BC_FRAME_NCYC_MIN to maxNumOfCyclPerFrame.
where `maxNumOfCyclPerFrame' is the output obtained from `wa4701_getTransParamsNVRAM()' function
Default Value: -

sequenceOfMessages

Variable Type: ViInt16 []
Control Name: sequenceOfMessages
Description: This parameter specifies the sequence of messages that are to be transmitted within the referenced cycle of the specified frame. The messages have to be defined previously. It is permissible to repeat the same message in the sequence any number of times.

Variable Type: ViInt16 []

Default Value: -

numOfMsgsInSequence

Variable Type: ViInt16
Control Name: numOfMsgsInSequence
Description: This parameter specifies the number of elements in the message sequence list.
Variable Type: ViInt16
Valid Range: wa4701_BC_FRAME_NCYC_MIN to preDefinedNumOfCycl.
where preDefinedNumOfCycl is the pre defined number of cycles for the specified frame. This value can be found by reading the `queryBCFrameParams` function.

Default Value: -

syncPulseFlag

Variable Type: ViInt16
Control Name: syncPulseFlag
Description: This parameter is a flag which indicates whether the function should set the state of the synchronisation pulse within the referenced cycle of the referenced frame.

Variable Type: ViInt16

Valid Range: wa4701_BC_FRAME_SPUL_YES(1223 / Set Pulse) or
wa4701_BC_FRAME_SPUL_NO (1224 / Don't set pulse)

Default Value: wa4701_BC_FRAME_SPUL_NO.

typeOfSyncPulse

Variable Type: ViInt16
Control Name: typeOfSyncPulse
Description: This parameter specifies the type of synchronising pulse to be applied within the referenced cycle of the referenced frame.

Variable Type: ViInt16

Valid Range : Value Interpretation
wa4701_BC_FRAME_TRIG_POS(1230) Positive
wa4701_BC_FRAME_TRIG_NEG(1231) Negative
wa4701_BC_FRAME_TRIG_HIGH (1232) High
wa4701_BC_FRAME_TRIG_LOW(1233) Low

Default Value: wa4701_BC_FRAME_TRIG_HIGH.

Return Value

Control Name: status.
Description: Displays the results of the function call.
Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_setupBCFrameParams

Function:

ViStatus wa4701_setupBCFrameParams (ViSession instrHndl, ViInt16 frameNum, ViInt16 numberOfCycles, ViInt16 syncAtEndOfCycle, ViReal64 cyclePeriodmilliseconds);

Panel Name: Setup BC Frame Parameters

Purpose:

This function configures the following BC Frame related parameters

- (1) Number Of Cycles within the frame (NCYC)
- (2) The Period of cycle in the frame(PER)
- (3) The Synchronisation at end of each cycle within the specified frame (SYNC)

Note : *The Cycle Period (PER) will be effective on the instrument only if the Synchronisation Type (SYNC) employed is Internal*

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument.
This parameter gets its value from the Initialise function.
Valid Range: -
Default Value: -

frameNum

Variable Type: ViInt16
Control Name: frameNum
Description: This parameter specifies the frame number of the frame to be configured.
The valid maximum range for this parameter depends on the last configured value in the NVRAM.
Variable Type: ViInt16.
Valid Range: wa4701_BC_FRAME_NUM_MIN to numOfBCFrames.
where `numOfBCFrames' is the output obtained from `wa4701_getTransParamsNVRAM()'
Default Value: -

numberOfCycles

Variable Type: ViInt16
Control Name: numberOfCycles
Description: This parameter sets the number of cycles within the selected frame.
Variable Type: ViInt16.
Valid Range: wa4701_BC_FRAME_NCYC_MIN to maxNumOfCyclPerFrame
where `maxNumOfCyclPerFrame' is the output obtained from `wa4701_getTransParamsNVRAM()' function.
Default Value: 1.

syncAtEndOfCycle

Variable Type: ViInt16
Control Name: syncAtEndOfCycle
Description: This parameter specifies the type of synchronisation to be applied at the end of each cycle.
Variable Type: ViInt16.
Valid Range: Value Interpretation
wa4701_BC_FRAME_SYNC_NONE (1235) None
wa4701_BC_FRAME_SYNC_INT(1236) Internal
wa4701_BC_FRAME_SYNC_EXT(1237) External
Default Value: wa4701_BC_FRAME_SYNC_NONE.

cyclePeriodmilliseconds

Variable Type: ViReal64
Control Name: cyclePeriod.
Description: This parameter sets the period for each cycle within a frame. This parameter takes effect ONLY if the synchronisation mode is set to 'INTERNAL' i.e. the formal parameter 'syncAtEndOfCycle' is set to wa4701_BC_FRAME_SYNC_INT. The input period value is assumed to be expressed in 'milliseconds'
Variable Type: ViInt16.
Valid Range: wa4701_BC_FRAME_CYCLE_PRD_MIN (0.5 milliseconds) to wa4701_BC_FRAME_CYCLE_PRD_MAX (500 milliseconds)
Default Value: wa4701_BC_FRAME_CYCLE_PRD_MIN.

Note: *If a value of 20.0 is specified for this parameter, then the function interprets the period value to be configured as 20 milliseconds.*

Return Value

Control Name: status.
Description: Displays the results of the function call.
Variable Type: ViStatus

Note: *Use wa4701_errorMessage() function to retrieve the error message corresponding to the error code returned by this function.*

wa4701_setupBCMMsgDTAG

Function:

ViStatus wa4701_setupBCMMsgDTAG (ViSession instrHndl, ViInt16 msgNum, ViInt16 dynamicTagOffset, ViInt16 dtag1Or2Flag, ViInt16 upperByteValue);

Panel Name: Setup BC Message Dynamic Tag

Purpose:

This function sets up the dynamic tag words for the specified message. Each message can have up to two dynamic tag words. The function provides an option to set up either DTAG1 or DTAG2.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument.
This parameter gets its value from the Initialise function.
Variable Type: ViSession
Valid Range: -
Default Value: -

msgNum

Variable Type: ViInt16
Control Name: msgNum
Description: This parameter specifies the message number of the message for which the Dynamic Tag is to be injected.
Variable Type: ViInt16.
Valid Range: wa4701_BC_MSG_NUM_MIN to maxNumOfBCMmsgs
where 'maxNumOfBCMmsgs' is obtained by executing
'wa4701_getTransParamsNVRAM()' function
Default Value: -

dynamicTagOffset

Variable Type: ViInt16
Control Name: : dynamicTagOffset
Description: This parameter specifies the offset of the dynamic tag word. Location 0 would indicate the first word in the data buffer associated with the specified message.
A value of -1 would disable the dynamic tag.
Variable Type: ViInt16
Valid Range: wa4701_DISABLE_DTAG (-1) to (preDefinedNumberOfWords - 1)
where 'preDefinedNumberOfWords' is the number of words that have been currently associated with the specified message number.
Default Value: -

dtag1Or2Flag

Variable Type: ViInt16
Control Name: dtag1Or2Flag
Description: Each message may have up to two dynamic tag words. This parameter specifies whether the dynamic tag to be set is DTAG1 or DTAG2.
Variable Type: ViInt16
Valid Range: wa4701_SET_DTAG1_FOR_MESG or wa4701_SET_DTAG2_FOR_MESG
Default Value: wa4701_SET_DTAG1_FOR_MESG

upperByteValue

Variable Type: ViInt16
Control Name: upperByteValue
Description: This parameter specifies the value of the upper byte of the word. This value is fixed.
Variable Type: ViInt16
Valid Range: wa4701_UPPER_BYTE_MIN (0) to wa4701_UPPER_BYTE_MAX (255)
Default Value: -

Return Value

Control Name: status.
Description: Displays the results of the function call.
Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_setupBCMsgErrorInj

Function:

ViStatus wa4701_setupBCMsgErrorInj (ViSession instrHndl, ViInt16 msgNum, ViInt16 errorType, ViInt16 lsErrorInjCapability, ViInt32 param_1, ViInt32 param_2, ViInt32 param_3);

Panel Name: Setup BC Error Injection Params

Purpose:

This function is used to inject either low speed or high speed errors into a message.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument.
This parameter gets its value from the Initialise function.
Valid Range: -
Default Value: -

msgNum

Variable Type: ViInt16
Control Name: msgNum
Description: This parameter specifies the message number of the message in which the specified error is to be injected. The maximum possible value for this parameter depends on the last configured value in the NVRAM. This value can be found by executing the `wa4701_getTransParamsNVRAM()` function.
Variable Type: ViInt16.
Valid Range: wa4701_BC_MSG_NUM_MIN to maxNumOfBCMsgs
where `maxNumOfBCMsgs` is obtained by executing
`wa4701_getTransParamsNVRAM()` function
Default Value: -

errorType

Variable Type: ViInt16
Control Name: errorType
Description: This parameter specifies the type of error that is to be injected into the message to be transmitted.
Variable Type: ViInt16.
Valid Range:

Value Interpretation, Low speed Words Injection	
wa4701_BC_MSG_ERR_LS_NONE(1200)	Disables Error Injection
wa4701_BC_MSG_ERR_LS_PAR (1201)	Parity Error Injection
wa4701_BC_MSG_ERR_LS_SYNC(1202)	Sync Error Injection
wa4701_BC_MSG_ERR_LS_MANCH (1203)	Manchester Error Injection
wa4701_BC_MSG_ERR_LS_WRD_CNT (1204)	Word Count Error Injection
wa4701_BC_MSG_ERR_LS_WRD_LEN (1205)	Word Length Error Injection
wa4701_BC_MSG_ERR_LS_WRG_BUS (1206)	Wrong Bus Error Injection
wa4701_BC_MSG_ERR_LS_BOTH_BUS(1207)	Transmit on both buses
Value Interpretation, HS Frame Injection	
wa4701_BC_MSG_ERR_HS_NONE(1208)	Disables Error Injection
wa4701_BC_MSG_ERR_HS_WRD_CNT (1209)	Word Count Error Injection
wa4701_BC_MSG_ERR_HS_FCS (1210)	Frame Check Sequence Error
wa4701_BC_MSG_ERR_HS_MANCH (1211)	Manchester Error Injection

Note: *The instrument supports only one Low Speed Error per message. Only one Manchester error may be injected in the HS frame - this may be in addition to FCS and / or Word Count on HS message. The use of 'None' clears all error injection associated with the message.*

Default Value: wa4701_BC_MSG_ERR_LS_NONE.

IsErrorInjCapability

Variable Type: ViInt16

Control Name: IsErrorInjCapability

Description: This parameter specifies which of the four different capabilities of a low speed transaction should the error injection be introduced into. For more details, refer the manual.

Variable Type: ViInt16.

Valid Range:	Values	Interpretation
	wa4701_BC_MSG_ERR_LS_CAPABILITY_1	(1)LS1
	wa4701_BC_MSG_ERR_LS_CAPABILITY_2	(2)LS2
	wa4701_BC_MSG_ERR_LS_CAPABILITY_3	(3)LS3
	wa4701_BC_MSG_ERR_LS_CAPABILITY_4	(4)LS4

Default Value: wa4701_BC_MSG_ERR_LS_CAPABILITY_1

Note: *If the error to be injected is in a HS Frame, then this parameter is ignored. Its value is considered only for low speed error injection.*

param_1

Variable Type: ViInt32

Control Name: param_1

Description: This parameter specifies the required input in order to set up the required error. Depending on the type of error to be injected, the valid range for this parameter changes.

Variable Type: ViInt32.

Valid Range:	Error Type	Valid Range
Low Speed Errors	None	This parameter is ignored
	Parity	wa4701_BC_MSG_ERR_WNUM_MIN (0) to wa4701_BC_MSG_ERR_WNUM_MAX (32).
	Synchro	wa4701_BC_MSG_ERR_WNUM_MIN (0) to wa4701_BC_MSG_ERR_WNUM_MAX (32).
	Manchester	wa4701_BC_MSG_ERR_WNUM_MIN (0) to wa4701_BC_MSG_ERR_WNUM_MAX (32).
	Word Count	wa4701_BC_MSG_ERR_LS_WCNT_MIN (-63) to wa4701_BC_MSG_ERR_LS_WCNT_MAX (63).
	Word Length	wa4701_BC_MSG_ERR_WNUM_MIN (0) to wa4701_BC_MSG_ERR_WNUM_MAX (32).
	Wrong Bus	This parameter is ignored
High Speed Errors	Both Buses	This parameter is ignored
	None	This parameter is ignored
	Word Count	wa4701_BC_MSG_ERR_HS_WCNT_MIN (-1) to wa4701_BC_MSG_ERR_HS_WCNT_MAX (1).
	Frame Check Sequence	wa4701_BC_MSG_ERR_HS_FCS_ON (1211) or wa4701_BC_MSG_ERR_HS_FCS_OFF (1212).
	Manchester	wa4701_BC_MSG_ERR_HS_MANC_BIT_MIN (0) to wa4701_BC_MSG_ERR_HS_MANC_BIT_MAX (65535)

param_2

Variable Type: ViInt32

Control Name: param_2

Description: This parameter specifies the required input in order to set up the required error. Depending on the type of error to be injected, the valid range for this parameter changes.

Variable Type: ViInt32.

Valid Range:	Error Type	Valid Range
Low Speed Errors	None	This parameter is ignored
	Parity	This parameter is ignored
	Synchro	wa4701_BC_MSG_ERR_LS_SYNC_MIN (1) to wa4701_BC_MSG_ERR_LS_SYNC_MAX (63).
	Manchester	wa4701_BC_MSG_ERR_LS_MANC_BIT_MIN (0) to wa4701_BC_MSG_ERR_LS_MANC_BIT_MAX (31).
	Word Count	This parameter is ignored.
	Word Length	wa4701_BC_MSG_ERR_LS_WLEN_MIN (-8) to wa4701_BC_MSG_ERR_LS_WLEN_MAX (8).
High Speed Errors	Wrong Bus	This parameter is ignored
	Both Buses	This parameter is ignored
	None	This parameter is ignored
	Word Count	This parameter is ignored
	Frame Check Sequence	This parameter is ignored
	Manchester	wa4701_BC_MSG_ERR_HS_MANC_LVL_ZERO or wa4701_BC_MSG_ERR_HS_MANC_LVL_ONE.

param_3

Variable Type: ViInt32

Control Name: param_3

Description: This parameter specifies the required input in order to set up the required error. Depending on the type of error to be injected, the valid range for this parameter changes.

Variable Type: ViInt32.

Valid Range:	Error Type	Valid Range
Low Speed Errors	None	This parameter is ignored
	Parity	This parameter is ignored
	Synchro	This parameter is ignored
	Manchester	This parameter is ignored
	Word Count	This parameter is ignored
	Word Length	This parameter is ignored
	Wrong Bus	This parameter is ignored
	Both Buses	This parameter is ignored
High Speed Errors	None	This parameter is ignored
	Word Count	This parameter is ignored
	Frame Check Sequence	This parameter is ignored
	Manchester	wa4701_BC_MSG_ERR_HS_MANC_PRE (1215) to wa4701_BC_MSG_ERR_HS_MANC_EDEL(1220).

Return Value

Control Name: status.

Description: Displays the results of the function call.

Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_setupBCMsgParams

Function:

ViStatus wa4701_setupBCMsgParams (ViSession instrHndl, ViInt16 msgNum, ViInt16 sourceRTNum, ViInt16 sourceSubAddress, ViInt16 destRTNum, ViInt16 destSubAddress, ViInt16 numOfWeeks, ViInt16 lsOrHsBus, ViInt16 typeOfLSBus, ViInt16 typeOfHSBus, ViInt16 commandFlag, ViInt16 modeCommand, ViInt32 modeCommandData);

Panel Name: Setup BC Message Parameters

Purpose:

This function sets up a message for transfer. The various parameters assist in setting up the message source and destination, the number of words to be transmitted, the bus to be used and to set up a mode command for transmission if so desired.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument.
This parameter gets its value from the Initialise function.
Valid Range: -
Default Value: -

msgNum

Variable Type: ViInt16
Control Name: msgNum
Description: This parameter specifies the message number of the message which is to be defined.
Variable Type: ViInt16.
Valid Range: wa4701_BC_MSG_NUM_MIN to maxNumOfBCMsgs
where 'maxNumOfBCMsgs' is obtained by executing
'wa4701_getTransParamsNVRAM()' function
Default Value: -

sourceRTNum

Variable Type: ViInt16
Control Name: sourceRTNum
Description: This parameter specifies the Source RT number of the message that is being set up.
Variable Type: ViInt16.
Valid Range: BC is source wa4701_BC_MSG_SOURCE_RT_MIN (-1)
RT is source wa4701_BC_MSG_SOURCE_RT_MIN + 1 (1) to
wa4701_BC_MSG_SOURCE_RT_MAX (30)
Default Value: -

sourceSubAddress

Variable Type: ViInt16
Control Name: sourceSubAddress
Description: This parameter specifies the source RT subaddress. If the source is the BC, then this value is ignored. If the source is an RT then it should fall in the range mentioned below.
Variable Type: ViInt16.
Valid Range: The range would depend on whether the configuration is being done for a low speed subaddress or a high speed identifier.

BC is source
wa4701_BC_MSG_LS_SOURCE_SA_MIN (0) or
wa4701_BC_MSG_HS_SOURCE_SA_MIN (0)

RT is source for Low Speed Transfer
wa4701_BC_MSG_LS_SOURCE_SA_MIN + 1 (1) to
wa4701_BC_MSG_LS_SOURCE_SA_MAX (30).

RT is source for High Speed Transfer
wa4701_BC_MSG_HS_SOURCE_SA_MIN + 1 (1) to
wa4701_BC_MSG_HS_SOURCE_SA_MAX (127).

Default Value: -

destRTNum

Variable Type: ViInt16
Control Name: destRTNum
Description: This parameter specifies the destination RT number of the message that is being set up.
Variable Type: ViInt16.
Valid Range: BC is destination
wa4701_BC_MSG_DEST_RT_MIN (-1)

RT is destination

wa4701_BC_MSG_DEST_RT_MIN + 1 (0) to
wa4701_BC_MSG_DEST_RT_MAX (31).

Default Value: -

Note: *If the destination RT number is specified as 31, then IIB-3910-VXI assumes the transfer to be a 'broadcast' type.*

destSubAddress

Variable Type: ViInt16
Control Name: destSubAddress
Description: This parameter specifies the destination RT subaddress. If the destination is the BC, then this value is ignored. If the destination is an RT then it should fall in the range mentioned below.

Variable Type: ViInt16.
Valid Range: The below mentioned range applies only if the destination RT Subaddress is specified for an RT. The valid range for this parameter depends on whether the configuration is being done for a low speed subaddress or a high speed identifier.

BC is destination
wa4701_BC_MSG_LS_DEST_SA_MIN (0) or
wa4701_BC_MSG_HS_DEST_SA_MIN (0)

RT is destination for Low Speed Transfers
wa4701_BC_MSG_LS_DEST_SA_MIN + 1 (1)to
wa4701_BC_MSG_LS_DEST_SA_MAX (30).

RT is destination for High Speed Transfers
wa4701_BC_MSG_HS_DEST_SA_MIN + 1 (1) to
wa4701_BC_MSG_HS_DEST_SA_MAX (127).

Default Value: -

numOfWords

Variable Type: ViInt16
Control Name: numOfWords
Description: This parameter specifies the number of words to be transferred in a message.
The range of values differ between a high speed message and low speed message.
Variable Type: ViInt16.
Valid Range : For a low speed message
wa4701_BC_LS_WRD_CNT_MIN (1) to
wa4701_BC_LS_WRD_CNT_MAX (32).
For a high speed message
wa4701_BC_HS_WRD_CNT_MIN (1) to
wa4701_BC_HS_WRD_CNT_MAX (4096).
Default Value: -

IsOrHsBus

Variable Type: ViInt16
Control Name: IsOrHsBus
Description: This parameter indicates whether the message is to be set up for transfer on the low speed bus or the high speed bus. This parameter also configures the message to be transmitted on the low or high speed bus.
Variable Type: ViInt16.
Valid Range: Value Interpretation
wa4701_BC_MSG_BUS_LS Low Speed Bus
wa4701_BC_MSG_BUS_HS High Speed Bus
Default Value: wa4701_BC_MSG_BUS_LS.

typeOfLSBus

Variable Type: ViInt16
Control Name: typeOfLSBus
Description: This parameter indicates whether the transmission is to take place on the primary or secondary bus if the transmission is to take place on the low speed bus.
Variable Type: ViInt16.
Valid Range: wa4701_BC_MSG_LS_BUS_PRI Primary Bus
wa4701_BC_MSG_LS_BUS_SEC Secondary Bus
Default Value: wa4701_BC_MSG_LS_BUS_PRI

typeOfHSBus

Variable Type: ViInt16
Control Name: typeOfHSBus
Description: This parameter could take two different meanings depending on whether the message to be setup consists of a mode command or a normal command. If the message consists of a mode command and the `IsOrHsBus' parameter specifies a high speed bus, then this parameter specifies the A/B `bus select' bit of the high speed action word.
Variable Type: ViInt16.
Valid Range:

Value	Interpretation
wa4701_BC_MSG_HS_BUS_NONE	None
wa4701_BC_MSG_HS_BUS_PRI	Primary Bus
wa4701_BC_MSG_HS_BUS_SEC	Secondary Bus

Default: wa4701_BC_MSG_HS_BUS_NONE

commandFlag

Variable Type: ViInt16
Control Name: commandFlag
Description: This parameter indicates whether the command word to be transmitted is a mode command or not.
Variable Type: ViInt16.
Valid Range:

Value	Interpretation
wa4701_BC_MSG_MCOM_WITH_DATA	Mode Command with data
wa4701_BC_MSG_MCOM_WITHOUT_DATA	Mode Comm. without data
wa4701_BC_MSG_NOT_MCOM	Not a mode command

Default Value: wa4701_BC_MSG_NOT_MCOM

modeCommand

Variable Type: ViInt16
Control Name: modeCommand
Description: This parameter specifies the mode command that is to be transmitted. This parameter takes effect only if the `modeCommFlag' parameter specifies a mode command. The valid range of values depend on whether it is a low speed mode command or a high speed mode command.
Variable Type: ViInt16.
Valid Range: For low speed mode command
wa4701_BC_MSG_MCOM_LS_NUM_MIN (0) to
wa4701_BC_MSG_MCOM_LS_NUM_MAX (31).

For high speed mode command
wa4701_BC_MSG_MCOM_HS_NUM_MIN (0) to
wa4701_BC_MSG_MCOM_HS_NUM_MAX (127).
Default Value: -

modeCommandData

Variable Type: ViInt32
Control Name: modeCommandData
Description: This parameter specifies the data to be associated with the mode command. This parameter takes effect only if the `modeCommFlag' flag specifies a value of `wa4701_BC_MSG_MCOM_WITH_DATA'.
Variable Type: ViInt16.
Valid Range: wa4701_BC_MSG_MCOM_DATA_MIN (0) to
wa4701_BC_MSG_MCOM_DATA_MAX (65535).
Default Value: -

Return Value

Control Name: status.
Description: Displays the results of the function call.
Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_setupBCMsgTiming

Function:

ViStatus wa4701_setupBCMsgTiming (ViSession instrHndl, ViInt16 msgNum, ViInt16 lsOrHsBus, ViReal32 imgValuemicoseconds, ViInt16 mmgValuemicoseconds, ViInt16 RTResponseTimemicoseconds, ViInt16 riRiOutFlag, ViInt16 riRiOutValuemicoseconds, ViInt16 nprValue);

Panel Name: Setup BC Msg Timing Parameters

Purpose:

This function sets up the below mentioned Timing Parameters related to a message.

- (1) Inter Message Gap (IMG)
- (2) Mid Message Gap (MMG)
- (3) RT Response Time (TIM:RTIM)
- (4) Receive Initialise (RI)
- (5) Receiver Initialisation time-out (RIOUT)
- (6) Number of Preamble Bits (NPR)

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument.
This parameter gets its value from the Initialise function.
Valid Range: -
Default Value: -

msgNum

Variable Type: ViInt16
Control Name: msgNum
Description: This parameter specifies the message number of the message which is to be defined.
Variable Type: ViInt16.
Valid Range: wa4701_BC_MSG_NUM_MIN to maxNumOfBCMsgs
where 'maxNumOfBCMsgs' is obtained by executing
'wa4701_getTransParamsNVRAM()' function
Default Value: -

lsOrHsBus

Variable Type: ViInt16
Control Name: lsOrHsBus
Description: This parameter specifies whether the Inter Message Gap (IMG) and the Response Time (RTIM) are to be set for the Low Speed Bus (or) whether the Inter Message Gap (IMG), the Mid Message Gap (MMG), the Response Time (RTIM), the Receive Initialise(RI)/Receiver Initialisation time-out (RIOUT),and the Number Of Preamble Bits (NPR) are to be set for the High Speed Bus.
Variable Type: ViInt16.
Valid Range:

Value	Interpretation
wa4701_BC_MSG_BUS_LS (1085)	Low Speed Bus
wa4701_BC_MSG_BUS_HS (1086)	High Speed Bus

Default Value: wa4701_BC_MSG_BUS_LS.

imgValuemicoseconds

Variable Type: ViReal32

Control Name: imgValue
Description: This parameter specifies the Inter Message Gap(IMG) that can be set for the specified message. This value is the gap time on the low speed bus at the end of the message before further transmission occurs. The valid range for this parameter is specified below. The function interprets the value specified in this parameter to be in microseconds.
Variable Type: ViReal32
Valid Range: wa4701_BC_ERR_TIM_IMG_MIN (20 microseconds) to wa4701_BC_ERR_TIM_IMG_MAX (6553.5 microseconds)

Note: *If a value of 20.0 is specified for this parameter, then the function interprets the IMG value to be configured as 20.0 microseconds.*

mmgValuemicroseconds

Variable Type: ViInt16
Control Name: mmgValue
Description: This parameter specifies the Mid-Message Gap (MMG) that is to be set for the specified message. This is the gap time on the low speed bus between the two messages of a high speed RT to RT message. If the specified message is not defined as a High Speed RT to RT transfer, then this parameter has no effect on the message. The function interprets the value specified for this parameter to be expressed in microseconds.
Variable Type: ViInt16
Valid Range: wa4701_BC_ERR_TIM_MMG_MIN (10 microseconds) to wa4701_BC_ERR_TIM_MMG_MAX (1000 microseconds)

Note: *If a value of 20 is specified for this parameter, then the function interprets the MMG value to be configured for the instrument as 20 microseconds.*

RTResponseTimemicroseconds

Variable Type: ViInt16
Control Name: RTResponseTime
Description: This parameter sets the response time for the simulated RT in BC mode which is accessed by the specified message.
Variable Type: ViInt16
Valid Range: wa4701_BC_ERR_TIM_RTIM_MIN (4 microseconds) to wa4701_BC_ERR_TIM_RTIM_MAX (31 microseconds)

riRiOutFlag

Variable Type: ViInt16
Control Name: riRiOutFlag
Description: This parameter is a flag that specifies whether the parameter `riRiOut' should set the Receive Initialise (Ri) time of a BC to RT high speed message (or) whether it should set the Receiver Initialise time-out (Riout) time of an RT to BC high speed message.
Variable Type: ViInt16
Valid Range:

Value	Interpretation
wa4701_BC_MSG_TIM_RI(1111)	Receive Initialise
wa4701_BC_MSG_TIM_RIOUT (1112)	Receiver Initialisation Timeout

riRiOutValuemicroseconds

Variable Type: ViInt16
Control Name: riRiOutValue

Description: : This parameter is used to specify whether the Receive Initialise (Ri) time for a BC to RT high speed message is to be set (or) a Receiver Initialisation Timeout (Riout) for an RT to BC high speed message is to be set depending on the value of the parameter 'riRiOutFlag'. The valid values of both the possible cases has been mentioned below. The function interprets the specified timing value to be expressed in microseconds.

Variable Type: ViInt16

Valid Range: For Ri
wa4701_BC_ERR_TIM_RI_MIN (20 microseconds) to
wa4701_BC_ERR_TIM_RI_MAX (255 microseconds).

For Riout
wa4701_BC_ERR_TIM_RIOUT_MIN (20 microseconds) to
wa4701_BC_ERR_TIM_RIOUT_MAX (255 microseconds).

Note: (1) If a value of 20 is specified for this parameter, then the function interprets the Ri/RiOut value to be configured for the instrument as 20 microseconds.

(2) This parameter will not take effect if the message number which has been input to this function has been injected with errors. The instrument will return error -221 into the error queue. The 'checkSystemError()' function can be used to read the error queue to check the same. So it is imperative so ensure that the response time not be set for a message which has been injected with error(s)

nprValue

Variable Type: ViInt16

Control Name: nprValue

Description: : This parameter is used to specify the number of preamble bits to be set for a high speed message. This is the number of preamble bits that the instrument will transmit at the beginning of a HS frame. If the message is not a high speed transfer, then this parameter has no effect on the instrument.

Variable Type: ViInt16

Valid Range: wa4701_BC_ERR_TIM_NPREAMB_MIN (2) to
wa4701_BC_ERR_TIM_NPREAMB_MAX (257).

Default Value: -

Return Value

Control Name: status.

Description: Displays the results of the function call.

Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_setupBCParameters

Function:

ViStatus wa4701_setupBCParameters (ViSession instrHndl, ViInt16 hsSubAddress, ViInt16 busTimeOutmicroseconds, ViInt16 destAddrMode, ViInt16 globalResponseTimeFlag, ViInt16 responseTime, ViInt16 msgUpdateType);

Panel Name: Setup BC Parameters

Purpose: This function is used to set the following parameters related to the BC.

- (1) High Speed Subaddress (HSS)
- (2) Bus Time Out (BTO)
- (3) Destination Address Mode (DAM)
- (4) Global Reponse Time for all simulated RTs (RTIM)
- (5) Message Update Type (UPD)

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument
Valid Range: -
Default Value: -

hsSubAddress

Variable Type: ViInt16
Control Name: hsSubAddress
Description: This parameter specifies the High Speed Subaddress for the system. This is used by the IIB-3910-VXI to correctly generate command words when executing the high level message commands.
Variable Type: ViInt16.
Valid Range: wa4701_SETUP_BC_HSS_MIN(1)
wa4701_SETUP_BC_HSS_MAX(30)
Default Value: wa4701_SETUP_BC_HSS_MIN(1)

busTimeOutmicroseconds

Variable Type: ViInt16
Control Name: busTimeOut (microseconds)
Description: : This parameter is used to specify the Bus Controller Bus Timeout (BTO). This is the time that the IIB-3910-VXI waits for an external RT to respond before assuming that it has not responded.
Variable Type: ViInt16.
Valid Range: wa4701_SETUP_BC_BTO_MIN(14 microseconds)
wa4701_SETUP_BC_BTO_MAX(1000microsecond)
Default Value: wa4701_SETUP_BC_BTO_MIN(14 microseconds)

destAddrMode

Variable Type: ViInt16
Control Name: destAddrMode
Description: : This parameter is used to specify the algorithm used by the IIB-3910-VXI to calculate the destination address for a high speed message.
Variable Type: ViInt16.

Valid Range:	Value	Interpretation
	wa4701_SETUP_BC_DAM_NONE	None
	wa4701_SETUP_BC_DAM_PHYS	Physical
	wa4701_SETUP_BC_DAM_LOG	Logical
Default Value:	wa4701_SETUP_BC_DAM_NONE.	

globalResponseTimeFlag

Variable Type:	ViInt16	
Control Name:	globalResponseTimeFlag	
Description:	: This parameter is used to specify whether the function should set the global response time for all the simulated RTs. If the value specified for this parameter is to NOT set the global response time, then the value specified in the `globalResponseTime` field is ignored.	
Variable Type:	ViInt16.	
Valid Range :	wa4701_SET_GLOBAL_RESPONSE_TIME	Set global response time
	wa4701_DONT_SET_GLOBAL_RESPONSE_TIME	Don't set the response time
Default Value:	wa4701_SET_GLOBAL_RESPONSE_TIME	

responseTime

Variable Type:	ViInt16	
Control Name:	responseTime	
Description:	This parameter is used to specify the Global response time which is to be set for all the simulated RTs. If the flag `globalResponseTimeFlag` specifies that the global response time should NOT be set, then this value is ignored.	
Variable Type:	ViInt16.	
Valid Range:	wa4701_BC_ERR_TIM_RTIM_GLOBAL_MIN (4 microseconds) to wa4701_BC_ERR_TIM_RTIM_GLOBAL_MAX (99 microseconds).	
Default Value:	wa4701_BC_ERR_TIM_RTIM_GLOBAL_MIN	

Note: *The function interprets the value entered into this field to be in microseconds(e.g. if a value of 20 was entered, then the Global Response Time to be set for all the simulated RTs is interpreted as 20 microseconds).*

msgUpdateType

Variable Type:	ViInt16	
Control Name:	msgUpdateType	
Description:	This parameter is used to specify the method adopted by the IIB-3910-VXI to update new messages. The mode adopted reflects whether the message update will be done immediately or whether the update will be done at the end of cycle or frame.	
Variable Type:	ViInt16.	
Valid Range:		Value Interpretation
	wa4701_UPDATE_MODE_IMM	Immediate
	wa4701_UPDATE_MODE_CYCLE	At the end of current cycle
	wa4701_UPDATE_MODE_FRAME	At the end of current frame
Default Value:	wa4701_UPDATE_MODE_IMM	

Return Value

Control Name:	status.
Description:	Displays the results of the function call.
Variable Type:	ViStatus

Note: Use `wa4701_errorMessage()` function to retrieve the error message corresponding to the error code returned by this function.

wa4701_setupCMStartSeqArmTrig

Function:

ViStatus wa4701_setupCMStartSeqArmTrig (ViSession instrHndl, ViInt16 armOrTriggerFlag, ViInt16 source, ViInt16 slope);

Panel Name: Setup CM Start Sequence

Purpose:

This function configures the Arm and Trigger Event Detection Layers of the Start Sequence of the Chronological Monitor.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument. This parameter gets its value from the Initialise function.
Variable Type: ViSession
Valid Range: -
Default: -

armOrTriggerFlag

Variable Type: ViInt16
Control Name: armOrTriggerFlag.
Description: This parameter specifies whether the Arm or Trigger Event Detection Layer is being configured for the Start Sequence.
Variable Type: ViInt16.
Valid Range: wa4701_CONF_CM_ARM(1027 / Arm) or wa4701_CONF_CM_TRIGGER(1028 / Trigger).
Default Value: wa4701_CONF_CM_ARM.

source

Variable Type: ViInt16
Control Name: source
Description: This parameter specifies the Source used by the specified event detection layer for the specified sequence. The valid values for this parameter depend on whether the configuration is being done for Arm Or Trigger.
Variable Type: ViInt16.
Valid Range: For Arm
wa4701_CONF_CM_SOURCE_BUS (1020 / Bus) or
wa4701_CONF_CM_SOURCE_IMM.(1023 / Immediate)

For Trigger
wa4701_CONF_CM_SOURCE_IMM (1023 / Immediate) or
wa4701_CONF_CM_SOURCE_EXT (1022 / External)
Default Value: wa4701_CONF_CM_SOURCE_BUS.

slope

Variable Type: ViInt16
Control Name: slope
Description: This parameter specifies the slope of the output pulse generated when TRIG:START condition is satisfied.

Variable Type: ViInt16.
Valid Range: wa4701_CONF_CM_SLOPE_NEG (1050 / Negative Pulse)
wa4701_CONF_CM_SLOPE_POS (1051 / Positive Pulse)
Default Value: wa4701_CONF_CM_SLOPE_NEG.

Note: *This parameter takes effect only if the Trigger is being configured and the source is set to External. If the Arm is being configured, then this parameter is ignored.*

Return Value

Control Name: status.
Description: Displays the results of the function call.
Variable Type: ViStatus

Note: *Use **wa4701_errorMessage()** function to retrieve the error message corresponding to the error code returned by this function.*

wa4701_setupCMStopSeqArmTrig

Function:

ViStatus wa4701_setupCMStopSeqArmTrig (ViSession instrHndl, ViInt16 armOrTriggerFlag, ViInt16 layerNum, ViInt16 source, ViInt16 wordNum, ViInt16 lsOrHsFlag, ViChar digitalMask[], ViInt16 wordType, ViInt16 bus, ViInt16 errorType, ViInt16 mode);

Panel Name: Setup CM Stop Sequence

Purpose:

This function is used to configure the Arm and Trigger Event Detection Layers of the Stop Sequence of the Chronological Monitor.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique handle to the instrument.
This parameter gets its value from the Initialise function.
Valid Range: -
Default: -

armOrTriggerFlag

Variable Type: ViInt16
Control Name: armOrTriggerFlag.
Description: This parameter specifies whether the Arm or Trigger Event Detection Layer is being configured for the Stop Sequence.
Variable Type: ViInt16.
Valid Range: wa4701_CONF_CM_ARM(1027 / Arm) or wa4701_CONF_CM_TRIGGER(1028 / Trigger).
Default Value: wa4701_CONF_CM_ARM.

layerNum

Variable Type: ViInt16
Control Name: layerNum.
Description: Selects the event layer number to be configured in the Stop Sequence.
Variable Type: ViInt16.
Valid Range: wa4701_CONF_CM_LAYER1 (1 / Layer 1)
wa4701_CONF_CM_LAYER2 (2 / Layer 2)
wa4701_CONF_CM_LAYER3 (3 / Layer 3).
Default Value: wa4701_CONF_CM_LAYER1.

Note: *This parameter is ignored if the Trigger event detection layer is being configured.*

source

Variable Type: ViInt16
Control Name: source
Description: This parameter configures the Source for activating the event detection layer.
Variable Type: ViInt16.
Valid Range: For Arm
wa4701_CONF_CM_SOURCE_INT (1021 / Internal) or
wa4701_CONF_CM_SOURCE_IMM.(1023 / Immediate)

For Trigger

Default Value: wa4701_CONF_CM_SOURCE_INT (1021 / Internal) or
wa4701_CONF_CM_SOURCE_IMM (1023 / Immediate)
wa4701_CONF_CM_SOURCE_INT.

wordNum

Variable Type: ViInt16
Control Name: wordNum.
Description: This parameter specifies the word number of the digital mask used for checking occurrence or non-occurrence of this word in the messages on the Lowspeed & Highspeed buses.
Variable Type: ViInt16.
Valid Range: wa4701_CONF_CM_WORD1(1 / Word 1)
wa4701_CONF_CM_WORD2(2 / Word 2)
wa4701_CONF_CM_WORD3(3 / Word 3) or
wa4701_CONF_CM_WORD4(4 / Word 4)
Default Value: wa4701_CONF_CM_WORD1.

lsOrHsFlag

Variable Type: ViInt16
Control Name: lsOrHsFlag
Description: This parameter is a flag which indicates whether a Low speed word or a High speed word is being defined for event detection.
Variable Type: ViInt16.
Valid Range: wa4701_CONF_CM_LS(1026 / Low Speed Word)
wa4701_CONF_CM_HS(1025 / High Speed Word)
Default Value: wa4701_CONF_CM_LS

digitalMask

Variable Type: ViChar []
Control Name: : digitalMask
Description: This parameter configures the digital mask for the LS word. Depending on the format in which the mask is being specified, the parameter contains character values to define binary, octal, hexadecimal or the MIL-STD-1553 word by pre-fixing the mask value with the qualifiers as shown below.

Qualifier for digital mask Interpretation

X	Don't care
#B	Binary
#O	Octal
#H	Hexadecimal
#W	MIL-STD-1553 Word

Variable Type: ViChar [].
Valid Range: -
Default Value: -

Note: For example, if a value of FFFF (Hex) was to be input to this parameter, then the input should be "#HFFFF" where #H qualifies the input value to indicate a hexadecimal value.

wordType

Variable Type: ViInt16
Control Name: wordType
Description: This parameter is used to qualify (specify) the digital mask values with the type shown below.
Variable Type: ViInt16.
Valid Range: Value Interpretation

	wa4701_CONF_CM_COMM_NONE(1040)	Don't Care
	wa4701_CONF_CM_COMM_COMM(1041)	Command Word
	wa4701_CONF_CM_COMM_DATA(1042)	Data Word
	wa4701_CONF_CM_COMM_RT_RT (1043)	RT-RT Word
	wa4701_CONF_CM_COMM_STATUS(1044)	Status
Default Value:	wa4701_CONF_CM_COMM_NONE.	

bus

Variable Type: ViInt16
Control Name: bus
Description: This parameter specifies the bus on which the CM parses the data.
Variable Type: ViInt16.
Valid Range:

	Value Interpretation
wa4701_CONF_CM_BUS_NONE (1045)	None
wa4701_CONF_CM_BUS_PRI(1046)	Primary Bus
wa4701_CONF_CM_BUS_SEC(1047)	Secondary Bus
wa4701_CONF_CM_BUS_BOTH (1048)	Both Buses

Default Value: wa4701_CONF_CM_BUS_NONE

errorType

Variable Type: ViInt16
Control Name: errorType.
Description: The error type is used to specify an additional qualifier to the digital mask. If specified, CM looks for the errors to satisfy the defined event detection conditions for the selected layer.
Variable Type: ViInt16.
Valid Range: For LS word :

- wa4701_CONF_CM_LS_ERR_VAL_MIN
- wa4701_CONF_CM_LS_ERR_VAL_MAX

wa4701_LS_ERR_NONE
wa4701_LS_ERR_PARITY
wa4701_LS_ERR_SYNC
wa4701_LS_ERR_SWORD
wa4701_LS_ERR_LWORD
wa4701_LS_ERR_MANC
wa4701_LS_ERR_NRES
wa4701_LS_ERR_WCO
wa4701_LS_ERR_TADD

For HS word :

- wa4701_CONF_CM_HS_ERR_VAL_MIN
- wa4701_CONF_CM_HS_ERR_VAL_MAX

wa4701_HS_ERR_NONE
wa4701_HS_ERR_FCS
wa4701_HS_ERR_EDEL
wa4701_HS_ERR_NRESP
wa4701_HS_ERR_WCOUNT

Default Value: -

Example:

ViStatus w4701_setupCMStopSeqArmTrig (ViSession instrHndl,
ViInt16 armOrTriggerFlag,
ViInt16 layerNum,
ViInt16 source,

```
ViInt16 wordNum,  
ViInt16 lsOrHsFlag,  
ViChar digitalMask[],  
ViInt16 wordType,  
ViInt16 bus,  
wa4701_HS_ERR_NRESP,  
wa4701_HS_ERR_WCOUNT,  
ViInt16 mode);
```

mode

Variable Type: ViInt16

Control Name: mode

Description: This parameter configures the mode in which word comparison is done by the CM on the defined Low speed or High speed word. If Normal then the event detection layer looks for the occurrence of the specified word in the messages on the bus else it looks for non-occurrence of the specified word in the messages on the bus.

Variable Type: ViInt16.

Valid Range: wa4701_CONF_CM_MODE_NORM (1027 / Normal Mode) or
wa4701_CONF_CM_MODE_INV(1028 / Inverted Mode)

Default Value: wa4701_CONF_CM_MODE_NORM

Return Value

Control Name: status.

Description: Displays the results of the function call.

Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_setupExternalTrigger

Function:

ViStatus wa4701_setupExternalTrigger (ViSession instrHndl, ViInt16 trigStopSlope, ViInt16 postTrigCountSlope);

Panel Name: Setup CM External Trigger

Purpose:

This function is used to configure the generation of hardware synchronisation pulses which appear at the `TRIG OUT' connector on the front panel and optionally, as a TTLTRG line on the VXIbus backplane at the moment when the TRIG:STOP condition is satisfied or at the moment the capturing of the post trigger count has been completed or at both moments.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique handle to the instrument.
This parameter gets its value from the Initialise function.
Valid Range: -
Default : -

trigStopSlope

Variable Type: ViInt16
Control Name: trigStopSlope
Description: This parameter specifies the slope of the pulse generated after TRIG:STOP condition is satisfied. A value of HIGH means no pulse is generated. A value of NEG means a Negative pulse is generated.
Variable Type: ViInt16.
Valid Range: wa4701_TRIG_STOP_SLOPE_NEG(1063 / Negative)
wa4701_TRIG_STOP_SLOPE_HIGH (1064 / High)
Default Value: wa4701_TRIG_STOP_SLOPE_NEG

postTrigCountSlope

Variable Type: ViInt16
Control Name: postTrigCountSlope
Description: This parameter specifies the slope of the pulse generated after the capturing of the post trigger count has completed. A value of HIGH means no pulse is generated. A value of NEG means a Negative pulse is generated.
Variable Type: ViInt16.
Valid Range: wa4701_POST_TRIG_SLOPE_NEG(1065 / Negative Pulse)
wa4701_POST_TRIG_SLOPE_HIGH (1066 / High Pulse)
Default Value: wa4701_POST_TRIG_SLOPE_NEG

Return Value

Control Name: status.
Description: Displays the results of the function call.
Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_setupRTHSErrInj

Function:

ViStatus wa4701_setupRTHSErrInj (ViSession instrHndl, ViInt16 RTNumber, ViInt16 uniqueOrGlobalHSIDFlag, ViInt16 hsIdentifier, ViInt16 errorType, ViInt32 param_1, ViInt32 param_2, ViInt32 param_3);

Panel Name: Setup RT High Speed Err Inject

Purpose:

This function is used to inject errors into the HS Frame associated with the specified unique or global high speed identifier.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument.
This parameter gets its value from the Initialise function.
Variable Type: ViSession
Valid Range: -
Default Value: -

RTNumber

Variable Type: ViInt16
Control Name: RTNumber
Description: This parameter specifies the RT Number for which errors are to be injected into the HS Frame associated with the specified unique or global High Speed Identifier.
Variable Type: ViInt16.
Valid Range: wa4701_RT_NUMBER_MIN (0)
wa4701_RT_NUMBER_MAX (31)
Default: -

uniqueOrGlobalHSIDFlag

Variable Type: ViInt16
Control Name: uniqueOrGlobalHSIDFlag
Description: This parameter is used to specify whether the HSID, for which errors are to be injected, is a unique or global high speed identifier.
Variable Type: ViInt16.
Valid Range: wa4701_CONFIGURE_FOR_UNIQUE (2205 / Unique HSID)
wa4701_CONFIGURE_FOR_GLOBAL (2206 / Global HSID)
Default: wa4701_CONFIGURE_FOR_UNIQUE

hsIdentifier

Variable Type: ViInt16
Control Name: hsIdentifier.
Description: This parameter specifies the High Speed Identifier of the RT whose associated HS Frame is to be injected with the specified error.
Variable Type: ViInt16.
Valid Range: wa4701_RT_HSID_NUM_MIN (1)
wa4701_RT_HSID_NUM_MAX (127)
Default: -

Note: *If the `uniqueOrGlobal` flag is set to inject errors into the global high speed identifier, then this parameter is ignored. This parameter is range checked only when the error injection is to done for the unique high speed identifier.*

errorType

Variable Type: ViInt16

Control Name: errorType.

Description: This parameter is used to specify the type of error to be injected into the HS Frame associated with the global/unique high speed identifier.

Variable Type: ViInt16.

Valid Range :	Value Interpretation
wa4701_RT_HS_ERR_NONE (1085)	Disable HS Errors
wa4701_RT_HS_ERR_WCO(1086)	Word Count
wa4701_RT_HS_ERR_FCS(1087)	Frame Check Sequence
wa4701_RT_HS_ERR_MANC (1088)	Manchester
Default:	wa4701_RT_HS_ERR_NONE

param_1

Variable Type: ViInt32

Control Name: param_1

Description: This parameter is used to specify values depending on the type of error chosen. The valid ranges for this parameter for the different error types is listed below.

Variable Type: ViInt32

Valid Values:	Error Type	Valid Range
	None	This parameter is ignored.
	Word Count	wa4701_RT_HS_ERR_WCO_MIN (-1) to wa4701_RT_HS_ERR_WCO_MAX (1).
	Frame Check Sequence	wa4701_RT_FCS_ENABLE(1100/Enable) or wa4701_RT_FCS_DISABLE (1101/Disable).
	Manchester	wa4701_RT_HS_ERR_BITPOS_MIN (0) to wa4701_RT_HS_ERR_BITPOS_MAX (65535).

param_2

Variable Type: ViInt32

Control Name: param_2

Description: This parameter is used to values depending on the type of error chosen. The valid ranges for this parameter for the different error types is listed below.

Variable Type: ViInt32

Valid Values:	Error Type	Valid Range
	None	This parameter is ignored.
	Word Count	This parameter is ignored
	Frame Check Sequence	This parameter is ignored
	Manchester	wa4701_RT_HS_ERR_LEVEL_MIN (0) to wa4701_RT_HS_ERR_LEVEL_MAX (1).

param_3

Variable Type: ViInt32

Control Name: param_3

Description: This parameter is used to specify values depending on the type of error chosen. The valid ranges for this parameter for the different error types is listed below.

Variable Type: ViInt32

Valid Values:	<i>Error Type</i>	<i>Valid Range</i>
	None	This parameter is ignored.

Word Count	This parameter is ignored
Frame Check Sequence	This parameter is ignored
Manchester	wa4701_RT_HS_ERR_MANC_PRE (1090/Preamble) wa4701_RT_HS_ERR_MANC_SDEL(1091/Start Delimiter) wa4701_RT_HS_ERR_MANC_FCPA(1092/FCPA) wa4701_RT_HS_ERR_MANC_INFO(1093/INFO) wa4701_RT_HS_ERR_MANC_FCS (1094/FCS) wa4701_RT_HS_ERR_MANC_EDEL(1095/End Delimiter)

Return Value

Control Name: status.

Description: Displays the results of the function call.

Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_setupRTLSErrInj

Function:

ViStatus wa4701_setupRTLSErrInj (ViSession instrHndl, ViInt16 RTNumber, ViInt16 uniqueOrGlobalSAFlag, ViInt16 subAddress, ViInt16 errorType, ViInt16 param_1, ViInt16 param_2);

Panel Name: Setup RT Low Speed Err Inject

Purpose:

This function is used to inject errors into the low speed words of the specified unique or global low speed subaddress.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument. This parameter gets its value from the Initialise function.
Variable Type: ViSession
Valid Range: -
Default Value: -

RTNumber

Variable Type: ViInt16
Control Name: RTNumber
Description: This parameter specifies the RT Number for which errors are to be injected into low speed words of the low speed subaddress.
Variable Type: ViInt16.
Valid Range: wa4701_RT_NUMBER_MIN (0)
wa4701_RT_NUMBER_MAX (31)
Default: -

uniqueOrGlobalSAFlag

Variable Type: ViInt16

Control Name: uniqueOrGlobalSAFlag
Description: This parameter is used to specify whether the subaddress for which errors are to be injected is a unique or global low speed subaddress.
Variable Type: ViInt16.
Valid Range: wa4701_CONFIGURE_FOR_UNIQUE (2205 / Unique) or
wa4701_CONFIGURE_FOR_GLOBAL (2206 / Global)
Default: wa4701_CONFIGURE_FOR_UNIQUE

subAddress

Variable Type: ViInt16
Control Name: subAddress
Description: This parameter specifies the Low Speed Subaddress where the errors are to be injected into the low speed words.
Variable Type: ViInt16.
Valid Range: wa4701_RT_SA_MIN (1)
wa4701_RT_SA_MAX (30)
Default: wa4701_RT_SA_MIN (1)

Note: If the `uniqueOrGlobal` flag specifies a Global Subaddress, then the value of this parameter is ignored by the function. The range check is done only for unique low speed subaddresses.

errorType

Variable Type: ViInt16
Control Name: errorType
Description: This parameter specifies the type of error that is to be injected into the low speed words of the specified unique/global low speed subaddress.
Variable Type: ViInt16.
Valid Range

Value	Interpretation
wa4701_RT_ERR_NONE	Disable LS Errors
wa4701_RT_ERR_PARITY	Parity
wa4701_RT_ERR_SYNC	Synchro
wa4701_RT_ERR_MANCH	Manchester
wa4701_RT_ERR_WRD_CNT	Word Count
wa4701_RT_ERR_WRD_LEN	Wrong Length
wa4701_RT_ERR_WRNG_BUS	Wrong Bus
wa4701_RT_ERR_BOTH_BUS	Both Buses

Default: wa4701_RT_ERR_NONE

param_1

Variable Type: ViInt16
Control Name: param_1
Description: This parameter is used to specify the additional parameter required to be input in order to set up a low speed error injection. This parameter is expected to contain values depending on the type of error chosen. The valid ranges for the parameter for the different error types is listed below.

Variable Type: ViInt16
Valid Values:

Error Type	Valid Range
None	This parameter is ignored.
Parity	wa4701_RT_ERR_WNUM_MIN (0) to wa4701_RT_ERR_WNUM_MAX (32).
Synchro	wa4701_RT_ERR_WNUM_MIN (0) to wa4701_RT_ERR_WNUM_MAX (32).
Manchester	wa4701_RT_ERR_WNUM_MIN (0) to wa4701_RT_ERR_WNUM_MAX (32).
Word Count	wa4701_RT_ERR_WRD_CNT_MIN (-63) to wa4701_RT_ERR_WRD_CNT_MAX (63).
Word Length	wa4701_RT_ERR_WNUM_MIN (0) to wa4701_RT_ERR_WNUM_MAX (32).
Wrong Bus	This parameter is ignored.
Both Buses	This parameter is ignored.

param_2

Variable Type: ViInt16
Control Name: param_2
Description: This is an additional parameter required depending on the type of error chosen. The valid ranges for this parameter for the different error types is listed below.

Variable Type: ViInt16
Valid Values:

Error Type	Valid Range
None	This parameter is ignored.
Parity	This parameter is ignored.
Synchro	wa4701_RT_ERR_SYNC_PATTERN_MIN (1) to wa4701_RT_ERR_SYNC_PATTERN_MAX (63).

Manchester	wa4701_RT_ERR_MANCH_BIT_POS_MIN (0) to wa4701_RT_ERR_MANCH_BIT_POS_MAX (31).
Word Count	This parameter is ignored.
Word Length	wa4701_RT_ERR_WRD_LEN_MIN (-8) to wa4701_RT_ERR_WRD_LEN_MAX (8).
Wrong Bus	This parameter is ignored.
Both Buses	This parameter is ignored.

Return Value

- Control Name:** status.
- Description:** Displays the results of the function call.
- Variable Type:** ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_setupRTParams

Function:

ViStatus wa4701_setupRTParams (ViSession instrHndl, ViInt16 RTNumber, ViInt32 lsStatusWord, ViInt32 hsStatusWord, ViInt32 lsBitWord, ViInt32 hsBitWord);

Panel Name: Setup RT Parameters

Purpose:

This function is used to set up the Low Speed and High Speed Status Words and the Low Speed and High Speed BIT (Built In Test) Words.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument.
Valid Range: -
Default Value: -

RTNumber

Variable Type: ViInt16
Control Name: RTNumber.
Description: This parameter is used to specify the RT Number of the RT whose Status and BIT words are configured.
Variable Type: ViInt16.
Valid Range: wa4701_RT_NUMBER_MIN (0)
wa4701_RT_NUMBER_MAX (31)
Default: wa4701_RT_NUMBER_MIN (0)

lsStatusWord

Variable Type: ViInt32
Control Name: lsStatusWord
Description: This parameter specifies the low speed Status Word to be associated with the specified RT. This low speed status word will be transmitted by the simulated RT in response to all messages.
Variable Type: ViInt32.
Valid Range: wa4701_RT_LS_STATUS_WRD_MIN (0)
wa4701_RT_LS_STATUS_WRD_MAX (65535)
Default: wa4701_RT_LS_STATUS_WRD_MIN (0)

hsStatusWord

Variable Type: ViInt32
Control Name: hsStatusWord
Description: This parameter specifies the high speed Status Word to be associated with the specified RT. This high speed status word will be transmitted by the simulated RT in response to all messages.
Variable Type: ViInt32.
Valid Range: wa4701_RT_HS_STATUS_WRD_MIN (0)
wa4701_RT_HS_STATUS_WRD_MAX (65535)
Default: wa4701_RT_HS_STATUS_WRD_MIN (0)

lsBitWord

Variable Type: ViInt32
Control Name: lsBitWord
Description: This parameter specifies the low speed BIT word (Built In Test) to be returned by the simulated RT in response to the appropriate mode command. The value transmitted by the external RT when IIB-3910-VXI is not simulating the RT is not stored. If the RT number specified to this function is not simulated by the instrument, then specifying this parameter has no effect on the system.
Variable Type: ViInt32.
Valid Range: wa4701_RT_LS_BIT_WRD_MIN (0)
wa4701_RT_LS_BIT_WRD_MAX (65535)
Default: wa4701_RT_LS_BIT_WRD_MIN (0)

hsBitWord

Variable Type: ViInt32
Control Name: hsBitWord
Description: This parameter specified the high speed BIT word (Built In Test) to be returned by the simulated RT in response to a transmit command from a high speed subaddress with a word count of 3 or more. The value transmitted by the external RT when IIB-3910-VXI is not simulating the RT is not stored. So if the RT number specified to this function is not simulated by the instrument, then specifying this parameter has no effect on the system.
Variable Type: ViInt32.
Valid Range: wa4701_RT_HS_BIT_WRD_MIN (0)
wa4701_RT_HS_BIT_WRD_MAX (65535)
Default: wa4701_RT_HS_BIT_WRD_MIN (0)

Return Value

Control Name: status.
Description: Displays the results of the function call.
Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_setupRTSubAddressParams

Function:

ViStatus wa4701_setupRTSubAddressParams (ViSession instrHndl, ViInt16 RTNumber, ViInt16 lsOrHsFlag, ViInt16 subAddressOrHSID, ViInt16 setAliasOrWrapAndUniqueSAFlag, ViInt16 uniqueOrGlobalFlag, ViInt16 setWrapFlag, ViInt16 aliasList[], ViInt16 sizeOfAliasList, ViInt16 HSIDBufferSize);

Panel Name: Setup RT Subaddress Parameters

Purpose:

This function is used to set up the subaddress parameters of the specified RT. The function allows configuration of either the global or unique low speed subaddress or high speed identifier. It configures the Alias List, the Wrap Around facility of the data buffers and configures the specified low speed subaddress/high speed identifier as unique based on the user input.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument. This parameter gets its value from the Initialise function.
Variable Type: ViSession
Valid Range: -
Default Value: -

RTNumber

Variable Type: ViInt16
Control Name: RTNumber.
Description: This parameter is used to specify the RT Number of the RT whose subaddress parameters are to be configured.
Variable Type: ViInt16.
Valid Range: wa4701_RT_NUMBER_MIN (0)
wa4701_RT_NUMBER_MAX (31)
Default: -

lsOrHsFlag

Variable Type: ViInt16
Control Name: lsOrHsFlag
Description: This parameter is used to select the low speed or high speed subaddress for configuration.
Variable Type: ViInt16.
Valid Range:

Value	Interpretation
wa4701_RT_LS (1074)	Unique/Global Low Speed Subaddress
wa4701_RT_HS (1075)	Unique/Global High Speed Identifier

Default Value: wa4701_RT_LS

subAddressOrHSID

Variable Type: ViInt16
Control Name: subAddressOrHSID
Description: This parameter specifies the low speed subaddress or high speed identifier for which the various subaddress parameters are to be set.
Variable Type: ViInt16
Valid Range: For Low Speed Subaddress

wa4701_RT_SA_MIN (1) to
wa4701_RT_SA_MAX (30),

For High Speed Identifier
wa4701_RT_HSID_MIN (1) to
wa4701_RT_HSID_MAX (127).

Default Value: -

Note : *This parameter is ignored by the function if the `uniqueOrGlobalFlag' is set for global low speed subaddress/high speed identifier.*

setAliasOrWrapAndUniqueSAFlag

Variable Type: ViInt16

Control Name: setAliasOrWrapAndUniqueSAFlag

Description: This parameter selects one of the following to be executed by the function,

a) Configure Alias list to the specified Subaddress.

OR

b) Switch On/Off the Wrap around feature to the selected Subaddress & enable or disable the selected Subaddress as a unique Subaddress.

Variable Type: ViInt16

Valid Range: wa4701_SET_WRAP_AND_SA_UNIQUE (2200) or
wa4701_SET_ALIAS_LIST (2201).

Default Value: wa4701_SET_WRAP_AND_SA_UNIQUE

Note: *In any one execution, the function will EITHER set up the Alias List for the specified RT (or) it will set the `Wrap Around' facility and switch On/OFF the specified low speed subaddress/high speed identifier as unique. This execution path can be decided by specifying the appropriate value to this parameter.*

uniqueOrGlobalFlag

Variable Type: ViInt16

Control Name: uniqueOrGlobalFlag

Description: This parameter specifies whether the specified RT's subaddress parameters are to be configured for the global low speed subaddress/high speed identifier (or) for a unique low speed subaddress/high speed identifier.

Variable Type: ViInt16

Valid Range: wa4701_CONFIGURE_FOR_UNIQUE (2205 / Unique) or
wa4701_CONFIGURE_FOR_GLOBAL.(2206 / Global)

Default Value: wa4701_CONFIGURE_FOR_UNIQUE

setWrapFlag

Variable Type: iInt16

Control Name: setWrapFlag

Description: This parameter specifies whether the `Wrap Around' feature is to be set ON or OFF for the specified low speed subaddress or high speed identifier.

Variable Type: ViInt16

Valid Range: wa4701_SET_WRAP_ON (2202 / Set Wrap On) or
wa4701_SET_WRAP_OFF (2203 / Set Wrap Off)

Default Value: wa4701_SET_WRAP_OFF

Note: *If the `setAliasOrWrapAndUniqueSAFlag' specifies that the configuration to be done is setting up the Alias List, then this parameter is ignored by the function.*

aliasList

Variable Type: ViInt16 []
Control Name: aliasList
Description: This parameter should contain a List of all the subaddresses which are to be aliased to the specified unique/global low speed or subaddress/high speed identifier.
Variable Type: ViInt16 []
Valid Range: The individual elements of the input array have to be valid low speed subaddresses/high speed identifiers.
For Low Speed Subaddress
wa4701_RT_SA_MIN (1) to
wa4701_RT_SA_MAX (30),

For High Speed Identifier
wa4701_RT_HSID_MIN (1) to
wa4701_RT_HSID_MAX (127).
Default Value: -

Note: (1) It must be ensured that the subaddresses specified in the input array are not duplicated.
(2) This parameter is ignored by the function if the `setAliasOrWrapAndUniqueSAFlag' is set for configuring the Wrap and Unique Subaddress.

sizeofAliasList

Variable Type: ViInt16
Control Name: sizeofAliasList
Description: This parameter specifies the number of elements in the `aliasList' input array.
Variable Type: ViInt16
Valid Range: -
Default Value: -

HSIDBuffSize

Variable Type: ViInt16
Control Name: HSIDBuffSize
Description: This parameter specifies the size of the transmit & receive data buffer associated with the high speed subaddress (identifier).
Variable Type: ViInt16
Valid Range: wa4701_HSID_BUFF_SIZE_MIN (32) to
wa4701_HSID_BUFF_SIZE_MAX (4096).
Default Value: -

Return Value

Control Name: status.
Description: Displays the results of the function call.
Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_setupRTTimingParams

Function:

ViStatus wa4701_setupRTTimingParams (ViSession instrHndl, ViInt16 RTNumber, ViInt16 subAddressOrHSID, ViInt16 lsOrHsFlag, ViInt16 uniqueOrGlobalFlag, ViInt16 rtmicroseconds, ViInt16 timicroseconds, ViInt16 riOutmicroseconds, ViInt16 npr);

Panel Name: Setup RT Timing Parameters

Purpose:

This function configures the timing parameters related with the specified RT for either all the low speed subaddresses/high speed identifiers (or) for a single unique low speed subaddress/high speed identifier.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument. This parameter gets its value from the Initialise function.
Variable Type: ViSession
Valid Range: -
Default Value: -

RTNumber

Variable Type: ViInt16
Control Name: RTNumber.
Description: This parameter is used to specify the RT Number of the RT which is to be configured for the timing parameters.
Variable Type: ViInt16.
Valid Range: wa4701_RT_NUMBER_MIN (0)
wa4701_RT_NUMBER_MAX (31)
Default: -

subAddressOrHSID

Variable Type: ViInt16
Control Name: subAddressOrHSID
Description: This parameter specifies the low speed subaddress or high speed identifier for which the various timing parameters are to be configured.
Variable Type: ViInt16
Valid Range: For Low Speed Subaddress
wa4701_RT_SA_MIN (1) to
wa4701_RT_SA_MAX (30),

For High Speed Identifier
wa4701_RT_HSID_MIN (1) to
wa4701_RT_HSID_MAX (127).
Default Value: -

Note :

- (1) This parameter is ignored by the function if the `uniqueOrGlobalFlag` is set for global low speed subaddress/high speed identifier.
- (2) If the `lsOrHsFlag` specifies a low speed transfer and the `uniqueOrGlobalFlag` specifies a unique low speed subaddress, then the Response Time is set for the specified low speed subaddress. On the other hand, if the `lsOrHsFlag` specifies high speed transfer and the `uniqueOrGlobalFlag`

specifies a unique HSID, then the Transmitter Initialise, Receiver Initialisation TimeOut and the Number Of Preamble Bits will be set for the specified HSID.

- (3) *If the `uniqueOrGlobalFlag' specifies a global subaddress/HSID, then all the timing parameters will be applied globally to the specified RT.*

lsOrHsFlag

Variable Type: ViInt16
Control Name: lsOrHsFlag.
Description: This parameter is used to indicate whether the Response Time is to be set for the specified low speed subaddress (or) whether the Transmitter Initialise, Receiver Initialisation TimeOut and Number Of Preamble Bits are to be set for the high speed identifier.

Variable Type: ViInt16.
Valid Range:

Value	Interpretation
wa4701_RT_LS (1074)	Low Speed Subaddress
wa4701_RT_HS (1075)	High Speed Identifier

Default Value: wa4701_RT_LS

uniqueOrGlobalFlag

Variable Type: ViInt16
Control Name: uniqueOrGlobalFlag
Description: This parameter specifies whether the Response Time, the Transmitter Initialise Time, the Receiver Initialisation TimeOut and Number Of Preamble Bits are to be set globally for the specified RT (or) the same timing parameters are to be applied to a unique low speed subaddress/high speed identifier.

Variable Type: ViInt16
Valid Range: wa4701_CONFIGURE_FOR_UNIQUE (2205 / Unique) or wa4701_CONFIGURE_FOR_GLOBAL.(2206 / Global)
Default Value: wa4701_CONFIGURE_FOR_UNIQUE.

rtimmicroseconds

Variable Type: ViInt16
Control Name: rtim
Description: This parameter specifies the Response Time to be set either globally for the specified RT or for a unique low speed subaddress depending on the value specified for the `uniqueOrGlobalFlag' and `lsOrHsFlag' input parameters.

Variable Type: ViInt16
Valid Range: wa4701_RT_TIM_RTIM_MIN (20 microseconds) to wa4701_RT_TIM_RTIM_MAX (255 microseconds).

Note: *This function interprets the value specified to this parameter to be in microseconds. For example, if a value of 20 was input to this parameter, then the response time to be set is interpreted as 20 microseconds.*

timicroseconds

Variable Type: ViInt16
Control Name: ti
Description: This parameter specifies the Transmitter Initialise time to be configured either globally for the simulated RT for all the high speed identifiers or for a unique high speed identifier depending on the value specified for the `uniqueOrGlobalFlag' and `lsOrHsFlag' input parameters.

Variable Type: ViInt16
Valid Range: wa4701_RT_TIM_TI_MIN (20 microseconds) to wa4701_RT_TIM_TI_MAX (255 microseconds).

Note: *This function interprets the value specified to this parameter to be in microseconds. For example, if a value of 20 was input to this parameter, then the Transmitter Initialise Time to be set is interpreted as 20 microseconds.*

riOutmicroseconds

Variable Type: ViInt16

Control Name: riOut

Description: This parameter specifies the Receiver Initialisation Timeout to be configured either globally for the simulated RTs of all the high speed identifiers or for a unique high speed identifier depending on the values specified for 'uniqueOrGlobalFlag' and 'IsOrHsFlag' input parameters.

Variable Type: ViInt16

Valid Range: wa4701_RT_TIM_RIOUT_MIN (20 microseconds) to wa4701_RT_TIM_RIOUT_MAX (255 microseconds).

Note: *This function interprets the value specified to this parameter to be in microseconds. For example, if a value of 20 was input to this parameter, then the Receiver Initialisation Timeout to be set is interpreted as 20 microseconds.*

npr

Variable Type: ViInt16

Control Name: npr

Description: This parameter specifies the Number Of Preamble Bits to be set either globally for the simulated RTs for all high speed identifiers or for a unique high speed identifier depending on the values specified for the 'uniqueOrGlobalFlag' and 'IsOrHsFlag' input parameters.

Variable Type: ViInt16

Valid Range: wa4701_RT_TIM_NPR_MIN (2 Bits) to wa4701_RT_TIM_NPR_MAX (255 Bits).

Default Value: -

Return Value

Control Name: status.

Description: Displays the results of the function call.

Variable Type: ViStatus

Note: *Use wa4701_errorMessage() function to retrieve the error message corresponding to the error code returned by this function.*

wa4701_setupScapAndWindParams

Function:

ViStatus wa4701_setupScapAndWindParams (ViSession instrHndl,
ViInt16 armOrTriggerFlag, ViInt16 scapOrWindowFlag,
ViInt16 layerNum, ViInt16 state, ViInt32 value);

Panel Name: Setup Scapture/Window Params

Purpose:

This function is used to configure the Selective Capture/Window Mode for the specified Arm/Trigger Event Detection Layer of the Chronological Monitor.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument. This parameter gets its value from the Initialise function.
Variable Type: ViSession
Valid Range: -
Default: -

armOrTriggerFlag

Variable Type: ViInt16
Control Name: armOrTriggerFlag.
Description: This parameter specifies whether the Arm or Trigger Event Detection Layer is being configured for the Selective Capture Mode or Window Mode.
Variable Type: ViInt16.
Valid Range: wa4701_CONF_CM_ARM(1027 / Arm) or
wa4701_CONF_CM_TRIGGER(1028 / Trigger).
Default Value: wa4701_CONF_CM_ARM.

scapOrWindowFlag

Variable Type: ViInt16
Control Name: scapOrWindowFlag
Description: This parameter is a flag which indicates whether Selective Capture or WINDOW mode is to be configured for the specified Arm/Trigger Event Detection layer
Variable Type: ViInt16.
Valid Range: wa4701_SETUP_SCAP(1052 / Selective Capture)
wa4701_SETUP_WIND(1053 / Window Mode)
Default Value: wa4701_SETUP_SCAP

layerNum

Variable Type: ViInt16
Control Name: layerNum.
Description: This parameter specifies the layer number for which the Selective Capture/Window Mode is to be configured.
Variable Type: ViInt16.
Valid Range: wa4701_CONF_CM_LAYER1 (1 / Layer 1)
wa4701_CONF_CM_LAYER2 (2 / Layer 2)
wa4701_CONF_CM_LAYER3 (3 / Layer 3).
Default Value: wa4701_CONF_CM_LAYER1.

state

Variable Type: ViInt16
Control Name: state
Description: This parameter specifies whether the Selective Capture/Window Mode for the specified Arm/Trigger is to be switched on or off..
Variable Type: ViInt16.
Valid Range: wa4701_SETUP_SCAP_WIND_STAT_ON (1055 / Set ON)
wa4701_SETUP_SCAP_WIND_STAT_OFF(1056 / Set OFF)
Default Value: wa4701_SETUP_SCAP_WIND_STAT_ON

value

Variable Type: ViInt32
Control Name: value
Description: When SCAPture mode is set ON, then this parameter specifies the number of the messages to be stored before the event detector is satisfied. When WINDow mode is set ON, then this parameter specifies the position in the message where the CM looks for WORD2 in the message.
Variable Type: ViInt32.
Valid Range: For SCAPture mode
wa4701_SETUP_SCAP_CNT_MIN (0)
wa4701_SETUP_SCAP_CNT_MAX (32768)

For WINDow mode
wa4701_SETUP_WIN_POS_MIN (0),
wa4701_SETUP_WIN_POS_MAX (33)
Default Value: -

Return Value

Control Name: status.
Description: Displays the results of the function call.
Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_startStopBC

Function:

ViStatus wa4701_startStopBC (ViSession instrHndl, ViInt16 frameNum, ViInt16 startStopFlag, ViInt32 frameCount);

Panel Name: Start/Stop BC

Purpose:

This function starts/stops the BC transmission. It sets up the BC to either actively transmit messages or stop the message transmission on the bus.

Parameter List

instrHndl

Variable Type: ViSession
Control Name: instrHndl.
Description: A unique session handle to the instrument. This parameter gets its value from the Initialise function.
Variable Type: ViSession
Valid Range: -
Default Value: -

frameNum

Variable Type: ViInt16
Control Name: frameNum
Description: This parameter specifies the frame number of the frame to be transmitted.
Variable Type: ViInt16.
Valid Range: wa4701_BC_FRAME_NUM_MIN (1) to numOfBCFrames.
where `numOfBCFrames' is the output obtained from
`wa4701_getTransParamsNVRAM()' function.
Default Value: -

startStopFlag

Variable Type: ViInt16
Control Name: startStopFlag
Description: This parameter is used to indicate whether the BC is to start transmitting the frames on the bus or to stop any currently active transmission. If the BC is not transmitting when this function instructs it to stop, then the function has no effect on the instrument.
Variable Type: ViInt16.
Valid Range:

Value	Interpretation
wa4701_BC_FRAME_START (1238)	Start transmission
wa4701_BC_FRAME_STOP(1239)	Stop transmission

Default Value: wa4701_BC_FRAME_START

frameCount

Variable Type: ViInt32
Control Name: frameCount
Description: This parameter specifies the number of times the specified frame is to be transmitted over the bus. If a value of `wa4701_BC_FRAME_COUNT_MAX' is specified, then the value equates to `MAX' and the specified frame will be transmitted `forever' over the bus.
Variable Type: ViInt16.
Valid Range: wa4701_BC_FRAME_COUNT_MIN (0) to wa4701_BC_FRAME_COUNT_MAX.(65537)
Default Value: 1

Return Value**Control Name:** status.**Description:** Displays the results of the function call.**Variable Type:** ViStatus**Note:** Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.

wa4701_startStopCMAcq

Function:

ViStatus wa4701_startStopCMAcq (ViSession instrHndl, ViInt16 startOrStopFlag);

Panel Name: Start/Stop CM Acquisition

Purpose:

This function enables or disables the CM from monitoring the bus.

Parameter List

instrHndl

Variable Type: ViSession

Control Name: instrHndl.

Description: A unique session handle to the instrument. This parameter gets its value from the Initialise function,

Variable Type: ViSession

Valid Range: -

Default: -

startOrStopFlag

Variable Type: ViInt16

Control Name: startOrStopFlag

Description: This parameter specifies whether the CM is to start actively monitoring the bus traffic (or) whether it should stop a currently active monitoring process.

Variable Type: ViInt16.

Valid Range: wa4701_CM_START (1070 / Start monitoring bus)
wa4701_CM_STOP(1071 / Stop monitoring bus)

Default Value: wa4701_CM_START.

Return Value

Control Name: status.

Description: Displays the results of the function call.

Variable Type: ViStatus

Note: Use *wa4701_errorMessage()* function to retrieve the error message corresponding to the error code returned by this function.