

2015 Test beam Run Control

Generated by Doxygen 1.8.9.1

Fri Jul 24 2015 08:32:19

Contents

1	Module Index	1
1.1	Modules	1
2	Namespace Index	3
2.1	Namespace List	3
3	Hierarchical Index	5
3.1	Class Hierarchy	5
4	Data Structure Index	7
4.1	Data Structures	7
5	Module Documentation	9
5.1	Socket communication objects	9
5.1.1	Detailed Description	9
5.1.2	Enumeration Type Documentation	9
5.1.2.1	SocketType	9
6	Namespace Documentation	11
6.1	VME Namespace Reference	11
6.1.1	Typedef Documentation	13
6.1.1.1	CFDCollection	13
6.1.1.2	TDCCollection	13
6.1.1.3	TDCEventCollection	13
6.1.2	Enumeration Type Documentation	13
6.1.2.1	AcquisitionMode	13
6.1.2.2	BridgeType	13
6.1.2.3	CAENETControllerV288Register	13
6.1.2.4	CFDV812Register	13
6.1.2.5	DetectionMode	14
6.1.2.6	FPGAUnitV1495Register	14
6.1.2.7	IOModuleV262Register	15
6.1.2.8	micro_handshake	15

6.1.2.9	TDCV1x90Register	15
6.1.2.10	trailead_edge_lsb	16
6.1.2.11	trig_conf	16
6.2	VME::TDCV1x90Opcodes Namespace Reference	16
6.2.1	Function Documentation	18
6.2.1.1	AUTOLOAD_DEF_CONFI	18
6.2.1.2	AUTOLOAD_USER_CONF	18
6.2.1.3	CLEAR_KEEP_TOKEN	18
6.2.1.4	CONT_STOR	18
6.2.1.5	DEFAULT_SETUP_REG	18
6.2.1.6	DIS_ALL_CHANNEL	18
6.2.1.7	DIS_CHANNEL	18
6.2.1.8	DIS_ERROR_BYPASS	18
6.2.1.9	DIS_ERROR_MARK	18
6.2.1.10	DIS_HEAD_TRAILER	18
6.2.1.11	DIS_SUB_TRG	18
6.2.1.12	DISABLE_TEST_MODE	18
6.2.1.13	EN_ALL_CHANNEL	18
6.2.1.14	EN_CHANNEL	18
6.2.1.15	EN_ERROR_BYPASS	18
6.2.1.16	EN_ERROR_MARK	18
6.2.1.17	EN_HEAD_TRAILER	18
6.2.1.18	EN_SUB_TRG	18
6.2.1.19	ENABLE_TEST_MODE	18
6.2.1.20	LOAD_DEF_CONFIG	18
6.2.1.21	LOAD_USER_CONFIG	18
6.2.1.22	READ_ACQ_MOD	18
6.2.1.23	READ_ADJUST_CH	18
6.2.1.24	READ_DEAD_TIME	18
6.2.1.25	READ_DETECTION	18
6.2.1.26	READ_DLL_LOCK	18
6.2.1.27	READ_EEPROM	18
6.2.1.28	READ_EN_PATTERN	19
6.2.1.29	READ_EN_PATTERN32	19
6.2.1.30	READ_ERROR_STATUS	19
6.2.1.31	READ_ERROR_TYPES	19
6.2.1.32	READ_EVENT_SIZE	19
6.2.1.33	READ_FIFO_SIZE	19
6.2.1.34	READ_GLOB_OFFS	19
6.2.1.35	READ_HEAD_TRAILER	19

6.2.1.36	READ_MICRO_REV	19
6.2.1.37	READ_RC_ADJ	19
6.2.1.38	READ_RES	19
6.2.1.39	READ_SETUP_REG	19
6.2.1.40	READ_SETUP_SCANPATH	19
6.2.1.41	READ_SPARE	19
6.2.1.42	READ_STATUS_STREAM	19
6.2.1.43	READ_TDC_ID	19
6.2.1.44	READ_TRG_CONF	19
6.2.1.45	RESET_DLL_PLL	19
6.2.1.46	REV_DATE_MICRO_FW	19
6.2.1.47	SAVE_RC_ADJ	19
6.2.1.48	SAVE_USER_CONFIG	19
6.2.1.49	SET_ADJUST_CH	19
6.2.1.50	SET_DEAD_TIME	19
6.2.1.51	SET_DETECTION	19
6.2.1.52	SET_DLL_CLOCK	19
6.2.1.53	SET_ERROR_TYPES	19
6.2.1.54	SET_EVENT_SIZE	19
6.2.1.55	SET_FIFO_SIZE	19
6.2.1.56	SET_GLOB_OFFS	20
6.2.1.57	SET_KEEP_TOKEN	20
6.2.1.58	SET_PAIR_RES	20
6.2.1.59	SET_RC_ADJ	20
6.2.1.60	SET_REJ_MARGIN	20
6.2.1.61	SET_SW_MARGIN	20
6.2.1.62	SET_TDC_TSET_OUTPUT	20
6.2.1.63	SET_TR_LEAD_LSB	20
6.2.1.64	SET_WIN_OFFS	20
6.2.1.65	SET_WIN_WIDTH	20
6.2.1.66	TRG_MATCH	20
6.2.1.67	UPDATE_SETUP_REG	20
6.2.1.68	UPDATE_SETUP_TDC	20
6.2.1.69	WRITE_EEPROM	20
6.2.1.70	WRITE_EN_PATTERN	20
6.2.1.71	WRITE_EN_PATTERN32	20
6.2.1.72	WRITE_SETUP_REG	20
6.2.1.73	WRITE_SPARE	20

7.1	VME::BridgeVx718 Class Reference	21
7.1.1	Detailed Description	23
7.1.2	Member Enumeration Documentation	23
7.1.2.1	IRQId	23
7.1.3	Constructor & Destructor Documentation	23
7.1.3.1	BridgeVx718	23
7.1.3.2	~BridgeVx718	24
7.1.4	Member Function Documentation	24
7.1.4.1	CheckConfiguration	24
7.1.4.2	CheckPCIIInterface	24
7.1.4.3	GetHandle	24
7.1.4.4	GetIRQStatus	24
7.1.4.5	GetStatus	24
7.1.4.6	InputConf	24
7.1.4.7	InputRead	25
7.1.4.8	OutputConf	25
7.1.4.9	OutputOff	25
7.1.4.10	OutputOn	25
7.1.4.11	Reset	25
7.1.4.12	SetIRQ	25
7.1.4.13	SinglePulse	25
7.1.4.14	StartPulser	25
7.1.4.15	StopPulser	25
7.1.4.16	TestOutputs	25
7.1.4.17	WaitIRQ	25
7.1.5	Field Documentation	25
7.1.5.1	fHasIRQ	25
7.2	VME::BridgeVx718Control Class Reference	26
7.2.1	Constructor & Destructor Documentation	26
7.2.1.1	BridgeVx718Control	26
7.2.1.2	~BridgeVx718Control	26
7.2.2	Member Function Documentation	26
7.2.2.1	GetAddressIncrement	26
7.2.2.2	GetArbiterType	26
7.2.2.3	GetBusReqLevel	27
7.2.2.4	GetBusTimeout	27
7.2.2.5	GetInterruptReq	27
7.2.2.6	GetReleaseType	27
7.2.2.7	GetRequesterType	27
7.2.2.8	GetSysRes	27

7.2.3	Field Documentation	27
7.2.3.1	fWord	27
7.3	VME::BridgeVx718Status Class Reference	27
7.3.1	Constructor & Destructor Documentation	28
7.3.1.1	BridgeVx718Status	28
7.3.1.2	~BridgeVx718Status	28
7.3.2	Member Function Documentation	28
7.3.2.1	Dump	28
7.3.2.2	GetBERR	28
7.3.2.3	GetDipSwitch	28
7.3.2.4	GetDTACK	28
7.3.2.5	GetSystemControl	28
7.3.2.6	GetSystemReset	28
7.3.2.7	GetUSBType	28
7.3.3	Field Documentation	29
7.3.3.1	fWord	29
7.4	VME::CAENETControllerV288 Class Reference	29
7.4.1	Detailed Description	30
7.4.2	Constructor & Destructor Documentation	30
7.4.2.1	CAENETControllerV288	30
7.4.2.2	~CAENETControllerV288	30
7.4.3	Member Function Documentation	30
7.4.3.1	GetStatus	30
7.4.3.2	Receive	31
7.4.3.3	Send	31
7.4.3.4	WaitForResponse	31
7.4.4	Friends And Related Function Documentation	31
7.4.4.1	operator<<	31
7.4.4.2	operator>>	31
7.4.5	Field Documentation	31
7.4.5.1	fNumWordsInBuffer	31
7.5	VME::CAENETControllerV288Status Class Reference	31
7.5.1	Member Enumeration Documentation	32
7.5.1.1	OperationStatus	32
7.5.2	Constructor & Destructor Documentation	32
7.5.2.1	CAENETControllerV288Status	32
7.5.2.2	~CAENETControllerV288Status	32
7.5.3	Member Function Documentation	32
7.5.3.1	GetOperationStatus	32
7.5.4	Field Documentation	32

7.5.4.1	fWord	32
7.6	VME::CFDV812 Class Reference	32
7.6.1	Detailed Description	34
7.6.2	Constructor & Destructor Documentation	34
7.6.2.1	CFDV812	34
7.6.2.2	~CFDV812	34
7.6.3	Member Function Documentation	34
7.6.3.1	CheckConfiguration	34
7.6.3.2	DeadTimeCalculator	34
7.6.3.3	GetFixedCode	35
7.6.3.4	GetManufacturerId	35
7.6.3.5	GetModuleType	35
7.6.3.6	GetModuleVersion	36
7.6.3.7	GetSerialNumber	36
7.6.3.8	OutputWidthCalculator	36
7.6.3.9	SetDeadTime	36
7.6.3.10	SetOutputWidth	37
7.6.3.11	SetPOI	38
7.6.3.12	SetThreshold	38
7.7	Client Class Reference	39
7.7.1	Detailed Description	40
7.7.2	Constructor & Destructor Documentation	40
7.7.2.1	Client	40
7.7.2.2	Client	40
7.7.2.3	~Client	41
7.7.3	Member Function Documentation	41
7.7.3.1	Announce	41
7.7.3.2	Connect	41
7.7.3.3	Disconnect	42
7.7.3.4	GetType	42
7.7.3.5	ParseMessage	42
7.7.3.6	Receive	42
7.7.3.7	Send	43
7.7.3.8	Send	43
7.7.3.9	SendAndReceive	44
7.7.4	Field Documentation	44
7.7.4.1	fClientId	44
7.7.4.2	fIsConnected	44
7.7.4.3	fType	44
7.8	file_header_t Struct Reference	44

7.8.1	Detailed Description	44
7.8.2	Field Documentation	45
7.8.2.1	acq_mode	45
7.8.2.2	det_mode	45
7.8.2.3	magic	45
7.8.2.4	num_hptdc	45
7.8.2.5	run_id	45
7.8.2.6	spill_id	45
7.9	FileReader Class Reference	45
7.9.1	Detailed Description	46
7.9.2	Constructor & Destructor Documentation	46
7.9.2.1	FileReader	46
7.9.2.2	~FileReader	46
7.9.3	Member Function Documentation	46
7.9.3.1	Dump	46
7.9.3.2	GetNextEvent	47
7.9.3.3	GetNextMeasurement	47
7.9.3.4	GetNumEvents	47
7.9.3.5	GetNumTDCs	47
7.9.4	Field Documentation	47
7.9.4.1	fFile	47
7.9.4.2	fHeader	47
7.9.4.3	fNumEvents	48
7.9.4.4	fReadoutMode	48
7.9.4.5	fWriteTime	48
7.10	VME::FPGAUnitV1495 Class Reference	48
7.10.1	Detailed Description	50
7.10.2	Member Enumeration Documentation	50
7.10.2.1	TDCBits	50
7.10.3	Constructor & Destructor Documentation	50
7.10.3.1	FPGAUnitV1495	50
7.10.3.2	~FPGAUnitV1495	50
7.10.4	Member Function Documentation	50
7.10.4.1	CheckBoardVersion	51
7.10.4.2	ClearOutputPulser	51
7.10.4.3	DumpFWInformation	51
7.10.4.4	GetCAENFirmwareRevision	52
7.10.4.5	GetControl	52
7.10.4.6	GetGeoAddress	52
7.10.4.7	GetHardwareRevision	53

7.10.4.8	GetInternalClockPeriod	53
7.10.4.9	GetInternalTriggerPeriod	53
7.10.4.10	GetOutputDelay	54
7.10.4.11	GetOutputPulser	54
7.10.4.12	GetScalerValue	54
7.10.4.13	GetSerialNumber	55
7.10.4.14	GetTDCBits	55
7.10.4.15	GetUserFirmwareRevision	55
7.10.4.16	PulseTDCBits	55
7.10.4.17	ResetFPGA	56
7.10.4.18	SetControl	56
7.10.4.19	SetInternalClockPeriod	56
7.10.4.20	SetInternalTriggerPeriod	57
7.10.4.21	SetOutputDelay	57
7.10.4.22	SetOutputPulser	58
7.10.4.23	SetOutputPulserPOI	58
7.10.4.24	SetTDCBits	58
7.10.4.25	StartScaler	58
7.10.4.26	StopScaler	59
7.10.5	Field Documentation	59
7.10.5.1	fScalerStarted	59
7.11	VME::FPGAUnitV1495Control Class Reference	59
7.11.1	Detailed Description	60
7.11.2	Member Enumeration Documentation	60
7.11.2.1	ClockSource	60
7.11.2.2	SignalSource	61
7.11.2.3	TriggerSource	61
7.11.3	Constructor & Destructor Documentation	61
7.11.3.1	FPGAUnitV1495Control	61
7.11.3.2	~FPGAUnitV1495Control	61
7.11.4	Member Function Documentation	61
7.11.4.1	Dump	61
7.11.4.2	GetBit	61
7.11.4.3	GetClockSource	61
7.11.4.4	GetScalerStatus	61
7.11.4.5	GetSignalSource	62
7.11.4.6	GetTriggerSource	62
7.11.4.7	GetWord	62
7.11.4.8	SetBit	62
7.11.4.9	SetClockSource	62

7.11.4.10 SetScalerReset	63
7.11.4.11 SetScalerStatus	63
7.11.4.12 SetSignalSource	63
7.11.4.13 SetTriggerSource	63
7.11.5 Field Documentation	64
7.11.5.1 fWord	64
7.12 VME::GenericBoard< Register, am > Class Template Reference	64
7.12.1 Constructor & Destructor Documentation	64
7.12.1.1 GenericBoard	64
7.12.1.2 ~GenericBoard	64
7.12.2 Member Function Documentation	65
7.12.2.1 ReadRegister	65
7.12.2.2 ReadRegister	65
7.12.2.3 WriteRegister	65
7.12.2.4 WriteRegister	65
7.12.3 Field Documentation	65
7.12.3.1 fBaseAddr	65
7.12.3.2 fHandle	66
7.13 VME::GlobalOffset Struct Reference	66
7.13.1 Field Documentation	66
7.13.1.1 coarse	66
7.13.1.2 fine	66
7.14 HTTPMessage Class Reference	66
7.14.1 Detailed Description	67
7.14.2 Constructor & Destructor Documentation	67
7.14.2.1 HTTPMessage	68
7.14.2.2 HTTPMessage	68
7.14.3 Member Function Documentation	68
7.14.3.1 Decode	68
7.14.3.2 Dump	68
7.14.3.3 Encode	68
7.14.3.4 GetKey	68
7.14.4 Field Documentation	68
7.14.4.1 fOriginalString	68
7.14.4.2 fWS	68
7.15 VME::IOModuleV262 Class Reference	69
7.15.1 Constructor & Destructor Documentation	70
7.15.1.1 IOModuleV262	70
7.15.1.2 ~IOModuleV262	70
7.15.2 Member Function Documentation	70

7.15.2.1	GetIdentifier	70
7.15.2.2	GetManufacturerId	71
7.15.2.3	GetModuleType	71
7.15.2.4	GetModuleVersion	71
7.15.2.5	GetSerialNumber	72
7.16	Message Class Reference	72
7.16.1	Detailed Description	73
7.16.2	Constructor & Destructor Documentation	73
7.16.2.1	Message	73
7.16.2.2	Message	73
7.16.2.3	Message	73
7.16.2.4	~Message	73
7.16.3	Member Function Documentation	73
7.16.3.1	Dump	73
7.16.3.2	GetKey	73
7.16.3.3	GetString	73
7.16.3.4	IsFromWeb	74
7.16.4	Field Documentation	74
7.16.4.1	fString	74
7.17	Messenger Class Reference	74
7.17.1	Detailed Description	75
7.17.2	Constructor & Destructor Documentation	76
7.17.2.1	Messenger	76
7.17.2.2	Messenger	76
7.17.2.3	~Messenger	76
7.17.3	Member Function Documentation	76
7.17.3.1	AddClient	76
7.17.3.2	Broadcast	77
7.17.3.3	Connect	77
7.17.3.4	Disconnect	78
7.17.3.5	DisconnectClient	78
7.17.3.6	GetType	79
7.17.3.7	ProcessMessage	79
7.17.3.8	Receive	79
7.17.3.9	Send	80
7.17.3.10	StartAcquisition	80
7.17.3.11	StopAcquisition	80
7.17.3.12	SwitchClientType	81
7.17.4	Field Documentation	81
7.17.4.1	fNumAttempts	81

7.17.4.2	fPID	81
7.17.4.3	fStderrPipe	81
7.17.4.4	fStdoutPipe	81
7.17.4.5	fWS	81
7.18	VME::PCIInterfaceA2818 Class Reference	81
7.18.1	Constructor & Destructor Documentation	81
7.18.1.1	PCIInterfaceA2818	81
7.18.1.2	~PCIInterfaceA2818	81
7.18.2	Member Function Documentation	81
7.18.2.1	GetFWRevision	82
7.18.3	Field Documentation	82
7.18.3.1	fHandle	82
7.19	Socket Class Reference	82
7.19.1	Detailed Description	84
7.19.2	Member Typedef Documentation	84
7.19.2.1	SocketCollection	84
7.19.3	Constructor & Destructor Documentation	84
7.19.3.1	Socket	84
7.19.3.2	Socket	84
7.19.3.3	~Socket	84
7.19.4	Member Function Documentation	84
7.19.4.1	AcceptConnections	84
7.19.4.2	Bind	84
7.19.4.3	Configure	85
7.19.4.4	Create	85
7.19.4.5	DumpConnected	85
7.19.4.6	FetchMessage	85
7.19.4.7	GetPort	85
7.19.4.8	GetSocketId	85
7.19.4.9	GetSocketType	85
7.19.4.10	IsWebSocket	86
7.19.4.11	Listen	86
7.19.4.12	PrepareConnection	86
7.19.4.13	SelectConnections	86
7.19.4.14	SendMessage	86
7.19.4.15	SetPort	87
7.19.4.16	SetSocketId	87
7.19.4.17	Start	87
7.19.4.18	Stop	87
7.19.5	Field Documentation	87

7.19.5.1	fAddress	87
7.19.5.2	fBuffer	87
7.19.5.3	fMaster	87
7.19.5.4	fPort	88
7.19.5.5	fReadFds	88
7.19.5.6	fSocketId	88
7.19.5.7	fSocketsConnected	88
7.20	SocketMessage Class Reference	88
7.20.1	Detailed Description	90
7.20.2	Constructor & Destructor Documentation	90
7.20.2.1	SocketMessage	90
7.20.2.2	SocketMessage	90
7.20.2.3	SocketMessage	90
7.20.2.4	SocketMessage	91
7.20.2.5	SocketMessage	91
7.20.2.6	SocketMessage	91
7.20.2.7	SocketMessage	91
7.20.2.8	SocketMessage	92
7.20.2.9	SocketMessage	92
7.20.2.10	SocketMessage	92
7.20.2.11	SocketMessage	92
7.20.2.12	~SocketMessage	92
7.20.3	Member Function Documentation	92
7.20.3.1	Dump	93
7.20.3.2	GetIntValue	93
7.20.3.3	GetKey	93
7.20.3.4	GetString	93
7.20.3.5	GetValue	93
7.20.3.6	GetVectorValue	93
7.20.3.7	Object	93
7.20.3.8	SetKeyValue	94
7.20.3.9	SetKeyValue	94
7.20.3.10	SetKeyValue	94
7.20.3.11	SetKeyValue	94
7.20.3.12	String	95
7.20.4	Field Documentation	95
7.20.4.1	fMessage	95
7.21	VME::TDCErrorFlag Class Reference	95
7.21.1	Detailed Description	95
7.21.2	Constructor & Destructor Documentation	96

7.21.2.1	TDCErrFlag	96
7.21.2.2	~TDCErrFlag	96
7.21.3	Member Function Documentation	96
7.21.3.1	Dump	96
7.21.3.2	GetWord	96
7.21.3.3	HasGroupError	96
7.21.3.4	HasInternalChipError	96
7.21.3.5	HasL1BufferOverflow	96
7.21.3.6	HasReachedEventSizeLimit	96
7.21.3.7	HasReadoutFIFOOverflow	96
7.21.3.8	HasTriggerFIFOOverflow	96
7.21.4	Friends And Related Function Documentation	96
7.21.4.1	operator<<	96
7.21.5	Field Documentation	96
7.21.5.1	fWord	96
7.22	VME::TDCEvent Class Reference	97
7.22.1	Detailed Description	98
7.22.2	Member Enumeration Documentation	98
7.22.2.1	EventType	98
7.22.3	Constructor & Destructor Documentation	98
7.22.3.1	TDCEvent	98
7.22.3.2	TDCEvent	98
7.22.3.3	TDCEvent	98
7.22.3.4	~TDCEvent	98
7.22.4	Member Function Documentation	98
7.22.4.1	Dump	98
7.22.4.2	GetBunchId	99
7.22.4.3	GetChannelId	99
7.22.4.4	GetErrorFlags	99
7.22.4.5	GetETTT	99
7.22.4.6	GetEventCount	100
7.22.4.7	GetEventId	100
7.22.4.8	GetGeo	100
7.22.4.9	GetStatus	101
7.22.4.10	GetTDCId	101
7.22.4.11	GetTime	101
7.22.4.12	GetType	101
7.22.4.13	GetWidth	102
7.22.4.14	GetWord	102
7.22.4.15	GetWordCount	102

7.22.4.16	IsTrailing	102
7.22.4.17	SetWord	102
7.22.5	Field Documentation	102
7.22.5.1	fWord	102
7.23	VME::TDCMeasurement Class Reference	103
7.23.1	Detailed Description	103
7.23.2	Constructor & Destructor Documentation	103
7.23.2.1	TDCMeasurement	103
7.23.2.2	TDCMeasurement	104
7.23.2.3	~TDCMeasurement	104
7.23.3	Member Function Documentation	104
7.23.3.1	Dump	105
7.23.3.2	GetBunchId	105
7.23.3.3	GetChannelId	105
7.23.3.4	GetETTT	105
7.23.3.5	GetEventId	105
7.23.3.6	GetLeadingTime	105
7.23.3.7	GetTDCId	105
7.23.3.8	GetToT	106
7.23.3.9	GetTrailingTime	106
7.23.3.10	NumErrors	106
7.23.3.11	NumEvents	106
7.23.3.12	SetEventsCollection	106
7.23.4	Field Documentation	106
7.23.4.1	fEvents	106
7.23.4.2	fMap	106
7.24	VME::TDCV1x90 Class Reference	106
7.24.1	Detailed Description	109
7.24.2	Member Enumeration Documentation	109
7.24.2.1	DLLMode	109
7.24.3	Constructor & Destructor Documentation	109
7.24.3.1	TDCV1x90	110
7.24.3.2	~TDCV1x90	110
7.24.4	Member Function Documentation	110
7.24.4.1	abort	110
7.24.4.2	CheckConfiguration	110
7.24.4.3	DisableChannel	111
7.24.4.4	EnableChannel	111
7.24.4.5	FetchEvents	111
7.24.4.6	GetAcquisitionMode	112

7.24.4.7	GetBLTEventNumberRegister	112
7.24.4.8	GetChannelDeadTime	112
7.24.4.9	GetControl	113
7.24.4.10	GetDetectionMode	113
7.24.4.11	GetDLLClock	113
7.24.4.12	GetErrorMarks	113
7.24.4.13	GetETTT	113
7.24.4.14	GetEventCounter	113
7.24.4.15	GetEventStored	114
7.24.4.16	GetFIFOSize	114
7.24.4.17	GetFirmwareRevision	115
7.24.4.18	GetGlobalOffset	115
7.24.4.19	GetModel	115
7.24.4.20	GetOUI	116
7.24.4.21	GetPol	116
7.24.4.22	GetRCAdjust	116
7.24.4.23	GetResolution	117
7.24.4.24	GetSerialNumber	117
7.24.4.25	GetStatus	117
7.24.4.26	GetTDCEncapsulation	118
7.24.4.27	GetTestMode	118
7.24.4.28	GetTriggerConfiguration	118
7.24.4.29	GetWindowOffset	118
7.24.4.30	GetWindowWidth	118
7.24.4.31	HardwareReset	118
7.24.4.32	ReadAcquisitionMode	119
7.24.4.33	ReadDetectionMode	119
7.24.4.34	SetAcquisitionMode	119
7.24.4.35	SetBLTEventNumberRegister	120
7.24.4.36	SetChannelDeadTime	120
7.24.4.37	SetContinuousStorage	120
7.24.4.38	SetControl	121
7.24.4.39	SetDetectionMode	121
7.24.4.40	SetDLLClock	121
7.24.4.41	SetErrorMarks	122
7.24.4.42	SetETTT	122
7.24.4.43	SetFIFOSize	122
7.24.4.44	SetGlobalOffset	123
7.24.4.45	SetLSBTraileadEdge	123
7.24.4.46	SetPairModeResolution	123

7.24.4.47 SetPol	124
7.24.4.48 SetRCAdjust	124
7.24.4.49 SetStatus	124
7.24.4.50 SetTDCEncapsulation	125
7.24.4.51 SetTestMode	125
7.24.4.52 SetTriggerMatching	125
7.24.4.53 SetVerboseLevel	125
7.24.4.54 SetWindowOffset	126
7.24.4.55 SetWindowWidth	127
7.24.4.56 SoftwareClear	127
7.24.4.57 SoftwareReset	128
7.24.4.58 WaitMicro	128
7.24.5 Field Documentation	128
7.24.5.1 fAcquisitionMode	128
7.24.5.2 fBuffer	128
7.24.5.3 fDetectionMode	128
7.24.5.4 fErrorMarks	128
7.24.5.5 fVerb	128
7.24.5.6 fWindowWidth	128
7.24.5.7 gEnd	128
7.24.5.8 nchannels	128
7.24.5.9 pair_lead_res	128
7.24.5.10 pair_width_res	128
7.25 VME::TDCV1x90Control Class Reference	129
7.25.1 Detailed Description	129
7.25.2 Constructor & Destructor Documentation	130
7.25.2.1 TDCV1x90Control	130
7.25.2.2 ~TDCV1x90Control	130
7.25.3 Member Function Documentation	130
7.25.3.1 Dump	131
7.25.3.2 GetAlign64	131
7.25.3.3 GetBusError	131
7.25.3.4 GetCompensation	132
7.25.3.5 GetEmptyEvent	132
7.25.3.6 GetETTT	132
7.25.3.7 GetEventFIFO	132
7.25.3.8 GetMEBAccess	132
7.25.3.9 GetSRAMCompensation	132
7.25.3.10 GetSWTermination	132
7.25.3.11 GetTermination	132

7.25.3.12	GetTestFIFO	132
7.25.3.13	GetValue	132
7.25.3.14	SetAlign64	132
7.25.3.15	SetBusError	132
7.25.3.16	SetCompensation	133
7.25.3.17	SetEmptyEvent	133
7.25.3.18	SetETTT	133
7.25.3.19	SetEventFIFO	134
7.25.3.20	SetMEBAccess	134
7.25.3.21	SetSRAMCompensation	134
7.25.3.22	SetSWTermination	135
7.25.3.23	SetTermination	135
7.25.3.24	SetTestFIFO	135
7.25.4	Field Documentation	135
7.25.4.1	fWord	135
7.26	VME::TDCV1x90Status Class Reference	136
7.26.1	Detailed Description	136
7.26.2	Member Enumeration Documentation	136
7.26.2.1	TDCResolution	136
7.26.3	Constructor & Destructor Documentation	137
7.26.3.1	TDCV1x90Status	137
7.26.3.2	~TDCV1x90Status	137
7.26.4	Member Function Documentation	137
7.26.4.1	AlmostFull	137
7.26.4.2	BusError	137
7.26.4.3	DataReady	137
7.26.4.4	Dump	138
7.26.4.5	Error	138
7.26.4.6	Error	139
7.26.4.7	Full	139
7.26.4.8	GetValue	139
7.26.4.9	HeadersEnabled	139
7.26.4.10	PairMode	139
7.26.4.11	Purged	139
7.26.4.12	Resolution	139
7.26.4.13	TerminationOn	139
7.26.4.14	TriggerLost	139
7.26.4.15	TriggerMatching	139
7.26.5	Field Documentation	139
7.26.5.1	fWord	139

7.27	VME::trailead_t Struct Reference	139
7.27.1	Field Documentation	140
7.27.1.1	ettt	140
7.27.1.2	event_count	140
7.27.1.3	leading	140
7.27.1.4	total_hits	140
7.27.1.5	trailing	140
7.28	VMEReader Class Reference	140
7.28.1	Detailed Description	142
7.28.2	Constructor & Destructor Documentation	142
7.28.2.1	VMEReader	142
7.28.2.2	~VMEReader	143
7.28.3	Member Function Documentation	143
7.28.3.1	Abort	143
7.28.3.2	AddCFD	143
7.28.3.3	AddFPGAUnit	144
7.28.3.4	AddIOModule	144
7.28.3.5	AddTDC	144
7.28.3.6	GetCFD	145
7.28.3.7	GetCFDCollection	145
7.28.3.8	GetFPGAUnit	145
7.28.3.9	GetIOModule	145
7.28.3.10	GetNumCFD	145
7.28.3.11	GetNumTDC	145
7.28.3.12	GetOutputFile	145
7.28.3.13	GetRunNumber	145
7.28.3.14	GetTDC	145
7.28.3.15	GetTDCCollection	146
7.28.3.16	SendClear	146
7.28.3.17	SendPulse	146
7.28.3.18	SetOutputFile	146
7.28.3.19	StartPulser	146
7.28.3.20	StopPulser	147
7.28.4	Field Documentation	147
7.28.4.1	fBridge	147
7.28.4.2	fCFDCollection	147
7.28.4.3	fFPGA	147
7.28.4.4	fIsPulserStarted	147
7.28.4.5	fOnSocket	147
7.28.4.6	fOutputFile	147

7.28.4.7 fSG	148
7.28.4.8 fTDCCollection	148
Index	149

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Socket communication objects	9
--	---

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

VME	11
VME::TDCV1x90Opcodes	16

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

VME::BridgeVx718Control	26
VME::BridgeVx718Status	27
VME::CAENETControllerV288Status	31
file_header_t	44
FileReader	45
VME::FPGAUnitV1495Control	59
VME::GenericBoard< Register, am >	64
VME::GenericBoard< CAENETControllerV288Register, cvA24_U_DATA >	64
VME::CAENETControllerV288	29
VME::GenericBoard< CFDV812Register, cvA24_U_DATA >	64
VME::CFDV812	32
VME::GenericBoard< CVRegisters, cvA32_U_DATA >	64
VME::BridgeVx718	21
VME::GenericBoard< FPGAUnitV1495Register, cvA32_U_DATA >	64
VME::FPGAUnitV1495	48
VME::GenericBoard< IOModuleV262Register, cvA24_U_DATA >	64
VME::IOModuleV262	69
VME::GenericBoard< TDCV1x90Register, cvA32_U_DATA >	64
VME::TDCV1x90	106
VME::GlobalOffset	66
Message	72
HTTPMessage	66
SocketMessage	88
VME::PCIInterfaceA2818	81
Socket	82
Client	39
VMEReader	140
Messenger	74
VME::TDCErrorFlag	95
VME::TDCEvent	97
VME::TDCMeasurement	103
VME::TDCV1x90Control	129
VME::TDCV1x90Status	136
VME::trailead_t	139

Chapter 4

Data Structure Index

4.1 Data Structures

Here are the data structures with brief descriptions:

VME::BridgeVx718	
Class defining the VME bridge	21
VME::BridgeVx718Control	26
VME::BridgeVx718Status	27
VME::CAENETControllerV288	
Handler for a CAEN V288 CAENET controller	29
VME::CAENETControllerV288Status	31
VME::CFDV812	
Controller for a CAEN V812 constant fraction discriminator	32
Client	
Base client object for the socket	39
file_header_t	
Header to the output files	44
FileReader	
Handler for a TDC output file readout	45
VME::FPGAUnitV1495	48
VME::FPGAUnitV1495Control	59
VME::GenericBoard< Register, am >	64
VME::GlobalOffset	66
HTTPMessage	
Message to be transmitted through a WebSocket protocol	66
VME::IOModuleV262	69
Message	
Base socket message type	72
Messenger	
Base master object for the socket	74
VME::PCIInterfaceA2818	81
Socket	
Base socket object from which clients/master from a socket inherit	82
SocketMessage	
Socket-passed message type	88
VME::TDCErrorFlag	
Error flags handler	95
VME::TDCEvent	
HPTDC event parser	97
VME::TDCMeasurement	103
VME::TDCV1x90	106

VME::TDCV1x90Control	
TDC control register	129
VME::TDCV1x90Status	
TDC status register	136
VME::trailead_t	139
VMEReader	140

Chapter 5

Module Documentation

5.1 Socket communication objects

Data Structures

- class [Client](#)
Base client object for the socket.
- class [HTTPMessage](#)
Message to be transmitted through a WebSocket protocol.
- class [Messenger](#)
Base master object for the socket.
- class [Socket](#)
Base socket object from which clients/master from a socket inherit.
- class [SocketMessage](#)
Socket-passed message type.

Enumerations

- enum [Socket::SocketType](#) {
[Socket::INVALID](#) = -1, [Socket::MASTER](#) = 0, [Socket::WEBSOCKET_CLIENT](#), [Socket::CLIENT](#),
[Socket::DETECTOR](#), [Socket::DQM](#) }
Type of actor playing a role on the socket.

5.1.1 Detailed Description

5.1.2 Enumeration Type Documentation

5.1.2.1 enum [Socket::SocketType](#)

Type of actor playing a role on the socket.

Enumerator

[INVALID](#)
[MASTER](#)
[WEBSOCKET_CLIENT](#)
[CLIENT](#)
[DETECTOR](#)
[DQM](#)

Chapter 6

Namespace Documentation

6.1 VME Namespace Reference

Namespaces

- [TDCV1x90Opcodes](#)

Data Structures

- class [BridgeVx718](#)
class defining the VME bridge
- class [BridgeVx718Control](#)
- class [BridgeVx718Status](#)
- class [CAENETControllerV288](#)
Handler for a CAEN V288 CAENET controller.
- class [CAENETControllerV288Status](#)
- class [CFDV812](#)
Controller for a CAEN V812 constant fraction discriminator.
- class [FPGAUnitV1495](#)
- class [FPGAUnitV1495Control](#)
- class [GenericBoard](#)
- struct [GlobalOffset](#)
- class [IOModuleV262](#)
- class [PCIInterfaceA2818](#)
- class [TDCErrorFlag](#)
Error flags handler.
- class [TDCEvent](#)
HPTDC event parser.
- class [TDCMeasurement](#)
- class [TDCV1x90](#)
- class [TDCV1x90Control](#)
TDC control register.
- class [TDCV1x90Status](#)
TDC status register.
- struct [trailead_t](#)

Typedefs

- typedef std::map< uint32_t, VME::CFDV812 * > CFDCollection
Mapper from physical VME addresses to pointers to CFD objects.
- typedef std::vector< TDCEvent > TDCEventCollection
- typedef std::map< uint32_t, VME::TDCV1x90 * > TDCCollection
Mapper from physical VME addresses to pointers to TDC objects.

Enumerations

- enum BridgeType { CAEN_V1718, CAEN_V2718 }
Compatible bridge types.
- enum CAENETControllerV288Register {
kV288DataBuffer = 0x00, kV288Status = 0x02, kV288Transmission = 0x04, kV288ModuleReset = 0x06,
kV288IRQVector = 0x08 }
- enum CFDV812Register {
kV812ThresholdChannel0 = 0x00, kV812OutputWidthGroup0 = 0x40, kV812OutputWidthGroup1 = 0x42, kV812DeadTimeGroup0 = 0x44,
kV812DeadTimeGroup1 = 0x46, kV812MajorityThreshold = 0x48, kV812PatternOfInhibit = 0x4a, kV812TestPulse = 0x4c,
kV812FixedCode = 0xfa, kV812Info0 = 0xfc, kV812Info1 = 0xfe }
- enum FPGAUnitV1495Register {
kV1495ScalerCounter = 0x100c, kV1495DelaySettings = 0x1010, kV1495UserFWRevision = 0x1014, kV1495TDCBoardInterface = 0x1018,
kV1495ClockSettings = 0x101c, kV1495Control = 0x1020, kV1495TriggerSettings = 0x1024, kV1495OutputSettings = 0x1028,
kV1495GeoAddress = 0x8008, kV1495UserFPGAFlashMem = 0x8014, kV1495UserFPGAConfig = 0x8016, kV1495ModuleReset = 0x800a,
kV1495FWRevision = 0x800c, kV1495ConfigurationROM = 0x8100, kV1495OUI2 = 0x8124, kV1495OUI1 = 0x8128,
kV1495OUI0 = 0x812c, kV1495Board2 = 0x8134, kV1495Board1 = 0x8138, kV1495Board0 = 0x813c, kV1495HWRevision3 = 0x8140, kV1495HWRevision2 = 0x8144, kV1495HWRevision1 = 0x8148, kV1495HWRevision0 = 0x814c,
kV1495SerNum0 = 0x8180, kV1495SerNum1 = 0x8184 }
- enum IOModuleV262Register {
kECLLevelWrite = 0x04, kNIMLevelWrite = 0x06, kNIMPulseWrite = 0x08, kNIMPulseRead = 0x0a, kIdentifier = 0xfa, kBoardInfo0 = 0xfc, kBoardInfo1 = 0xfe }
- enum AcquisitionMode { CONT_STORAGE, TRIG_MATCH }
TDC acquisition mode.
- enum DetectionMode { PAIR = 0x0, OTRAILING = 0x1, OLEADING = 0x2, TRAILEAD = 0x3 }
- enum trig_conf {
MATCH_WIN_WIDTH = 0, WIN_OFFSET = 1, EXTRA_SEARCH_WIN_WIDTH = 2, REJECT_MARGIN = 3, TRIG_TIME_SUB = 4 }
- enum trailead_edge_lsb { r800ps = 0, r200ps = 1, r100ps = 2, r25ps = 3 }
- enum micro_handshake { WRITE_OK = 0, READ_OK = 1 }
- enum TDCV1x90Register {
kOutputBuffer = 0x0000, kControl = 0x1000, kStatus = 0x1002, kInterruptLevel = 0x100a, kInterruptVector = 0x100c, kGeoAddress = 0x100e, kMCSTBase = 0x1010, kMCSTControl = 0x1012, kModuleReset = 0x1014, kSoftwareClear = 0x1016, kEventCounter = 0x101c, kEventStored = 0x1020, kBLTEventNumber = 0x1024, kFirmwareRev = 0x1026, kMicro = 0x102e, kMicroHandshake = 0x1030, kEventFIFO = 0x1038, kEventFIFOStoredRegister = 0x103c, kEventFIFOStatusRegister = 0x103e, kROMOUI2 = 0x4024,
kROMOUI1 = 0x4028, kROMOUI0 = 0x402c, kROMBoard2 = 0x4034, kROMBoard1 = 0x4038, kROMBoard0 = 0x403c, kROMRevis3 = 0x4040, kROMRevis2 = 0x4044, kROMRevis1 = 0x4048, kROMRevis0 = 0x404c, kROMSerNum1 = 0x4080, kROMSerNum0 = 0x4084 }

6.1.1 Typedef Documentation

6.1.1.1 typedef std::map<uint32_t,VME::CFDV812*> VME::CFDCollection

Mapper from physical [VME](#) addresses to pointers to CFD objects.

6.1.1.2 typedef std::map<uint32_t,VME::TDCV1x90*> VME::TDCCollection

Mapper from physical [VME](#) addresses to pointers to TDC objects.

6.1.1.3 typedef std::vector<TDCEvent> VME::TDCEventCollection

6.1.2 Enumeration Type Documentation

6.1.2.1 enum VME::AcquisitionMode

TDC acquisition mode.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Enumerator

CONT_STORAGE

TRIG_MATCH

6.1.2.2 enum VME::BridgeType

Compatible bridge types.

Enumerator

CAEN_V1718

CAEN_V2718

6.1.2.3 enum VME::CAENETControllerV288Register

Enumerator

kV288DataBuffer

kV288Status

kV288Transmission

kV288ModuleReset

kV288IRQVector

6.1.2.4 enum VME::CFDV812Register

Enumerator

kV812ThresholdChannel0

kV812OutputWidthGroup0

kV812OutputWidthGroup1
kV812DeadTimeGroup0
kV812DeadTimeGroup1
kV812MajorityThreshold
kV812PatternOfInhibit
kV812TestPulse
kV812FixedCode
kV812Info0
kV812Info1

6.1.2.5 enum VME::DetectionMode

Enumerator

PAIR
OTRILING
OLEADING
TRAILEAD

6.1.2.6 enum VME::FPGAUnitV1495Register

Enumerator

kV1495ScalerCounter
kV1495DelaySettings
kV1495UserFWRevision
kV1495TDCBoardInterface
kV1495ClockSettings
kV1495Control
kV1495TriggerSettings
kV1495OutputSettings
kV1495GeoAddress
kV1495UserFPGAFlashMem
kV1495UserFPGAConfig
kV1495ModuleReset
kV1495FWRevision
kV1495ConfigurationROM
kV1495OUI2
kV1495OUI1
kV1495OUI0
kV1495Board2
kV1495Board1
kV1495Board0
kV1495HWRevision3
kV1495HWRevision2
kV1495HWRevision1
kV1495HWRevision0
kV1495SerNum0
kV1495SerNum1

6.1.2.7 enum VME::IOModuleV262Register

Enumerator

kECLLevelWrite
kNIMLevelWrite
kNIMPulseWrite
kNIMPulseRead
kIdentifier
kBoardInfo0
kBoardInfo1

6.1.2.8 enum VME::micro_handshake

Enumerator

WRITE_OK Is the TDC ready for writing?
READ_OK Is the TDC ready for reading?

6.1.2.9 enum VME::TDCV1x90Register

Enumerator

kOutputBuffer
kControl
kStatus
kInterruptLevel
kInterruptVector
kGeoAddress
kMCSTBase
kMCSTControl
kModuleReset
kSoftwareClear
kEventCounter
kEventStored
kBLTEventNumber
kFirmwareRev
kMicro
kMicroHandshake
kEventFIFO
kEventFIFOStoredRegister
kEventFIFOStatusRegister
kROMOui2
kROMOui1
kROMOui0
kROMBoard2
kROMBoard1
kROMBoard0

kROMRevis3
kROMRevis2
kROMRevis1
kROMRevis0
kROMSerNum1
kROMSerNum0

6.1.2.10 enum VME::trailead_edge_lsb

Enumerator

r800ps
r200ps
r100ps
r25ps

6.1.2.11 enum VME::trig_conf

Enumerator

MATCH_WIN_WIDTH
WIN_OFFSET
EXTRA_SEARCH_WIN_WIDTH
REJECT_MARGIN
TRIG_TIME_SUB

6.2 VME::TDCV1x90Opcodes Namespace Reference

Functions

- Opcode [TRG_MATCH](#) (0x0000)
- Opcode [CONT_STOR](#) (0x0100)
- Opcode [READ_ACQ_MOD](#) (0x0200)
- Opcode [SET_KEEP_TOKEN](#) (0x0300)
- Opcode [CLEAR_KEEP_TOKEN](#) (0x0400)
- Opcode [LOAD_DEF_CONFIG](#) (0x0500)
- Opcode [SAVE_USER_CONFIG](#) (0x0600)
- Opcode [LOAD_USER_CONFIG](#) (0x0700)
- Opcode [AUTOLOAD_USER_CONF](#) (0x0800)
- Opcode [AUTOLOAD_DEF_CONFI](#) (0x0900)
- Opcode [SET_WIN_WIDTH](#) (0x1000)
- Opcode [SET_WIN_OFFS](#) (0x1100)
- Opcode [SET_SW_MARGIN](#) (0x1200)
- Opcode [SET_REJ_MARGIN](#) (0x1300)
- Opcode [EN_SUB_TRG](#) (0x1400)
- Opcode [DIS_SUB_TRG](#) (0x1500)
- Opcode [READ_TRG_CONF](#) (0x1600)
- Opcode [SET_DETECTION](#) (0x2200)
- Opcode [READ_DETECTION](#) (0x2300)
- Opcode [SET_TR_LEAD_LSB](#) (0x2400)

- Opcode [SET_PAIR_RES](#) (0x2500)
- Opcode [READ_RES](#) (0x2600)
- Opcode [SET_DEAD_TIME](#) (0x2800)
- Opcode [READ_DEAD_TIME](#) (0x2900)
- Opcode [EN_HEAD_TRAILER](#) (0x3000)
- Opcode [DIS_HEAD_TRAILER](#) (0x3100)
- Opcode [READ_HEAD_TRAILER](#) (0x3200)
- Opcode [SET_EVENT_SIZE](#) (0x3300)
- Opcode [READ_EVENT_SIZE](#) (0x3400)
- Opcode [EN_ERROR_MARK](#) (0x3500)
- Opcode [DIS_ERROR_MARK](#) (0x3600)
- Opcode [EN_ERROR_BYPASS](#) (0x3700)
- Opcode [DIS_ERROR_BYPASS](#) (0x3800)
- Opcode [SET_ERROR_TYPES](#) (0x3900)
- Opcode [READ_ERROR_TYPES](#) (0x3a00)
- Opcode [SET_FIFO_SIZE](#) (0x3b00)
- Opcode [READ_FIFO_SIZE](#) (0x3c00)
- Opcode [EN_CHANNEL](#) (0x4000)
- Opcode [DIS_CHANNEL](#) (0x4100)
- Opcode [EN_ALL_CHANNEL](#) (0x4200)
- Opcode [DIS_ALL_CHANNEL](#) (0x4300)
- Opcode [WRITE_EN_PATTERN](#) (0x4400)
- Opcode [READ_EN_PATTERN](#) (0x4500)
- Opcode [WRITE_EN_PATTERN32](#) (0x4600)
- Opcode [READ_EN_PATTERN32](#) (0x4700)
- Opcode [SET_GLOB_OFFS](#) (0x5000)
- Opcode [READ_GLOB_OFFS](#) (0x5100)
- Opcode [SET_ADJUST_CH](#) (0x5200)
- Opcode [READ_ADJUST_CH](#) (0x5200)
- Opcode [SET_RC_ADJ](#) (0x5400)
- Opcode [READ_RC_ADJ](#) (0x5500)
- Opcode [SAVE_RC_ADJ](#) (0x5600)
- Opcode [READ_TDC_ID](#) (0x6000)
- Opcode [READ_MICRO_REV](#) (0x6100)
- Opcode [RESET_DLL_PLL](#) (0x6200)
- Opcode [WRITE_SETUP_REG](#) (0x7000)
- Opcode [READ_SETUP_REG](#) (0x7100)
- Opcode [UPDATE_SETUP_REG](#) (0x7200)
- Opcode [DEFAULT_SETUP_REG](#) (0x7300)
- Opcode [READ_ERROR_STATUS](#) (0x7400)
- Opcode [READ_DLL_LOCK](#) (0x7500)
- Opcode [READ_STATUS_STREAM](#) (0x7600)
- Opcode [UPDATE_SETUP_TDC](#) (0x7700)
- Opcode [WRITE_EEPROM](#) (0xc000)
- Opcode [READ_EEPROM](#) (0xc100)
- Opcode [REV_DATE_MICRO_FW](#) (0xc200)
- Opcode [WRITE_SPARE](#) (0xc300)
- Opcode [READ_SPARE](#) (0xc400)
- Opcode [ENABLE_TEST_MODE](#) (0xc500)
- Opcode [DISABLE_TEST_MODE](#) (0xc600)
- Opcode [SET_TDC_TSET_OUTPUT](#) (0xc700)
- Opcode [SET_DLL_CLOCK](#) (0xc800)
- Opcode [READ_SETUP_SCANPATH](#) (0xc900)

6.2.1 Function Documentation

6.2.1.1 Opcode VME::TDCV1x90Opcodes::AUTOLOAD_DEF_CONFI (0x0900)

6.2.1.2 Opcode VME::TDCV1x90Opcodes::AUTOLOAD_USER_CONF (0x0800)

6.2.1.3 Opcode VME::TDCV1x90Opcodes::CLEAR_KEEP_TOKEN (0x0400)

6.2.1.4 Opcode VME::TDCV1x90Opcodes::CONT_STOR (0x0100)

6.2.1.5 Opcode VME::TDCV1x90Opcodes::DEFAULT_SETUP_REG (0x7300)

6.2.1.6 Opcode VME::TDCV1x90Opcodes::DIS_ALL_CHANNEL (0x4300)

6.2.1.7 Opcode VME::TDCV1x90Opcodes::DIS_CHANNEL (0x4100)

6.2.1.8 Opcode VME::TDCV1x90Opcodes::DIS_ERROR_BYPASS (0x3800)

6.2.1.9 Opcode VME::TDCV1x90Opcodes::DIS_ERROR_MARK (0x3600)

6.2.1.10 Opcode VME::TDCV1x90Opcodes::DIS_HEAD_TRAILER (0x3100)

6.2.1.11 Opcode VME::TDCV1x90Opcodes::DIS_SUB_TRG (0x1500)

6.2.1.12 Opcode VME::TDCV1x90Opcodes::DISABLE_TEST_MODE (0xc600)

6.2.1.13 Opcode VME::TDCV1x90Opcodes::EN_ALL_CHANNEL (0x4200)

6.2.1.14 Opcode VME::TDCV1x90Opcodes::EN_CHANNEL (0x4000)

6.2.1.15 Opcode VME::TDCV1x90Opcodes::EN_ERROR_BYPASS (0x3700)

6.2.1.16 Opcode VME::TDCV1x90Opcodes::EN_ERROR_MARK (0x3500)

6.2.1.17 Opcode VME::TDCV1x90Opcodes::EN_HEAD_TRAILER (0x3000)

6.2.1.18 Opcode VME::TDCV1x90Opcodes::EN_SUB_TRG (0x1400)

6.2.1.19 Opcode VME::TDCV1x90Opcodes::ENABLE_TEST_MODE (0xc500)

6.2.1.20 Opcode VME::TDCV1x90Opcodes::LOAD_DEF_CONFIG (0x0500)

6.2.1.21 Opcode VME::TDCV1x90Opcodes::LOAD_USER_CONFIG (0x0700)

6.2.1.22 Opcode VME::TDCV1x90Opcodes::READ_ACQ_MOD (0x0200)

6.2.1.23 Opcode VME::TDCV1x90Opcodes::READ_ADJUST_CH (0x5200)

6.2.1.24 Opcode VME::TDCV1x90Opcodes::READ_DEAD_TIME (0x2900)

6.2.1.25 Opcode VME::TDCV1x90Opcodes::READ_DETECTION (0x2300)

6.2.1.26 Opcode VME::TDCV1x90Opcodes::READ_DLL_LOCK (0x7500)

6.2.1.27 Opcode VME::TDCV1x90Opcodes::READ_EEPROM (0xc100)

- 6.2.1.28 Opcode VME::TDCV1x90Opcodes::READ_EN_PATTERN (0x4500)
- 6.2.1.29 Opcode VME::TDCV1x90Opcodes::READ_EN_PATTERN32 (0x4700)
- 6.2.1.30 Opcode VME::TDCV1x90Opcodes::READ_ERROR_STATUS (0x7400)
- 6.2.1.31 Opcode VME::TDCV1x90Opcodes::READ_ERROR_TYPES (0x3a00)
- 6.2.1.32 Opcode VME::TDCV1x90Opcodes::READ_EVENT_SIZE (0x3400)
- 6.2.1.33 Opcode VME::TDCV1x90Opcodes::READ_FIFO_SIZE (0x3c00)
- 6.2.1.34 Opcode VME::TDCV1x90Opcodes::READ_GLOB_OFFS (0x5100)
- 6.2.1.35 Opcode VME::TDCV1x90Opcodes::READ_HEAD_TRAILER (0x3200)
- 6.2.1.36 Opcode VME::TDCV1x90Opcodes::READ_MICRO_REV (0x6100)
- 6.2.1.37 Opcode VME::TDCV1x90Opcodes::READ_RC_ADJ (0x5500)
- 6.2.1.38 Opcode VME::TDCV1x90Opcodes::READ_RES (0x2600)
- 6.2.1.39 Opcode VME::TDCV1x90Opcodes::READ_SETUP_REG (0x7100)
- 6.2.1.40 Opcode VME::TDCV1x90Opcodes::READ_SETUP_SCANPATH (0xc900)
- 6.2.1.41 Opcode VME::TDCV1x90Opcodes::READ_SPARE (0xc400)
- 6.2.1.42 Opcode VME::TDCV1x90Opcodes::READ_STATUS_STREAM (0x7600)
- 6.2.1.43 Opcode VME::TDCV1x90Opcodes::READ_TDC_ID (0x6000)
- 6.2.1.44 Opcode VME::TDCV1x90Opcodes::READ_TRG_CONF (0x1600)
- 6.2.1.45 Opcode VME::TDCV1x90Opcodes::RESET_DLL_PLL (0x6200)
- 6.2.1.46 Opcode VME::TDCV1x90Opcodes::REV_DATE_MICRO_FW (0xc200)
- 6.2.1.47 Opcode VME::TDCV1x90Opcodes::SAVE_RC_ADJ (0x5600)
- 6.2.1.48 Opcode VME::TDCV1x90Opcodes::SAVE_USER_CONFIG (0x0600)
- 6.2.1.49 Opcode VME::TDCV1x90Opcodes::SET_ADJUST_CH (0x5200)
- 6.2.1.50 Opcode VME::TDCV1x90Opcodes::SET_DEAD_TIME (0x2800)
- 6.2.1.51 Opcode VME::TDCV1x90Opcodes::SET_DETECTION (0x2200)
- 6.2.1.52 Opcode VME::TDCV1x90Opcodes::SET_DLL_CLOCK (0xc800)
- 6.2.1.53 Opcode VME::TDCV1x90Opcodes::SET_ERROR_TYPES (0x3900)
- 6.2.1.54 Opcode VME::TDCV1x90Opcodes::SET_EVENT_SIZE (0x3300)
- 6.2.1.55 Opcode VME::TDCV1x90Opcodes::SET_FIFO_SIZE (0x3b00)

- 6.2.1.56 Opcode VME::TDCV1x90OpCodes::SET_GLOB_OFFS (0x5000)
- 6.2.1.57 Opcode VME::TDCV1x90OpCodes::SET_KEEP_TOKEN (0x0300)
- 6.2.1.58 Opcode VME::TDCV1x90OpCodes::SET_PAIR_RES (0x2500)
- 6.2.1.59 Opcode VME::TDCV1x90OpCodes::SET_RC_ADJ (0x5400)
- 6.2.1.60 Opcode VME::TDCV1x90OpCodes::SET_REJ_MARGIN (0x1300)
- 6.2.1.61 Opcode VME::TDCV1x90OpCodes::SET_SW_MARGIN (0x1200)
- 6.2.1.62 Opcode VME::TDCV1x90OpCodes::SET_TDC_TSET_OUTPUT (0xc700)
- 6.2.1.63 Opcode VME::TDCV1x90OpCodes::SET_TR_LEAD_LSB (0x2400)
- 6.2.1.64 Opcode VME::TDCV1x90OpCodes::SET_WIN_OFFS (0x1100)
- 6.2.1.65 Opcode VME::TDCV1x90OpCodes::SET_WIN_WIDTH (0x1000)
- 6.2.1.66 Opcode VME::TDCV1x90OpCodes::TRG_MATCH (0x0000)
- 6.2.1.67 Opcode VME::TDCV1x90OpCodes::UPDATE_SETUP_REG (0x7200)
- 6.2.1.68 Opcode VME::TDCV1x90OpCodes::UPDATE_SETUP_TDC (0x7700)
- 6.2.1.69 Opcode VME::TDCV1x90OpCodes::WRITE_EEPROM (0xc000)
- 6.2.1.70 Opcode VME::TDCV1x90OpCodes::WRITE_EN_PATTERN (0x4400)
- 6.2.1.71 Opcode VME::TDCV1x90OpCodes::WRITE_EN_PATTERN32 (0x4600)
- 6.2.1.72 Opcode VME::TDCV1x90OpCodes::WRITE_SETUP_REG (0x7000)
- 6.2.1.73 Opcode VME::TDCV1x90OpCodes::WRITE_SPARE (0xc300)

Chapter 7

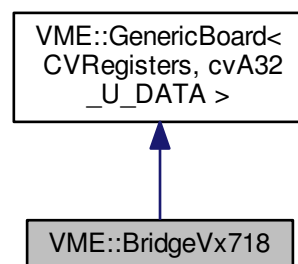
Data Structure Documentation

7.1 VME::BridgeVx718 Class Reference

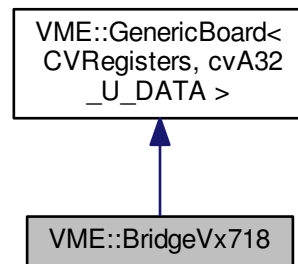
class defining the [VME](#) bridge

```
#include <VME_BridgeVx718.h>
```

Inheritance diagram for VME::BridgeVx718:



Collaboration diagram for VME::BridgeVx718:



Public Types

- enum [IRQId](#) {
[IRQ1](#) =0x1, [IRQ2](#) =0x2, [IRQ3](#) =0x4, [IRQ4](#) =0x8,
[IRQ5](#) =0x10, [IRQ6](#) =0x20, [IRQ7](#) =0x40 }

Public Member Functions

- [BridgeVx718](#) (const char *device, [BridgeType](#) type)
Constructor.
- [~BridgeVx718](#) ()
Destructor.
- int32_t [GetHandle](#) () const
Bridge's handle value.
- void [CheckPCIInterface](#) (const char *device) const
- void [CheckConfiguration](#) () const
- void [TestOutputs](#) () const
- void [Reset](#) () const
Perform a system reset of the module.
- [BridgeVx718Status](#) [GetStatus](#) () const
- void [SetIRQ](#) (unsigned int irq, bool enable=true)
- void [WaitIRQ](#) (unsigned int irq, unsigned long timeout=1000) const
- unsigned int [GetIRQStatus](#) () const
- void [OutputConf](#) (CVOutputSelect output) const
Set and control the output lines.
- void [OutputOn](#) (unsigned short output) const
- void [OutputOff](#) (unsigned short output) const
- void [InputConf](#) (CVInputSelect input) const
Set and read the input lines.
- void [InputRead](#) (CVInputSelect input) const
- void [StartPulser](#) (double period, double width, unsigned int num_pulses=0) const
- void [StopPulser](#) () const
- void [SinglePulse](#) (unsigned short channel) const

Private Attributes

- bool [fHasIRQ](#)

Additional Inherited Members

7.1.1 Detailed Description

class defining the [VME](#) bridge

This class initializes the CAEN V1718 [VME](#) bridge in order to control the crate.

Author

Laurent Forthomme laurent.forthomme@cern.ch
Bob Velghe bob.velghe@cern.ch

Date

Jun 2010

7.1.2 Member Enumeration Documentation

7.1.2.1 enum VME::BridgeVx718::IRQId

Enumerator

IRQ1

IRQ2

IRQ3

IRQ4

IRQ5

IRQ6

IRQ7

7.1.3 Constructor & Destructor Documentation

7.1.3.1 VME::BridgeVx718::BridgeVx718 (const char * *device*, BridgeType *type*)

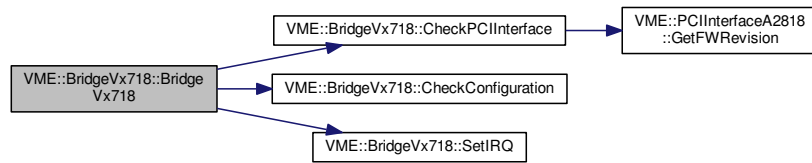
Constructor.

Bridge class constructor

Parameters

in	<i>device</i>	Device identifier on the VME crate
in	<i>type</i>	Device type (1718/2718)

Here is the call graph for this function:



7.1.3.2 VME::BridgeVx718::~~BridgeVx718 ()

Destructor.

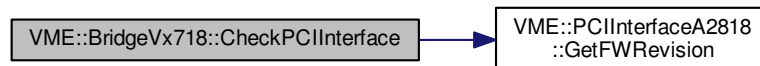
Bridge class destructor

7.1.4 Member Function Documentation

7.1.4.1 void VME::BridgeVx718::CheckConfiguration () const

7.1.4.2 void VME::BridgeVx718::CheckPCIInterface (const char * *device*) const

Here is the call graph for this function:



7.1.4.3 int32_t VME::BridgeVx718::GetHandle () const [inline]

Bridge's handle value.

Returns

Handle value

7.1.4.4 unsigned int VME::BridgeVx718::GetIRQStatus () const

7.1.4.5 BridgeVx718Status VME::BridgeVx718::GetStatus () const

7.1.4.6 void VME::BridgeVx718::InputConf (CVInputSelect *input*) const

Set and read the input lines.

7.1.4.7 void VME::BridgeVx718::InputRead (CVInputSelect *input*) const

7.1.4.8 void VME::BridgeVx718::OutputConf (CVOutputSelect *output*) const

Set and control the output lines.

7.1.4.9 void VME::BridgeVx718::OutputOff (unsigned short *output*) const

7.1.4.10 void VME::BridgeVx718::OutputOn (unsigned short *output*) const

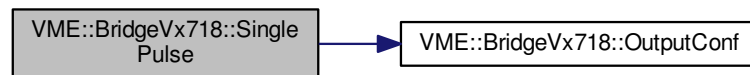
7.1.4.11 void VME::BridgeVx718::Reset () const

Perform a system reset of the module.

7.1.4.12 void VME::BridgeVx718::SetIRQ (unsigned int *irq*, bool *enable* = true)

7.1.4.13 void VME::BridgeVx718::SinglePulse (unsigned short *channel*) const

Here is the call graph for this function:

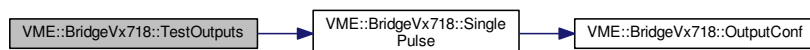


7.1.4.14 void VME::BridgeVx718::StartPulser (double *period*, double *width*, unsigned int *num_pulses* = 0) const

7.1.4.15 void VME::BridgeVx718::StopPulser () const

7.1.4.16 void VME::BridgeVx718::TestOutputs () const

Here is the call graph for this function:



7.1.4.17 void VME::BridgeVx718::WaitIRQ (unsigned int *irq*, unsigned long *timeout* = 1000) const

7.1.5 Field Documentation

7.1.5.1 bool VME::BridgeVx718::fHasIRQ [private]

The documentation for this class was generated from the following files:

- include/VME_BridgeVx718.h
- src/VME_BridgeVx718.cpp

7.2 VME::BridgeVx718Control Class Reference

```
#include <VME_BridgeVx718.h>
```

Public Member Functions

- [BridgeVx718Control](#) (uint16_t word)
- virtual [~BridgeVx718Control](#) ()
- bool [GetArbiterType](#) () const
Arbiter type.
- bool [GetRequesterType](#) () const
Requester type.
- bool [GetReleaseType](#) () const
Release type.
- unsigned int [GetBusReqLevel](#) () const
- bool [GetInterruptReq](#) () const
- bool [GetSysRes](#) () const
- bool [GetBusTimeout](#) () const
VME bus timeout.
- bool [GetAddressIncrement](#) () const
Address Increment.

Private Attributes

- uint16_t [fWord](#)

7.2.1 Constructor & Destructor Documentation

7.2.1.1 VME::BridgeVx718Control::BridgeVx718Control (uint16_t word) [inline]

7.2.1.2 virtual VME::BridgeVx718Control::~~BridgeVx718Control () [inline],[virtual]

7.2.2 Member Function Documentation

7.2.2.1 bool VME::BridgeVx718Control::GetAddressIncrement () const [inline]

Address Increment.

Returns

true if enabled, else false (FIFO mode)

7.2.2.2 bool VME::BridgeVx718Control::GetArbiterType () const [inline]

Arbiter type.

Returns

true if "Round Robin", else fixed priority

7.2.2.3 `unsigned int VME::BridgeVx718Control::GetBusReqLevel () const [inline]`

7.2.2.4 `bool VME::BridgeVx718Control::GetBusTimeout () const [inline]`

VME bus timeout.

Returns

true if 1400 us, else 50 us

7.2.2.5 `bool VME::BridgeVx718Control::GetInterruptReq () const [inline]`

7.2.2.6 `bool VME::BridgeVx718Control::GetReleaseType () const [inline]`

Release type.

Returns

true if release on request, else release when done

7.2.2.7 `bool VME::BridgeVx718Control::GetRequesterType () const [inline]`

Requester type.

Returns

true if demand, else fair

7.2.2.8 `bool VME::BridgeVx718Control::GetSysRes () const [inline]`

7.2.3 Field Documentation

7.2.3.1 `uint16_t VME::BridgeVx718Control::fWord [private]`

The documentation for this class was generated from the following file:

- include/VME_BridgeVx718.h

7.3 VME::BridgeVx718Status Class Reference

```
#include <VME_BridgeVx718.h>
```

Public Member Functions

- [BridgeVx718Status](#) (uint16_t word)
- virtual [~BridgeVx718Status](#) ()
- void [Dump](#) () const
- bool [GetSystemReset](#) () const
- bool [GetSystemControl](#) () const
- bool [GetDTACK](#) () const
- bool [GetBERR](#) () const
- bool [GetDipSwitch](#) (unsigned int sw) const
- bool [GetUSBType](#) () const

Private Attributes

- uint16_t [fWord](#)

7.3.1 Constructor & Destructor Documentation

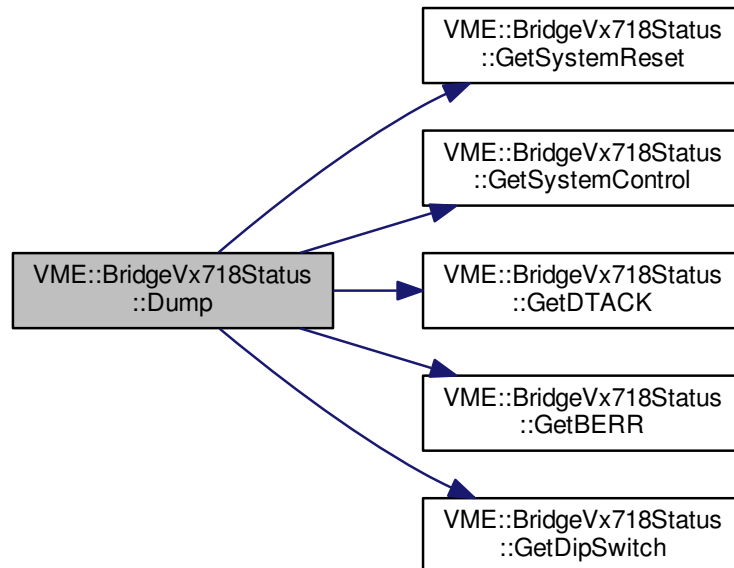
7.3.1.1 `VME::BridgeVx718Status::BridgeVx718Status (uint16_t word) [inline]`

7.3.1.2 `virtual VME::BridgeVx718Status::~~BridgeVx718Status () [inline], [virtual]`

7.3.2 Member Function Documentation

7.3.2.1 `void VME::BridgeVx718Status::Dump () const [inline]`

Here is the call graph for this function:



7.3.2.2 `bool VME::BridgeVx718Status::GetBERR () const [inline]`

7.3.2.3 `bool VME::BridgeVx718Status::GetDipSwitch (unsigned int sw) const [inline]`

7.3.2.4 `bool VME::BridgeVx718Status::GetDTACK () const [inline]`

7.3.2.5 `bool VME::BridgeVx718Status::GetSystemControl () const [inline]`

7.3.2.6 `bool VME::BridgeVx718Status::GetSystemReset () const [inline]`

7.3.2.7 `bool VME::BridgeVx718Status::GetUSBType () const [inline]`

7.3.3 Field Documentation

7.3.3.1 uint16_t VME::BridgeVx718Status::fWord [private]

The documentation for this class was generated from the following file:

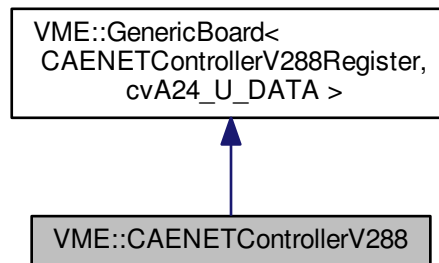
- include/VME_BridgeVx718.h

7.4 VME::CAENETControllerV288 Class Reference

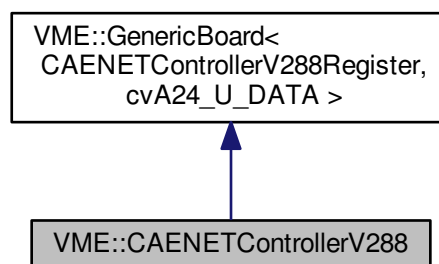
Handler for a CAEN V288 CAENET controller.

```
#include <VME_CAENETControllerV288.h>
```

Inheritance diagram for VME::CAENETControllerV288:



Collaboration diagram for VME::CAENETControllerV288:



Public Member Functions

- [CAENETControllerV288](#) (int32_t handle, uint32_t baseaddr)
- [~CAENETControllerV288](#) ()

- void [Send](#) ()
Send the whole buffer through the network.
- std::vector< uint16_t > [Receive](#) () const
Retrieve the network buffer.
- bool [WaitForResponse](#) (uint16_t *response, unsigned int max_trials=-1) const
- [CAENETControllerV288Status](#) [GetStatus](#) () const

Private Attributes

- unsigned int [fNumWordsInBuffer](#)

Friends

- void [operator<<](#) (uint16_t &word, [CAENETControllerV288](#) &cnt)
Fill the buffer with an additional 16-bit word.
- uint16_t & [operator>>](#) (uint16_t &word, const [CAENETControllerV288](#) &cnt)
Read back a 16-bit word from the buffer.

Additional Inherited Members

7.4.1 Detailed Description

Handler for a CAEN V288 CAENET controller.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

23 Jul 2015

7.4.2 Constructor & Destructor Documentation

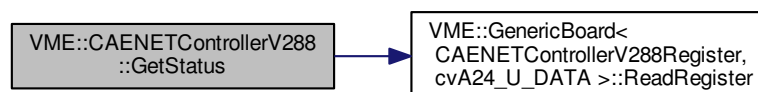
7.4.2.1 `VME::CAENETControllerV288::CAENETControllerV288 (int32_t handle, uint32_t baseaddr)`

7.4.2.2 `VME::CAENETControllerV288::~~CAENETControllerV288 () [inline]`

7.4.3 Member Function Documentation

7.4.3.1 `CAENETControllerV288Status VME::CAENETControllerV288::GetStatus () const`

Here is the call graph for this function:



7.4.3.2 `std::vector< uint16_t > VME::CAENETControllerV288::Receive () const`

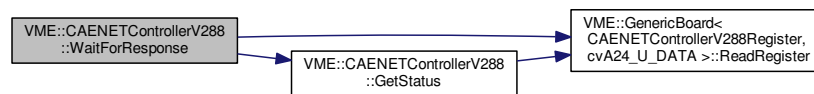
Retrieve the network buffer.

7.4.3.3 `void VME::CAENETControllerV288::Send ()`

Send the whole buffer through the network.

7.4.3.4 `bool VME::CAENETControllerV288::WaitForResponse (uint16_t * response, unsigned int max_trials = -1) const`

Here is the call graph for this function:



7.4.4 Friends And Related Function Documentation

7.4.4.1 `void operator<< (uint16_t & word, CAENETControllerV288 & cnt) [friend]`

Fill the buffer with an additional 16-bit word.

7.4.4.2 `uint16_t& operator>> (uint16_t & word, const CAENETControllerV288 & cnt) [friend]`

Read back a 16-bit word from the buffer.

7.4.5 Field Documentation

7.4.5.1 `unsigned int VME::CAENETControllerV288::fNumWordsInBuffer [private]`

The documentation for this class was generated from the following files:

- include/VME_CAENETControllerV288.h
- src/VME_CAENETControllerV288.cpp

7.5 VME::CAENETControllerV288Status Class Reference

```
#include <VME_CAENETControllerV288.h>
```

Public Types

- enum [OperationStatus](#) { [Valid](#) =0x0, [Invalid](#) =0x1 }

Public Member Functions

- [CAENETControllerV288Status](#) (uint16_t word)
- [~CAENETControllerV288Status](#) ()
- [OperationStatus](#) [GetOperationStatus](#) () const

Private Attributes

- uint16_t [fWord](#)

7.5.1 Member Enumeration Documentation

7.5.1.1 enum VME::CAENETControllerV288Status::OperationStatus

Enumerator

Valid

Invalid

7.5.2 Constructor & Destructor Documentation

7.5.2.1 VME::CAENETControllerV288Status::CAENETControllerV288Status (uint16_t *word*) [inline]

7.5.2.2 VME::CAENETControllerV288Status::~~CAENETControllerV288Status () [inline]

7.5.3 Member Function Documentation

7.5.3.1 OperationStatus VME::CAENETControllerV288Status::GetOperationStatus () const [inline]

7.5.4 Field Documentation

7.5.4.1 uint16_t VME::CAENETControllerV288Status::fWord [private]

The documentation for this class was generated from the following file:

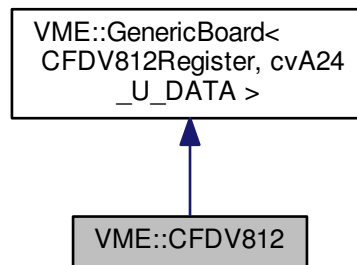
- include/VME_CAENETControllerV288.h

7.6 VME::CFDV812 Class Reference

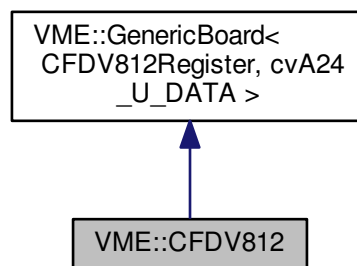
Controller for a CAEN V812 constant fraction discriminator.

```
#include <VME_CFDV812.h>
```

Inheritance diagram for VME::CFDV812:



Collaboration diagram for VME::CFDV812:



Public Member Functions

- `CFDV812` (int32_t bhandle, uint32_t baseaddr)
- `~CFDV812` ()
- void `CheckConfiguration` () const
- unsigned short `GetFixedCode` () const
- unsigned short `GetManufacturerId` () const
- unsigned short `GetModuleType` () const
- unsigned short `GetModuleVersion` () const
- unsigned short `GetSerialNumber` () const
- void `SetPOI` (unsigned short poi) const
Set the pattern of inhibit (list of enabled channels)
- void `SetThreshold` (unsigned short channel_id, unsigned short value) const
Set the threshold for one single channel, in units of 1 mV.
- void `SetOutputWidth` (unsigned short group_id, unsigned short value) const
Set the discriminated pulse output width for one group of 8 channels.
- void `SetDeadTime` (unsigned short group_id, unsigned short value) const
Set the discrimination dead time for one group of 8 channels.

Private Member Functions

- float [OutputWidthCalculator](#) (unsigned short value) const
- float [DeadTimeCalculator](#) (unsigned short value) const

Additional Inherited Members

7.6.1 Detailed Description

Controller for a CAEN V812 constant fraction discriminator.

Author

Laurent Forthomme laurent.forthomme@cern.ch

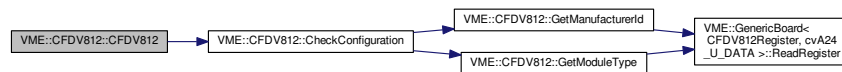
Date

22 Jul 2015

7.6.2 Constructor & Destructor Documentation

7.6.2.1 VME::CFDV812::CFDV812 (int32_t *bhandle*, uint32_t *baseaddr*)

Here is the call graph for this function:

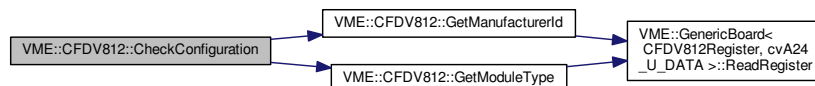


7.6.2.2 VME::CFDV812::~~CFDV812 () [inline]

7.6.3 Member Function Documentation

7.6.3.1 void VME::CFDV812::CheckConfiguration () const

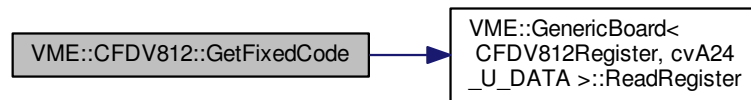
Here is the call graph for this function:



7.6.3.2 float VME::CFDV812::DeadTimeCalculator (unsigned short *value*) const [private]

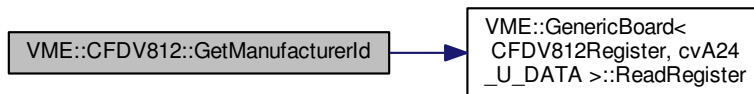
7.6.3.3 unsigned short VME::CFDV812::GetFixedCode () const

Here is the call graph for this function:



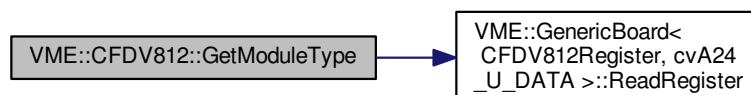
7.6.3.4 unsigned short VME::CFDV812::GetManufacturerId () const

Here is the call graph for this function:



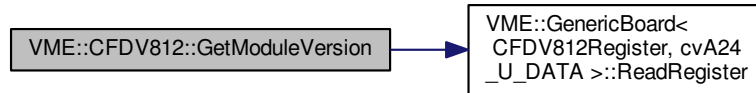
7.6.3.5 unsigned short VME::CFDV812::GetModuleType () const

Here is the call graph for this function:



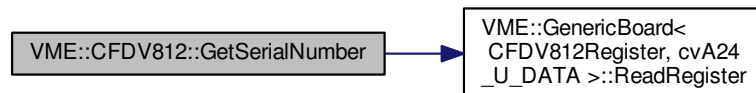
7.6.3.6 unsigned short VME::CFDV812::GetModuleVersion () const

Here is the call graph for this function:



7.6.3.7 unsigned short VME::CFDV812::GetSerialNumber () const

Here is the call graph for this function:



7.6.3.8 float VME::CFDV812::OutputWidthCalculator (unsigned short *value*) const [private]

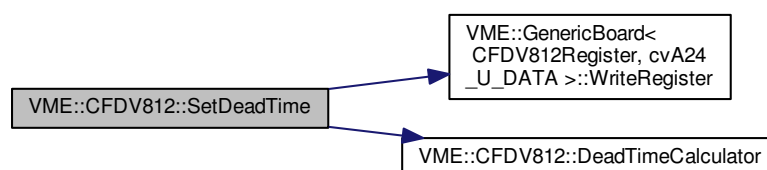
7.6.3.9 void VME::CFDV812::SetDeadTime (unsigned short *group_id*, unsigned short *value*) const

Set the discrimination dead time for one group of 8 channels.

Parameters

<i>in</i>	<i>group_id</i>	Group of 8 channels (either 0 for 0-7, or 1 for 8-15)
-----------	-----------------	---

Here is the call graph for this function:



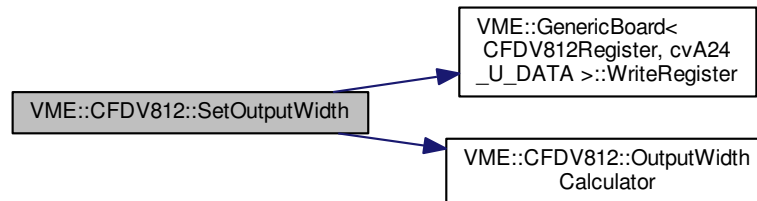
7.6.3.10 void VME::CFDV812::SetOutputWidth (unsigned short *group_id*, unsigned short *value*) const

Set the discriminated pulse output width for one group of 8 channels.

Parameters

<i>in</i>	<i>group_id</i>	Group of 8 channels (either 0 for 0-7, or 1 for 8-15)
-----------	-----------------	---

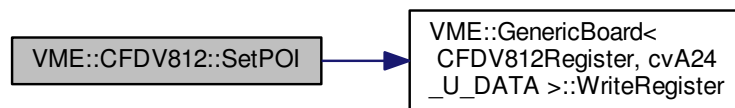
Here is the call graph for this function:



7.6.3.11 void VME::CFDV812::SetPOI (unsigned short *poi*) const

Set the pattern of inhibit (list of enabled channels)

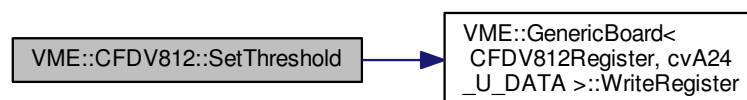
Here is the call graph for this function:



7.6.3.12 void VME::CFDV812::SetThreshold (unsigned short *channel_id*, unsigned short *value*) const

Set the threshold for one single channel, in units of 1 mV.

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- include/VME_CFDV812.h

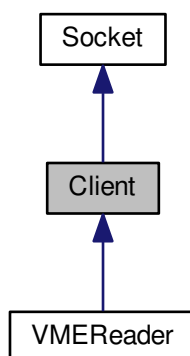
- `src/VME_CFDV812.cpp`

7.7 Client Class Reference

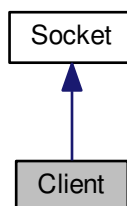
Base client object for the socket.

```
#include <Client.h>
```

Inheritance diagram for Client:



Collaboration diagram for Client:



Public Member Functions

- `Client ()`
General void client constructor.
- `Client (int port)`
Bind a socket client to a given port.
- `virtual ~Client ()`
- `bool Connect (const SocketType &type=CLIENT)`
Bind this client to the socket.

- void `Disconnect` ()
Unbind this client from the socket.
- void `Send` (const `Message` &m) const
Send a message to the master through the socket.
- void `Send` (const Exception &e) const
- `SocketMessage SendAndReceive` (const `SocketMessage` &m, const `MessageKey` &a) const
- void `Receive` ()
Receive a socket message from the master.
- virtual void `ParseMessage` (const `SocketMessage` &m)
Parse a `SocketMessage` received from the master.
- virtual `SocketType GetType` () const
`Socket` actor type retrieval method.

Private Member Functions

- void `Announce` ()
Announce our entry on the socket to its master.

Private Attributes

- int `fClientId`
- bool `flsConnected`
- `SocketType fType`

Additional Inherited Members

7.7.1 Detailed Description

Base client object for the socket.

`Client` object used by the server to send/receive commands from the messenger/broadcaster.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

24 Mar 2015

7.7.2 Constructor & Destructor Documentation

7.7.2.1 `Client::Client ()` `[inline]`

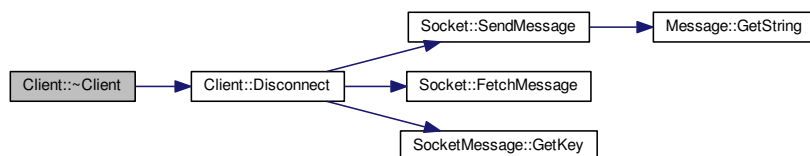
General void client constructor.

7.7.2.2 `Client::Client (int port)`

Bind a socket client to a given port.

7.7.2.3 Client::~~Client () [virtual]

Here is the call graph for this function:

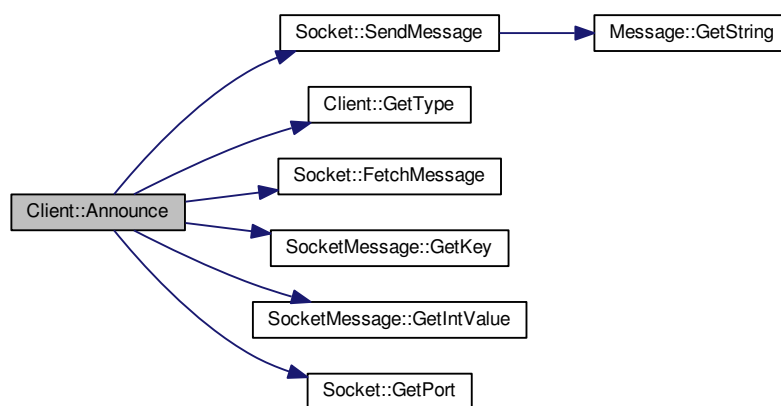


7.7.3 Member Function Documentation

7.7.3.1 void Client::Announce () [private]

Announce our entry on the socket to its master.

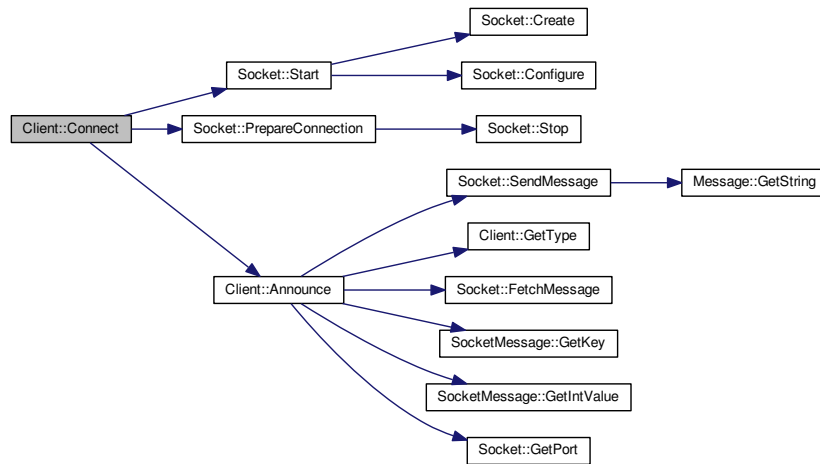
Here is the call graph for this function:



7.7.3.2 bool Client::Connect (const SocketType & type = CLIENT)

Bind this client to the socket.

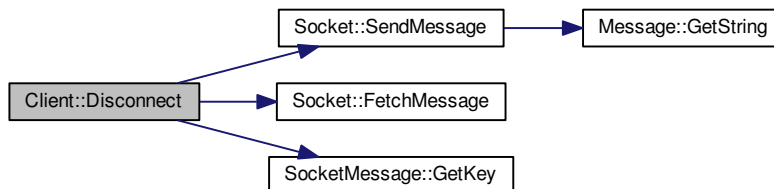
Here is the call graph for this function:



7.7.3.3 void Client::Disconnect ()

Unbind this client from the socket.

Here is the call graph for this function:



7.7.3.4 virtual SocketType Client::GetType () const [inline],[virtual]

`Socket` actor type retrieval method.

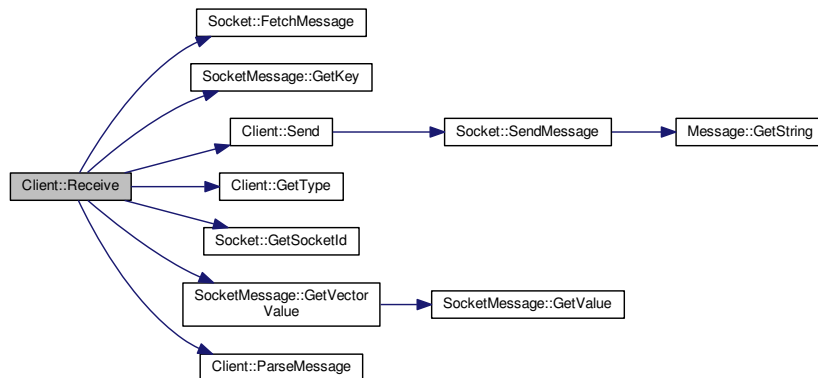
7.7.3.5 virtual void Client::ParseMessage (const SocketMessage & m) [inline],[virtual]

Parse a `SocketMessage` received from the master.

7.7.3.6 void Client::Receive ()

Receive a socket message from the master.

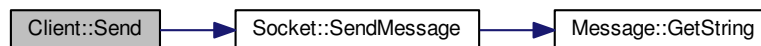
Here is the call graph for this function:



7.7.3.7 void Client::Send (const Message & m) const [inline]

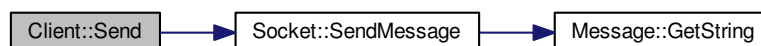
Send a message to the master through the socket.

Here is the call graph for this function:



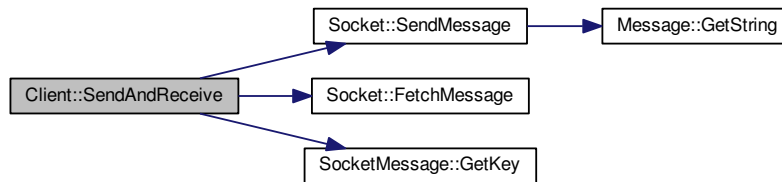
7.7.3.8 void Client::Send (const Exception & e) const [inline]

Here is the call graph for this function:



7.7.3.9 **SocketMessage** Client::SendAndReceive (const SocketMessage & m, const MessageKey & a) const
[inline]

Here is the call graph for this function:



7.7.4 Field Documentation

7.7.4.1 **int** Client::fClientId [private]

7.7.4.2 **bool** Client::fIsConnected [private]

7.7.4.3 **SocketType** Client::fType [private]

The documentation for this class was generated from the following files:

- include/Client.h
- src/Client.cpp

7.8 file_header_t Struct Reference

Header to the output files.

```
#include <FileConstants.h>
```

Data Fields

- uint32_t magic
- uint32_t run_id
- uint32_t spill_id
- uint8_t num_hptdc
- VME::AcquisitionMode acq_mode
- VME::DetectionMode det_mode

7.8.1 Detailed Description

Header to the output files.

General header to store in each collected data file for offline readout. It enable any reader to retrieve the run/spill number, as well as the HPTDC configuration during data collection.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

14 Apr 2015

7.8.2 Field Documentation

7.8.2.1 **VME::AcquisitionMode** file_header_t::acq_mode

7.8.2.2 **VME::DetectionMode** file_header_t::det_mode

7.8.2.3 **uint32_t** file_header_t::magic

7.8.2.4 **uint8_t** file_header_t::num_hptdc

7.8.2.5 **uint32_t** file_header_t::run_id

7.8.2.6 **uint32_t** file_header_t::spill_id

The documentation for this struct was generated from the following file:

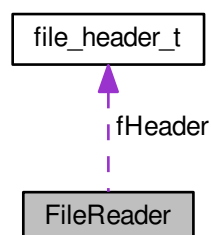
- include/FileConstants.h

7.9 FileReader Class Reference

Handler for a TDC output file readout.

```
#include <FileReader.h>
```

Collaboration diagram for FileReader:



Public Member Functions

- [FileReader](#) (std::string name)
Class constructor.
- [~FileReader](#) ()
- void [Dump](#) () const

- unsigned int [GetNumTDCs](#) () const
- unsigned long [GetNumEvents](#) () const
- bool [GetNextEvent](#) ([VME::TDCEvent](#) *)
- bool [GetNextMeasurement](#) (unsigned int channel_id, [VME::TDCMeasurement](#) *mc)

Fetch the next full measurement on a given channel.

Private Attributes

- std::ifstream [fFile](#)
- [file_header_t](#) [fHeader](#)
- [VME::AcquisitionMode](#) [fReadoutMode](#)
- time_t [fWriteTime](#)
- unsigned long [fNumEvents](#)

7.9.1 Detailed Description

Handler for a TDC output file readout.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

Jun 2015

7.9.2 Constructor & Destructor Documentation

7.9.2.1 [FileReader::FileReader](#) ([std::string](#) *name*)

Class constructor.

Parameters

in	<i>name</i>	Path to the file to read
in	<i>ro</i>	Data readout mode (continuous storage or trigger matching)

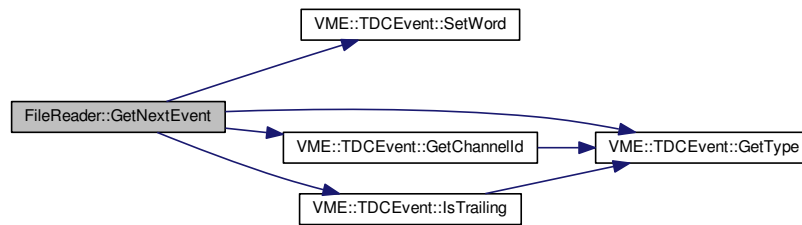
7.9.2.2 [FileReader::~~FileReader](#) ()

7.9.3 Member Function Documentation

7.9.3.1 [void FileReader::Dump](#) () const

7.9.3.2 bool FileReader::GetNextEvent (VME::TDCEvent * ev)

Here is the call graph for this function:



7.9.3.3 bool FileReader::GetNextMeasurement (unsigned int channel_id, VME::TDCMeasurement * mc)

Fetch the next full measurement on a given channel.

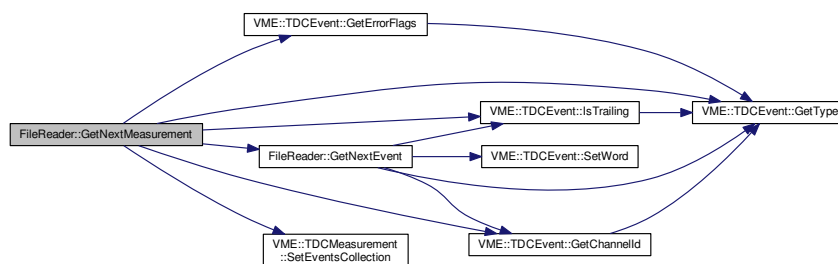
Parameters

in	<i>channel_id</i>	Unique identifier of the channel number to retrieve
out	<i>m</i>	A full measurement with leading, trailing times, ...

Returns

A boolean stating the success of retrieval operation

Here is the call graph for this function:



7.9.3.4 unsigned long FileReader::GetNumEvents () const [inline]

7.9.3.5 unsigned int FileReader::GetNumTDCs () const [inline]

7.9.4 Field Documentation

7.9.4.1 std::ifstream FileReader::fFile [private]

7.9.4.2 file_header_t FileReader::fHeader [private]

7.9.4.3 `unsigned long FileReader::fNumEvents` `[private]`

7.9.4.4 `VME::AcquisitionMode FileReader::fReadoutMode` `[private]`

7.9.4.5 `time_t FileReader::fWriteTime` `[private]`

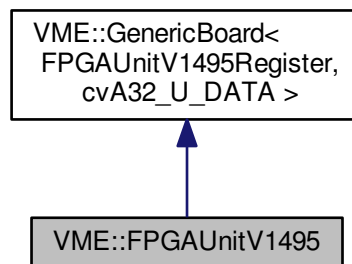
The documentation for this class was generated from the following files:

- `include/FileReader.h`
- `src/FileReader.cpp`

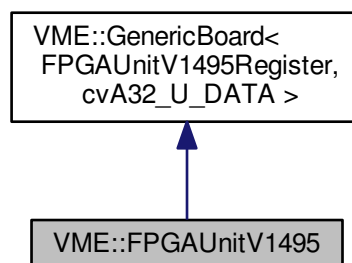
7.10 VME::FPGAUnitV1495 Class Reference

```
#include <VME_FPGAUnitV1495.h>
```

Inheritance diagram for VME::FPGAUnitV1495:



Collaboration diagram for VME::FPGAUnitV1495:



Public Types

- enum `TDCBits` { `kReset` =0x1, `kTrigger` =0x2, `kClear` =0x4 }

Public Member Functions

- [FPGAUnitV1495](#) (int32_t bhandle, uint32_t baseaddr)
- [~FPGAUnitV1495](#) ()
- unsigned short [GetCAENFirmwareRevision](#) () const
- unsigned short [GetUserFirmwareRevision](#) () const
- unsigned int [GetHardwareRevision](#) () const
- unsigned short [GetSerialNumber](#) () const
- unsigned short [GetGeoAddress](#) () const
- void [CheckBoardVersion](#) () const
- void [ResetFPGA](#) () const
- void [DumpFWInformation](#) () const
- void [SetTDCBits](#) (unsigned short bits) const
Set a pattern of bits to be sent to all TDCs through the ECL mezzanine.
- void [PulseTDCBits](#) (unsigned short bits, unsigned int time_us=10) const
Send a pulse to TDCs' front panel.
- unsigned short [GetTDCBits](#) () const
Retrieve the current bits sent to TDCs' front panel.
- [FPGAUnitV1495Control](#) [GetControl](#) () const
Retrieve the user-defined control word.
- void [SetControl](#) (const [FPGAUnitV1495Control](#) &control) const
Set the user-defined control word.
- void [SetInternalClockPeriod](#) (uint32_t period) const
Set the internal clock period.
- uint32_t [GetInternalClockPeriod](#) () const
Retrieve the internal clock period.
- void [SetInternalTriggerPeriod](#) (uint32_t period) const
Set the internal trigger period.
- uint32_t [GetInternalTriggerPeriod](#) () const
Retrieve the internal trigger period.
- uint32_t [GetOutputPulser](#) () const
- void [ClearOutputPulser](#) () const
- void [SetOutputPulser](#) (unsigned short id, bool enable=true) const
- void [SetOutputPulserPOI](#) (uint32_t poi) const
- uint32_t [GetOutputDelay](#) () const
- void [SetOutputDelay](#) (uint32_t delay) const
- void [StartScaler](#) ()
Start the inner triggers counter.
- void [StopScaler](#) ()
Stop the inner triggers counter.
- uint32_t [GetScalerValue](#) () const
Return the inner triggers counter value.

Private Attributes

- bool [fScalerStarted](#)

Additional Inherited Members

7.10.1 Detailed Description

Handler for the multi-purposes FPGA unit (CAEN V1495)

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

25 Jun 2015

7.10.2 Member Enumeration Documentation

7.10.2.1 enum VME::FPGAUnitV1495::TDCBits

Enumerator

kReset

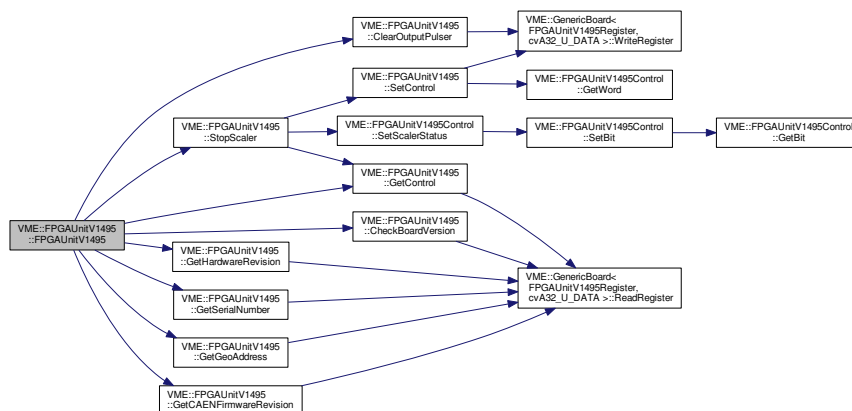
kTrigger

kClear

7.10.3 Constructor & Destructor Documentation

7.10.3.1 VME::FPGAUnitV1495::FPGAUnitV1495 (int32_t bhandle, uint32_t baseaddr)

Here is the call graph for this function:

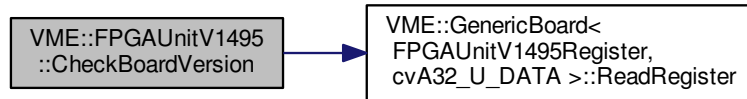


7.10.3.2 VME::FPGAUnitV1495::~~FPGAUnitV1495 ()

7.10.4 Member Function Documentation

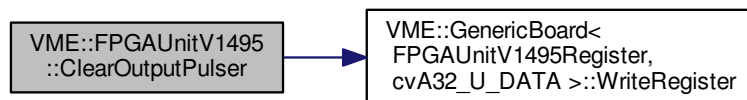
7.10.4.1 void VME::FPGAUnitV1495::CheckBoardVersion () const

Here is the call graph for this function:



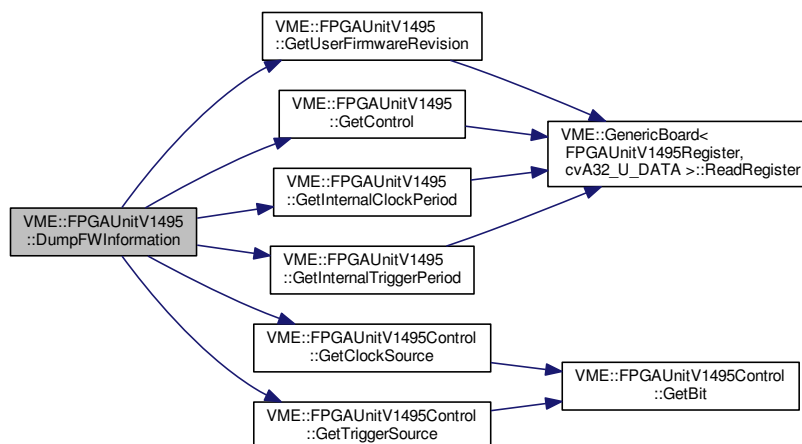
7.10.4.2 void VME::FPGAUnitV1495::ClearOutputPulser () const

Here is the call graph for this function:



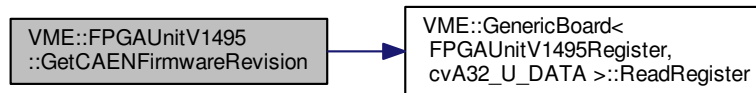
7.10.4.3 void VME::FPGAUnitV1495::DumpFWInformation () const

Here is the call graph for this function:



7.10.4.4 unsigned short VME::FPGAUnitV1495::GetCAENFirmwareRevision () const

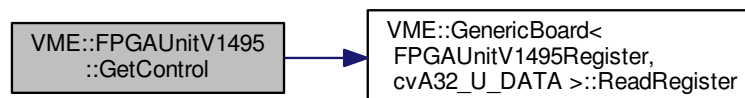
Here is the call graph for this function:



7.10.4.5 FPGAUnitV1495Control VME::FPGAUnitV1495::GetControl () const

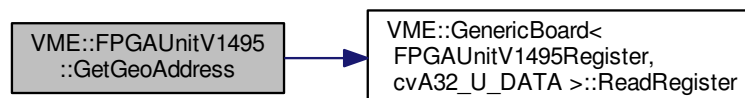
Retrieve the user-defined control word.

Here is the call graph for this function:



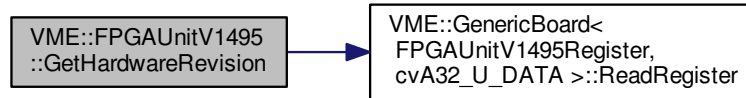
7.10.4.6 unsigned short VME::FPGAUnitV1495::GetGeoAddress () const

Here is the call graph for this function:



7.10.4.7 unsigned int VME::FPGAUnitV1495::GetHardwareRevision () const

Here is the call graph for this function:



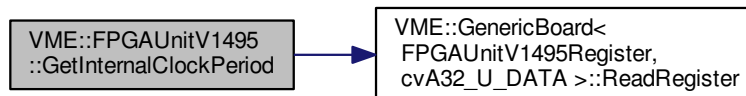
7.10.4.8 uint32_t VME::FPGAUnitV1495::GetInternalClockPeriod () const

Retrieve the internal clock period.

Returns

Clock period (in units of 25 ns)

Here is the call graph for this function:



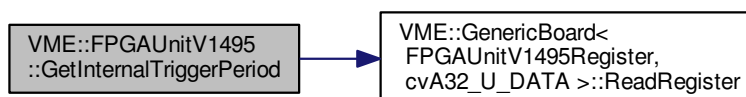
7.10.4.9 uint32_t VME::FPGAUnitV1495::GetInternalTriggerPeriod () const

Retrieve the internal trigger period.

Returns

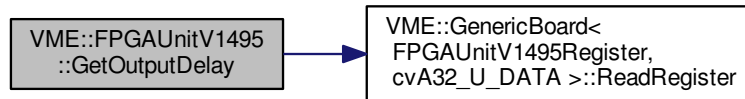
Trigger period (in units of 50 ns)

Here is the call graph for this function:



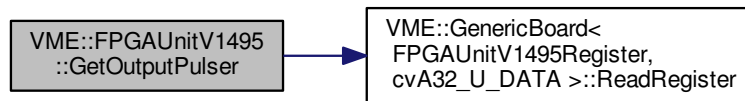
7.10.4.10 uint32_t VME::FPGAUnitV1495::GetOutputDelay () const

Here is the call graph for this function:



7.10.4.11 uint32_t VME::FPGAUnitV1495::GetOutputPulser () const

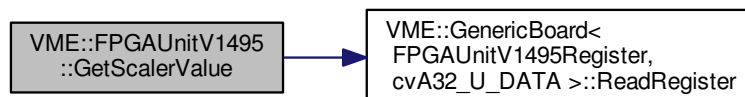
Here is the call graph for this function:



7.10.4.12 uint32_t VME::FPGAUnitV1495::GetScalerValue () const

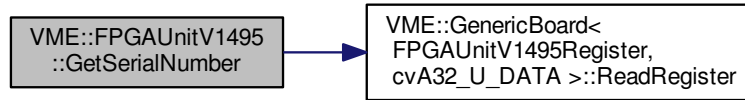
Return the inner triggers counter value.

Here is the call graph for this function:



7.10.4.13 unsigned short VME::FPGAUnitV1495::GetSerialNumber () const

Here is the call graph for this function:



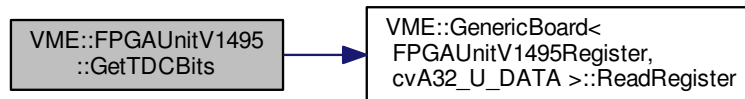
7.10.4.14 unsigned short VME::FPGAUnitV1495::GetTDCBits () const

Retrieve the current bits sent to TDCs' front panel.

Returns

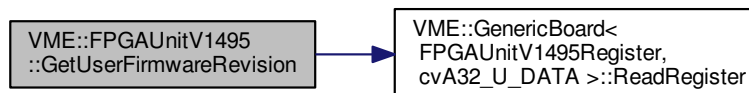
A 3-bit word Pol

Here is the call graph for this function:



7.10.4.15 unsigned short VME::FPGAUnitV1495::GetUserFirmwareRevision () const

Here is the call graph for this function:

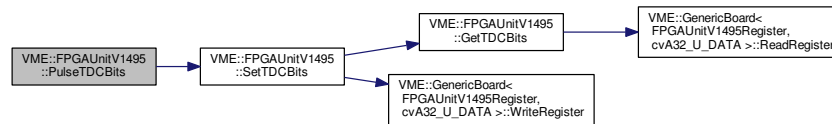
7.10.4.16 void VME::FPGAUnitV1495::PulseTDCBits (unsigned short *bits*, unsigned int *time_us* = 10) const

Send a pulse to TDCs' front panel.

Parameters

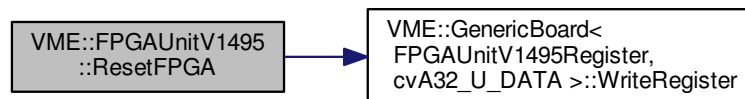
in	<i>bits</i>	The pattern to send (3 bits)
in	<i>time_us</i>	Pulse width (in us)

Here is the call graph for this function:



7.10.4.17 void VME::FPGAUnitV1495::ResetFPGA () const

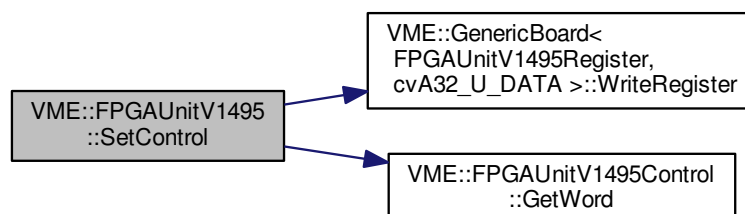
Here is the call graph for this function:



7.10.4.18 void VME::FPGAUnitV1495::SetControl (const FPGAUnitV1495Control & control) const

Set the user-defined control word.

Here is the call graph for this function:



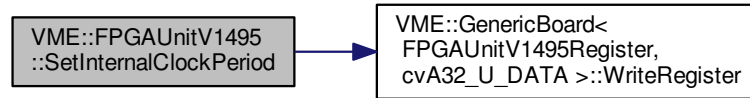
7.10.4.19 void VME::FPGAUnitV1495::SetInternalClockPeriod (uint32_t period) const

Set the internal clock period.

Parameters

<i>in</i>	<i>period</i>	Clock period (in units of 25 ns)
-----------	---------------	----------------------------------

Here is the call graph for this function:



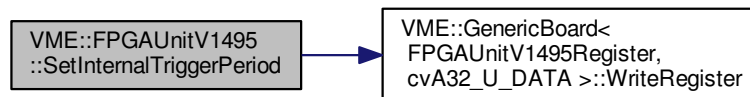
7.10.4.20 void VME::FPGAUnitV1495::SetInternalTriggerPeriod (uint32_t *period*) const

Set the internal trigger period.

Parameters

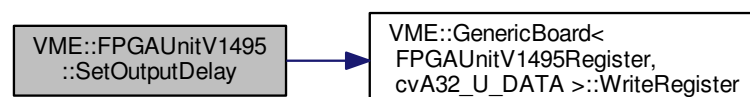
<i>in</i>	<i>period</i>	Trigger period (in units of 50 ns)
-----------	---------------	------------------------------------

Here is the call graph for this function:



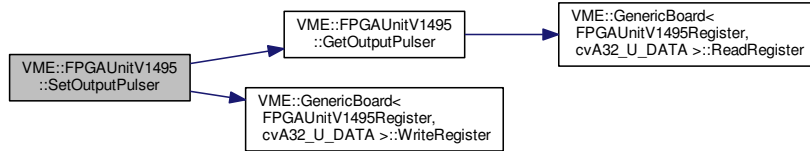
7.10.4.21 void VME::FPGAUnitV1495::SetOutputDelay (uint32_t *delay*) const

Here is the call graph for this function:



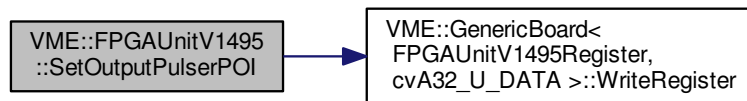
7.10.4.22 `void VME::FPGAUnitV1495::SetOutputPulser (unsigned short id, bool enable = true) const`

Here is the call graph for this function:



7.10.4.23 `void VME::FPGAUnitV1495::SetOutputPulserPOI (uint32_t poi) const`

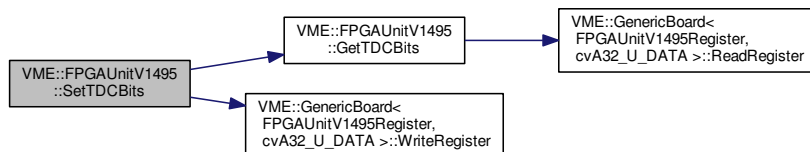
Here is the call graph for this function:



7.10.4.24 `void VME::FPGAUnitV1495::SetTDCBits (unsigned short bits) const`

Set a pattern of bits to be sent to all TDCs through the ECL mezzanine.

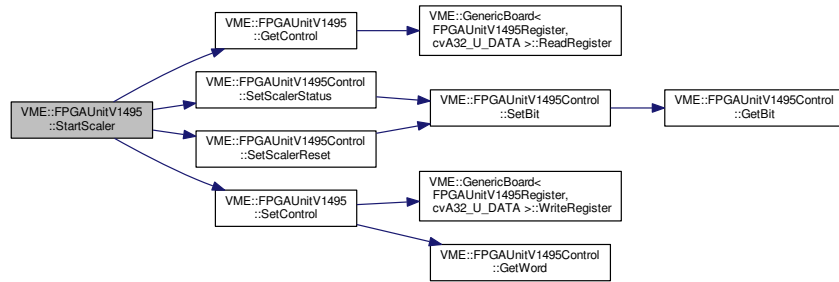
Here is the call graph for this function:



7.10.4.25 `void VME::FPGAUnitV1495::StartScaler ()`

Start the inner triggers counter.

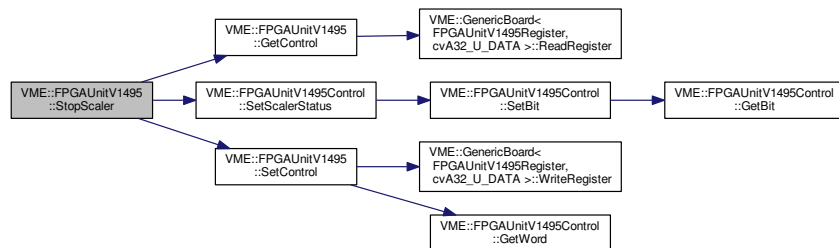
Here is the call graph for this function:



7.10.4.26 void VME::FPGAUnitV1495::StopScaler ()

Stop the inner triggers counter.

Here is the call graph for this function:



7.10.5 Field Documentation

7.10.5.1 bool VME::FPGAUnitV1495::fScalerStarted [private]

The documentation for this class was generated from the following files:

- include/VME_FPGAUnitV1495.h
- src/VME_FPGAUnitV1495.cpp

7.11 VME::FPGAUnitV1495Control Class Reference

```
#include <VME_FPGAUnitV1495.h>
```

Public Types

- enum [ClockSource](#) { [InternalClock](#) =0x0, [ExternalClock](#) =0x1 }
- enum [TriggerSource](#) { [InternalTrigger](#) =0x0, [ExternalTrigger](#) =0x1 }
- enum [SignalSource](#) { [InternalSignal](#) =0x0, [ExternalSignal](#) =0x1 }

Public Member Functions

- [FPGAUnitV1495Control](#) (uint32_t word)
- virtual [~FPGAUnitV1495Control](#) ()
- void [Dump](#) () const
- uint32_t [GetWord](#) () const
- [ClockSource](#) [GetClockSource](#) () const
Get the clock source.
- void [SetClockSource](#) (const [ClockSource](#) &cs)
Switch between internal and external clock source.
- [TriggerSource](#) [GetTriggerSource](#) () const
Get the trigger source.
- void [SetTriggerSource](#) (const [TriggerSource](#) &cs)
Switch between internal and external trigger source.
- bool [GetScalerStatus](#) () const
- void [SetScalerStatus](#) (bool start=true)
- void [SetScalerReset](#) (bool reset=true)
- [SignalSource](#) [GetSignalSource](#) (unsigned short map_id) const
- void [SetSignalSource](#) (unsigned short map_id, const [SignalSource](#) &s)

Private Member Functions

- bool [GetBit](#) (unsigned short id) const
- void [SetBit](#) (unsigned short id, unsigned short value=0x1)

Private Attributes

- uint32_t [fWord](#)

7.11.1 Detailed Description

User-defined control word to be propagated to the CAEN V1495 board firmware.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

27 Jun 2015

7.11.2 Member Enumeration Documentation

7.11.2.1 enum VME::FPGAUnitV1495Control::ClockSource

Enumerator

InternalClock

ExternalClock

7.11.2.2 enum VME::FPGAUnitV1495Control::SignalSource

Enumerator

InternalSignal
ExternalSignal

7.11.2.3 enum VME::FPGAUnitV1495Control::TriggerSource

Enumerator

InternalTrigger
ExternalTrigger

7.11.3 Constructor & Destructor Documentation

7.11.3.1 VME::FPGAUnitV1495Control::FPGAUnitV1495Control (uint32_t word) [inline]

7.11.3.2 virtual VME::FPGAUnitV1495Control::~~FPGAUnitV1495Control () [inline], [virtual]

7.11.4 Member Function Documentation

7.11.4.1 void VME::FPGAUnitV1495Control::Dump () const [inline]

7.11.4.2 bool VME::FPGAUnitV1495Control::GetBit (unsigned short id) const [inline], [private]

7.11.4.3 ClockSource VME::FPGAUnitV1495Control::GetClockSource () const [inline]

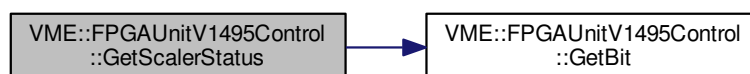
Get the clock source.

Here is the call graph for this function:



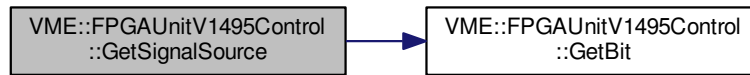
7.11.4.4 bool VME::FPGAUnitV1495Control::GetScalerStatus () const [inline]

Here is the call graph for this function:



7.11.4.5 SignalSource VME::FPGAUnitV1495Control::GetSignalSource (unsigned short *map_id*) const [inline]

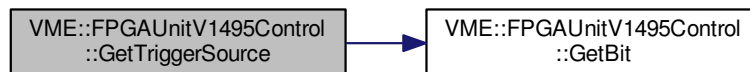
Here is the call graph for this function:



7.11.4.6 TriggerSource VME::FPGAUnitV1495Control::GetTriggerSource () const [inline]

Get the trigger source.

Here is the call graph for this function:



7.11.4.7 uint32_t VME::FPGAUnitV1495Control::GetWord () const [inline]

7.11.4.8 void VME::FPGAUnitV1495Control::SetBit (unsigned short *id*, unsigned short *value* = 0x1) [inline], [private]

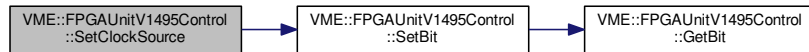
Here is the call graph for this function:



7.11.4.9 void VME::FPGAUnitV1495Control::SetClockSource (const ClockSource & *cs*) [inline]

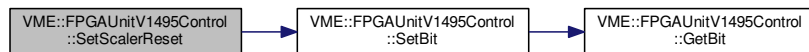
Switch between internal and external clock source.

Here is the call graph for this function:



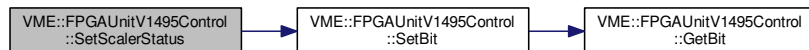
7.11.4.10 void VME::FPGAUnitV1495Control::SetScalerReset (bool *reset* = true) [inline]

Here is the call graph for this function:



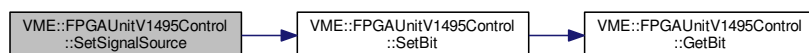
7.11.4.11 void VME::FPGAUnitV1495Control::SetScalerStatus (bool *start* = true) [inline]

Here is the call graph for this function:



7.11.4.12 void VME::FPGAUnitV1495Control::SetSignalSource (unsigned short *map_id*, const SignalSource & *s*) [inline]

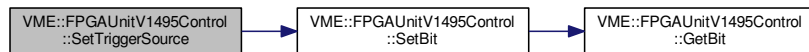
Here is the call graph for this function:



7.11.4.13 void VME::FPGAUnitV1495Control::SetTriggerSource (const TriggerSource & *cs*) [inline]

Switch between internal and external trigger source.

Here is the call graph for this function:



7.11.5 Field Documentation

7.11.5.1 uint32_t VME::FPGAUnitV1495Control::fWord [private]

The documentation for this class was generated from the following file:

- include/VME_FPGAUnitV1495.h

7.12 VME::GenericBoard< Register, am > Class Template Reference

```
#include <VME_GenericBoard.h>
```

Public Member Functions

- [GenericBoard](#) (int32_t bhandle, uint32_t baseaddr)
- virtual [~GenericBoard](#) ()

Protected Member Functions

- void [WriteRegister](#) (const Register ®, const uint16_t &data) const
Write on register.
- void [WriteRegister](#) (const Register ®, const uint32_t &data) const
Write on register.
- void [ReadRegister](#) (const Register ®, uint16_t *data) const
Read on register.
- void [ReadRegister](#) (const Register ®, uint32_t *data) const
Read on register.

Protected Attributes

- int32_t [fHandle](#)
- uint32_t [fBaseAddr](#)

7.12.1 Constructor & Destructor Documentation

7.12.1.1 `template<class Register, CVAddressModifier am> VME::GenericBoard< Register, am >::GenericBoard (int32_t bhandle, uint32_t baseaddr) [inline]`

7.12.1.2 `template<class Register, CVAddressModifier am> virtual VME::GenericBoard< Register, am >::~~GenericBoard () [inline],[virtual]`

7.12.2 Member Function Documentation

7.12.2.1 `template<class Register, CVAddressModifier am> void VME::GenericBoard< Register, am >::ReadRegister (const Register & reg, uint16_t * data) const` `[inline], [protected]`

Read on register.

Read a 16-bit word in the register

Parameters

<i>in</i>	<i>addr</i>	register
<i>out</i>	<i>data</i>	word

7.12.2.2 `template<class Register, CVAddressModifier am> void VME::GenericBoard< Register, am >::ReadRegister (const Register & reg, uint32_t * data) const` `[inline], [protected]`

Read on register.

Read a 32-bit word in the register

Parameters

<i>in</i>	<i>addr</i>	register
<i>out</i>	<i>data</i>	word

7.12.2.3 `template<class Register, CVAddressModifier am> void VME::GenericBoard< Register, am >::WriteRegister (const Register & reg, const uint16_t & data) const` `[inline], [protected]`

Write on register.

Write a 16-bit word in the register

Parameters

<i>in</i>	<i>addr</i>	register
<i>in</i>	<i>data</i>	word

7.12.2.4 `template<class Register, CVAddressModifier am> void VME::GenericBoard< Register, am >::WriteRegister (const Register & reg, const uint32_t & data) const` `[inline], [protected]`

Write on register.

Write a 32-bit word in the register

Parameters

<i>in</i>	<i>addr</i>	register
<i>in</i>	<i>data</i>	word

7.12.3 Field Documentation

7.12.3.1 `template<class Register, CVAddressModifier am> uint32_t VME::GenericBoard< Register, am >::fBaseAddr` `[protected]`

```
7.12.3.2  template<class Register, CVAddressModifier am> int32_t VME::GenericBoard< Register, am >::fHandle
[protected]
```

The documentation for this class was generated from the following file:

- include/VME_GenericBoard.h

7.13 VME::GlobalOffset Struct Reference

```
#include <VME_TDCV1x90.h>
```

Data Fields

- uint16_t [coarse](#)
- uint16_t [fine](#)

7.13.1 Field Documentation

7.13.1.1 uint16_t VME::GlobalOffset::coarse

7.13.1.2 uint16_t VME::GlobalOffset::fine

The documentation for this struct was generated from the following file:

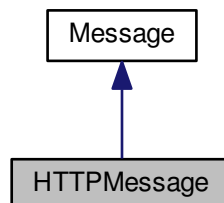
- include/VME_TDCV1x90.h

7.14 HTTPMessage Class Reference

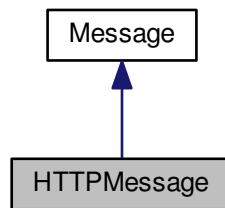
[Message](#) to be transmitted through a WebSocket protocol.

```
#include <HTTPMessage.h>
```

Inheritance diagram for HTTPMessage:



Collaboration diagram for HTTPMessage:



Public Member Functions

- `HTTPMessage` (WebSocket *ws, `Message` m, MessageAction a)
- `HTTPMessage` (WebSocket *ws, const char *msg, MessageAction a)
- void `Decode` ()
- void `Encode` ()
- MessageKey `GetKey` () const
- void `Dump` (std::ostream &os=std::cout) const

Private Attributes

- WebSocket * `fWS`
- std::string `fOriginalString`

Additional Inherited Members

7.14.1 Detailed Description

`Message` to be transmitted through a WebSocket protocol.

Type of message compatible to the transmission through a WebSocket protocol. It enables a direct conversion of standards from any socket message format used elsewhere in this code using the *MessageAction* statement.

Author

Laurent Forthomme laurent.forthomme@cern.ch

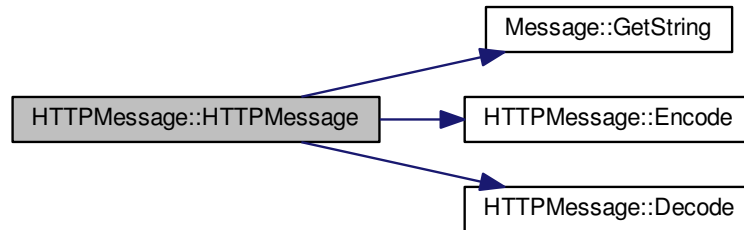
Date

1 Apr 2015

7.14.2 Constructor & Destructor Documentation

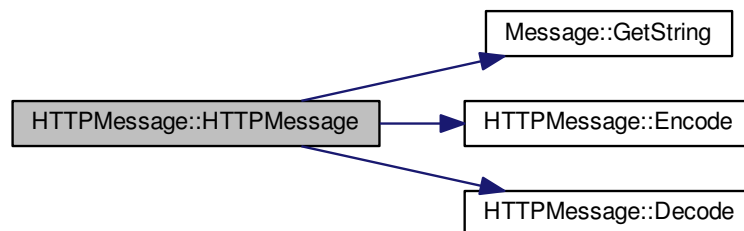
7.14.2.1 HTTPMessage::HTTPMessage (WebSocket * *ws*, Message *m*, MessageAction *a*) [inline]

Here is the call graph for this function:



7.14.2.2 HTTPMessage::HTTPMessage (WebSocket * *ws*, const char * *msg*, MessageAction *a*) [inline]

Here is the call graph for this function:



7.14.3 Member Function Documentation

7.14.3.1 void HTTPMessage::Decode () [inline]

7.14.3.2 void HTTPMessage::Dump (std::ostream & *os* = std::cout) const [inline]

7.14.3.3 void HTTPMessage::Encode () [inline]

7.14.3.4 MessageKey HTTPMessage::GetKey () const [inline]

7.14.4 Field Documentation

7.14.4.1 std::string HTTPMessage::fOriginalString [private]

7.14.4.2 WebSocket* HTTPMessage::fWS [private]

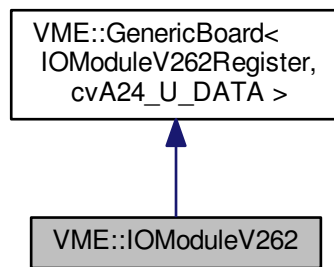
The documentation for this class was generated from the following file:

- include/HTTPMessage.h

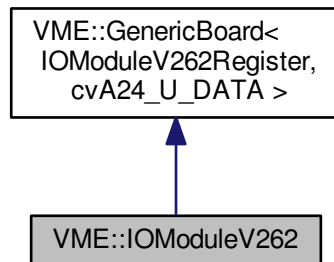
7.15 VME::IOModuleV262 Class Reference

```
#include <VME_IOModuleV262.h>
```

Inheritance diagram for VME::IOModuleV262:



Collaboration diagram for VME::IOModuleV262:



Public Member Functions

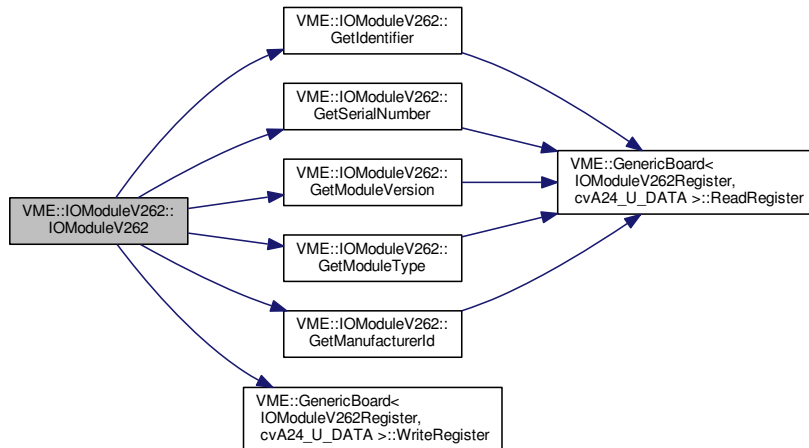
- [IOModuleV262](#) (int32_t bhandle, uint32_t baseaddr)
- [~IOModuleV262](#) ()
- unsigned short [GetSerialNumber](#) () const
- unsigned short [GetModuleVersion](#) () const
- unsigned short [GetModuleType](#) () const
- unsigned short [GetManufacturerId](#) () const
- unsigned short [GetIdentifier](#) () const

Additional Inherited Members

7.15.1 Constructor & Destructor Documentation

7.15.1.1 VME::IOModuleV262::IOModuleV262 (int32_t *bhandle*, uint32_t *baseaddr*)

Here is the call graph for this function:

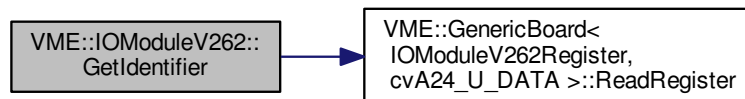


7.15.1.2 VME::IOModuleV262::~~IOModuleV262 () [inline]

7.15.2 Member Function Documentation

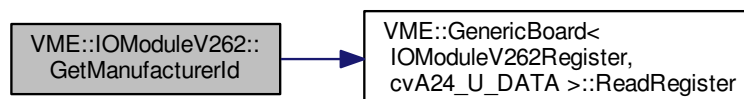
7.15.2.1 unsigned short VME::IOModuleV262::GetIdentifier () const

Here is the call graph for this function:



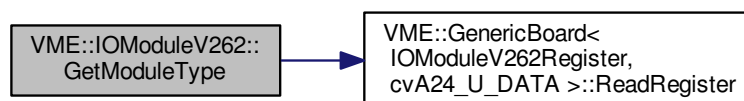
7.15.2.2 unsigned short VME::IOModuleV262::GetManufacturerId () const

Here is the call graph for this function:



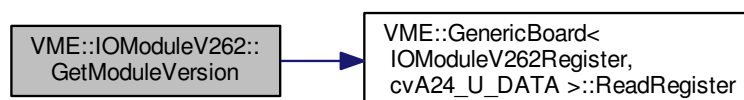
7.15.2.3 unsigned short VME::IOModuleV262::GetModuleType () const

Here is the call graph for this function:



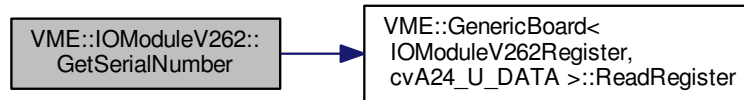
7.15.2.4 unsigned short VME::IOModuleV262::GetModuleVersion () const

Here is the call graph for this function:



7.15.2.5 unsigned short VME::IOModuleV262::GetSerialNumber () const

Here is the call graph for this function:



The documentation for this class was generated from the following files:

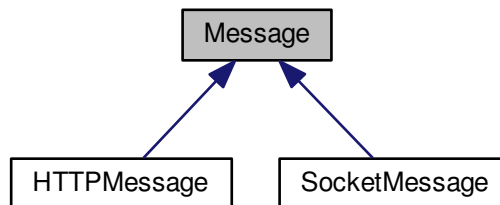
- `include/VME_IOModuleV262.h`
- `src/VME_IOModuleV262.cpp`

7.16 Message Class Reference

Base socket message type.

```
#include <Message.h>
```

Inheritance diagram for Message:



Public Member Functions

- [Message](#) ()
Void message constructor.
- [Message](#) (const char *msg)
Construct a message from a string.
- [Message](#) (std::string msg)
Construct a message from a string.
- virtual [~Message](#) ()
- MessageKey [GetKey](#) () const
Placeholder for the MessageKey retrieval method.
- std::string [GetString](#) () const

Retrieve the string carried by this message as a whole.

- bool `IsFromWeb` () const

Extract from any message its potential arrival from a WebSocket protocol.

- void `Dump` (std::ostream &os=std::cout) const

Protected Attributes

- std::string `fString`

7.16.1 Detailed Description

Base socket message type.

Base handler for messages to be transmitted through the socket

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

6 Apr 2015

7.16.2 Constructor & Destructor Documentation

7.16.2.1 `Message::Message ()` [inline]

Void message constructor.

7.16.2.2 `Message::Message (const char * msg)` [inline]

Construct a message from a string.

7.16.2.3 `Message::Message (std::string msg)` [inline]

Construct a message from a string.

7.16.2.4 `virtual Message::~~Message ()` [inline],[virtual]

7.16.3 Member Function Documentation

7.16.3.1 `void Message::Dump (std::ostream & os = std::cout) const` [inline]

7.16.3.2 `MessageKey Message::GetKey () const` [inline]

Placeholder for the MessageKey retrieval method.

7.16.3.3 `std::string Message::GetString () const` [inline]

Retrieve the string carried by this message as a whole.

7.16.3.4 `bool Message::IsFromWeb () const [inline]`

Extract from any message its potential arrival from a WebSocket protocol.

7.16.4 Field Documentation

7.16.4.1 `std::string Message::fString [protected]`

The documentation for this class was generated from the following file:

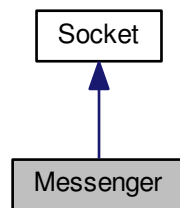
- `include/Message.h`

7.17 Messenger Class Reference

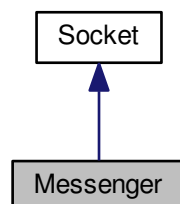
Base master object for the socket.

```
#include <Messenger.h>
```

Inheritance diagram for Messenger:



Collaboration diagram for Messenger:



Public Member Functions

- [Messenger \(\)](#)

- *Build a void master object or socket actor.*
- [Messenger](#) (int port)
- *Build a master object to control the socket.*
- [~Messenger](#) ()
- bool [Connect](#) ()
- *Connect the master to the socket.*
- void [Disconnect](#) ()
- *Remove the master and destroy the socket.*
- void [Send](#) (const [Message](#) &m, int sid) const
- *Send any type of message to any client.*
- void [Receive](#) ()
- *Handle a message reception from a client.*
- void [Broadcast](#) (const [Message](#) &m) const
- *Emit a message to all clients connected through the socket.*
- void [StartAcquisition](#) ()
- *Start the data acquisition.*
- void [StopAcquisition](#) ()
- [SocketType](#) [GetType](#) () const
- *[Socket](#) actor type retrieval method.*

Private Member Functions

- void [AddClient](#) ()
- *Add a client to listen to.*
- void [DisconnectClient](#) (int sid, MessageKey key, bool force=false)
- *Disconnect a client.*
- void [SwitchClientType](#) (int sid, [Socket::SocketType](#) type)
- void [ProcessMessage](#) ([SocketMessage](#) m, int sid)
- *Process a message received from the socket.*

Private Attributes

- [WebSocket](#) * [fWS](#)
- int [fNumAttempts](#)
- pid_t [fPID](#)
- int [fStdoutPipe](#) [2]
- int [fStderrPipe](#) [2]

Additional Inherited Members

7.17.1 Detailed Description

Base master object for the socket.

Messenger/broadcaster object used by the server to send/receive commands from the clients/listeners.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

23 Mar 2015

7.17.2 Constructor & Destructor Documentation

7.17.2.1 Messenger::Messenger ()

Build a void master object or socket actor.

7.17.2.2 Messenger::Messenger (int *port*)

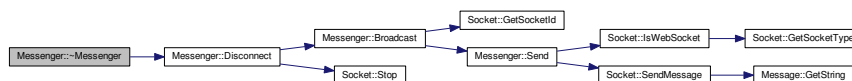
Build a master object to control the socket.

Here is the call graph for this function:



7.17.2.3 Messenger::~~Messenger ()

Here is the call graph for this function:



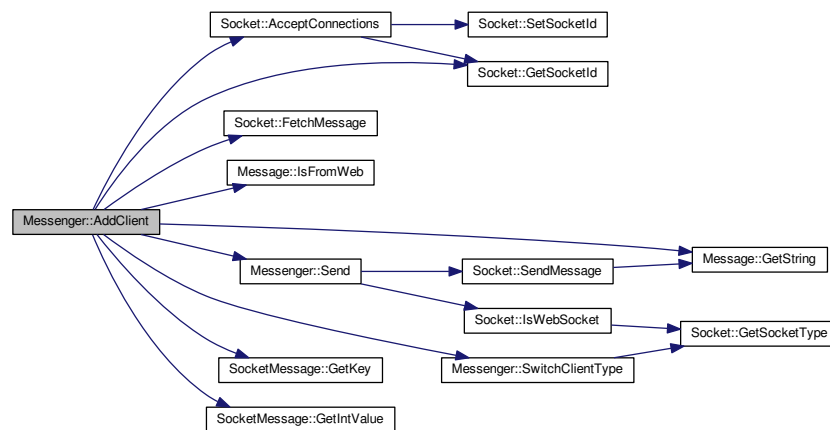
7.17.3 Member Function Documentation

7.17.3.1 void Messenger::AddClient () [private]

Add a client to listen to.

Add one client to the list of socket actors to monitor for message retrieval/submission.

Here is the call graph for this function:



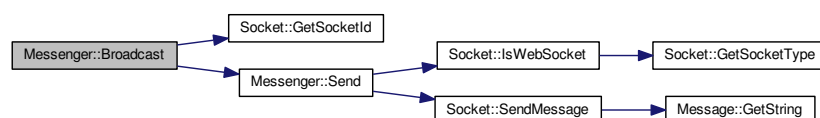
7.17.3.2 void Messenger::Broadcast (const Message & m) const

Emit a message to all clients connected through the socket.

Parameters

in	<i>m</i>	Message to transmit
----	----------	-------------------------------------

Here is the call graph for this function:

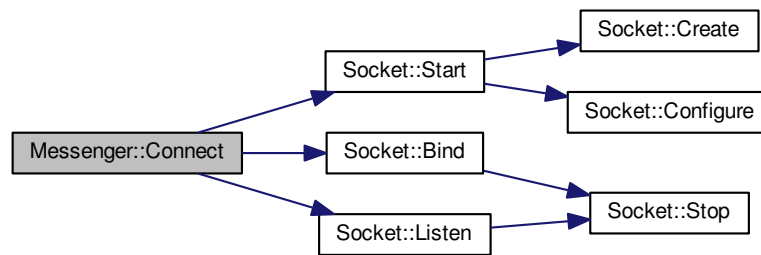


7.17.3.3 bool Messenger::Connect ()

Connect the master to the socket.

Connect this master to the socket for clients to be able to bind.

Here is the call graph for this function:

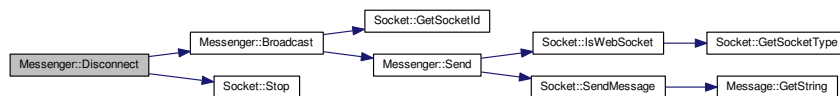


7.17.3.4 void Messenger::Disconnect ()

Remove the master and destroy the socket.

Remove this master from the socket, thus disconnecting automatically the clients connected.

Here is the call graph for this function:



7.17.3.5 void Messenger::DisconnectClient (int sid, MessageKey key, bool force = false) [private]

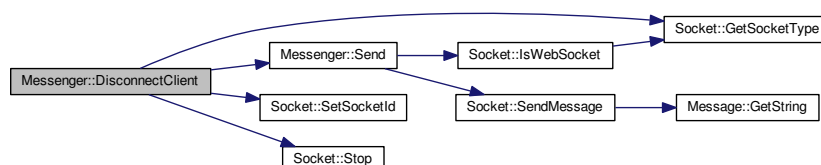
Disconnect a client.

Ask to a client to disconnect from this socket.

Parameters

in	<i>sid</i>	Unique identifier of the client to disconnect
in	<i>key</i>	Key to the message to transmit for disconnection
in	<i>force</i>	Do we need to force the client out of this socket ?

Here is the call graph for this function:



7.17.3.6 SocketType Messenger::GetType () const [inline]

Socket actor type retrieval method.

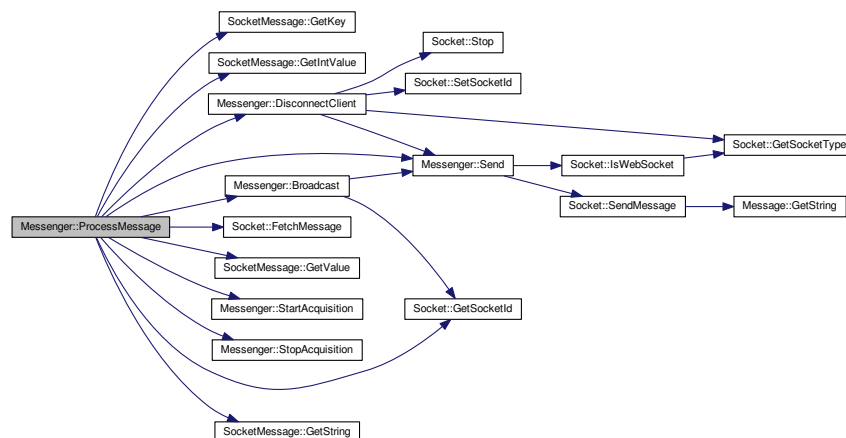
7.17.3.7 void Messenger::ProcessMessage (SocketMessage m, int sid) [private]

Process a message received from the socket.

Parameters

in	Unique	identifier of the client sending the message
----	--------	--

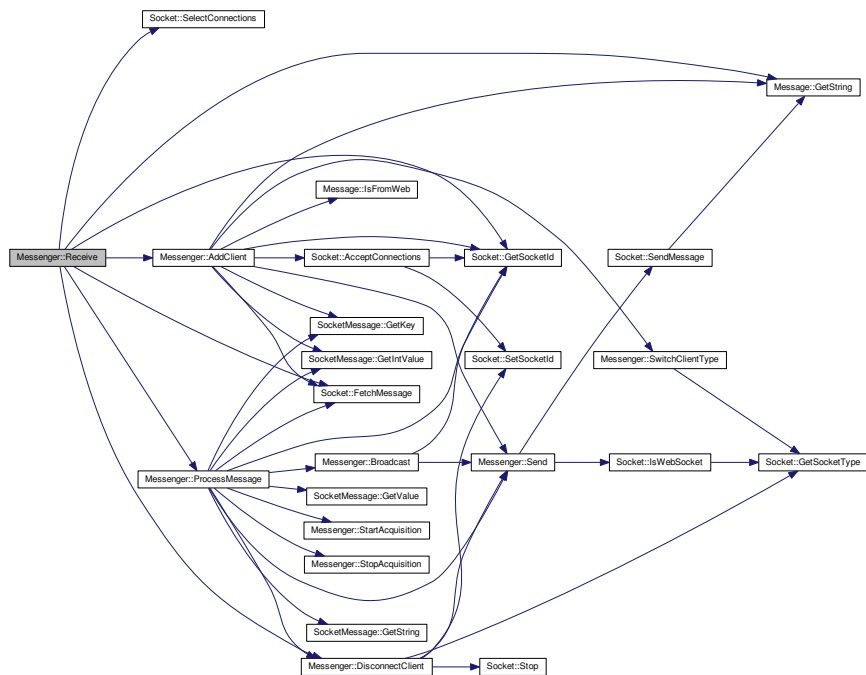
Here is the call graph for this function:



7.17.3.8 void Messenger::Receive ()

Handle a message reception from a client.

Here is the call graph for this function:



7.17.3.9 void Messenger::Send (const Message & m, int sid) const [inline]

Send any type of message to any client.

Parameters

in	<i>m</i>	Message to transmit
in	<i>sid</i>	Unique identifier of the client on this socket

Here is the call graph for this function:



7.17.3.10 void Messenger::StartAcquisition ()

Start the data acquisition.

7.17.3.11 void Messenger::StopAcquisition ()

7.17.3.12 `void Messenger::SwitchClientType (int sid, Socket::SocketType type)` [private]

Here is the call graph for this function:



7.17.4 Field Documentation

7.17.4.1 `int Messenger::fNumAttempts` [private]

7.17.4.2 `pid_t Messenger::fPID` [private]

7.17.4.3 `int Messenger::fStderrPipe[2]` [private]

7.17.4.4 `int Messenger::fStdoutPipe[2]` [private]

7.17.4.5 `WebSocket* Messenger::fWS` [private]

The documentation for this class was generated from the following files:

- include/Messenger.h
- src/Messenger.cpp

7.18 VME::PCIInterfaceA2818 Class Reference

```
#include <VME_PCIInterfaceA2818.h>
```

Public Member Functions

- [PCIInterfaceA2818](#) (const char *device)
- virtual [~PCIInterfaceA2818](#) ()
- std::string [GetFWRevision](#) () const

Private Attributes

- int [fHandle](#)

7.18.1 Constructor & Destructor Documentation

7.18.1.1 `VME::PCIInterfaceA2818::PCIInterfaceA2818 (const char * device)` [inline]

7.18.1.2 `virtual VME::PCIInterfaceA2818::~~PCIInterfaceA2818 ()` [inline], [virtual]

7.18.2 Member Function Documentation

7.18.2.1 `std::string VME::PCIInterfaceA2818::GetFWRevision () const [inline]`

7.18.3 Field Documentation

7.18.3.1 `int VME::PCIInterfaceA2818::fHandle [private]`

The documentation for this class was generated from the following file:

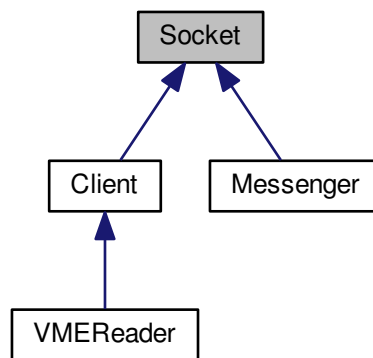
- `include/VME_PCIInterfaceA2818.h`

7.19 Socket Class Reference

Base socket object from which clients/master from a socket inherit.

```
#include <Socket.h>
```

Inheritance diagram for Socket:



Public Types

- enum `SocketType` {
`INVALID = -1`, `MASTER = 0`, `WEBSOCKET_CLIENT`, `CLIENT`,
`DETECTOR`, `DQM` }
Type of actor playing a role on the socket.
- typedef `std::set< std::pair< int, SocketType > > SocketCollection`

Public Member Functions

- `Socket ()`
- `Socket (int port)`
- virtual `~Socket ()`
- void `Stop ()`
Terminates the socket and all attached communications.
- void `SetPort (int port)`
- int `GetPort () const`

Retrieve the port used for this socket.

- void [AcceptConnections](#) ([Socket](#) &socket)

Accept connection from a client.

- void [SelectConnections](#) ()
- void [SetSocketId](#) (int sid)
- int [GetSocketId](#) () const
- [SocketType](#) [GetSocketType](#) (int sid) const
- bool [IsWebSocket](#) (int sid) const
- void [DumpConnected](#) () const

Protected Member Functions

- bool [Start](#) ()

Start the socket.

- void [Bind](#) ()

Bind a name to a socket.

- void [PrepareConnection](#) ()
- void [Listen](#) (int maxconn)

Listen to incoming messages.

- void [SendMessage](#) ([Message](#) message, int id=-1) const

Send a message on a socket.

- [Message](#) [FetchMessage](#) (int id=-1) const

Receive a message from a socket.

Protected Attributes

- int [fPort](#)
- char [fBuffer](#) [MAX_WORD_LENGTH]
- [SocketCollection](#) [fSocketsConnected](#)
- fd_set [fMaster](#)

Master file descriptor list.

- fd_set [fReadFds](#)

Temp file descriptor list for select()

Private Member Functions

- void [Create](#) ()

Create an endpoint for communication.

- void [Configure](#) ()

Configure the socket object for communication.

Private Attributes

- int [fSocketId](#)
- struct sockaddr_in [fAddress](#)

7.19.1 Detailed Description

Base socket object from which clients/master from a socket inherit.

General object providing all useful method to connect/bind/send/receive information through system sockets.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

23 Mar 2015

7.19.2 Member Typedef Documentation

7.19.2.1 `typedef std::set< std::pair<int,SocketType> > Socket::SocketCollection`

7.19.3 Constructor & Destructor Documentation

7.19.3.1 `Socket::Socket ()` `[inline]`

7.19.3.2 `Socket::Socket (int port)`

7.19.3.3 `Socket::~~Socket ()` `[virtual]`

7.19.4 Member Function Documentation

7.19.4.1 `void Socket::AcceptConnections (Socket & socket)`

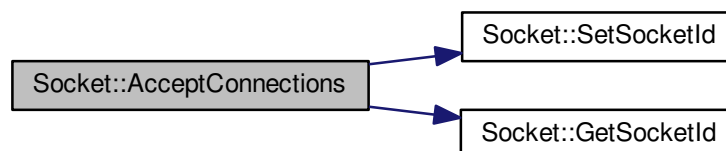
Accept connection from a client.

Set the socket to accept connections any client transmitting through the socket

Parameters

<code>in, out</code>	<code><i>socket</i></code>	Master/client object to enable on the socket
----------------------	----------------------------	--

Here is the call graph for this function:



7.19.4.2 `void Socket::Bind ()` `[protected]`

Bind a name to a socket.

Returns

Success of the operation

Here is the call graph for this function:



7.19.4.3 `void Socket::Configure () [private]`

Configure the socket object for communication.

7.19.4.4 `void Socket::Create () [private]`

Create an endpoint for communication.

7.19.4.5 `void Socket::DumpConnected () const`

7.19.4.6 `Message Socket::FetchMessage (int id = -1) const [protected]`

Receive a message from a socket.

Returns

Received message as a `std::string`

7.19.4.7 `int Socket::GetPort () const [inline]`

Retrieve the port used for this socket.

7.19.4.8 `int Socket::GetSocketId () const [inline]`

7.19.4.9 `SocketType Socket::GetSocketType (int sid) const [inline]`

7.19.4.10 `bool Socket::IsWebSocket (int sid) const` `[inline]`

Here is the call graph for this function:



7.19.4.11 `void Socket::Listen (int maxconn)` `[protected]`

Listen to incoming messages.

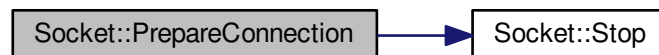
Set the socket to listen to any message coming from outside

Here is the call graph for this function:



7.19.4.12 `void Socket::PrepareConnection ()` `[protected]`

Here is the call graph for this function:



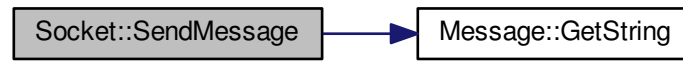
7.19.4.13 `void Socket::SelectConnections ()`

Register all open file descriptors to read their communication through the socket

7.19.4.14 `void Socket::SendMessage (Message message, int id = -1) const` `[protected]`

Send a message on a socket.

Here is the call graph for this function:



7.19.4.15 `void Socket::SetPort (int port)` `[inline]`

7.19.4.16 `void Socket::SetSocketId (int sid)` `[inline]`

7.19.4.17 `bool Socket::Start ()` `[protected]`

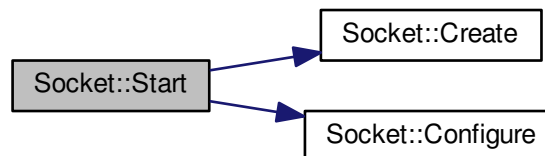
Start the socket.

Launch all mandatory operations to set the socket to be used

Returns

Success of the operation

Here is the call graph for this function:



7.19.4.18 `void Socket::Stop ()`

Terminates the socket and all attached communications.

7.19.5 Field Documentation

7.19.5.1 `struct sockaddr_in Socket::fAddress` `[private]`

7.19.5.2 `char Socket::fBuffer[MAX_WORD_LENGTH]` `[protected]`

7.19.5.3 `fd_set Socket::fMaster` `[protected]`

Master file descriptor list.

7.19.5.4 `int Socket::fPort` [protected]

7.19.5.5 `fd_set Socket::fReadFds` [protected]

Temp file descriptor list for select()

7.19.5.6 `int Socket::fSocketId` [private]

A file descriptor for this socket, if *Create* was performed beforehand.

7.19.5.7 **SocketCollection** `Socket::fSocketsConnected` [protected]

The documentation for this class was generated from the following files:

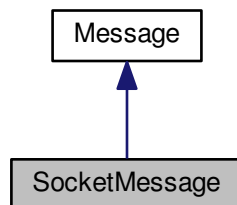
- include/Socket.h
- src/Socket.cpp

7.20 SocketMessage Class Reference

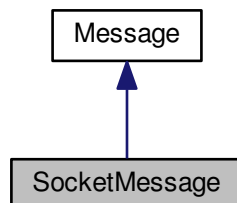
Socket-passed message type.

```
#include <SocketMessage.h>
```

Inheritance diagram for SocketMessage:



Collaboration diagram for SocketMessage:



Public Member Functions

- [SocketMessage](#) ()
- [SocketMessage](#) (const [Message](#) &msg)
- [SocketMessage](#) (const char *msg_s)
- [SocketMessage](#) (std::string msg_s)
- [SocketMessage](#) (const MessageKey &key)
Construct a socket message out of a key.
- [SocketMessage](#) (const MessageKey &key, const char *value)
Construct a socket message out of a key and a string-type value.
- [SocketMessage](#) (const MessageKey &key, std::string value)
Construct a socket message out of a key and a string-type value.
- [SocketMessage](#) (const MessageKey &key, const int value)
Construct a socket message out of a key and an integer-type value.
- [SocketMessage](#) (const MessageKey &key, const float value)
Construct a socket message out of a key and a float-type value.
- [SocketMessage](#) (const MessageKey &key, const double value)
Construct a socket message out of a key and a double precision-type value.
- [SocketMessage](#) (MessageMap msg_m)
Construct a socket message out of a map of key/string-type value.
- [~SocketMessage](#) ()
- void [SetKeyValue](#) (const MessageKey &key, const char *value)
String-valued message.
- void [SetKeyValue](#) (const MessageKey &key, int int_value)
Send an integer-valued message.
- void [SetKeyValue](#) (const MessageKey &key, float float_value)
Float-valued message.
- void [SetKeyValue](#) (const MessageKey &key, double double_value)
Double-valued message.
- std::string [GetString](#) () const
Extract the whole key:value message.
- MessageKey [GetKey](#) () const
Extract the message's key.
- std::string [GetValue](#) () const
Extract the message's string value.
- int [GetIntValue](#) () const
Extract the message's integer value.
- VectorValue [GetVectorValue](#) () const
Extract the message's vector of string value.
- void [Dump](#) (std::ostream &os=std::cout) const

Private Member Functions

- MessageMap [Object](#) () const
- std::string [String](#) () const

Private Attributes

- MessageMap [fMessage](#)

Additional Inherited Members

7.20.1 Detailed Description

Socket-passed message type.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

26 Mar 2015

7.20.2 Constructor & Destructor Documentation

7.20.2.1 `SocketMessage::SocketMessage ()` `[inline]`

7.20.2.2 `SocketMessage::SocketMessage (const Message & msg)` `[inline]`

Here is the call graph for this function:



7.20.2.3 `SocketMessage::SocketMessage (const char * msg_s)` `[inline]`

Here is the call graph for this function:



7.20.2.4 SocketMessage::SocketMessage (std::string msg_s) [inline]

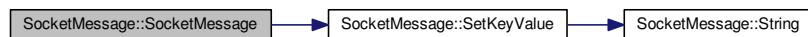
Here is the call graph for this function:



7.20.2.5 SocketMessage::SocketMessage (const MessageKey & key) [inline]

Construct a socket message out of a key.

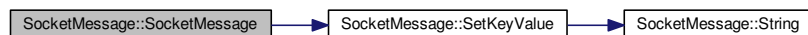
Here is the call graph for this function:



7.20.2.6 SocketMessage::SocketMessage (const MessageKey & key, const char * value) [inline]

Construct a socket message out of a key and a string-type value.

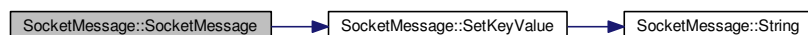
Here is the call graph for this function:



7.20.2.7 SocketMessage::SocketMessage (const MessageKey & key, std::string value) [inline]

Construct a socket message out of a key and a string-type value.

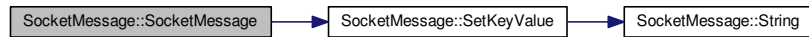
Here is the call graph for this function:



7.20.2.8 SocketMessage::SocketMessage (const MessageKey & key, const int value) [inline]

Construct a socket message out of a key and an integer-type value.

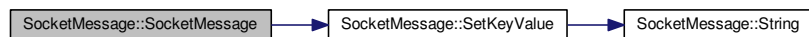
Here is the call graph for this function:



7.20.2.9 SocketMessage::SocketMessage (const MessageKey & key, const float value) [inline]

Construct a socket message out of a key and a float-type value.

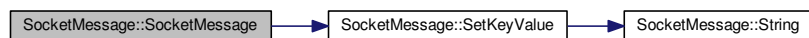
Here is the call graph for this function:



7.20.2.10 SocketMessage::SocketMessage (const MessageKey & key, const double value) [inline]

Construct a socket message out of a key and a double precision-type value.

Here is the call graph for this function:



7.20.2.11 SocketMessage::SocketMessage (MessageMap msg_m) [inline]

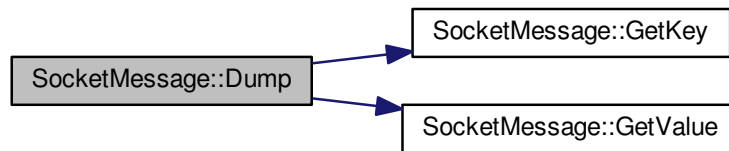
Construct a socket message out of a map of key/string-type value.

7.20.2.12 SocketMessage::~~SocketMessage () [inline]

7.20.3 Member Function Documentation

7.20.3.1 `void SocketMessage::Dump (std::ostream & os = std::cout) const [inline]`

Here is the call graph for this function:



7.20.3.2 `int SocketMessage::GetIntValue () const [inline]`

Extract the message's integer value.

7.20.3.3 `MessageKey SocketMessage::GetKey () const [inline]`

Extract the message's key.

7.20.3.4 `std::string SocketMessage::GetString () const [inline]`

Extract the whole key:value message.

7.20.3.5 `std::string SocketMessage::GetValue () const [inline]`

Extract the message's string value.

7.20.3.6 `VectorValue SocketMessage::GetVectorValue () const [inline]`

Extract the message's vector of string value.

Here is the call graph for this function:

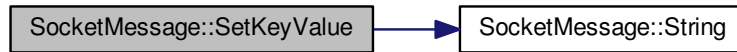


7.20.3.7 `MessageMap SocketMessage::Object () const [inline],[private]`

7.20.3.8 `void SocketMessage::SetKeyValue (const MessageKey & key, const char * value) [inline]`

String-valued message.

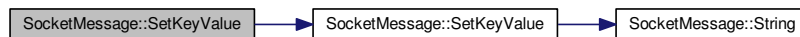
Here is the call graph for this function:



7.20.3.9 `void SocketMessage::SetKeyValue (const MessageKey & key, int int_value) [inline]`

Send an integer-valued message.

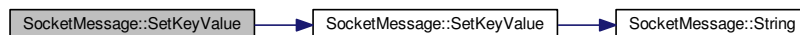
Here is the call graph for this function:



7.20.3.10 `void SocketMessage::SetKeyValue (const MessageKey & key, float float_value) [inline]`

Float-valued message.

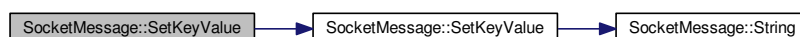
Here is the call graph for this function:



7.20.3.11 `void SocketMessage::SetKeyValue (const MessageKey & key, double double_value) [inline]`

Double-valued message.

Here is the call graph for this function:



7.20.3.12 `std::string SocketMessage::String () const` `[inline], [private]`

7.20.4 Field Documentation

7.20.4.1 `MessageMap SocketMessage::fMessage` `[private]`

The documentation for this class was generated from the following file:

- `include/SocketMessage.h`

7.21 VME::TDCErrorFlag Class Reference

Error flags handler.

```
#include <VME_TDCEvent.h>
```

Public Member Functions

- [TDCErrorFlag](#) (uint16_t ef)
- virtual [~TDCErrorFlag](#) ()
- uint16_t [GetWord](#) () const
- void [Dump](#) () const
- bool [HasReadoutFIFOOverflow](#) (unsigned int group_id) const
Check whether hits have been lost from read-out FIFO overflow in a given group.
- bool [HasL1BufferOverflow](#) (unsigned int group_id) const
Check whether hits have been lost from L1 buffer overflow in a given group.
- bool [HasGroupError](#) (unsigned int group_id) const
Check whether hits have been lost due to error in a given group.
- bool [HasReachedEventSizeLimit](#) () const
Hits rejected because of programmed event size limit.
- bool [HasTriggerFIFOOverflow](#) () const
Event lost (trigger FIFO overflow)
- bool [HasInternalChipError](#) () const
Internal fatal chip error has been detected.

Private Attributes

- uint16_t [fWord](#)

Friends

- `std::ostream & operator<< (std::ostream &os, const TDCErrorFlag &ef)`

7.21.1 Detailed Description

Error flags handler.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

22 Jun 2015

7.21.2 Constructor & Destructor Documentation

7.21.2.1 `VME::TDCErrorFlag::TDCErrorFlag (uint16_t ef)` [inline]

7.21.2.2 `virtual VME::TDCErrorFlag::~~TDCErrorFlag ()` [inline],[virtual]

7.21.3 Member Function Documentation

7.21.3.1 `void VME::TDCErrorFlag::Dump () const` [inline]

7.21.3.2 `uint16_t VME::TDCErrorFlag::GetWord () const` [inline]

7.21.3.3 `bool VME::TDCErrorFlag::HasGroupError (unsigned int group_id) const` [inline]

Check whether hits have been lost due to error in a given group.

7.21.3.4 `bool VME::TDCErrorFlag::HasInternalChipError () const` [inline]

Internal fatal chip error has been detected.

7.21.3.5 `bool VME::TDCErrorFlag::HasL1BufferOverflow (unsigned int group_id) const` [inline]

Check whether hits have been lost from L1 buffer overflow in a given group.

7.21.3.6 `bool VME::TDCErrorFlag::HasReachedEventSizeLimit () const` [inline]

Hits rejected because of programmed event size limit.

7.21.3.7 `bool VME::TDCErrorFlag::HasReadoutFIFOOverflow (unsigned int group_id) const` [inline]

Check whether hits have been lost from read-out FIFO overflow in a given group.

7.21.3.8 `bool VME::TDCErrorFlag::HasTriggerFIFOOverflow () const` [inline]

Event lost (trigger FIFO overflow)

7.21.4 Friends And Related Function Documentation

7.21.4.1 `std::ostream& operator<< (std::ostream & os, const TDCErrorFlag & ef)` [friend]

7.21.5 Field Documentation

7.21.5.1 `uint16_t VME::TDCErrorFlag::fWord` [private]

The documentation for this class was generated from the following file:

- include/VME_TDCEvent.h

7.22 VME::TDCEvent Class Reference

HPTDC event parser.

```
#include <VME_TDCEvent.h>
```

Public Types

- enum [EventType](#) {
[TDCMeasurement](#) = 0x0, [TDCHeader](#) = 0x1, [TDCTrailer](#) = 0x3, [TDCError](#) = 0x4,
[GlobalHeader](#) = 0x8, [GlobalTrailer](#) = 0x10, [ETTT](#) = 0x11, [Filler](#) = 0x18 }

Public Member Functions

- [TDCEvent](#) ()
- [TDCEvent](#) (const [TDCEvent](#) &ev)
- [TDCEvent](#) (const uint32_t &word)
- virtual [~TDCEvent](#) ()
- void [Dump](#) () const
- void [SetWord](#) (const uint32_t &word)
- uint32_t [GetWord](#) () const
- [EventType](#) [GetType](#) () const
Type of packet read out from the TDC.
- unsigned int [GetTDCId](#) () const
Programmed identifier of master TDC providing the event.
- uint16_t [GetEventId](#) () const
Event identifier from event counter.
- uint16_t [GetWordCount](#) () const
Total number of words in event (including headers and trailers)
- unsigned int [GetGeo](#) () const
- unsigned int [GetChannelId](#) () const
Channel number for.
- uint32_t [GetEventCount](#) () const
Total number of events.
- uint16_t [GetBunchId](#) () const
Bunch identifier of trigger (or trigger time tag)
- bool [IsTrailing](#) () const
Are we dealing with a trailing or a leading measurement?
- uint32_t [GetETTT](#) () const
Extended trigger time tag.
- uint32_t [GetTime](#) (bool pair=false) const
Edge measurement in programmed time resolution.
- unsigned int [GetWidth](#) () const
Width of pulse in programmed time resolution.
- unsigned int [GetStatus](#) () const
- [TDCErrorFlag](#) [GetErrorFlags](#) () const
Return error flags if an error condition has been detected.

Private Attributes

- uint32_t [fWord](#)

7.22.1 Detailed Description

HPTDC event parser.

Object enabling to decipher any measurement/error/debug event returned by the HPTDC chip

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

4 May 2015

7.22.2 Member Enumeration Documentation

7.22.2.1 enum VME::TDCEvent::EventType

Enumerator

TDCMeasurement

TDCHeader

TDCTrailer

TDCError

GlobalHeader

GlobalTrailer

ETTT

Filler

7.22.3 Constructor & Destructor Documentation

7.22.3.1 VME::TDCEvent::TDCEvent () [inline]

7.22.3.2 VME::TDCEvent::TDCEvent (const TDCEvent & ev) [inline]

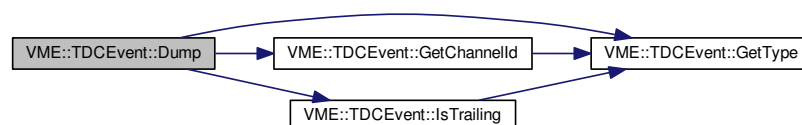
7.22.3.3 VME::TDCEvent::TDCEvent (const uint32_t & word) [inline]

7.22.3.4 virtual VME::TDCEvent::~~TDCEvent () [inline],[virtual]

7.22.4 Member Function Documentation

7.22.4.1 void VME::TDCEvent::Dump () const [inline]

Here is the call graph for this function:



7.22.4.2 uint16_t VME::TDCEvent::GetBunchId () const [inline]

Bunch identifier of trigger (or trigger time tag)

Here is the call graph for this function:



7.22.4.3 unsigned int VME::TDCEvent::GetChannelId () const [inline]

Channel number for.

Here is the call graph for this function:



7.22.4.4 TDCErrorFlag VME::TDCEvent::GetErrorFlags () const [inline]

Return error flags if an error condition has been detected.

Here is the call graph for this function:



7.22.4.5 uint32_t VME::TDCEvent::GetETTT () const [inline]

Extended trigger time tag.

Here is the call graph for this function:



7.22.4.6 `uint32_t VME::TDCEvent::GetEventCount () const [inline]`

Total number of events.

Here is the call graph for this function:



7.22.4.7 `uint16_t VME::TDCEvent::GetEventId () const [inline]`

Event identifier from event counter.

Here is the call graph for this function:



7.22.4.8 `unsigned int VME::TDCEvent::GetGeo () const [inline]`

Here is the call graph for this function:



7.22.4.9 unsigned int VME::TDCEvent::GetStatus () const [inline]

Here is the call graph for this function:



7.22.4.10 unsigned int VME::TDCEvent::GetTDCId () const [inline]

Programmed identifier of master TDC providing the event.

Here is the call graph for this function:

7.22.4.11 uint32_t VME::TDCEvent::GetTime (bool *pair* = false) const [inline]

Edge measurement in programmed time resolution.

Parameters

in	<i>pair</i>	Are we dealing with a pair measurement? (only for leading time word)
----	-------------	--

Here is the call graph for this function:



7.22.4.12 EventType VME::TDCEvent::GetType () const [inline]

Type of packet read out from the TDC.

7.22.4.13 `unsigned int VME::TDCEvent::GetWidth () const [inline]`

Width of pulse in programmed time resolution.

Here is the call graph for this function:



7.22.4.14 `uint32_t VME::TDCEvent::GetWord () const [inline]`

7.22.4.15 `uint16_t VME::TDCEvent::GetWordCount () const [inline]`

Total number of words in event (including headers and trailers)

Here is the call graph for this function:



7.22.4.16 `bool VME::TDCEvent::IsTrailing () const [inline]`

Are we dealing with a trailing or a leading measurement?

Here is the call graph for this function:



7.22.4.17 `void VME::TDCEvent::SetWord (const uint32_t & word) [inline]`

7.22.5 Field Documentation

7.22.5.1 `uint32_t VME::TDCEvent::fWord [private]`

The documentation for this class was generated from the following file:

- `include/VME_TDCEvent.h`

7.23 VME::TDCMeasurement Class Reference

```
#include <VME_TDCMeasurement.h>
```

Public Member Functions

- [TDCMeasurement](#) ()
- [TDCMeasurement](#) (const std::vector< [TDCEvent](#) > &v)
- [~TDCMeasurement](#) ()
- void [Dump](#) ()
- void [SetEventsCollection](#) (const std::vector< [TDCEvent](#) > &v)
- uint32_t [GetLeadingTime](#) (unsigned short event_id=0)
- uint32_t [GetTrailingTime](#) (unsigned short event_id=0)
- uint16_t [GetToT](#) (unsigned short event_id=0)
- uint16_t [GetChannelId](#) (unsigned short event_id=0)
- uint16_t [GetTDCId](#) ()
- uint16_t [GetEventId](#) ()
- uint16_t [GetBunchId](#) ()
- uint32_t [GetETTT](#) ()
- size_t [NumEvents](#) () const
- size_t [NumErrors](#) () const

Private Attributes

- std::map< [TDCEvent::EventType](#), [TDCEvent](#) > [fMap](#)
- std::vector< std::pair< [TDCEvent](#), [TDCEvent](#) > > [fEvents](#)

7.23.1 Detailed Description

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

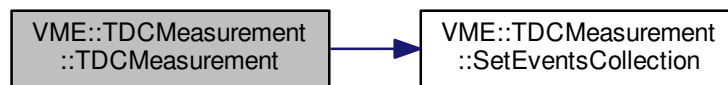
Jun 2015

7.23.2 Constructor & Destructor Documentation

7.23.2.1 VME::TDCMeasurement::TDCMeasurement () `[inline]`

7.23.2.2 `VME::TDCMeasurement::TDCMeasurement (const std::vector< TDCEvent > & v) [inline]`

Here is the call graph for this function:

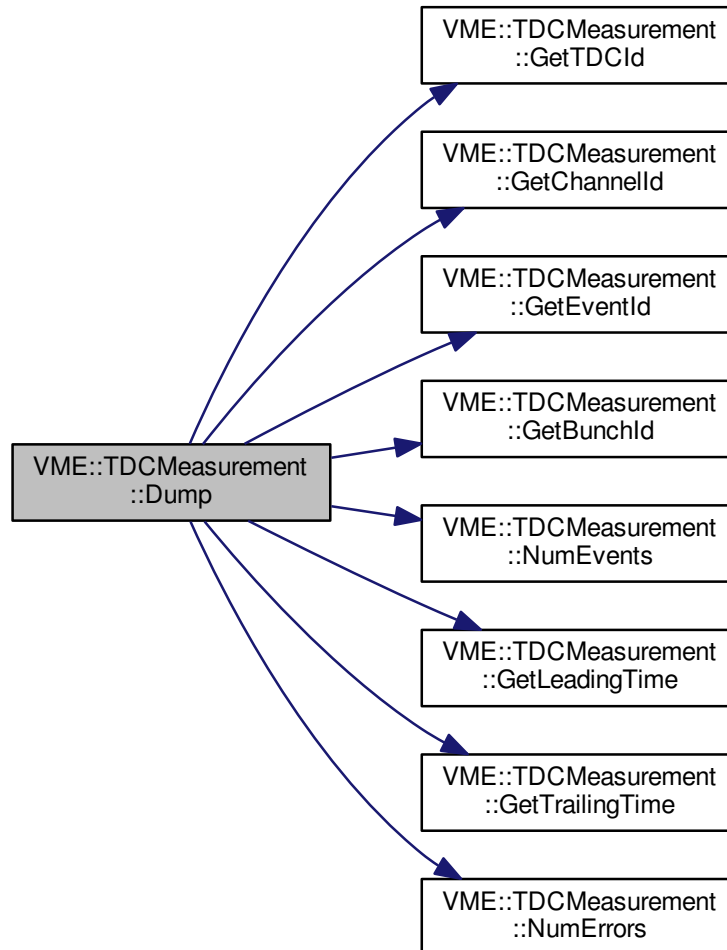


7.23.2.3 `VME::TDCMeasurement::~~TDCMeasurement () [inline]`

7.23.3 Member Function Documentation

7.23.3.1 void VME::TDCMeasurement::Dump () [inline]

Here is the call graph for this function:



7.23.3.2 uint16_t VME::TDCMeasurement::GetBunchId () [inline]

7.23.3.3 uint16_t VME::TDCMeasurement::GetChannelId (unsigned short *event_id* = 0) [inline]

7.23.3.4 uint32_t VME::TDCMeasurement::GetETTT () [inline]

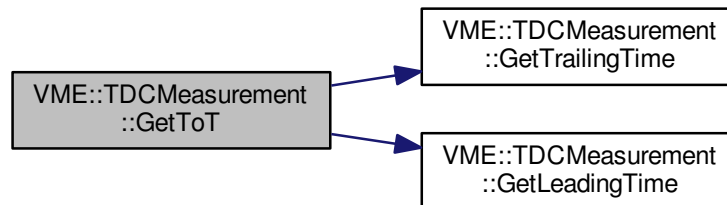
7.23.3.5 uint16_t VME::TDCMeasurement::GetEventId () [inline]

7.23.3.6 uint32_t VME::TDCMeasurement::GetLeadingTime (unsigned short *event_id* = 0) [inline]

7.23.3.7 uint16_t VME::TDCMeasurement::GetTDCId () [inline]

7.23.3.8 `uint16_t VME::TDCMeasurement::GetToT (unsigned short event_id = 0) [inline]`

Here is the call graph for this function:



7.23.3.9 `uint32_t VME::TDCMeasurement::GetTrailingTime (unsigned short event_id = 0) [inline]`

7.23.3.10 `size_t VME::TDCMeasurement::NumErrors () const [inline]`

7.23.3.11 `size_t VME::TDCMeasurement::NumEvents () const [inline]`

7.23.3.12 `void VME::TDCMeasurement::SetEventsCollection (const std::vector< TDCEvent > & v) [inline]`

7.23.4 Field Documentation

7.23.4.1 `std::vector< std::pair<TDCEvent,TDCEvent> > VME::TDCMeasurement::fEvents [private]`

7.23.4.2 `std::map<TDCEvent::EventType,TDCEvent> VME::TDCMeasurement::fMap [private]`

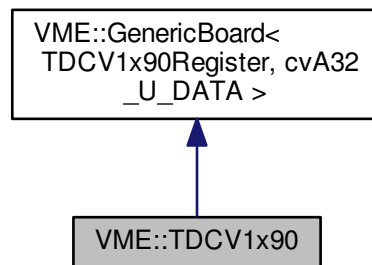
The documentation for this class was generated from the following file:

- `include/VME_TDCMeasurement.h`

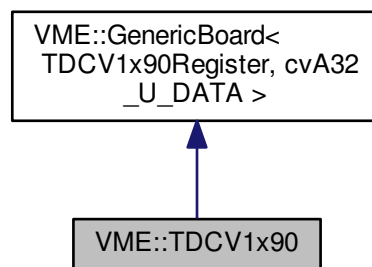
7.24 VME::TDCV1x90 Class Reference

```
#include <VME_TDCV1x90.h>
```


Inheritance diagram for VME::TDCV1x90:



Collaboration diagram for VME::TDCV1x90:



Public Types

- enum `DLLMode` { `DLL_Direct_LowRes` = 0x0, `DLL_PLL_LowRes` = 0x1, `DLL_PLL_MedRes` = 0x2, `DLL_PLL_HighRes` = 0x3 }

Public Member Functions

- `TDCV1x90` (int32_t bhandle, uint32_t baseaddr)
- `~TDCV1x90` ()
- void `SetVerboseLevel` (unsigned short verb=1)
- void `SetTestMode` (bool en=true) const
- bool `GetTestMode` () const
- uint32_t `GetModel` () const
- uint32_t `GetOUI` () const
- uint32_t `GetSerialNumber` () const
- uint16_t `GetFirmwareRevision` () const
- void `CheckConfiguration` () const
- void `EnableChannel` (short) const

- void [DisableChannel](#) (short) const
- void [SetPol](#) (uint16_t word1, uint16_t word2) const
- std::map< unsigned short, bool > [GetPol](#) () const
- void [SetLSBTrailEdge](#) ([trailEdge_lsb](#)) const
- void [SetAcquisitionMode](#) (const [AcquisitionMode](#) &)
- [AcquisitionMode](#) [GetAcquisitionMode](#) ()
- void [SetTriggerMatching](#) ()
- void [SetContinuousStorage](#) ()
- void [SetDetectionMode](#) (const [DetectionMode](#) &detm)
- [DetectionMode](#) [GetDetectionMode](#) ()
- void [SetDLLClock](#) (const [DLLMode](#) &dll) const
- [DLLMode](#) [GetDLLClock](#) () const
- void [SetGlobalOffset](#) (const [GlobalOffset](#) &) const
- [GlobalOffset](#) [GetGlobalOffset](#) () const
- void [SetRCAdjust](#) (int, uint16_t) const
- uint16_t [GetRCAdjust](#) (int) const
- uint32_t [GetEventCounter](#) () const
- *Number of occurred triggers.*
- uint16_t [GetEventStored](#) () const
- *Number of events currently stored in the output buffer.*
- void [SetTDCEncapsulation](#) (bool) const
- bool [GetTDCEncapsulation](#) () const
- void [SetErrorMarks](#) (bool mode=true)
- bool [GetErrorMarks](#) () const
- void [SetPairModeResolution](#) (int, int) const
- uint16_t [GetResolution](#) () const
- void [SetBLTEventNumberRegister](#) (const uint16_t &) const
- uint16_t [GetBLTEventNumberRegister](#) () const
- void [SetWindowWidth](#) (const uint16_t &)
- uint16_t [GetWindowWidth](#) () const
- void [SetWindowOffset](#) (const int16_t &) const
- int16_t [GetWindowOffset](#) () const
- uint16_t [GetTriggerConfiguration](#) (const [trig_conf](#) &) const
- bool [SoftwareClear](#) () const
- bool [SoftwareReset](#) () const
- bool [HardwareReset](#) () const
- void [SetETTT](#) (bool ettt=true) const
- bool [GetETTT](#) () const
- void [SetStatus](#) (const [TDCV1x90Status](#) &) const
- [TDCV1x90Status](#) [GetStatus](#) () const
- void [SetControl](#) (const [TDCV1x90Control](#) &) const
- [TDCV1x90Control](#) [GetControl](#) () const
- [TDCEventCollection](#) [FetchEvents](#) ()
- void [SetChannelDeadTime](#) (unsigned short dt) const
- unsigned short [GetChannelDeadTime](#) () const
- void [SetFIFOSize](#) (const uint16_t &) const
- uint16_t [GetFIFOSize](#) () const
- void [abort](#) ()

Private Member Functions

- bool [WaitMicro](#) (const [micro_handshake](#) &mode) const
- void [ReadAcquisitionMode](#) ()
- void [ReadDetectionMode](#) ()

Private Attributes

- unsigned short `fVerb`
- `AcquisitionMode` `fAcquisitionMode`
- `DetectionMode` `fDetectionMode`
- bool `fErrorMarks`
- `uint16_t` `fWindowWidth`
- `uint32_t` * `fBuffer`
- `uint32_t` `nchannels`
- bool `gEnd`
- `std::string` `pair_lead_res` [8]
- `std::string` `pair_width_res` [16]

Additional Inherited Members

7.24.1 Detailed Description

Author

Laurent Forthomme laurent.forthomme@cern.ch
Bob Velghe bob.velghe@cern.ch

Date

Jun 2010 (NA62-Gigatracker)
May 2015 (CMS-TOTEM PPS)

7.24.2 Member Enumeration Documentation

7.24.2.1 enum VME::TDCV1x90::DLLMode

Enumerator

DLL_Direct_LowRes

DLL_PLL_LowRes

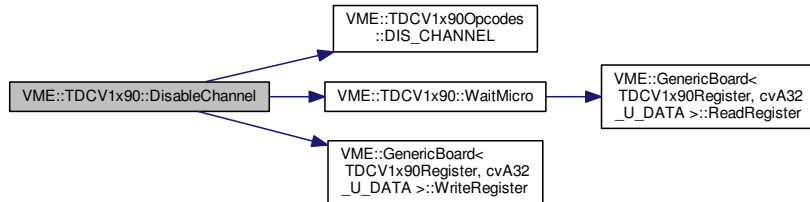
DLL_PLL_MedRes

DLL_PLL_HighRes

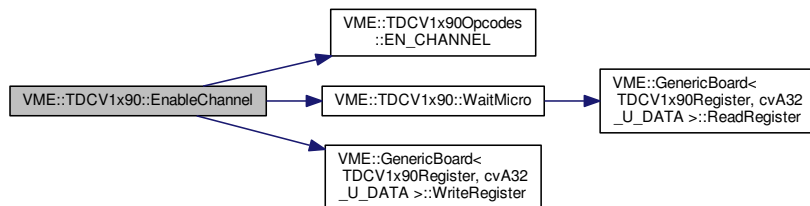
7.24.3 Constructor & Destructor Documentation

7.24.4.3 void VME::TDCV1x90::DisableChannel (short *channel_id*) const

Here is the call graph for this function:

7.24.4.4 void VME::TDCV1x90::EnableChannel (short *channel_id*) const

Here is the call graph for this function:



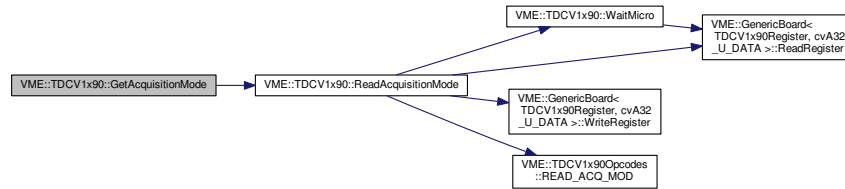
7.24.4.5 TDCEventCollection VME::TDCV1x90::FetchEvents ()

Here is the call graph for this function:



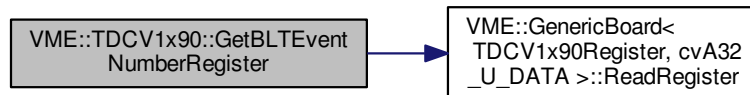
7.24.4.6 AcquisitionMode VME::TDCV1x90::GetAcquisitionMode () [inline]

Here is the call graph for this function:



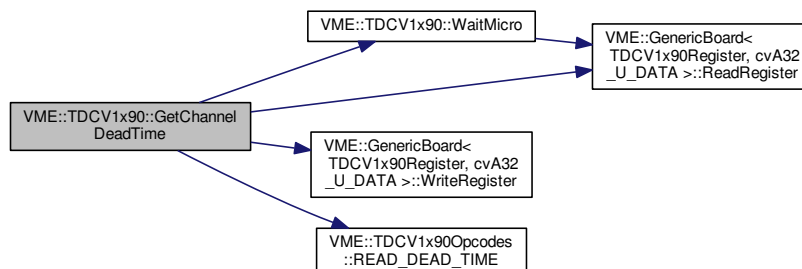
7.24.4.7 uint16_t VME::TDCV1x90::GetBLTEventNumberRegister () const

Here is the call graph for this function:



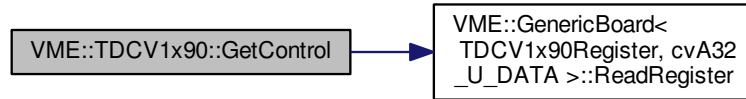
7.24.4.8 unsigned short VME::TDCV1x90::GetChannelDeadTime () const

Here is the call graph for this function:



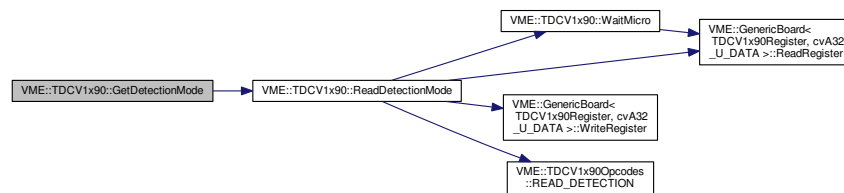
7.24.4.9 TDCV1x90Control VME::TDCV1x90::GetControl () const

Here is the call graph for this function:



7.24.4.10 DetectionMode VME::TDCV1x90::GetDetectionMode () [inline]

Here is the call graph for this function:

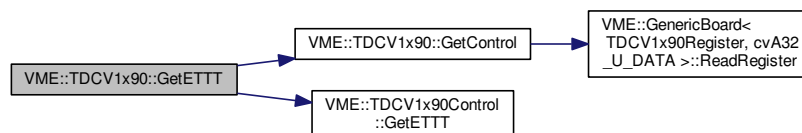


7.24.4.11 DLLMode VME::TDCV1x90::GetDLLClock () const

7.24.4.12 bool VME::TDCV1x90::GetErrorMarks () const [inline]

7.24.4.13 bool VME::TDCV1x90::GetETTT () const [inline]

Here is the call graph for this function:

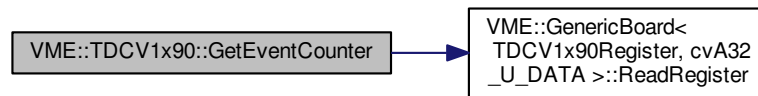


7.24.4.14 uint32_t VME::TDCV1x90::GetEventCounter () const

Number of occurred triggers.

Number of acquired events since the latest module's reset/clear; this counter works in trigger Matching Mode only.

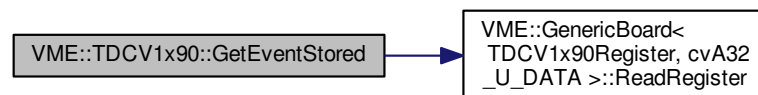
Here is the call graph for this function:



7.24.4.15 `uint16_t VME::TDCV1x90::GetEventStored () const`

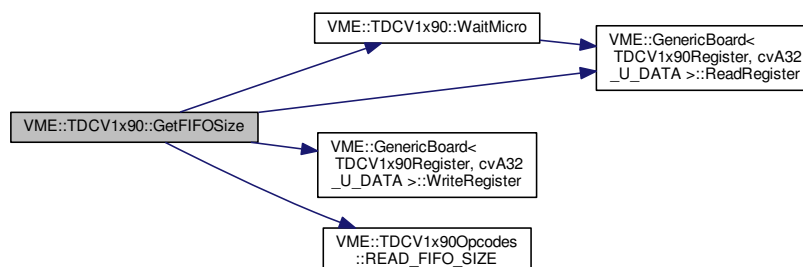
Number of events currently stored in the output buffer.

Here is the call graph for this function:



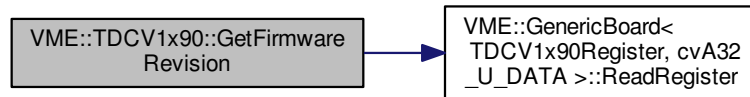
7.24.4.16 `uint16_t VME::TDCV1x90::GetFIFOSize () const`

Here is the call graph for this function:



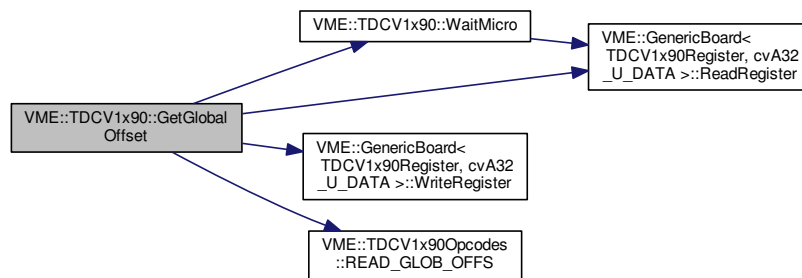
7.24.4.17 uint16_t VME::TDCV1x90::GetFirmwareRevision () const

Here is the call graph for this function:



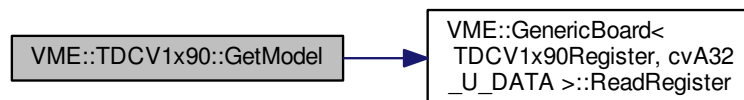
7.24.4.18 GlobalOffset VME::TDCV1x90::GetGlobalOffset () const

Here is the call graph for this function:



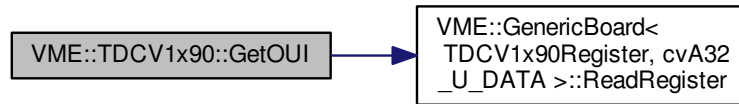
7.24.4.19 uint32_t VME::TDCV1x90::GetModel () const

Here is the call graph for this function:



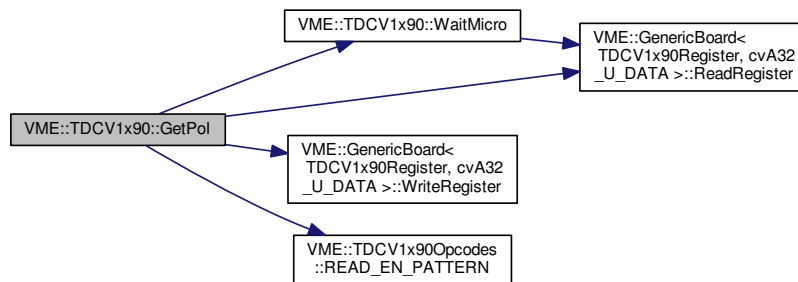
7.24.4.20 `uint32_t VME::TDCV1x90::GetOUI () const`

Here is the call graph for this function:



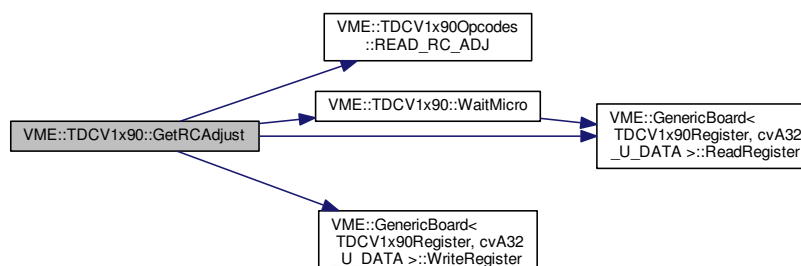
7.24.4.21 `std::map< unsigned short, bool > VME::TDCV1x90::GetPol () const`

Here is the call graph for this function:



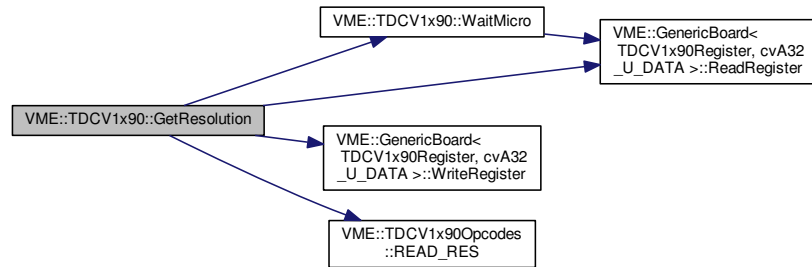
7.24.4.22 `uint16_t VME::TDCV1x90::GetRCAdjust (int tdc) const`

Here is the call graph for this function:



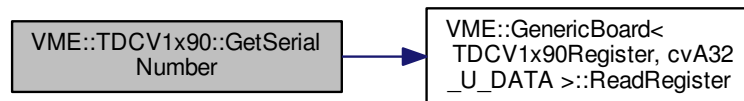
7.24.4.23 uint16_t VME::TDCV1x90::GetResolution () const

Here is the call graph for this function:



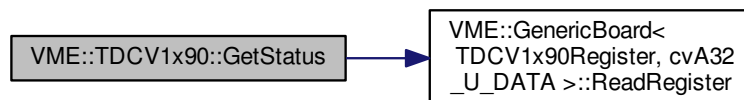
7.24.4.24 uint32_t VME::TDCV1x90::GetSerialNumber () const

Here is the call graph for this function:



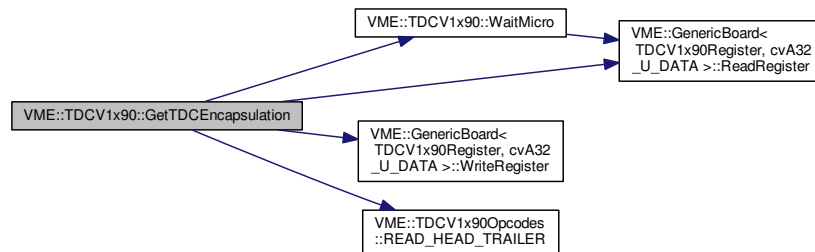
7.24.4.25 TDCV1x90Status VME::TDCV1x90::GetStatus () const

Here is the call graph for this function:



7.24.4.26 `bool VME::TDCV1x90::GetTDCEncapsulation () const`

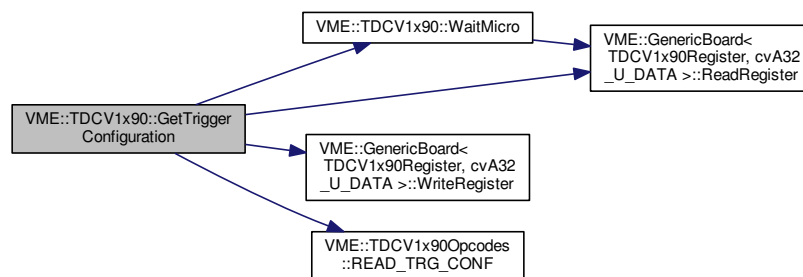
Here is the call graph for this function:



7.24.4.27 `bool VME::TDCV1x90::GetTestMode () const`

7.24.4.28 `uint16_t VME::TDCV1x90::GetTriggerConfiguration (const trig_conf & type) const`

Here is the call graph for this function:



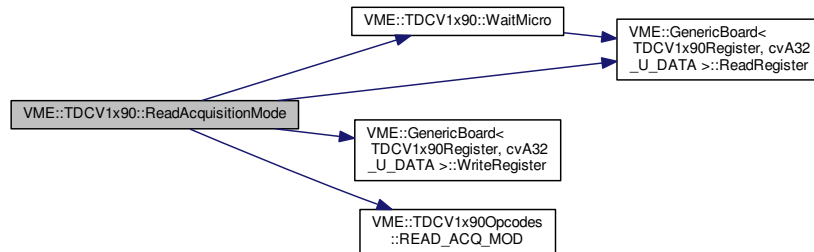
7.24.4.29 `int16_t VME::TDCV1x90::GetWindowOffset () const`

7.24.4.30 `uint16_t VME::TDCV1x90::GetWindowWidth () const` `[inline]`

7.24.4.31 `bool VME::TDCV1x90::HardwareReset () const`

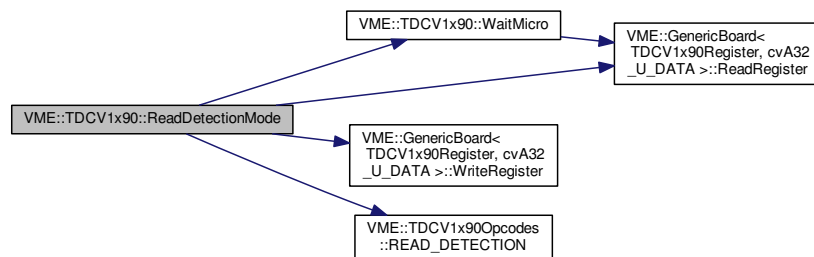
7.24.4.32 void VME::TDCV1x90::ReadAcquisitionMode () [private]

Here is the call graph for this function:



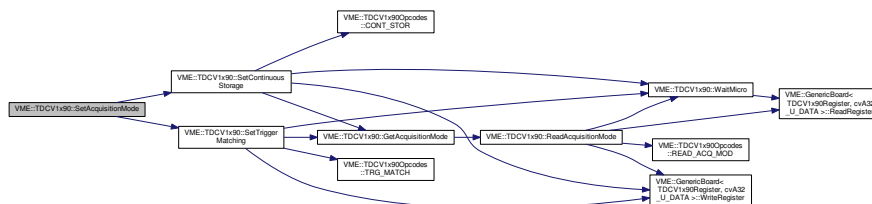
7.24.4.33 void VME::TDCV1x90::ReadDetectionMode () [private]

Here is the call graph for this function:



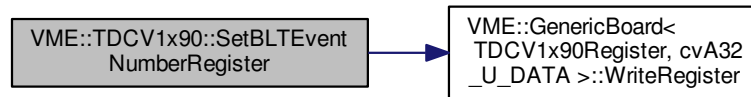
7.24.4.34 void VME::TDCV1x90::SetAcquisitionMode (const AcquisitionMode & mode)

Here is the call graph for this function:



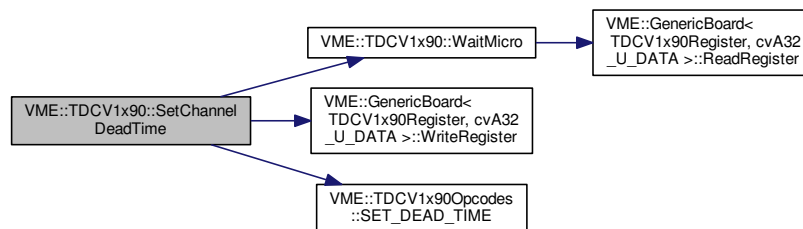
7.24.4.35 void VME::TDCV1x90::SetBLTEventNumberRegister (const uint16_t & value) const

Here is the call graph for this function:



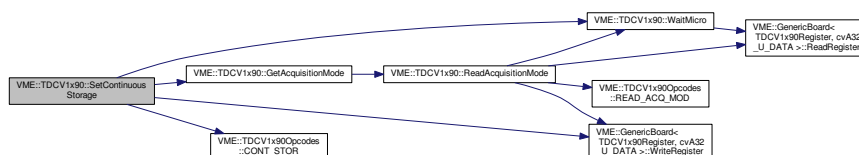
7.24.4.36 void VME::TDCV1x90::SetChannelDeadTime (unsigned short dt) const

Here is the call graph for this function:



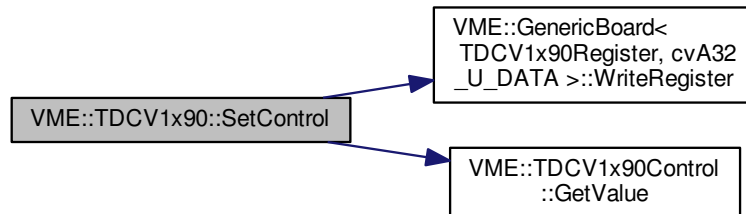
7.24.4.37 void VME::TDCV1x90::SetContinuousStorage ()

Here is the call graph for this function:



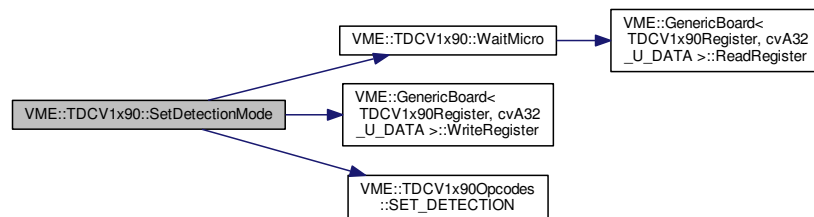
7.24.4.38 void VME::TDCV1x90::SetControl (const TDCV1x90Control & control) const

Here is the call graph for this function:



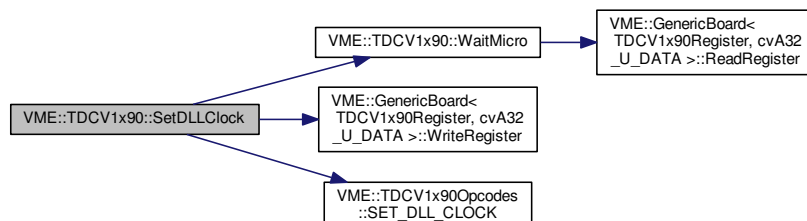
7.24.4.39 void VME::TDCV1x90::SetDetectionMode (const DetectionMode & detm)

Here is the call graph for this function:



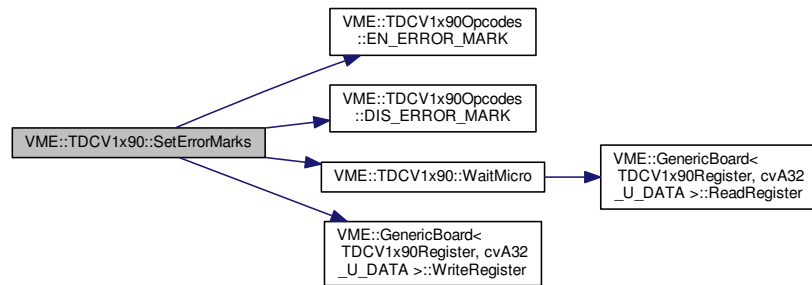
7.24.4.40 void VME::TDCV1x90::SetDLLClock (const DLLMode & dll) const

Here is the call graph for this function:



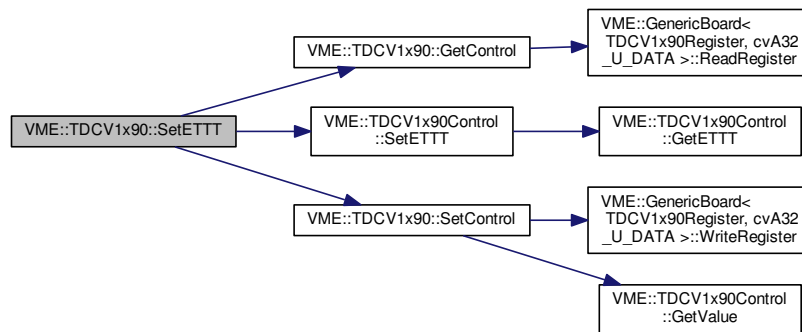
7.24.4.41 void VME::TDCV1x90::SetErrorMarks (bool *mode* = true)

Here is the call graph for this function:



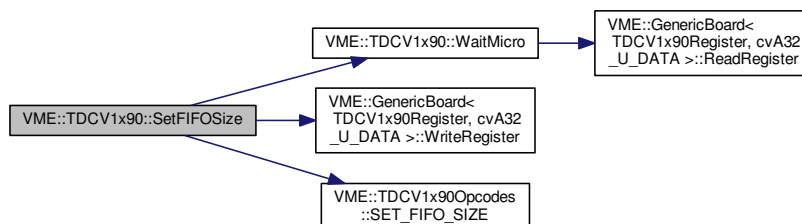
7.24.4.42 void VME::TDCV1x90::SetETTT (bool *ettt* = true) const [inline]

Here is the call graph for this function:



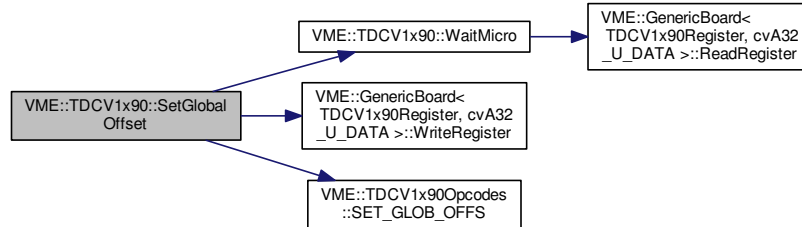
7.24.4.43 void VME::TDCV1x90::SetFIFOSize (const uint16_t & *size*) const

Here is the call graph for this function:



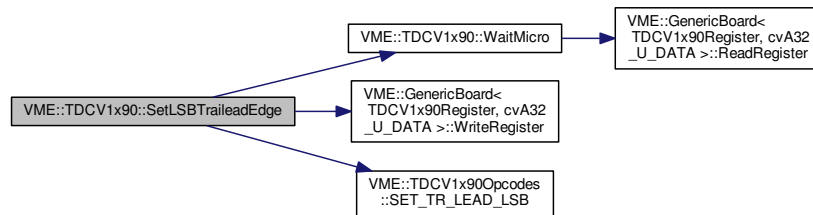
7.24.4.44 void VME::TDCV1x90::SetGlobalOffset (const GlobalOffset & offs) const

Here is the call graph for this function:



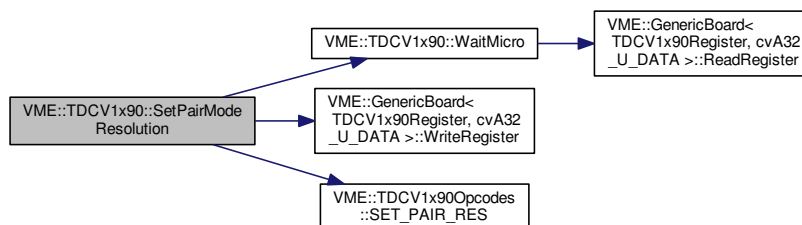
7.24.4.45 void VME::TDCV1x90::SetLSBTraileadEdge (trailead_edge_lsb conf) const

Here is the call graph for this function:



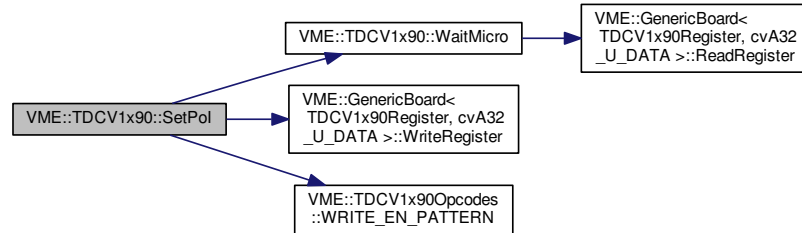
7.24.4.46 void VME::TDCV1x90::SetPairModeResolution (int lead_time_res, int pulse_width_res) const

Here is the call graph for this function:



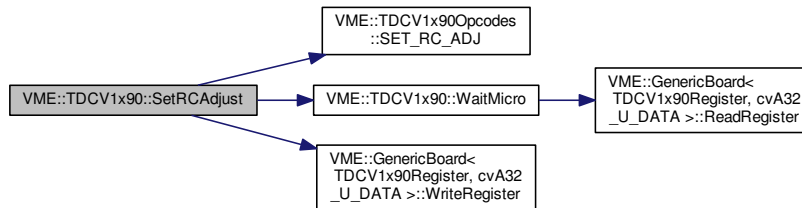
7.24.4.47 void VME::TDCV1x90::SetPol (uint16_t word1, uint16_t word2) const

Here is the call graph for this function:



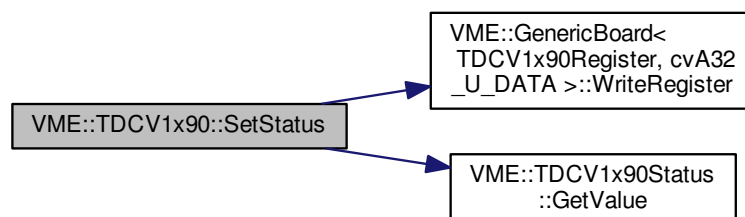
7.24.4.48 void VME::TDCV1x90::SetRCAdjust (int tdc, uint16_t value) const

Here is the call graph for this function:



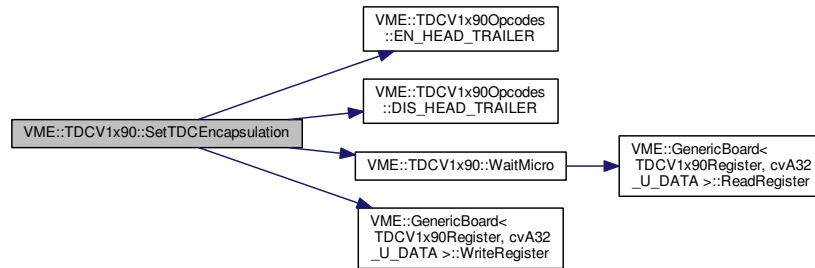
7.24.4.49 void VME::TDCV1x90::SetStatus (const TDCV1x90Status & status) const

Here is the call graph for this function:

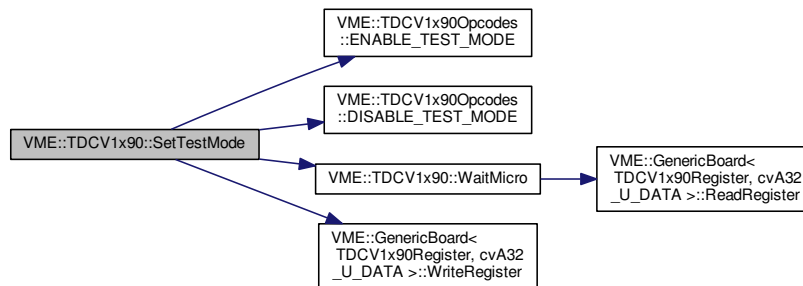


7.24.4.50 void VME::TDCV1x90::SetTDCEncapsulation (bool *mode*) const

Here is the call graph for this function:

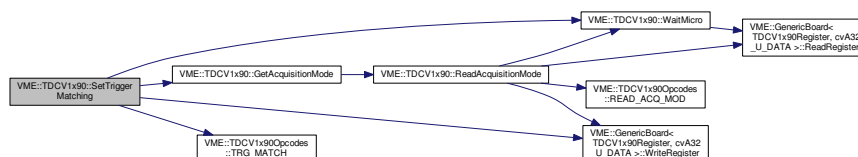
7.24.4.51 void VME::TDCV1x90::SetTestMode (bool *en = true*) const

Here is the call graph for this function:



7.24.4.52 void VME::TDCV1x90::SetTriggerMatching ()

Here is the call graph for this function:

7.24.4.53 void VME::TDCV1x90::SetVerboseLevel (unsigned short *verb = 1*) [inline]

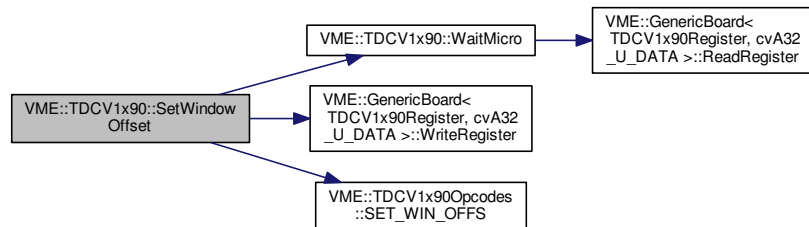
7.24.4.54 `void VME::TDCV1x90::SetWindowOffset (const int16_t & offs) const`

Set the offset of the match window with respect to the trigger itself, i.e. the time difference (expressed in clock cycles) between the start of the match window and the trigger time

Parameters

in	Window	offset, in units of clock cycles
----	--------	----------------------------------

Here is the call graph for this function:



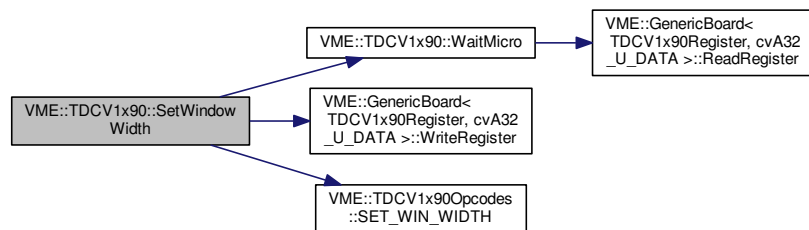
7.24.4.55 void VME::TDCV1x90::SetWindowWidth (const uint16_t & width)

Set the width of the match window (in number of clock cycles)

Parameters

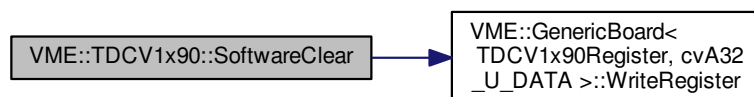
in	Window	width, in units of clock cycles
----	--------	---------------------------------

Here is the call graph for this function:



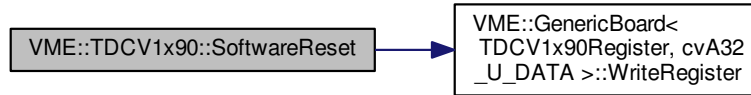
7.24.4.56 bool VME::TDCV1x90::SoftwareClear () const

Here is the call graph for this function:



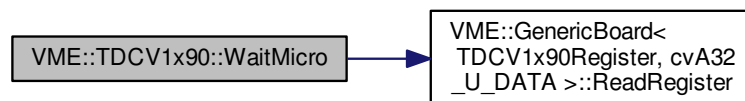
7.24.4.57 `bool VME::TDCV1x90::SoftwareReset () const`

Here is the call graph for this function:



7.24.4.58 `bool VME::TDCV1x90::WaitMicro (const micro_handshake & mode) const` [private]

Here is the call graph for this function:



7.24.5 Field Documentation

7.24.5.1 `AcquisitionMode VME::TDCV1x90::fAcquisitionMode` [private]

7.24.5.2 `uint32_t* VME::TDCV1x90::fBuffer` [private]

7.24.5.3 `DetectionMode VME::TDCV1x90::fDetectionMode` [private]

7.24.5.4 `bool VME::TDCV1x90::fErrorMarks` [private]

7.24.5.5 `unsigned short VME::TDCV1x90::fVerb` [private]

7.24.5.6 `uint16_t VME::TDCV1x90::fWindowWidth` [private]

7.24.5.7 `bool VME::TDCV1x90::gEnd` [private]

7.24.5.8 `uint32_t VME::TDCV1x90::nchannels` [private]

7.24.5.9 `std::string VME::TDCV1x90::pair_lead_res[8]` [private]

7.24.5.10 `std::string VME::TDCV1x90::pair_width_res[16]` [private]

The documentation for this class was generated from the following files:

- `include/VME_TDCV1x90.h`
- `src/VME_TDCV1x90.cpp`

7.25 VME::TDCV1x90Control Class Reference

TDC control register.

```
#include <VME_TDCV1x90.h>
```

Public Member Functions

- [TDCV1x90Control](#) (const uint16_t &word)
- virtual [~TDCV1x90Control](#) ()
- void [Dump](#) () const
- uint16_t [GetValue](#) () const
- bool [GetBusError](#) () const
- void [SetBusError](#) (bool sw)
- bool [GetTermination](#) () const
- void [SetTermination](#) (bool sw)
- bool [GetSWTermination](#) () const
- void [SetSWTermination](#) (bool sw)
- bool [GetEmptyEvent](#) () const
- void [SetEmptyEvent](#) (bool sw)
- bool [GetAlign64](#) () const
- void [SetAlign64](#) (bool sw)
- bool [GetCompensation](#) () const
- void [SetCompensation](#) (bool sw)
- bool [GetTestFIFO](#) () const
- void [SetTestFIFO](#) (bool sw)
- bool [GetSRAMCompensation](#) () const
- void [SetSRAMCompensation](#) (bool sw)
- bool [GetEventFIFO](#) () const
- void [SetEventFIFO](#) (bool sw)
- bool [GetETTT](#) () const
- void [SetETTT](#) (bool sw)
- bool [GetMEBAccess](#) () const
- void [SetMEBAccess](#) (bool sw)

Private Attributes

- uint16_t [fWord](#)

7.25.1 Detailed Description

TDC control register.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

Jun 2015

7.25.2 Constructor & Destructor Documentation

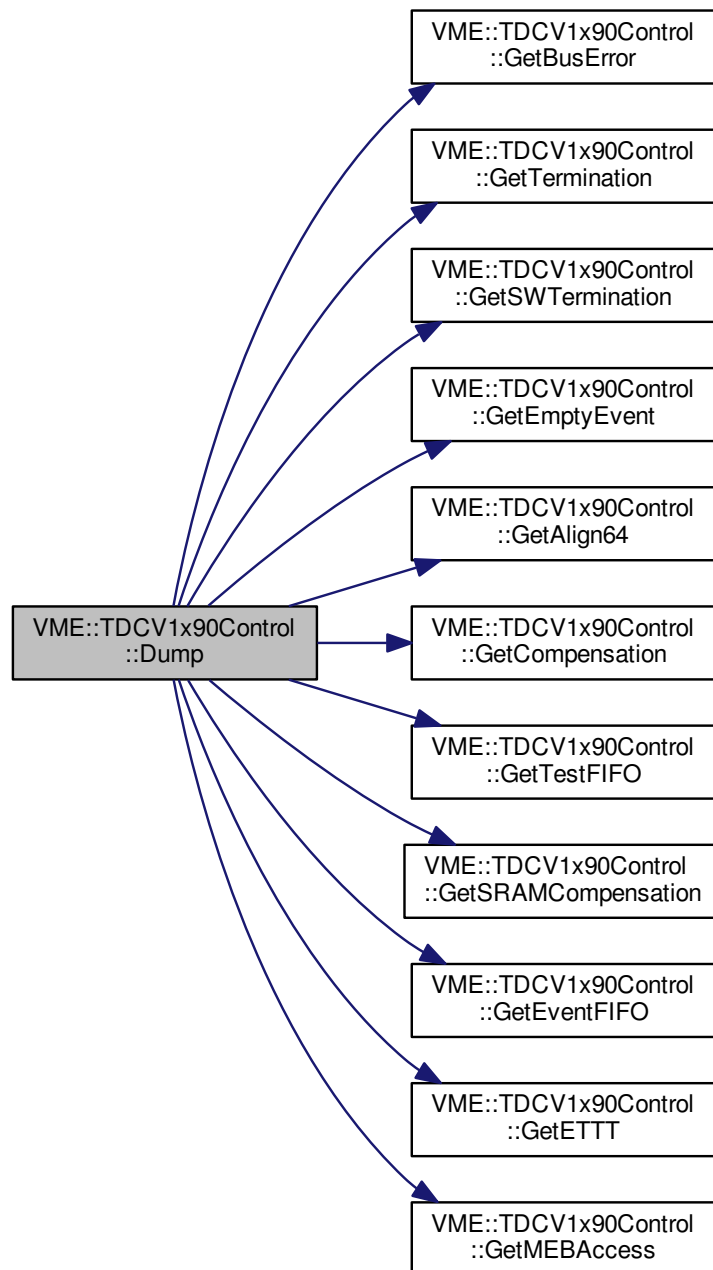
7.25.2.1 VME::TDCV1x90Control::TDCV1x90Control (const uint16_t & *word*) [inline]

7.25.2.2 virtual VME::TDCV1x90Control::~~TDCV1x90Control () [inline],[virtual]

7.25.3 Member Function Documentation

7.25.3.1 void VME::TDCV1x90Control::Dump () const [inline]

Here is the call graph for this function:



7.25.3.2 bool VME::TDCV1x90Control::GetAlign64 () const [inline]

7.25.3.3 bool VME::TDCV1x90Control::GetBusError () const [inline]

7.25.3.4 `bool VME::TDCV1x90Control::GetCompensation () const [inline]`

7.25.3.5 `bool VME::TDCV1x90Control::GetEmptyEvent () const [inline]`

7.25.3.6 `bool VME::TDCV1x90Control::GetETTT () const [inline]`

7.25.3.7 `bool VME::TDCV1x90Control::GetEventFIFO () const [inline]`

7.25.3.8 `bool VME::TDCV1x90Control::GetMEBAccess () const [inline]`

7.25.3.9 `bool VME::TDCV1x90Control::GetSRAMCompensation () const [inline]`

7.25.3.10 `bool VME::TDCV1x90Control::GetSWTermination () const [inline]`

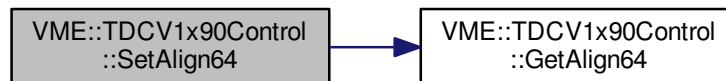
7.25.3.11 `bool VME::TDCV1x90Control::GetTermination () const [inline]`

7.25.3.12 `bool VME::TDCV1x90Control::GetTestFIFO () const [inline]`

7.25.3.13 `uint16_t VME::TDCV1x90Control::GetValue () const [inline]`

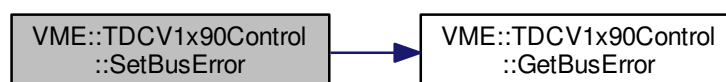
7.25.3.14 `void VME::TDCV1x90Control::SetAlign64 (bool sw) [inline]`

Here is the call graph for this function:



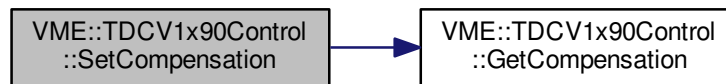
7.25.3.15 `void VME::TDCV1x90Control::SetBusError (bool sw) [inline]`

Here is the call graph for this function:



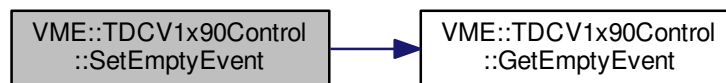
7.25.3.16 void VME::TDCV1x90Control::SetCompensation (bool *sw*) [inline]

Here is the call graph for this function:



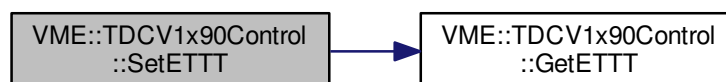
7.25.3.17 void VME::TDCV1x90Control::SetEmptyEvent (bool *sw*) [inline]

Here is the call graph for this function:



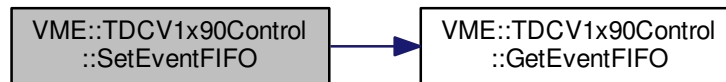
7.25.3.18 void VME::TDCV1x90Control::SetETTT (bool *sw*) [inline]

Here is the call graph for this function:



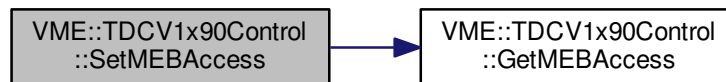
7.25.3.19 void VME::TDCV1x90Control::SetEventFIFO (bool *sw*) [inline]

Here is the call graph for this function:



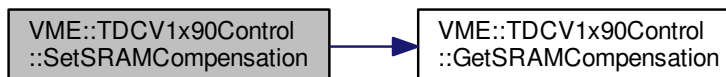
7.25.3.20 void VME::TDCV1x90Control::SetMEBAccess (bool *sw*) [inline]

Here is the call graph for this function:



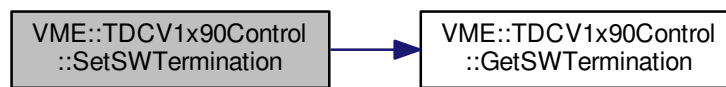
7.25.3.21 void VME::TDCV1x90Control::SetSRAMCompensation (bool *sw*) [inline]

Here is the call graph for this function:



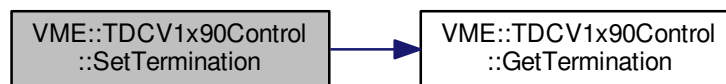
7.25.3.22 `void VME::TDCV1x90Control::SetSWTermination (bool sw) [inline]`

Here is the call graph for this function:



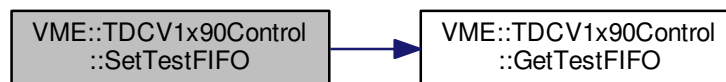
7.25.3.23 `void VME::TDCV1x90Control::SetTermination (bool sw) [inline]`

Here is the call graph for this function:



7.25.3.24 `void VME::TDCV1x90Control::SetTestFIFO (bool sw) [inline]`

Here is the call graph for this function:



7.25.4 Field Documentation

7.25.4.1 `uint16_t VME::TDCV1x90Control::fWord [private]`

The documentation for this class was generated from the following file:

- `include/VME_TDCV1x90.h`

7.26 VME::TDCV1x90Status Class Reference

TDC status register.

```
#include <VME_TDCV1x90.h>
```

Public Types

- enum [TDCResolution](#) { [R_800ps](#) = 0x0, [R_200ps](#) = 0x1, [R_100ps](#) = 0x2, [R_25ps](#) = 0x3 }

Public Member Functions

- [TDCV1x90Status](#) (const uint16_t &word)
- virtual [~TDCV1x90Status](#) ()
- void [Dump](#) () const
- uint16_t [GetValue](#) () const
- bool [DataReady](#) () const
- bool [AlmostFull](#) () const
- bool [Full](#) () const
- bool [TriggerMatching](#) () const
- bool [HeadersEnabled](#) () const
- bool [TerminationOn](#) () const
- bool [Error](#) (const unsigned int &id) const
- bool [Error](#) () const
- bool [BusError](#) () const
- bool [Purged](#) () const
- [TDCResolution Resolution](#) () const
- bool [PairMode](#) () const
- bool [TriggerLost](#) () const

Private Attributes

- uint16_t [fWord](#)

7.26.1 Detailed Description

TDC status register.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

Jun 2015

7.26.2 Member Enumeration Documentation

7.26.2.1 enum VME::TDCV1x90Status::TDCResolution

Enumerator

R_800ps

R_200ps

R_100ps

R_25ps

7.26.3 Constructor & Destructor Documentation

7.26.3.1 VME::TDCV1x90Status::TDCV1x90Status (const uint16_t & *word*) [inline]

7.26.3.2 virtual VME::TDCV1x90Status::~~TDCV1x90Status () [inline],[virtual]

7.26.4 Member Function Documentation

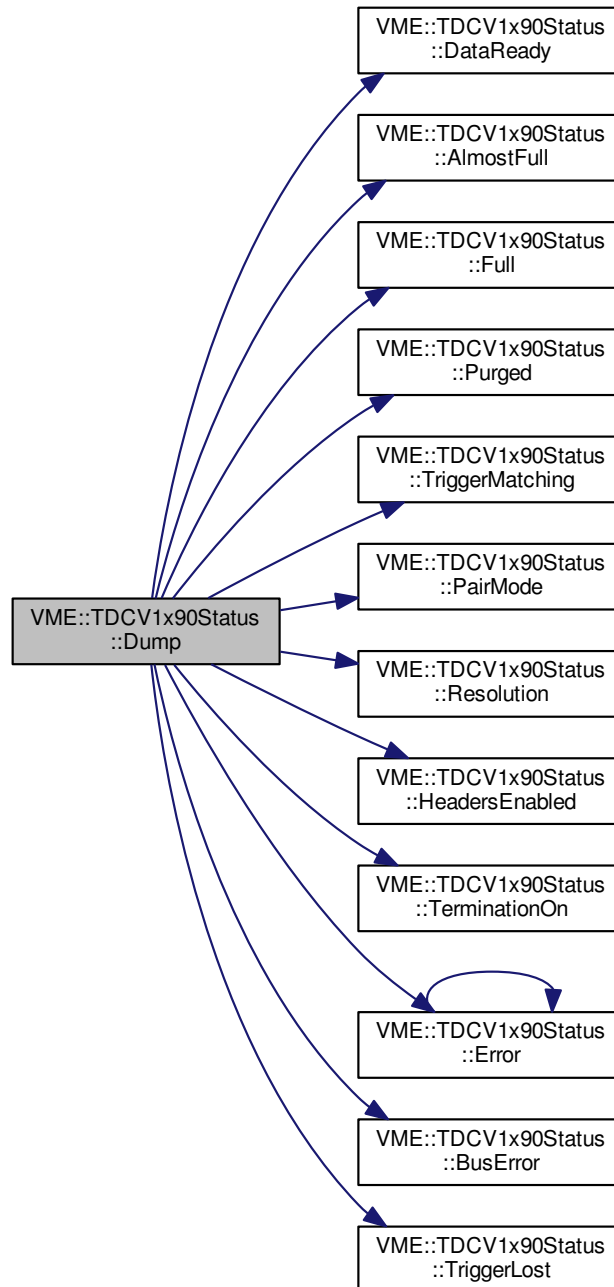
7.26.4.1 bool VME::TDCV1x90Status::AlmostFull () const [inline]

7.26.4.2 bool VME::TDCV1x90Status::BusError () const [inline]

7.26.4.3 bool VME::TDCV1x90Status::DataReady () const [inline]

7.26.4.4 `void VME::TDCV1x90Status::Dump () const [inline]`

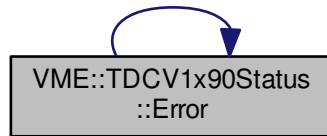
Here is the call graph for this function:



7.26.4.5 `bool VME::TDCV1x90Status::Error (const unsigned int & id) const [inline]`

7.26.4.6 `bool VME::TDCV1x90Status::Error () const [inline]`

Here is the call graph for this function:



7.26.4.7 `bool VME::TDCV1x90Status::Full () const [inline]`

7.26.4.8 `uint16_t VME::TDCV1x90Status::GetValue () const [inline]`

7.26.4.9 `bool VME::TDCV1x90Status::HeadersEnabled () const [inline]`

7.26.4.10 `bool VME::TDCV1x90Status::PairMode () const [inline]`

7.26.4.11 `bool VME::TDCV1x90Status::Purged () const [inline]`

7.26.4.12 `TDCResolution VME::TDCV1x90Status::Resolution () const [inline]`

7.26.4.13 `bool VME::TDCV1x90Status::TerminationOn () const [inline]`

7.26.4.14 `bool VME::TDCV1x90Status::TriggerLost () const [inline]`

7.26.4.15 `bool VME::TDCV1x90Status::TriggerMatching () const [inline]`

7.26.5 Field Documentation

7.26.5.1 `uint16_t VME::TDCV1x90Status::fWord [private]`

The documentation for this class was generated from the following file:

- `include/VME_TDCV1x90.h`

7.27 VME::trailead_t Struct Reference

```
#include <VME_TDCV1x90.h>
```

Data Fields

- `uint32_t event_count`
- `int total_hits [16]`
- `std::multimap< int32_t, int32_t > leading`
- `std::multimap< int32_t, int32_t > trailing`
- `uint32_t ettt`

7.27.1 Field Documentation

7.27.1.1 `uint32_t VME::trailead_t::ettt`

7.27.1.2 `uint32_t VME::trailead_t::event_count`

7.27.1.3 `std::multimap<int32_t,int32_t> VME::trailead_t::leading`

7.27.1.4 `int VME::trailead_t::total_hits[16]`

7.27.1.5 `std::multimap<int32_t,int32_t> VME::trailead_t::trailing`

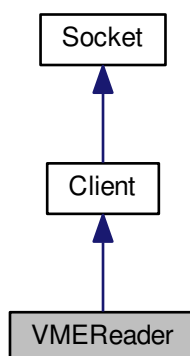
The documentation for this struct was generated from the following file:

- `include/VME_TDCV1x90.h`

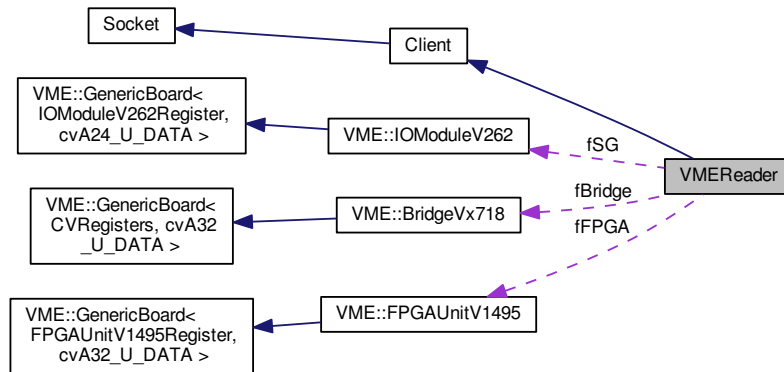
7.28 VMEReader Class Reference

```
#include <VMEReader.h>
```

Inheritance diagram for VMEReader:



Collaboration diagram for VMEReader:



Public Member Functions

- [VMEReader](#) (const char *device, [VME::BridgeType](#) type, bool on_socket=true)
- virtual [~VMEReader](#) ()
- void [AddTDC](#) (uint32_t address)
Add a TDC to handle.
- [VME::TDCV1x90](#) * [GetTDC](#) (uint32_t address)
Get a TDC on the VME bus Return a pointer to the TDC object, given its physical address on the VME bus.
- size_t [GetNumTDC](#) () const
- [VME::TDCCollection](#) [GetTDCCollection](#) ()
- void [AddIOModule](#) (uint32_t address)
- [VME::IOModuleV262](#) * [GetIOModule](#) ()
- void [AddCFD](#) (uint32_t address)
Add a CFD to handle.
- [VME::CFDV812](#) * [GetCFD](#) (uint32_t address)
Get a CFD on the VME bus Return a pointer to the CFD object, given its physical address on the VME bus.
- size_t [GetNumCFD](#) () const
- [VME::CFDCollection](#) [GetCFDCollection](#) ()
- void [AddFPGAUnit](#) (uint32_t address)
Add a multi-purposes FPGA board (CAEN V1495) to the crate controller.
- [VME::FPGAUnitV1495](#) * [GetFPGAUnit](#) ()
Return the pointer to the FPGA board connected to this controller (if any ; 0 otherwise)
- unsigned int [GetRunNumber](#) ()
Ask the socket master a run number.
- void [StartPulser](#) (double period, double width, unsigned int num_pulses=0)
Start the bridge's pulse generator [faulty].
- void [StopPulser](#) ()
Stop the bridge's pulse generator [faulty].
- void [SendPulse](#) (unsigned short output=0) const
Send a single pulse to the output register/plug connected to TDC boards.
- void [SendClear](#) () const
Send a clear signal to both the TDC boards.
- void [SetOutputFile](#) (std::string filename)

Set the path to the output file where the DAQ is to write.

- `std::string GetOutputFile () const`

Return the path to the output file the DAQ is currently writing to.

- `void Abort ()`

Abort data collection for all modules on the bus handled by the bridge.

Private Attributes

- `VME::BridgeVx718 * fBridge`

The VME bridge object to handle.

- `VME::TDCCollection fTDCCollection`

A set of pointers to TDC objects indexed by their physical VME address.

- `VME::CFDCollection fCFDCollection`

A set of pointers to CFD objects indexed by their physical VME address.

- `VME::IOModuleV262 * fSG`

Pointer to the VME input/output module object.

- `VME::FPGAUnitV1495 * fFPGA`

Pointer to the VME general purpose FPGA unit object.

- `bool fOnSocket`

Are we dealing with socket message passing?

- `bool fIsPulserStarted`

Is the bridge's pulser already started?

- `std::string fOutputFile`

Path to the current output file the DAQ is writing to.

Additional Inherited Members

7.28.1 Detailed Description

VME reader object to fetch events on a HPTDC board

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

4 May 2015

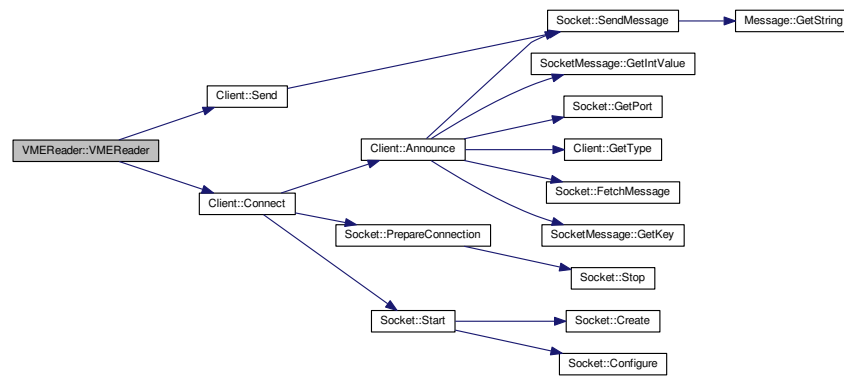
7.28.2 Constructor & Destructor Documentation

7.28.2.1 VMEReader::VMEReader (const char * device, VME::BridgeType type, bool on_socket = true)

Parameters

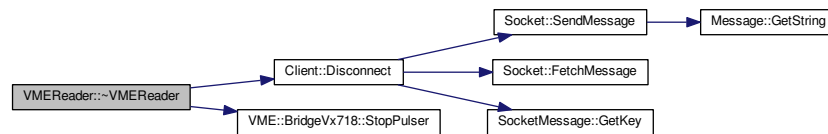
in	<i>device</i>	Path to the device (/dev/xxx)
in	<i>type</i>	Bridge model
in	<i>on_socket</i>	Are we trying to connect through the socket?

Here is the call graph for this function:



7.28.2.2 VMEReader::~~VMEReader () [virtual]

Here is the call graph for this function:

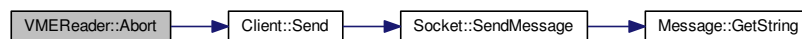


7.28.3 Member Function Documentation

7.28.3.1 void VMEReader::Abort ()

Abort data collection for all modules on the bus handled by the bridge.

Here is the call graph for this function:



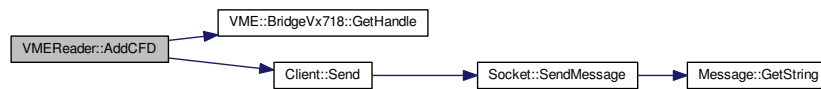
7.28.3.2 void VMEReader::AddCFD (uint32_t address)

Add a CFD to handle.

Parameters

in	<i>address</i>	32-bit address of the CFD module on the VME bus Create a new CFD handler for the VME bus
----	----------------	--

Here is the call graph for this function:



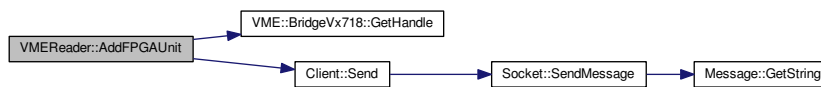
7.28.3.3 void VMEReaders::AddFPGAUnit (uint32_t *address*)

Add a multi-purposes FPGA board (CAEN V1495) to the crate controller.

Parameters

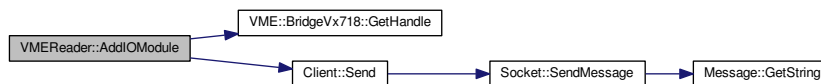
in	<i>address</i>	32-bit address of the TDC module on the VME bus
----	----------------	---

Here is the call graph for this function:



7.28.3.4 void VMEReaders::AddIOModule (uint32_t *address*)

Here is the call graph for this function:



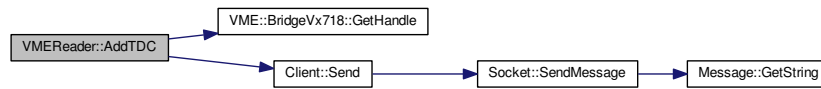
7.28.3.5 void VMEReaders::AddTDC (uint32_t *address*)

Add a TDC to handle.

Parameters

in	<i>address</i>	32-bit address of the TDC module on the VME bus Create a new TDC handler for the VME bus
----	----------------	--

Here is the call graph for this function:



7.28.3.6 VME::CFDV812* VMEReader::GetCFD (uint32_t address) [inline]

Get a CFD on the [VME](#) bus Return a pointer to the CFD object, given its physical address on the [VME](#) bus.

7.28.3.7 VME::CFDCollection VMEReader::GetCFDCollection () [inline]

7.28.3.8 VME::FPGAUnitV1495* VMEReader::GetFPGAUnit () [inline]

Return the pointer to the FPGA board connected to this controller (if any ; 0 otherwise)

7.28.3.9 VME::IOModuleV262* VMEReader::GetIOModule () [inline]

7.28.3.10 size_t VMEReader::GetNumCFD () const [inline]

7.28.3.11 size_t VMEReader::GetNumTDC () const [inline]

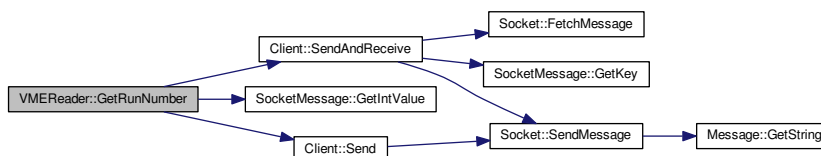
7.28.3.12 std::string VMEReader::GetOutputFile () const [inline]

Return the path to the output file the DAQ is currently writing to.

7.28.3.13 unsigned int VMEReader::GetRunNumber ()

Ask the socket master a run number.

Here is the call graph for this function:



7.28.3.14 VME::TDCV1x90* VMEReader::GetTDC (uint32_t address) [inline]

Get a TDC on the [VME](#) bus Return a pointer to the TDC object, given its physical address on the [VME](#) bus.

7.28.3.15 **VME::TDCCollection** VMEReader::GetTDCCollection () [inline]

7.28.3.16 **void** VMEReader::SendClear () const [inline]

Send a clear signal to both the TDC boards.

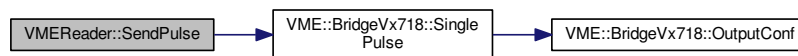
Here is the call graph for this function:



7.28.3.17 **void** VMEReader::SendPulse (unsigned short *output* = 0) const [inline]

Send a single pulse to the output register/plug connected to TDC boards.

Here is the call graph for this function:



7.28.3.18 **void** VMEReader::SetOutputFile (std::string *filename*)

Set the path to the output file where the DAQ is to write.

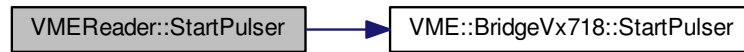
Here is the call graph for this function:



7.28.3.19 **void** VMEReader::StartPulser (double *period*, double *width*, unsigned int *num_pulses* = 0) [inline]

Start the bridge's pulse generator [faulty].

Here is the call graph for this function:



7.28.3.20 `void VMEReader::StopPulser () [inline]`

Stop the bridge's pulse generator [faulty].

Here is the call graph for this function:



7.28.4 Field Documentation

7.28.4.1 `VME::BridgeVx718* VMEReader::fBridge [private]`

The [VME](#) bridge object to handle.

7.28.4.2 `VME::CFDCollection VMEReader::fCFDCollection [private]`

A set of pointers to CFD objects indexed by their physical [VME](#) address.

7.28.4.3 `VME::FPGAUnitV1495* VMEReader::fFPGA [private]`

Pointer to the [VME](#) general purpose FPGA unit object.

7.28.4.4 `bool VMEReader::fIsPulserStarted [private]`

Is the bridge's pulser already started?

7.28.4.5 `bool VMEReader::fOnSocket [private]`

Are we dealing with socket message passing?

7.28.4.6 `std::string VMEReader::fOutputFile [private]`

Path to the current output file the DAQ is writing to.

7.28.4.7 VME::IOModuleV262* VMEReader::fSG [private]

Pointer to the [VME](#) input/output module object.

7.28.4.8 VME::TDCCollection VMEReader::fTDCCollection [private]

A set of pointers to TDC objects indexed by their physical [VME](#) address.

The documentation for this class was generated from the following files:

- include/VMEReader.h
- src/VMEReader.cpp

Index

- ~BridgeVx718
 - VME::BridgeVx718, [24](#)
- ~BridgeVx718Control
 - VME::BridgeVx718Control, [26](#)
- ~BridgeVx718Status
 - VME::BridgeVx718Status, [28](#)
- ~CAENETControllerV288
 - VME::CAENETControllerV288, [30](#)
- ~CAENETControllerV288Status
 - VME::CAENETControllerV288Status, [32](#)
- ~CFDV812
 - VME::CFDV812, [34](#)
- ~Client
 - Client, [40](#)
- ~FPGAUnitV1495
 - VME::FPGAUnitV1495, [50](#)
- ~FPGAUnitV1495Control
 - VME::FPGAUnitV1495Control, [61](#)
- ~FileReader
 - FileReader, [46](#)
- ~GenericBoard
 - VME::GenericBoard, [64](#)
- ~IOModuleV262
 - VME::IOModuleV262, [70](#)
- ~Message
 - Message, [73](#)
- ~Messenger
 - Messenger, [76](#)
- ~PCIInterfaceA2818
 - VME::PCIInterfaceA2818, [81](#)
- ~Socket
 - Socket, [84](#)
- ~SocketMessage
 - SocketMessage, [92](#)
- ~TDCErrorFlag
 - VME::TDCErrorFlag, [96](#)
- ~TDCEvent
 - VME::TDCEvent, [98](#)
- ~TDCMeasurement
 - VME::TDCMeasurement, [104](#)
- ~TDCV1x90
 - VME::TDCV1x90, [110](#)
- ~TDCV1x90Control
 - VME::TDCV1x90Control, [130](#)
- ~TDCV1x90Status
 - VME::TDCV1x90Status, [137](#)
- ~VMEReader
 - VMEReader, [143](#)
- AUTOLOAD_DEF_CONFI
 - VME::TDCV1x90OpCodes, [18](#)
- AUTOLOAD_USER_CONF
 - VME::TDCV1x90OpCodes, [18](#)
- Abort
 - VMEReader, [143](#)
- abort
 - VME::TDCV1x90, [110](#)
- AcceptConnections
 - Socket, [84](#)
- acq_mode
 - file_header_t, [45](#)
- AcquisitionMode
 - VME, [13](#)
- AddCFD
 - VMEReader, [143](#)
- AddClient
 - Messenger, [76](#)
- AddFPGAUnit
 - VMEReader, [144](#)
- AddIOModule
 - VMEReader, [144](#)
- AddTDC
 - VMEReader, [144](#)
- AlmostFull
 - VME::TDCV1x90Status, [137](#)
- Announce
 - Client, [41](#)
- Bind
 - Socket, [84](#)
- BridgeType
 - VME, [13](#)
- BridgeVx718
 - VME::BridgeVx718, [23](#)
- BridgeVx718Control
 - VME::BridgeVx718Control, [26](#)
- BridgeVx718Status
 - VME::BridgeVx718Status, [28](#)
- Broadcast
 - Messenger, [77](#)
- BusError
 - VME::TDCV1x90Status, [137](#)
- CAEN_V1718
 - VME, [13](#)
- CAEN_V2718
 - VME, [13](#)
- CAENETControllerV288
 - VME::CAENETControllerV288, [30](#)
- CAENETControllerV288Register

- VME, [13](#)
- CAENETControllerV288Status
 - VME::CAENETControllerV288Status, [32](#)
- CFDCollection
 - VME, [13](#)
- CFDV812
 - VME::CFDV812, [34](#)
- CFDV812Register
 - VME, [13](#)
- CLEAR_KEEP_TOKEN
 - VME::TDCV1x90Opcodes, [18](#)
- CLIENT
 - Socket communication objects, [9](#)
- CONT_STOR
 - VME::TDCV1x90Opcodes, [18](#)
- CONT_STORAGE
 - VME, [13](#)
- CheckBoardVersion
 - VME::FPGAUnitV1495, [50](#)
- CheckConfiguration
 - VME::BridgeVx718, [24](#)
 - VME::CFDV812, [34](#)
 - VME::TDCV1x90, [110](#)
- CheckPCIInterface
 - VME::BridgeVx718, [24](#)
- ClearOutputPulser
 - VME::FPGAUnitV1495, [51](#)
- Client, [39](#)
 - ~Client, [40](#)
 - Announce, [41](#)
 - Client, [40](#)
 - Connect, [41](#)
 - Disconnect, [42](#)
 - fClientId, [44](#)
 - flsConnected, [44](#)
 - fType, [44](#)
 - GetType, [42](#)
 - ParseMessage, [42](#)
 - Receive, [42](#)
 - Send, [43](#)
 - SendAndReceive, [43](#)
- ClockSource
 - VME::FPGAUnitV1495Control, [60](#)
- coarse
 - VME::GlobalOffset, [66](#)
- Configure
 - Socket, [85](#)
- Connect
 - Client, [41](#)
 - Messenger, [77](#)
- Create
 - Socket, [85](#)
- DEFAULT_SETUP_REG
 - VME::TDCV1x90Opcodes, [18](#)
- DETECTOR
 - Socket communication objects, [9](#)
- DIS_ALL_CHANNEL
 - VME::TDCV1x90Opcodes, [18](#)
- DIS_CHANNEL
 - VME::TDCV1x90Opcodes, [18](#)
- DIS_ERROR_BYPASS
 - VME::TDCV1x90Opcodes, [18](#)
- DIS_ERROR_MARK
 - VME::TDCV1x90Opcodes, [18](#)
- DIS_HEAD_TRAILER
 - VME::TDCV1x90Opcodes, [18](#)
- DIS_SUB_TRG
 - VME::TDCV1x90Opcodes, [18](#)
- DISABLE_TEST_MODE
 - VME::TDCV1x90Opcodes, [18](#)
- DLL_Direct_LowRes
 - VME::TDCV1x90, [109](#)
- DLL_PLL_HighRes
 - VME::TDCV1x90, [109](#)
- DLL_PLL_LowRes
 - VME::TDCV1x90, [109](#)
- DLL_PLL_MedRes
 - VME::TDCV1x90, [109](#)
- DLLMode
 - VME::TDCV1x90, [109](#)
- DQM
 - Socket communication objects, [9](#)
- DataReady
 - VME::TDCV1x90Status, [137](#)
- DeadTimeCalculator
 - VME::CFDV812, [34](#)
- Decode
 - HTTPMessage, [68](#)
- det_mode
 - file_header_t, [45](#)
- DetectionMode
 - VME, [14](#)
- DisableChannel
 - VME::TDCV1x90, [110](#)
- Disconnect
 - Client, [42](#)
 - Messenger, [78](#)
- DisconnectClient
 - Messenger, [78](#)
- Dump
 - FileReader, [46](#)
 - HTTPMessage, [68](#)
 - Message, [73](#)
 - SocketMessage, [92](#)
 - VME::BridgeVx718Status, [28](#)
 - VME::FPGAUnitV1495Control, [61](#)
 - VME::TDCErrorFlag, [96](#)
 - VME::TDCEvent, [98](#)
 - VME::TDCMeasurement, [104](#)
 - VME::TDCV1x90Control, [130](#)
 - VME::TDCV1x90Status, [137](#)
- DumpConnected
 - Socket, [85](#)
- DumpFWInformation
 - VME::FPGAUnitV1495, [51](#)
- EN_ALL_CHANNEL

- VME::TDCV1x90Opcodes, 18
- EN_CHANNEL
 - VME::TDCV1x90Opcodes, 18
- EN_ERROR_BYPASS
 - VME::TDCV1x90Opcodes, 18
- EN_ERROR_MARK
 - VME::TDCV1x90Opcodes, 18
- EN_HEAD_TRAILER
 - VME::TDCV1x90Opcodes, 18
- EN_SUB_TRG
 - VME::TDCV1x90Opcodes, 18
- ENABLE_TEST_MODE
 - VME::TDCV1x90Opcodes, 18
- ETTT
 - VME::TDCEvent, 98
- EXTRA_SEARCH_WIN_WIDTH
 - VME, 16
- EnableChannel
 - VME::TDCV1x90, 111
- Encode
 - HTTPMessage, 68
- Error
 - VME::TDCV1x90Status, 138
- ettt
 - VME::trailead_t, 140
- event_count
 - VME::trailead_t, 140
- EventType
 - VME::TDCEvent, 98
- ExternalClock
 - VME::FPGAUnitV1495Control, 60
- ExternalSignal
 - VME::FPGAUnitV1495Control, 61
- ExternalTrigger
 - VME::FPGAUnitV1495Control, 61
- fAcquisitionMode
 - VME::TDCV1x90, 128
- fAddress
 - Socket, 87
- fBaseAddr
 - VME::GenericBoard, 65
- fBridge
 - VMEReader, 147
- fBuffer
 - Socket, 87
 - VME::TDCV1x90, 128
- fCFDCollection
 - VMEReader, 147
- fClientId
 - Client, 44
- fDetectionMode
 - VME::TDCV1x90, 128
- fErrorMarks
 - VME::TDCV1x90, 128
- fEvents
 - VME::TDCMeasurement, 106
- fFPGA
 - VMEReader, 147
- fFile
 - FileReader, 47
- fHandle
 - VME::GenericBoard, 65
 - VME::PCIInterfaceA2818, 82
- fHasIRQ
 - VME::BridgeVx718, 25
- fHeader
 - FileReader, 47
- fIsConnected
 - Client, 44
- fIsPulserStarted
 - VMEReader, 147
- fMap
 - VME::TDCMeasurement, 106
- fMaster
 - Socket, 87
- fMessage
 - SocketMessage, 95
- fNumAttempts
 - Messenger, 81
- fNumEvents
 - FileReader, 47
- fNumWordsInBuffer
 - VME::CAENETControllerV288, 31
- fOnSocket
 - VMEReader, 147
- fOriginalString
 - HTTPMessage, 68
- fOutputFile
 - VMEReader, 147
- FPGAUnitV1495
 - VME::FPGAUnitV1495, 50
- FPGAUnitV1495Control
 - VME::FPGAUnitV1495Control, 61
- FPGAUnitV1495Register
 - VME, 14
- fPID
 - Messenger, 81
- fPort
 - Socket, 87
- fReadFds
 - Socket, 88
- fReadoutMode
 - FileReader, 48
- fSG
 - VMEReader, 147
- fScalerStarted
 - VME::FPGAUnitV1495, 59
- fSocketId
 - Socket, 88
- fSocketsConnected
 - Socket, 88
- fStderrPipe
 - Messenger, 81
- fStdoutPipe
 - Messenger, 81
- fString

- Message, 74
- fTDCCollection
 - VMEReader, 148
- fType
 - Client, 44
- fVerb
 - VME::TDCV1x90, 128
- fWS
 - HTTPMessage, 68
 - Messenger, 81
- fWindowWidth
 - VME::TDCV1x90, 128
- fWord
 - VME::BridgeVx718Control, 27
 - VME::BridgeVx718Status, 29
 - VME::CAENETControllerV288Status, 32
 - VME::FPGAUnitV1495Control, 64
 - VME::TDCErrorFlag, 96
 - VME::TDCEvent, 102
 - VME::TDCV1x90Control, 135
 - VME::TDCV1x90Status, 139
- fWriteTime
 - FileReader, 48
- FetchEvents
 - VME::TDCV1x90, 111
- FetchMessage
 - Socket, 85
- file_header_t, 44
 - acq_mode, 45
 - det_mode, 45
 - magic, 45
 - num_hptdc, 45
 - run_id, 45
 - spill_id, 45
- FileReader, 45
 - ~FileReader, 46
 - Dump, 46
 - fFile, 47
 - fHeader, 47
 - fNumEvents, 47
 - fReadoutMode, 48
 - fWriteTime, 48
 - FileReader, 46
 - GetNextEvent, 46
 - GetNextMeasurement, 47
 - GetNumEvents, 47
 - GetNumTDCs, 47
- Filler
 - VME::TDCEvent, 98
- fine
 - VME::GlobalOffset, 66
- Full
 - VME::TDCV1x90Status, 139
- gEnd
 - VME::TDCV1x90, 128
- GenericBoard
 - VME::GenericBoard, 64
- GetAcquisitionMode
 - VME::TDCV1x90, 111
- GetAddressIncrement
 - VME::BridgeVx718Control, 26
- GetAlign64
 - VME::TDCV1x90Control, 131
- GetArbiterType
 - VME::BridgeVx718Control, 26
- GetBERR
 - VME::BridgeVx718Status, 28
- GetBLTEventNumberRegister
 - VME::TDCV1x90, 112
- GetBit
 - VME::FPGAUnitV1495Control, 61
- GetBunchId
 - VME::TDCEvent, 98
 - VME::TDCMeasurement, 105
- GetBusError
 - VME::TDCV1x90Control, 131
- GetBusReqLevel
 - VME::BridgeVx718Control, 26
- GetBusTimeout
 - VME::BridgeVx718Control, 27
- GetCAENFirmwareRevision
 - VME::FPGAUnitV1495, 51
- GetCFD
 - VMEReader, 145
- GetCFDCollection
 - VMEReader, 145
- GetChannelDeadTime
 - VME::TDCV1x90, 112
- GetChannelId
 - VME::TDCEvent, 99
 - VME::TDCMeasurement, 105
- GetClockSource
 - VME::FPGAUnitV1495Control, 61
- GetCompensation
 - VME::TDCV1x90Control, 131
- GetControl
 - VME::FPGAUnitV1495, 52
 - VME::TDCV1x90, 112
- GetDLLClock
 - VME::TDCV1x90, 113
- GetDTACK
 - VME::BridgeVx718Status, 28
- GetDetectionMode
 - VME::TDCV1x90, 113
- GetDipSwitch
 - VME::BridgeVx718Status, 28
- GetETTT
 - VME::TDCEvent, 99
 - VME::TDCMeasurement, 105
 - VME::TDCV1x90, 113
 - VME::TDCV1x90Control, 132
- GetEmptyEvent
 - VME::TDCV1x90Control, 132
- GetErrorFlags
 - VME::TDCEvent, 99
- GetErrorMarks

- VME::TDCV1x90, [113](#)
- GetEventCount
 - VME::TDCEvent, [100](#)
- GetEventCounter
 - VME::TDCV1x90, [113](#)
- GetEventFIFO
 - VME::TDCV1x90Control, [132](#)
- GetEventId
 - VME::TDCEvent, [100](#)
 - VME::TDCMeasurement, [105](#)
- GetEventStored
 - VME::TDCV1x90, [114](#)
- GetFIFOSize
 - VME::TDCV1x90, [114](#)
- GetFPGAUnit
 - VMEReader, [145](#)
- GetFWRevision
 - VME::PCIInterfaceA2818, [81](#)
- GetFirmwareRevision
 - VME::TDCV1x90, [114](#)
- GetFixedCode
 - VME::CFDV812, [34](#)
- GetGeo
 - VME::TDCEvent, [100](#)
- GetGeoAddress
 - VME::FPGAUnitV1495, [52](#)
- GetGlobalOffset
 - VME::TDCV1x90, [115](#)
- GetHandle
 - VME::BridgeVx718, [24](#)
- GetHardwareRevision
 - VME::FPGAUnitV1495, [52](#)
- GetIOModule
 - VMEReader, [145](#)
- GetIRQStatus
 - VME::BridgeVx718, [24](#)
- GetIdentifier
 - VME::IOModuleV262, [70](#)
- GetIntValue
 - SocketMessage, [93](#)
- GetInternalClockPeriod
 - VME::FPGAUnitV1495, [53](#)
- GetInternalTriggerPeriod
 - VME::FPGAUnitV1495, [53](#)
- GetInterruptReq
 - VME::BridgeVx718Control, [27](#)
- GetKey
 - HTTPMessage, [68](#)
 - Message, [73](#)
 - SocketMessage, [93](#)
- GetLeadingTime
 - VME::TDCMeasurement, [105](#)
- GetMEBAccess
 - VME::TDCV1x90Control, [132](#)
- GetManufacturerId
 - VME::CFDV812, [35](#)
 - VME::IOModuleV262, [70](#)
- GetModel
 - VME::TDCV1x90, [115](#)
- GetModuleType
 - VME::CFDV812, [35](#)
 - VME::IOModuleV262, [71](#)
- GetModuleVersion
 - VME::CFDV812, [35](#)
 - VME::IOModuleV262, [71](#)
- GetNextEvent
 - FileReader, [46](#)
- GetNextMeasurement
 - FileReader, [47](#)
- GetNumCFD
 - VMEReader, [145](#)
- GetNumEvents
 - FileReader, [47](#)
- GetNumTDC
 - VMEReader, [145](#)
- GetNumTDCs
 - FileReader, [47](#)
- GetOUI
 - VME::TDCV1x90, [115](#)
- GetOperationStatus
 - VME::CAENETControllerV288Status, [32](#)
- GetOutputDelay
 - VME::FPGAUnitV1495, [53](#)
- GetOutputFile
 - VMEReader, [145](#)
- GetOutputPulser
 - VME::FPGAUnitV1495, [54](#)
- GetPol
 - VME::TDCV1x90, [116](#)
- GetPort
 - Socket, [85](#)
- GetRCAdjust
 - VME::TDCV1x90, [116](#)
- GetReleaseType
 - VME::BridgeVx718Control, [27](#)
- GetRequesterType
 - VME::BridgeVx718Control, [27](#)
- GetResolution
 - VME::TDCV1x90, [116](#)
- GetRunNumber
 - VMEReader, [145](#)
- GetSRAMCompensation
 - VME::TDCV1x90Control, [132](#)
- GetSWTermination
 - VME::TDCV1x90Control, [132](#)
- GetScalerStatus
 - VME::FPGAUnitV1495Control, [61](#)
- GetScalerValue
 - VME::FPGAUnitV1495, [54](#)
- GetSerialNumber
 - VME::CFDV812, [36](#)
 - VME::FPGAUnitV1495, [54](#)
 - VME::IOModuleV262, [71](#)
 - VME::TDCV1x90, [117](#)
- GetSignalSource
 - VME::FPGAUnitV1495Control, [61](#)

- GetSocketId
 - Socket, [85](#)
- GetSocketType
 - Socket, [85](#)
- GetStatus
 - VME::BridgeVx718, [24](#)
 - VME::CAENETControllerV288, [30](#)
 - VME::TDCEvent, [101](#)
 - VME::TDCV1x90, [117](#)
- GetString
 - Message, [73](#)
 - SocketMessage, [93](#)
- GetSysRes
 - VME::BridgeVx718Control, [27](#)
- GetSystemControl
 - VME::BridgeVx718Status, [28](#)
- GetSystemReset
 - VME::BridgeVx718Status, [28](#)
- GetTDC
 - VMEReader, [145](#)
- GetTDCBits
 - VME::FPGAUnitV1495, [55](#)
- GetTDCCollection
 - VMEReader, [145](#)
- GetTDCEncapsulation
 - VME::TDCV1x90, [117](#)
- GetTDCId
 - VME::TDCEvent, [101](#)
 - VME::TDCMeasurement, [105](#)
- GetTermination
 - VME::TDCV1x90Control, [132](#)
- GetTestFIFO
 - VME::TDCV1x90Control, [132](#)
- GetTestMode
 - VME::TDCV1x90, [118](#)
- GetTime
 - VME::TDCEvent, [101](#)
- GetToT
 - VME::TDCMeasurement, [105](#)
- GetTrailingTime
 - VME::TDCMeasurement, [106](#)
- GetTriggerConfiguration
 - VME::TDCV1x90, [118](#)
- GetTriggerSource
 - VME::FPGAUnitV1495Control, [62](#)
- GetType
 - Client, [42](#)
 - Messenger, [78](#)
 - VME::TDCEvent, [101](#)
- GetUSBType
 - VME::BridgeVx718Status, [28](#)
- GetUserFirmwareRevision
 - VME::FPGAUnitV1495, [55](#)
- GetValue
 - SocketMessage, [93](#)
 - VME::TDCV1x90Control, [132](#)
 - VME::TDCV1x90Status, [139](#)
- GetVectorValue
 - SocketMessage, [93](#)
- GetWidth
 - VME::TDCEvent, [101](#)
- GetWindowOffset
 - VME::TDCV1x90, [118](#)
- GetWindowWidth
 - VME::TDCV1x90, [118](#)
- GetWord
 - VME::FPGAUnitV1495Control, [62](#)
 - VME::TDCErrorFlag, [96](#)
 - VME::TDCEvent, [102](#)
- GetWordCount
 - VME::TDCEvent, [102](#)
- GlobalHeader
 - VME::TDCEvent, [98](#)
- GlobalTrailer
 - VME::TDCEvent, [98](#)
- HTTPMessage, [66](#)
 - Decode, [68](#)
 - Dump, [68](#)
 - Encode, [68](#)
 - fOriginalString, [68](#)
 - fWS, [68](#)
 - GetKey, [68](#)
 - HTTPMessage, [67](#), [68](#)
- HardwareReset
 - VME::TDCV1x90, [118](#)
- HasGroupError
 - VME::TDCErrorFlag, [96](#)
- HasInternalChipError
 - VME::TDCErrorFlag, [96](#)
- HasL1BufferOverflow
 - VME::TDCErrorFlag, [96](#)
- HasReachedEventSizeLimit
 - VME::TDCErrorFlag, [96](#)
- HasReadoutFIFOOverflow
 - VME::TDCErrorFlag, [96](#)
- HasTriggerFIFOOverflow
 - VME::TDCErrorFlag, [96](#)
- HeadersEnabled
 - VME::TDCV1x90Status, [139](#)
- INVALID
 - Socket communication objects, [9](#)
- IOModuleV262
 - VME::IOModuleV262, [70](#)
- IOModuleV262Register
 - VME, [14](#)
- IRQ1
 - VME::BridgeVx718, [23](#)
- IRQ2
 - VME::BridgeVx718, [23](#)
- IRQ3
 - VME::BridgeVx718, [23](#)
- IRQ4
 - VME::BridgeVx718, [23](#)
- IRQ5
 - VME::BridgeVx718, [23](#)

- IRQ6
 - VME::BridgeVx718, [23](#)
- IRQ7
 - VME::BridgeVx718, [23](#)
- IRQId
 - VME::BridgeVx718, [23](#)
- InputConf
 - VME::BridgeVx718, [24](#)
- InputRead
 - VME::BridgeVx718, [24](#)
- InternalClock
 - VME::FPGAUnitV1495Control, [60](#)
- InternalSignal
 - VME::FPGAUnitV1495Control, [61](#)
- InternalTrigger
 - VME::FPGAUnitV1495Control, [61](#)
- Invalid
 - VME::CAENETControllerV288Status, [32](#)
- IsFromWeb
 - Message, [73](#)
- IsTrailing
 - VME::TDCEvent, [102](#)
- IsWebSocket
 - Socket, [85](#)
- kBLTEventNumber
 - VME, [15](#)
- kBoardInfo0
 - VME, [15](#)
- kBoardInfo1
 - VME, [15](#)
- kClear
 - VME::FPGAUnitV1495, [50](#)
- kControl
 - VME, [15](#)
- kECLLevelWrite
 - VME, [15](#)
- kEventCounter
 - VME, [15](#)
- kEventFIFO
 - VME, [15](#)
- kEventFIFOStatusRegister
 - VME, [15](#)
- kEventFIFOStoredRegister
 - VME, [15](#)
- kEventStored
 - VME, [15](#)
- kFirmwareRev
 - VME, [15](#)
- kGeoAddress
 - VME, [15](#)
- kIdentifier
 - VME, [15](#)
- kInterruptLevel
 - VME, [15](#)
- kInterruptVector
 - VME, [15](#)
- kMCSTBase
 - VME, [15](#)
- kMCSTControl
 - VME, [15](#)
- kMicro
 - VME, [15](#)
- kMicroHandshake
 - VME, [15](#)
- kModuleReset
 - VME, [15](#)
- kNIMLevelWrite
 - VME, [15](#)
- kNIMPulseRead
 - VME, [15](#)
- kNIMPulseWrite
 - VME, [15](#)
- kOutputBuffer
 - VME, [15](#)
- kROMBoard0
 - VME, [15](#)
- kROMBoard1
 - VME, [15](#)
- kROMBoard2
 - VME, [15](#)
- kROMOui0
 - VME, [15](#)
- kROMOui1
 - VME, [15](#)
- kROMOui2
 - VME, [15](#)
- kROMRevis0
 - VME, [16](#)
- kROMRevis1
 - VME, [16](#)
- kROMRevis2
 - VME, [16](#)
- kROMRevis3
 - VME, [15](#)
- kROMSerNum0
 - VME, [16](#)
- kROMSerNum1
 - VME, [16](#)
- kReset
 - VME::FPGAUnitV1495, [50](#)
- kSoftwareClear
 - VME, [15](#)
- kStatus
 - VME, [15](#)
- kTrigger
 - VME::FPGAUnitV1495, [50](#)
- kV1495Board0
 - VME, [14](#)
- kV1495Board1
 - VME, [14](#)
- kV1495Board2
 - VME, [14](#)
- kV1495ClockSettings
 - VME, [14](#)
- kV1495ConfigurationROM
 - VME, [14](#)

- kV1495Control
 - VME, [14](#)
- kV1495DelaySettings
 - VME, [14](#)
- kV1495FWRevision
 - VME, [14](#)
- kV1495GeoAddress
 - VME, [14](#)
- kV1495HWRevision0
 - VME, [14](#)
- kV1495HWRevision1
 - VME, [14](#)
- kV1495HWRevision2
 - VME, [14](#)
- kV1495HWRevision3
 - VME, [14](#)
- kV1495ModuleReset
 - VME, [14](#)
- kV1495OUI0
 - VME, [14](#)
- kV1495OUI1
 - VME, [14](#)
- kV1495OUI2
 - VME, [14](#)
- kV1495OutputSettings
 - VME, [14](#)
- kV1495ScalerCounter
 - VME, [14](#)
- kV1495SerNum0
 - VME, [14](#)
- kV1495SerNum1
 - VME, [14](#)
- kV1495TDCBoardInterface
 - VME, [14](#)
- kV1495TriggerSettings
 - VME, [14](#)
- kV1495UserFPGAConfig
 - VME, [14](#)
- kV1495UserFPGAFlashMem
 - VME, [14](#)
- kV1495UserFWRevision
 - VME, [14](#)
- kV288DataBuffer
 - VME, [13](#)
- kV288IRQVector
 - VME, [13](#)
- kV288ModuleReset
 - VME, [13](#)
- kV288Status
 - VME, [13](#)
- kV288Transmission
 - VME, [13](#)
- kV812DeadTimeGroup0
 - VME, [14](#)
- kV812DeadTimeGroup1
 - VME, [14](#)
- kV812FixedCode
 - VME, [14](#)
- kV812Info0
 - VME, [14](#)
- kV812Info1
 - VME, [14](#)
- kV812MajorityThreshold
 - VME, [14](#)
- kV812OutputWidthGroup0
 - VME, [13](#)
- kV812OutputWidthGroup1
 - VME, [13](#)
- kV812PatternOfInhibit
 - VME, [14](#)
- kV812TestPulse
 - VME, [14](#)
- kV812ThresholdChannel0
 - VME, [13](#)
- LOAD_DEF_CONFIG
 - VME::TDCV1x90Opcodes, [18](#)
- LOAD_USER_CONFIG
 - VME::TDCV1x90Opcodes, [18](#)
- leading
 - VME::trailead_t, [140](#)
- Listen
 - Socket, [86](#)
- MASTER
 - Socket communication objects, [9](#)
- MATCH_WIN_WIDTH
 - VME, [16](#)
- magic
 - file_header_t, [45](#)
- Message, [72](#)
 - ~Message, [73](#)
 - Dump, [73](#)
 - fString, [74](#)
 - GetKey, [73](#)
 - GetString, [73](#)
 - IsFromWeb, [73](#)
 - Message, [73](#)
- Messenger, [74](#)
 - ~Messenger, [76](#)
 - AddClient, [76](#)
 - Broadcast, [77](#)
 - Connect, [77](#)
 - Disconnect, [78](#)
 - DisconnectClient, [78](#)
 - fNumAttempts, [81](#)
 - fPID, [81](#)
 - fStderrPipe, [81](#)
 - fStdoutPipe, [81](#)
 - fWS, [81](#)
 - GetType, [78](#)
 - Messenger, [76](#)
 - ProcessMessage, [79](#)
 - Receive, [79](#)
 - Send, [80](#)
 - StartAcquisition, [80](#)
 - StopAcquisition, [80](#)

- SwitchClientType, [80](#)
- micro_handshake
 - VME, [15](#)
- nchannels
 - VME::TDCV1x90, [128](#)
- num_hptdc
 - file_header_t, [45](#)
- NumErrors
 - VME::TDCMeasurement, [106](#)
- NumEvents
 - VME::TDCMeasurement, [106](#)
- OLEADING
 - VME, [14](#)
- OTRAILING
 - VME, [14](#)
- Object
 - SocketMessage, [93](#)
- OperationStatus
 - VME::CAENETControllerV288Status, [32](#)
- operator<<
 - VME::CAENETControllerV288, [31](#)
 - VME::TDCErrorFlag, [96](#)
- operator>>
 - VME::CAENETControllerV288, [31](#)
- OutputConf
 - VME::BridgeVx718, [25](#)
- OutputOff
 - VME::BridgeVx718, [25](#)
- OutputOn
 - VME::BridgeVx718, [25](#)
- OutputWidthCalculator
 - VME::CFDV812, [36](#)
- PAIR
 - VME, [14](#)
- PCIInterfaceA2818
 - VME::PCIInterfaceA2818, [81](#)
- pair_lead_res
 - VME::TDCV1x90, [128](#)
- pair_width_res
 - VME::TDCV1x90, [128](#)
- PairMode
 - VME::TDCV1x90Status, [139](#)
- ParseMessage
 - Client, [42](#)
- PrepareConnection
 - Socket, [86](#)
- ProcessMessage
 - Messenger, [79](#)
- PulseTDCBits
 - VME::FPGAUnitV1495, [55](#)
- Purged
 - VME::TDCV1x90Status, [139](#)
- r100ps
 - VME, [16](#)
- r200ps
 - VME, [16](#)
- r25ps
 - VME, [16](#)
- r800ps
 - VME, [16](#)
- R_100ps
 - VME::TDCV1x90Status, [136](#)
- R_200ps
 - VME::TDCV1x90Status, [136](#)
- R_25ps
 - VME::TDCV1x90Status, [136](#)
- R_800ps
 - VME::TDCV1x90Status, [136](#)
- READ_ACQ_MOD
 - VME::TDCV1x90Opcodes, [18](#)
- READ_ADJUST_CH
 - VME::TDCV1x90Opcodes, [18](#)
- READ_DEAD_TIME
 - VME::TDCV1x90Opcodes, [18](#)
- READ_DETECTION
 - VME::TDCV1x90Opcodes, [18](#)
- READ_DLL_LOCK
 - VME::TDCV1x90Opcodes, [18](#)
- READ_EEPROM
 - VME::TDCV1x90Opcodes, [18](#)
- READ_EN_PATTERN
 - VME::TDCV1x90Opcodes, [18](#)
- READ_EN_PATTERN32
 - VME::TDCV1x90Opcodes, [19](#)
- READ_ERROR_STATUS
 - VME::TDCV1x90Opcodes, [19](#)
- READ_ERROR_TYPES
 - VME::TDCV1x90Opcodes, [19](#)
- READ_EVENT_SIZE
 - VME::TDCV1x90Opcodes, [19](#)
- READ_FIFO_SIZE
 - VME::TDCV1x90Opcodes, [19](#)
- READ_GLOB_OFFS
 - VME::TDCV1x90Opcodes, [19](#)
- READ_HEAD_TRAILER
 - VME::TDCV1x90Opcodes, [19](#)
- READ_MICRO_REV
 - VME::TDCV1x90Opcodes, [19](#)
- READ_OK
 - VME, [15](#)
- READ_RC_ADJ
 - VME::TDCV1x90Opcodes, [19](#)
- READ_RES
 - VME::TDCV1x90Opcodes, [19](#)
- READ_SETUP_REG
 - VME::TDCV1x90Opcodes, [19](#)
- READ_SETUP_SCANPATH
 - VME::TDCV1x90Opcodes, [19](#)
- READ_SPARE
 - VME::TDCV1x90Opcodes, [19](#)
- READ_STATUS_STREAM
 - VME::TDCV1x90Opcodes, [19](#)
- READ_TDC_ID

- VME::TDCV1x90Opcodes, [19](#)
- READ_TRG_CONF
 - VME::TDCV1x90Opcodes, [19](#)
- REJECT_MARGIN
 - VME, [16](#)
- RESET_DLL_PLL
 - VME::TDCV1x90Opcodes, [19](#)
- REV_DATE_MICRO_FW
 - VME::TDCV1x90Opcodes, [19](#)
- ReadAcquisitionMode
 - VME::TDCV1x90, [118](#)
- ReadDetectionMode
 - VME::TDCV1x90, [119](#)
- ReadRegister
 - VME::GenericBoard, [65](#)
- Receive
 - Client, [42](#)
 - Messenger, [79](#)
 - VME::CAENETControllerV288, [30](#)
- Reset
 - VME::BridgeVx718, [25](#)
- ResetFPGA
 - VME::FPGAUnitV1495, [56](#)
- Resolution
 - VME::TDCV1x90Status, [139](#)
- run_id
 - file_header_t, [45](#)
- SAVE_RC_ADJ
 - VME::TDCV1x90Opcodes, [19](#)
- SAVE_USER_CONFIG
 - VME::TDCV1x90Opcodes, [19](#)
- SET_ADJUST_CH
 - VME::TDCV1x90Opcodes, [19](#)
- SET_DEAD_TIME
 - VME::TDCV1x90Opcodes, [19](#)
- SET_DETECTION
 - VME::TDCV1x90Opcodes, [19](#)
- SET_DLL_CLOCK
 - VME::TDCV1x90Opcodes, [19](#)
- SET_ERROR_TYPES
 - VME::TDCV1x90Opcodes, [19](#)
- SET_EVENT_SIZE
 - VME::TDCV1x90Opcodes, [19](#)
- SET_FIFO_SIZE
 - VME::TDCV1x90Opcodes, [19](#)
- SET_GLOB_OFFS
 - VME::TDCV1x90Opcodes, [19](#)
- SET_KEEP_TOKEN
 - VME::TDCV1x90Opcodes, [20](#)
- SET_PAIR_RES
 - VME::TDCV1x90Opcodes, [20](#)
- SET_RC_ADJ
 - VME::TDCV1x90Opcodes, [20](#)
- SET_REJ_MARGIN
 - VME::TDCV1x90Opcodes, [20](#)
- SET_SW_MARGIN
 - VME::TDCV1x90Opcodes, [20](#)
- SET_TDC_TSET_OUTPUT
 - VME::TDCV1x90Opcodes, [20](#)
- SET_TR_LEAD_LSB
 - VME::TDCV1x90Opcodes, [20](#)
- SET_WIN_OFFS
 - VME::TDCV1x90Opcodes, [20](#)
- SET_WIN_WIDTH
 - VME::TDCV1x90Opcodes, [20](#)
- SelectConnections
 - Socket, [86](#)
- Send
 - Client, [43](#)
 - Messenger, [80](#)
 - VME::CAENETControllerV288, [31](#)
- SendAndReceive
 - Client, [43](#)
- SendClear
 - VMEReader, [146](#)
- SendMessage
 - Socket, [86](#)
- SendPulse
 - VMEReader, [146](#)
- SetAcquisitionMode
 - VME::TDCV1x90, [119](#)
- SetAlign64
 - VME::TDCV1x90Control, [132](#)
- SetBLTEventNumberRegister
 - VME::TDCV1x90, [119](#)
- SetBit
 - VME::FPGAUnitV1495Control, [62](#)
- SetBusError
 - VME::TDCV1x90Control, [132](#)
- SetChannelDeadTime
 - VME::TDCV1x90, [120](#)
- SetClockSource
 - VME::FPGAUnitV1495Control, [62](#)
- SetCompensation
 - VME::TDCV1x90Control, [132](#)
- SetContinuousStorage
 - VME::TDCV1x90, [120](#)
- SetControl
 - VME::FPGAUnitV1495, [56](#)
 - VME::TDCV1x90, [120](#)
- SetDLLClock
 - VME::TDCV1x90, [121](#)
- SetDeadTime
 - VME::CFDV812, [36](#)
- SetDetectionMode
 - VME::TDCV1x90, [121](#)
- SetETTT
 - VME::TDCV1x90, [122](#)
 - VME::TDCV1x90Control, [133](#)
- SetEmptyEvent
 - VME::TDCV1x90Control, [133](#)
- SetErrorMarks
 - VME::TDCV1x90, [121](#)
- SetEventFIFO
 - VME::TDCV1x90Control, [133](#)
- SetEventsCollection

- VME::TDCMeasurement, 106
- SetFIFOSize
 - VME::TDCV1x90, 122
- SetGlobalOffset
 - VME::TDCV1x90, 122
- SetIRQ
 - VME::BridgeVx718, 25
- SetInternalClockPeriod
 - VME::FPGAUnitV1495, 56
- SetInternalTriggerPeriod
 - VME::FPGAUnitV1495, 57
- SetKeyValue
 - SocketMessage, 93, 94
- SetLSBTraileadEdge
 - VME::TDCV1x90, 123
- SetMEBAccess
 - VME::TDCV1x90Control, 134
- SetOutputDelay
 - VME::FPGAUnitV1495, 57
- SetOutputFile
 - VMEReader, 146
- SetOutputPulser
 - VME::FPGAUnitV1495, 57
- SetOutputPulserPOI
 - VME::FPGAUnitV1495, 58
- SetOutputWidth
 - VME::CFDV812, 36
- SetPOI
 - VME::CFDV812, 38
- SetPairModeResolution
 - VME::TDCV1x90, 123
- SetPol
 - VME::TDCV1x90, 123
- SetPort
 - Socket, 87
- SetRCAdjust
 - VME::TDCV1x90, 124
- SetSRAMCompensation
 - VME::TDCV1x90Control, 134
- SetSWTermination
 - VME::TDCV1x90Control, 134
- SetScalerReset
 - VME::FPGAUnitV1495Control, 63
- SetScalerStatus
 - VME::FPGAUnitV1495Control, 63
- SetSignalSource
 - VME::FPGAUnitV1495Control, 63
- SetSocketId
 - Socket, 87
- SetStatus
 - VME::TDCV1x90, 124
- SetTDCBits
 - VME::FPGAUnitV1495, 58
- SetTDCEncapsulation
 - VME::TDCV1x90, 124
- SetTermination
 - VME::TDCV1x90Control, 135
- SetTestFIFO
 - VME::TDCV1x90Control, 135
- SetTestMode
 - VME::TDCV1x90, 125
- SetThreshold
 - VME::CFDV812, 38
- SetTriggerMatching
 - VME::TDCV1x90, 125
- SetTriggerSource
 - VME::FPGAUnitV1495Control, 63
- SetVerboseLevel
 - VME::TDCV1x90, 125
- SetWindowOffset
 - VME::TDCV1x90, 125
- SetWindowWidth
 - VME::TDCV1x90, 127
- SetWord
 - VME::TDCEvent, 102
- SignalSource
 - VME::FPGAUnitV1495Control, 60
- SinglePulse
 - VME::BridgeVx718, 25
- Socket, 82
 - ~Socket, 84
 - AcceptConnections, 84
 - Bind, 84
 - Configure, 85
 - Create, 85
 - DumpConnected, 85
 - fAddress, 87
 - fBuffer, 87
 - fMaster, 87
 - fPort, 87
 - fReadFds, 88
 - fSocketId, 88
 - fSocketsConnected, 88
 - FetchMessage, 85
 - GetPort, 85
 - GetSocketId, 85
 - GetSocketType, 85
 - IsWebSocket, 85
 - Listen, 86
 - PrepareConnection, 86
 - SelectConnections, 86
 - SendMessage, 86
 - SetPort, 87
 - SetSocketId, 87
 - Socket, 84
 - SocketCollection, 84
 - Start, 87
 - Stop, 87
- Socket communication objects, 9
 - CLIENT, 9
 - DETECTOR, 9
 - DQM, 9
 - INVALID, 9
 - MASTER, 9
 - SocketType, 9
 - WEBSOCKET_CLIENT, 9

- SocketCollection
 - Socket, [84](#)
- SocketMessage, [88](#)
 - ~SocketMessage, [92](#)
 - Dump, [92](#)
 - fMessage, [95](#)
 - GetIntValue, [93](#)
 - GetKey, [93](#)
 - GetString, [93](#)
 - GetValue, [93](#)
 - GetVectorValue, [93](#)
 - Object, [93](#)
 - SetKeyValue, [93](#), [94](#)
 - SocketMessage, [90–92](#)
 - String, [94](#)
- SocketType
 - Socket communication objects, [9](#)
- SoftwareClear
 - VME::TDCV1x90, [127](#)
- SoftwareReset
 - VME::TDCV1x90, [127](#)
- spill_id
 - file_header_t, [45](#)
- Start
 - Socket, [87](#)
- StartAcquisition
 - Messenger, [80](#)
- StartPulser
 - VME::BridgeVx718, [25](#)
 - VMEReader, [146](#)
- StartScaler
 - VME::FPGAUnitV1495, [58](#)
- Stop
 - Socket, [87](#)
- StopAcquisition
 - Messenger, [80](#)
- StopPulser
 - VME::BridgeVx718, [25](#)
 - VMEReader, [147](#)
- StopScaler
 - VME::FPGAUnitV1495, [59](#)
- String
 - SocketMessage, [94](#)
- SwitchClientType
 - Messenger, [80](#)
- TDCBits
 - VME::FPGAUnitV1495, [50](#)
- TDCCollection
 - VME, [13](#)
- TDCError
 - VME::TDCEvent, [98](#)
- TDCErrorFlag
 - VME::TDCErrorFlag, [96](#)
- TDCEvent
 - VME::TDCEvent, [98](#)
- TDCEventCollection
 - VME, [13](#)
- TDCHeader
 - VME::TDCEvent, [98](#)
- TDCMeasurement
 - VME::TDCEvent, [98](#)
 - VME::TDCMeasurement, [103](#)
- TDCResolution
 - VME::TDCV1x90Status, [136](#)
- TDCTrailer
 - VME::TDCEvent, [98](#)
- TDCV1x90
 - VME::TDCV1x90, [109](#)
- TDCV1x90Control
 - VME::TDCV1x90Control, [130](#)
- TDCV1x90Register
 - VME, [15](#)
- TDCV1x90Status
 - VME::TDCV1x90Status, [137](#)
- TRAILHEAD
 - VME, [14](#)
- TRG_MATCH
 - VME::TDCV1x90Opcodes, [20](#)
- TRIG_MATCH
 - VME, [13](#)
- TRIG_TIME_SUB
 - VME, [16](#)
- TerminationOn
 - VME::TDCV1x90Status, [139](#)
- TestOutputs
 - VME::BridgeVx718, [25](#)
- total_hits
 - VME::trailead_t, [140](#)
- trailead_edge_lsb
 - VME, [16](#)
- trailing
 - VME::trailead_t, [140](#)
- trig_conf
 - VME, [16](#)
- TriggerLost
 - VME::TDCV1x90Status, [139](#)
- TriggerMatching
 - VME::TDCV1x90Status, [139](#)
- TriggerSource
 - VME::FPGAUnitV1495Control, [61](#)
- UPDATE_SETUP_REG
 - VME::TDCV1x90Opcodes, [20](#)
- UPDATE_SETUP_TDC
 - VME::TDCV1x90Opcodes, [20](#)
- VME, [11](#)
 - AcquisitionMode, [13](#)
 - BridgeType, [13](#)
 - CAEN_V1718, [13](#)
 - CAEN_V2718, [13](#)
 - CAENETControllerV288Register, [13](#)
 - CFDCollection, [13](#)
 - CFDV812Register, [13](#)
 - CONT_STORAGE, [13](#)
 - DetectionMode, [14](#)
 - EXTRA_SEARCH_WIN_WIDTH, [16](#)

FPGAUnitV1495Register, 14
 IOModuleV262Register, 14
 kBLTEventNumber, 15
 kBoardInfo0, 15
 kBoardInfo1, 15
 kControl, 15
 kECLLevelWrite, 15
 kEventCounter, 15
 kEventFIFO, 15
 kEventFIFOStatusRegister, 15
 kEventFIFOStoredRegister, 15
 kEventStored, 15
 kFirmwareRev, 15
 kGeoAddress, 15
 kIdentifier, 15
 kInterruptLevel, 15
 kInterruptVector, 15
 kMCSTBase, 15
 kMCSTControl, 15
 kMicro, 15
 kMicroHandshake, 15
 kModuleReset, 15
 kNIMLevelWrite, 15
 kNIMPulseRead, 15
 kNIMPulseWrite, 15
 kOutputBuffer, 15
 kROMBoard0, 15
 kROMBoard1, 15
 kROMBoard2, 15
 kROMOui0, 15
 kROMOui1, 15
 kROMOui2, 15
 kROMRevis0, 16
 kROMRevis1, 16
 kROMRevis2, 16
 kROMRevis3, 15
 kROMSerNum0, 16
 kROMSerNum1, 16
 kSoftwareClear, 15
 kStatus, 15
 kV1495Board0, 14
 kV1495Board1, 14
 kV1495Board2, 14
 kV1495ClockSettings, 14
 kV1495ConfigurationROM, 14
 kV1495Control, 14
 kV1495DelaySettings, 14
 kV1495FWRevision, 14
 kV1495GeoAddress, 14
 kV1495HWRevision0, 14
 kV1495HWRevision1, 14
 kV1495HWRevision2, 14
 kV1495HWRevision3, 14
 kV1495ModuleReset, 14
 kV1495OUI0, 14
 kV1495OUI1, 14
 kV1495OUI2, 14
 kV1495OutputSettings, 14
 kV1495ScalerCounter, 14
 kV1495SerNum0, 14
 kV1495SerNum1, 14
 kV1495TDCBoardInterface, 14
 kV1495TriggerSettings, 14
 kV1495UserFPGAConfig, 14
 kV1495UserFPGAFlashMem, 14
 kV1495UserFWRevision, 14
 kV288DataBuffer, 13
 kV288IRQVector, 13
 kV288ModuleReset, 13
 kV288Status, 13
 kV288Transmission, 13
 kV812DeadTimeGroup0, 14
 kV812DeadTimeGroup1, 14
 kV812FixedCode, 14
 kV812Info0, 14
 kV812Info1, 14
 kV812MajorityThreshold, 14
 kV812OutputWidthGroup0, 13
 kV812OutputWidthGroup1, 13
 kV812PatternOfInhibit, 14
 kV812TestPulse, 14
 kV812ThresholdChannel0, 13
 MATCH_WIN_WIDTH, 16
 micro_handshake, 15
 OLEADING, 14
 OTRAILING, 14
 PAIR, 14
 r100ps, 16
 r200ps, 16
 r25ps, 16
 r800ps, 16
 READ_OK, 15
 REJECT_MARGIN, 16
 TDCCollection, 13
 TDCEventCollection, 13
 TDCV1x90Register, 15
 TRAILEAD, 14
 TRIG_MATCH, 13
 TRIG_TIME_SUB, 16
 trailead_edge_lsb, 16
 trig_conf, 16
 WIN_OFFSET, 16
 WRITE_OK, 15
 VME::BridgeVx718, 21
 ~BridgeVx718, 24
 BridgeVx718, 23
 CheckConfiguration, 24
 CheckPCIInterface, 24
 fHasIRQ, 25
 GetHandle, 24
 GetIRQStatus, 24
 GetStatus, 24
 IRQ1, 23
 IRQ2, 23
 IRQ3, 23
 IRQ4, 23

- IRQ5, [23](#)
- IRQ6, [23](#)
- IRQ7, [23](#)
- IRQId, [23](#)
- InputConf, [24](#)
- InputRead, [24](#)
- OutputConf, [25](#)
- OutputOff, [25](#)
- OutputOn, [25](#)
- Reset, [25](#)
- SetIRQ, [25](#)
- SinglePulse, [25](#)
- StartPulser, [25](#)
- StopPulser, [25](#)
- TestOutputs, [25](#)
- WaitIRQ, [25](#)
- VME::BridgeVx718Control, [26](#)
 - ~BridgeVx718Control, [26](#)
 - BridgeVx718Control, [26](#)
 - fWord, [27](#)
 - GetAddressIncrement, [26](#)
 - GetArbiterType, [26](#)
 - GetBusReqLevel, [26](#)
 - GetBusTimeout, [27](#)
 - GetInterruptReq, [27](#)
 - GetReleaseType, [27](#)
 - GetRequesterType, [27](#)
 - GetSysRes, [27](#)
- VME::BridgeVx718Status, [27](#)
 - ~BridgeVx718Status, [28](#)
 - BridgeVx718Status, [28](#)
 - Dump, [28](#)
 - fWord, [29](#)
 - GetBERR, [28](#)
 - GetDTACK, [28](#)
 - GetDipSwitch, [28](#)
 - GetSystemControl, [28](#)
 - GetSystemReset, [28](#)
 - GetUSBType, [28](#)
- VME::CAENETControllerV288, [29](#)
 - ~CAENETControllerV288, [30](#)
 - CAENETControllerV288, [30](#)
 - fNumWordsInBuffer, [31](#)
 - GetStatus, [30](#)
 - operator<<, [31](#)
 - operator>>, [31](#)
 - Receive, [30](#)
 - Send, [31](#)
 - WaitForResponse, [31](#)
- VME::CAENETControllerV288Status, [31](#)
 - ~CAENETControllerV288Status, [32](#)
 - CAENETControllerV288Status, [32](#)
 - fWord, [32](#)
 - GetOperationStatus, [32](#)
 - Invalid, [32](#)
 - OperationStatus, [32](#)
 - Valid, [32](#)
- VME::CFDV812, [32](#)
 - ~CFDV812, [34](#)
 - CFDV812, [34](#)
 - CheckConfiguration, [34](#)
 - DeadTimeCalculator, [34](#)
 - GetFixedCode, [34](#)
 - GetManufacturerId, [35](#)
 - GetModuleType, [35](#)
 - GetModuleVersion, [35](#)
 - GetSerialNumber, [36](#)
 - OutputWidthCalculator, [36](#)
 - SetDeadTime, [36](#)
 - SetOutputWidth, [36](#)
 - SetPOI, [38](#)
 - SetThreshold, [38](#)
- VME::FPGAUnitV1495, [48](#)
 - ~FPGAUnitV1495, [50](#)
 - CheckBoardVersion, [50](#)
 - ClearOutputPulser, [51](#)
 - DumpFWInformation, [51](#)
 - FPGAUnitV1495, [50](#)
 - fScalerStarted, [59](#)
 - GetCAENFirmwareRevision, [51](#)
 - GetControl, [52](#)
 - GetGeoAddress, [52](#)
 - GetHardwareRevision, [52](#)
 - GetInternalClockPeriod, [53](#)
 - GetInternalTriggerPeriod, [53](#)
 - GetOutputDelay, [53](#)
 - GetOutputPulser, [54](#)
 - GetScalerValue, [54](#)
 - GetSerialNumber, [54](#)
 - GetTDCBits, [55](#)
 - GetUserFirmwareRevision, [55](#)
 - kClear, [50](#)
 - kReset, [50](#)
 - kTrigger, [50](#)
 - PulseTDCBits, [55](#)
 - ResetFPGA, [56](#)
 - SetControl, [56](#)
 - SetInternalClockPeriod, [56](#)
 - SetInternalTriggerPeriod, [57](#)
 - SetOutputDelay, [57](#)
 - SetOutputPulser, [57](#)
 - SetOutputPulserPOI, [58](#)
 - SetTDCBits, [58](#)
 - StartScaler, [58](#)
 - StopScaler, [59](#)
 - TDCBits, [50](#)
- VME::FPGAUnitV1495Control, [59](#)
 - ~FPGAUnitV1495Control, [61](#)
 - ClockSource, [60](#)
 - Dump, [61](#)
 - ExternalClock, [60](#)
 - ExternalSignal, [61](#)
 - ExternalTrigger, [61](#)
 - FPGAUnitV1495Control, [61](#)
 - fWord, [64](#)
 - GetBit, [61](#)

- GetClockSource, 61
- GetScalerStatus, 61
- GetSignalSource, 61
- GetTriggerSource, 62
- GetWord, 62
- InternalClock, 60
- InternalSignal, 61
- InternalTrigger, 61
- SetBit, 62
- SetClockSource, 62
- SetScalerReset, 63
- SetScalerStatus, 63
- SetSignalSource, 63
- SetTriggerSource, 63
- SignalSource, 60
- TriggerSource, 61
- VME::GenericBoard
 - ~GenericBoard, 64
 - fBaseAddr, 65
 - fHandle, 65
 - GenericBoard, 64
 - ReadRegister, 65
 - WriteRegister, 65
- VME::GenericBoard< Register, am >, 64
- VME::GlobalOffset, 66
 - coarse, 66
 - fine, 66
- VME::IOModuleV262, 69
 - ~IOModuleV262, 70
 - GetIdentifier, 70
 - GetManufacturerId, 70
 - GetModuleType, 71
 - GetModuleVersion, 71
 - GetSerialNumber, 71
 - IOModuleV262, 70
- VME::PCIInterfaceA2818, 81
 - ~PCIInterfaceA2818, 81
 - fHandle, 82
 - GetFWRevision, 81
 - PCIInterfaceA2818, 81
- VME::TDCErrorFlag, 95
 - ~TDCErrorFlag, 96
 - Dump, 96
 - fWord, 96
 - GetWord, 96
 - HasGroupError, 96
 - HasInternalChipError, 96
 - HasL1BufferOverflow, 96
 - HasReachedEventSizeLimit, 96
 - HasReadoutFIFOOverflow, 96
 - HasTriggerFIFOOverflow, 96
 - operator<<, 96
 - TDCErrorFlag, 96
- VME::TDCEvent, 97
 - ~TDCEvent, 98
 - Dump, 98
 - ETTT, 98
 - EventType, 98
- fWord, 102
- Filler, 98
- GetBunchId, 98
- GetChannelId, 99
- GetETTT, 99
- GetErrorFlags, 99
- GetEventCount, 100
- GetEventId, 100
- GetGeo, 100
- GetStatus, 101
- GetTDCId, 101
- GetTime, 101
- GetType, 101
- GetWidth, 101
- GetWord, 102
- GetWordCount, 102
- GlobalHeader, 98
- GlobalTrailer, 98
- IsTrailing, 102
- SetWord, 102
- TDCError, 98
- TDCEvent, 98
- TDCHeader, 98
- TDCMeasurement, 98
- TDCTrailer, 98
- VME::TDCMeasurement, 103
 - ~TDCMeasurement, 104
 - Dump, 104
 - fEvents, 106
 - fMap, 106
 - GetBunchId, 105
 - GetChannelId, 105
 - GetETTT, 105
 - GetEventId, 105
 - GetLeadingTime, 105
 - GetTDCId, 105
 - GetToT, 105
 - GetTrailingTime, 106
 - NumErrors, 106
 - NumEvents, 106
 - SetEventsCollection, 106
 - TDCMeasurement, 103
- VME::TDCV1x90, 106
 - ~TDCV1x90, 110
 - abort, 110
 - CheckConfiguration, 110
 - DLL_Direct_LowRes, 109
 - DLL_PLL_HighRes, 109
 - DLL_PLL_LowRes, 109
 - DLL_PLL_MedRes, 109
 - DLLMode, 109
 - DisableChannel, 110
 - EnableChannel, 111
 - fAcquisitionMode, 128
 - fBuffer, 128
 - fDetectionMode, 128
 - fErrorMarks, 128
 - fVerb, 128

- fWindowWidth, 128
- FetchEvents, 111
- gEnd, 128
- GetAcquisitionMode, 111
- GetBLTEventNumberRegister, 112
- GetChannelDeadTime, 112
- GetControl, 112
- GetDLLClock, 113
- GetDetectionMode, 113
- GetETTT, 113
- GetErrorMarks, 113
- GetEventCounter, 113
- GetEventStored, 114
- GetFIFOSize, 114
- GetFirmwareRevision, 114
- GetGlobalOffset, 115
- GetModel, 115
- GetOUI, 115
- GetPol, 116
- GetRCAdjust, 116
- GetResolution, 116
- GetSerialNumber, 117
- GetStatus, 117
- GetTDCEncapsulation, 117
- GetTestMode, 118
- GetTriggerConfiguration, 118
- GetWindowOffset, 118
- GetWindowWidth, 118
- HardwareReset, 118
- nchannels, 128
- pair_lead_res, 128
- pair_width_res, 128
- ReadAcquisitionMode, 118
- ReadDetectionMode, 119
- SetAcquisitionMode, 119
- SetBLTEventNumberRegister, 119
- SetChannelDeadTime, 120
- SetContinuousStorage, 120
- SetControl, 120
- SetDLLClock, 121
- SetDetectionMode, 121
- SetETTT, 122
- SetErrorMarks, 121
- SetFIFOSize, 122
- SetGlobalOffset, 122
- SetLSBTraileadEdge, 123
- SetPairModeResolution, 123
- SetPol, 123
- SetRCAdjust, 124
- SetStatus, 124
- SetTDCEncapsulation, 124
- SetTestMode, 125
- SetTriggerMatching, 125
- SetVerboseLevel, 125
- SetWindowOffset, 125
- SetWindowWidth, 127
- SoftwareClear, 127
- SoftwareReset, 127
- TDCV1x90, 109
- WaitMicro, 128
- VME::TDCV1x90Control, 129
 - ~TDCV1x90Control, 130
 - Dump, 130
 - fWord, 135
 - GetAlign64, 131
 - GetBusError, 131
 - GetCompensation, 131
 - GetETTT, 132
 - GetEmptyEvent, 132
 - GetEventFIFO, 132
 - GetMEBAccess, 132
 - GetSRAMCompensation, 132
 - GetSWTermination, 132
 - GetTermination, 132
 - GetTestFIFO, 132
 - GetValue, 132
 - SetAlign64, 132
 - SetBusError, 132
 - SetCompensation, 132
 - SetETTT, 133
 - SetEmptyEvent, 133
 - SetEventFIFO, 133
 - SetMEBAccess, 134
 - SetSRAMCompensation, 134
 - SetSWTermination, 134
 - SetTermination, 135
 - SetTestFIFO, 135
 - TDCV1x90Control, 130
- VME::TDCV1x90Opcodes, 16
 - AUTOLOAD_DEF_CONF1, 18
 - AUTOLOAD_USER_CONF, 18
 - CLEAR_KEEP_TOKEN, 18
 - CONT_STOR, 18
 - DEFAULT_SETUP_REG, 18
 - DIS_ALL_CHANNEL, 18
 - DIS_CHANNEL, 18
 - DIS_ERROR_BYPASS, 18
 - DIS_ERROR_MARK, 18
 - DIS_HEAD_TRAILER, 18
 - DIS_SUB_TRG, 18
 - DISABLE_TEST_MODE, 18
 - EN_ALL_CHANNEL, 18
 - EN_CHANNEL, 18
 - EN_ERROR_BYPASS, 18
 - EN_ERROR_MARK, 18
 - EN_HEAD_TRAILER, 18
 - EN_SUB_TRG, 18
 - ENABLE_TEST_MODE, 18
 - LOAD_DEF_CONFIG, 18
 - LOAD_USER_CONFIG, 18
 - READ_ACQ_MOD, 18
 - READ_ADJUST_CH, 18
 - READ_DEAD_TIME, 18
 - READ_DETECTION, 18
 - READ_DLL_LOCK, 18
 - READ_EEPROM, 18

- READ_EN_PATTERN, 18
- READ_EN_PATTERN32, 19
- READ_ERROR_STATUS, 19
- READ_ERROR_TYPES, 19
- READ_EVENT_SIZE, 19
- READ_FIFO_SIZE, 19
- READ_GLOB_OFFS, 19
- READ_HEAD_TRAILER, 19
- READ_MICRO_REV, 19
- READ_RC_ADJ, 19
- READ_RES, 19
- READ_SETUP_REG, 19
- READ_SETUP_SCANPATH, 19
- READ_SPARE, 19
- READ_STATUS_STREAM, 19
- READ_TDC_ID, 19
- READ_TRG_CONF, 19
- RESET_DLL_PLL, 19
- REV_DATE_MICRO_FW, 19
- SAVE_RC_ADJ, 19
- SAVE_USER_CONFIG, 19
- SET_ADJUST_CH, 19
- SET_DEAD_TIME, 19
- SET_DETECTION, 19
- SET_DLL_CLOCK, 19
- SET_ERROR_TYPES, 19
- SET_EVENT_SIZE, 19
- SET_FIFO_SIZE, 19
- SET_GLOB_OFFS, 19
- SET_KEEP_TOKEN, 20
- SET_PAIR_RES, 20
- SET_RC_ADJ, 20
- SET_REJ_MARGIN, 20
- SET_SW_MARGIN, 20
- SET_TDC_TSET_OUTPUT, 20
- SET_TR_LEAD_LSB, 20
- SET_WIN_OFFS, 20
- SET_WIN_WIDTH, 20
- TRG_MATCH, 20
- UPDATE_SETUP_REG, 20
- UPDATE_SETUP_TDC, 20
- WRITE_EEPROM, 20
- WRITE_EN_PATTERN, 20
- WRITE_EN_PATTERN32, 20
- WRITE_SETUP_REG, 20
- WRITE_SPARE, 20
- VME::TDCV1x90Status, 136
 - ~TDCV1x90Status, 137
 - AlmostFull, 137
 - BusError, 137
 - DataReady, 137
 - Dump, 137
 - Error, 138
 - fWord, 139
 - Full, 139
 - GetValue, 139
 - HeadersEnabled, 139
 - PairMode, 139
 - Purged, 139
 - R_100ps, 136
 - R_200ps, 136
 - R_25ps, 136
 - R_800ps, 136
 - Resolution, 139
 - TDCResolution, 136
 - TDCV1x90Status, 137
 - TerminationOn, 139
 - TriggerLost, 139
 - TriggerMatching, 139
- VME::trailead_t, 139
 - ettt, 140
 - event_count, 140
 - leading, 140
 - total_hits, 140
 - trailing, 140
- VMEReader, 140
 - ~VMEReader, 143
 - Abort, 143
 - AddCFD, 143
 - AddFPGAUnit, 144
 - AddIOModule, 144
 - AddTDC, 144
 - fBridge, 147
 - fCFDCollection, 147
 - fFPGA, 147
 - fIsPulserStarted, 147
 - fOnSocket, 147
 - fOutputFile, 147
 - fSG, 147
 - fTDCCollection, 148
 - GetCFD, 145
 - GetCFDCollection, 145
 - GetFPGAUnit, 145
 - GetIOModule, 145
 - GetNumCFD, 145
 - GetNumTDC, 145
 - GetOutputFile, 145
 - GetRunNumber, 145
 - GetTDC, 145
 - GetTDCCollection, 145
 - SendClear, 146
 - SendPulse, 146
 - SetOutputFile, 146
 - StartPulser, 146
 - StopPulser, 147
 - VMEReader, 142
- Valid
 - VME::CAENETControllerV288Status, 32
- WEBSOCKET_CLIENT
 - Socket communication objects, 9
- WIN_OFFSET
 - VME, 16
- WRITE_EEPROM
 - VME::TDCV1x90Opcodes, 20
- WRITE_EN_PATTERN
 - VME::TDCV1x90Opcodes, 20

WRITE_EN_PATTERN32
 VME::TDCV1x90Opcodes, [20](#)
WRITE_OK
 VME, [15](#)
WRITE_SETUP_REG
 VME::TDCV1x90Opcodes, [20](#)
WRITE_SPARE
 VME::TDCV1x90Opcodes, [20](#)
WaitForResponse
 VME::CAENETControllerV288, [31](#)
WaitIRQ
 VME::BridgeVx718, [25](#)
WaitMicro
 VME::TDCV1x90, [128](#)
WriteRegister
 VME::GenericBoard, [65](#)