

2015 Test beam Run Control

Generated by Doxygen 1.8.9.1

Tue Sep 15 2015 17:19:15

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	DQM Namespace Reference	11
6.2	NIM Namespace Reference	11
6.2.1	Enumeration Type Documentation	11
6.2.1.1	HVModuleN470Opcodes	12
6.3	VME Namespace Reference	12
6.3.1	Typedef Documentation	14
6.3.1.1	CFDCollection	14
6.3.1.2	FPGAUnitCollection	14
6.3.1.3	TDCCollection	14
6.3.1.4	TDCEventCollection	14
6.3.2	Enumeration Type Documentation	14
6.3.2.1	AcquisitionMode	14
6.3.2.2	BridgeType	15
6.3.2.3	CAENETControllerV288Answer	15

6.3.2.4	CAENETControllerV288Register	15
6.3.2.5	CFDV812Register	15
6.3.2.6	DetectionMode	16
6.3.2.7	FPGAUnitV1495DACCH	16
6.3.2.8	FPGAUnitV1495Register	16
6.3.2.9	IOModuleV262Register	17
6.3.2.10	micro_handshake	17
6.3.2.11	TDCV1x90Register	17
6.3.2.12	trailead_edge_lsb	18
6.3.2.13	trig_conf	18
6.3.3	Variable Documentation	18
6.3.3.1	OneVolt	18
6.4	VME::TDCV1x90Opcodes Namespace Reference	18
6.4.1	Function Documentation	20
6.4.1.1	AUTOLOAD_DEF_CONF1	20
6.4.1.2	AUTOLOAD_USER_CONF	20
6.4.1.3	CLEAR_KEEP_TOKEN	20
6.4.1.4	CONT_STOR	20
6.4.1.5	DEFAULT_SETUP_REG	20
6.4.1.6	DIS_ALL_CHANNEL	20
6.4.1.7	DIS_CHANNEL	20
6.4.1.8	DIS_ERROR_BYPASS	20
6.4.1.9	DIS_ERROR_MARK	20
6.4.1.10	DIS_HEAD_TRAILER	20
6.4.1.11	DIS_SUB_TRG	20
6.4.1.12	DISABLE_TEST_MODE	20
6.4.1.13	EN_ALL_CHANNEL	20
6.4.1.14	EN_CHANNEL	20
6.4.1.15	EN_ERROR_BYPASS	20
6.4.1.16	EN_ERROR_MARK	20
6.4.1.17	EN_HEAD_TRAILER	20
6.4.1.18	EN_SUB_TRG	20
6.4.1.19	ENABLE_TEST_MODE	20
6.4.1.20	LOAD_DEF_CONFIG	20
6.4.1.21	LOAD_USER_CONFIG	20
6.4.1.22	READ_ACQ_MOD	20
6.4.1.23	READ_ADJUST_CH	20
6.4.1.24	READ_DEAD_TIME	20
6.4.1.25	READ_DETECTION	20
6.4.1.26	READ_DLL_LOCK	20

6.4.1.27	READ_EEPROM	20
6.4.1.28	READ_EN_PATTERN	21
6.4.1.29	READ_EN_PATTERN32	21
6.4.1.30	READ_ERROR_STATUS	21
6.4.1.31	READ_ERROR_TYPES	21
6.4.1.32	READ_EVENT_SIZE	21
6.4.1.33	READ_FIFO_SIZE	21
6.4.1.34	READ_GLOB_OFFS	21
6.4.1.35	READ_HEAD_TRAILER	21
6.4.1.36	READ_MICRO_REV	21
6.4.1.37	READ_RC_ADJ	21
6.4.1.38	READ_RES	21
6.4.1.39	READ_SETUP_REG	21
6.4.1.40	READ_SETUP_SCANPATH	21
6.4.1.41	READ_SPARE	21
6.4.1.42	READ_STATUS_STREAM	21
6.4.1.43	READ_TDC_ID	21
6.4.1.44	READ_TRG_CONF	21
6.4.1.45	RESET_DLL_PLL	21
6.4.1.46	REV_DATE_MICRO_FW	21
6.4.1.47	SAVE_RC_ADJ	21
6.4.1.48	SAVE_USER_CONFIG	21
6.4.1.49	SET_ADJUST_CH	21
6.4.1.50	SET_DEAD_TIME	21
6.4.1.51	SET_DETECTION	21
6.4.1.52	SET_DLL_CLOCK	21
6.4.1.53	SET_ERROR_TYPES	21
6.4.1.54	SET_EVENT_SIZE	21
6.4.1.55	SET_FIFO_SIZE	21
6.4.1.56	SET_GLOB_OFFS	22
6.4.1.57	SET_KEEP_TOKEN	22
6.4.1.58	SET_PAIR_RES	22
6.4.1.59	SET_RC_ADJ	22
6.4.1.60	SET_REJ_MARGIN	22
6.4.1.61	SET_SW_MARGIN	22
6.4.1.62	SET_TDC_TSET_OUTPUT	22
6.4.1.63	SET_TR_LEAD_LSB	22
6.4.1.64	SET_WIN_OFFS	22
6.4.1.65	SET_WIN_WIDTH	22
6.4.1.66	TRG_MATCH	22

6.4.1.67	UPDATE_SETUP_REG	22
6.4.1.68	UPDATE_SETUP_TDC	22
6.4.1.69	WRITE_EEPROM	22
6.4.1.70	WRITE_EN_PATTERN	22
6.4.1.71	WRITE_EN_PATTERN32	22
6.4.1.72	WRITE_SETUP_REG	22
6.4.1.73	WRITE_SPARE	22
7	Data Structure Documentation	23
7.1	VME::BridgeVx718 Class Reference	23
7.1.1	Detailed Description	25
7.1.2	Member Enumeration Documentation	25
7.1.2.1	IRQId	25
7.1.3	Constructor & Destructor Documentation	25
7.1.3.1	BridgeVx718	25
7.1.3.2	~BridgeVx718	26
7.1.4	Member Function Documentation	26
7.1.4.1	CheckConfiguration	26
7.1.4.2	CheckPCIInterface	26
7.1.4.3	GetHandle	26
7.1.4.4	GetIRQStatus	26
7.1.4.5	GetStatus	26
7.1.4.6	InputConf	26
7.1.4.7	InputRead	26
7.1.4.8	OutputConf	26
7.1.4.9	OutputOff	26
7.1.4.10	OutputOn	26
7.1.4.11	Reset	26
7.1.4.12	SetIRQ	27
7.1.4.13	SinglePulse	27
7.1.4.14	StartPulser	27
7.1.4.15	StopPulser	27
7.1.4.16	TestOutputs	27
7.1.4.17	WaitIRQ	27
7.1.5	Field Documentation	27
7.1.5.1	fHasIRQ	27
7.2	VME::BridgeVx718Control Class Reference	27
7.2.1	Constructor & Destructor Documentation	28
7.2.1.1	BridgeVx718Control	28
7.2.1.2	~BridgeVx718Control	28

7.2.2	Member Function Documentation	28
7.2.2.1	GetAddressIncrement	28
7.2.2.2	GetArbiterType	28
7.2.2.3	GetBusReqLevel	28
7.2.2.4	GetBusTimeout	28
7.2.2.5	GetInterruptReq	29
7.2.2.6	GetReleaseType	29
7.2.2.7	GetRequesterType	29
7.2.2.8	GetSysRes	29
7.2.3	Field Documentation	29
7.2.3.1	fWord	29
7.3	VME::BridgeVx718Status Class Reference	29
7.3.1	Constructor & Destructor Documentation	30
7.3.1.1	BridgeVx718Status	30
7.3.1.2	~BridgeVx718Status	30
7.3.2	Member Function Documentation	30
7.3.2.1	Dump	30
7.3.2.2	GetBERR	30
7.3.2.3	GetDipSwitch	30
7.3.2.4	GetDTACK	30
7.3.2.5	GetSystemControl	30
7.3.2.6	GetSystemReset	30
7.3.2.7	GetUSBType	30
7.3.3	Field Documentation	30
7.3.3.1	fWord	30
7.4	OnlineDBHandler::BurstInfo Struct Reference	31
7.4.1	Field Documentation	31
7.4.1.1	burst_id	31
7.4.1.2	time_start	31
7.5	VME::CAENETControllerV288 Class Reference	31
7.5.1	Detailed Description	32
7.5.2	Constructor & Destructor Documentation	33
7.5.2.1	CAENETControllerV288	33
7.5.2.2	~CAENETControllerV288	33
7.5.3	Member Function Documentation	33
7.5.3.1	FetchBuffer	33
7.5.3.2	GetStatus	33
7.5.3.3	Reset	33
7.5.3.4	SendBuffer	33
7.5.3.5	WaitForResponse	34

7.5.4	Friends And Related Function Documentation	34
7.5.4.1	operator<<	34
7.5.4.2	operator>>	34
7.6	VME::CAENETControllerV288Status Class Reference	34
7.6.1	Member Enumeration Documentation	34
7.6.1.1	OperationStatus	34
7.6.2	Constructor & Destructor Documentation	35
7.6.2.1	CAENETControllerV288Status	35
7.6.2.2	~CAENETControllerV288Status	35
7.6.3	Member Function Documentation	35
7.6.3.1	GetOperationStatus	35
7.6.4	Field Documentation	35
7.6.4.1	fWord	35
7.7	VME::CFDV812 Class Reference	35
7.7.1	Detailed Description	36
7.7.2	Constructor & Destructor Documentation	37
7.7.2.1	CFDV812	37
7.7.2.2	~CFDV812	37
7.7.3	Member Function Documentation	37
7.7.3.1	CheckConfiguration	37
7.7.3.2	DeadTimeCalculator	37
7.7.3.3	GetFixedCode	37
7.7.3.4	GetManufacturerId	38
7.7.3.5	GetModuleType	38
7.7.3.6	GetModuleVersion	38
7.7.3.7	GetSerialNumber	39
7.7.3.8	OutputWidthCalculator	39
7.7.3.9	SetDeadTime	39
7.7.3.10	SetOutputWidth	39
7.7.3.11	SetPOI	40
7.7.3.12	SetThreshold	40
7.8	NIM::HVModuleN470Values::ChannelStatus Class Reference	41
7.8.1	Member Enumeration Documentation	41
7.8.1.1	SignalStandard	41
7.8.2	Constructor & Destructor Documentation	42
7.8.2.1	ChannelStatus	42
7.8.2.2	~ChannelStatus	42
7.8.3	Member Function Documentation	42
7.8.3.1	Alarm	42
7.8.3.2	Dump	42

7.8.3.3	Enabled	42
7.8.3.4	HVEnabled	42
7.8.3.5	Isel	42
7.8.3.6	Kill	42
7.8.3.7	MaxV	42
7.8.3.8	NonCalibrated	42
7.8.3.9	OVC	42
7.8.3.10	OVV	42
7.8.3.11	Polarity	42
7.8.3.12	RampDown	42
7.8.3.13	RampUp	42
7.8.3.14	Standard	42
7.8.3.15	Trip	42
7.8.3.16	UNV	42
7.8.3.17	Vsel	42
7.8.4	Friends And Related Function Documentation	42
7.8.4.1	operator<<	42
7.8.5	Field Documentation	42
7.8.5.1	fWord	42
7.9	Client Class Reference	43
7.9.1	Detailed Description	44
7.9.2	Constructor & Destructor Documentation	44
7.9.2.1	Client	44
7.9.2.2	Client	44
7.9.2.3	~Client	45
7.9.3	Member Function Documentation	45
7.9.3.1	Announce	45
7.9.3.2	Connect	45
7.9.3.3	Disconnect	46
7.9.3.4	GetType	46
7.9.3.5	ParseMessage	46
7.9.3.6	Receive	46
7.9.3.7	Receive	47
7.9.3.8	Send	47
7.9.3.9	Send	48
7.9.3.10	SendAndReceive	48
7.9.4	Field Documentation	48
7.9.4.1	fClientId	48
7.9.4.2	fIsConnected	48
7.9.4.3	fType	48

7.10	DQM::GastofCanvas::Coord Struct Reference	48
7.10.1	Field Documentation	48
7.10.1.1	x	49
7.10.1.2	y	49
7.11	DQM::QuarticCanvas::Coord Struct Reference	49
7.11.1	Field Documentation	49
7.11.1.1	x	49
7.11.1.2	y	49
7.12	DQM::DQMProcess Class Reference	49
7.12.1	Detailed Description	50
7.12.2	Member Enumeration Documentation	51
7.12.2.1	Action	51
7.12.3	Constructor & Destructor Documentation	51
7.12.3.1	DQMProcess	51
7.12.3.2	~DQMProcess	51
7.12.4	Member Function Documentation	51
7.12.4.1	IsInRun	52
7.12.4.2	ParseMessage	52
7.12.4.3	Run	52
7.12.4.4	Run	53
7.12.5	Field Documentation	53
7.12.5.1	fAddressesCanProcess	53
7.12.5.2	fDetectorType	53
7.12.5.3	fOrder	53
7.12.5.4	fRunNumber	53
7.13	file_header_t Struct Reference	53
7.13.1	Detailed Description	53
7.13.2	Field Documentation	54
7.13.2.1	acq_mode	54
7.13.2.2	det_mode	54
7.13.2.3	magic	54
7.13.2.4	num_hptdc	54
7.13.2.5	run_id	54
7.13.2.6	spill_id	54
7.14	FileReader Class Reference	54
7.14.1	Detailed Description	55
7.14.2	Constructor & Destructor Documentation	55
7.14.2.1	FileReader	55
7.14.2.2	FileReader	55
7.14.2.3	~FileReader	56

7.14.3	Member Function Documentation	56
7.14.3.1	Clear	56
7.14.3.2	Dump	56
7.14.3.3	GetAcquisitionMode	56
7.14.3.4	GetBurstId	56
7.14.3.5	GetDetectionMode	56
7.14.3.6	GetNextEvent	56
7.14.3.7	GetNextMeasurement	56
7.14.3.8	GetNumEvents	57
7.14.3.9	GetNumTDCs	57
7.14.3.10	GetRunId	57
7.14.3.11	IsOpen	57
7.14.3.12	Open	57
7.14.4	Field Documentation	57
7.14.4.1	fFile	57
7.14.4.2	fHeader	57
7.14.4.3	fNumEvents	57
7.14.4.4	fReadoutMode	57
7.14.4.5	fWriteTime	57
7.15	VME::FPGAUnitV1495 Class Reference	57
7.15.1	Detailed Description	59
7.15.2	Member Enumeration Documentation	59
7.15.2.1	TDCBits	59
7.15.3	Constructor & Destructor Documentation	60
7.15.3.1	FPGAUnitV1495	60
7.15.3.2	~FPGAUnitV1495	60
7.15.4	Member Function Documentation	60
7.15.4.1	CheckBoardVersion	60
7.15.4.2	ClearOutputPulser	61
7.15.4.3	DumpFWInformation	61
7.15.4.4	GetCAENFirmwareRevision	61
7.15.4.5	GetControl	62
7.15.4.6	GetGeoAddress	62
7.15.4.7	GetHardwareRevision	62
7.15.4.8	GetInternalClockPeriod	62
7.15.4.9	GetInternalTriggerPeriod	63
7.15.4.10	GetOutputDelay	63
7.15.4.11	GetOutputPulser	64
7.15.4.12	GetScalerValue	64
7.15.4.13	GetSerialNumber	64

7.15.4.14	GetTDCBits	64
7.15.4.15	GetThresholdVoltage	65
7.15.4.16	GetUserFirmwareRevision	65
7.15.4.17	IsTDCControlFanout	65
7.15.4.18	PulseTDCBits	66
7.15.4.19	ResetFPGA	66
7.15.4.20	SetControl	66
7.15.4.21	SetInternalClockPeriod	67
7.15.4.22	SetInternalTriggerPeriod	68
7.15.4.23	SetOutputDelay	68
7.15.4.24	SetOutputPulser	69
7.15.4.25	SetOutputPulserPOI	69
7.15.4.26	SetTDCBits	69
7.15.4.27	SetTDCControlFanout	69
7.15.4.28	SetThresholdVoltage	69
7.15.4.29	StartScaler	70
7.15.4.30	StopScaler	70
7.15.5	Field Documentation	71
7.15.5.1	fIsTDCControlFanout	71
7.15.5.2	fScalerStarted	71
7.16	VME::FPGAUnitV1495Control Class Reference	71
7.16.1	Detailed Description	72
7.16.2	Member Enumeration Documentation	72
7.16.2.1	ClockSource	72
7.16.2.2	SignalSource	72
7.16.2.3	TriggeringMode	72
7.16.2.4	TriggerSource	73
7.16.3	Constructor & Destructor Documentation	73
7.16.3.1	FPGAUnitV1495Control	73
7.16.3.2	~FPGAUnitV1495Control	73
7.16.4	Member Function Documentation	73
7.16.4.1	Dump	73
7.16.4.2	GetBit	73
7.16.4.3	GetClockSource	73
7.16.4.4	GetScalerStatus	73
7.16.4.5	GetSignalSource	74
7.16.4.6	GetTriggeringMode	74
7.16.4.7	GetTriggerSource	74
7.16.4.8	GetWord	74
7.16.4.9	SetBit	75

7.16.4.10 SetClockSource	75
7.16.4.11 SetScalerReset	75
7.16.4.12 SetScalerStatus	75
7.16.4.13 SetSignalSource	76
7.16.4.14 SetTriggeringMode	76
7.16.4.15 SetTriggerSource	76
7.16.5 Field Documentation	76
7.16.5.1 fWord	76
7.17 DQM::GastofCanvas Class Reference	76
7.17.1 Detailed Description	78
7.17.2 Constructor & Destructor Documentation	78
7.17.2.1 GastofCanvas	78
7.17.2.2 GastofCanvas	78
7.17.2.3 GastofCanvas	79
7.17.2.4 ~GastofCanvas	79
7.17.3 Member Function Documentation	79
7.17.3.1 Build	79
7.17.3.2 DrawGrid	79
7.17.3.3 FillChannel	79
7.17.3.4 GetCoordinates	79
7.17.3.5 Grid	79
7.17.3.6 Save	80
7.17.3.7 SetRunInfo	80
7.17.3.8 SetUpperLabel	80
7.17.4 Field Documentation	80
7.17.4.1 c1	80
7.17.4.2 c2	80
7.17.4.3 fBoardId	80
7.17.4.4 fHeight	80
7.17.4.5 fHist	80
7.17.4.6 fLabel1	80
7.17.4.7 fLabel2	80
7.17.4.8 fLabel3	80
7.17.4.9 fLabel4	80
7.17.4.10 fLabelsDrawn	80
7.17.4.11 fLegend	80
7.17.4.12 fLegendNumEntries	80
7.17.4.13 fLegendX	80
7.17.4.14 fLegendY	80
7.17.4.15 fRunDate	80

7.17.4.16 fRunId	81
7.17.4.17 fSpillId	81
7.17.4.18 fUpperLabel	81
7.17.4.19 fUpperLabelText	81
7.17.4.20 fWidth	81
7.18 VME::GenericBoard< Register, am > Class Template Reference	81
7.18.1 Constructor & Destructor Documentation	81
7.18.1.1 GenericBoard	81
7.18.1.2 ~GenericBoard	81
7.18.2 Member Function Documentation	81
7.18.2.1 ReadRegister	82
7.18.2.2 ReadRegister	82
7.18.2.3 WriteRegister	82
7.18.2.4 WriteRegister	82
7.18.3 Field Documentation	82
7.18.3.1 fBaseAddr	82
7.18.3.2 fHandle	82
7.19 VME::GlobalOffset Struct Reference	83
7.19.1 Field Documentation	83
7.19.1.1 coarse	83
7.19.1.2 fine	83
7.20 NIM::HVModuleN470 Class Reference	83
7.20.1 Constructor & Destructor Documentation	84
7.20.1.1 HVModuleN470	84
7.20.1.2 ~HVModuleN470	84
7.20.2 Member Function Documentation	84
7.20.2.1 DisableChannel	84
7.20.2.2 EnableChannel	85
7.20.2.3 GetFWRevision	85
7.20.2.4 GetModuleId	85
7.20.2.5 ReadChannelValues	85
7.20.2.6 ReadMonitoringValues	85
7.20.2.7 ReadRegister	85
7.20.2.8 SetChannelI0	86
7.20.2.9 SetChannelI1	86
7.20.2.10 SetChannelV0	86
7.20.2.11 SetChannelV1	87
7.20.2.12 WriteRegister	87
7.20.2.13 WriteRegister	87
7.20.3 Field Documentation	87

7.20.3.1	fAddress	87
7.20.3.2	fController	88
7.21	NIM::HVModuleN470ChannelValues Class Reference	88
7.21.1	Detailed Description	88
7.21.2	Constructor & Destructor Documentation	89
7.21.2.1	HVModuleN470ChannelValues	89
7.21.2.2	~HVModuleN470ChannelValues	89
7.21.3	Member Function Documentation	89
7.21.3.1	ChannelStatus	89
7.21.3.2	Dump	89
7.21.3.3	I0	90
7.21.3.4	I1	90
7.21.3.5	lmon	90
7.21.3.6	MaxV	90
7.21.3.7	RampDown	90
7.21.3.8	RampUp	90
7.21.3.9	Trip	90
7.21.3.10	V0	90
7.21.3.11	V1	90
7.21.3.12	Vmon	90
7.21.4	Field Documentation	90
7.21.4.1	fChannelId	90
7.21.4.2	fValues	90
7.22	NIM::HVModuleN470Values Class Reference	90
7.22.1	Detailed Description	91
7.22.2	Constructor & Destructor Documentation	91
7.22.2.1	HVModuleN470Values	91
7.22.2.2	~HVModuleN470Values	91
7.22.3	Member Function Documentation	91
7.22.3.1	Dump	91
7.22.3.2	GetChannelStatus	91
7.22.3.3	lmon	91
7.22.3.4	Vmax	91
7.22.3.5	Vmon	91
7.22.4	Field Documentation	91
7.22.4.1	fValues	92
7.23	VME::IOModuleV262 Class Reference	92
7.23.1	Constructor & Destructor Documentation	93
7.23.1.1	IOModuleV262	93
7.23.1.2	~IOModuleV262	93

7.23.2	Member Function Documentation	93
7.23.2.1	GetIdentifier	93
7.23.2.2	GetManufacturerId	94
7.23.2.3	GetModuleType	94
7.23.2.4	GetModuleVersion	94
7.23.2.5	GetSerialNumber	95
7.24	Logger Class Reference	95
7.24.1	Detailed Description	95
7.24.2	Constructor & Destructor Documentation	95
7.24.2.1	Logger	95
7.24.2.2	~Logger	95
7.24.3	Field Documentation	95
7.24.3.1	fBuffer	95
7.24.3.2	fStream	95
7.25	LogRedirector Class Reference	96
7.25.1	Detailed Description	96
7.25.2	Constructor & Destructor Documentation	96
7.25.2.1	LogRedirector	96
7.25.3	Member Function Documentation	96
7.25.3.1	contents	97
7.25.4	Field Documentation	97
7.25.4.1	fRedirect	97
7.25.4.2	fSS	97
7.26	Message Class Reference	97
7.26.1	Detailed Description	98
7.26.2	Constructor & Destructor Documentation	98
7.26.2.1	Message	98
7.26.2.2	Message	98
7.26.2.3	Message	98
7.26.2.4	~Message	98
7.26.3	Member Function Documentation	98
7.26.3.1	Dump	98
7.26.3.2	GetKey	98
7.26.3.3	GetString	98
7.26.3.4	IsFromWeb	98
7.26.4	Field Documentation	98
7.26.4.1	fString	98
7.27	Messenger Class Reference	99
7.27.1	Detailed Description	100
7.27.2	Constructor & Destructor Documentation	100

7.27.2.1	Messenger	100
7.27.2.2	Messenger	101
7.27.2.3	~Messenger	101
7.27.3	Member Function Documentation	101
7.27.3.1	AddClient	101
7.27.3.2	Broadcast	101
7.27.3.3	Connect	102
7.27.3.4	Disconnect	102
7.27.3.5	DisconnectClient	102
7.27.3.6	GetType	104
7.27.3.7	ProcessMessage	104
7.27.3.8	Receive	104
7.27.3.9	Send	105
7.27.3.10	SendAll	105
7.27.3.11	SendAll	106
7.27.3.12	StartAcquisition	106
7.27.3.13	StopAcquisition	106
7.27.3.14	SwitchClientType	107
7.27.4	Field Documentation	107
7.27.4.1	fNumAttempts	107
7.27.4.2	fPID	107
7.27.4.3	fStderrPipe	107
7.27.4.4	fStdoutPipe	107
7.28	OnlineDBHandler Class Reference	107
7.28.1	Detailed Description	108
7.28.2	Member Typedef Documentation	108
7.28.2.1	BurstInfos	108
7.28.2.2	RunCollection	108
7.28.2.3	TDCCConditionsCollection	108
7.28.3	Constructor & Destructor Documentation	108
7.28.3.1	OnlineDBHandler	108
7.28.3.2	~OnlineDBHandler	109
7.28.4	Member Function Documentation	109
7.28.4.1	BuildTables	109
7.28.4.2	GetLastBurst	109
7.28.4.3	GetLastRun	109
7.28.4.4	GetRunInfo	109
7.28.4.5	GetRuns	109
7.28.4.6	GetTDCCConditions	109
7.28.4.7	NewBurst	109

7.28.4.8	NewRun	109
7.28.4.9	Select	109
7.28.4.10	SetHVConditions	109
7.28.4.11	SetTDCConditions	110
7.28.5	Field Documentation	110
7.28.5.1	fDB	110
7.29	VME::PCIInterfaceA2818 Class Reference	110
7.29.1	Constructor & Destructor Documentation	110
7.29.1.1	PCIInterfaceA2818	110
7.29.1.2	~PCIInterfaceA2818	110
7.29.2	Member Function Documentation	110
7.29.2.1	GetFWRevision	110
7.29.3	Field Documentation	110
7.29.3.1	fHandle	110
7.30	DQM::PPSCanvas Class Reference	111
7.30.1	Detailed Description	112
7.30.2	Constructor & Destructor Documentation	112
7.30.2.1	PPSCanvas	112
7.30.2.2	PPSCanvas	112
7.30.2.3	PPSCanvas	112
7.30.2.4	~PPSCanvas	113
7.30.3	Member Function Documentation	113
7.30.3.1	Build	113
7.30.3.2	DrawGrid	113
7.30.3.3	Grid	113
7.30.3.4	Save	113
7.30.3.5	SetRunInfo	113
7.30.3.6	SetUpperLabel	113
7.30.4	Field Documentation	113
7.30.4.1	c1	113
7.30.4.2	c2	113
7.30.4.3	fHeight	113
7.30.4.4	fLabel1	113
7.30.4.5	fLabel2	113
7.30.4.6	fLabel3	113
7.30.4.7	fLabelsDrawn	114
7.30.4.8	fLegend	114
7.30.4.9	fLegendNumEntries	114
7.30.4.10	fLegendX	114
7.30.4.11	fLegendY	114

7.30.4.12 fRunDate	114
7.30.4.13 fRunId	114
7.30.4.14 fUpperLabel	114
7.30.4.15 fUpperLabelText	114
7.30.4.16 fWidth	114
7.31 DQM::QuarticCanvas Class Reference	114
7.31.1 Detailed Description	116
7.31.2 Constructor & Destructor Documentation	116
7.31.2.1 QuarticCanvas	116
7.31.2.2 QuarticCanvas	116
7.31.2.3 QuarticCanvas	116
7.31.2.4 ~QuarticCanvas	117
7.31.3 Member Function Documentation	117
7.31.3.1 Build	117
7.31.3.2 DrawGrid	117
7.31.3.3 FillChannel	117
7.31.3.4 GetCoordinates	117
7.31.3.5 Grid	117
7.31.3.6 Save	117
7.31.3.7 SetRunInfo	117
7.31.3.8 SetUpperLabel	117
7.31.4 Field Documentation	117
7.31.4.1 c1	118
7.31.4.2 c2	118
7.31.4.3 fBoardId	118
7.31.4.4 fHeight	118
7.31.4.5 fHist	118
7.31.4.6 fLabel1	118
7.31.4.7 fLabel2	118
7.31.4.8 fLabel3	118
7.31.4.9 fLabel4	118
7.31.4.10 fLabelsDrawn	118
7.31.4.11 fLegend	118
7.31.4.12 fLegendNumEntries	118
7.31.4.13 fLegendX	118
7.31.4.14 fLegendY	118
7.31.4.15 fRunDate	118
7.31.4.16 fRunId	118
7.31.4.17 fSpillId	118
7.31.4.18 fUpperLabel	118

7.31.4.19 fUpperLabelText	118
7.31.4.20 fWidth	118
7.32 Socket Class Reference	118
7.32.1 Detailed Description	120
7.32.2 Member Typedef Documentation	120
7.32.2.1 SocketCollection	120
7.32.3 Constructor & Destructor Documentation	120
7.32.3.1 Socket	120
7.32.3.2 Socket	121
7.32.3.3 ~Socket	121
7.32.4 Member Function Documentation	121
7.32.4.1 AcceptConnections	121
7.32.4.2 Bind	121
7.32.4.3 Configure	121
7.32.4.4 Create	122
7.32.4.5 DumpConnected	122
7.32.4.6 FetchMessage	122
7.32.4.7 GetPort	122
7.32.4.8 GetSocketId	122
7.32.4.9 GetSocketType	122
7.32.4.10 IsWebSocket	122
7.32.4.11 Listen	122
7.32.4.12 PrepareConnection	123
7.32.4.13 SelectConnections	123
7.32.4.14 SendMessage	123
7.32.4.15 SetPort	123
7.32.4.16 SetSocketId	123
7.32.4.17 Start	123
7.32.4.18 Stop	124
7.32.5 Field Documentation	124
7.32.5.1 fAddress	124
7.32.5.2 fBuffer	124
7.32.5.3 fMaster	124
7.32.5.4 fPort	124
7.32.5.5 fReadFds	124
7.32.5.6 fSocketId	124
7.32.5.7 fSocketsConnected	124
7.33 SocketMessage Class Reference	125
7.33.1 Detailed Description	127
7.33.2 Constructor & Destructor Documentation	127

7.33.2.1	SocketMessage	127
7.33.2.2	SocketMessage	127
7.33.2.3	SocketMessage	127
7.33.2.4	SocketMessage	128
7.33.2.5	SocketMessage	128
7.33.2.6	SocketMessage	128
7.33.2.7	SocketMessage	128
7.33.2.8	SocketMessage	129
7.33.2.9	SocketMessage	129
7.33.2.10	SocketMessage	129
7.33.2.11	SocketMessage	129
7.33.2.12	SocketMessage	130
7.33.2.13	SocketMessage	130
7.33.2.14	~SocketMessage	130
7.33.3	Member Function Documentation	130
7.33.3.1	Dump	130
7.33.3.2	GetCleanedValue	130
7.33.3.3	GetIntValue	130
7.33.3.4	GetKey	130
7.33.3.5	GetString	130
7.33.3.6	GetValue	131
7.33.3.7	GetVectorValue	131
7.33.3.8	Object	131
7.33.3.9	SetKeyValue	131
7.33.3.10	SetKeyValue	131
7.33.3.11	SetKeyValue	131
7.33.3.12	SetKeyValue	132
7.33.3.13	SetKeyValue	132
7.33.3.14	SetKeyValue	132
7.33.3.15	String	132
7.33.4	Field Documentation	132
7.33.4.1	fMessage	133
7.34	OnlineDBHandler::TDCConditions Struct Reference	133
7.34.1	Member Function Documentation	133
7.34.1.1	operator=	133
7.34.1.2	operator==	133
7.34.2	Field Documentation	133
7.34.2.1	detector	133
7.34.2.2	run_id	133
7.34.2.3	tdc_acq_mode	133

7.34.2.4	tdc_address	133
7.34.2.5	tdc_det_mode	133
7.34.2.6	tdc_id	133
7.35	VME::TDCErrorFlag Class Reference	133
7.35.1	Detailed Description	134
7.35.2	Constructor & Destructor Documentation	134
7.35.2.1	TDCErrorFlag	134
7.35.2.2	~TDCErrorFlag	134
7.35.3	Member Function Documentation	134
7.35.3.1	Dump	134
7.35.3.2	GetWord	134
7.35.3.3	HasGroupError	134
7.35.3.4	HasInternalChipError	135
7.35.3.5	HasL1BufferOverflow	135
7.35.3.6	HasReachedEventSizeLimit	135
7.35.3.7	HasReadoutFIFOOverflow	135
7.35.3.8	HasTriggerFIFOOverflow	135
7.35.4	Friends And Related Function Documentation	135
7.35.4.1	operator<<	135
7.35.5	Field Documentation	135
7.35.5.1	fWord	135
7.36	VME::TDCEvent Class Reference	135
7.36.1	Detailed Description	136
7.36.2	Member Enumeration Documentation	137
7.36.2.1	EventType	137
7.36.3	Constructor & Destructor Documentation	137
7.36.3.1	TDCEvent	137
7.36.3.2	TDCEvent	137
7.36.3.3	TDCEvent	137
7.36.3.4	TDCEvent	137
7.36.3.5	~TDCEvent	137
7.36.4	Member Function Documentation	137
7.36.4.1	Dump	137
7.36.4.2	GetBunchId	137
7.36.4.3	GetChannelId	138
7.36.4.4	GetErrorFlags	138
7.36.4.5	GetETTT	138
7.36.4.6	GetEventCount	139
7.36.4.7	GetEventId	139
7.36.4.8	GetGeo	139

7.36.4.9	GetStatus	140
7.36.4.10	GetTDCId	140
7.36.4.11	GetTime	140
7.36.4.12	GetType	140
7.36.4.13	GetWidth	141
7.36.4.14	GetWord	141
7.36.4.15	GetWordCount	141
7.36.4.16	IsTrailing	141
7.36.4.17	SetWord	141
7.36.5	Field Documentation	141
7.36.5.1	fWord	141
7.37	VME::TDCMeasurement Class Reference	142
7.37.1	Detailed Description	142
7.37.2	Constructor & Destructor Documentation	142
7.37.2.1	TDCMeasurement	142
7.37.2.2	TDCMeasurement	143
7.37.2.3	~TDCMeasurement	143
7.37.3	Member Function Documentation	143
7.37.3.1	Dump	144
7.37.3.2	GetBunchId	144
7.37.3.3	GetChannelId	144
7.37.3.4	GetETTT	144
7.37.3.5	GetEventId	144
7.37.3.6	GetLeadingTime	144
7.37.3.7	GetTDCId	144
7.37.3.8	GetToT	145
7.37.3.9	GetTrailingTime	145
7.37.3.10	NumErrors	145
7.37.3.11	NumEvents	145
7.37.3.12	SetEventsCollection	145
7.37.4	Field Documentation	145
7.37.4.1	fEvents	145
7.37.4.2	fMap	145
7.38	VME::TDCV1x90 Class Reference	145
7.38.1	Detailed Description	148
7.38.2	Member Enumeration Documentation	148
7.38.2.1	DLLMode	148
7.38.3	Constructor & Destructor Documentation	148
7.38.3.1	TDCV1x90	149
7.38.3.2	~TDCV1x90	149

7.38.4	Member Function Documentation	149
7.38.4.1	abort	149
7.38.4.2	CheckConfiguration	149
7.38.4.3	DisableChannel	150
7.38.4.4	EnableChannel	150
7.38.4.5	FetchEvents	150
7.38.4.6	GetAcquisitionMode	151
7.38.4.7	GetBLTEventNumberRegister	151
7.38.4.8	GetChannelDeadTime	151
7.38.4.9	GetControl	152
7.38.4.10	GetDetectionMode	152
7.38.4.11	GetDLLClock	152
7.38.4.12	GetErrorMarks	152
7.38.4.13	GetETTT	152
7.38.4.14	GetEventCounter	152
7.38.4.15	GetEventStored	153
7.38.4.16	GetFIFOSize	153
7.38.4.17	GetFirmwareRevision	154
7.38.4.18	GetGlobalOffset	154
7.38.4.19	GetModel	154
7.38.4.20	GetOUI	155
7.38.4.21	GetPol	155
7.38.4.22	GetRCAdjust	155
7.38.4.23	GetResolution	156
7.38.4.24	GetSerialNumber	156
7.38.4.25	GetStatus	156
7.38.4.26	GetTDCEncapsulation	157
7.38.4.27	GetTestMode	157
7.38.4.28	GetTriggerConfiguration	157
7.38.4.29	GetWindowOffset	157
7.38.4.30	GetWindowWidth	157
7.38.4.31	HardwareReset	157
7.38.4.32	ReadAcquisitionMode	158
7.38.4.33	ReadDetectionMode	158
7.38.4.34	SetAcquisitionMode	158
7.38.4.35	SetBLTEventNumberRegister	159
7.38.4.36	SetChannelDeadTime	159
7.38.4.37	SetContinuousStorage	159
7.38.4.38	SetControl	160
7.38.4.39	SetDetectionMode	160

7.38.4.40 SetDLLClock	160
7.38.4.41 SetErrorMarks	161
7.38.4.42 SetETTT	161
7.38.4.43 SetFIFOSize	161
7.38.4.44 SetGlobalOffset	162
7.38.4.45 SetLSBTraileadEdge	162
7.38.4.46 SetPairModeResolution	162
7.38.4.47 SetPol	163
7.38.4.48 SetRCAdjust	163
7.38.4.49 SetStatus	163
7.38.4.50 SetTDCEncapsulation	164
7.38.4.51 SetTestMode	164
7.38.4.52 SetTriggerMatching	164
7.38.4.53 SetTriggerTimeSubtraction	165
7.38.4.54 SetVerboseLevel	165
7.38.4.55 SetWindowOffset	165
7.38.4.56 SetWindowWidth	165
7.38.4.57 SoftwareClear	166
7.38.4.58 SoftwareReset	166
7.38.4.59 WaitMicro	167
7.38.5 Field Documentation	167
7.38.5.1 fAcquisitionMode	167
7.38.5.2 fBuffer	167
7.38.5.3 fDetectionMode	167
7.38.5.4 fErrorMarks	167
7.38.5.5 fVerb	167
7.38.5.6 fWindowWidth	167
7.38.5.7 gEnd	167
7.38.5.8 nchannels	167
7.38.5.9 pair_lead_res	167
7.38.5.10 pair_width_res	167
7.39 VME::TDCV1x90Control Class Reference	167
7.39.1 Detailed Description	168
7.39.2 Constructor & Destructor Documentation	168
7.39.2.1 TDCV1x90Control	168
7.39.2.2 ~TDCV1x90Control	168
7.39.3 Member Function Documentation	168
7.39.3.1 Dump	169
7.39.3.2 GetAlign64	169
7.39.3.3 GetBusError	169

7.39.3.4	GetCompensation	170
7.39.3.5	GetEmptyEvent	170
7.39.3.6	GetETTT	170
7.39.3.7	GetEventFIFO	170
7.39.3.8	GetMEBAccess	170
7.39.3.9	GetSRAMCompensation	170
7.39.3.10	GetSWTermination	170
7.39.3.11	GetTermination	170
7.39.3.12	GetTestFIFO	170
7.39.3.13	GetValue	170
7.39.3.14	SetAlign64	170
7.39.3.15	SetBusError	170
7.39.3.16	SetCompensation	171
7.39.3.17	SetEmptyEvent	171
7.39.3.18	SetETTT	171
7.39.3.19	SetEventFIFO	172
7.39.3.20	SetMEBAccess	172
7.39.3.21	SetSRAMCompensation	172
7.39.3.22	SetSWTermination	173
7.39.3.23	SetTermination	173
7.39.3.24	SetTestFIFO	173
7.39.4	Field Documentation	173
7.39.4.1	fWord	173
7.40	VME::TDCV1x90Status Class Reference	174
7.40.1	Detailed Description	174
7.40.2	Member Enumeration Documentation	174
7.40.2.1	TDCResolution	174
7.40.3	Constructor & Destructor Documentation	175
7.40.3.1	TDCV1x90Status	175
7.40.3.2	~TDCV1x90Status	175
7.40.4	Member Function Documentation	175
7.40.4.1	AlmostFull	175
7.40.4.2	BusError	175
7.40.4.3	DataReady	175
7.40.4.4	Dump	176
7.40.4.5	Error	176
7.40.4.6	Error	177
7.40.4.7	Full	177
7.40.4.8	GetValue	177
7.40.4.9	HeadersEnabled	177

7.40.4.10 PairMode	177
7.40.4.11 Purged	177
7.40.4.12 Resolution	177
7.40.4.13 TerminationOn	177
7.40.4.14 TriggerLost	177
7.40.4.15 TriggerMatching	177
7.40.5 Field Documentation	177
7.40.5.1 fWord	177
7.41 VME::TDCV1x90TriggerConfig Class Reference	177
7.41.1 Constructor & Destructor Documentation	178
7.41.1.1 TDCV1x90TriggerConfig	178
7.41.1.2 ~TDCV1x90TriggerConfig	178
7.41.2 Member Function Documentation	178
7.41.2.1 Dump	178
7.41.2.2 GetExtraSearchWindowWidth	178
7.41.2.3 GetRejectMargin	178
7.41.2.4 GetWindowOffset	178
7.41.2.5 GetWindowWidth	178
7.41.2.6 HasTriggerTimeSubtraction	179
7.41.2.7 SetWord	179
7.41.3 Field Documentation	179
7.41.3.1 fWords	179
7.42 VME::trailead_t Struct Reference	179
7.42.1 Field Documentation	179
7.42.1.1 ettt	179
7.42.1.2 event_count	179
7.42.1.3 leading	179
7.42.1.4 total_hits	179
7.42.1.5 trailing	179
7.43 VMEReader Class Reference	179
7.43.1 Detailed Description	182
7.43.2 Member Typedef Documentation	182
7.43.2.1 OutputFiles	182
7.43.3 Member Enumeration Documentation	182
7.43.3.1 GlobalAcqMode	182
7.43.4 Constructor & Destructor Documentation	182
7.43.4.1 VMEReader	182
7.43.4.2 ~VMEReader	183
7.43.5 Member Function Documentation	183
7.43.5.1 Abort	183

7.43.5.2	AddCFD	183
7.43.5.3	AddFPGAUnit	184
7.43.5.4	AddHVModule	184
7.43.5.5	AddIOModule	185
7.43.5.6	AddTDC	185
7.43.5.7	BroadcastHVStatus	185
7.43.5.8	BroadcastNewBurst	186
7.43.5.9	BroadcastTriggerRate	186
7.43.5.10	GetBurstNumber	186
7.43.5.11	GetCFD	186
7.43.5.12	GetCFDCollection	186
7.43.5.13	GetFPGAUnit	186
7.43.5.14	GetFPGAUnitCollection	186
7.43.5.15	GetGlobalAcquisitionMode	186
7.43.5.16	GetHVModule	186
7.43.5.17	GetIOModule	187
7.43.5.18	GetNumCFD	187
7.43.5.19	GetNumTDC	187
7.43.5.20	GetOutputFile	187
7.43.5.21	GetRunNumber	187
7.43.5.22	GetTDC	187
7.43.5.23	GetTDCCollection	187
7.43.5.24	LogHVValues	187
7.43.5.25	NewBurst	188
7.43.5.26	NewRun	188
7.43.5.27	ReadXML	188
7.43.5.28	ReadXML	190
7.43.5.29	SendClear	190
7.43.5.30	SendOutputFile	190
7.43.5.31	SendPulse	190
7.43.5.32	SetOutputFile	191
7.43.5.33	StartPulser	191
7.43.5.34	StopPulser	191
7.43.5.35	UseSocket	191
7.43.6	Field Documentation	191
7.43.6.1	fBridge	191
7.43.6.2	fCAENET	191
7.43.6.3	fCFDCollection	191
7.43.6.4	fFPGACollection	192
7.43.6.5	fGlobalAcqMode	192

7.43.6.6 fHV	192
7.43.6.7 flsPulserStarted	192
7.43.6.8 fOnSocket	192
7.43.6.9 fOutputFiles	192
7.43.6.10 fSG	192
7.43.6.11 fTDCCollection	192
 Index	 193

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:

DQM	11
NIM	11
VME	12
VME::TDCV1x90Opcodes	18

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

VME::BridgeVx718Control	27
VME::BridgeVx718Status	29
OnlineDBHandler::BurstInfo	31
VME::CAENETControllerV288Status	34
NIM::HVModuleN470Values::ChannelStatus	41
DQM::GastofCanvas::Coord	48
DQM::QuarticCanvas::Coord	49
file_header_t	53
FileReader	54
VME::FPGAUnitV1495Control	71
VME::GenericBoard< Register, am >	81
VME::GenericBoard< CAENETControllerV288Register, cvA24_U_DATA >	81
VME::CAENETControllerV288	31
VME::GenericBoard< CFDV812Register, cvA24_U_DATA >	81
VME::CFDV812	35
VME::GenericBoard< CVRegisters, cvA32_U_DATA >	81
VME::BridgeVx718	23
VME::GenericBoard< FPGAUnitV1495Register, cvA32_U_DATA >	81
VME::FPGAUnitV1495	57
VME::GenericBoard< IOModuleV262Register, cvA24_U_DATA >	81
VME::IOModuleV262	92
VME::GenericBoard< TDCV1x90Register, cvA32_U_DATA >	81
VME::TDCV1x90	145
VME::GlobalOffset	83
NIM::HVModuleN470	83
NIM::HVModuleN470ChannelValues	88
NIM::HVModuleN470Values	90
Logger	95
LogRedirector	96
Message	97
SocketMessage	125
OnlineDBHandler	107
VME::PCIInterfaceA2818	110
Socket	118
Client	43
DQM::DQMProcess	49

VMEReader	179
Messenger	99
TCanvas	
DQM::GastofCanvas	76
DQM::PPSCanvas	111
DQM::QuarticCanvas	114
OnlineDBHandler::TDCConditions	133
VME::TDCErrorFlag	133
VME::TDCEvent	135
VME::TDCMeasurement	142
VME::TDCV1x90Control	167
VME::TDCV1x90Status	174
VME::TDCV1x90TriggerConfig	177
VME::trailead_t	179

Chapter 4

Data Structure Index

4.1 Data Structures

Here are the data structures with brief descriptions:

VME::BridgeVx718	
Class defining the VME bridge	23
VME::BridgeVx718Control	27
VME::BridgeVx718Status	29
OnlineDBHandler::BurstInfo	31
VME::CAENETControllerV288	
Handler for a CAEN V288 CAENET controller	31
VME::CAENETControllerV288Status	34
VME::CFDV812	
Controller for a CAEN V812 constant fraction discriminator	35
NIM::HVModuleN470Values::ChannelStatus	41
Client	
Base client object for the socket	43
DQM::GastofCanvas::Coord	48
DQM::QuarticCanvas::Coord	49
DQM::DQMProcess	
Handler for a common DQM process to run on the socket	49
file_header_t	
Header to the output files	53
FileReader	
Handler for a TDC output file readout	54
VME::FPGAUnitV1495	57
VME::FPGAUnitV1495Control	71
DQM::GastofCanvas	76
VME::GenericBoard< Register, am >	81
VME::GlobalOffset	83
NIM::HVModuleN470	83
NIM::HVModuleN470ChannelValues	
Single channel monitoring values for the HV power supply	88
NIM::HVModuleN470Values	
General monitoring values for the HV power supply	90
VME::IOModuleV262	92
Logger	
Redirect outputs to another output stream	95
LogRedirector	
Redirect output stream to a string	96
Message	
Base socket message type	97

Messenger	
Base master object for the socket	99
OnlineDBHandler	
Handler for the run information online database	107
VME::PCIInterfaceA2818	110
DQM::PPSCanvas	111
DQM::QuarticCanvas	114
Socket	
Base socket object from which clients/master from a socket inherit	118
SocketMessage	
Socket-passed message type	125
OnlineDBHandler::TDCConditions	133
VME::TDCErrorFlag	
Error flags handler	133
VME::TDCEvent	
HPTDC event parser	135
VME::TDCMeasurement	142
VME::TDCV1x90	145
VME::TDCV1x90Control	
TDC control register	167
VME::TDCV1x90Status	
TDC status register	174
VME::TDCV1x90TriggerConfig	177
VME::trailead_t	179
VMEReader	179

Chapter 5

Module Documentation

5.1 Socket communication objects

Data Structures

- class [Client](#)
Base client object for the socket.
- 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](#), [Socket::DAQ](#) }
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
DAQ

Chapter 6

Namespace Documentation

6.1 DQM Namespace Reference

Data Structures

- class [DQMProcess](#)
Handler for a common [DQM](#) process to run on the socket.
- class [GastofCanvas](#)
- class [PPSCanvas](#)
- class [QuarticCanvas](#)

6.2 NIM Namespace Reference

Data Structures

- class [HVModuleN470](#)
- class [HVModuleN470ChannelValues](#)
Single channel monitoring values for the HV power supply.
- class [HVModuleN470Values](#)
General monitoring values for the HV power supply.

Enumerations

- enum [HVModuleN470Opcodes](#) {
 [kN470GeneralInfo](#) = 0x00, [kN470MonStatus](#) = 0x01, [kN470OperationalParams](#) = 0x02, [kN470V0Value](#) = 0x03,
 [kN470I0Value](#) = 0x04, [kN470V1Value](#) = 0x05, [kN470I1Value](#) = 0x06, [kN470TripValue](#) = 0x07,
 [kN470RampUpValue](#) = 0x08, [kN470RampDownValue](#) = 0x09, [kN470ChannelOn](#) = 0x0a, [kN470ChannelOff](#) = 0x0b,
 [kN470KillAllChannels](#) = 0x0c, [kN470ClearAlarm](#) = 0x0d, [kN470EnableFrontPanel](#) = 0x0e, [kN470DisableFrontPanel](#) = 0x0f,
 [kN470TTLLLevel](#) = 0x10, [kN470NIMLevel](#) = 0x11 }

6.2.1 Enumeration Type Documentation

6.2.1.1 enum NIM::HVModuleN470Opcodes

Enumerator

kN470GeneralInfo
kN470MonStatus
kN470OperationalParams
kN470V0Value
kN470I0Value
kN470V1Value
kN470I1Value
kN470TripValue
kN470RampUpValue
kN470RampDownValue
kN470ChannelOn
kN470ChannelOff
kN470KillAllChannels
kN470ClearAlarm
kN470EnableFrontPanel
kN470DisableFrontPanel
kN470TTLLLevel
kN470NIMLevel

6.3 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.

- class [TDCV1x90TriggerConfig](#)
- struct [trailead_t](#)

Typedefs

- typedef std::map< uint32_t, [VME::CFDV812](#) * > [CFDCollection](#)
Mapper from physical [VME](#) addresses to pointers to CFD objects.
- typedef std::map< uint32_t, [VME::FPGAUnitV1495](#) * > [FPGAUnitCollection](#)
Mapper from physical [VME](#) addresses to pointers to FPGA 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 [CAENETControllerV288Answer](#) {
 [cnSuccess](#) = 0x0000, [cnBusy](#) = 0xff00, [cnUnrecognizedCode](#) = 0xff01, [cnIncorrectValue](#) = 0xff02,
 [cnNoData](#) = 0xfffd, [cnIncorrectHCC](#) = 0xfffe, [cnWrongModuleAddress](#) = 0xffff }
- 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,
 [kV1495ThresholdVoltage0](#) = 0x1028, [kV1495ThresholdVoltage1](#) = 0x1010, [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 [FPGAUnitV1495DACCH](#) { [ch0](#) = 0, [ch1](#) = 4096, [ch2](#) = 8192, [ch3](#) = 12288 }
- 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, [OTRILING](#) = 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, [kROM←](#)
[Oui2](#) = 0x4024,
[kROMOui1](#) = 0x4028, [kROMOui0](#) = 0x402c, [kROMBoard2](#) = 0x4034, [kROMBoard1](#) = 0x4038,
[kROMBoard0](#) = 0x403c, [kROMRevis3](#) = 0x4040, [kROMRevis2](#) = 0x4044, [kROMRevis1](#) = 0x4048,
[kROMRevis0](#) = 0x404c, [kROMSerNum1](#) = 0x4080, [kROMSerNum0](#) = 0x4084 }

Variables

- const int [OneVolt](#) = 1634

6.3.1 Typedef Documentation

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

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

6.3.1.2 typedef std::map<uint32_t,VME::FPGAUnitV1495*> VME::FPGAUnitCollection

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

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

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

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

6.3.2 Enumeration Type Documentation

6.3.2.1 enum VME::AcquisitionMode

TDC acquisition mode.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Enumerator

CONT_STORAGE

TRIG_MATCH

6.3.2.2 enum VME::BridgeType

Compatible bridge types.

Enumerator

CAEN_V1718

CAEN_V2718

6.3.2.3 enum VME::CAENETControllerV288Answer

Enumerator

cnSuccess

cnBusy

cnUnrecognizedCode

cnIncorrectValue

cnNoData

cnIncorrectHCC

cnWrongModuleAddress

6.3.2.4 enum VME::CAENETControllerV288Register

Enumerator

kV288DataBuffer

kV288Status

kV288Transmission

kV288ModuleReset

kV288IRQVector

6.3.2.5 enum VME::CFDV812Register

Enumerator

kV812ThresholdChannel0

kV812OutputWidthGroup0

kV812OutputWidthGroup1

kV812DeadTimeGroup0

kV812DeadTimeGroup1

kV812MajorityThreshold

kV812PatternOfInhibit

kV812TestPulse

kV812FixedCode

kV812Info0

kV812Info1

6.3.2.6 enum VME::DetectionMode

Enumerator

PAIR
OTRILING
OLEADING
TRAILEAD

6.3.2.7 enum VME::FPGAUnitV1495DACCH

Enumerator

cH0
cH1
cH2
cH3

6.3.2.8 enum VME::FPGAUnitV1495Register

Enumerator

kV1495ScalerCounter
kV1495DelaySettings
kV1495UserFWRevision
kV1495TDCBoardInterface
kV1495ClockSettings
kV1495Control
kV1495TriggerSettings
kV1495OutputSettings
kV1495ThresholdVoltage0
kV1495ThresholdVoltage1
kV1495GeoAddress
kV1495UserFPGAFlashMem
kV1495UserFPGAConfig
kV1495ModuleReset
kV1495FWRevision
kV1495ConfigurationROM
kV1495OUI2
kV1495OUI1
kV1495OUI0
kV1495Board2
kV1495Board1
kV1495Board0
kV1495HWRevision3
kV1495HWRevision2
kV1495HWRevision1
kV1495HWRevision0
kV1495SerNum0
kV1495SerNum1

6.3.2.9 enum VME::IOModuleV262Register

Enumerator

kECLLevelWrite
kNIMLevelWrite
kNIMPulseWrite
kNIMPulseRead
kIdentifier
kBoardInfo0
kBoardInfo1

6.3.2.10 enum VME::micro_handshake

Enumerator

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

6.3.2.11 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

6.3.2.12 enum VME::trailead_edge_lsb

Enumerator

r800ps
r200ps
r100ps
r25ps

6.3.2.13 enum VME::trig_conf

Enumerator

MATCH_WIN_WIDTH
WIN_OFFSET
EXTRA_SEARCH_WIN_WIDTH
REJECT_MARGIN
TRIG_TIME_SUB

6.3.3 Variable Documentation

6.3.3.1 const int VME::OneVolt = 1634

6.4 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)

6.4.1 Function Documentation

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

- 6.4.1.28 Opcode VME::TDCV1x90Opcodes::READ_EN_PATTERN (0x4500)
- 6.4.1.29 Opcode VME::TDCV1x90Opcodes::READ_EN_PATTERN32 (0x4700)
- 6.4.1.30 Opcode VME::TDCV1x90Opcodes::READ_ERROR_STATUS (0x7400)
- 6.4.1.31 Opcode VME::TDCV1x90Opcodes::READ_ERROR_TYPES (0x3a00)
- 6.4.1.32 Opcode VME::TDCV1x90Opcodes::READ_EVENT_SIZE (0x3400)
- 6.4.1.33 Opcode VME::TDCV1x90Opcodes::READ_FIFO_SIZE (0x3c00)
- 6.4.1.34 Opcode VME::TDCV1x90Opcodes::READ_GLOB_OFFS (0x5100)
- 6.4.1.35 Opcode VME::TDCV1x90Opcodes::READ_HEAD_TRAILER (0x3200)
- 6.4.1.36 Opcode VME::TDCV1x90Opcodes::READ_MICRO_REV (0x6100)
- 6.4.1.37 Opcode VME::TDCV1x90Opcodes::READ_RC_ADJ (0x5500)
- 6.4.1.38 Opcode VME::TDCV1x90Opcodes::READ_RES (0x2600)
- 6.4.1.39 Opcode VME::TDCV1x90Opcodes::READ_SETUP_REG (0x7100)
- 6.4.1.40 Opcode VME::TDCV1x90Opcodes::READ_SETUP_SCANPATH (0xc900)
- 6.4.1.41 Opcode VME::TDCV1x90Opcodes::READ_SPARE (0xc400)
- 6.4.1.42 Opcode VME::TDCV1x90Opcodes::READ_STATUS_STREAM (0x7600)
- 6.4.1.43 Opcode VME::TDCV1x90Opcodes::READ_TDC_ID (0x6000)
- 6.4.1.44 Opcode VME::TDCV1x90Opcodes::READ_TRG_CONF (0x1600)
- 6.4.1.45 Opcode VME::TDCV1x90Opcodes::RESET_DLL_PLL (0x6200)
- 6.4.1.46 Opcode VME::TDCV1x90Opcodes::REV_DATE_MICRO_FW (0xc200)
- 6.4.1.47 Opcode VME::TDCV1x90Opcodes::SAVE_RC_ADJ (0x5600)
- 6.4.1.48 Opcode VME::TDCV1x90Opcodes::SAVE_USER_CONFIG (0x0600)
- 6.4.1.49 Opcode VME::TDCV1x90Opcodes::SET_ADJUST_CH (0x5200)
- 6.4.1.50 Opcode VME::TDCV1x90Opcodes::SET_DEAD_TIME (0x2800)
- 6.4.1.51 Opcode VME::TDCV1x90Opcodes::SET_DETECTION (0x2200)
- 6.4.1.52 Opcode VME::TDCV1x90Opcodes::SET_DLL_CLOCK (0xc800)
- 6.4.1.53 Opcode VME::TDCV1x90Opcodes::SET_ERROR_TYPES (0x3900)
- 6.4.1.54 Opcode VME::TDCV1x90Opcodes::SET_EVENT_SIZE (0x3300)
- 6.4.1.55 Opcode VME::TDCV1x90Opcodes::SET_FIFO_SIZE (0x3b00)

- 6.4.1.56 Opcode VME::TDCV1x90OpCodes::SET_GLOB_OFFS (0x5000)
- 6.4.1.57 Opcode VME::TDCV1x90OpCodes::SET_KEEP_TOKEN (0x0300)
- 6.4.1.58 Opcode VME::TDCV1x90OpCodes::SET_PAIR_RES (0x2500)
- 6.4.1.59 Opcode VME::TDCV1x90OpCodes::SET_RC_ADJ (0x5400)
- 6.4.1.60 Opcode VME::TDCV1x90OpCodes::SET_REJ_MARGIN (0x1300)
- 6.4.1.61 Opcode VME::TDCV1x90OpCodes::SET_SW_MARGIN (0x1200)
- 6.4.1.62 Opcode VME::TDCV1x90OpCodes::SET_TDC_TSET_OUTPUT (0xc700)
- 6.4.1.63 Opcode VME::TDCV1x90OpCodes::SET_TR_LEAD_LSB (0x2400)
- 6.4.1.64 Opcode VME::TDCV1x90OpCodes::SET_WIN_OFFS (0x1100)
- 6.4.1.65 Opcode VME::TDCV1x90OpCodes::SET_WIN_WIDTH (0x1000)
- 6.4.1.66 Opcode VME::TDCV1x90OpCodes::TRG_MATCH (0x0000)
- 6.4.1.67 Opcode VME::TDCV1x90OpCodes::UPDATE_SETUP_REG (0x7200)
- 6.4.1.68 Opcode VME::TDCV1x90OpCodes::UPDATE_SETUP_TDC (0x7700)
- 6.4.1.69 Opcode VME::TDCV1x90OpCodes::WRITE_EEPROM (0xc000)
- 6.4.1.70 Opcode VME::TDCV1x90OpCodes::WRITE_EN_PATTERN (0x4400)
- 6.4.1.71 Opcode VME::TDCV1x90OpCodes::WRITE_EN_PATTERN32 (0x4600)
- 6.4.1.72 Opcode VME::TDCV1x90OpCodes::WRITE_SETUP_REG (0x7000)
- 6.4.1.73 Opcode VME::TDCV1x90OpCodes::WRITE_SPARE (0xc300)

Chapter 7

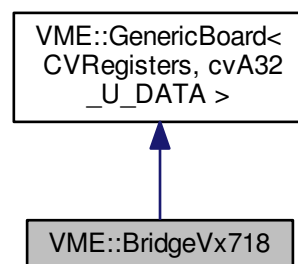
Data Structure Documentation

7.1 VME::BridgeVx718 Class Reference

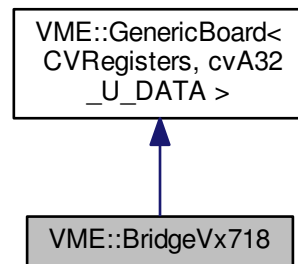
class defining the [VME](#) bridge

```
#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*)

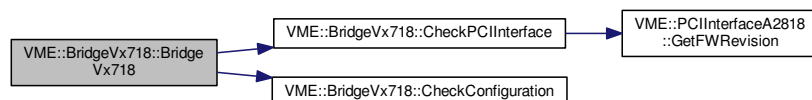
Constructor.

Bridge class constructor

Parameters

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

Here is the call graph for this function:



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

Destructor.

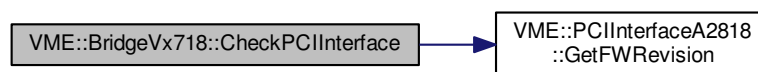
Bridge class destructor

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

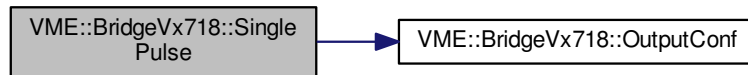
7.1.4.11 void VME::BridgeVx718::Reset () const

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:

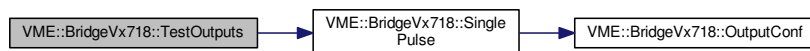


7.1.4.14 void VME::BridgeVx718::StartPulser (double *period*, double *width*, unsigned int *num_pulses* = 0) const

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

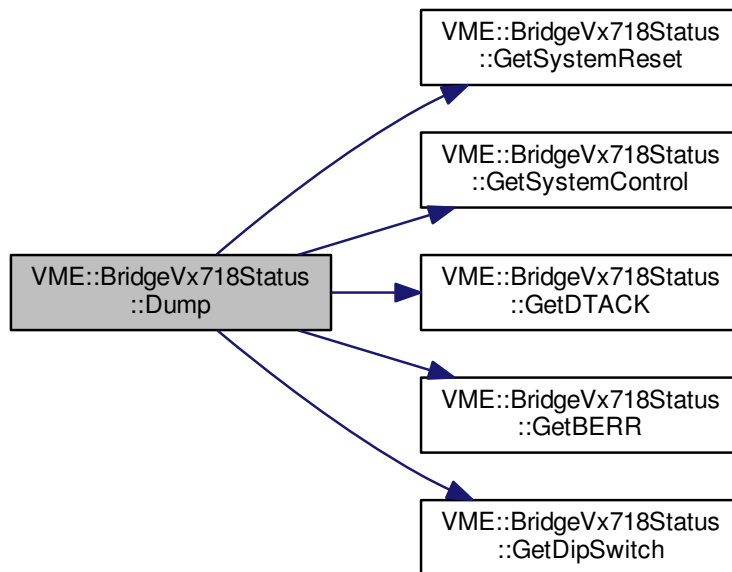
7.3.1.1 `VME::BridgeVx718Status::BridgeVx718Status (uint16_t word) [inline]`

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 OnlineDBHandler::BurstInfo Struct Reference

```
#include <OnlineDBHandler.h>
```

Data Fields

- unsigned int [burst_id](#)
- unsigned int [time_start](#)

7.4.1 Field Documentation

7.4.1.1 unsigned int OnlineDBHandler::BurstInfo::burst_id

7.4.1.2 unsigned int OnlineDBHandler::BurstInfo::time_start

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

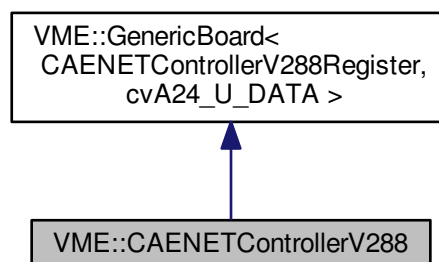
- include/OnlineDBHandler.h

7.5 VME::CAENETControllerV288 Class Reference

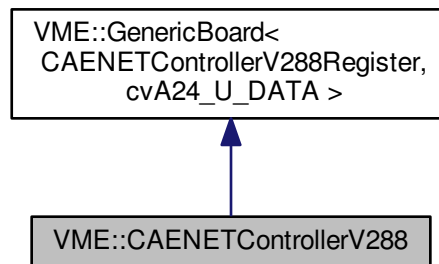
Handler for a CAEN V288 CAENET controller.

```
#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 [Reset](#) () const
- [CAENETControllerV288Status](#) [GetStatus](#) () const
- void [SendBuffer](#) () const
Send the whole buffer through the network.
- std::vector< uint16_t > [FetchBuffer](#) (unsigned int num_words) const
Retrieve the network buffer.
- bool [WaitForResponse](#) ([CAENETControllerV288Answer](#) *response, unsigned int max_trials=-1) const

Friends

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

Additional Inherited Members

7.5.1 Detailed Description

Handler for a CAEN V288 CAENET controller.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

23 Jul 2015

7.5.2 Constructor & Destructor Documentation

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

7.5.2.2 `VME::CAENETControllerV288::~~CAENETControllerV288 ()`

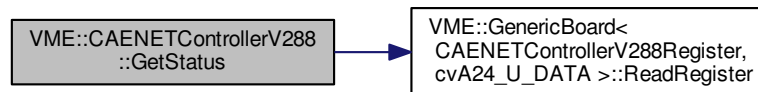
7.5.3 Member Function Documentation

7.5.3.1 `std::vector< uint16_t > VME::CAENETControllerV288::FetchBuffer (unsigned int num_words = 1) const`

Retrieve the network buffer.

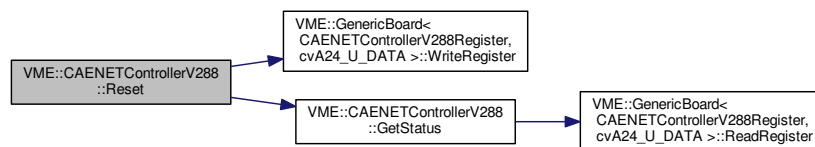
7.5.3.2 `CAENETControllerV288Status VME::CAENETControllerV288::GetStatus () const`

Here is the call graph for this function:



7.5.3.3 `void VME::CAENETControllerV288::Reset () const`

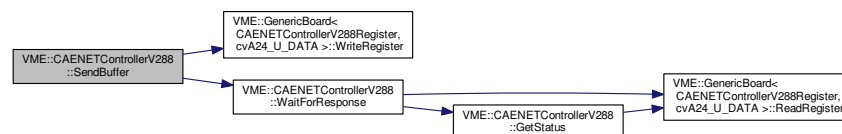
Here is the call graph for this function:



7.5.3.4 `void VME::CAENETControllerV288::SendBuffer () const`

Send the whole buffer through the network.

Here is the call graph for this function:



7.5.3.5 `bool VME::CAENETControllerV288::WaitForResponse (CAENETControllerV288Answer * response, unsigned int max_trials = -1) const`

Here is the call graph for this function:



7.5.4 Friends And Related Function Documentation

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

Fill the buffer with an additional 16-bit word.

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

Read back a 16-bit word from the buffer.

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

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

7.6 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](#)

7.6.1 Member Enumeration Documentation

7.6.1.1 `enum VME::CAENETControllerV288Status::OperationStatus`

Enumerator

Valid

Invalid

7.6.2 Constructor & Destructor Documentation

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

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

7.6.3 Member Function Documentation

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

7.6.4 Field Documentation

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

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

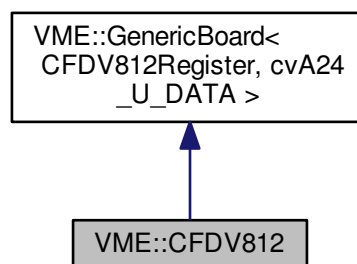
- include/VME_CAENETControllerV288.h

7.7 VME::CFDV812 Class Reference

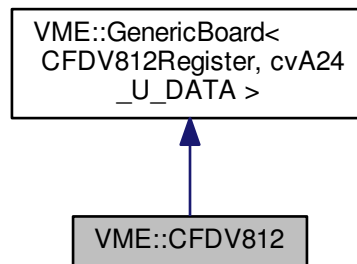
Controller for a CAEN V812 constant fraction discriminator.

```
#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

7.7.1 Detailed Description

Controller for a CAEN V812 constant fraction discriminator.

Author

Laurent Forthomme laurent.forthomme@cern.ch

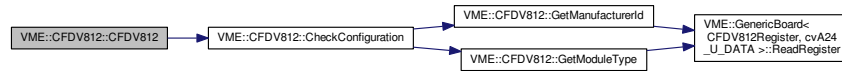
Date

22 Jul 2015

7.7.2 Constructor & Destructor Documentation

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

Here is the call graph for this function:

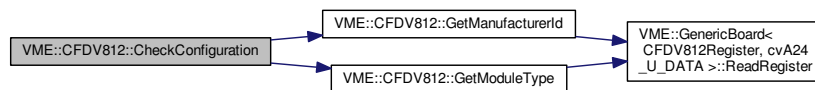


7.7.2.2 VME::CFDV812::~~CFDV812 () [inline]

7.7.3 Member Function Documentation

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

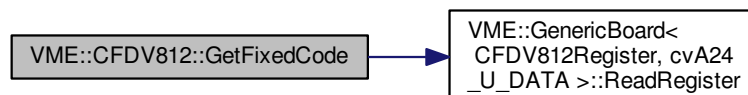
Here is the call graph for this function:



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

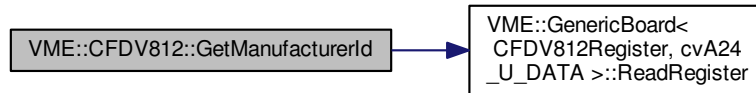
7.7.3.3 unsigned short VME::CFDV812::GetFixedCode () const

Here is the call graph for this function:



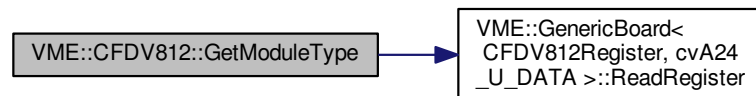
7.7.3.4 unsigned short VME::CFDV812::GetManufacturerId () const

Here is the call graph for this function:



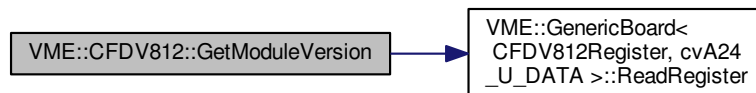
7.7.3.5 unsigned short VME::CFDV812::GetModuleType () const

Here is the call graph for this function:



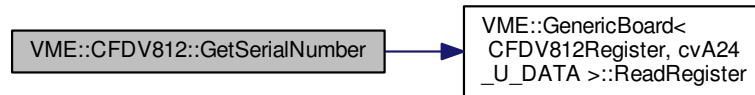
7.7.3.6 unsigned short VME::CFDV812::GetModuleVersion () const

Here is the call graph for this function:



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

Here is the call graph for this function:

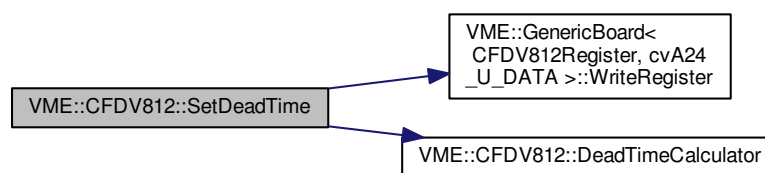
7.7.3.8 float VME::CFDV812::OutputWidthCalculator (unsigned short *value*) const [private]7.7.3.9 void VME::CFDV812::SetDeadTime (unsigned short *group_id*, unsigned short *value*) const

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

Parameters

in	<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:

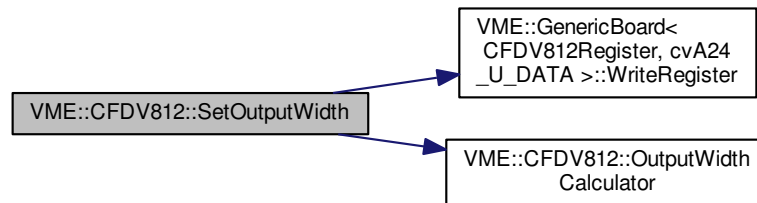
7.7.3.10 void VME::CFDV812::SetOutputWidth (unsigned short *group_id*, unsigned short *value*) const

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

Parameters

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

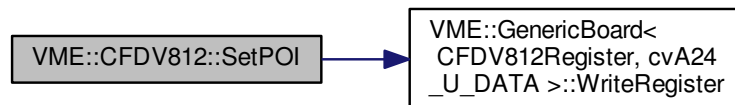
Here is the call graph for this function:



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

Set the pattern of inhibit (list of enabled channels)

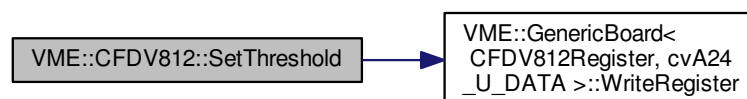
Here is the call graph for this function:



7.7.3.12 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

7.8 NIM::HVModuleN470Values::ChannelStatus Class Reference

```
#include <NIM_HVModuleN470.h>
```

Public Types

- enum [SignalStandard](#) { [NIM](#) =0x0, [TTL](#) =0x1 }

Public Member Functions

- [ChannelStatus](#) (unsigned short word)
- [~ChannelStatus](#) ()
- bool [Enabled](#) () const
- bool [OVC](#) () const
- bool [OVV](#) () const
- bool [UNV](#) () const
- bool [Trip](#) () const
- bool [RampUp](#) () const
- bool [RampDown](#) () const
- bool [MaxV](#) () const
- bool [Polarity](#) () const
- bool [Vsel](#) () const
- bool [Isel](#) () const
- bool [Kill](#) () const
- bool [HVEnabled](#) () const
- [SignalStandard Standard](#) () const
- bool [NonCalibrated](#) () const
- bool [Alarm](#) () const
- void [Dump](#) () const

Private Attributes

- unsigned short [fWord](#)

Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [ChannelStatus](#) &cs)

7.8.1 Member Enumeration Documentation

7.8.1.1 enum NIM::HVModuleN470Values::ChannelStatus::SignalStandard

Enumerator

NIM

TTL

7.8.2 Constructor & Destructor Documentation

7.8.2.1 `NIM::HVModuleN470Values::ChannelStatus::ChannelStatus (unsigned short word)` `[inline]`

7.8.2.2 `NIM::HVModuleN470Values::ChannelStatus::~~ChannelStatus ()` `[inline]`

7.8.3 Member Function Documentation

7.8.3.1 `bool NIM::HVModuleN470Values::ChannelStatus::Alarm () const` `[inline]`

7.8.3.2 `void NIM::HVModuleN470Values::ChannelStatus::Dump () const` `[inline]`

7.8.3.3 `bool NIM::HVModuleN470Values::ChannelStatus::Enabled () const` `[inline]`

7.8.3.4 `bool NIM::HVModuleN470Values::ChannelStatus::HVEnabled () const` `[inline]`

7.8.3.5 `bool NIM::HVModuleN470Values::ChannelStatus::Isel () const` `[inline]`

7.8.3.6 `bool NIM::HVModuleN470Values::ChannelStatus::Kill () const` `[inline]`

7.8.3.7 `bool NIM::HVModuleN470Values::ChannelStatus::MaxV () const` `[inline]`

7.8.3.8 `bool NIM::HVModuleN470Values::ChannelStatus::NonCalibrated () const` `[inline]`

7.8.3.9 `bool NIM::HVModuleN470Values::ChannelStatus::OVC () const` `[inline]`

7.8.3.10 `bool NIM::HVModuleN470Values::ChannelStatus::OVV () const` `[inline]`

7.8.3.11 `bool NIM::HVModuleN470Values::ChannelStatus::Polarity () const` `[inline]`

7.8.3.12 `bool NIM::HVModuleN470Values::ChannelStatus::RampDown () const` `[inline]`

7.8.3.13 `bool NIM::HVModuleN470Values::ChannelStatus::RampUp () const` `[inline]`

7.8.3.14 `SignalStandard NIM::HVModuleN470Values::ChannelStatus::Standard () const` `[inline]`

7.8.3.15 `bool NIM::HVModuleN470Values::ChannelStatus::Trip () const` `[inline]`

7.8.3.16 `bool NIM::HVModuleN470Values::ChannelStatus::UNV () const` `[inline]`

7.8.3.17 `bool NIM::HVModuleN470Values::ChannelStatus::Vsel () const` `[inline]`

7.8.4 Friends And Related Function Documentation

7.8.4.1 `std::ostream& operator<< (std::ostream & os, const ChannelStatus & cs)` `[friend]`

7.8.5 Field Documentation

7.8.5.1 `unsigned short NIM::HVModuleN470Values::ChannelStatus::fWord` `[private]`

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

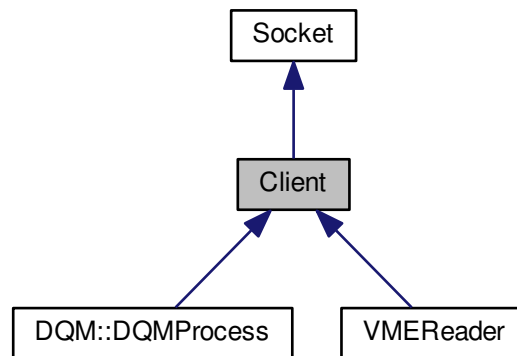
- `include/NIM_HVModuleN470.h`

7.9 Client Class Reference

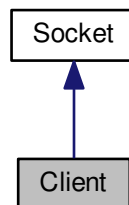
Base client object for the socket.

```
#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.

- [SocketMessage Receive](#) (const MessageKey &key)
- 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

7.9.1 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

7.9.2 Constructor & Destructor Documentation

7.9.2.1 [Client::Client](#) () [inline]

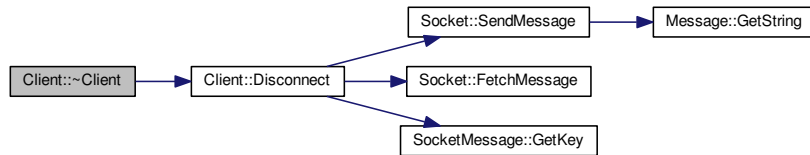
General void client constructor.

7.9.2.2 [Client::Client](#) (int *port*)

Bind a socket client to a given port.

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

Here is the call graph for this function:

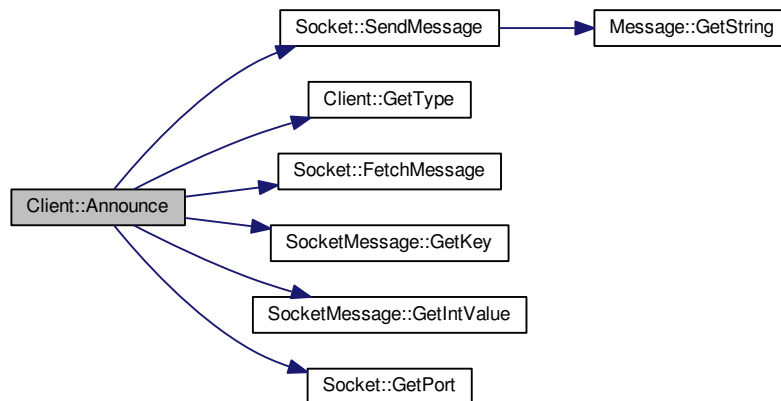


7.9.3 Member Function Documentation

7.9.3.1 void Client::Announce () [private]

Announce our entry on the socket to its master.

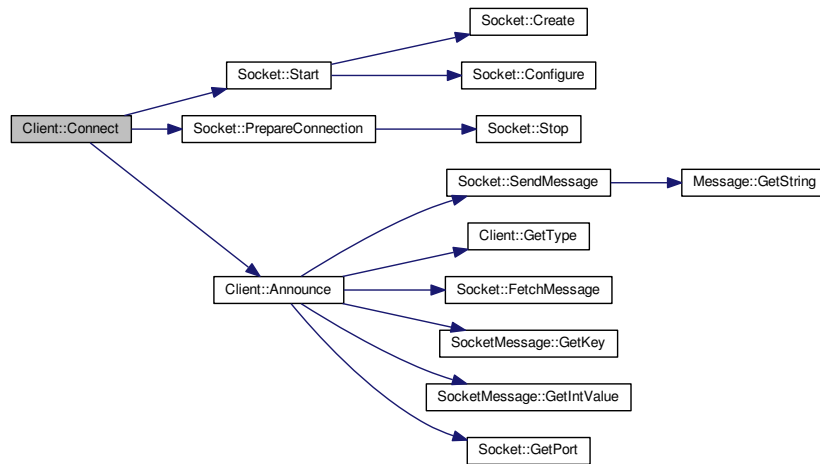
Here is the call graph for this function:



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

Bind this client to the socket.

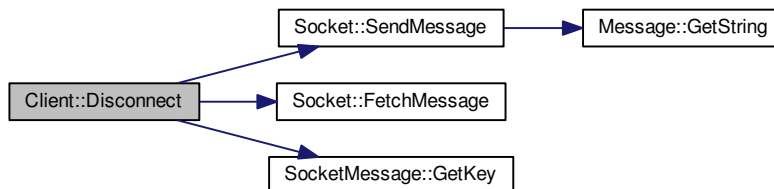
Here is the call graph for this function:



7.9.3.3 void Client::Disconnect ()

Unbind this client from the socket.

Here is the call graph for this function:



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

`Socket` actor type retrieval method.

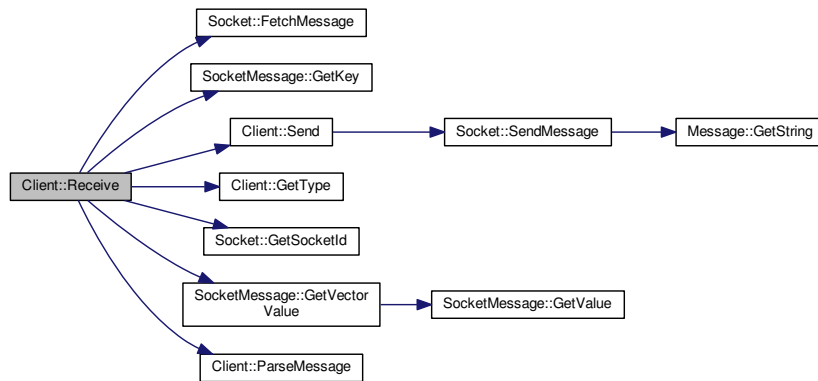
7.9.3.5 virtual void Client::ParseMessage (const SocketMessage & m) [inline],[virtual]

Parse a `SocketMessage` received from the master.

7.9.3.6 void Client::Receive ()

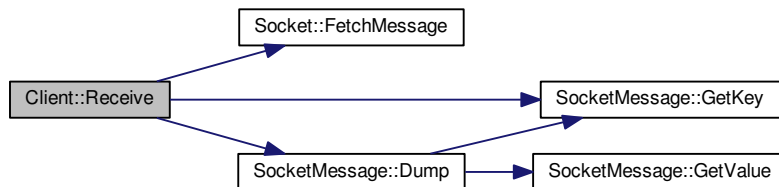
Receive a socket message from the master.

Here is the call graph for this function:



7.9.3.7 SocketMessage Client::Receive (const MessageKey & key)

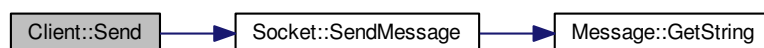
Here is the call graph for this function:



7.9.3.8 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:



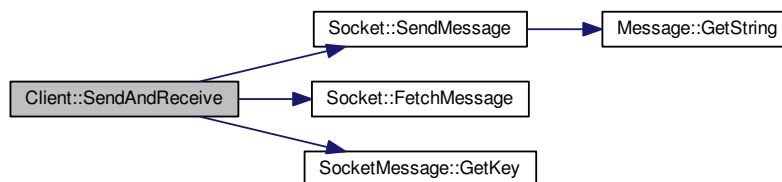
7.9.3.9 `void Client::Send (const Exception & e) const` `[inline]`

Here is the call graph for this function:



7.9.3.10 `SocketMessage Client::SendAndReceive (const SocketMessage & m, const MessageKey & a) const` `[inline]`

Here is the call graph for this function:



7.9.4 Field Documentation

7.9.4.1 `int Client::fClientId` `[private]`

7.9.4.2 `bool Client::fIsConnected` `[private]`

7.9.4.3 `SocketType Client::fType` `[private]`

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

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

7.10 DQM::GastofCanvas::Coord Struct Reference

Data Fields

- unsigned int [x](#)
- unsigned int [y](#)

7.10.1 Field Documentation

7.10.1.1 unsigned int DQM::GastofCanvas::Coord::x

7.10.1.2 unsigned int DQM::GastofCanvas::Coord::y

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

- include/GastofCanvas.h

7.11 DQM::QuarticCanvas::Coord Struct Reference

Data Fields

- unsigned int [x](#)
- unsigned int [y](#)

7.11.1 Field Documentation

7.11.1.1 unsigned int DQM::QuarticCanvas::Coord::x

7.11.1.2 unsigned int DQM::QuarticCanvas::Coord::y

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

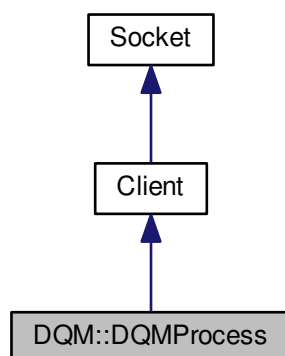
- include/QuarticCanvas.h

7.12 DQM::DQMProcess Class Reference

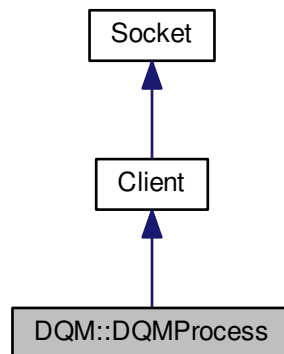
Handler for a common [DQM](#) process to run on the socket.

```
#include <DQMProcess.h>
```

Inheritance diagram for DQM::DQMProcess:



Collaboration diagram for DQM::DQMProcess:



Public Types

- enum [Action](#) { [NewPlot](#) = 0x0, [UpdatedPlot](#) = 0x1 }

Public Member Functions

- [DQMProcess](#) (int port, unsigned short order=0, const char *det_type="")
- [~DQMProcess](#) ()
- void [Run](#) (bool(*fcn)(unsigned int addr, std::string filename, std::vector< std::string > *outputs), const [Action](#) &act=[NewPlot](#))
Run a [DQM](#) plotter making use of the board/output filename information.
- void [Run](#) (bool(*fcn)(std::vector< std::string > *outputs), const [Action](#) &act=[NewPlot](#))
Run a [DQM](#) plotter without any information on the board/output filename.

Private Member Functions

- int [ParseMessage](#) (uint32_t *board_address, std::string *filename)
- bool [IsInRun](#) ()

Private Attributes

- unsigned short [fOrder](#)
- unsigned int [fRunNumber](#)
- std::string [fDetectorType](#)
- std::map< unsigned long, std::string > [fAddressesCanProcess](#)

Additional Inherited Members

7.12.1 Detailed Description

Handler for a common [DQM](#) process to run on the socket.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

3 Aug 2015

7.12.2 Member Enumeration Documentation

7.12.2.1 enum DQM::DQMProcess::Action

Enumerator

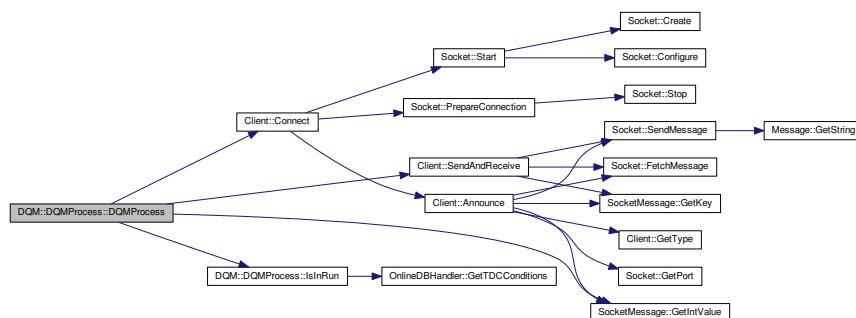
NewPlot

UpdatedPlot

7.12.3 Constructor & Destructor Documentation

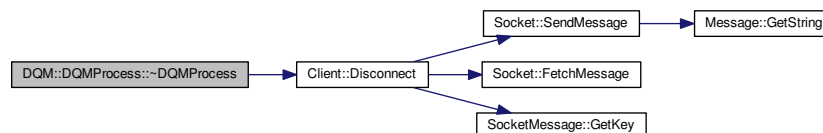
7.12.3.1 DQM::DQMProcess::DQMProcess (int port, unsigned short order = 0, const char * det_type = " ") [inline]

Here is the call graph for this function:



7.12.3.2 DQM::DQMProcess::~~DQMProcess () [inline]

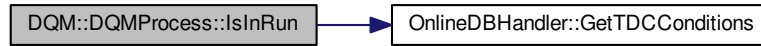
Here is the call graph for this function:



7.12.4 Member Function Documentation

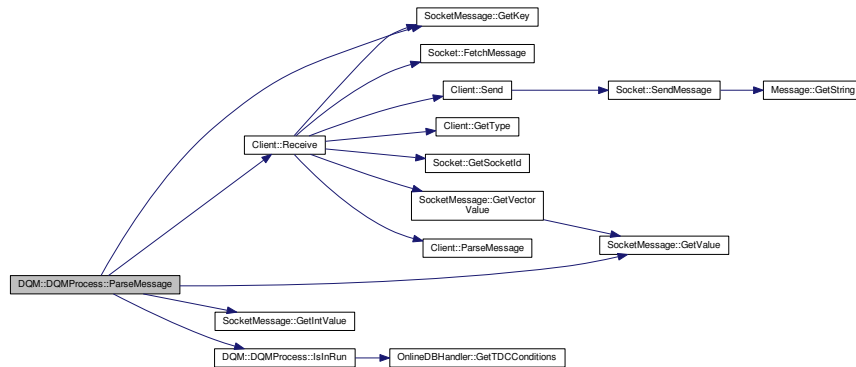
7.12.4.1 `bool DQM::DQMProcess::IsInRun () [inline],[private]`

Here is the call graph for this function:



7.12.4.2 `int DQM::DQMProcess::ParseMessage (uint32_t * board_address, std::string * filename) [inline],[private]`

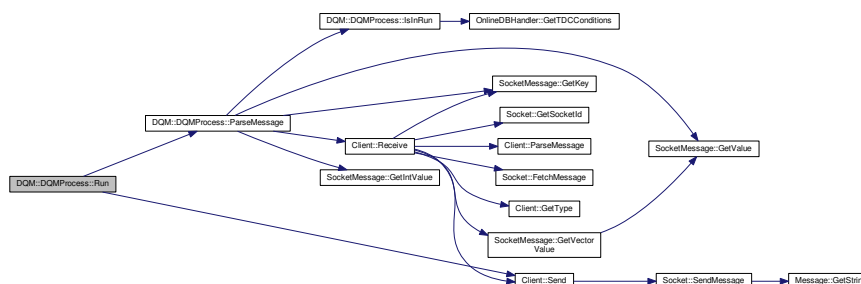
Here is the call graph for this function:



7.12.4.3 `void DQM::DQMProcess::Run (bool(*) (unsigned int addr, std::string filename, std::vector< std::string > *outputs) fcn, const Action & act = NewPlot) [inline]`

Run a [DQM](#) plotter making use of the board/output filename information.

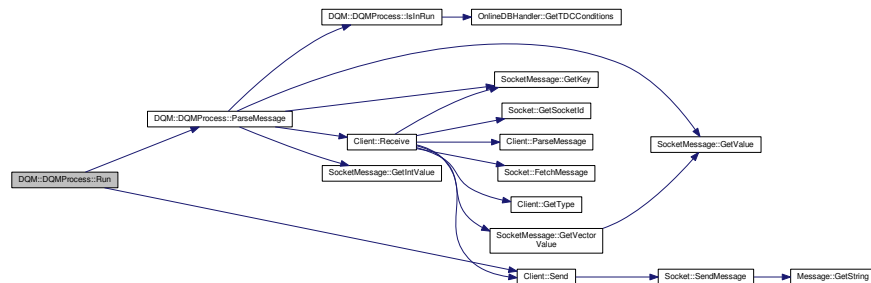
Here is the call graph for this function:



7.12.4.4 `void DQM::DQMProcess::Run (bool(*) (std::vector< std::string > *outputs) fcn, const Action & act = NewPlot)`
`[inline]`

Run a [DQM](#) plotter without any information on the board/output filename.

Here is the call graph for this function:



7.12.5 Field Documentation

7.12.5.1 `std::map< unsigned long, std::string > DQM::DQMProcess::fAddressesCanProcess` `[private]`

7.12.5.2 `std::string DQM::DQMProcess::fDetectorType` `[private]`

7.12.5.3 `unsigned short DQM::DQMProcess::fOrder` `[private]`

7.12.5.4 `unsigned int DQM::DQMProcess::fRunNumber` `[private]`

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

- `include/DQMProcess.h`

7.13 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`

7.13.1 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

7.13.2 Field Documentation

7.13.2.1 **VME::AcquisitionMode** file_header_t::acq_mode

7.13.2.2 **VME::DetectionMode** file_header_t::det_mode

7.13.2.3 uint32_t file_header_t::magic

7.13.2.4 uint8_t file_header_t::num_hptdc

7.13.2.5 uint32_t file_header_t::run_id

7.13.2.6 uint32_t file_header_t::spill_id

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

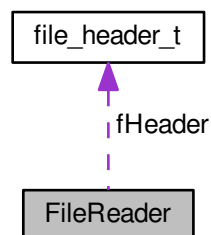
- include/FileConstants.h

7.14 FileReader Class Reference

Handler for a TDC output file readout.

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

Collaboration diagram for FileReader:



Public Member Functions

- [FileReader](#) ()
 - [FileReader](#) (std::string name)
- Class constructor.*
- [~FileReader](#) ()

- void [Open](#) (std::string name)
- bool [IsOpen](#) () const
- void [Clear](#) ()
- void [Dump](#) () const
- unsigned int [GetNumTDCs](#) () const
- unsigned int [GetRunId](#) () const
- unsigned int [GetBurstId](#) () const
- unsigned int [GetAcquisitionMode](#) () const
- unsigned int [GetDetectionMode](#) () 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](#)

7.14.1 Detailed Description

Handler for a TDC output file readout.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

Jun 2015

7.14.2 Constructor & Destructor Documentation

7.14.2.1 FileReader::FileReader () [inline]

7.14.2.2 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)

Here is the call graph for this function:



7.14.2.3 FileReader::~FileReader ()

7.14.3 Member Function Documentation

7.14.3.1 void FileReader::Clear () [inline]

7.14.3.2 void FileReader::Dump () const

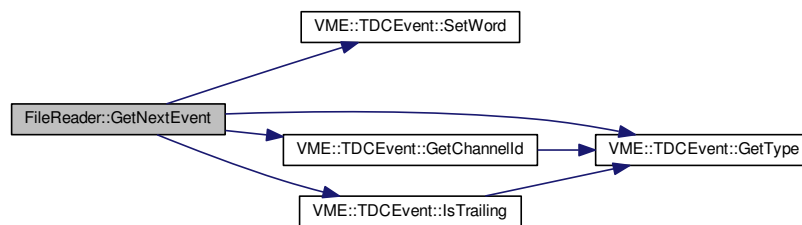
7.14.3.3 unsigned int FileReader::GetAcquisitionMode () const [inline]

7.14.3.4 unsigned int FileReader::GetBurstId () const [inline]

7.14.3.5 unsigned int FileReader::GetDetectionMode () const [inline]

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

Here is the call graph for this function:



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

Fetch the next full measurement on a given channel.

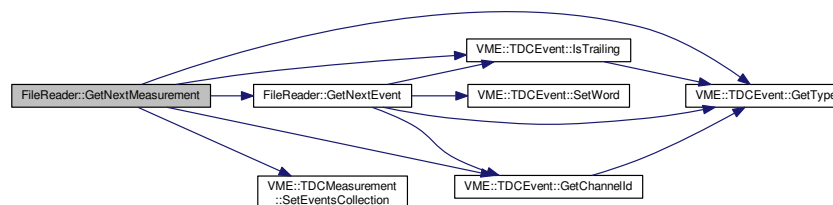
Parameters

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:



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

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

7.14.3.10 unsigned int FileReader::GetRunId () const [inline]

7.14.3.11 bool FileReader::IsOpen () const [inline]

7.14.3.12 void FileReader::Open (std::string name)

7.14.4 Field Documentation

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

7.14.4.2 file_header_t FileReader::fHeader [private]

7.14.4.3 unsigned long FileReader::fNumEvents [private]

7.14.4.4 VME::AcquisitionMode FileReader::fReadoutMode [private]

7.14.4.5 time_t FileReader::fWriteTime [private]

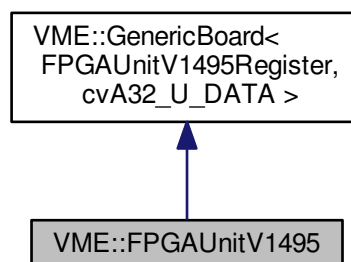
The documentation for this class was generated from the following files:

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

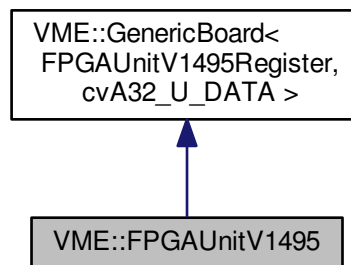
7.15 VME::FPGAUnitV1495 Class Reference

```
#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 [GetThresholdVoltage](#) (uint32_t tdc_number) const
Retrieve the threshold voltage.
- void [SetThresholdVoltage](#) (uint32_t voltage, uint32_t tdc_number) const
Set the threshold voltage.
- 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.
- void [SetTDCControlFanout](#) (bool sw=true)
Is this FPGA board used as a mean to propagate the control signal to HPTDCs?
- bool [IsTDCControlFanout](#) () const
Is this FPGA board used as a mean to propagate the control signal to HPTDCs?

Private Attributes

- bool [fScalerStarted](#)
- bool [flsTDCControlFanout](#)

Additional Inherited Members

7.15.1 Detailed Description

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

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

25 Jun 2015

7.15.2 Member Enumeration Documentation

7.15.2.1 enum VME::FPGAUnitV1495::TDCBits

Enumerator

kReset

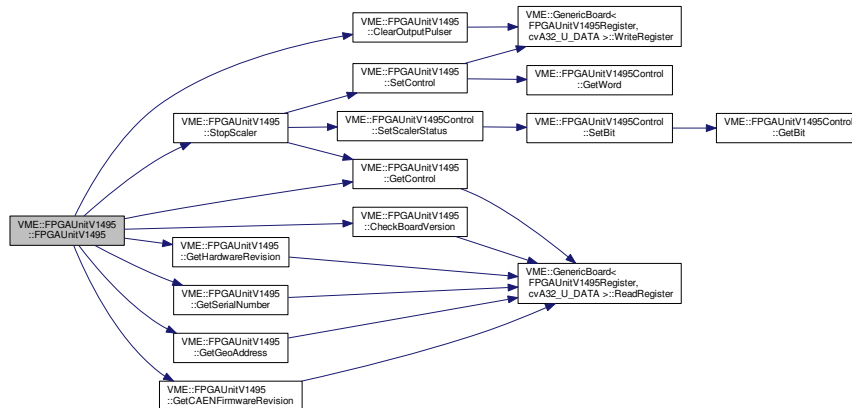
kTrigger

kClear

7.15.3 Constructor & Destructor Documentation

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

Here is the call graph for this function:

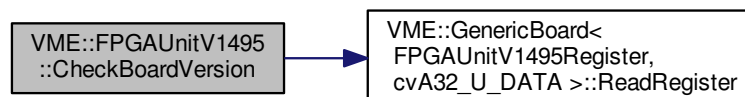


7.15.3.2 VME::FPGAUnitV1495::~~FPGAUnitV1495 ()

7.15.4 Member Function Documentation

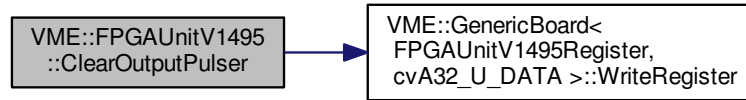
7.15.4.1 void VME::FPGAUnitV1495::CheckBoardVersion () const

Here is the call graph for this function:



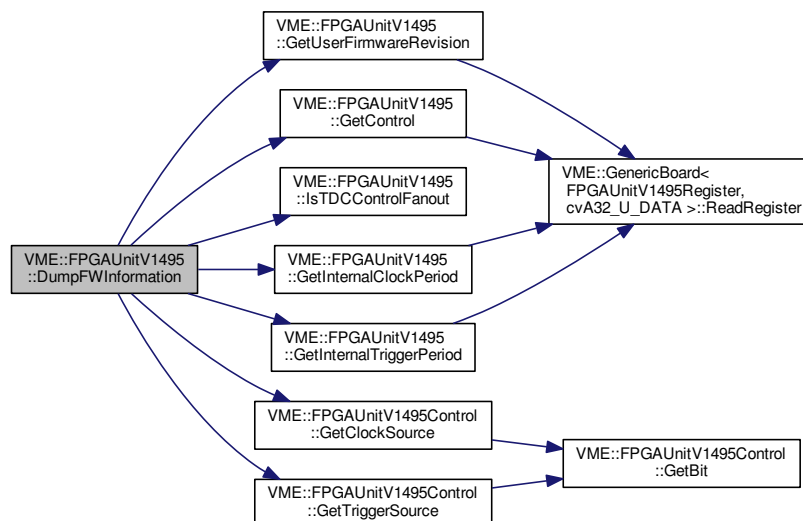
7.15.4.2 void VME::FPGAUnitV1495::ClearOutputPulser () const

Here is the call graph for this function:



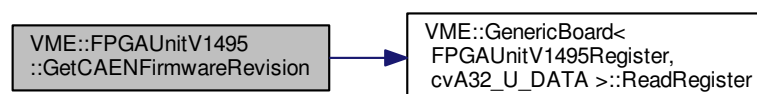
7.15.4.3 void VME::FPGAUnitV1495::DumpFWInformation () const

Here is the call graph for this function:



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

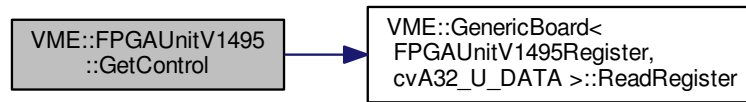
Here is the call graph for this function:



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

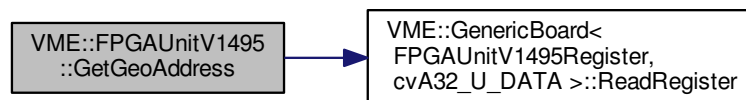
Retrieve the user-defined control word.

Here is the call graph for this function:



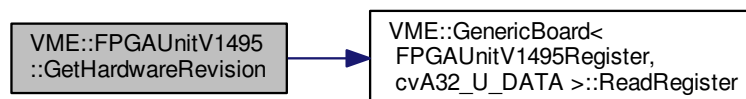
7.15.4.6 unsigned short VME::FPGAUnitV1495::GetGeoAddress () const

Here is the call graph for this function:



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

Here is the call graph for this function:



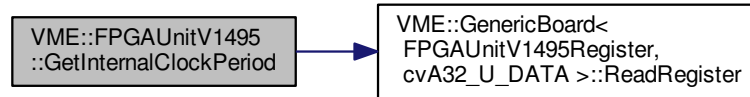
7.15.4.8 uint32_t VME::FPGAUnitV1495::GetInternalClockPeriod () const

Retrieve the internal clock period.

Returns

Clock period (in units of 25 ns)

Here is the call graph for this function:

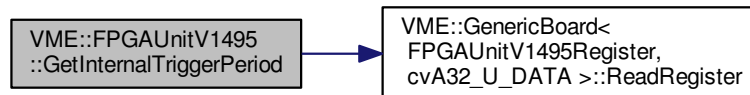
**7.15.4.9 uint32_t VME::FPGAUnitV1495::GetInternalTriggerPeriod () const**

Retrieve the internal trigger period.

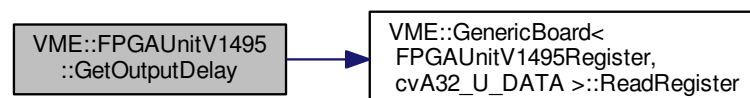
Returns

Trigger period (in units of 50 ns)

Here is the call graph for this function:

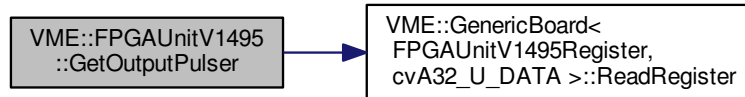
**7.15.4.10 uint32_t VME::FPGAUnitV1495::GetOutputDelay () const**

Here is the call graph for this function:



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

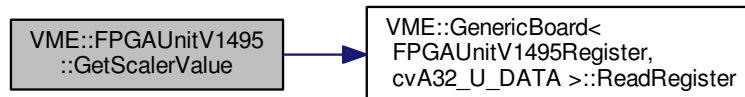
Here is the call graph for this function:



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

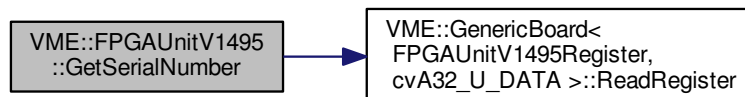
Return the inner triggers counter value.

Here is the call graph for this function:



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

Here is the call graph for this function:



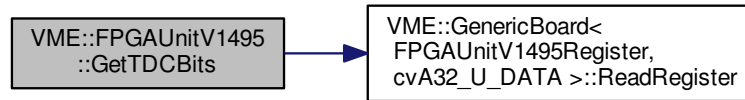
7.15.4.14 unsigned short VME::FPGAUnitV1495::GetTDCBits () const

Retrieve the current bits sent to TDCs' front panel.

Returns

A 3-bit word Pol

Here is the call graph for this function:



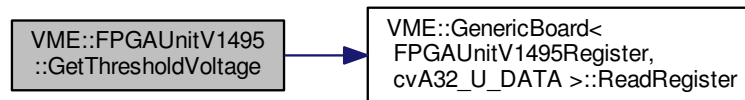
7.15.4.15 `uint32_t VME::FPGAUnitV1495::GetThresholdVoltage (uint32_t tdc_number) const`

Retrieve the threshold voltage.

Returns

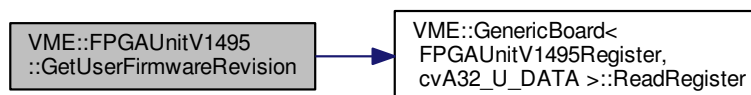
Threshold voltage (in units of 50 ns)

Here is the call graph for this function:



7.15.4.16 `unsigned short VME::FPGAUnitV1495::GetUserFirmwareRevision () const`

Here is the call graph for this function:



7.15.4.17 `bool VME::FPGAUnitV1495::IsTDCControlFanout () const` `[inline]`

Is this FPGA board used as a mean to propagate the control signal to HPTDCs?

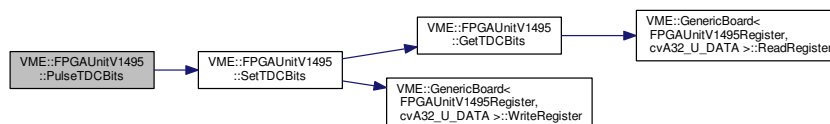
7.15.4.18 void VME::FPGAUnitV1495::PulseTDCBits (unsigned short *bits*, unsigned int *time_us* = 10) const

Send a pulse to TDCs' front panel.

Parameters

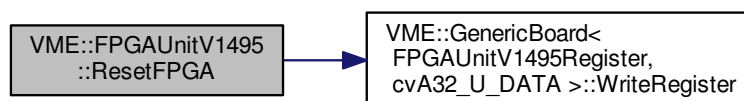
in	<i>bits</i>	The pattern to send (3 bits)
in	<i>time_us</i>	Pulse width (in us)

Here is the call graph for this function:



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

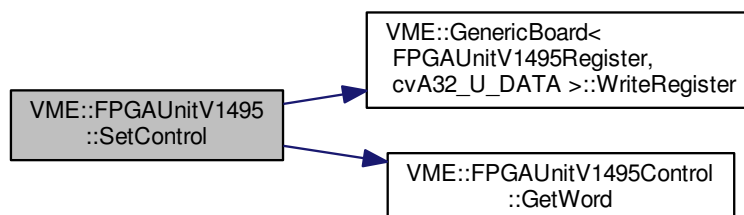
Here is the call graph for this function:



7.15.4.20 void VME::FPGAUnitV1495::SetControl (const FPGAUnitV1495Control & *control*) const

Set the user-defined control word.

Here is the call graph for this function:



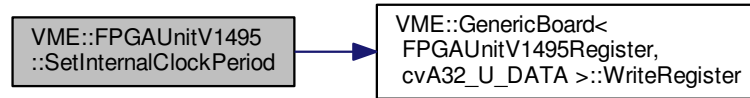
7.15.4.21 void VME::FPGAUnitV1495::SetInternalClockPeriod (uint32_t *period*) const

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:



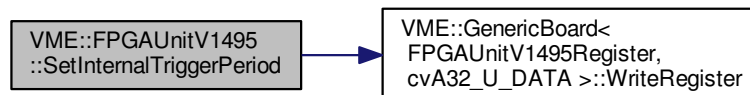
7.15.4.22 void VME::FPGAUnitV1495::SetInternalTriggerPeriod (uint32_t *period*) const

Set the internal trigger period.

Parameters

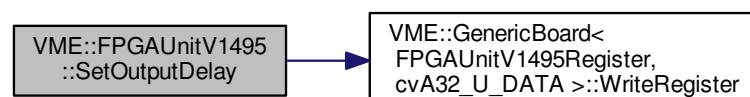
<i>in</i>	<i>period</i>	Trigger period (in units of 50 ns)
-----------	---------------	------------------------------------

Here is the call graph for this function:



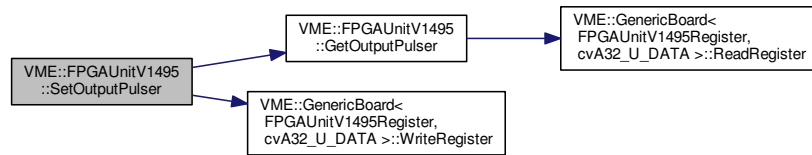
7.15.4.23 void VME::FPGAUnitV1495::SetOutputDelay (uint32_t *delay*) const

Here is the call graph for this function:



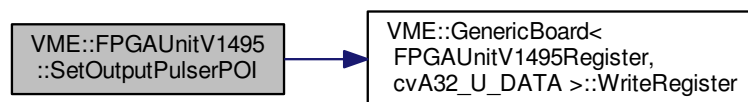
7.15.4.24 `void VME::FPGAUnitV1495::SetOutputPulser (unsigned short id, bool enable = true) const`

Here is the call graph for this function:



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

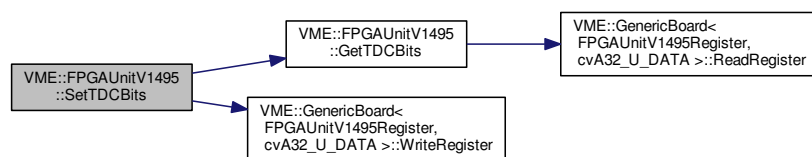
Here is the call graph for this function:



7.15.4.26 `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:



7.15.4.27 `void VME::FPGAUnitV1495::SetTDCControlFanout (bool sw = true) [inline]`

Is this FPGA board used as a mean to propagate the control signal to HPTDCs?

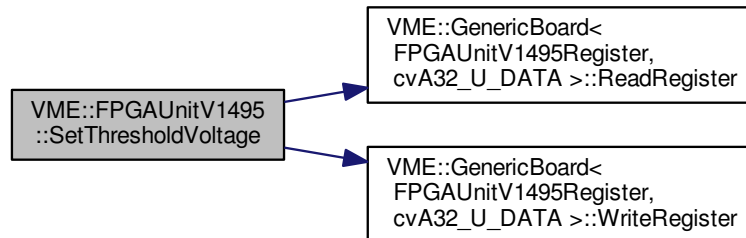
7.15.4.28 `void VME::FPGAUnitV1495::SetThresholdVoltage (uint32_t voltage, uint32_t tdc_number) const`

Set the threshold voltage.

Parameters

<i>in</i>	<i>Threshold</i>	voltage (in units of 50 ns)
-----------	------------------	-----------------------------

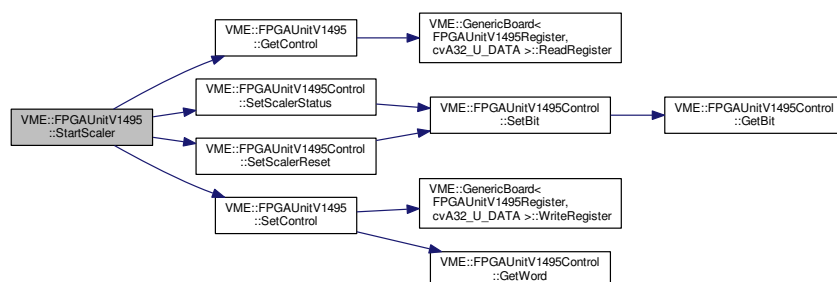
Here is the call graph for this function:



7.15.4.29 void VME::FPGAUnitV1495::StartScaler ()

Start the inner triggers counter.

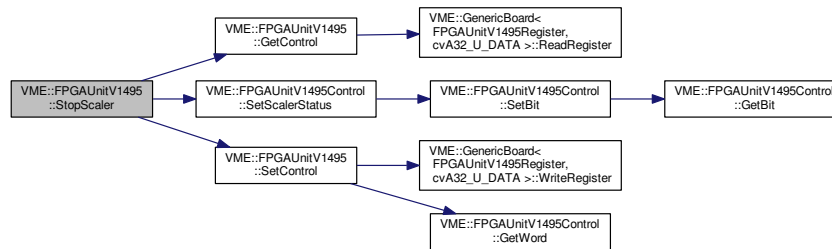
Here is the call graph for this function:



7.15.4.30 void VME::FPGAUnitV1495::StopScaler ()

Stop the inner triggers counter.

Here is the call graph for this function:



7.15.5 Field Documentation

7.15.5.1 `bool VME::FPGAUnitV1495::flsTDCCControlFanout` [private]

7.15.5.2 `bool VME::FPGAUnitV1495::fScalerStarted` [private]

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

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

7.16 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 }
- enum `TriggeringMode` { `ContinuousStorage` =0x0, `TriggerMatching` =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)
- [TriggeringMode](#) [GetTriggeringMode](#) () const
- void [SetTriggeringMode](#) (const [TriggeringMode](#) &tm)

Private Member Functions

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

Private Attributes

- uint32_t [fWord](#)

7.16.1 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

7.16.2 Member Enumeration Documentation

7.16.2.1 enum VME::FPGAUnitV1495Control::ClockSource

Enumerator

InternalClock
ExternalClock

7.16.2.2 enum VME::FPGAUnitV1495Control::SignalSource

Enumerator

InternalSignal
ExternalSignal

7.16.2.3 enum VME::FPGAUnitV1495Control::TriggeringMode

Enumerator

ContinuousStorage
TriggerMatching

7.16.2.4 enum VME::FPGAUnitV1495Control::TriggerSource

Enumerator

InternalTrigger

ExternalTrigger

7.16.3 Constructor & Destructor Documentation

7.16.3.1 VME::FPGAUnitV1495Control::FPGAUnitV1495Control (uint32_t *word*) [inline]

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

7.16.4 Member Function Documentation

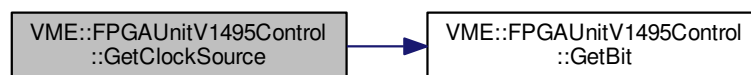
7.16.4.1 void VME::FPGAUnitV1495Control::Dump () const [inline]

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

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

Get the clock source.

Here is the call graph for this function:



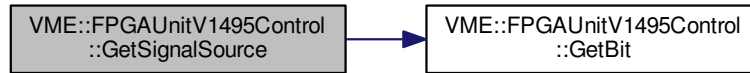
7.16.4.4 bool VME::FPGAUnitV1495Control::GetScalerStatus () const [inline]

Here is the call graph for this function:



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

Here is the call graph for this function:



7.16.4.6 TriggeringMode VME::FPGAUnitV1495Control::GetTriggeringMode () const [inline]

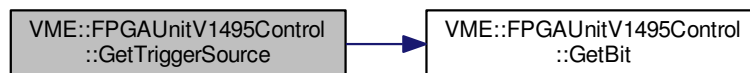
Here is the call graph for this function:



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

Get the trigger source.

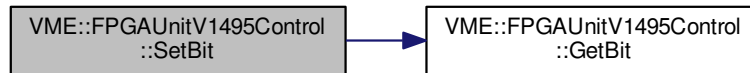
Here is the call graph for this function:



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

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

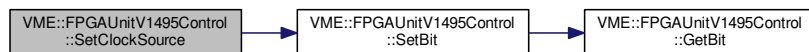
Here is the call graph for this function:



7.16.4.10 `void VME::FPGAUnitV1495Control::SetClockSource (const ClockSource & cs) [inline]`

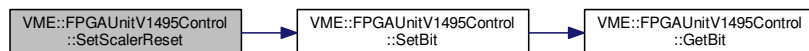
Switch between internal and external clock source.

Here is the call graph for this function:



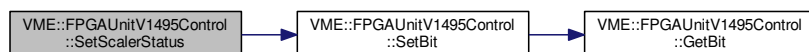
7.16.4.11 `void VME::FPGAUnitV1495Control::SetScalerReset (bool reset = true) [inline]`

Here is the call graph for this function:



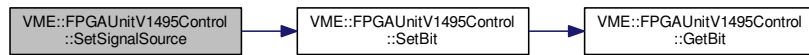
7.16.4.12 `void VME::FPGAUnitV1495Control::SetScalerStatus (bool start = true) [inline]`

Here is the call graph for this function:



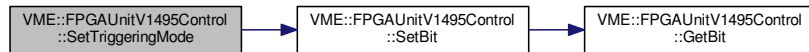
7.16.4.13 `void VME::FPGAUnitV1495Control::SetSignalSource (unsigned short map_id, const SignalSource & s)`
`[inline]`

Here is the call graph for this function:



7.16.4.14 `void VME::FPGAUnitV1495Control::SetTriggeringMode (const TriggeringMode & tm)` `[inline]`

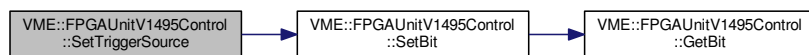
Here is the call graph for this function:



7.16.4.15 `void VME::FPGAUnitV1495Control::SetTriggerSource (const TriggerSource & cs)` `[inline]`

Switch between internal and external trigger source.

Here is the call graph for this function:



7.16.5 Field Documentation

7.16.5.1 `uint32_t VME::FPGAUnitV1495Control::fWord` `[private]`

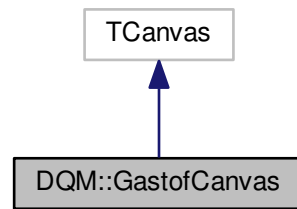
The documentation for this class was generated from the following file:

- `include/VME_FPGAUnitV1495.h`

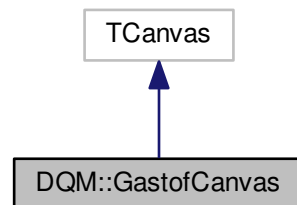
7.17 DQM::GastofCanvas Class Reference

```
#include <GastofCanvas.h>
```

Inheritance diagram for DQM::GastofCanvas:



Collaboration diagram for DQM::GastofCanvas:



Data Structures

- struct [Coord](#)

Public Member Functions

- [GastofCanvas](#) ()
- [GastofCanvas](#) (TString name, unsigned int width=500, unsigned int height=500, TString upper_label="")
- [GastofCanvas](#) (TString name, TString upper_label)
- virtual [~GastofCanvas](#) ()
- void [SetRunInfo](#) (unsigned int board_id, unsigned int run_id, unsigned int spill_id, TString date)
- void [SetUpperLabel](#) (TString text)
- void [FillChannel](#) (unsigned short nino_id, unsigned short channel_id, double content)
- TH2D * [Grid](#) ()
- void [Save](#) (TString ext="png", TString path=".")

Private Member Functions

- void [Build](#) ()
- void [DrawGrid](#) ()
- [Coord](#) [GetCoordinates](#) (unsigned short nino_id, unsigned short channel_id) const

Private Attributes

- TPad * [c1](#)
- TPad * [c2](#)
- TH2D * [fHist](#)
- double [fWidth](#)
- double [fHeight](#)
- TLegend * [fLegend](#)
- double [fLegendX](#)
- double [fLegendY](#)
- unsigned int [fLegendNumEntries](#)
- TPaveText * [fLabel1](#)
- TPaveText * [fLabel2](#)
- TPaveText * [fLabel3](#)
- TPaveText * [fLabel4](#)
- TString [fUpperLabelText](#)
- TPaveText * [fUpperLabel](#)
- bool [fLabelsDrawn](#)
- unsigned int [fBoardId](#)
- unsigned int [fRunId](#)
- unsigned int [fSpillId](#)
- TString [fRunDate](#)

7.17.1 Detailed Description

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

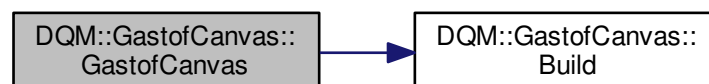
25 Jul 2015

7.17.2 Constructor & Destructor Documentation

7.17.2.1 DQM::GastofCanvas::GastofCanvas () [\[inline\]](#)

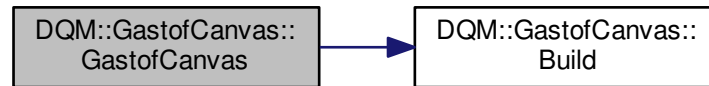
7.17.2.2 DQM::GastofCanvas::GastofCanvas (TString *name*, unsigned int *width* = 500, unsigned int *height* = 500, TString *upper_label* = " ") [\[inline\]](#)

Here is the call graph for this function:



7.17.2.3 DQM::GastofCanvas::GastofCanvas (TString *name*, TString *upper_label*) [inline]

Here is the call graph for this function:



7.17.2.4 virtual DQM::GastofCanvas::~~GastofCanvas () [inline],[virtual]

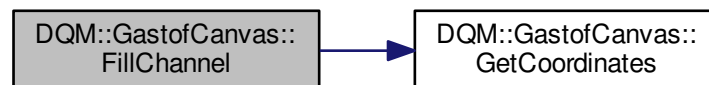
7.17.3 Member Function Documentation

7.17.3.1 void DQM::GastofCanvas::Build () [inline],[private]

7.17.3.2 void DQM::GastofCanvas::DrawGrid () [inline],[private]

7.17.3.3 void DQM::GastofCanvas::FillChannel (unsigned short *nino_id*, unsigned short *channel_id*, double *content*) [inline]

Here is the call graph for this function:

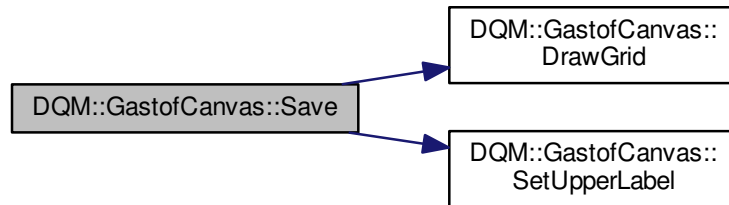


7.17.3.4 Coord DQM::GastofCanvas::GetCoordinates (unsigned short *nino_id*, unsigned short *channel_id*) const [inline],[private]

7.17.3.5 TH2D* DQM::GastofCanvas::Grid () [inline]

7.17.3.6 `void DQM::GastofCanvas::Save (TString ext = "png", TString path = " . ") [inline]`

Here is the call graph for this function:



7.17.3.7 `void DQM::GastofCanvas::SetRunInfo (unsigned int board_id, unsigned int run_id, unsigned int spill_id, TString date) [inline]`

7.17.3.8 `void DQM::GastofCanvas::SetUpperLabel (TString text) [inline]`

7.17.4 Field Documentation

7.17.4.1 `TPad* DQM::GastofCanvas::c1 [private]`

7.17.4.2 `TPad * DQM::GastofCanvas::c2 [private]`

7.17.4.3 `unsigned int DQM::GastofCanvas::fBoardId [private]`

7.17.4.4 `double DQM::GastofCanvas::fHeight [private]`

7.17.4.5 `TH2D* DQM::GastofCanvas::fHist [private]`

7.17.4.6 `TPaveText* DQM::GastofCanvas::fLabel1 [private]`

7.17.4.7 `TPaveText * DQM::GastofCanvas::fLabel2 [private]`

7.17.4.8 `TPaveText * DQM::GastofCanvas::fLabel3 [private]`

7.17.4.9 `TPaveText * DQM::GastofCanvas::fLabel4 [private]`

7.17.4.10 `bool DQM::GastofCanvas::fLabelsDrawn [private]`

7.17.4.11 `TLegend* DQM::GastofCanvas::fLegend [private]`

7.17.4.12 `unsigned int DQM::GastofCanvas::fLegendNumEntries [private]`

7.17.4.13 `double DQM::GastofCanvas::fLegendX [private]`

7.17.4.14 `double DQM::GastofCanvas::fLegendY [private]`

7.17.4.15 `TString DQM::GastofCanvas::fRunDate [private]`

- 7.17.4.16 unsigned int DQM::GastofCanvas::fRunId [private]
- 7.17.4.17 unsigned int DQM::GastofCanvas::fSpillId [private]
- 7.17.4.18 TPaveText* DQM::GastofCanvas::fUpperLabel [private]
- 7.17.4.19 TString DQM::GastofCanvas::fUpperLabelText [private]
- 7.17.4.20 double DQM::GastofCanvas::fWidth [private]

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

- include/GastofCanvas.h

7.18 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](#)

7.18.1 Constructor & Destructor Documentation

- 7.18.1.1 `template<class Register, CVAddressModifier am> VME::GenericBoard< Register, am >::GenericBoard (int32_t bhandle, uint32_t baseaddr) [inline]`
- 7.18.1.2 `template<class Register, CVAddressModifier am> virtual VME::GenericBoard< Register, am >::~~GenericBoard () [inline],[virtual]`

7.18.2 Member Function Documentation

7.18.2.1 `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

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

7.18.2.2 `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

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

7.18.2.3 `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

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

7.18.2.4 `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

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

7.18.3 Field Documentation

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

7.18.3.2 `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`

7.19 VME::GlobalOffset Struct Reference

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

Data Fields

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

7.19.1 Field Documentation

7.19.1.1 `uint16_t` [VME::GlobalOffset::coarse](#)

7.19.1.2 `uint16_t` [VME::GlobalOffset::fine](#)

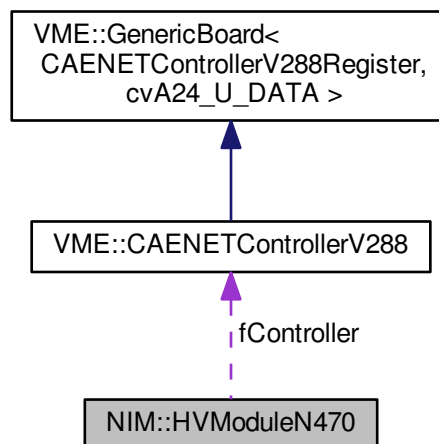
The documentation for this struct was generated from the following file:

- `include/VME_TDCV1x90.h`

7.20 NIM::HVModuleN470 Class Reference

```
#include <NIM_HVModuleN470.h>
```

Collaboration diagram for NIM::HVModuleN470:



Public Member Functions

- [HVModuleN470](#) (`uint16_t` addr, [VME::CAENETControllerV288](#) &cont)

- `~HVMModuleN470 ()`
- `std::string GetModuleId () const`
- `unsigned short GetFWRevision () const`
- `HVMModuleN470Values ReadMonitoringValues () const`
- `HVMModuleN470ChannelValues ReadChannelValues (unsigned short ch_id) const`
- `void SetChannelV0 (unsigned short ch_id, unsigned short v0) const`
- `void SetChannelI0 (unsigned short ch_id, unsigned short i0) const`
- `void SetChannelV1 (unsigned short ch_id, unsigned short v1) const`
- `void SetChannelI1 (unsigned short ch_id, unsigned short i1) const`
- `void EnableChannel (unsigned short ch_id) const`
- `void DisableChannel (unsigned short ch_id) const`

Private Member Functions

- `void ReadRegister (const HVMModuleN470Opcodes ®, std::vector< uint16_t > *data, unsigned int num←_words=1) const`
Read in register.
- `void WriteRegister (const HVMModuleN470Opcodes ®, const std::vector< uint16_t > &data) const`
Write on register.
- `void WriteRegister (const HVMModuleN470Opcodes ®, const uint16_t &data) const`
Write on register.

Private Attributes

- `VME::CAENETControllerV288 fController`
- `uint16_t fAddress`

7.20.1 Constructor & Destructor Documentation

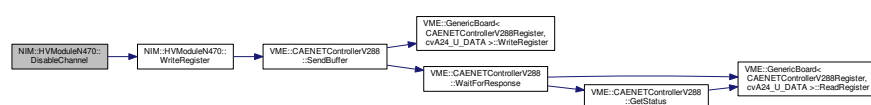
7.20.1.1 `NIM::HVMModuleN470::HVMModuleN470 (uint16_t addr, VME::CAENETControllerV288 & cont)`

7.20.1.2 `NIM::HVMModuleN470::~~HVMModuleN470 () [inline]`

7.20.2 Member Function Documentation

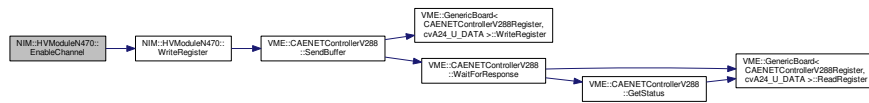
7.20.2.1 `void NIM::HVMModuleN470::DisableChannel (unsigned short ch_id) const`

Here is the call graph for this function:



7.20.2.2 void NIM::HVModuleN470::EnableChannel (unsigned short *ch_id*) const

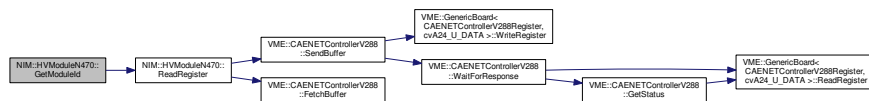
Here is the call graph for this function:



7.20.2.3 unsigned short NIM::HVModuleN470::GetFWRevision () const

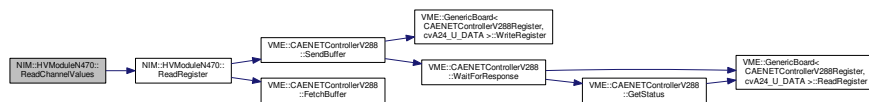
7.20.2.4 std::string NIM::HVModuleN470::GetModuleId () const

Here is the call graph for this function:



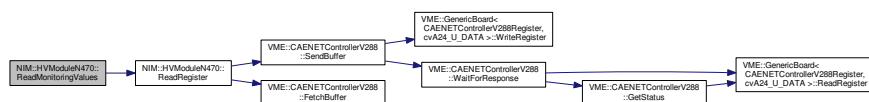
7.20.2.5 HVModuleN470ChannelValues NIM::HVModuleN470::ReadChannelValues (unsigned short *ch_id*) const

Here is the call graph for this function:



7.20.2.6 HVModuleN470Values NIM::HVModuleN470::ReadMonitoringValues () const

Here is the call graph for this function:



7.20.2.7 void NIM::HVModuleN470::ReadRegister (const HVModuleN470Opcodes & *reg*, std::vector< uint16_t > * *data*, unsigned int *num_words* = 1) const [private]

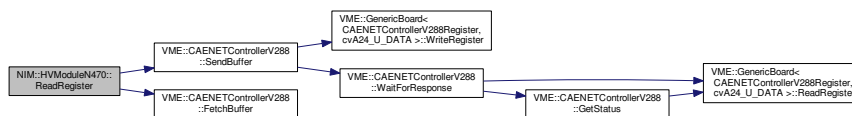
Read in register.

Read a vector of 16-bit words in the register

Parameters

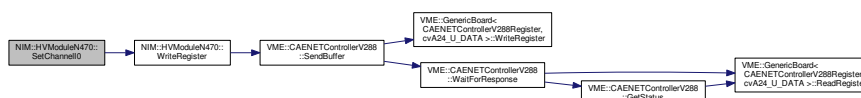
in	<i>addr</i>	register
out	<i>vector</i>	of data words

Here is the call graph for this function:



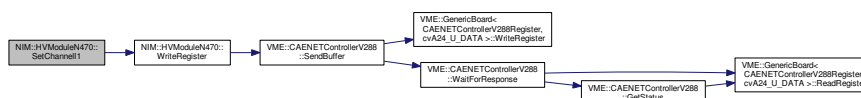
7.20.2.8 void NIM::HVModuleN470::SetChannelI0 (unsigned short *ch_id*, unsigned short *i0*) const

Here is the call graph for this function:



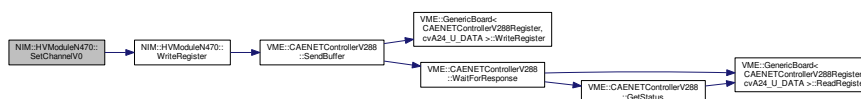
7.20.2.9 void NIM::HVModuleN470::SetChannelI1 (unsigned short *ch_id*, unsigned short *i1*) const

Here is the call graph for this function:



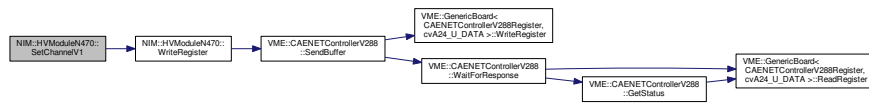
7.20.2.10 void NIM::HVModuleN470::SetChannelV0 (unsigned short *ch_id*, unsigned short *v0*) const

Here is the call graph for this function:



7.20.2.11 void NIM::HVModuleN470::SetChannelV1 (unsigned short *ch_id*, unsigned short *v1*) const

Here is the call graph for this function:



7.20.2.12 void NIM::HVModuleN470::WriteRegister (const HVModuleN470Opcodes & *reg*, const std::vector< uint16_t > & *data*) const [private]

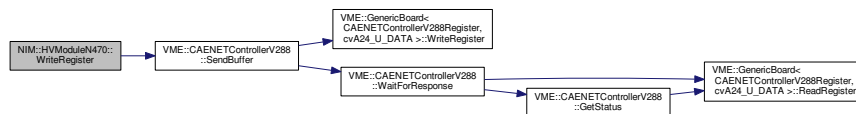
Write on register.

Write a vector of 16-bit words in the register

Parameters

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

Here is the call graph for this function:



7.20.2.13 void NIM::HVModuleN470::WriteRegister (const HVModuleN470Opcodes & *reg*, const uint16_t & *data*) const [private]

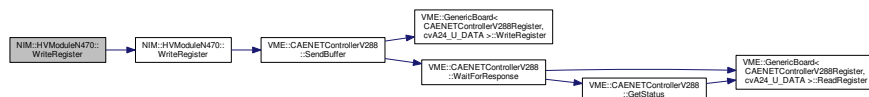
Write on register.

Write a 16-bit word in the register

Parameters

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

Here is the call graph for this function:



7.20.3 Field Documentation

7.20.3.1 uint16_t NIM::HVModuleN470::fAddress [private]

7.20.3.2 VME::CAENETControllerV288 NIM::HVModuleN470::fController [private]

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

- include/NIM_HVModuleN470.h
- src/NIM_HVModuleN470.cpp

7.21 NIM::HVModuleN470ChannelValues Class Reference

Single channel monitoring values for the HV power supply.

```
#include <NIM_HVModuleN470.h>
```

Public Member Functions

- [HVModuleN470ChannelValues](#) (unsigned short ch_id, std::vector< unsigned short > vals)
- [~HVModuleN470ChannelValues](#) ()
- void [Dump](#) () const
- unsigned short [ChannelStatus](#) () const
- unsigned short [Vmon](#) () const
- unsigned short [Imon](#) () const
- unsigned short [V0](#) () const
- unsigned short [I0](#) () const
- unsigned short [V1](#) () const
- unsigned short [I1](#) () const
- unsigned short [Trip](#) () const
- unsigned short [RampUp](#) () const
- unsigned short [RampDown](#) () const
- unsigned short [MaxV](#) () const

Private Attributes

- unsigned short [fChannelId](#)
- std::vector< unsigned short > [fValues](#)

7.21.1 Detailed Description

Single channel monitoring values for the HV power supply.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

24 Jul 2015

7.21.2 Constructor & Destructor Documentation

7.21.2.1 NIM::HVModuleN470ChannelValues::HVModuleN470ChannelValues (unsigned short *ch_id*, std::vector< unsigned short > *vals*) [inline]

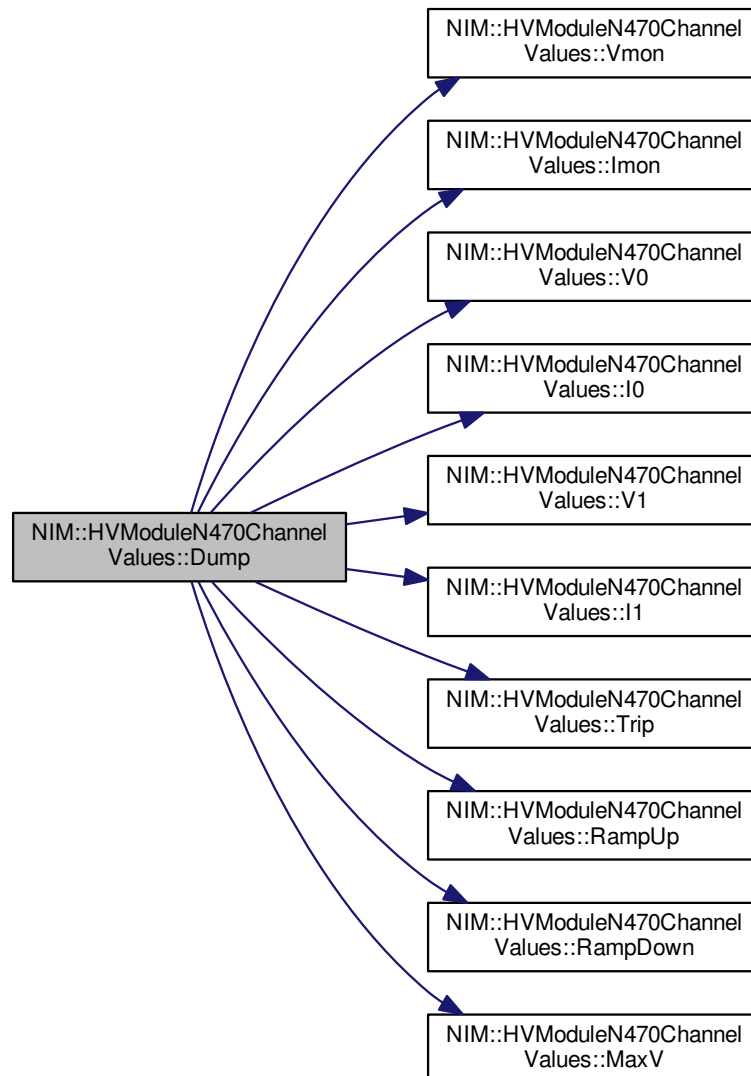
7.21.2.2 NIM::HVModuleN470ChannelValues::~~HVModuleN470ChannelValues () [inline]

7.21.3 Member Function Documentation

7.21.3.1 unsigned short NIM::HVModuleN470ChannelValues::ChannelStatus () const [inline]

7.21.3.2 void NIM::HVModuleN470ChannelValues::Dump () const [inline]

Here is the call graph for this function:



- 7.21.3.3 unsigned short NIM::HVModuleN470ChannelValues::l0 () const [inline]
- 7.21.3.4 unsigned short NIM::HVModuleN470ChannelValues::l1 () const [inline]
- 7.21.3.5 unsigned short NIM::HVModuleN470ChannelValues::lmon () const [inline]
- 7.21.3.6 unsigned short NIM::HVModuleN470ChannelValues::MaxV () const [inline]
- 7.21.3.7 unsigned short NIM::HVModuleN470ChannelValues::RampDown () const [inline]
- 7.21.3.8 unsigned short NIM::HVModuleN470ChannelValues::RampUp () const [inline]
- 7.21.3.9 unsigned short NIM::HVModuleN470ChannelValues::Trip () const [inline]
- 7.21.3.10 unsigned short NIM::HVModuleN470ChannelValues::V0 () const [inline]
- 7.21.3.11 unsigned short NIM::HVModuleN470ChannelValues::V1 () const [inline]
- 7.21.3.12 unsigned short NIM::HVModuleN470ChannelValues::Vmon () const [inline]

7.21.4 Field Documentation

- 7.21.4.1 unsigned short NIM::HVModuleN470ChannelValues::fChannelId [private]
- 7.21.4.2 std::vector<unsigned short> NIM::HVModuleN470ChannelValues::fValues [private]

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

- include/NIM_HVModuleN470.h

7.22 NIM::HVModuleN470Values Class Reference

General monitoring values for the HV power supply.

```
#include <NIM_HVModuleN470.h>
```

Data Structures

- class [ChannelStatus](#)

Public Member Functions

- [HVModuleN470Values](#) (std::vector< unsigned short > vals)
- [~HVModuleN470Values](#) ()
- void [Dump](#) () const
- unsigned short [Vmon](#) () const
- unsigned short [lmon](#) () const
- unsigned short [Vmax](#) () const
- [ChannelStatus GetChannelStatus](#) (unsigned short ch_id) const

Private Attributes

- std::vector< unsigned short > [fValues](#)

7.22.1 Detailed Description

General monitoring values for the HV power supply.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

24 Jul 2015

7.22.2 Constructor & Destructor Documentation

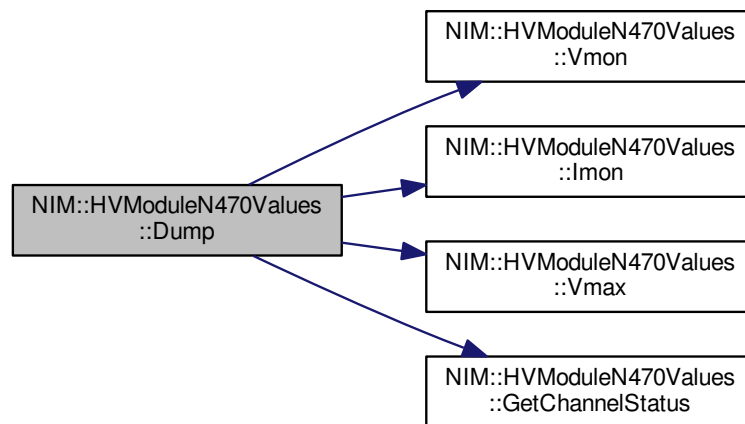
7.22.2.1 NIM::HVModuleN470Values::HVModuleN470Values (`std::vector< unsigned short > vals`) `[inline]`

7.22.2.2 NIM::HVModuleN470Values::~~HVModuleN470Values () `[inline]`

7.22.3 Member Function Documentation

7.22.3.1 void NIM::HVModuleN470Values::Dump () `const` `[inline]`

Here is the call graph for this function:



7.22.3.2 ChannelStatus NIM::HVModuleN470Values::GetChannelStatus (`unsigned short ch_id`) `const` `[inline]`

7.22.3.3 unsigned short NIM::HVModuleN470Values::Imon () `const` `[inline]`

7.22.3.4 unsigned short NIM::HVModuleN470Values::Vmax () `const` `[inline]`

7.22.3.5 unsigned short NIM::HVModuleN470Values::Vmon () `const` `[inline]`

7.22.4 Field Documentation

7.22.4.1 `std::vector<unsigned short> NIM::HVModuleN470Values::fValues` [private]

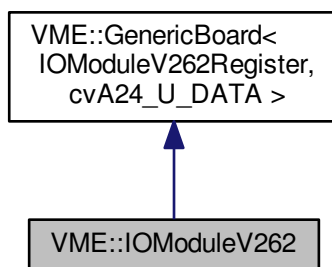
The documentation for this class was generated from the following file:

- `include/NIM_HVModuleN470.h`

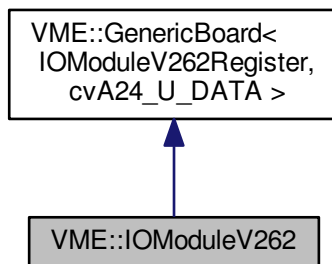
7.23 VME::IOModuleV262 Class Reference

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

Inheritance diagram for VME::IOModuleV262:



Collaboration diagram for VME::IOModuleV262:



Public Member Functions

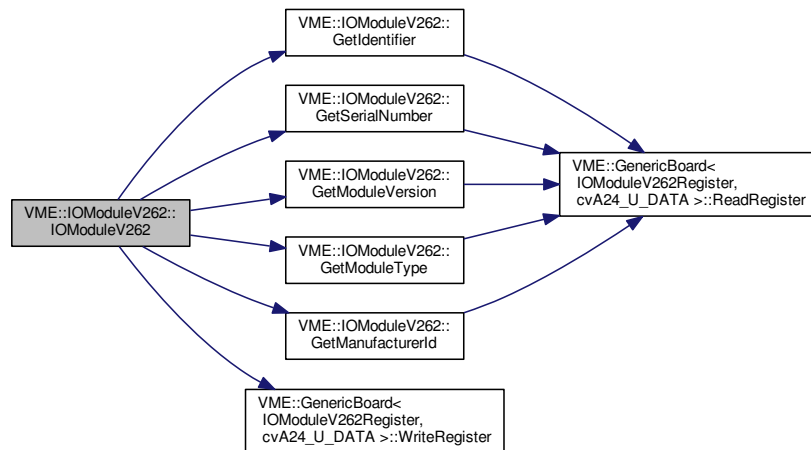
- `IOModuleV262` (`int32_t bhandle`, `uint32_t baseaddr`)
- `~IOModuleV262` ()
- `unsigned short GetSerialNumber` () const
- `unsigned short GetModuleVersion` () const
- `unsigned short GetModuleType` () const
- `unsigned short GetManufacturerId` () const
- `unsigned short GetIdentifier` () const

Additional Inherited Members

7.23.1 Constructor & Destructor Documentation

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

Here is the call graph for this function:

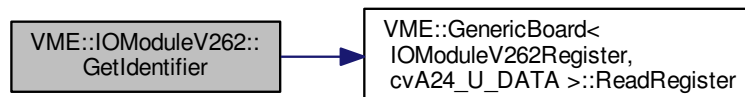


7.23.1.2 VME::IOModuleV262::~~IOModuleV262 () [inline]

7.23.2 Member Function Documentation

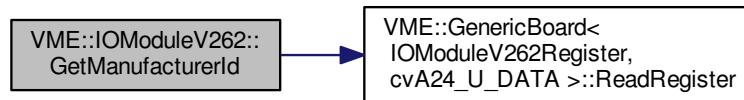
7.23.2.1 unsigned short VME::IOModuleV262::GetIdentifier () const

Here is the call graph for this function:



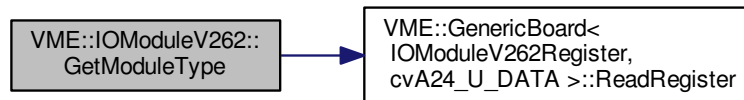
7.23.2.2 unsigned short VME::IOModuleV262::GetManufacturerId () const

Here is the call graph for this function:



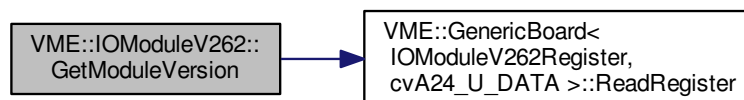
7.23.2.3 unsigned short VME::IOModuleV262::GetModuleType () const

Here is the call graph for this function:



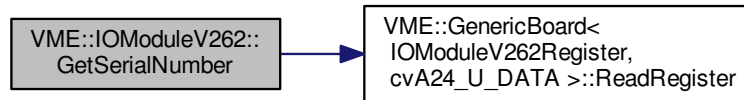
7.23.2.4 unsigned short VME::IOModuleV262::GetModuleVersion () const

Here is the call graph for this function:



7.23.2.5 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

7.24 Logger Class Reference

Redirect outputs to another output stream.

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

Public Member Functions

- [Logger](#) (std::ostream &lhs, std::ostream &rhs=std::cout)
- [~Logger](#) ()

Private Attributes

- std::ostream & [fStream](#)
- std::streambuf *const [fBuffer](#)

7.24.1 Detailed Description

Redirect outputs to another output stream.

7.24.2 Constructor & Destructor Documentation

7.24.2.1 `Logger::Logger (std::ostream & lhs, std::ostream & rhs = std::cout) [inline]`

7.24.2.2 `Logger::~~Logger () [inline]`

7.24.3 Field Documentation

7.24.3.1 `std::streambuf* const Logger::fBuffer [private]`

7.24.3.2 `std::ostream& Logger::fStream [private]`

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

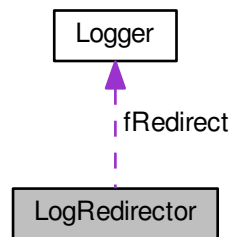
- `include/FileConstants.h`

7.25 LogRedirector Class Reference

Redirect output stream to a string.

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

Collaboration diagram for LogRedirector:



Public Member Functions

- [LogRedirector](#) (`std::ostream &stm=std::cout`)
- `std::string` [contents](#) () const

Private Attributes

- `std::ostringstream` [fSS](#)
- const [Logger](#) [fRedirect](#)

7.25.1 Detailed Description

Redirect output stream to a string.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

3 Aug 2015

7.25.2 Constructor & Destructor Documentation

7.25.2.1 `LogRedirector::LogRedirector (std::ostream & stm = std::cout) [inline]`

7.25.3 Member Function Documentation

7.25.3.1 `std::string LogRedirector::contents () const` `[inline]`

7.25.4 Field Documentation

7.25.4.1 `const Logger LogRedirector::fRedirect` `[private]`

7.25.4.2 `std::ostringstream LogRedirector::fSS` `[private]`

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

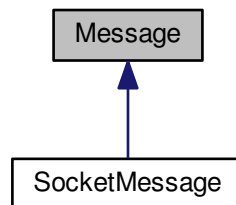
- `include/FileConstants.h`

7.26 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`

7.26.1 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

7.26.2 Constructor & Destructor Documentation

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

Void message constructor.

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

Construct a message from a string.

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

Construct a message from a string.

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

7.26.3 Member Function Documentation

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

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

Placeholder for the MessageKey retrieval method.

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

Retrieve the string carried by this message as a whole.

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

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

7.26.4 Field Documentation

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

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

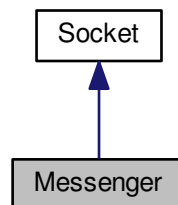
- include/Message.h

7.27 Messenger Class Reference

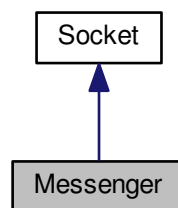
Base master object for the socket.

```
#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 [SendAll](#) (const [Socket::SocketType](#) &type, const [Message](#) &m) const
Send any type of message to all clients of one type.
- void [SendAll](#) (const [Socket::SocketType](#) &type, const [Exception](#) &e) const

- 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

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

Additional Inherited Members

7.27.1 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

7.27.2 Constructor & Destructor Documentation

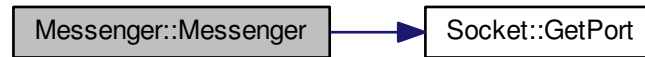
7.27.2.1 [Messenger::Messenger](#) ()

Build a void master object or socket actor.

7.27.2.2 Messenger::Messenger (int port)

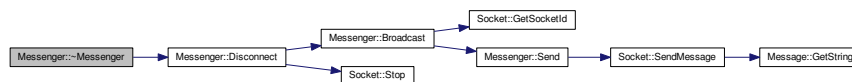
Build a master object to control the socket.

Here is the call graph for this function:



7.27.2.3 Messenger::~~Messenger ()

Here is the call graph for this function:



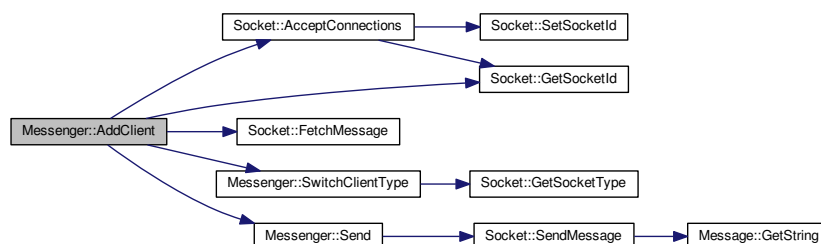
7.27.3 Member Function Documentation

7.27.3.1 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:



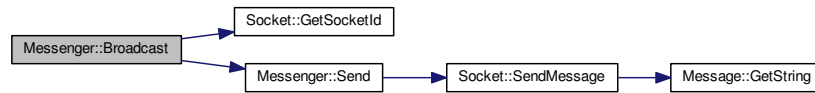
7.27.3.2 void Messenger::Broadcast (const Message & m) const

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:

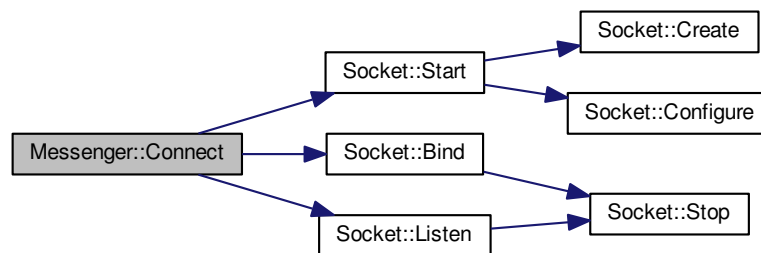


7.27.3.3 bool Messenger::Connect ()

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:

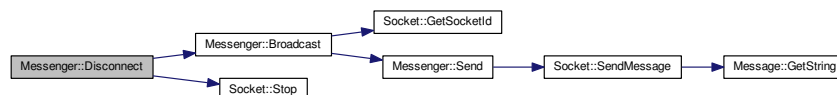


7.27.3.4 void Messenger::Disconnect ()

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:



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

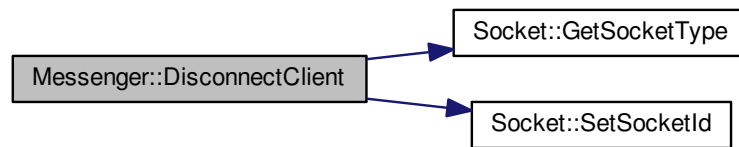
Disconnect a client.

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:



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

Socket actor type retrieval method.

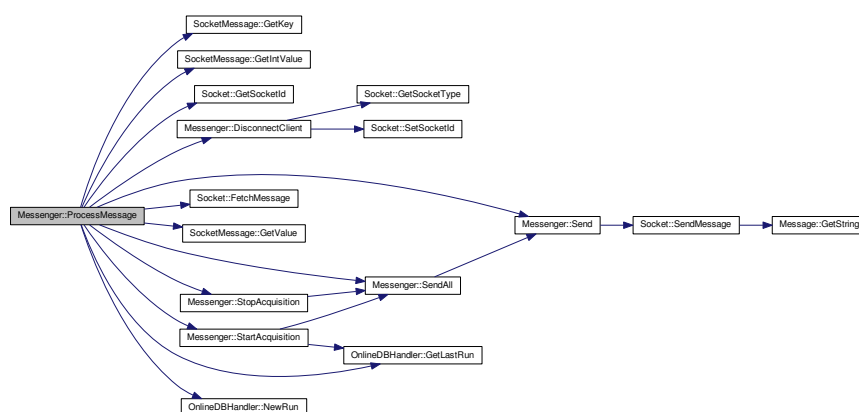
7.27.3.7 void Messenger::ProcessMessage (SocketMessage m, int sid) [private]

Process a message received from the socket.

Parameters

in	<i>Unique</i>	identifier of the client sending the message
----	---------------	--

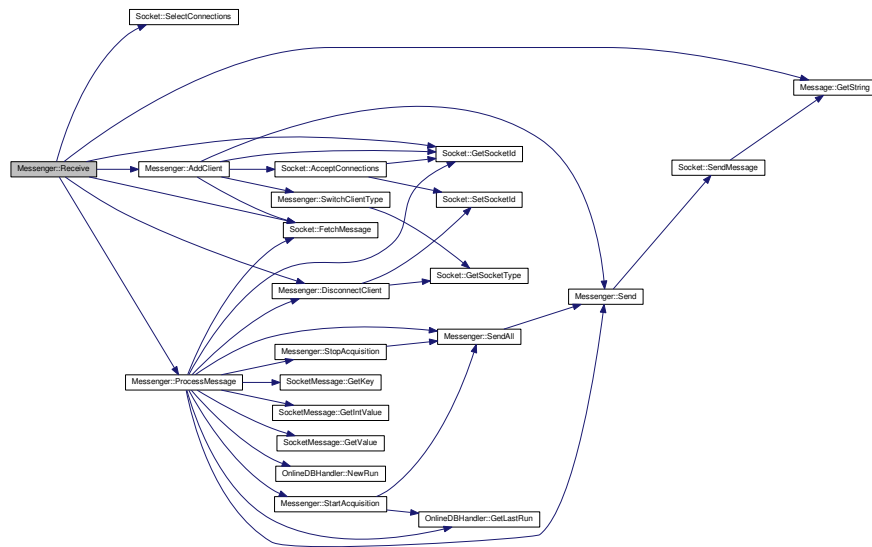
Here is the call graph for this function:



7.27.3.8 void Messenger::Receive ()

Handle a message reception from a client.

Here is the call graph for this function:



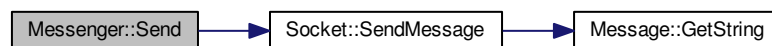
7.27.3.9 void Messenger::Send (const Message & m, int sid) const

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:



7.27.3.10 void Messenger::SendAll (const Socket::SocketType & type, const Message & m) const [inline]

Send any type of message to all clients of one type.

Parameters

in	<i>type</i>	Client type
in	<i>m</i>	Message to transmit

Here is the call graph for this function:



7.27.3.11 `void Messenger::SendAll (const Socket::SocketType & type, const Exception & e) const` `[inline]`

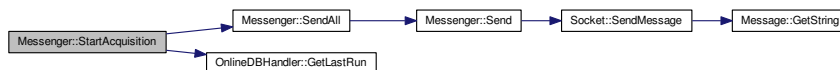
Here is the call graph for this function:



7.27.3.12 `void Messenger::StartAcquisition ()`

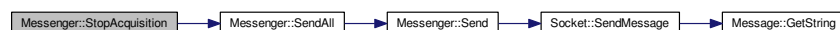
Start the data acquisition.

Here is the call graph for this function:



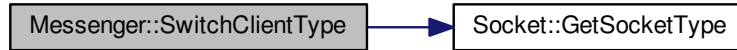
7.27.3.13 `void Messenger::StopAcquisition ()`

Here is the call graph for this function:



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

Here is the call graph for this function:



7.27.4 Field Documentation

7.27.4.1 `int Messenger::fNumAttempts` [private]

7.27.4.2 `pid_t Messenger::fPID` [private]

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

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

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

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

7.28 OnlineDBHandler Class Reference

Handler for the run information online database.

```
#include <OnlineDBHandler.h>
```

Data Structures

- struct [BurstInfo](#)
- struct [TDCConditions](#)

Public Types

- typedef std::map< unsigned int, unsigned int > [RunCollection](#)
- typedef std::vector< [BurstInfo](#) > [BurstInfos](#)
- typedef std::vector< [TDCConditions](#) > [TDCConditionsCollection](#)

Public Member Functions

- [OnlineDBHandler](#) (std::string path=std::string(std::getenv("PPS_PATH"))+"/run_infos.db")
- [~OnlineDBHandler](#) ()
- void [NewRun](#) ()
- void [NewBurst](#) ()
- [RunCollection](#) [GetRuns](#) () const

- unsigned int [GetLastRun](#) () const
Retrieve the last run acquired.
- int [GetLastBurst](#) (unsigned int run) const
- [BurstInfos GetRunInfo](#) (unsigned int run) const
Retrieve information on a given run (spill IDs / timestamp)
- void [SetTDCConditions](#) (unsigned short tdc_id, unsigned long tdc_address, unsigned short tdc_acq_mode, unsigned short tdc_det_mode, std::string detector)
- [TDCConditionsCollection GetTDCConditions](#) (unsigned int run_id) const
- void [SetHVConditions](#) (unsigned short channel_id, unsigned int vmax, unsigned int imax)

Private Member Functions

- void [BuildTables](#) ()
- template<class T >
std::vector< std::vector< T > > [Select](#) (std::string req, int num_fields=-1) const

Private Attributes

- sqlite3 * [fDB](#)

7.28.1 Detailed Description

Handler for the run information online database.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

3 Aug 2015

7.28.2 Member Typedef Documentation

7.28.2.1 typedef std::vector<[BurstInfo](#)> [OnlineDBHandler::BurstInfos](#)

7.28.2.2 typedef std::map<unsigned int, unsigned int> [OnlineDBHandler::RunCollection](#)

7.28.2.3 typedef std::vector<[TDCConditions](#)> [OnlineDBHandler::TDCConditionsCollection](#)

7.28.3 Constructor & Destructor Documentation

7.28.3.1 [OnlineDBHandler::OnlineDBHandler](#) (std::string *path* = std::string(std::getenv("PPS_↔
PATH")) + "/run_infos.db") [inline]

Here is the call graph for this function:



7.28.3.2 `OnlineDBHandler::~~OnlineDBHandler () [inline]`

7.28.4 Member Function Documentation

7.28.4.1 `void OnlineDBHandler::BuildTables () [inline],[private]`

7.28.4.2 `int OnlineDBHandler::GetLastBurst (unsigned int run) const [inline]`

7.28.4.3 `unsigned int OnlineDBHandler::GetLastRun () const [inline]`

Retrieve the last run acquired.

7.28.4.4 **BurstInfos** `OnlineDBHandler::GetRunInfo (unsigned int run) const [inline]`

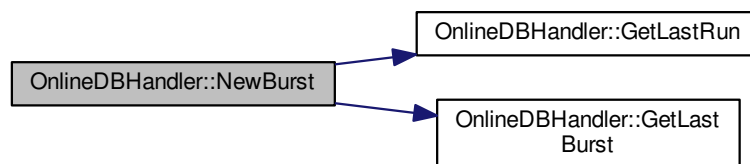
Retrieve information on a given run (spill IDs / timestamp)

7.28.4.5 **RunCollection** `OnlineDBHandler::GetRuns () const [inline]`

7.28.4.6 **TDCConditionsCollection** `OnlineDBHandler::GetTDCConditions (unsigned int run_id) const [inline]`

7.28.4.7 `void OnlineDBHandler::NewBurst () [inline]`

Here is the call graph for this function:



7.28.4.8 `void OnlineDBHandler::NewRun () [inline]`

7.28.4.9 `template<class T> std::vector< std::vector<T>> OnlineDBHandler::Select (std::string req, int num_fields = -1) const [inline],[private]`

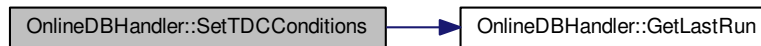
7.28.4.10 `void OnlineDBHandler::SetHVConditions (unsigned short channel_id, unsigned int vmax, unsigned int imax) [inline]`

Here is the call graph for this function:



7.28.4.11 `void OnlineDBHandler::SetTDCConditions (unsigned short tdc_id, unsigned long tdc_address, unsigned short tdc_acq_mode, unsigned short tdc_det_mode, std::string detector) [inline]`

Here is the call graph for this function:



7.28.5 Field Documentation

7.28.5.1 `sqlite3* OnlineDBHandler::fDB [private]`

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

- include/OnlineDBHandler.h

7.29 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](#)

7.29.1 Constructor & Destructor Documentation

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

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

7.29.2 Member Function Documentation

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

7.29.3 Field Documentation

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

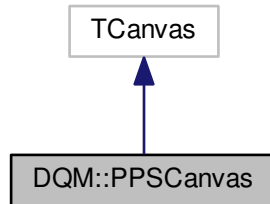
The documentation for this class was generated from the following file:

- include/VME_PCIInterfaceA2818.h

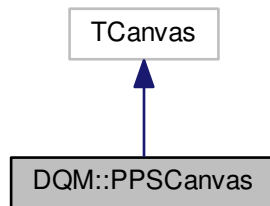
7.30 DQM::PPSCanvas Class Reference

```
#include <PPSCanvas.h>
```

Inheritance diagram for DQM::PPSCanvas:



Collaboration diagram for DQM::PPSCanvas:



Public Member Functions

- [PPSCanvas](#) ()
- [PPSCanvas](#) (TString name, unsigned int width=500, unsigned int height=500, TString upper_label="")
- [PPSCanvas](#) (TString name, TString upper_label)
- virtual [~PPSCanvas](#) ()
- void [SetRunInfo](#) (unsigned int run_id, TString date)
- void [SetUpperLabel](#) (TString text)
- TPad * [Grid](#) ()
- void [Save](#) (TString ext="png", TString path=".")

Private Member Functions

- void [Build](#) ()
- void [DrawGrid](#) ()

Private Attributes

- TPad * [c1](#)
- TPad * [c2](#)
- double [fWidth](#)
- double [fHeight](#)
- TLegend * [fLegend](#)
- double [fLegendX](#)
- double [fLegendY](#)
- unsigned int [fLegendNumEntries](#)
- TPaveText * [fLabel1](#)
- TPaveText * [fLabel2](#)
- TPaveText * [fLabel3](#)
- TString [fUpperLabelText](#)
- TPaveText * [fUpperLabel](#)
- bool [fLabelsDrawn](#)
- unsigned int [fRunId](#)
- TString [fRunDate](#)

7.30.1 Detailed Description

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

3 Aug 2015

7.30.2 Constructor & Destructor Documentation

7.30.2.1 DQM::PPSCanvas::PPSCanvas () [inline]

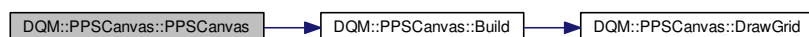
7.30.2.2 DQM::PPSCanvas::PPSCanvas (TString *name*, unsigned int *width* = 500, unsigned int *height* = 500, TString *upper_label* = " ") [inline]

Here is the call graph for this function:



7.30.2.3 DQM::PPSCanvas::PPSCanvas (TString *name*, TString *upper_label*) [inline]

Here is the call graph for this function:



7.30.2.4 virtual DQM::PPSCanvas::~~PPSCanvas () [inline],[virtual]

7.30.3 Member Function Documentation

7.30.3.1 void DQM::PPSCanvas::Build () [inline],[private]

Here is the call graph for this function:



7.30.3.2 void DQM::PPSCanvas::DrawGrid () [inline],[private]

7.30.3.3 TPad* DQM::PPSCanvas::Grid () [inline]

7.30.3.4 void DQM::PPSCanvas::Save (TString ext = "png", TString path = " . ") [inline]

Here is the call graph for this function:



7.30.3.5 void DQM::PPSCanvas::SetRunInfo (unsigned int run_id, TString date) [inline]

7.30.3.6 void DQM::PPSCanvas::SetUpperLabel (TString text) [inline]

7.30.4 Field Documentation

7.30.4.1 TPad* DQM::PPSCanvas::c1 [private]

7.30.4.2 TPad * DQM::PPSCanvas::c2 [private]

7.30.4.3 double DQM::PPSCanvas::fHeight [private]

7.30.4.4 TPaveText* DQM::PPSCanvas::fLabel1 [private]

7.30.4.5 TPaveText * DQM::PPSCanvas::fLabel2 [private]

7.30.4.6 TPaveText * DQM::PPSCanvas::fLabel3 [private]

- 7.30.4.7 `bool DQM::PPSCanvas::fLabelsDrawn` `[private]`
- 7.30.4.8 `TLegend* DQM::PPSCanvas::fLegend` `[private]`
- 7.30.4.9 `unsigned int DQM::PPSCanvas::fLegendNumEntries` `[private]`
- 7.30.4.10 `double DQM::PPSCanvas::fLegendX` `[private]`
- 7.30.4.11 `double DQM::PPSCanvas::fLegendY` `[private]`
- 7.30.4.12 `TString DQM::PPSCanvas::fRunDate` `[private]`
- 7.30.4.13 `unsigned int DQM::PPSCanvas::fRunId` `[private]`
- 7.30.4.14 `TPaveText* DQM::PPSCanvas::fUpperLabel` `[private]`
- 7.30.4.15 `TString DQM::PPSCanvas::fUpperLabelText` `[private]`
- 7.30.4.16 `double DQM::PPSCanvas::fWidth` `[private]`

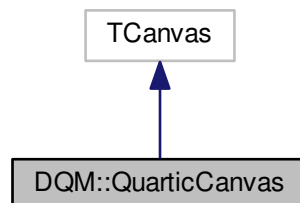
The documentation for this class was generated from the following file:

- `include/PPSCanvas.h`

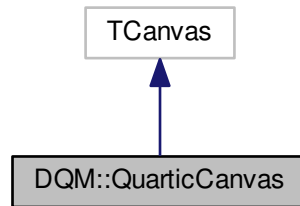
7.31 DQM::QuarticCanvas Class Reference

```
#include <QuarticCanvas.h>
```

Inheritance diagram for DQM::QuarticCanvas:



Collaboration diagram for DQM::QuarticCanvas:



Data Structures

- struct [Coord](#)

Public Member Functions

- [QuarticCanvas](#) ()
- [QuarticCanvas](#) (TString name, unsigned int width=500, unsigned int height=500, TString upper_label="")
- [QuarticCanvas](#) (TString name, TString upper_label)
- virtual [~QuarticCanvas](#) ()
- void [SetRunInfo](#) (unsigned int board_id, unsigned int run_id, unsigned int spill_id, TString date)
- void [SetUpperLabel](#) (TString text)
- void [FillChannel](#) (unsigned short channel_id, double content)
- TH2D * [Grid](#) ()
- void [Save](#) (TString ext="png", TString path=".")

Private Member Functions

- void [Build](#) ()
- void [DrawGrid](#) ()
- [Coord GetCoordinates](#) (unsigned short channel_id) const

Private Attributes

- TPad * [c1](#)
- TPad * [c2](#)
- TH2D * [fHist](#)
- double [fWidth](#)
- double [fHeight](#)
- TLegend * [fLegend](#)
- double [fLegendX](#)
- double [fLegendY](#)
- unsigned int [fLegendNumEntries](#)
- TPaveText * [fLabel1](#)
- TPaveText * [fLabel2](#)
- TPaveText * [fLabel3](#)

- TPaveText * [fLabel4](#)
- TString [fUpperLabelText](#)
- TPaveText * [fUpperLabel](#)
- bool [fLabelsDrawn](#)
- unsigned int [fBoardId](#)
- unsigned int [fRunId](#)
- unsigned int [fSpillId](#)
- TString [fRunDate](#)

7.31.1 Detailed Description

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

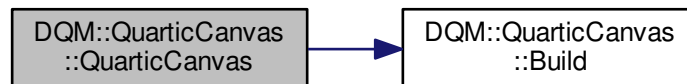
3 Aug 2015

7.31.2 Constructor & Destructor Documentation

7.31.2.1 `DQM::QuarticCanvas::QuarticCanvas ()` `[inline]`

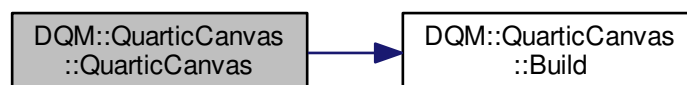
7.31.2.2 `DQM::QuarticCanvas::QuarticCanvas (TString name, unsigned int width = 500, unsigned int height = 500, TString upper_label = " ")` `[inline]`

Here is the call graph for this function:



7.31.2.3 `DQM::QuarticCanvas::QuarticCanvas (TString name, TString upper_label)` `[inline]`

Here is the call graph for this function:



7.31.2.4 virtual DQM::QuarticCanvas::~~QuarticCanvas () [inline],[virtual]

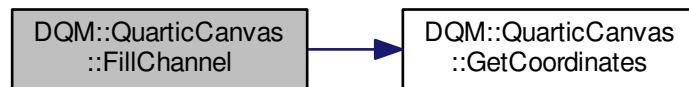
7.31.3 Member Function Documentation

7.31.3.1 void DQM::QuarticCanvas::Build () [inline],[private]

7.31.3.2 void DQM::QuarticCanvas::DrawGrid () [inline],[private]

7.31.3.3 void DQM::QuarticCanvas::FillChannel (unsigned short *channel_id*, double *content*) [inline]

Here is the call graph for this function:

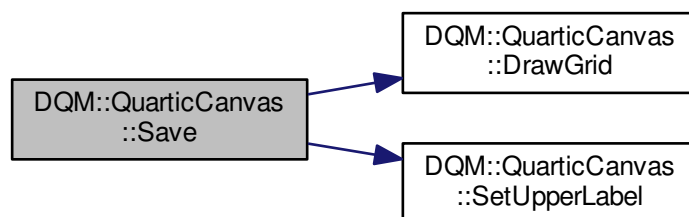


7.31.3.4 Coord DQM::QuarticCanvas::GetCoordinates (unsigned short *channel_id*) const [inline],[private]

7.31.3.5 TH2D* DQM::QuarticCanvas::Grid () [inline]

7.31.3.6 void DQM::QuarticCanvas::Save (TString *ext* = "png", TString *path* = " . ") [inline]

Here is the call graph for this function:



7.31.3.7 void DQM::QuarticCanvas::SetRunInfo (unsigned int *board_id*, unsigned int *run_id*, unsigned int *spill_id*, TString *date*) [inline]

7.31.3.8 void DQM::QuarticCanvas::SetUpperLabel (TString *text*) [inline]

7.31.4 Field Documentation

- 7.31.4.1 TPad* DQM::QuarticCanvas::c1 [private]
- 7.31.4.2 TPad * DQM::QuarticCanvas::c2 [private]
- 7.31.4.3 unsigned int DQM::QuarticCanvas::fBoardId [private]
- 7.31.4.4 double DQM::QuarticCanvas::fHeight [private]
- 7.31.4.5 TH2D* DQM::QuarticCanvas::fHist [private]
- 7.31.4.6 TPaveText* DQM::QuarticCanvas::fLabel1 [private]
- 7.31.4.7 TPaveText * DQM::QuarticCanvas::fLabel2 [private]
- 7.31.4.8 TPaveText * DQM::QuarticCanvas::fLabel3 [private]
- 7.31.4.9 TPaveText * DQM::QuarticCanvas::fLabel4 [private]
- 7.31.4.10 bool DQM::QuarticCanvas::fLabelsDrawn [private]
- 7.31.4.11 TLegend* DQM::QuarticCanvas::fLegend [private]
- 7.31.4.12 unsigned int DQM::QuarticCanvas::fLegendNumEntries [private]
- 7.31.4.13 double DQM::QuarticCanvas::fLegendX [private]
- 7.31.4.14 double DQM::QuarticCanvas::fLegendY [private]
- 7.31.4.15 TString DQM::QuarticCanvas::fRunDate [private]
- 7.31.4.16 unsigned int DQM::QuarticCanvas::fRunId [private]
- 7.31.4.17 unsigned int DQM::QuarticCanvas::fSpillId [private]
- 7.31.4.18 TPaveText* DQM::QuarticCanvas::fUpperLabel [private]
- 7.31.4.19 TString DQM::QuarticCanvas::fUpperLabelText [private]
- 7.31.4.20 double DQM::QuarticCanvas::fWidth [private]

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

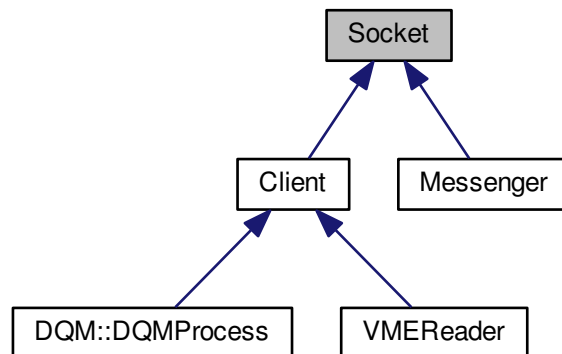
- include/QuarticCanvas.h

7.32 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](#), [DAQ](#) }
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](#)

7.32.1 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

7.32.2 Member Typedef Documentation

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

7.32.3 Constructor & Destructor Documentation

7.32.3.1 `Socket::Socket () \[inline\]`

7.32.3.2 `Socket::Socket (int port)`

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

7.32.4 Member Function Documentation

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

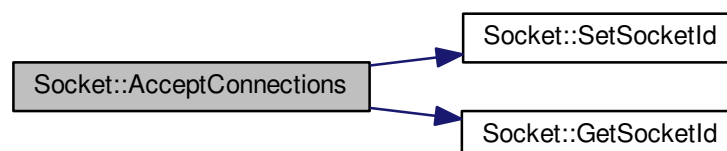
Accept connection from a client.

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

Parameters

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

Here is the call graph for this function:



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

Bind a name to a socket.

Returns

Success of the operation

Here is the call graph for this function:



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

Configure the socket object for communication.

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

Create an endpoint for communication.

7.32.4.5 `void Socket::DumpConnected () const`

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

Receive a message from a socket.

Returns

Received message as a `std::string`

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

Retrieve the port used for this socket.

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

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

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

Here is the call graph for this function:



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

Listen to incoming messages.

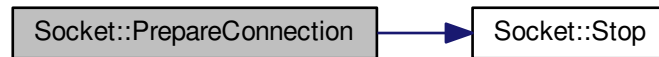
Set the socket to listen to any message coming from outside

Here is the call graph for this function:



7.32.4.12 void Socket::PrepareConnection () [protected]

Here is the call graph for this function:



7.32.4.13 void Socket::SelectConnections ()

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

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

Send a message on a socket.

Here is the call graph for this function:



7.32.4.15 void Socket::SetPort (int port) [inline]

7.32.4.16 void Socket::SetSocketId (int sid) [inline]

7.32.4.17 bool Socket::Start () [protected]

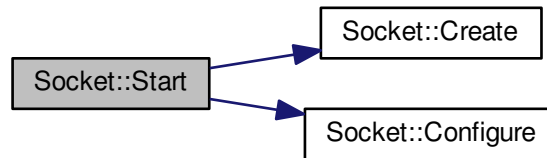
Start the socket.

Launch all mandatory operations to set the socket to be used

Returns

Success of the operation

Here is the call graph for this function:

**7.32.4.18 void Socket::Stop ()**

Terminates the socket and all attached communications.

7.32.5 Field Documentation

7.32.5.1 struct sockaddr_in Socket::fAddress [private]

7.32.5.2 char Socket::fBuffer[MAX_WORD_LENGTH] [protected]

7.32.5.3 fd_set Socket::fMaster [protected]

Master file descriptor list.

7.32.5.4 int Socket::fPort [protected]

7.32.5.5 fd_set Socket::fReadFds [protected]

Temp file descriptor list for select()

7.32.5.6 int Socket::fSocketId [private]

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

7.32.5.7 SocketCollection Socket::fSocketsConnected [protected]

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

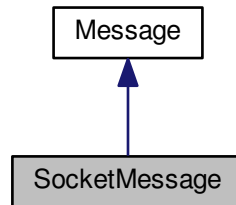
- include/Socket.h
- src/Socket.cpp

7.33 SocketMessage Class Reference

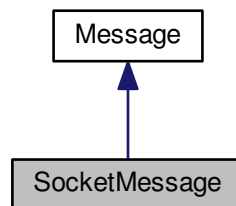
Socket-passed message type.

```
#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 short value)

Construct a socket message out of a key and a short integer-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 long value)
Construct a socket message out of a key and a long 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, short int_value)
Send a short integer-valued message.
- void [SetKeyValue](#) (const MessageKey &key, int int_value)
Send an integer-valued message.
- void [SetKeyValue](#) (const MessageKey &key, long int_value)
Send a long 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.
- std::string [GetCleanedValue](#) () const
Extract the message's string value (without the trailing endlines)
- 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

7.33.1 Detailed Description

Socket-passed message type.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

26 Mar 2015

7.33.2 Constructor & Destructor Documentation

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

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

Here is the call graph for this function:



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

Here is the call graph for this function:



7.33.2.4 `SocketMessage::SocketMessage (std::string msg_s) [inline]`

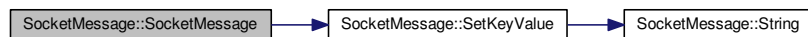
Here is the call graph for this function:



7.33.2.5 `SocketMessage::SocketMessage (const MessageKey & key) [inline]`

Construct a socket message out of a key.

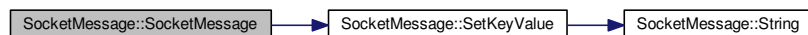
Here is the call graph for this function:



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

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

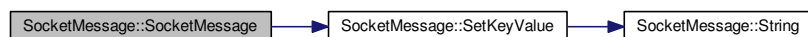
Here is the call graph for this function:



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

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

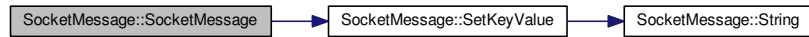
Here is the call graph for this function:



7.33.2.8 `SocketMessage::SocketMessage (const MessageKey & key, const short value)` `[inline]`

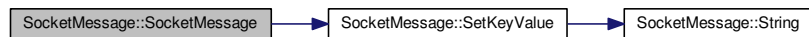
Construct a socket message out of a key and a short integer-type value.

Here is the call graph for this function:

**7.33.2.9** `SocketMessage::SocketMessage (const MessageKey & key, const int value)` `[inline]`

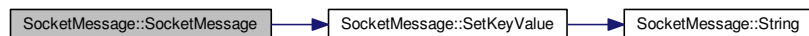
Construct a socket message out of a key and an integer-type value.

Here is the call graph for this function:

**7.33.2.10** `SocketMessage::SocketMessage (const MessageKey & key, const long value)` `[inline]`

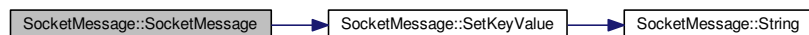
Construct a socket message out of a key and a long integer-type value.

Here is the call graph for this function:

**7.33.2.11** `SocketMessage::SocketMessage (const MessageKey & key, const float value)` `[inline]`

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

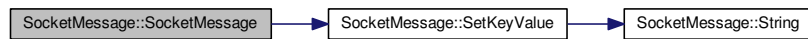
Here is the call graph for this function:



7.33.2.12 `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:



7.33.2.13 `SocketMessage::SocketMessage (MessageMap msg_m) [inline]`

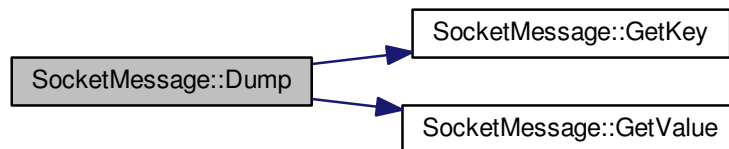
Construct a socket message out of a map of key/string-type value.

7.33.2.14 `SocketMessage::~~SocketMessage () [inline]`

7.33.3 Member Function Documentation

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

Here is the call graph for this function:



7.33.3.2 `std::string SocketMessage::GetCleanedValue () const [inline]`

Extract the message's string value (without the trailing endlines)

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

Extract the message's integer value.

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

Extract the message's key.

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

Extract the whole key:value message.

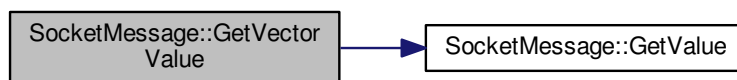
7.33.3.6 `std::string SocketMessage::GetValue () const` `[inline]`

Extract the message's string value.

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

Extract the message's vector of string value.

Here is the call graph for this function:



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

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

String-valued message.

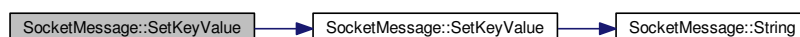
Here is the call graph for this function:



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

Send a short integer-valued message.

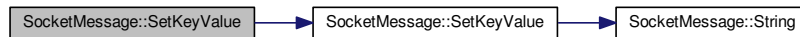
Here is the call graph for this function:



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

Send an integer-valued message.

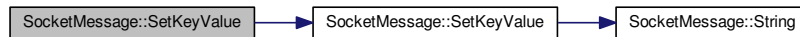
Here is the call graph for this function:



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

Send a long integer-valued message.

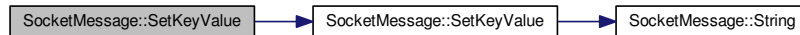
Here is the call graph for this function:



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

Float-valued message.

Here is the call graph for this function:



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

Double-valued message.

Here is the call graph for this function:



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

7.33.4 Field Documentation

7.33.4.1 MessageMap SocketMessage::fMessage [private]

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

- include/SocketMessage.h

7.34 OnlineDBHandler::TDCConditions Struct Reference

```
#include <OnlineDBHandler.h>
```

Public Member Functions

- bool [operator==](#) (const [TDCConditions](#) &rhs) const
- [TDCConditions](#) & [operator=](#) (const [TDCConditions](#) &rhs)

Data Fields

- unsigned int [run_id](#)
- unsigned short [tdc_id](#)
- unsigned long [tdc_address](#)
- unsigned short [tdc_acq_mode](#)
- unsigned short [tdc_det_mode](#)
- std::string [detector](#)

7.34.1 Member Function Documentation

7.34.1.1 [TDCConditions& OnlineDBHandler::TDCConditions::operator= \(const \[TDCConditions\]\(#\) & rhs \)](#) [inline]

7.34.1.2 [bool OnlineDBHandler::TDCConditions::operator== \(const \[TDCConditions\]\(#\) & rhs \) const](#) [inline]

7.34.2 Field Documentation

7.34.2.1 [std::string OnlineDBHandler::TDCConditions::detector](#)

7.34.2.2 [unsigned int OnlineDBHandler::TDCConditions::run_id](#)

7.34.2.3 [unsigned short OnlineDBHandler::TDCConditions::tdc_acq_mode](#)

7.34.2.4 [unsigned long OnlineDBHandler::TDCConditions::tdc_address](#)

7.34.2.5 [unsigned short OnlineDBHandler::TDCConditions::tdc_det_mode](#)

7.34.2.6 [unsigned short OnlineDBHandler::TDCConditions::tdc_id](#)

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

- include/OnlineDBHandler.h

7.35 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)

7.35.1 Detailed Description

Error flags handler.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

22 Jun 2015

7.35.2 Constructor & Destructor Documentation

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

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

7.35.3 Member Function Documentation

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

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

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

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

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

Internal fatal chip error has been detected.

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

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

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

Hits rejected because of programmed event size limit.

7.35.3.7 `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.

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

Event lost (trigger FIFO overflow)

7.35.4 Friends And Related Function Documentation

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

7.35.5 Field Documentation

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

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

- include/VME_TDCEvent.h

7.36 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,
[Trigger](#) = 0x1f }

Public Member Functions

- [TDCEvent](#) ()
- [TDCEvent](#) (const [TDCEvent](#) &ev)
- [TDCEvent](#) (const uint32_t &word)
- [TDCEvent](#) (const [EventType](#) &ev)

- 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`

7.36.1 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

7.36.2 Member Enumeration Documentation

7.36.2.1 enum VME::TDCEvent::EventType

Enumerator

TDMeasurement

TDCHeader

TDCTrailer

TDCError

GlobalHeader

GlobalTrailer

ETTT

Filler

Trigger

7.36.3 Constructor & Destructor Documentation

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

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

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

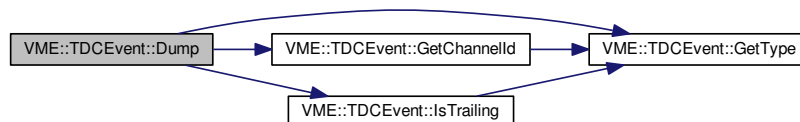
7.36.3.4 VME::TDCEvent::TDCEvent (const EventType & ev) [inline]

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

7.36.4 Member Function Documentation

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

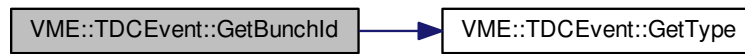
Here is the call graph for this function:



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

Bunch identifier of trigger (or trigger time tag)

Here is the call graph for this function:



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

Channel number for.

Here is the call graph for this function:



7.36.4.4 `TDCErrorFlag VME::TDCEvent::GetErrorFlags () const [inline]`

Return error flags if an error condition has been detected.

Here is the call graph for this function:



7.36.4.5 `uint32_t VME::TDCEvent::GetETTT () const [inline]`

Extended trigger time tag.

Here is the call graph for this function:



7.36.4.6 uint32_t VME::TDCEvent::GetEventCount () const [inline]

Total number of events.

Here is the call graph for this function:



7.36.4.7 uint16_t VME::TDCEvent::GetEventId () const [inline]

Event identifier from event counter.

Here is the call graph for this function:



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

Here is the call graph for this function:



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

Here is the call graph for this function:



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

Programmed identifier of master TDC providing the event.

Here is the call graph for this function:



7.36.4.11 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:



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

Type of packet read out from the TDC.

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

Width of pulse in programmed time resolution.

Here is the call graph for this function:



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

7.36.4.15 `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:



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

Are we dealing with a trailing or a leading measurement?

Here is the call graph for this function:



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

7.36.5 Field Documentation

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

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

- `include/VME_TDCEvent.h`

7.37 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](#)

7.37.1 Detailed Description

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

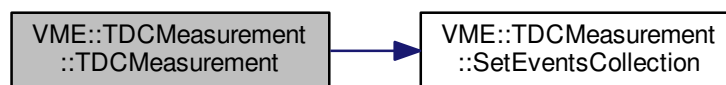
Jun 2015

7.37.2 Constructor & Destructor Documentation

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

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

Here is the call graph for this function:

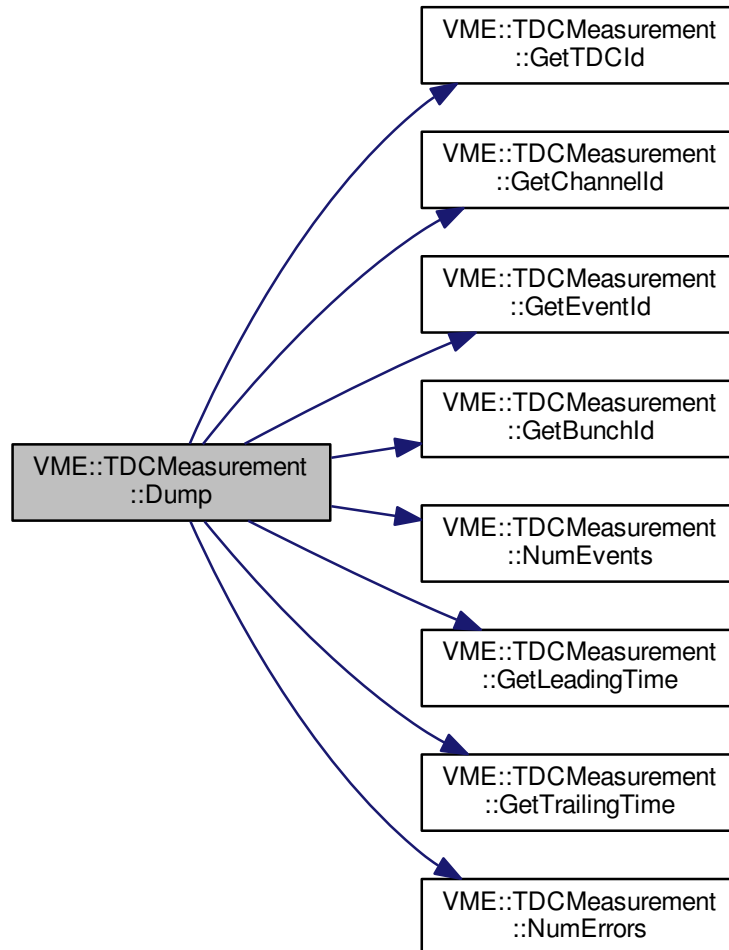


7.37.2.3 VME::TDCMeasurement::~~TDCMeasurement () [inline]

7.37.3 Member Function Documentation

7.37.3.1 `void VME::TDCMeasurement::Dump () [inline]`

Here is the call graph for this function:



7.37.3.2 `uint16_t VME::TDCMeasurement::GetBunchId () [inline]`

7.37.3.3 `uint16_t VME::TDCMeasurement::GetChannelId (unsigned short event_id = 0) [inline]`

7.37.3.4 `uint32_t VME::TDCMeasurement::GetETTT () [inline]`

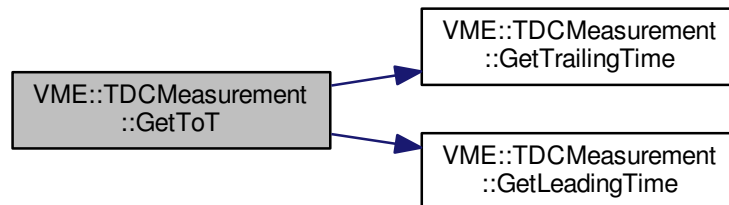
7.37.3.5 `uint16_t VME::TDCMeasurement::GetEventId () [inline]`

7.37.3.6 `uint32_t VME::TDCMeasurement::GetLeadingTime (unsigned short event_id = 0) [inline]`

7.37.3.7 `uint16_t VME::TDCMeasurement::GetTDCId () [inline]`

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

Here is the call graph for this function:



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

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

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

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

7.37.4 Field Documentation

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

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

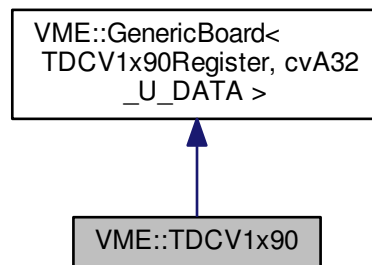
The documentation for this class was generated from the following file:

- `include/VME_TDCMeasurement.h`

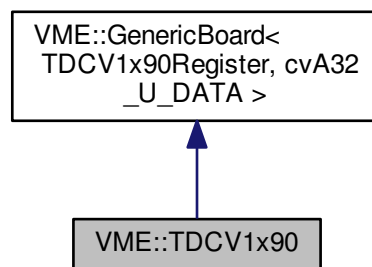
7.38 VME::TDCV1x90 Class Reference

```
#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 [SetLSBTraileadEdge](#) (trailead_edge_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
- void [SetTriggerTimeSubtraction](#) (bool enabled=true) const
- [TDCV1x90TriggerConfig](#) [GetTriggerConfiguration](#) () 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

7.38.1 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)

7.38.2 Member Enumeration Documentation

7.38.2.1 enum `VME::TDCV1x90::DLLMode`

Enumerator

`DLL_Direct_LowRes`

`DLL_PLL_LowRes`

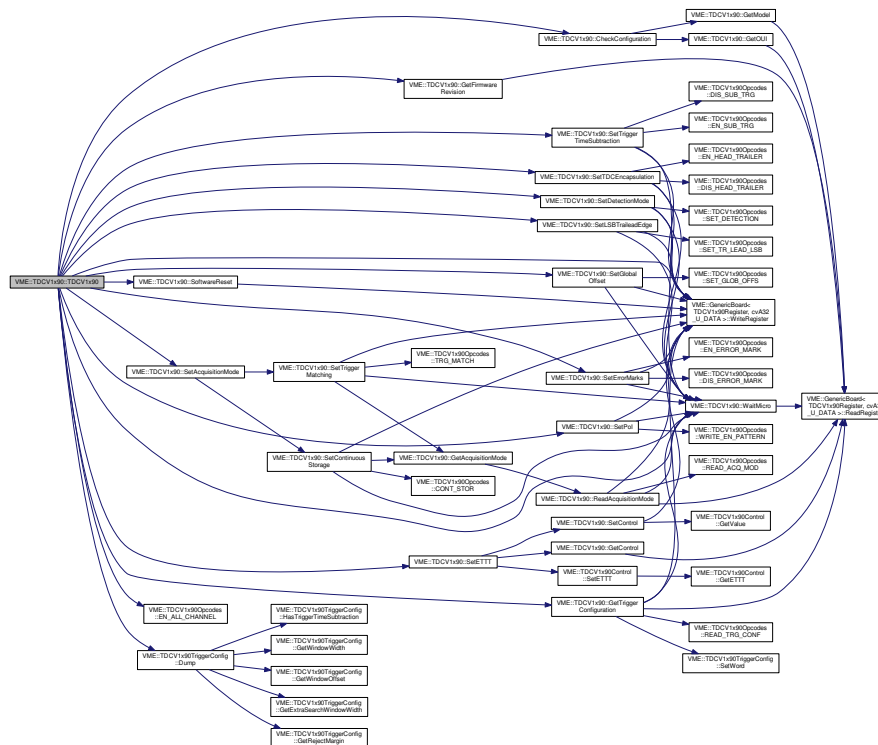
`DLL_PLL_MedRes`

`DLL_PLL_HighRes`

7.38.3 Constructor & Destructor Documentation

7.38.3.1 VME::TDCV1x90::TDCV1x90 (int32_t *bhandle*, uint32_t *baseaddr*)

Here is the call graph for this function:



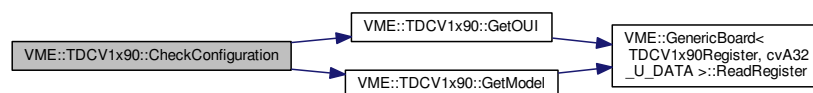
7.38.3.2 VME::TDCV1x90::~~TDCV1x90 ()

7.38.4 Member Function Documentation

7.38.4.1 void VME::TDCV1x90::abort ()

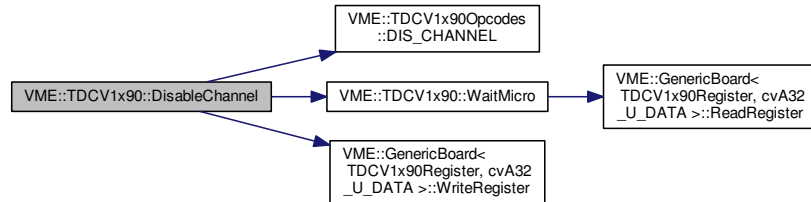
7.38.4.2 void VME::TDCV1x90::CheckConfiguration () const

Here is the call graph for this function:



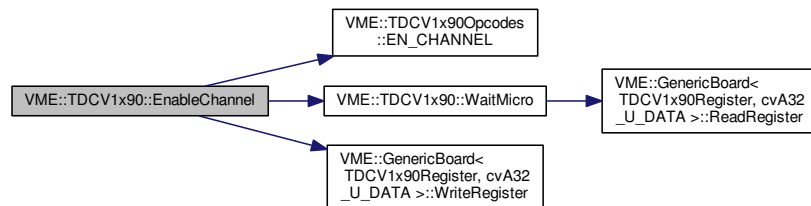
7.38.4.3 void VME::TDCV1x90::DisableChannel (short *channel_id*) const

Here is the call graph for this function:



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

Here is the call graph for this function:



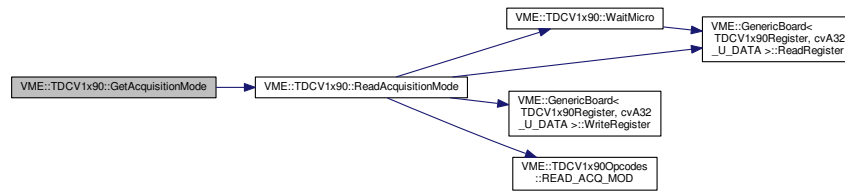
7.38.4.5 TDCEventCollection VME::TDCV1x90::FetchEvents ()

Here is the call graph for this function:



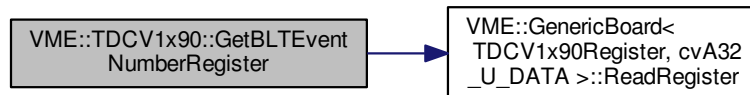
7.38.4.6 AcquisitionMode VME::TDCV1x90::GetAcquisitionMode () [inline]

Here is the call graph for this function:



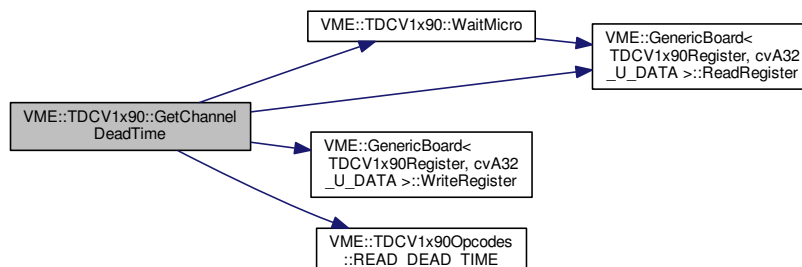
7.38.4.7 uint16_t VME::TDCV1x90::GetBLTEventNumberRegister () const

Here is the call graph for this function:



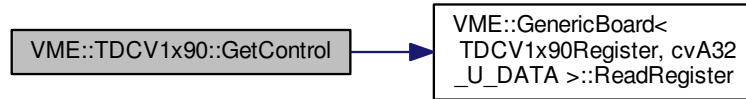
7.38.4.8 unsigned short VME::TDCV1x90::GetChannelDeadTime () const

Here is the call graph for this function:



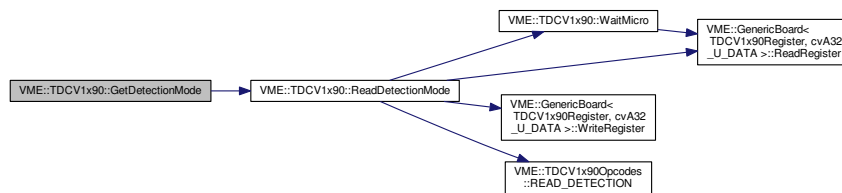
7.38.4.9 TDCV1x90Control VME::TDCV1x90::GetControl () const

Here is the call graph for this function:



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

Here is the call graph for this function:

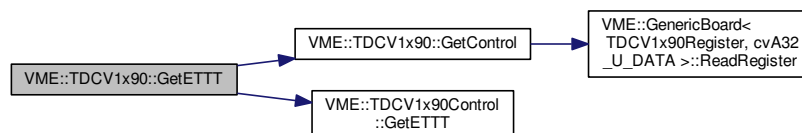


7.38.4.11 DLLMode VME::TDCV1x90::GetDLLClock () const

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

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

Here is the call graph for this function:

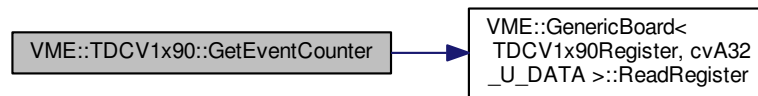


7.38.4.14 uint32_t VME::TDCV1x90::GetEventCounter () const

Number of occurred triggers.

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

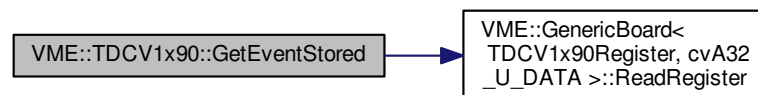
Here is the call graph for this function:



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

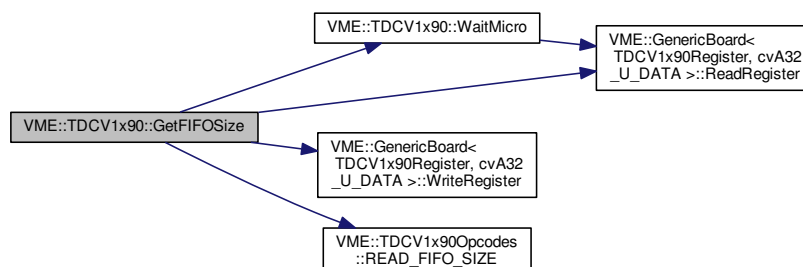
Number of events currently stored in the output buffer.

Here is the call graph for this function:



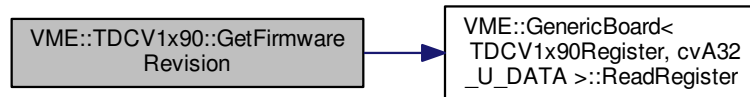
7.38.4.16 `uint16_t VME::TDCV1x90::GetFIFOSize () const`

Here is the call graph for this function:



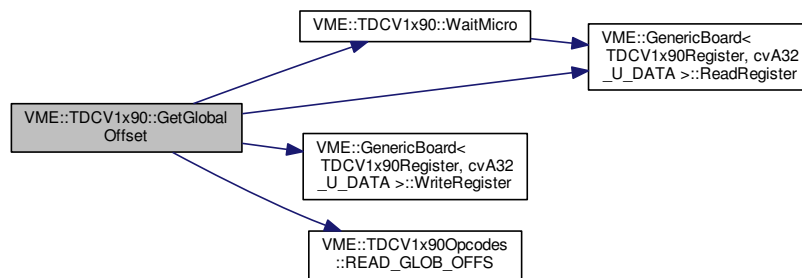
7.38.4.17 uint16_t VME::TDCV1x90::GetFirmwareRevision () const

Here is the call graph for this function:



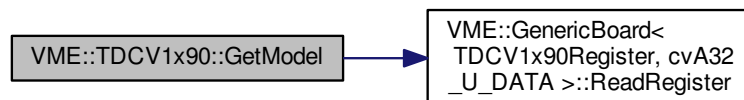
7.38.4.18 GlobalOffset VME::TDCV1x90::GetGlobalOffset () const

Here is the call graph for this function:



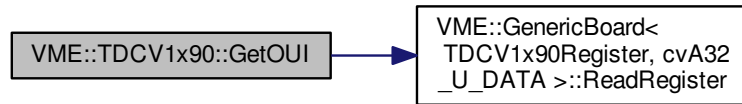
7.38.4.19 uint32_t VME::TDCV1x90::GetModel () const

Here is the call graph for this function:



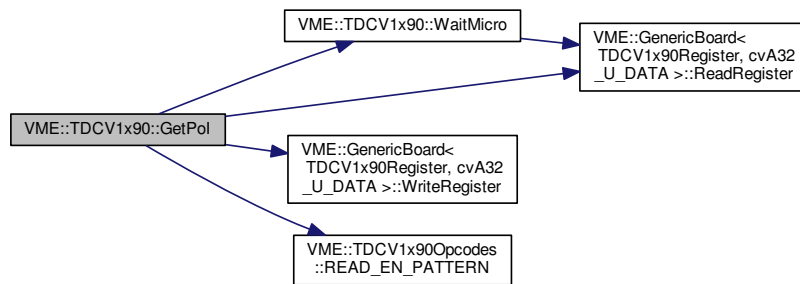
7.38.4.20 uint32_t VME::TDCV1x90::GetOUI () const

Here is the call graph for this function:



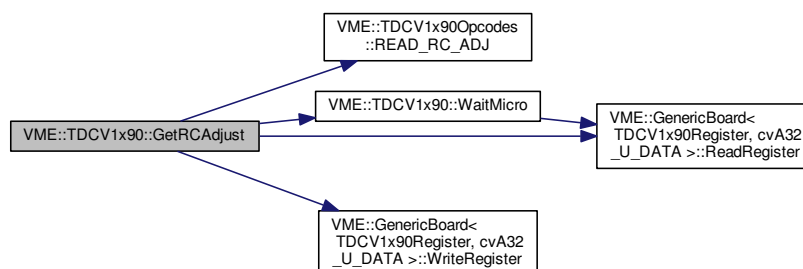
7.38.4.21 std::map< unsigned short, bool > VME::TDCV1x90::GetPol () const

Here is the call graph for this function:



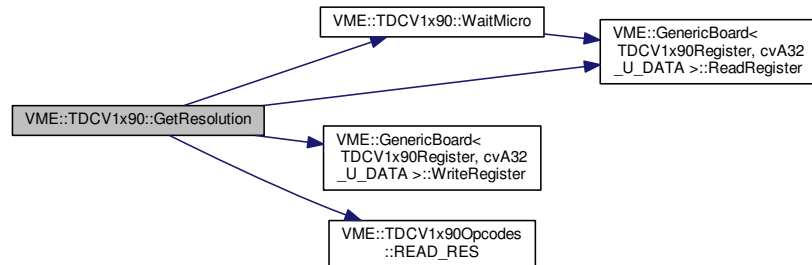
7.38.4.22 uint16_t VME::TDCV1x90::GetRCAdjust (int tdc) const

Here is the call graph for this function:



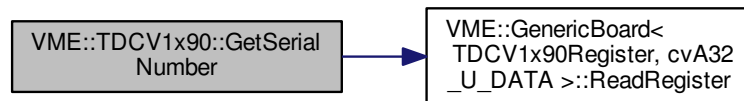
7.38.4.23 uint16_t VME::TDCV1x90::GetResolution () const

Here is the call graph for this function:



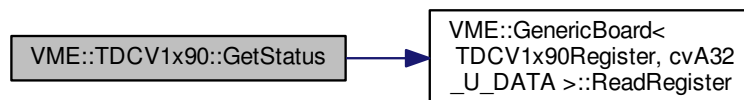
7.38.4.24 uint32_t VME::TDCV1x90::GetSerialNumber () const

Here is the call graph for this function:



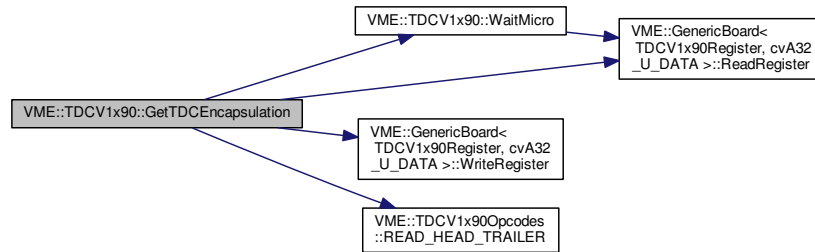
7.38.4.25 TDCV1x90Status VME::TDCV1x90::GetStatus () const

Here is the call graph for this function:



7.38.4.26 bool VME::TDCV1x90::GetTDCEncapsulation () const

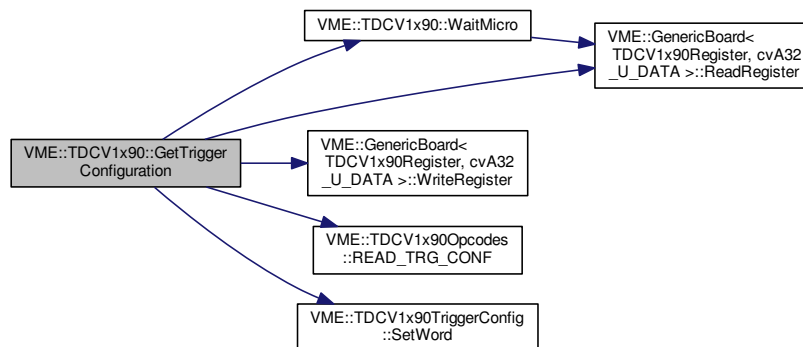
Here is the call graph for this function:



7.38.4.27 bool VME::TDCV1x90::GetTestMode () const

7.38.4.28 TDCV1x90TriggerConfig VME::TDCV1x90::GetTriggerConfiguration () const

Here is the call graph for this function:



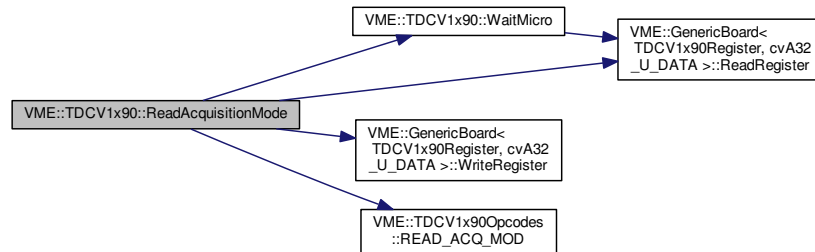
7.38.4.29 int16_t VME::TDCV1x90::GetWindowOffset () const

7.38.4.30 uint16_t VME::TDCV1x90::GetWindowWidth () const [inline]

7.38.4.31 bool VME::TDCV1x90::HardwareReset () const

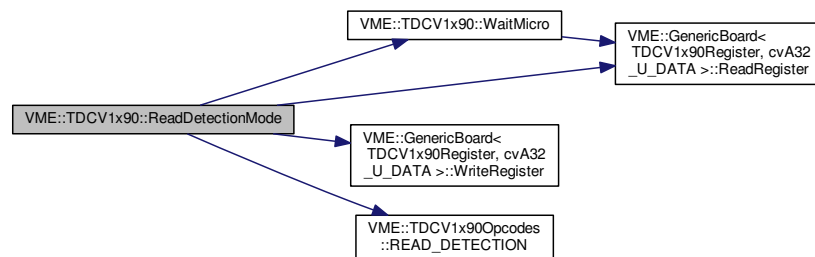
7.38.4.32 void VME::TDCV1x90::ReadAcquisitionMode () [private]

Here is the call graph for this function:



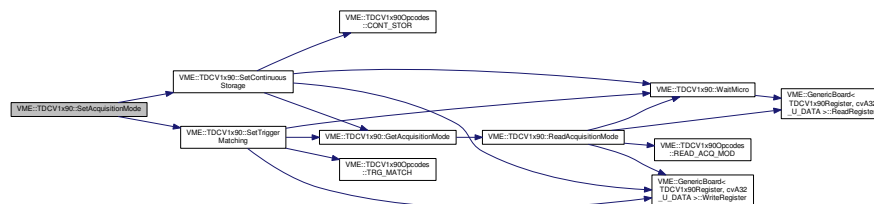
7.38.4.33 void VME::TDCV1x90::ReadDetectionMode () [private]

Here is the call graph for this function:



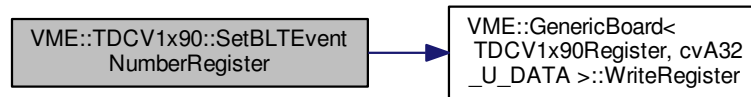
7.38.4.34 void VME::TDCV1x90::SetAcquisitionMode (const AcquisitionMode & mode)

Here is the call graph for this function:



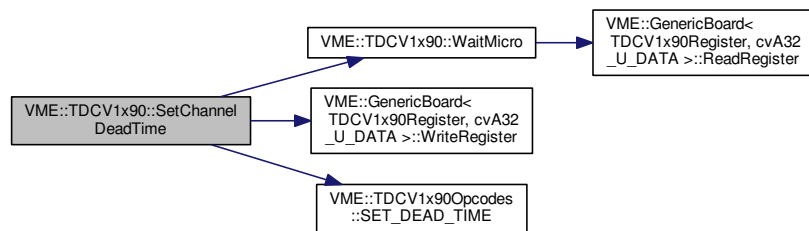
7.38.4.35 void VME::TDCV1x90::SetBLTEventNumberRegister (const uint16_t & value) const

Here is the call graph for this function:



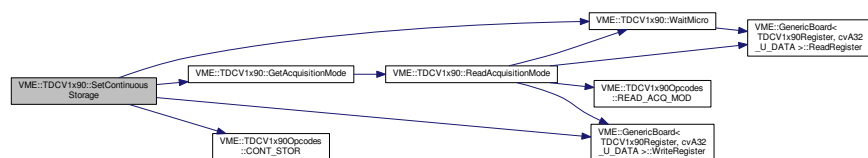
7.38.4.36 void VME::TDCV1x90::SetChannelDeadTime (unsigned short dt) const

Here is the call graph for this function:



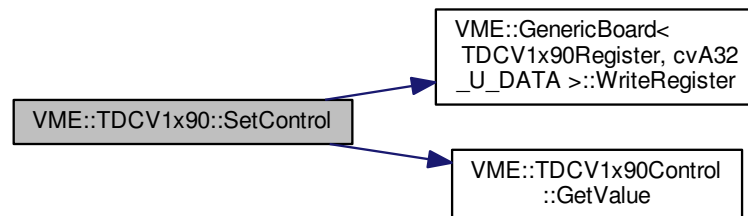
7.38.4.37 void VME::TDCV1x90::SetContinuousStorage ()

Here is the call graph for this function:



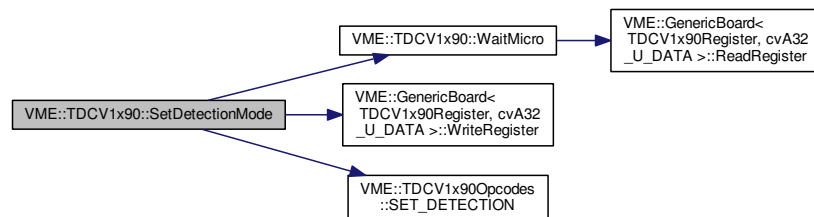
7.38.4.38 void VME::TDCV1x90::SetControl (const TDCV1x90Control & control) const

Here is the call graph for this function:



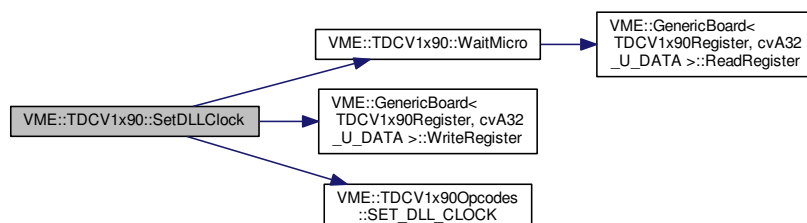
7.38.4.39 void VME::TDCV1x90::SetDetectionMode (const DetectionMode & detm)

Here is the call graph for this function:



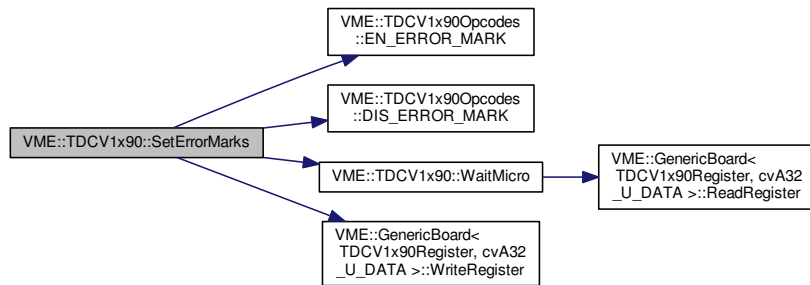
7.38.4.40 void VME::TDCV1x90::SetDLLClock (const DLLMode & dll) const

Here is the call graph for this function:

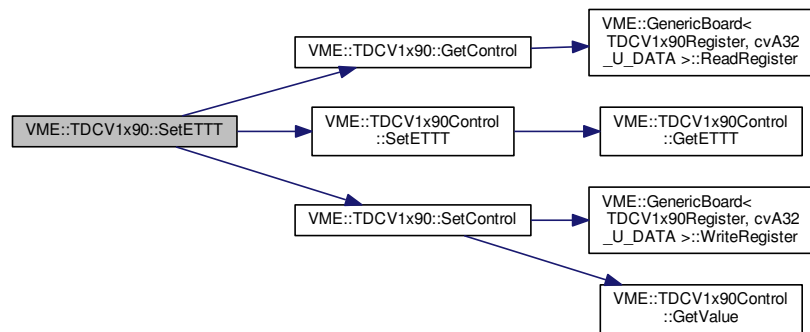


7.38.4.41 void VME::TDCV1x90::SetErrorMarks (bool *mode* = true)

Here is the call graph for this function:

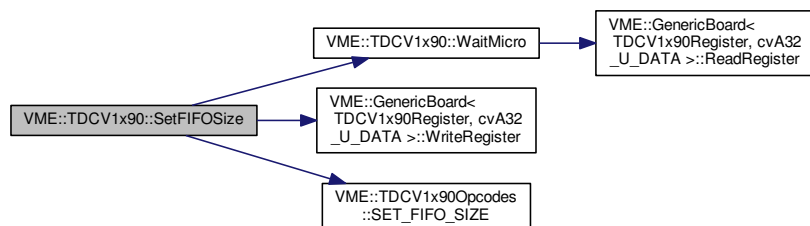
7.38.4.42 void VME::TDCV1x90::SetETTT (bool *ettt* = true) const [inline]

Here is the call graph for this function:



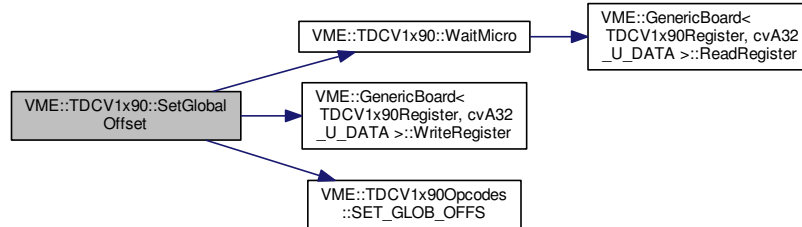
7.38.4.43 void VME::TDCV1x90::SetFIFOSize (const uint16_t & size) const

Here is the call graph for this function:



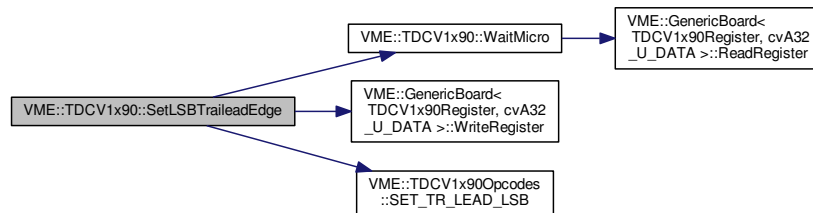
7.38.4.44 void VME::TDCV1x90::SetGlobalOffset (const GlobalOffset & offs) const

Here is the call graph for this function:



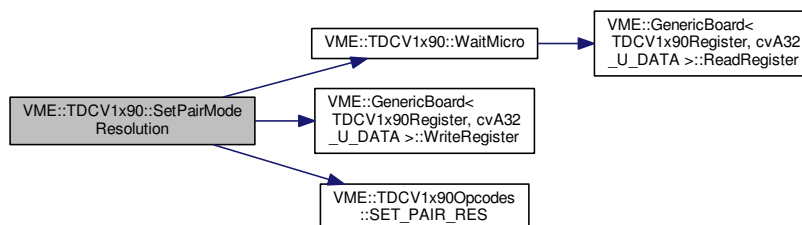
7.38.4.45 void VME::TDCV1x90::SetLSBTraileadEdge (trailead_edge_lsb conf) const

Here is the call graph for this function:



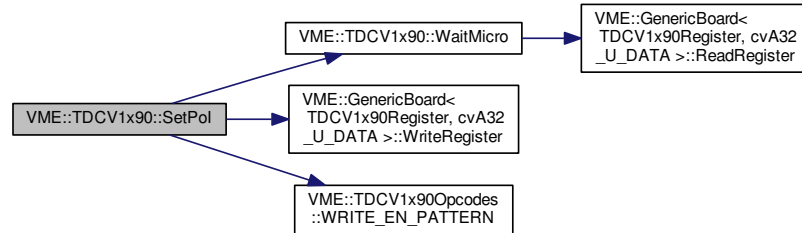
7.38.4.46 void VME::TDCV1x90::SetPairModeResolution (int lead_time_res, int pulse_width_res) const

Here is the call graph for this function:



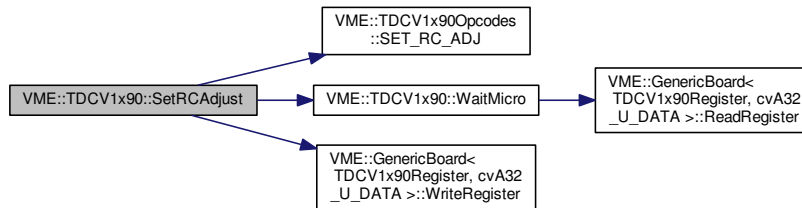
7.38.4.47 void VME::TDCV1x90::SetPol (uint16_t *word1*, uint16_t *word2*) const

Here is the call graph for this function:



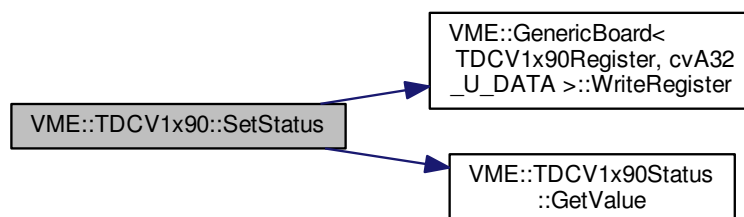
7.38.4.48 void VME::TDCV1x90::SetRCAdjust (int *tdc*, uint16_t *value*) const

Here is the call graph for this function:



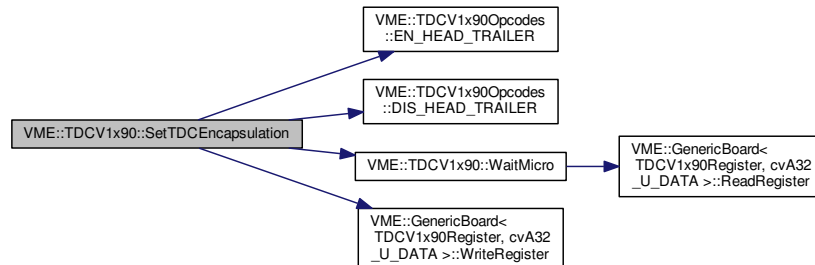
7.38.4.49 void VME::TDCV1x90::SetStatus (const TDCV1x90Status & *status*) const

Here is the call graph for this function:



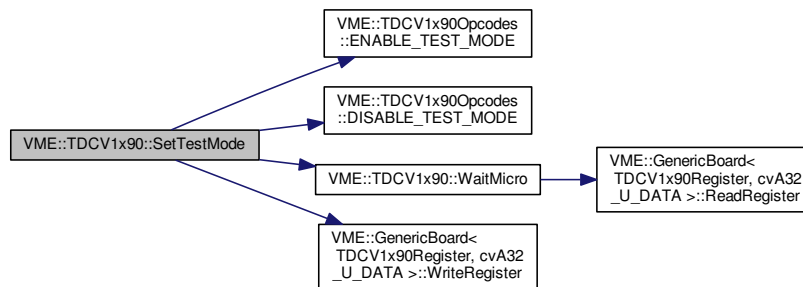
7.38.4.50 void VME::TDCV1x90::SetTDCEncapsulation (bool mode) const

Here is the call graph for this function:



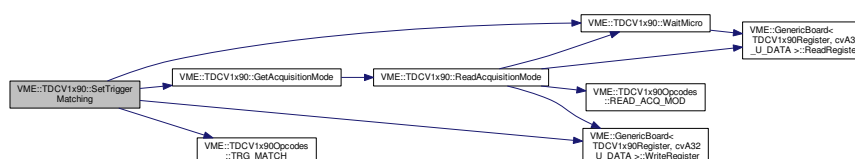
7.38.4.51 void VME::TDCV1x90::SetTestMode (bool en = true) const

Here is the call graph for this function:



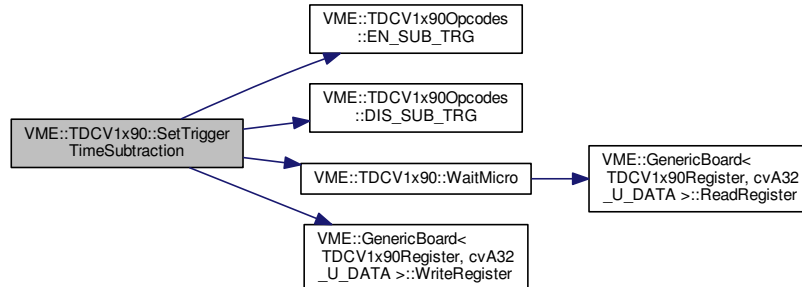
7.38.4.52 void VME::TDCV1x90::SetTriggerMatching ()

Here is the call graph for this function:



7.38.4.53 `void VME::TDCV1x90::SetTriggerTimeSubtraction (bool enabled = true) const`

Here is the call graph for this function:



7.38.4.54 `void VME::TDCV1x90::SetVerboseLevel (unsigned short verb = 1) [inline]`

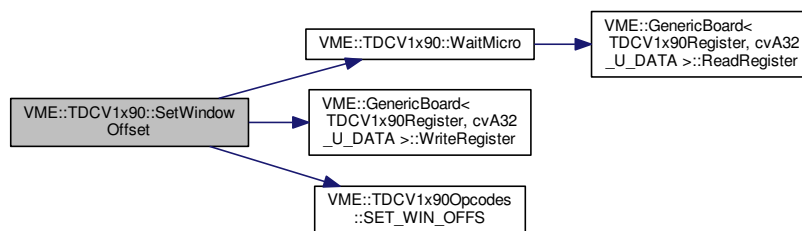
7.38.4.55 `void VME::TDCV1x90::SetWindowOffset (const int16_t & offs) const`

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

<code>in</code>	<i>Window</i>	offset, in units of clock cycles
-----------------	---------------	----------------------------------

Here is the call graph for this function:



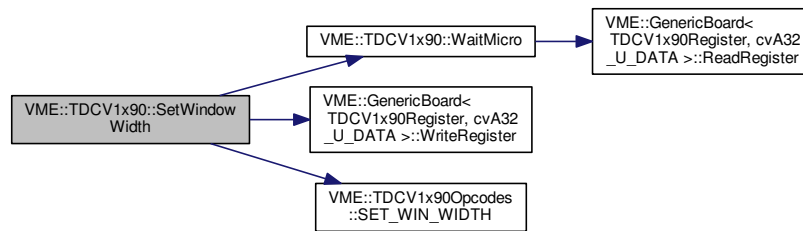
7.38.4.56 `void VME::TDCV1x90::SetWindowWidth (const uint16_t & width)`

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

Parameters

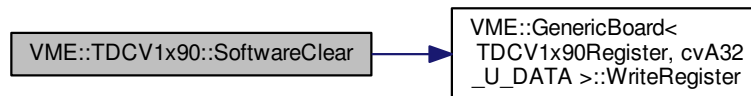
<code>in</code>	<i>Window</i>	width, in units of clock cycles
-----------------	---------------	---------------------------------

Here is the call graph for this function:



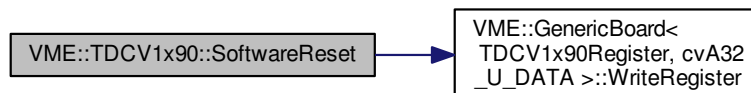
7.38.4.57 `bool VME::TDCV1x90::SoftwareClear () const`

Here is the call graph for this function:



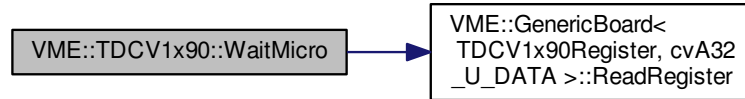
7.38.4.58 `bool VME::TDCV1x90::SoftwareReset () const`

Here is the call graph for this function:



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

Here is the call graph for this function:



7.38.5 Field Documentation

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

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

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

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

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

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

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

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

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

7.38.5.10 `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

7.39 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](#)

7.39.1 Detailed Description

TDC control register.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

Jun 2015

7.39.2 Constructor & Destructor Documentation

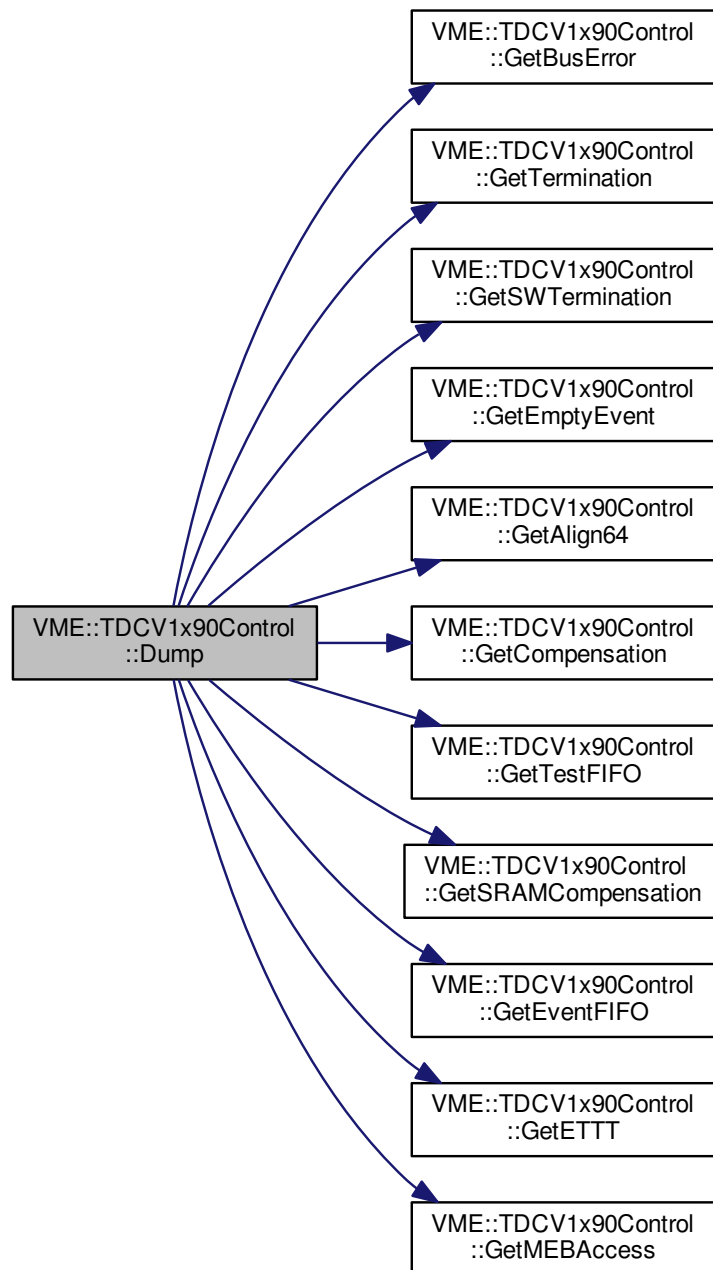
7.39.2.1 `VME::TDCV1x90Control::TDCV1x90Control (const uint16_t & word)` `[inline]`

7.39.2.2 `virtual VME::TDCV1x90Control::~~TDCV1x90Control ()` `[inline],[virtual]`

7.39.3 Member Function Documentation

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

Here is the call graph for this function:



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

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

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

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

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

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

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

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

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

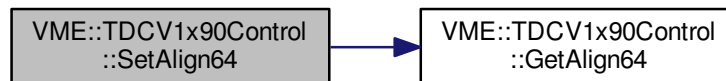
7.39.3.11 `bool VME::TDCV1x90Control::GetTermination () const [inline]`

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

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

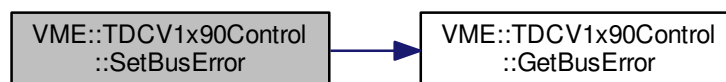
7.39.3.14 `void VME::TDCV1x90Control::SetAlign64 (bool sw) [inline]`

Here is the call graph for this function:



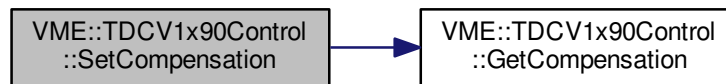
7.39.3.15 `void VME::TDCV1x90Control::SetBusError (bool sw) [inline]`

Here is the call graph for this function:



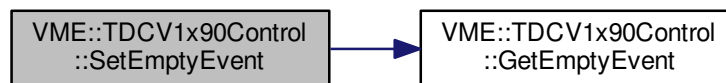
7.39.3.16 void VME::TDCV1x90Control::SetCompensation (bool *sw*) [inline]

Here is the call graph for this function:



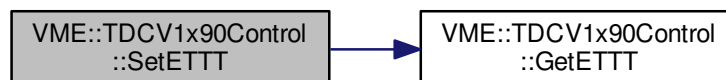
7.39.3.17 void VME::TDCV1x90Control::SetEmptyEvent (bool *sw*) [inline]

Here is the call graph for this function:



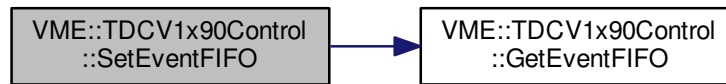
7.39.3.18 void VME::TDCV1x90Control::SetETTT (bool *sw*) [inline]

Here is the call graph for this function:



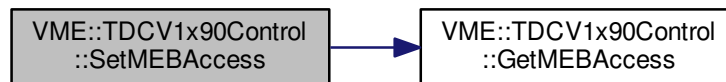
7.39.3.19 void VME::TDCV1x90Control::SetEventFIFO (bool *sw*) [inline]

Here is the call graph for this function:



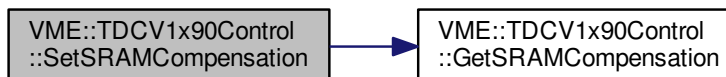
7.39.3.20 void VME::TDCV1x90Control::SetMEBAccess (bool *sw*) [inline]

Here is the call graph for this function:



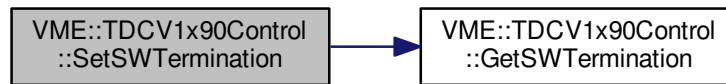
7.39.3.21 void VME::TDCV1x90Control::SetSRAMCompensation (bool *sw*) [inline]

Here is the call graph for this function:



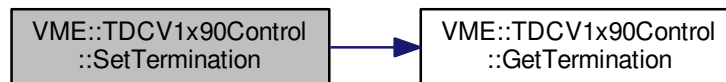
7.39.3.22 `void VME::TDCV1x90Control::SetSWTermination (bool sw) [inline]`

Here is the call graph for this function:



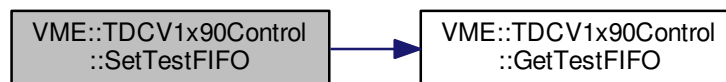
7.39.3.23 `void VME::TDCV1x90Control::SetTermination (bool sw) [inline]`

Here is the call graph for this function:



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

Here is the call graph for this function:



7.39.4 Field Documentation

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

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

- `include/VME_TDCV1x90.h`

7.40 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](#)

7.40.1 Detailed Description

TDC status register.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

Jun 2015

7.40.2 Member Enumeration Documentation

7.40.2.1 enum VME::TDCV1x90Status::TDCResolution

Enumerator

R_800ps

R_200ps

R_100ps

R_25ps

7.40.3 Constructor & Destructor Documentation

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

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

7.40.4 Member Function Documentation

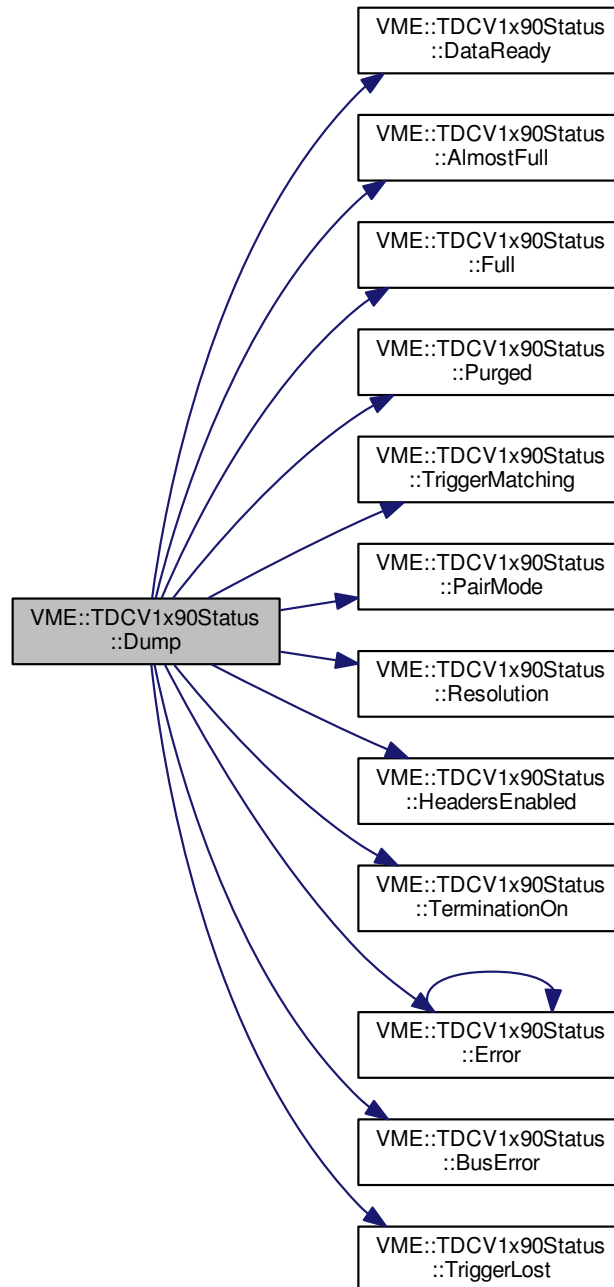
7.40.4.1 bool VME::TDCV1x90Status::AlmostFull () const [inline]

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

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

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

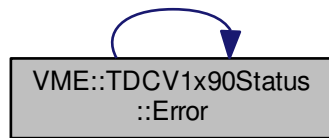
Here is the call graph for this function:



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

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

Here is the call graph for this function:



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

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

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

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

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

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

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

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

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

7.40.5 Field Documentation

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

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

- `include/VME_TDCV1x90.h`

7.41 VME::TDCV1x90TriggerConfig Class Reference

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

Public Member Functions

- [TDCV1x90TriggerConfig \(\)](#)
- [~TDCV1x90TriggerConfig \(\)](#)
- void [SetWord](#) (unsigned int word_id, uint16_t word_content)
- void [Dump](#) () const

- unsigned short [GetWidthWidth](#) () const
- short [GetWindowOffset](#) () const
- unsigned short [GetExtraSearchWindowWidth](#) () const
- unsigned short [GetRejectMargin](#) () const
- bool [HasTriggerTimeSubtraction](#) () const

Private Attributes

- std::vector< uint16_t > [fWords](#)

7.41.1 Constructor & Destructor Documentation

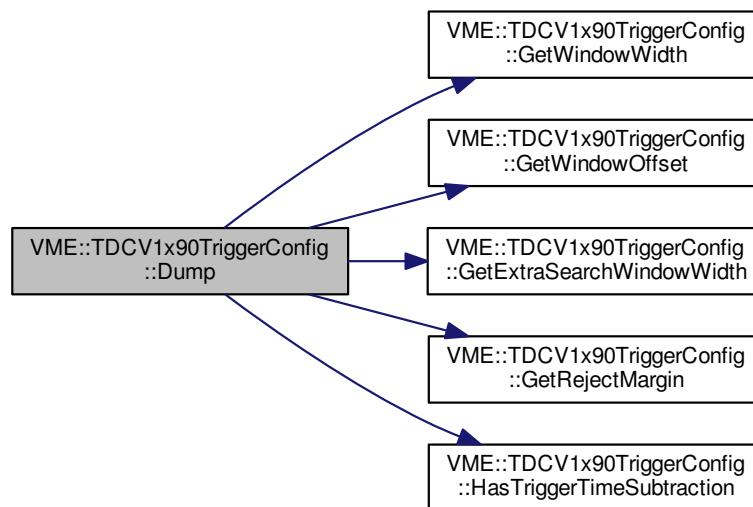
7.41.1.1 VME::TDCV1x90TriggerConfig::TDCV1x90TriggerConfig () [\[inline\]](#)

7.41.1.2 VME::TDCV1x90TriggerConfig::~~TDCV1x90TriggerConfig () [\[inline\]](#)

7.41.2 Member Function Documentation

7.41.2.1 void VME::TDCV1x90TriggerConfig::Dump () const [\[inline\]](#)

Here is the call graph for this function:



7.41.2.2 unsigned short VME::TDCV1x90TriggerConfig::GetExtraSearchWindowWidth () const [\[inline\]](#)

7.41.2.3 unsigned short VME::TDCV1x90TriggerConfig::GetRejectMargin () const [\[inline\]](#)

7.41.2.4 short VME::TDCV1x90TriggerConfig::GetWindowOffset () const [\[inline\]](#)

7.41.2.5 unsigned short VME::TDCV1x90TriggerConfig::GetWidthWidth () const [\[inline\]](#)

7.41.2.6 `bool VME::TDCV1x90TriggerConfig::HasTriggerTimeSubtraction () const` `[inline]`

7.41.2.7 `void VME::TDCV1x90TriggerConfig::SetWord (unsigned int word_id, uint16_t word_content)` `[inline]`

7.41.3 Field Documentation

7.41.3.1 `std::vector<uint16_t> VME::TDCV1x90TriggerConfig::fWords` `[private]`

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

- `include/VME_TDCV1x90.h`

7.42 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](#)

7.42.1 Field Documentation

7.42.1.1 `uint32_t VME::trailead_t::ettt`

7.42.1.2 `uint32_t VME::trailead_t::event_count`

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

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

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

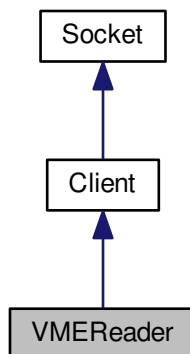
The documentation for this struct was generated from the following file:

- `include/VME_TDCV1x90.h`

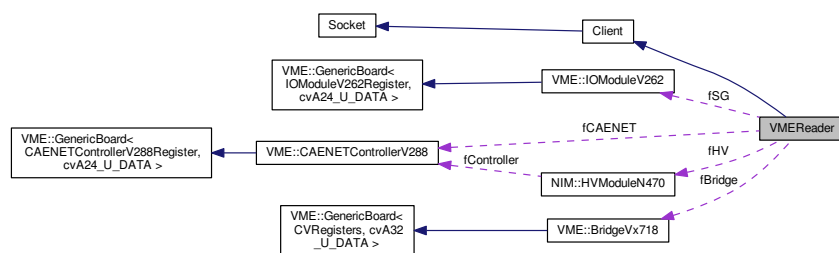
7.43 VMEReader Class Reference

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

Inheritance diagram for VMEReader:



Collaboration diagram for VMEReader:



Public Types

- enum [GlobalAcqMode](#) { [ContinuousStorage](#) = 0x0, [TriggerStart](#) = 0x1, [TriggerMatching](#) = 0x2 }

Public Member Functions

- [VMEReader](#) (const char *device, [VME::BridgeType](#) type, bool on_socket=true)
- virtual [~VMEReader](#) ()
- void [ReadXML](#) (const char *filename)
Load an XML configuration file.
- void [ReadXML](#) (std::string filename)
- [GlobalAcqMode](#) [GetGlobalAcquisitionMode](#) () const
- 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](#) (uint32_t address)
Return the pointer to the FPGA board connected to this controller (if any ; 0 otherwise)
- [VME::FPGAUnitCollection GetFPGAUnitCollection](#) ()
- void [NewRun](#) () const
- void [NewBurst](#) () const
- unsigned int [GetRunNumber](#) () const
Ask the socket master a run number.
- unsigned int [GetBurstNumber](#) () const
- 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](#) (uint32_t addr)
Send a clear signal to both the TDC boards.
- void [AddHVModule](#) (uint32_t vme_address, uint16_t nim_address)
Add a high voltage module (controlled by a VME-CAENET controller) to the DAQ.
- [NIM::HVModuleN470 * GetHVModule](#) ()
Retrieve the [NIM](#) high voltage module.
- void [SetOutputFile](#) (uint32_t tdc_address, std::string filename)
Set the path to the output file where the DAQ is to write.
- std::string [GetOutputFile](#) (uint32_t tdc_address)
Return the path to the output file the DAQ is currently writing to.
- void [SendOutputFile](#) (uint32_t tdc_address) const
Send the path to the output file through the socket.
- void [BroadcastNewBurst](#) (unsigned int burst_id) const
- void [BroadcastTriggerRate](#) (unsigned int burst_id, unsigned long num_triggers) const
- void [BroadcastHVStatus](#) (unsigned short channel_id, const [NIM::HVModuleN470ChannelValues](#) &val) const
- void [LogHVValues](#) (unsigned short channel_id, const [NIM::HVModuleN470ChannelValues](#) &val) const
- bool [UseSocket](#) () const
- void [Abort](#) ()
Abort data collection for all modules on the bus handled by the bridge.

Private Types

- typedef std::map< uint32_t, std::string > [OutputFiles](#)

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::FPGAUnitCollection](#) [fFPGACollection](#)
Pointer to the *VME* general purpose FPGA unit object.
- [VME::CAENETControllerV288](#) * [fCAENET](#)
Pointer to the *VME* CAENET controller.
- [NIM::HVModuleN470](#) * [fHV](#)
Pointer to the *NIM* high voltage module (passing through the CAENET controller)
- bool [fOnSocket](#)
Are we dealing with socket message passing?
- bool [fIsPulserStarted](#)
Is the bridge's pulser already started?
- [OutputFiles](#) [fOutputFiles](#)
- [GlobalAcqMode](#) [fGlobalAcqMode](#)

Additional Inherited Members

7.43.1 Detailed Description

VME reader object to fetch events on a HPTDC board

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

4 May 2015

7.43.2 Member Typedef Documentation

7.43.2.1 `typedef std::map<uint32_t, std::string> VMEReader::OutputFiles` [private]

7.43.3 Member Enumeration Documentation

7.43.3.1 `enum VMEReader::GlobalAcqMode`

Enumerator

ContinuousStorage

TriggerStart

TriggerMatching

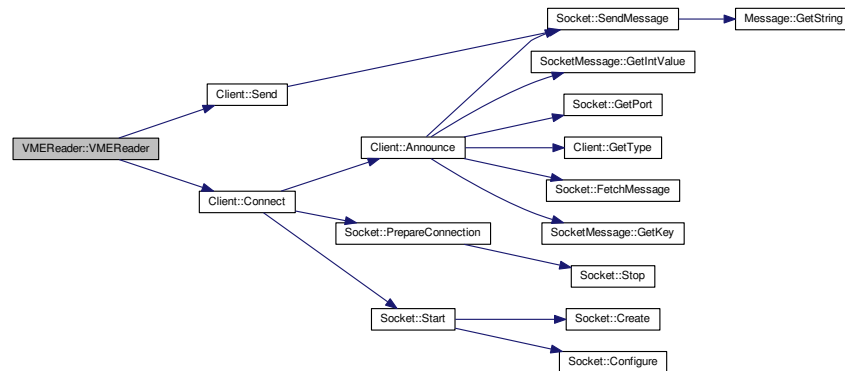
7.43.4 Constructor & Destructor Documentation

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

Parameters

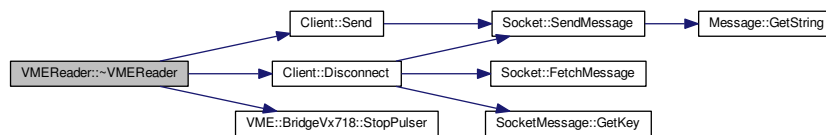
in	<i>device</i>	Path to the device (/dev/xxx)
in	<i>type</i>	Bridge model
in	<i>on_socket</i>	Are we trying to connect through the socket?

Here is the call graph for this function:



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

Here is the call graph for this function:

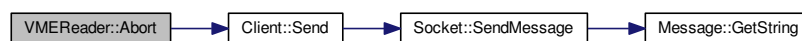


7.43.5 Member Function Documentation

7.43.5.1 void VMEReader::Abort ()

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

Here is the call graph for this function:



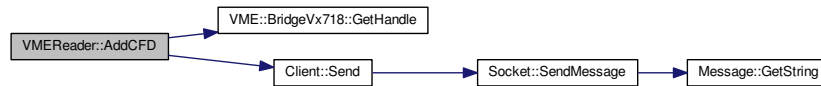
7.43.5.2 void VMEReader::AddCFD (uint32_t address)

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:



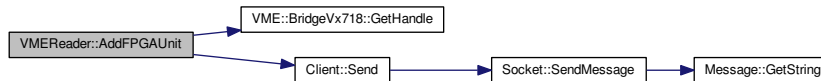
7.43.5.3 void VMEReaders::AddFPGAUnit (uint32_t *address*)

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
----	----------------	---

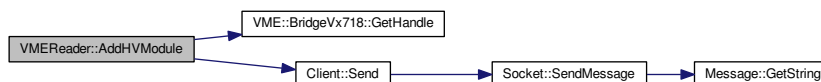
Here is the call graph for this function:



7.43.5.4 void VMEReaders::AddHVModule (uint32_t *vme_address*, uint16_t *nim_address*)

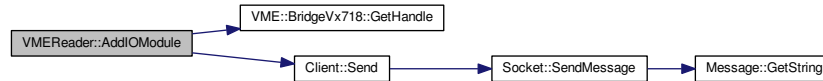
Add a high voltage module (controlled by a VME-CAENET controller) to the DAQ.

Here is the call graph for this function:



7.43.5.5 void VMEReader::AddIOModule (uint32_t address)

Here is the call graph for this function:



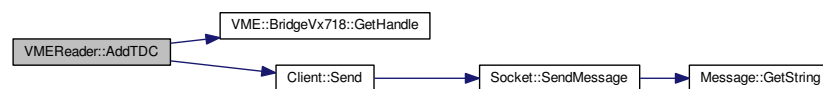
7.43.5.6 void VMEReader::AddTDC (uint32_t address)

Add a TDC to handle.

Parameters

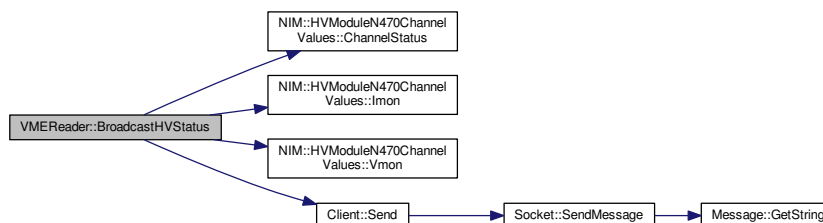
in	address	32-bit address of the TDC module on the VME bus Create a new TDC handler for the VME bus
----	---------	--

Here is the call graph for this function:



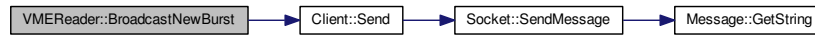
7.43.5.7 void VMEReader::BroadcastHVStatus (unsigned short channel_id, const NIM::HVModuleN470ChannelValues & val) const

Here is the call graph for this function:



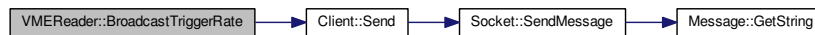
7.43.5.8 void VMEReader::BroadcastNewBurst (unsigned int *burst_id*) const

Here is the call graph for this function:



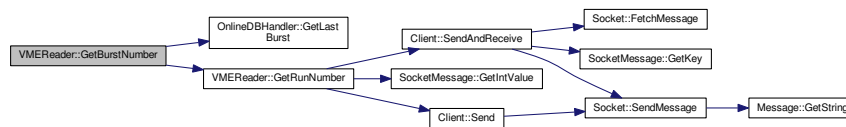
7.43.5.9 void VMEReader::BroadcastTriggerRate (unsigned int *burst_id*, unsigned long *num_triggers*) const

Here is the call graph for this function:



7.43.5.10 unsigned int VMEReader::GetBurstNumber () const [inline]

Here is the call graph for this function:



7.43.5.11 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.

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

7.43.5.13 VME::FPGAUnitV1495* VMEReader::GetFPGAUnit (uint32_t *address*) [inline]

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

7.43.5.14 VME::FPGAUnitCollection VMEReader::GetFPGAUnitCollection () [inline]

7.43.5.15 GlobalAcqMode VMEReader::GetGlobalAcquisitionMode () const [inline]

7.43.5.16 NIM::HVModuleN470* VMEReader::GetHVModule () [inline]

Retrieve the [NIM](#) high voltage module.

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

7.43.5.18 **size_t** VMEReader::GetNumCFD () const [inline]

7.43.5.19 **size_t** VMEReader::GetNumTDC () const [inline]

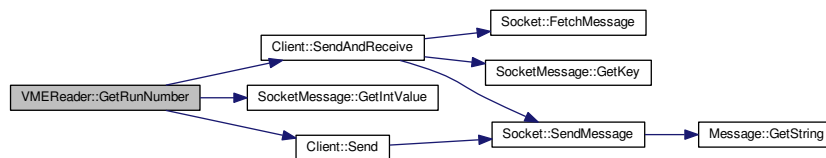
7.43.5.20 **std::string** VMEReader::GetOutputFile (**uint32_t** *tdc_address*) [inline]

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

7.43.5.21 **unsigned int** VMEReader::GetRunNumber () const

Ask the socket master a run number.

Here is the call graph for this function:



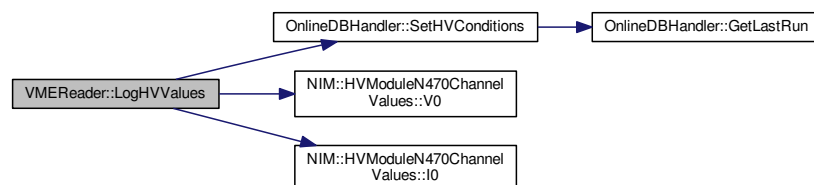
7.43.5.22 **VME::TDCV1x90*** VMEReader::GetTDC (**uint32_t** *address*) [inline]

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

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

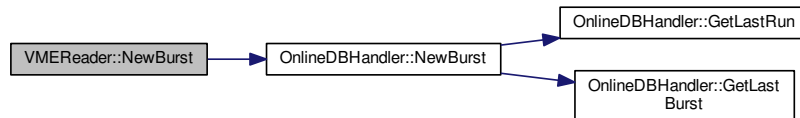
7.43.5.24 **void** VMEReader::LogHVValues (**unsigned short** *channel_id*, **const** **NIM::HVModuleN470ChannelValues** & *val*) const

Here is the call graph for this function:



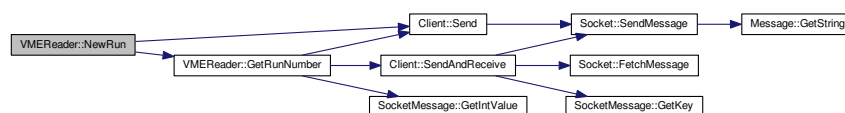
7.43.5.25 void VMEReader::NewBurst () const [inline]

Here is the call graph for this function:



7.43.5.26 void VMEReader::NewRun () const

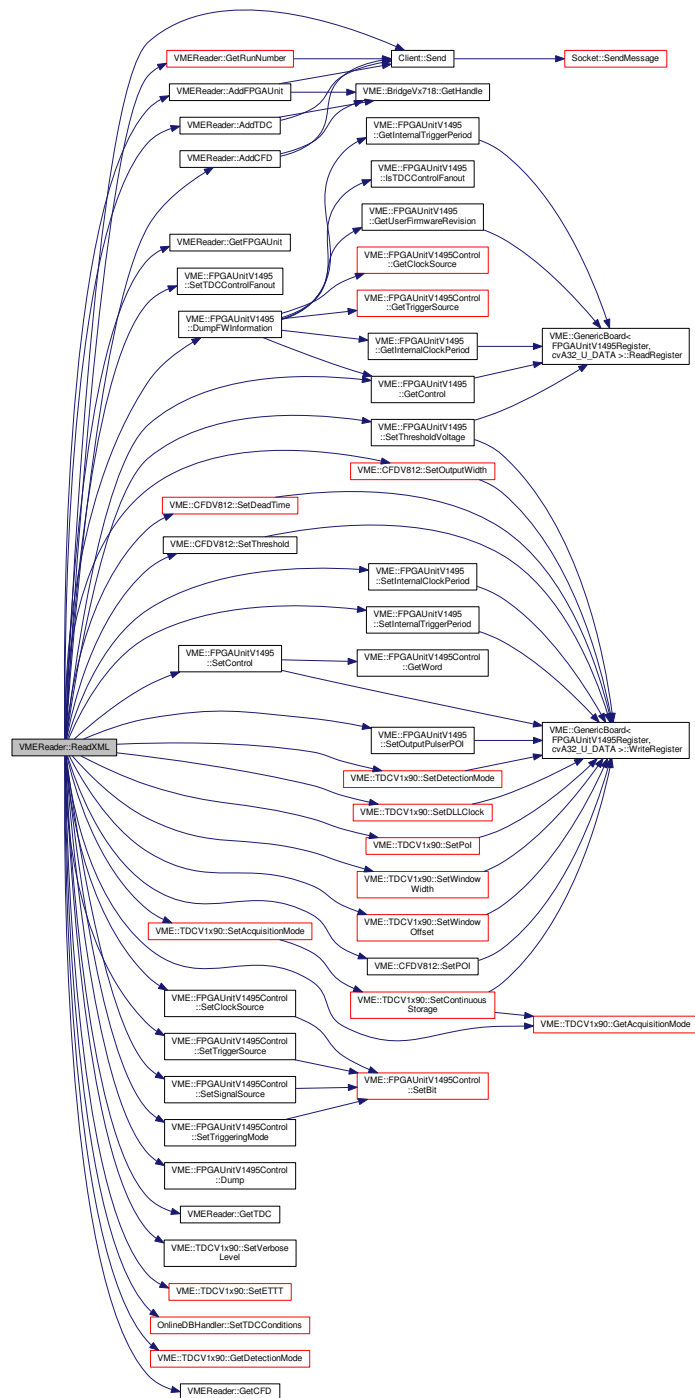
Here is the call graph for this function:



7.43.5.27 void VMEReader::ReadXML (const char * filename)

Load an XML configuration file.

Here is the call graph for this function:



7.43.5.28 `void VMEReader::ReadXML (std::string filename) [inline]`

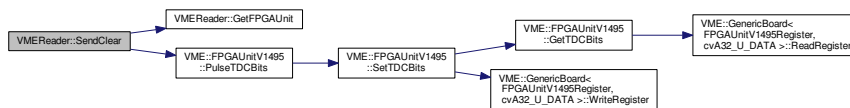
Here is the call graph for this function:



7.43.5.29 `void VMEReader::SendClear (uint32_t addr) [inline]`

Send a clear signal to both the TDC boards.

Here is the call graph for this function:



7.43.5.30 `void VMEReader::SendOutputFile (uint32_t tdc_address) const`

Send the path to the output file through the socket.

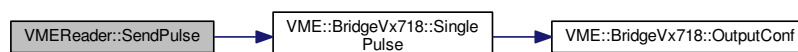
Here is the call graph for this function:



7.43.5.31 `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:



7.43.5.32 `void VMEReader::SetOutputFile (uint32_t tdc_address, std::string filename)`

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

7.43.5.33 `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:



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

Stop the bridge's pulse generator [faulty].

Here is the call graph for this function:



7.43.5.35 `bool VMEReader::UseSocket () const` `[inline]`

7.43.6 Field Documentation

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

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

7.43.6.2 `VME::CAENETControllerV288* VMEReader::fCAENET` `[private]`

Pointer to the [VME](#) CAENET controller.

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

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

7.43.6.4 VME::FPGAUnitCollection VMEReaders::fFPGACollection [private]

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

7.43.6.5 GlobalAcqMode VMEReaders::fGlobalAcqMode [private]**7.43.6.6 NIM::HVModuleN470* VMEReaders::fHV** [private]

Pointer to the [NIM](#) high voltage module (passing through the CAENET controller)

7.43.6.7 bool VMEReaders::fIsPulserStarted [private]

Is the bridge's pulser already started?

7.43.6.8 bool VMEReaders::fOnSocket [private]

Are we dealing with socket message passing?

7.43.6.9 OutputFiles VMEReaders::fOutputFiles [private]

Path to the current output files the DAQ is writing to (indexed by the TDC id)

7.43.6.10 VME::IOModuleV262* VMEReaders::fSG [private]

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

7.43.6.11 VME::TDCCollection VMEReaders::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/VMEReaders.h
- src/VMEReaders.cpp

Index

- ~BridgeVx718
 - VME::BridgeVx718, [25](#)
- ~BridgeVx718Control
 - VME::BridgeVx718Control, [28](#)
- ~BridgeVx718Status
 - VME::BridgeVx718Status, [30](#)
- ~CAENETControllerV288
 - VME::CAENETControllerV288, [33](#)
- ~CAENETControllerV288Status
 - VME::CAENETControllerV288Status, [35](#)
- ~CFDV812
 - VME::CFDV812, [37](#)
- ~ChannelStatus
 - NIM::HVModuleN470Values::ChannelStatus, [42](#)
- ~Client
 - Client, [44](#)
- ~DQMProcess
 - DQM::DQMProcess, [51](#)
- ~FPGAUnitV1495
 - VME::FPGAUnitV1495, [60](#)
- ~FPGAUnitV1495Control
 - VME::FPGAUnitV1495Control, [73](#)
- ~FileReader
 - FileReader, [56](#)
- ~GastofCanvas
 - DQM::GastofCanvas, [79](#)
- ~GenericBoard
 - VME::GenericBoard, [81](#)
- ~HVModuleN470
 - NIM::HVModuleN470, [84](#)
- ~HVModuleN470ChannelValues
 - NIM::HVModuleN470ChannelValues, [89](#)
- ~HVModuleN470Values
 - NIM::HVModuleN470Values, [91](#)
- ~IOModuleV262
 - VME::IOModuleV262, [93](#)
- ~Logger
 - Logger, [95](#)
- ~Message
 - Message, [98](#)
- ~Messenger
 - Messenger, [101](#)
- ~OnlineDBHandler
 - OnlineDBHandler, [108](#)
- ~PCIInterfaceA2818
 - VME::PCIInterfaceA2818, [110](#)
- ~PPSCanvas
 - DQM::PPSCanvas, [112](#)
- ~QuarticCanvas
 - DQM::QuarticCanvas, [116](#)
- ~Socket
 - Socket, [121](#)
- ~SocketMessage
 - SocketMessage, [130](#)
- ~TDCErrorFlag
 - VME::TDCErrorFlag, [134](#)
- ~TDCEvent
 - VME::TDCEvent, [137](#)
- ~TDCMeasurement
 - VME::TDCMeasurement, [143](#)
- ~TDCV1x90
 - VME::TDCV1x90, [149](#)
- ~TDCV1x90Control
 - VME::TDCV1x90Control, [168](#)
- ~TDCV1x90Status
 - VME::TDCV1x90Status, [175](#)
- ~TDCV1x90TriggerConfig
 - VME::TDCV1x90TriggerConfig, [178](#)
- ~VMEReader
 - VMEReader, [183](#)
- AUTOLOAD_DEF_CONFI
 - VME::TDCV1x90Opcodes, [20](#)
- AUTOLOAD_USER_CONF
 - VME::TDCV1x90Opcodes, [20](#)
- Abort
 - VMEReader, [183](#)
- abort
 - VME::TDCV1x90, [149](#)
- AcceptConnections
 - Socket, [121](#)
- acq_mode
 - file_header_t, [54](#)
- AcquisitionMode
 - VME, [14](#)
- Action
 - DQM::DQMProcess, [51](#)
- AddCFD
 - VMEReader, [183](#)
- AddClient
 - Messenger, [101](#)
- AddFPGAUnit
 - VMEReader, [184](#)
- AddHVModule
 - VMEReader, [184](#)
- AddIOModule
 - VMEReader, [184](#)
- AddTDC
 - VMEReader, [185](#)

- Alarm
 - NIM::HVModuleN470Values::ChannelStatus, 42
- AlmostFull
 - VME::TDCV1x90Status, 175
- Announce
 - Client, 45
- Bind
 - Socket, 121
- BridgeType
 - VME, 14
- BridgeVx718
 - VME::BridgeVx718, 25
- BridgeVx718Control
 - VME::BridgeVx718Control, 28
- BridgeVx718Status
 - VME::BridgeVx718Status, 30
- Broadcast
 - Messenger, 101
- BroadcastHVStatus
 - VMEReader, 185
- BroadcastNewBurst
 - VMEReader, 185
- BroadcastTriggerRate
 - VMEReader, 186
- Build
 - DQM::GastofCanvas, 79
 - DQM::PPSCanvas, 113
 - DQM::QuarticCanvas, 117
- BuildTables
 - OnlineDBHandler, 109
- burst_id
 - OnlineDBHandler::BurstInfo, 31
- BurstInfos
 - OnlineDBHandler, 108
- BusError
 - VME::TDCV1x90Status, 175
- c1
 - DQM::GastofCanvas, 80
 - DQM::PPSCanvas, 113
 - DQM::QuarticCanvas, 117
- c2
 - DQM::GastofCanvas, 80
 - DQM::PPSCanvas, 113
 - DQM::QuarticCanvas, 118
- CAEN_V1718
 - VME, 15
- CAEN_V2718
 - VME, 15
- CAENETControllerV288
 - VME::CAENETControllerV288, 33
- CAENETControllerV288Answer
 - VME, 15
- CAENETControllerV288Register
 - VME, 15
- CAENETControllerV288Status
 - VME::CAENETControllerV288Status, 35
- CFDCollection
 - VME, 14
- CFDV812
 - VME::CFDV812, 37
- CFDV812Register
 - VME, 15
- cH0
 - VME, 16
- cH1
 - VME, 16
- cH2
 - VME, 16
- cH3
 - VME, 16
- CLEAR_KEEP_TOKEN
 - VME::TDCV1x90Opcodes, 20
- CLIENT
 - Socket communication objects, 9
- CONT_STOR
 - VME::TDCV1x90Opcodes, 20
- CONT_STORAGE
 - VME, 14
- ChannelStatus
 - NIM::HVModuleN470ChannelValues, 89
 - NIM::HVModuleN470Values::ChannelStatus, 42
- CheckBoardVersion
 - VME::FPGAUnitV1495, 60
- CheckConfiguration
 - VME::BridgeVx718, 26
 - VME::CFDV812, 37
 - VME::TDCV1x90, 149
- CheckPCIInterface
 - VME::BridgeVx718, 26
- Clear
 - FileReader, 56
- ClearOutputPulser
 - VME::FPGAUnitV1495, 60
- Client, 43
 - ~Client, 44
 - Announce, 45
 - Client, 44
 - Connect, 45
 - Disconnect, 46
 - fClientId, 48
 - flsConnected, 48
 - fType, 48
 - GetType, 46
 - ParseMessage, 46
 - Receive, 46, 47
 - Send, 47
 - SendAndReceive, 48
- ClockSource
 - VME::FPGAUnitV1495Control, 72
- cnBusy
 - VME, 15
- cnIncorrectHCC
 - VME, 15
- cnIncorrectValue
 - VME, 15

- cnNoData
 - VME, [15](#)
- cnSuccess
 - VME, [15](#)
- cnUnrecognizedCode
 - VME, [15](#)
- cnWrongModuleAddress
 - VME, [15](#)
- coarse
 - VME::GlobalOffset, [83](#)
- Configure
 - Socket, [121](#)
- Connect
 - Client, [45](#)
 - Messenger, [102](#)
- contents
 - LogRedirector, [96](#)
- ContinuousStorage
 - VME::FPGAUnitV1495Control, [72](#)
 - VMEReader, [182](#)
- Create
 - Socket, [121](#)
- DAQ
 - Socket communication objects, [9](#)
- DEFAULT_SETUP_REG
 - VME::TDCV1x90Opcodes, [20](#)
- DETECTOR
 - Socket communication objects, [9](#)
- DIS_ALL_CHANNEL
 - VME::TDCV1x90Opcodes, [20](#)
- DIS_CHANNEL
 - VME::TDCV1x90Opcodes, [20](#)
- DIS_ERROR_BYPASS
 - VME::TDCV1x90Opcodes, [20](#)
- DIS_ERROR_MARK
 - VME::TDCV1x90Opcodes, [20](#)
- DIS_HEAD_TRAILER
 - VME::TDCV1x90Opcodes, [20](#)
- DIS_SUB_TRG
 - VME::TDCV1x90Opcodes, [20](#)
- DISABLE_TEST_MODE
 - VME::TDCV1x90Opcodes, [20](#)
- DLL_Direct_LowRes
 - VME::TDCV1x90, [148](#)
- DLL_PLL_HighRes
 - VME::TDCV1x90, [148](#)
- DLL_PLL_LowRes
 - VME::TDCV1x90, [148](#)
- DLL_PLL_MedRes
 - VME::TDCV1x90, [148](#)
- DLLMode
 - VME::TDCV1x90, [148](#)
- DQM, [11](#)
 - Socket communication objects, [9](#)
- DQM::DQMProcess, [49](#)
 - ~DQMProcess, [51](#)
 - Action, [51](#)
 - DQMProcess, [51](#)
- fAddressesCanProcess, [53](#)
- fDetectorType, [53](#)
- fOrder, [53](#)
- fRunNumber, [53](#)
- IsInRun, [51](#)
- NewPlot, [51](#)
- ParseMessage, [52](#)
- Run, [52](#)
- UpdatedPlot, [51](#)
- DQM::GastofCanvas, [76](#)
 - ~GastofCanvas, [79](#)
 - Build, [79](#)
 - c1, [80](#)
 - c2, [80](#)
 - DrawGrid, [79](#)
 - fBoardId, [80](#)
 - fHeight, [80](#)
 - fHist, [80](#)
 - fLabel1, [80](#)
 - fLabel2, [80](#)
 - fLabel3, [80](#)
 - fLabel4, [80](#)
 - fLabelsDrawn, [80](#)
 - fLegend, [80](#)
 - fLegendNumEntries, [80](#)
 - fLegendX, [80](#)
 - fLegendY, [80](#)
 - fRunDate, [80](#)
 - fRunId, [80](#)
 - fSpillId, [81](#)
 - fUpperLabel, [81](#)
 - fUpperLabelText, [81](#)
 - fWidth, [81](#)
 - FillChannel, [79](#)
 - GastofCanvas, [78](#)
 - GetCoordinates, [79](#)
 - Grid, [79](#)
 - Save, [79](#)
 - SetRunInfo, [80](#)
 - SetUpperLabel, [80](#)
- DQM::GastofCanvas::Coord, [48](#)
 - x, [48](#)
 - y, [49](#)
- DQM::PPSCanvas, [111](#)
 - ~PPSCanvas, [112](#)
 - Build, [113](#)
 - c1, [113](#)
 - c2, [113](#)
 - DrawGrid, [113](#)
 - fHeight, [113](#)
 - fLabel1, [113](#)
 - fLabel2, [113](#)
 - fLabel3, [113](#)
 - fLabelsDrawn, [113](#)
 - fLegend, [114](#)
 - fLegendNumEntries, [114](#)
 - fLegendX, [114](#)
 - fLegendY, [114](#)

- fRunDate, 114
- fRunId, 114
- fUpperLabel, 114
- fUpperLabelText, 114
- fWidth, 114
- Grid, 113
- PPSCanvas, 112
- Save, 113
- SetRunInfo, 113
- SetUpperLabel, 113
- DQM::QuarticCanvas, 114
- ~QuarticCanvas, 116
- Build, 117
- c1, 117
- c2, 118
- DrawGrid, 117
- fBoardId, 118
- fHeight, 118
- fHist, 118
- fLabel1, 118
- fLabel2, 118
- fLabel3, 118
- fLabel4, 118
- fLabelsDrawn, 118
- fLegend, 118
- fLegendNumEntries, 118
- fLegendX, 118
- fLegendY, 118
- fRunDate, 118
- fRunId, 118
- fSpillId, 118
- fUpperLabel, 118
- fUpperLabelText, 118
- fWidth, 118
- FillChannel, 117
- GetCoordinates, 117
- Grid, 117
- QuarticCanvas, 116
- Save, 117
- SetRunInfo, 117
- SetUpperLabel, 117
- DQM::QuarticCanvas::Coord, 49
- x, 49
- y, 49
- DQMProcess
 - DQM::DQMProcess, 51
- DataReady
 - VME::TDCV1x90Status, 175
- DeadTimeCalculator
 - VME::CFDV812, 37
- det_mode
 - file_header_t, 54
- DetectionMode
 - VME, 15
- detector
 - OnlineDBHandler::TDCConditions, 133
- DisableChannel
 - NIM::HVModuleN470, 84
- VME::TDCV1x90, 149
- Disconnect
 - Client, 46
 - Messenger, 102
- DisconnectClient
 - Messenger, 102
- DrawGrid
 - DQM::GastofCanvas, 79
 - DQM::PPSCanvas, 113
 - DQM::QuarticCanvas, 117
- Dump
 - FileReader, 56
 - Message, 98
 - NIM::HVModuleN470ChannelValues, 89
 - NIM::HVModuleN470Values, 91
 - NIM::HVModuleN470Values::ChannelStatus, 42
 - SocketMessage, 130
 - VME::BridgeVx718Status, 30
 - VME::FPGAUnitV1495Control, 73
 - VME::TDCErrorFlag, 134
 - VME::TDCEvent, 137
 - VME::TDCMeasurement, 143
 - VME::TDCV1x90Control, 168
 - VME::TDCV1x90Status, 175
 - VME::TDCV1x90TriggerConfig, 178
- DumpConnected
 - Socket, 122
- DumpFWInformation
 - VME::FPGAUnitV1495, 61
- EN_ALL_CHANNEL
 - VME::TDCV1x90Opcodes, 20
- EN_CHANNEL
 - VME::TDCV1x90Opcodes, 20
- EN_ERROR_BYPASS
 - VME::TDCV1x90Opcodes, 20
- EN_ERROR_MARK
 - VME::TDCV1x90Opcodes, 20
- EN_HEAD_TRAILER
 - VME::TDCV1x90Opcodes, 20
- EN_SUB_TRG
 - VME::TDCV1x90Opcodes, 20
- ENABLE_TEST_MODE
 - VME::TDCV1x90Opcodes, 20
- ETTT
 - VME::TDCEvent, 137
- EXTRA_SEARCH_WIN_WIDTH
 - VME, 18
- EnableChannel
 - NIM::HVModuleN470, 84
 - VME::TDCV1x90, 150
- Enabled
 - NIM::HVModuleN470Values::ChannelStatus, 42
- Error
 - VME::TDCV1x90Status, 176
- ettt
 - VME::trailead_t, 179
- event_count
 - VME::trailead_t, 179

- EventType
 - VME::TDCEvent, [137](#)
- ExternalClock
 - VME::FPGAUnitV1495Control, [72](#)
- ExternalSignal
 - VME::FPGAUnitV1495Control, [72](#)
- ExternalTrigger
 - VME::FPGAUnitV1495Control, [73](#)
- fAcquisitionMode
 - VME::TDCV1x90, [167](#)
- fAddress
 - NIM::HVModuleN470, [87](#)
 - Socket, [124](#)
- fAddressesCanProcess
 - DQM::DQMProcess, [53](#)
- fBaseAddr
 - VME::GenericBoard, [82](#)
- fBoardId
 - DQM::GastofCanvas, [80](#)
 - DQM::QuarticCanvas, [118](#)
- fBridge
 - VMEReader, [191](#)
- fBuffer
 - Logger, [95](#)
 - Socket, [124](#)
 - VME::TDCV1x90, [167](#)
- fCAENET
 - VMEReader, [191](#)
- fCFDCollection
 - VMEReader, [191](#)
- fChannelId
 - NIM::HVModuleN470ChannelValues, [90](#)
- fClientId
 - Client, [48](#)
- fController
 - NIM::HVModuleN470, [87](#)
- fDB
 - OnlineDBHandler, [110](#)
- fDetectionMode
 - VME::TDCV1x90, [167](#)
- fDetectorType
 - DQM::DQMProcess, [53](#)
- fErrorMarks
 - VME::TDCV1x90, [167](#)
- fEvents
 - VME::TDCMeasurement, [145](#)
- fFPGACollection
 - VMEReader, [191](#)
- fFile
 - FileReader, [57](#)
- fGlobalAcqMode
 - VMEReader, [192](#)
- fHV
 - VMEReader, [192](#)
- fHandle
 - VME::GenericBoard, [82](#)
 - VME::PCIInterfaceA2818, [110](#)
- fHasIRQ
 - VME::BridgeVx718, [27](#)
- fHeader
 - FileReader, [57](#)
- fHeight
 - DQM::GastofCanvas, [80](#)
 - DQM::PPSCanvas, [113](#)
 - DQM::QuarticCanvas, [118](#)
- fHist
 - DQM::GastofCanvas, [80](#)
 - DQM::QuarticCanvas, [118](#)
- fIsConnected
 - Client, [48](#)
- fIsPulserStarted
 - VMEReader, [192](#)
- fIsTDCCControlFanout
 - VME::FPGAUnitV1495, [71](#)
- fLabel1
 - DQM::GastofCanvas, [80](#)
 - DQM::PPSCanvas, [113](#)
 - DQM::QuarticCanvas, [118](#)
- fLabel2
 - DQM::GastofCanvas, [80](#)
 - DQM::PPSCanvas, [113](#)
 - DQM::QuarticCanvas, [118](#)
- fLabel3
 - DQM::GastofCanvas, [80](#)
 - DQM::PPSCanvas, [113](#)
 - DQM::QuarticCanvas, [118](#)
- fLabel4
 - DQM::GastofCanvas, [80](#)
 - DQM::QuarticCanvas, [118](#)
- fLabelsDrawn
 - DQM::GastofCanvas, [80](#)
 - DQM::PPSCanvas, [113](#)
 - DQM::QuarticCanvas, [118](#)
- fLegend
 - DQM::GastofCanvas, [80](#)
 - DQM::PPSCanvas, [114](#)
 - DQM::QuarticCanvas, [118](#)
- fLegendNumEntries
 - DQM::GastofCanvas, [80](#)
 - DQM::PPSCanvas, [114](#)
 - DQM::QuarticCanvas, [118](#)
- fLegendX
 - DQM::GastofCanvas, [80](#)
 - DQM::PPSCanvas, [114](#)
 - DQM::QuarticCanvas, [118](#)
- fLegendY
 - DQM::GastofCanvas, [80](#)
 - DQM::PPSCanvas, [114](#)
 - DQM::QuarticCanvas, [118](#)
- fMap
 - VME::TDCMeasurement, [145](#)
- fMaster
 - Socket, [124](#)
- fMessage
 - SocketMessage, [132](#)
- fNumAttempts

- Messenger, 107
- fNumEvents
 - FileReader, 57
- fOnSocket
 - VMEReader, 192
- fOrder
 - DQM::DQMProcess, 53
- fOutputFiles
 - VMEReader, 192
- FPGAUnitCollection
 - VME, 14
- FPGAUnitV1495
 - VME::FPGAUnitV1495, 60
- FPGAUnitV1495Control
 - VME::FPGAUnitV1495Control, 73
- FPGAUnitV1495DACCH
 - VME, 16
- FPGAUnitV1495Register
 - VME, 16
- fPID
 - Messenger, 107
- fPort
 - Socket, 124
- fReadFds
 - Socket, 124
- fReadoutMode
 - FileReader, 57
- fRedirect
 - LogRedirector, 97
- fRunDate
 - DQM::GastofCanvas, 80
 - DQM::PPSCanvas, 114
 - DQM::QuarticCanvas, 118
- fRunId
 - DQM::GastofCanvas, 80
 - DQM::PPSCanvas, 114
 - DQM::QuarticCanvas, 118
- fRunNumber
 - DQM::DQMProcess, 53
- fSG
 - VMEReader, 192
- fSS
 - LogRedirector, 97
- fScalerStarted
 - VME::FPGAUnitV1495, 71
- fSocketId
 - Socket, 124
- fSocketsConnected
 - Socket, 124
- fSpillId
 - DQM::GastofCanvas, 81
 - DQM::QuarticCanvas, 118
- fStderrPipe
 - Messenger, 107
- fStdoutPipe
 - Messenger, 107
- fStream
 - Logger, 95
- fString
 - Message, 98
- fTDCCollection
 - VMEReader, 192
- fType
 - Client, 48
- fUpperLabel
 - DQM::GastofCanvas, 81
 - DQM::PPSCanvas, 114
 - DQM::QuarticCanvas, 118
- fUpperLabelText
 - DQM::GastofCanvas, 81
 - DQM::PPSCanvas, 114
 - DQM::QuarticCanvas, 118
- fValues
 - NIM::HVModuleN470ChannelValues, 90
 - NIM::HVModuleN470Values, 91
- fVerb
 - VME::TDCV1x90, 167
- fWidth
 - DQM::GastofCanvas, 81
 - DQM::PPSCanvas, 114
 - DQM::QuarticCanvas, 118
- fWindowWidth
 - VME::TDCV1x90, 167
- fWord
 - NIM::HVModuleN470Values::ChannelStatus, 42
 - VME::BridgeVx718Control, 29
 - VME::BridgeVx718Status, 30
 - VME::CAENETControllerV288Status, 35
 - VME::FPGAUnitV1495Control, 76
 - VME::TDCErrorFlag, 135
 - VME::TDCEvent, 141
 - VME::TDCV1x90Control, 173
 - VME::TDCV1x90Status, 177
- fWords
 - VME::TDCV1x90TriggerConfig, 179
- fWriteTime
 - FileReader, 57
- FetchBuffer
 - VME::CAENETControllerV288, 33
- FetchEvents
 - VME::TDCV1x90, 150
- FetchMessage
 - Socket, 122
- file_header_t, 53
 - acq_mode, 54
 - det_mode, 54
 - magic, 54
 - num_hptdc, 54
 - run_id, 54
 - spill_id, 54
- FileReader, 54
 - ~FileReader, 56
 - Clear, 56
 - Dump, 56
 - fFile, 57
 - fHeader, 57

- fNumEvents, [57](#)
- fReadoutMode, [57](#)
- fWriteTime, [57](#)
- FileReader, [55](#)
- GetAcquisitionMode, [56](#)
- GetBurstId, [56](#)
- GetDetectionMode, [56](#)
- GetNextEvent, [56](#)
- GetNextMeasurement, [56](#)
- GetNumEvents, [56](#)
- GetNumTDCs, [57](#)
- GetRunId, [57](#)
- IsOpen, [57](#)
- Open, [57](#)
- FillChannel
 - DQM::GastofCanvas, [79](#)
 - DQM::QuarticCanvas, [117](#)
- Filler
 - VME::TDCEvent, [137](#)
- fine
 - VME::GlobalOffset, [83](#)
- Full
 - VME::TDCV1x90Status, [177](#)
- gEnd
 - VME::TDCV1x90, [167](#)
- GastofCanvas
 - DQM::GastofCanvas, [78](#)
- GenericBoard
 - VME::GenericBoard, [81](#)
- GetAcquisitionMode
 - FileReader, [56](#)
 - VME::TDCV1x90, [150](#)
- GetAddressIncrement
 - VME::BridgeVx718Control, [28](#)
- GetAlign64
 - VME::TDCV1x90Control, [169](#)
- GetArbiterType
 - VME::BridgeVx718Control, [28](#)
- GetBERR
 - VME::BridgeVx718Status, [30](#)
- GetBLTEventNumberRegister
 - VME::TDCV1x90, [151](#)
- GetBit
 - VME::FPGAUnitV1495Control, [73](#)
- GetBunchId
 - VME::TDCEvent, [137](#)
 - VME::TDCMeasurement, [144](#)
- GetBurstId
 - FileReader, [56](#)
- GetBurstNumber
 - VMEReader, [186](#)
- GetBusError
 - VME::TDCV1x90Control, [169](#)
- GetBusReqLevel
 - VME::BridgeVx718Control, [28](#)
- GetBusTimeout
 - VME::BridgeVx718Control, [28](#)
- GetCAENFirmwareRevision
 - VME::FPGAUnitV1495, [61](#)
- GetCFD
 - VMEReader, [186](#)
- GetCFDCollection
 - VMEReader, [186](#)
- GetChannelDeadTime
 - VME::TDCV1x90, [151](#)
- GetChannelId
 - VME::TDCEvent, [138](#)
 - VME::TDCMeasurement, [144](#)
- GetChannelStatus
 - NIM::HVModuleN470Values, [91](#)
- GetCleanedValue
 - SocketMessage, [130](#)
- GetClockSource
 - VME::FPGAUnitV1495Control, [73](#)
- GetCompensation
 - VME::TDCV1x90Control, [169](#)
- GetControl
 - VME::FPGAUnitV1495, [61](#)
 - VME::TDCV1x90, [151](#)
- GetCoordinates
 - DQM::GastofCanvas, [79](#)
 - DQM::QuarticCanvas, [117](#)
- GetDLLClock
 - VME::TDCV1x90, [152](#)
- GetDTACK
 - VME::BridgeVx718Status, [30](#)
- GetDetectionMode
 - FileReader, [56](#)
 - VME::TDCV1x90, [152](#)
- GetDipSwitch
 - VME::BridgeVx718Status, [30](#)
- GetETTT
 - VME::TDCEvent, [138](#)
 - VME::TDCMeasurement, [144](#)
 - VME::TDCV1x90, [152](#)
 - VME::TDCV1x90Control, [170](#)
- GetEmptyEvent
 - VME::TDCV1x90Control, [170](#)
- GetErrorFlags
 - VME::TDCEvent, [138](#)
- GetErrorMarks
 - VME::TDCV1x90, [152](#)
- GetEventCount
 - VME::TDCEvent, [139](#)
- GetEventCounter
 - VME::TDCV1x90, [152](#)
- GetEventFIFO
 - VME::TDCV1x90Control, [170](#)
- GetEventId
 - VME::TDCEvent, [139](#)
 - VME::TDCMeasurement, [144](#)
- GetEventStored
 - VME::TDCV1x90, [153](#)
- GetExtraSearchWindowWidth
 - VME::TDCV1x90TriggerConfig, [178](#)
- GetFIFOSize

- VME::TDCV1x90, [153](#)
- GetFPGAUnit
 - VMEReader, [186](#)
- GetFPGAUnitCollection
 - VMEReader, [186](#)
- GetFWRevision
 - NIM::HVModuleN470, [85](#)
 - VME::PCIInterfaceA2818, [110](#)
- GetFirmwareRevision
 - VME::TDCV1x90, [153](#)
- GetFixedCode
 - VME::CFDV812, [37](#)
- GetGeo
 - VME::TDCEvent, [139](#)
- GetGeoAddress
 - VME::FPGAUnitV1495, [62](#)
- GetGlobalAcquisitionMode
 - VMEReader, [186](#)
- GetGlobalOffset
 - VME::TDCV1x90, [154](#)
- GetHVModule
 - VMEReader, [186](#)
- GetHandle
 - VME::BridgeVx718, [26](#)
- GetHardwareRevision
 - VME::FPGAUnitV1495, [62](#)
- GetIOModule
 - VMEReader, [186](#)
- GetIRQStatus
 - VME::BridgeVx718, [26](#)
- GetIdentifier
 - VME::IOModuleV262, [93](#)
- GetIntValue
 - SocketMessage, [130](#)
- GetInternalClockPeriod
 - VME::FPGAUnitV1495, [62](#)
- GetInternalTriggerPeriod
 - VME::FPGAUnitV1495, [63](#)
- GetInterruptReq
 - VME::BridgeVx718Control, [28](#)
- GetKey
 - Message, [98](#)
 - SocketMessage, [130](#)
- GetLastBurst
 - OnlineDBHandler, [109](#)
- GetLastRun
 - OnlineDBHandler, [109](#)
- GetLeadingTime
 - VME::TDCMeasurement, [144](#)
- GetMEBAccess
 - VME::TDCV1x90Control, [170](#)
- GetManufacturerId
 - VME::CFDV812, [37](#)
 - VME::IOModuleV262, [93](#)
- GetModel
 - VME::TDCV1x90, [154](#)
- GetModuleId
 - NIM::HVModuleN470, [85](#)
- GetModuleType
 - VME::CFDV812, [38](#)
 - VME::IOModuleV262, [94](#)
- GetModuleVersion
 - VME::CFDV812, [38](#)
 - VME::IOModuleV262, [94](#)
- GetNextEvent
 - FileReader, [56](#)
- GetNextMeasurement
 - FileReader, [56](#)
- GetNumCFD
 - VMEReader, [187](#)
- GetNumEvents
 - FileReader, [56](#)
- GetNumTDC
 - VMEReader, [187](#)
- GetNumTDCs
 - FileReader, [57](#)
- GetOUI
 - VME::TDCV1x90, [154](#)
- GetOperationStatus
 - VME::CAENETControllerV288Status, [35](#)
- GetOutputDelay
 - VME::FPGAUnitV1495, [63](#)
- GetOutputFile
 - VMEReader, [187](#)
- GetOutputPulser
 - VME::FPGAUnitV1495, [63](#)
- GetPol
 - VME::TDCV1x90, [155](#)
- GetPort
 - Socket, [122](#)
- GetRCAdjust
 - VME::TDCV1x90, [155](#)
- GetRejectMargin
 - VME::TDCV1x90TriggerConfig, [178](#)
- GetReleaseType
 - VME::BridgeVx718Control, [29](#)
- GetRequesterType
 - VME::BridgeVx718Control, [29](#)
- GetResolution
 - VME::TDCV1x90, [155](#)
- GetRunId
 - FileReader, [57](#)
- GetRunInfo
 - OnlineDBHandler, [109](#)
- GetRunNumber
 - VMEReader, [187](#)
- GetRuns
 - OnlineDBHandler, [109](#)
- GetSRAMCompensation
 - VME::TDCV1x90Control, [170](#)
- GetSWTermination
 - VME::TDCV1x90Control, [170](#)
- GetScalerStatus
 - VME::FPGAUnitV1495Control, [73](#)
- GetScalerValue
 - VME::FPGAUnitV1495, [64](#)

- GetSerialNumber
 - VME::CFDV812, [38](#)
 - VME::FPGAUnitV1495, [64](#)
 - VME::IOModuleV262, [94](#)
 - VME::TDCV1x90, [156](#)
- GetSignalSource
 - VME::FPGAUnitV1495Control, [73](#)
- GetSocketId
 - Socket, [122](#)
- GetSocketType
 - Socket, [122](#)
- GetStatus
 - VME::BridgeVx718, [26](#)
 - VME::CAENETControllerV288, [33](#)
 - VME::TDCEvent, [140](#)
 - VME::TDCV1x90, [156](#)
- GetString
 - Message, [98](#)
 - SocketMessage, [130](#)
- GetSysRes
 - VME::BridgeVx718Control, [29](#)
- GetSystemControl
 - VME::BridgeVx718Status, [30](#)
- GetSystemReset
 - VME::BridgeVx718Status, [30](#)
- GetTDC
 - VMEReader, [187](#)
- GetTDCBits
 - VME::FPGAUnitV1495, [64](#)
- GetTDCCollection
 - VMEReader, [187](#)
- GetTDCConditions
 - OnlineDBHandler, [109](#)
- GetTDCEncapsulation
 - VME::TDCV1x90, [156](#)
- GetTDCId
 - VME::TDCEvent, [140](#)
 - VME::TDCMeasurement, [144](#)
- GetTermination
 - VME::TDCV1x90Control, [170](#)
- GetTestFIFO
 - VME::TDCV1x90Control, [170](#)
- GetTestMode
 - VME::TDCV1x90, [157](#)
- GetThresholdVoltage
 - VME::FPGAUnitV1495, [65](#)
- GetTime
 - VME::TDCEvent, [140](#)
- GetToT
 - VME::TDCMeasurement, [144](#)
- GetTrailingTime
 - VME::TDCMeasurement, [145](#)
- GetTriggerConfiguration
 - VME::TDCV1x90, [157](#)
- GetTriggerSource
 - VME::FPGAUnitV1495Control, [74](#)
- GetTriggeringMode
 - VME::FPGAUnitV1495Control, [74](#)
- GetType
 - Client, [46](#)
 - Messenger, [104](#)
 - VME::TDCEvent, [140](#)
- GetUSBType
 - VME::BridgeVx718Status, [30](#)
- GetUserFirmwareRevision
 - VME::FPGAUnitV1495, [65](#)
- GetValue
 - SocketMessage, [130](#)
 - VME::TDCV1x90Control, [170](#)
 - VME::TDCV1x90Status, [177](#)
- GetVectorValue
 - SocketMessage, [131](#)
- GetWidth
 - VME::TDCEvent, [140](#)
- GetWindowOffset
 - VME::TDCV1x90, [157](#)
 - VME::TDCV1x90TriggerConfig, [178](#)
- GetWindowWidth
 - VME::TDCV1x90, [157](#)
 - VME::TDCV1x90TriggerConfig, [178](#)
- GetWord
 - VME::FPGAUnitV1495Control, [74](#)
 - VME::TDCErrorFlag, [134](#)
 - VME::TDCEvent, [141](#)
- GetWordCount
 - VME::TDCEvent, [141](#)
- GlobalAcqMode
 - VMEReader, [182](#)
- GlobalHeader
 - VME::TDCEvent, [137](#)
- GlobalTrailer
 - VME::TDCEvent, [137](#)
- Grid
 - DQM::GastofCanvas, [79](#)
 - DQM::PPSCanvas, [113](#)
 - DQM::QuarticCanvas, [117](#)
- HVEnabled
 - NIM::HVModuleN470Values::ChannelStatus, [42](#)
- HVModuleN470
 - NIM::HVModuleN470, [84](#)
- HVModuleN470ChannelValues
 - NIM::HVModuleN470ChannelValues, [89](#)
- HVModuleN470Opcodes
 - NIM, [11](#)
- HVModuleN470Values
 - NIM::HVModuleN470Values, [91](#)
- HardwareReset
 - VME::TDCV1x90, [157](#)
- HasGroupError
 - VME::TDCErrorFlag, [134](#)
- HasInternalChipError
 - VME::TDCErrorFlag, [134](#)
- HasL1BufferOverflow
 - VME::TDCErrorFlag, [135](#)
- HasReachedEventSizeLimit
 - VME::TDCErrorFlag, [135](#)

- HasReadoutFIFOOverflow
 - VME::TDCErrrorFlag, [135](#)
- HasTriggerFIFOOverflow
 - VME::TDCErrrorFlag, [135](#)
- HasTriggerTimeSubtraction
 - VME::TDCV1x90TriggerConfig, [178](#)
- HeadersEnabled
 - VME::TDCV1x90Status, [177](#)
- I0
 - NIM::HVModuleN470ChannelValues, [89](#)
- I1
 - NIM::HVModuleN470ChannelValues, [90](#)
- INVALID
 - Socket communication objects, [9](#)
- IOModuleV262
 - VME::IOModuleV262, [93](#)
- IOModuleV262Register
 - VME, [16](#)
- IRQ1
 - VME::BridgeVx718, [25](#)
- IRQ2
 - VME::BridgeVx718, [25](#)
- IRQ3
 - VME::BridgeVx718, [25](#)
- IRQ4
 - VME::BridgeVx718, [25](#)
- IRQ5
 - VME::BridgeVx718, [25](#)
- IRQ6
 - VME::BridgeVx718, [25](#)
- IRQ7
 - VME::BridgeVx718, [25](#)
- IRQId
 - VME::BridgeVx718, [25](#)
- Imon
 - NIM::HVModuleN470ChannelValues, [90](#)
 - NIM::HVModuleN470Values, [91](#)
- InputConf
 - VME::BridgeVx718, [26](#)
- InputRead
 - VME::BridgeVx718, [26](#)
- InternalClock
 - VME::FPGAUnitV1495Control, [72](#)
- InternalSignal
 - VME::FPGAUnitV1495Control, [72](#)
- InternalTrigger
 - VME::FPGAUnitV1495Control, [73](#)
- Invalid
 - VME::CAENETControllerV288Status, [34](#)
- IsFromWeb
 - Message, [98](#)
- IsInRun
 - DQM::DQMProcess, [51](#)
- IsOpen
 - FileReader, [57](#)
- IsTDCControlFanout
 - VME::FPGAUnitV1495, [65](#)
- IsTrailing
 - VME::TDCEvent, [141](#)
- IsWebSocket
 - Socket, [122](#)
- Isel
 - NIM::HVModuleN470Values::ChannelStatus, [42](#)
- kBLTEventNumber
 - VME, [17](#)
- kBoardInfo0
 - VME, [17](#)
- kBoardInfo1
 - VME, [17](#)
- kClear
 - VME::FPGAUnitV1495, [59](#)
- kControl
 - VME, [17](#)
- kECLLevelWrite
 - VME, [17](#)
- kEventCounter
 - VME, [17](#)
- kEventFIFO
 - VME, [17](#)
- kEventFIFOStatusRegister
 - VME, [17](#)
- kEventFIFOStoredRegister
 - VME, [17](#)
- kEventStored
 - VME, [17](#)
- kFirmwareRev
 - VME, [17](#)
- kGeoAddress
 - VME, [17](#)
- kIdentifier
 - VME, [17](#)
- kInterruptLevel
 - VME, [17](#)
- kInterruptVector
 - VME, [17](#)
- kMCSTBase
 - VME, [17](#)
- kMCSTControl
 - VME, [17](#)
- kMicro
 - VME, [17](#)
- kMicroHandshake
 - VME, [17](#)
- kModuleReset
 - VME, [17](#)
- kN470ChannelOff
 - NIM, [12](#)
- kN470ChannelOn
 - NIM, [12](#)
- kN470ClearAlarm
 - NIM, [12](#)
- kN470DisableFrontPanel
 - NIM, [12](#)
- kN470EnableFrontPanel
 - NIM, [12](#)
- kN470GeneralInfo

NIM, [12](#)
 kN470IOValue
 NIM, [12](#)
 kN470I1Value
 NIM, [12](#)
 kN470KillAllChannels
 NIM, [12](#)
 kN470MonStatus
 NIM, [12](#)
 kN470NIMLevel
 NIM, [12](#)
 kN470OperationalParams
 NIM, [12](#)
 kN470RampDownValue
 NIM, [12](#)
 kN470RampUpValue
 NIM, [12](#)
 kN470TTLLLevel
 NIM, [12](#)
 kN470TripValue
 NIM, [12](#)
 kN470V0Value
 NIM, [12](#)
 kN470V1Value
 NIM, [12](#)
 kNIMLevelWrite
 VME, [17](#)
 kNIMPulseRead
 VME, [17](#)
 kNIMPulseWrite
 VME, [17](#)
 kOutputBuffer
 VME, [17](#)
 kROMBoard0
 VME, [17](#)
 kROMBoard1
 VME, [17](#)
 kROMBoard2
 VME, [17](#)
 kROMOui0
 VME, [17](#)
 kROMOui1
 VME, [17](#)
 kROMOui2
 VME, [17](#)
 kROMRevis0
 VME, [18](#)
 kROMRevis1
 VME, [18](#)
 kROMRevis2
 VME, [18](#)
 kROMRevis3
 VME, [17](#)
 kROMSerNum0
 VME, [18](#)
 kROMSerNum1
 VME, [18](#)
 kReset
 VME::FPGAUnitV1495, [59](#)
 kSoftwareClear
 VME, [17](#)
 kStatus
 VME, [17](#)
 kTrigger
 VME::FPGAUnitV1495, [59](#)
 kV1495Board0
 VME, [16](#)
 kV1495Board1
 VME, [16](#)
 kV1495Board2
 VME, [16](#)
 kV1495ClockSettings
 VME, [16](#)
 kV1495ConfigurationROM
 VME, [16](#)
 kV1495Control
 VME, [16](#)
 kV1495DelaySettings
 VME, [16](#)
 kV1495FWRevision
 VME, [16](#)
 kV1495GeoAddress
 VME, [16](#)
 kV1495HWRRevision0
 VME, [16](#)
 kV1495HWRRevision1
 VME, [16](#)
 kV1495HWRRevision2
 VME, [16](#)
 kV1495HWRRevision3
 VME, [16](#)
 kV1495ModuleReset
 VME, [16](#)
 kV1495OUI0
 VME, [16](#)
 kV1495OUI1
 VME, [16](#)
 kV1495OUI2
 VME, [16](#)
 kV1495OutputSettings
 VME, [16](#)
 kV1495ScalerCounter
 VME, [16](#)
 kV1495SerNum0
 VME, [16](#)
 kV1495SerNum1
 VME, [16](#)
 kV1495TDCBoardInterface
 VME, [16](#)
 kV1495ThresholdVoltage0
 VME, [16](#)
 kV1495ThresholdVoltage1
 VME, [16](#)
 kV1495TriggerSettings
 VME, [16](#)
 kV1495UserFPGAConfig

- VME, [16](#)
- kV1495UserFPGAFlashMem
 - VME, [16](#)
- kV1495UserFWRevision
 - VME, [16](#)
- kV288DataBuffer
 - VME, [15](#)
- kV288IRQVector
 - VME, [15](#)
- kV288ModuleReset
 - VME, [15](#)
- kV288Status
 - VME, [15](#)
- kV288Transmission
 - VME, [15](#)
- kV812DeadTimeGroup0
 - VME, [15](#)
- kV812DeadTimeGroup1
 - VME, [15](#)
- kV812FixedCode
 - VME, [15](#)
- kV812Info0
 - VME, [15](#)
- kV812Info1
 - VME, [15](#)
- kV812MajorityThreshold
 - VME, [15](#)
- kV812OutputWidthGroup0
 - VME, [15](#)
- kV812OutputWidthGroup1
 - VME, [15](#)
- kV812PatternOfInhibit
 - VME, [15](#)
- kV812TestPulse
 - VME, [15](#)
- kV812ThresholdChannel0
 - VME, [15](#)
- Kill
 - NIM::HVModuleN470Values::ChannelStatus, [42](#)
- LOAD_DEF_CONFIG
 - VME::TDCV1x90Opcodes, [20](#)
- LOAD_USER_CONFIG
 - VME::TDCV1x90Opcodes, [20](#)
- leading
 - VME::trailead_t, [179](#)
- Listen
 - Socket, [122](#)
- LogHVValues
 - VMEReader, [187](#)
- LogRedirector, [96](#)
 - contents, [96](#)
 - fRedirect, [97](#)
 - fSS, [97](#)
 - LogRedirector, [96](#)
- Logger, [95](#)
 - ~Logger, [95](#)
 - fBuffer, [95](#)
 - fStream, [95](#)
- Logger, [95](#)
- MASTER
 - Socket communication objects, [9](#)
- MATCH_WIN_WIDTH
 - VME, [18](#)
- magic
 - file_header_t, [54](#)
- MaxV
 - NIM::HVModuleN470ChannelValues, [90](#)
 - NIM::HVModuleN470Values::ChannelStatus, [42](#)
- Message, [97](#)
 - ~Message, [98](#)
 - Dump, [98](#)
 - fString, [98](#)
 - GetKey, [98](#)
 - GetString, [98](#)
 - IsFromWeb, [98](#)
 - Message, [98](#)
- Messenger, [99](#)
 - ~Messenger, [101](#)
 - AddClient, [101](#)
 - Broadcast, [101](#)
 - Connect, [102](#)
 - Disconnect, [102](#)
 - DisconnectClient, [102](#)
 - fNumAttempts, [107](#)
 - fPID, [107](#)
 - fStderrPipe, [107](#)
 - fStdoutPipe, [107](#)
 - GetType, [104](#)
 - Messenger, [100](#)
 - ProcessMessage, [104](#)
 - Receive, [104](#)
 - Send, [105](#)
 - SendAll, [105](#), [106](#)
 - StartAcquisition, [106](#)
 - StopAcquisition, [106](#)
 - SwitchClientType, [106](#)
- micro_handshake
 - VME, [17](#)
- NIM, [11](#)
 - HVModuleN470Opcodes, [11](#)
 - kN470ChannelOff, [12](#)
 - kN470ChannelOn, [12](#)
 - kN470ClearAlarm, [12](#)
 - kN470DisableFrontPanel, [12](#)
 - kN470EnableFrontPanel, [12](#)
 - kN470GeneralInfo, [12](#)
 - kN470IOValue, [12](#)
 - kN470I1Value, [12](#)
 - kN470KillAllChannels, [12](#)
 - kN470MonStatus, [12](#)
 - kN470NIMLevel, [12](#)
 - kN470OperationalParams, [12](#)
 - kN470RampDownValue, [12](#)
 - kN470RampUpValue, [12](#)
 - kN470TTLLLevel, [12](#)

- kN470TripValue, 12
- kN470V0Value, 12
- kN470V1Value, 12
- NIM::HVModuleN470Values::ChannelStatus, 41
- NIM::HVModuleN470, 83
 - ~HVModuleN470, 84
 - DisableChannel, 84
 - EnableChannel, 84
 - fAddress, 87
 - fController, 87
 - GetFWRevision, 85
 - GetModuleId, 85
 - HVModuleN470, 84
 - ReadChannelValues, 85
 - ReadMonitoringValues, 85
 - ReadRegister, 85
 - SetChannelI0, 86
 - SetChannelI1, 86
 - SetChannelV0, 86
 - SetChannelV1, 86
 - WriteRegister, 87
- NIM::HVModuleN470ChannelValues, 88
 - ~HVModuleN470ChannelValues, 89
 - ChannelStatus, 89
 - Dump, 89
 - fChannelId, 90
 - fValues, 90
 - HVModuleN470ChannelValues, 89
 - I0, 89
 - I1, 90
 - Imon, 90
 - MaxV, 90
 - RampDown, 90
 - RampUp, 90
 - Trip, 90
 - V0, 90
 - V1, 90
 - Vmon, 90
- NIM::HVModuleN470Values, 90
 - ~HVModuleN470Values, 91
 - Dump, 91
 - fValues, 91
 - GetChannelStatus, 91
 - HVModuleN470Values, 91
 - Imon, 91
 - Vmax, 91
 - Vmon, 91
- NIM::HVModuleN470Values::ChannelStatus, 41
 - ~ChannelStatus, 42
 - Alarm, 42
 - ChannelStatus, 42
 - Dump, 42
 - Enabled, 42
 - fWord, 42
 - HVEnabled, 42
 - Isel, 42
 - Kill, 42
 - MaxV, 42
 - NIM, 41
 - NonCalibrated, 42
 - OVC, 42
 - OVV, 42
 - operator<<, 42
 - Polarity, 42
 - RampDown, 42
 - RampUp, 42
 - SignalStandard, 41
 - Standard, 42
 - TTL, 41
 - Trip, 42
 - UNV, 42
 - Vsel, 42
- nchannels
 - VME::TDCV1x90, 167
- NewBurst
 - OnlineDBHandler, 109
 - VMEReader, 187
- NewPlot
 - DQM::DQMProcess, 51
- NewRun
 - OnlineDBHandler, 109
 - VMEReader, 188
- NonCalibrated
 - NIM::HVModuleN470Values::ChannelStatus, 42
- num_hptdc
 - file_header_t, 54
- NumErrors
 - VME::TDCMeasurement, 145
- NumEvents
 - VME::TDCMeasurement, 145
- OLEADING
 - VME, 16
- OTRILING
 - VME, 16
- OVC
 - NIM::HVModuleN470Values::ChannelStatus, 42
- OVV
 - NIM::HVModuleN470Values::ChannelStatus, 42
- Object
 - SocketMessage, 131
- OneVolt
 - VME, 18
- OnlineDBHandler, 107
 - ~OnlineDBHandler, 108
 - BuildTables, 109
 - BurstInfos, 108
 - fDB, 110
 - GetLastBurst, 109
 - GetLastRun, 109
 - GetRunInfo, 109
 - GetRuns, 109
 - GetTDCConditions, 109
 - NewBurst, 109
 - NewRun, 109
 - OnlineDBHandler, 108
 - RunCollection, 108

- Select, [109](#)
- SetHVConditions, [109](#)
- SetTDCConditions, [110](#)
- TDCConditionsCollection, [108](#)
- OnlineDBHandler::BurstInfo, [31](#)
 - burst_id, [31](#)
 - time_start, [31](#)
- OnlineDBHandler::TDCConditions, [133](#)
 - detector, [133](#)
 - operator=, [133](#)
 - operator==, [133](#)
 - run_id, [133](#)
 - tdc_acq_mode, [133](#)
 - tdc_address, [133](#)
 - tdc_det_mode, [133](#)
 - tdc_id, [133](#)
- Open
 - FileReader, [57](#)
- OperationStatus
 - VME::CAENETControllerV288Status, [34](#)
- operator<<
 - NIM::HVModuleN470Values::ChannelStatus, [42](#)
 - VME::CAENETControllerV288, [34](#)
 - VME::TDCErrorFlag, [135](#)
- operator>>
 - VME::CAENETControllerV288, [34](#)
- operator=
 - OnlineDBHandler::TDCConditions, [133](#)
- operator==
 - OnlineDBHandler::TDCConditions, [133](#)
- OutputConf
 - VME::BridgeVx718, [26](#)
- OutputFiles
 - VMEReader, [182](#)
- OutputOff
 - VME::BridgeVx718, [26](#)
- OutputOn
 - VME::BridgeVx718, [26](#)
- OutputWidthCalculator
 - VME::CFDV812, [39](#)
- PAIR
 - VME, [16](#)
- PCIInterfaceA2818
 - VME::PCIInterfaceA2818, [110](#)
- PPSCanvas
 - DQM::PPSCanvas, [112](#)
- pair_lead_res
 - VME::TDCV1x90, [167](#)
- pair_width_res
 - VME::TDCV1x90, [167](#)
- PairMode
 - VME::TDCV1x90Status, [177](#)
- ParseMessage
 - Client, [46](#)
 - DQM::DQMProcess, [52](#)
- Polarity
 - NIM::HVModuleN470Values::ChannelStatus, [42](#)
- PrepareConnection
 - Socket, [122](#)
- ProcessMessage
 - Messenger, [104](#)
- PulseTDCBits
 - VME::FPGAUnitV1495, [65](#)
- Purged
 - VME::TDCV1x90Status, [177](#)
- QuarticCanvas
 - DQM::QuarticCanvas, [116](#)
- r100ps
 - VME, [18](#)
- r200ps
 - VME, [18](#)
- r25ps
 - VME, [18](#)
- r800ps
 - VME, [18](#)
- R_100ps
 - VME::TDCV1x90Status, [174](#)
- R_200ps
 - VME::TDCV1x90Status, [174](#)
- R_25ps
 - VME::TDCV1x90Status, [174](#)
- R_800ps
 - VME::TDCV1x90Status, [174](#)
- READ_ACQ_MOD
 - VME::TDCV1x90Opcodes, [20](#)
- READ_ADJUST_CH
 - VME::TDCV1x90Opcodes, [20](#)
- READ_DEAD_TIME
 - VME::TDCV1x90Opcodes, [20](#)
- READ_DETECTION
 - VME::TDCV1x90Opcodes, [20](#)
- READ_DLL_LOCK
 - VME::TDCV1x90Opcodes, [20](#)
- READ_EEPROM
 - VME::TDCV1x90Opcodes, [20](#)
- READ_EN_PATTERN
 - VME::TDCV1x90Opcodes, [20](#)
- READ_EN_PATTERN32
 - VME::TDCV1x90Opcodes, [21](#)
- READ_ERROR_STATUS
 - VME::TDCV1x90Opcodes, [21](#)
- READ_ERROR_TYPES
 - VME::TDCV1x90Opcodes, [21](#)
- READ_EVENT_SIZE
 - VME::TDCV1x90Opcodes, [21](#)
- READ_FIFO_SIZE
 - VME::TDCV1x90Opcodes, [21](#)
- READ_GLOB_OFFS
 - VME::TDCV1x90Opcodes, [21](#)
- READ_HEAD_TRAILER
 - VME::TDCV1x90Opcodes, [21](#)
- READ_MICRO_REV
 - VME::TDCV1x90Opcodes, [21](#)
- READ_OK
 - VME, [17](#)

- READ_RC_ADJ
 - VME::TDCV1x90Opcodes, [21](#)
- READ_RES
 - VME::TDCV1x90Opcodes, [21](#)
- READ_SETUP_REG
 - VME::TDCV1x90Opcodes, [21](#)
- READ_SETUP_SCANPATH
 - VME::TDCV1x90Opcodes, [21](#)
- READ_SPARE
 - VME::TDCV1x90Opcodes, [21](#)
- READ_STATUS_STREAM
 - VME::TDCV1x90Opcodes, [21](#)
- READ_TDC_ID
 - VME::TDCV1x90Opcodes, [21](#)
- READ_TRG_CONF
 - VME::TDCV1x90Opcodes, [21](#)
- REJECT_MARGIN
 - VME, [18](#)
- RESET_DLL_PLL
 - VME::TDCV1x90Opcodes, [21](#)
- REV_DATE_MICRO_FW
 - VME::TDCV1x90Opcodes, [21](#)
- RampDown
 - NIM::HVModuleN470ChannelValues, [90](#)
 - NIM::HVModuleN470Values::ChannelStatus, [42](#)
- RampUp
 - NIM::HVModuleN470ChannelValues, [90](#)
 - NIM::HVModuleN470Values::ChannelStatus, [42](#)
- ReadAcquisitionMode
 - VME::TDCV1x90, [157](#)
- ReadChannelValues
 - NIM::HVModuleN470, [85](#)
- ReadDetectionMode
 - VME::TDCV1x90, [158](#)
- ReadMonitoringValues
 - NIM::HVModuleN470, [85](#)
- ReadRegister
 - NIM::HVModuleN470, [85](#)
 - VME::GenericBoard, [81](#), [82](#)
- ReadXML
 - VMEReader, [188](#), [189](#)
- Receive
 - Client, [46](#), [47](#)
 - Messenger, [104](#)
- Reset
 - VME::BridgeVx718, [26](#)
 - VME::CAENETControllerV288, [33](#)
- ResetFPGA
 - VME::FPGAUnitV1495, [66](#)
- Resolution
 - VME::TDCV1x90Status, [177](#)
- Run
 - DQM::DQMProcess, [52](#)
- run_id
 - file_header_t, [54](#)
 - OnlineDBHandler::TDCConditions, [133](#)
- RunCollection
 - OnlineDBHandler, [108](#)
- SAVE_RC_ADJ
 - VME::TDCV1x90Opcodes, [21](#)
- SAVE_USER_CONFIG
 - VME::TDCV1x90Opcodes, [21](#)
- SET_ADJUST_CH
 - VME::TDCV1x90Opcodes, [21](#)
- SET_DEAD_TIME
 - VME::TDCV1x90Opcodes, [21](#)
- SET_DETECTION
 - VME::TDCV1x90Opcodes, [21](#)
- SET_DLL_CLOCK
 - VME::TDCV1x90Opcodes, [21](#)
- SET_ERROR_TYPES
 - VME::TDCV1x90Opcodes, [21](#)
- SET_EVENT_SIZE
 - VME::TDCV1x90Opcodes, [21](#)
- SET_FIFO_SIZE
 - VME::TDCV1x90Opcodes, [21](#)
- SET_GLOB_OFFS
 - VME::TDCV1x90Opcodes, [21](#)
- SET_KEEP_TOKEN
 - VME::TDCV1x90Opcodes, [22](#)
- SET_PAIR_RES
 - VME::TDCV1x90Opcodes, [22](#)
- SET_RC_ADJ
 - VME::TDCV1x90Opcodes, [22](#)
- SET_REJ_MARGIN
 - VME::TDCV1x90Opcodes, [22](#)
- SET_SW_MARGIN
 - VME::TDCV1x90Opcodes, [22](#)
- SET_TDC_TSET_OUTPUT
 - VME::TDCV1x90Opcodes, [22](#)
- SET_TR_LEAD_LSB
 - VME::TDCV1x90Opcodes, [22](#)
- SET_WIN_OFFS
 - VME::TDCV1x90Opcodes, [22](#)
- SET_WIN_WIDTH
 - VME::TDCV1x90Opcodes, [22](#)
- Save
 - DQM::GastofCanvas, [79](#)
 - DQM::PPSCanvas, [113](#)
 - DQM::QuarticCanvas, [117](#)
- Select
 - OnlineDBHandler, [109](#)
- SelectConnections
 - Socket, [123](#)
- Send
 - Client, [47](#)
 - Messenger, [105](#)
- SendAll
 - Messenger, [105](#), [106](#)
- SendAndReceive
 - Client, [48](#)
- SendBuffer
 - VME::CAENETControllerV288, [33](#)
- SendClear
 - VMEReader, [190](#)
- SendMessage

- Socket, [123](#)
- SendOutputFile
 - VME::VMEReader, [190](#)
- SendPulse
 - VME::VMEReader, [190](#)
- SetAcquisitionMode
 - VME::TDCV1x90, [158](#)
- SetAlign64
 - VME::TDCV1x90Control, [170](#)
- SetBLTEventNumberRegister
 - VME::TDCV1x90, [158](#)
- SetBit
 - VME::FPGAUnitV1495Control, [74](#)
- SetBusError
 - VME::TDCV1x90Control, [170](#)
- SetChannelDeadTime
 - VME::TDCV1x90, [159](#)
- SetChannelI0
 - NIM::HVModuleN470, [86](#)
- SetChannelI1
 - NIM::HVModuleN470, [86](#)
- SetChannelV0
 - NIM::HVModuleN470, [86](#)
- SetChannelV1
 - NIM::HVModuleN470, [86](#)
- SetClockSource
 - VME::FPGAUnitV1495Control, [75](#)
- SetCompensation
 - VME::TDCV1x90Control, [170](#)
- SetContinuousStorage
 - VME::TDCV1x90, [159](#)
- SetControl
 - VME::FPGAUnitV1495, [66](#)
 - VME::TDCV1x90, [159](#)
- SetDLLClock
 - VME::TDCV1x90, [160](#)
- SetDeadTime
 - VME::CFDV812, [39](#)
- SetDetectionMode
 - VME::TDCV1x90, [160](#)
- SetETTT
 - VME::TDCV1x90, [161](#)
 - VME::TDCV1x90Control, [171](#)
- SetEmptyEvent
 - VME::TDCV1x90Control, [171](#)
- SetErrorMarks
 - VME::TDCV1x90, [160](#)
- SetEventFIFO
 - VME::TDCV1x90Control, [171](#)
- SetEventsCollection
 - VME::TDCMeasurement, [145](#)
- SetFIFOSize
 - VME::TDCV1x90, [161](#)
- SetGlobalOffset
 - VME::TDCV1x90, [161](#)
- SetHVConditions
 - OnlineDBHandler, [109](#)
- SetIRQ
 - VME::BridgeVx718, [26](#)
- SetInternalClockPeriod
 - VME::FPGAUnitV1495, [66](#)
- SetInternalTriggerPeriod
 - VME::FPGAUnitV1495, [68](#)
- SetKeyValue
 - SocketMessage, [131](#), [132](#)
- SetLSBTrailEdge
 - VME::TDCV1x90, [162](#)
- SetMEBAccess
 - VME::TDCV1x90Control, [172](#)
- SetOutputDelay
 - VME::FPGAUnitV1495, [68](#)
- SetOutputFile
 - VME::VMEReader, [190](#)
- SetOutputPulser
 - VME::FPGAUnitV1495, [68](#)
- SetOutputPulserPOI
 - VME::FPGAUnitV1495, [69](#)
- SetOutputWidth
 - VME::CFDV812, [39](#)
- SetPOI
 - VME::CFDV812, [40](#)
- SetPairModeResolution
 - VME::TDCV1x90, [162](#)
- SetPol
 - VME::TDCV1x90, [162](#)
- SetPort
 - Socket, [123](#)
- SetRCAdjust
 - VME::TDCV1x90, [163](#)
- SetRunInfo
 - DQM::GastofCanvas, [80](#)
 - DQM::PPSCanvas, [113](#)
 - DQM::QuarticCanvas, [117](#)
- SetSRAMCompensation
 - VME::TDCV1x90Control, [172](#)
- SetSWTermination
 - VME::TDCV1x90Control, [172](#)
- SetScalerReset
 - VME::FPGAUnitV1495Control, [75](#)
- SetScalerStatus
 - VME::FPGAUnitV1495Control, [75](#)
- SetSignalSource
 - VME::FPGAUnitV1495Control, [75](#)
- SetSocketId
 - Socket, [123](#)
- SetStatus
 - VME::TDCV1x90, [163](#)
- SetTDCBits
 - VME::FPGAUnitV1495, [69](#)
- SetTDCConditions
 - OnlineDBHandler, [110](#)
- SetTDCCControlFanout
 - VME::FPGAUnitV1495, [69](#)
- SetTDCEncapsulation
 - VME::TDCV1x90, [163](#)
- SetTermination

- VME::TDCV1x90Control, [173](#)
- SetTestFIFO
 - VME::TDCV1x90Control, [173](#)
- SetTestMode
 - VME::TDCV1x90, [164](#)
- SetThreshold
 - VME::CFDV812, [40](#)
- SetThresholdVoltage
 - VME::FPGAUnitV1495, [69](#)
- SetTriggerMatching
 - VME::TDCV1x90, [164](#)
- SetTriggerSource
 - VME::FPGAUnitV1495Control, [76](#)
- SetTriggerTimeSubtraction
 - VME::TDCV1x90, [164](#)
- SetTriggeringMode
 - VME::FPGAUnitV1495Control, [76](#)
- SetUpperLabel
 - DQM::GastofCanvas, [80](#)
 - DQM::PPSCanvas, [113](#)
 - DQM::QuarticCanvas, [117](#)
- SetVerboseLevel
 - VME::TDCV1x90, [165](#)
- SetWindowOffset
 - VME::TDCV1x90, [165](#)
- SetWindowWidth
 - VME::TDCV1x90, [165](#)
- SetWord
 - VME::TDCEvent, [141](#)
 - VME::TDCV1x90TriggerConfig, [179](#)
- SignalSource
 - VME::FPGAUnitV1495Control, [72](#)
- SignalStandard
 - NIM::HVModuleN470Values::ChannelStatus, [41](#)
- SinglePulse
 - VME::BridgeVx718, [27](#)
- Socket, [118](#)
 - ~Socket, [121](#)
 - AcceptConnections, [121](#)
 - Bind, [121](#)
 - Configure, [121](#)
 - Create, [121](#)
 - DumpConnected, [122](#)
 - fAddress, [124](#)
 - fBuffer, [124](#)
 - fMaster, [124](#)
 - fPort, [124](#)
 - fReadFds, [124](#)
 - fSocketId, [124](#)
 - fSocketsConnected, [124](#)
 - FetchMessage, [122](#)
 - GetPort, [122](#)
 - GetSocketId, [122](#)
 - GetSocketType, [122](#)
 - IsWebSocket, [122](#)
 - Listen, [122](#)
 - PrepareConnection, [122](#)
 - SelectConnections, [123](#)
 - SendMessage, [123](#)
 - SetPort, [123](#)
 - SetSocketId, [123](#)
 - Socket, [120](#)
 - SocketCollection, [120](#)
 - Start, [123](#)
 - Stop, [124](#)
- Socket communication objects, [9](#)
 - CLIENT, [9](#)
 - DAQ, [9](#)
 - DETECTOR, [9](#)
 - DQM, [9](#)
 - INVALID, [9](#)
 - MASTER, [9](#)
 - SocketType, [9](#)
 - WEBSOCKET_CLIENT, [9](#)
- SocketCollection
 - Socket, [120](#)
- SocketMessage, [125](#)
 - ~SocketMessage, [130](#)
 - Dump, [130](#)
 - fMessage, [132](#)
 - GetCleanedValue, [130](#)
 - GetIntValue, [130](#)
 - GetKey, [130](#)
 - GetString, [130](#)
 - GetValue, [130](#)
 - GetVectorValue, [131](#)
 - Object, [131](#)
 - SetKeyValue, [131](#), [132](#)
 - SocketMessage, [127–130](#)
 - String, [132](#)
- SocketType
 - Socket communication objects, [9](#)
- SoftwareClear
 - VME::TDCV1x90, [166](#)
- SoftwareReset
 - VME::TDCV1x90, [166](#)
- spill_id
 - file_header_t, [54](#)
- Standard
 - NIM::HVModuleN470Values::ChannelStatus, [42](#)
- Start
 - Socket, [123](#)
- StartAcquisition
 - Messenger, [106](#)
- StartPulser
 - VME::BridgeVx718, [27](#)
 - VMEReader, [191](#)
- StartScaler
 - VME::FPGAUnitV1495, [70](#)
- Stop
 - Socket, [124](#)
- StopAcquisition
 - Messenger, [106](#)
- StopPulser
 - VME::BridgeVx718, [27](#)
 - VMEReader, [191](#)

- StopScaler
 - VME::FPGAUnitV1495, [70](#)
- String
 - SocketMessage, [132](#)
- SwitchClientType
 - Messenger, [106](#)
- TDCBits
 - VME::FPGAUnitV1495, [59](#)
- TDCCollection
 - VME, [14](#)
- TDCConditionsCollection
 - OnlineDBHandler, [108](#)
- TDCError
 - VME::TDCEvent, [137](#)
- TDCErrorFlag
 - VME::TDCErrorFlag, [134](#)
- TDCEvent
 - VME::TDCEvent, [137](#)
- TDCEventCollection
 - VME, [14](#)
- TDCHeader
 - VME::TDCEvent, [137](#)
- TDCMeasurement
 - VME::TDCEvent, [137](#)
 - VME::TDCMeasurement, [142](#)
- TDCResolution
 - VME::TDCV1x90Status, [174](#)
- TDCTrailer
 - VME::TDCEvent, [137](#)
- TDCV1x90
 - VME::TDCV1x90, [148](#)
- TDCV1x90Control
 - VME::TDCV1x90Control, [168](#)
- TDCV1x90Register
 - VME, [17](#)
- TDCV1x90Status
 - VME::TDCV1x90Status, [175](#)
- TDCV1x90TriggerConfig
 - VME::TDCV1x90TriggerConfig, [178](#)
- TRAILHEAD
 - VME, [16](#)
- TRG_MATCH
 - VME::TDCV1x90Opcodes, [22](#)
- TRIG_MATCH
 - VME, [14](#)
- TRIG_TIME_SUB
 - VME, [18](#)
- TTL
 - NIM::HVModuleN470Values::ChannelStatus, [41](#)
- tdc_acq_mode
 - OnlineDBHandler::TDCConditions, [133](#)
- tdc_address
 - OnlineDBHandler::TDCConditions, [133](#)
- tdc_det_mode
 - OnlineDBHandler::TDCConditions, [133](#)
- tdc_id
 - OnlineDBHandler::TDCConditions, [133](#)
- TerminationOn
 - VME::TDCV1x90Status, [177](#)
- TestOutputs
 - VME::BridgeVx718, [27](#)
- time_start
 - OnlineDBHandler::BurstInfo, [31](#)
- total_hits
 - VME::trailead_t, [179](#)
- trailead_edge_lsb
 - VME, [18](#)
- trailing
 - VME::trailead_t, [179](#)
- trig_conf
 - VME, [18](#)
- Trigger
 - VME::TDCEvent, [137](#)
- TriggerLost
 - VME::TDCV1x90Status, [177](#)
- TriggerMatching
 - VME::FPGAUnitV1495Control, [72](#)
 - VME::TDCV1x90Status, [177](#)
 - VMEReader, [182](#)
- TriggerSource
 - VME::FPGAUnitV1495Control, [72](#)
- TriggerStart
 - VMEReader, [182](#)
- TriggeringMode
 - VME::FPGAUnitV1495Control, [72](#)
- Trip
 - NIM::HVModuleN470ChannelValues, [90](#)
 - NIM::HVModuleN470Values::ChannelStatus, [42](#)
- UNV
 - NIM::HVModuleN470Values::ChannelStatus, [42](#)
- UPDATE_SETUP_REG
 - VME::TDCV1x90Opcodes, [22](#)
- UPDATE_SETUP_TDC
 - VME::TDCV1x90Opcodes, [22](#)
- UpdatedPlot
 - DQM::DQMProcess, [51](#)
- UseSocket
 - VMEReader, [191](#)
- V0
 - NIM::HVModuleN470ChannelValues, [90](#)
- V1
 - NIM::HVModuleN470ChannelValues, [90](#)
- VME, [12](#)
 - AcquisitionMode, [14](#)
 - BridgeType, [14](#)
 - CAEN_V1718, [15](#)
 - CAEN_V2718, [15](#)
 - CAENETControllerV288Answer, [15](#)
 - CAENETControllerV288Register, [15](#)
 - CFDCollection, [14](#)
 - CFDV812Register, [15](#)
 - ch0, [16](#)
 - ch1, [16](#)
 - ch2, [16](#)
 - ch3, [16](#)

CONT_STORAGE, 14
cnBusy, 15
cnIncorrectHCC, 15
cnIncorrectValue, 15
cnNoData, 15
cnSuccess, 15
cnUnrecognizedCode, 15
cnWrongModuleAddress, 15
DetectionMode, 15
EXTRA_SEARCH_WIN_WIDTH, 18
FPGAUnitCollection, 14
FPGAUnitV1495DACCH, 16
FPGAUnitV1495Register, 16
IOModuleV262Register, 16
kBLTEventNumber, 17
kBoardInfo0, 17
kBoardInfo1, 17
kControl, 17
kECLLevelWrite, 17
kEventCounter, 17
kEventFIFO, 17
kEventFIFOStatusRegister, 17
kEventFIFOStoredRegister, 17
kEventStored, 17
kFirmwareRev, 17
kGeoAddress, 17
kIdentifier, 17
kInterruptLevel, 17
kInterruptVector, 17
kMCSTBase, 17
kMCSTControl, 17
kMicro, 17
kMicroHandshake, 17
kModuleReset, 17
kNIMLevelWrite, 17
kNIMPulseRead, 17
kNIMPulseWrite, 17
kOutputBuffer, 17
kROMBoard0, 17
kROMBoard1, 17
kROMBoard2, 17
kROMOui0, 17
kROMOui1, 17
kROMOui2, 17
kROMRevis0, 18
kROMRevis1, 18
kROMRevis2, 18
kROMRevis3, 17
kROMSerNum0, 18
kROMSerNum1, 18
kSoftwareClear, 17
kStatus, 17
kV1495Board0, 16
kV1495Board1, 16
kV1495Board2, 16
kV1495ClockSettings, 16
kV1495ConfigurationROM, 16
kV1495Control, 16
kV1495DelaySettings, 16
kV1495FWRevision, 16
kV1495GeoAddress, 16
kV1495HWRRevision0, 16
kV1495HWRRevision1, 16
kV1495HWRRevision2, 16
kV1495HWRRevision3, 16
kV1495ModuleReset, 16
kV1495OUI0, 16
kV1495OUI1, 16
kV1495OUI2, 16
kV1495OutputSettings, 16
kV1495ScalerCounter, 16
kV1495SerNum0, 16
kV1495SerNum1, 16
kV1495TDCBoardInterface, 16
kV1495ThresholdVoltage0, 16
kV1495ThresholdVoltage1, 16
kV1495TriggerSettings, 16
kV1495UserFPGAConfig, 16
kV1495UserFPGAFlashMem, 16
kV1495UserFWRevision, 16
kV288DataBuffer, 15
kV288IRQVector, 15
kV288ModuleReset, 15
kV288Status, 15
kV288Transmission, 15
kV812DeadTimeGroup0, 15
kV812DeadTimeGroup1, 15
kV812FixedCode, 15
kV812Info0, 15
kV812Info1, 15
kV812MajorityThreshold, 15
kV812OutputWidthGroup0, 15
kV812OutputWidthGroup1, 15
kV812PatternOfInhibit, 15
kV812TestPulse, 15
kV812ThresholdChannel0, 15
MATCH_WIN_WIDTH, 18
micro_handshake, 17
OLEADING, 16
OTRILING, 16
OneVolt, 18
PAIR, 16
r100ps, 18
r200ps, 18
r25ps, 18
r800ps, 18
READ_OK, 17
REJECT_MARGIN, 18
TDCCollection, 14
TDCEventCollection, 14
TDCV1x90Register, 17
TRAILEAD, 16
TRIG_MATCH, 14
TRIG_TIME_SUB, 18
trailead_edge_lsb, 18
trig_conf, 18

- WIN_OFFSET, 18
- WRITE_OK, 17
- VME::BridgeVx718, 23
 - ~BridgeVx718, 25
 - BridgeVx718, 25
 - CheckConfiguration, 26
 - CheckPCIIInterface, 26
 - fHasIRQ, 27
 - GetHandle, 26
 - GetIRQStatus, 26
 - GetStatus, 26
 - IRQ1, 25
 - IRQ2, 25
 - IRQ3, 25
 - IRQ4, 25
 - IRQ5, 25
 - IRQ6, 25
 - IRQ7, 25
 - IRQId, 25
 - InputConf, 26
 - InputRead, 26
 - OutputConf, 26
 - OutputOff, 26
 - OutputOn, 26
 - Reset, 26
 - SetIRQ, 26
 - SinglePulse, 27
 - StartPulser, 27
 - StopPulser, 27
 - TestOutputs, 27
 - WaitIRQ, 27
- VME::BridgeVx718Control, 27
 - ~BridgeVx718Control, 28
 - BridgeVx718Control, 28
 - fWord, 29
 - GetAddressIncrement, 28
 - GetArbiterType, 28
 - GetBusReqLevel, 28
 - GetBusTimeout, 28
 - GetInterruptReq, 28
 - GetReleaseType, 29
 - GetRequesterType, 29
 - GetSysRes, 29
- VME::BridgeVx718Status, 29
 - ~BridgeVx718Status, 30
 - BridgeVx718Status, 30
 - Dump, 30
 - fWord, 30
 - GetBERR, 30
 - GetDTACK, 30
 - GetDipSwitch, 30
 - GetSystemControl, 30
 - GetSystemReset, 30
 - GetUSBType, 30
- VME::CAENETControllerV288, 31
 - ~CAENETControllerV288, 33
 - CAENETControllerV288, 33
 - FetchBuffer, 33
 - GetStatus, 33
 - operator<<, 34
 - operator>>, 34
 - Reset, 33
 - SendBuffer, 33
 - WaitForResponse, 33
- VME::CAENETControllerV288Status, 34
 - ~CAENETControllerV288Status, 35
 - CAENETControllerV288Status, 35
 - fWord, 35
 - GetOperationStatus, 35
 - Invalid, 34
 - OperationStatus, 34
 - Valid, 34
- VME::CFDV812, 35
 - ~CFDV812, 37
 - CFDV812, 37
 - CheckConfiguration, 37
 - DeadTimeCalculator, 37
 - GetFixedCode, 37
 - GetManufacturerId, 37
 - GetModuleType, 38
 - GetModuleVersion, 38
 - GetSerialNumber, 38
 - OutputWidthCalculator, 39
 - SetDeadTime, 39
 - SetOutputWidth, 39
 - SetPOI, 40
 - SetThreshold, 40
- VME::FPGAUnitV1495, 57
 - ~FPGAUnitV1495, 60
 - CheckBoardVersion, 60
 - ClearOutputPulser, 60
 - DumpFWInformation, 61
 - fIsTDCCControlFanout, 71
 - FPGAUnitV1495, 60
 - fScalerStarted, 71
 - GetCAENFirmwareRevision, 61
 - GetControl, 61
 - GetGeoAddress, 62
 - GetHardwareRevision, 62
 - GetInternalClockPeriod, 62
 - GetInternalTriggerPeriod, 63
 - GetOutputDelay, 63
 - GetOutputPulser, 63
 - GetScalerValue, 64
 - GetSerialNumber, 64
 - GetTDCBits, 64
 - GetThresholdVoltage, 65
 - GetUserFirmwareRevision, 65
 - IsTDCCControlFanout, 65
 - kClear, 59
 - kReset, 59
 - kTrigger, 59
 - PulseTDCBits, 65
 - ResetFPGA, 66
 - SetControl, 66
 - SetInternalClockPeriod, 66

- SetInternalTriggerPeriod, [68](#)
- SetOutputDelay, [68](#)
- SetOutputPulser, [68](#)
- SetOutputPulserPOI, [69](#)
- SetTDCBits, [69](#)
- SetTDCControlFanout, [69](#)
- SetThresholdVoltage, [69](#)
- StartScaler, [70](#)
- StopScaler, [70](#)
- TDCBits, [59](#)
- VME::FPGAUnitV1495Control, [71](#)
 - ~FPGAUnitV1495Control, [73](#)
 - ClockSource, [72](#)
 - ContinuousStorage, [72](#)
 - Dump, [73](#)
 - ExternalClock, [72](#)
 - ExternalSignal, [72](#)
 - ExternalTrigger, [73](#)
 - FPGAUnitV1495Control, [73](#)
 - fWord, [76](#)
 - GetBit, [73](#)
 - GetClockSource, [73](#)
 - GetScalerStatus, [73](#)
 - GetSignalSource, [73](#)
 - GetTriggerSource, [74](#)
 - GetTriggeringMode, [74](#)
 - GetWord, [74](#)
 - InternalClock, [72](#)
 - InternalSignal, [72](#)
 - InternalTrigger, [73](#)
 - SetBit, [74](#)
 - SetClockSource, [75](#)
 - SetScalerReset, [75](#)
 - SetScalerStatus, [75](#)
 - SetSignalSource, [75](#)
 - SetTriggerSource, [76](#)
 - SetTriggeringMode, [76](#)
 - SignalSource, [72](#)
 - TriggerMatching, [72](#)
 - TriggerSource, [72](#)
 - TriggeringMode, [72](#)
- VME::GenericBoard
 - ~GenericBoard, [81](#)
 - fBaseAddr, [82](#)
 - fHandle, [82](#)
 - GenericBoard, [81](#)
 - ReadRegister, [81](#), [82](#)
 - WriteRegister, [82](#)
- VME::GenericBoard< Register, am >, [81](#)
- VME::GlobalOffset, [83](#)
 - coarse, [83](#)
 - fine, [83](#)
- VME::IOModuleV262, [92](#)
 - ~IOModuleV262, [93](#)
 - GetIdentifier, [93](#)
 - GetManufacturerId, [93](#)
 - GetModuleType, [94](#)
 - GetModuleVersion, [94](#)
 - GetSerialNumber, [94](#)
 - IOModuleV262, [93](#)
- VME::PCIInterfaceA2818, [110](#)
 - ~PCIInterfaceA2818, [110](#)
 - fHandle, [110](#)
 - GetFWRevision, [110](#)
 - PCIInterfaceA2818, [110](#)
- VME::TDCErrorFlag, [133](#)
 - ~TDCErrorFlag, [134](#)
 - Dump, [134](#)
 - fWord, [135](#)
 - GetWord, [134](#)
 - HasGroupError, [134](#)
 - HasInternalChipError, [134](#)
 - HasL1BufferOverflow, [135](#)
 - HasReachedEventSizeLimit, [135](#)
 - HasReadoutFIFOOverflow, [135](#)
 - HasTriggerFIFOOverflow, [135](#)
 - operator<<, [135](#)
 - TDCErrorFlag, [134](#)
- VME::TDCEvent, [135](#)
 - ~TDCEvent, [137](#)
 - Dump, [137](#)
 - ETTT, [137](#)
 - EventType, [137](#)
 - fWord, [141](#)
 - Filler, [137](#)
 - GetBunchId, [137](#)
 - GetChannelId, [138](#)
 - GetETTT, [138](#)
 - GetErrorFlags, [138](#)
 - GetEventCount, [139](#)
 - GetEventId, [139](#)
 - GetGeo, [139](#)
 - GetStatus, [140](#)
 - GetTDCId, [140](#)
 - GetTime, [140](#)
 - GetType, [140](#)
 - GetWidth, [140](#)
 - GetWord, [141](#)
 - GetWordCount, [141](#)
 - GlobalHeader, [137](#)
 - GlobalTrailer, [137](#)
 - IsTrailing, [141](#)
 - SetWord, [141](#)
 - TDCError, [137](#)
 - TDCEvent, [137](#)
 - TDCHeader, [137](#)
 - TDCMeasurement, [137](#)
 - TDCTrailer, [137](#)
 - Trigger, [137](#)
- VME::TDCMeasurement, [142](#)
 - ~TDCMeasurement, [143](#)
 - Dump, [143](#)
 - fEvents, [145](#)
 - fMap, [145](#)
 - GetBunchId, [144](#)
 - GetChannelId, [144](#)

- GetETTT, 144
- GetEventId, 144
- GetLeadingTime, 144
- GetTDCId, 144
- GetToT, 144
- GetTrailingTime, 145
- NumErrors, 145
- NumEvents, 145
- SetEventsCollection, 145
- TDCMeasurement, 142
- VME::TDCV1x90, 145
 - ~TDCV1x90, 149
 - abort, 149
 - CheckConfiguration, 149
 - DLL_Direct_LowRes, 148
 - DLL_PLL_HighRes, 148
 - DLL_PLL_LowRes, 148
 - DLL_PLL_MedRes, 148
 - DLLMode, 148
 - DisableChannel, 149
 - EnableChannel, 150
 - fAcquisitionMode, 167
 - fBuffer, 167
 - fDetectionMode, 167
 - fErrorMarks, 167
 - fVerb, 167
 - fWindowWidth, 167
 - FetchEvents, 150
 - gEnd, 167
 - GetAcquisitionMode, 150
 - GetBLTEventNumberRegister, 151
 - GetChannelDeadTime, 151
 - GetControl, 151
 - GetDLLClock, 152
 - GetDetectionMode, 152
 - GetETTT, 152
 - GetErrorMarks, 152
 - GetEventCounter, 152
 - GetEventStored, 153
 - GetFIFOSize, 153
 - GetFirmwareRevision, 153
 - GetGlobalOffset, 154
 - GetModel, 154
 - GetOUI, 154
 - GetPol, 155
 - GetRCAdjust, 155
 - GetResolution, 155
 - GetSerialNumber, 156
 - GetStatus, 156
 - GetTDCEncapsulation, 156
 - GetTestMode, 157
 - GetTriggerConfiguration, 157
 - GetWindowOffset, 157
 - GetWindowWidth, 157
 - HardwareReset, 157
 - nchannels, 167
 - pair_lead_res, 167
 - pair_width_res, 167
 - ReadAcquisitionMode, 157
 - ReadDetectionMode, 158
 - SetAcquisitionMode, 158
 - SetBLTEventNumberRegister, 158
 - SetChannelDeadTime, 159
 - SetContinuousStorage, 159
 - SetControl, 159
 - SetDLLClock, 160
 - SetDetectionMode, 160
 - SetETTT, 161
 - SetErrorMarks, 160
 - SetFIFOSize, 161
 - SetGlobalOffset, 161
 - SetLSBTrailEdge, 162
 - SetPairModeResolution, 162
 - SetPol, 162
 - SetRCAdjust, 163
 - SetStatus, 163
 - SetTDCEncapsulation, 163
 - SetTestMode, 164
 - SetTriggerMatching, 164
 - SetTriggerTimeSubtraction, 164
 - SetVerboseLevel, 165
 - SetWindowOffset, 165
 - SetWindowWidth, 165
 - SoftwareClear, 166
 - SoftwareReset, 166
 - TDCV1x90, 148
 - WaitMicro, 166
- VME::TDCV1x90Control, 167
 - ~TDCV1x90Control, 168
 - Dump, 168
 - fWord, 173
 - GetAlign64, 169
 - GetBusError, 169
 - GetCompensation, 169
 - GetETTT, 170
 - GetEmptyEvent, 170
 - GetEventFIFO, 170
 - GetMEBAccess, 170
 - GetSRAMCompensation, 170
 - GetSWTermination, 170
 - GetTermination, 170
 - GetTestFIFO, 170
 - GetValue, 170
 - SetAlign64, 170
 - SetBusError, 170
 - SetCompensation, 170
 - SetETTT, 171
 - SetEmptyEvent, 171
 - SetEventFIFO, 171
 - SetMEBAccess, 172
 - SetSRAMCompensation, 172
 - SetSWTermination, 172
 - SetTermination, 173
 - SetTestFIFO, 173
 - TDCV1x90Control, 168
- VME::TDCV1x90Opcodes, 18

AUTOLOAD_DEF_CONFI, 20
 AUTOLOAD_USER_CONF, 20
 CLEAR_KEEP_TOKEN, 20
 CONT_STOR, 20
 DEFAULT_SETUP_REG, 20
 DIS_ALL_CHANNEL, 20
 DIS_CHANNEL, 20
 DIS_ERROR_BYPASS, 20
 DIS_ERROR_MARK, 20
 DIS_HEAD_TRAILER, 20
 DIS_SUB_TRG, 20
 DISABLE_TEST_MODE, 20
 EN_ALL_CHANNEL, 20
 EN_CHANNEL, 20
 EN_ERROR_BYPASS, 20
 EN_ERROR_MARK, 20
 EN_HEAD_TRAILER, 20
 EN_SUB_TRG, 20
 ENABLE_TEST_MODE, 20
 LOAD_DEF_CONFIG, 20
 LOAD_USER_CONFIG, 20
 READ_ACQ_MOD, 20
 READ_ADJUST_CH, 20
 READ_DEAD_TIME, 20
 READ_DETECTION, 20
 READ_DLL_LOCK, 20
 READ_EEPROM, 20
 READ_EN_PATTERN, 20
 READ_EN_PATTERN32, 21
 READ_ERROR_STATUS, 21
 READ_ERROR_TYPES, 21
 READ_EVENT_SIZE, 21
 READ_FIFO_SIZE, 21
 READ_GLOB_OFFS, 21
 READ_HEAD_TRAILER, 21
 READ_MICRO_REV, 21
 READ_RC_ADJ, 21
 READ_RES, 21
 READ_SETUP_REG, 21
 READ_SETUP_SCANPATH, 21
 READ_SPARE, 21
 READ_STATUS_STREAM, 21
 READ_TDC_ID, 21
 READ_TRG_CONF, 21
 RESET_DLL_PLL, 21
 REV_DATE_MICRO_FW, 21
 SAVE_RC_ADJ, 21
 SAVE_USER_CONFIG, 21
 SET_ADJUST_CH, 21
 SET_DEAD_TIME, 21
 SET_DETECTION, 21
 SET_DLL_CLOCK, 21
 SET_ERROR_TYPES, 21
 SET_EVENT_SIZE, 21
 SET_FIFO_SIZE, 21
 SET_GLOB_OFFS, 21
 SET_KEEP_TOKEN, 22
 SET_PAIR_RES, 22
 SET_RC_ADJ, 22
 SET_REJ_MARGIN, 22
 SET_SW_MARGIN, 22
 SET_TDC_TSET_OUTPUT, 22
 SET_TR_LEAD_LSB, 22
 SET_WIN_OFFS, 22
 SET_WIN_WIDTH, 22
 TRG_MATCH, 22
 UPDATE_SETUP_REG, 22
 UPDATE_SETUP_TDC, 22
 WRITE_EEPROM, 22
 WRITE_EN_PATTERN, 22
 WRITE_EN_PATTERN32, 22
 WRITE_SETUP_REG, 22
 WRITE_SPARE, 22
 VME::TDCV1x90Status, 174
 ~TDCV1x90Status, 175
 AlmostFull, 175
 BusError, 175
 DataReady, 175
 Dump, 175
 Error, 176
 fWord, 177
 Full, 177
 GetValue, 177
 HeadersEnabled, 177
 PairMode, 177
 Purged, 177
 R_100ps, 174
 R_200ps, 174
 R_25ps, 174
 R_800ps, 174
 Resolution, 177
 TDCResolution, 174
 TDCV1x90Status, 175
 TerminationOn, 177
 TriggerLost, 177
 TriggerMatching, 177
 VME::TDCV1x90TriggerConfig, 177
 ~TDCV1x90TriggerConfig, 178
 Dump, 178
 fWords, 179
 GetExtraSearchWindowWidth, 178
 GetRejectMargin, 178
 GetWindowOffset, 178
 GetWindowWidth, 178
 HasTriggerTimeSubtraction, 178
 SetWord, 179
 TDCV1x90TriggerConfig, 178
 VME::trailead_t, 179
 ettt, 179
 event_count, 179
 leading, 179
 total_hits, 179
 trailing, 179
 VMEReader, 179
 ~VMEReader, 183
 Abort, 183

- AddCFD, [183](#)
- AddFPGAUnit, [184](#)
- AddHVModule, [184](#)
- AddIOModule, [184](#)
- AddTDC, [185](#)
- BroadcastHVStatus, [185](#)
- BroadcastNewBurst, [185](#)
- BroadcastTriggerRate, [186](#)
- ContinuousStorage, [182](#)
- fBridge, [191](#)
- fCAENET, [191](#)
- fCFDCollection, [191](#)
- fFPGACollection, [191](#)
- fGlobalAcqMode, [192](#)
- fHV, [192](#)
- fIsPulserStarted, [192](#)
- fOnSocket, [192](#)
- fOutputFiles, [192](#)
- fSG, [192](#)
- fTDCCollection, [192](#)
- GetBurstNumber, [186](#)
- GetCFD, [186](#)
- GetCFDCollection, [186](#)
- GetFPGAUnit, [186](#)
- GetFPGAUnitCollection, [186](#)
- GetGlobalAcquisitionMode, [186](#)
- GetHVModule, [186](#)
- GetIOModule, [186](#)
- GetNumCFD, [187](#)
- GetNumTDC, [187](#)
- GetOutputFile, [187](#)
- GetRunNumber, [187](#)
- GetTDC, [187](#)
- GetTDCCollection, [187](#)
- GlobalAcqMode, [182](#)
- LogHVValues, [187](#)
- NewBurst, [187](#)
- NewRun, [188](#)
- OutputFiles, [182](#)
- ReadXML, [188](#), [189](#)
- SendClear, [190](#)
- SendOutputFile, [190](#)
- SendPulse, [190](#)
- SetOutputFile, [190](#)
- StartPulser, [191](#)
- StopPulser, [191](#)
- TriggerMatching, [182](#)
- TriggerStart, [182](#)
- UseSocket, [191](#)
- VMEReader, [182](#)
- Valid
 - VME::CAENETControllerV288Status, [34](#)
- Vmax
 - NIM::HVModuleN470Values, [91](#)
- Vmon
 - NIM::HVModuleN470ChannelValues, [90](#)
 - NIM::HVModuleN470Values, [91](#)
- Vsel
 - NIM::HVModuleN470Values::ChannelStatus, [42](#)
- WEBSOCKET_CLIENT
 - Socket communication objects, [9](#)
- WIN_OFFSET
 - VME, [18](#)
- WRITE_EEPROM
 - VME::TDCV1x90Opcodes, [22](#)
- WRITE_EN_PATTERN
 - VME::TDCV1x90Opcodes, [22](#)
- WRITE_EN_PATTERN32
 - VME::TDCV1x90Opcodes, [22](#)
- WRITE_OK
 - VME, [17](#)
- WRITE_SETUP_REG
 - VME::TDCV1x90Opcodes, [22](#)
- WRITE_SPARE
 - VME::TDCV1x90Opcodes, [22](#)
- WaitForResponse
 - VME::CAENETControllerV288, [33](#)
- WaitIRQ
 - VME::BridgeVx718, [27](#)
- WaitMicro
 - VME::TDCV1x90, [166](#)
- WriteRegister
 - NIM::HVModuleN470, [87](#)
 - VME::GenericBoard, [82](#)
- x
 - DQM::GastofCanvas::Coord, [48](#)
 - DQM::QuarticCanvas::Coord, [49](#)
- y
 - DQM::GastofCanvas::Coord, [49](#)
 - DQM::QuarticCanvas::Coord, [49](#)