

AC784xx_DFP PWM

5.1.0

Generated by Doxygen 1.8.13

Contents

| | | |
|----------|--|----------|
| 1 | Class Index | 1 |
| 1.1 | Class List | 1 |
| 2 | File Index | 2 |
| 2.1 | File List | 2 |
| 3 | Class Documentation | 3 |
| 3.1 | Pwm_Hal_CombineChnPairCfg Struct Reference | 3 |
| 3.1.1 | Detailed Description | 3 |
| 3.1.2 | Member Data Documentation | 4 |
| 3.1.2.1 | Ch1stActivePolarity | 4 |
| 3.1.2.2 | Ch1stDither | 4 |
| 3.1.2.3 | Ch1stInitLevel | 4 |
| 3.1.2.4 | Ch1stMatchDir | 4 |
| 3.1.2.5 | Ch1stValue | 4 |
| 3.1.2.6 | Ch2ndActivePolarity | 4 |
| 3.1.2.7 | Ch2ndDither | 5 |
| 3.1.2.8 | Ch2ndInitLevel | 5 |
| 3.1.2.9 | Ch2ndMatchDir | 5 |
| 3.1.2.10 | Ch2ndValue | 5 |
| 3.1.2.11 | DeadTimeValue | 5 |
| 3.1.2.12 | EnableCh1stEventDmaReq | 5 |
| 3.1.2.13 | EnableCh1stInterrupt | 6 |
| 3.1.2.14 | EnableCh1stMatchTrigger | 6 |
| 3.1.2.15 | EnableCh2ndEventDmaReq | 6 |

| | | |
|----------|--|----|
| 3.1.2.16 | EnableCh2ndInterrupt | 6 |
| 3.1.2.17 | EnableCh2ndMatchTrigger | 6 |
| 3.1.2.18 | EnableChComplementation | 6 |
| 3.1.2.19 | EnableDeadTime | 7 |
| 3.1.2.20 | EnableSymmetric | 7 |
| 3.1.2.21 | LevelMode | 7 |
| 3.1.2.22 | TimePsc | 7 |
| 3.2 | Pwm_Hal_CommonCfg Struct Reference | 7 |
| 3.2.1 | Detailed Description | 8 |
| 3.2.2 | Member Data Documentation | 8 |
| 3.2.2.1 | ChannelCallback | 8 |
| 3.2.2.2 | ClockSource | 8 |
| 3.2.2.3 | DmaTransLen | 8 |
| 3.2.2.4 | EnableDmaTransLen | 8 |
| 3.2.2.5 | EnableOverflowDmaReq | 8 |
| 3.2.2.6 | EnableOverflowEvent | 9 |
| 3.2.2.7 | EnableOverflowInterrupt | 9 |
| 3.2.2.8 | EnableUnderflowEvent | 9 |
| 3.2.2.9 | MaxCount | 9 |
| 3.2.2.10 | MinCount | 9 |
| 3.2.2.11 | OverflowCallback | 9 |
| 3.2.2.12 | OverflowFreq | 10 |
| 3.2.2.13 | PeriodDither | 10 |
| 3.2.2.14 | Prescaler | 10 |
| 3.3 | Pwm_Hal_CompareModeChnCfg Struct Reference | 10 |
| 3.3.1 | Detailed Description | 10 |
| 3.3.2 | Member Data Documentation | 11 |
| 3.3.2.1 | Action | 11 |
| 3.3.2.2 | ActivePolarity | 11 |
| 3.3.2.3 | ChnValue | 11 |
| 3.3.2.4 | EnableChnEventDmaReq | 11 |

| | | |
|---------|--|----|
| 3.3.2.5 | EnableInterrupt | 11 |
| 3.3.2.6 | EnableMatchTrigger | 11 |
| 3.3.2.7 | InitLevel | 12 |
| 3.4 | Pwm_Hal_FaultCfg Struct Reference | 12 |
| 3.4.1 | Detailed Description | 12 |
| 3.4.2 | Member Data Documentation | 12 |
| 3.4.2.1 | EnableChannelOutputCtrl | 12 |
| 3.4.2.2 | EnableHiz | 13 |
| 3.4.2.3 | EnableInterrupt | 13 |
| 3.4.2.4 | FaultCallback | 13 |
| 3.4.2.5 | FaultCtrlMode | 13 |
| 3.4.2.6 | FaultPinCfg | 13 |
| 3.4.2.7 | FilterPsc | 14 |
| 3.4.2.8 | FilterValue | 14 |
| 3.5 | Pwm_Hal_FaultPinCfg Struct Reference | 14 |
| 3.5.1 | Detailed Description | 14 |
| 3.5.2 | Member Data Documentation | 14 |
| 3.5.2.1 | EnableFaultFilter | 15 |
| 3.5.2.2 | EnableFaultInput | 15 |
| 3.5.2.3 | FaultPolarity | 15 |
| 3.6 | Pwm_Hal_IndependentChnCfg Struct Reference | 15 |
| 3.6.1 | Detailed Description | 16 |
| 3.6.2 | Member Data Documentation | 16 |
| 3.6.2.1 | ActivePolarity | 16 |
| 3.6.2.2 | ChnDither | 16 |
| 3.6.2.3 | ChnValue | 16 |
| 3.6.2.4 | EnableChnEventDmaReq | 16 |
| 3.6.2.5 | EnableInterrupt | 16 |
| 3.6.2.6 | EnableMatchTrigger | 17 |
| 3.6.2.7 | InitLevel | 17 |
| 3.6.2.8 | LevelMode | 17 |

| | | |
|----------|--|----|
| 3.7 | Pwm_Hal_InputCfg Struct Reference | 17 |
| 3.7.1 | Detailed Description | 17 |
| 3.7.2 | Member Data Documentation | 17 |
| 3.7.2.1 | ChannelCfg | 18 |
| 3.7.2.2 | ChannelNum | 18 |
| 3.7.2.3 | CommonCfg | 18 |
| 3.8 | Pwm_Hal_InputChannelCfg Struct Reference | 18 |
| 3.8.1 | Detailed Description | 18 |
| 3.8.2 | Member Data Documentation | 19 |
| 3.8.2.1 | CaptureEdge | 19 |
| 3.8.2.2 | Channel | 19 |
| 3.8.2.3 | ChannelMode | 19 |
| 3.8.2.4 | DualInputContinuousMode | 19 |
| 3.8.2.5 | DualInputMeasureType | 19 |
| 3.8.2.6 | EnableChnEventDmaReq | 19 |
| 3.8.2.7 | EnableCounterReset | 20 |
| 3.8.2.8 | EnableInterrupt | 20 |
| 3.8.2.9 | EnablePulseWidthMeasure | 20 |
| 3.8.2.10 | EventPsc | 20 |
| 3.8.2.11 | FilterValue | 20 |
| 3.9 | Pwm_Hal_InputCommonCfg Struct Reference | 20 |
| 3.9.1 | Detailed Description | 21 |
| 3.9.2 | Member Data Documentation | 21 |
| 3.9.2.1 | BaseCfg | 21 |
| 3.9.2.2 | EnableHall | 21 |
| 3.9.2.3 | FilterPsc | 21 |
| 3.10 | Pwm_Hal_OutputCfg Struct Reference | 21 |
| 3.10.1 | Detailed Description | 22 |
| 3.10.2 | Member Data Documentation | 22 |
| 3.10.2.1 | ChannelCfg | 22 |
| 3.10.2.2 | ChannelNum | 22 |

| | |
|--|----|
| 3.10.2.3 CommonCfg | 22 |
| 3.11 Pwm_Hal_OutputChannelCfg Struct Reference | 22 |
| 3.11.1 Detailed Description | 23 |
| 3.11.2 Member Data Documentation | 23 |
| 3.11.2.1 Channel | 23 |
| 3.11.2.2 ChannelMode | 23 |
| 3.11.2.3 ChnPairCfg | 23 |
| 3.11.2.4 CompareChnCfg | 23 |
| 3.11.2.5 IndependentChnCfg | 24 |
| 3.12 Pwm_Hal_OutputCommonCfg Struct Reference | 24 |
| 3.12.1 Detailed Description | 24 |
| 3.12.2 Member Data Documentation | 24 |
| 3.12.2.1 BaseCfg | 24 |
| 3.12.2.2 CombineCenterDutyMode | 24 |
| 3.12.2.3 CountMode | 25 |
| 3.12.2.4 EnableInitTrigger | 25 |
| 3.12.2.5 EnableMaxTrigger | 25 |
| 3.12.2.6 InitOutput | 25 |
| 3.12.2.7 TriggerRatio | 25 |
| 3.13 Pwm_Hal_QuadDecoderCfg Struct Reference | 25 |
| 3.13.1 Detailed Description | 26 |
| 3.13.2 Member Data Documentation | 26 |
| 3.13.2.1 BaseCfg | 26 |
| 3.13.2.2 EnablePhaseZReset | 26 |
| 3.13.2.3 EnableQuad | 26 |
| 3.13.2.4 FilterPsc | 27 |
| 3.13.2.5 Mode | 27 |
| 3.13.2.6 PhaseAConfig | 27 |
| 3.13.2.7 PhaseBConfig | 27 |
| 3.13.2.8 PhaseZConfig | 27 |
| 3.14 Pwm_Hal_QuadPhaseCfg Struct Reference | 28 |

| | |
|---|----|
| 3.14.1 Detailed Description | 28 |
| 3.14.2 Member Data Documentation | 28 |
| 3.14.2.1 FilterValue | 28 |
| 3.14.2.2 Polarity | 28 |
| 3.15 Pwm_Hal_SyncCfg Struct Reference | 28 |
| 3.15.1 Detailed Description | 29 |
| 3.15.2 Member Data Documentation | 29 |
| 3.15.2.1 DisableHwTriggerAfterTriggered | 29 |
| 3.15.2.2 EnableCounterInitSync | 29 |
| 3.15.2.3 EnableDualChannelInvertSync | 30 |
| 3.15.2.4 EnableHwSync0 | 30 |
| 3.15.2.5 EnableHwSync1 | 30 |
| 3.15.2.6 EnableHwSync2 | 30 |
| 3.15.2.7 EnableMaxLoadingPoint | 30 |
| 3.15.2.8 EnableMinLoadingPoint | 31 |
| 3.15.2.9 EnableOutputMaskSync | 31 |
| 3.15.2.10 EnablePairedChnValSync | 31 |
| 3.15.2.11 EnablePolaritySync | 31 |
| 3.15.2.12 EnableSwOutputCtrlSync | 31 |
| 3.15.2.13 EnableSync | 32 |
| 3.15.2.14 EnableSyncBypass | 32 |
| 3.15.2.15 TriggerType | 32 |

| | | |
|----------|---|-----------|
| 4 | File Documentation | 33 |
| 4.1 | AC784xx_API_Reference_Manual_PWM.pdf File Reference | 33 |
| 4.2 | AC784xx_Pwm_Reg.h File Reference | 33 |
| 4.2.1 | Detailed Description | 36 |
| 4.2.2 | Function Documentation | 36 |
| 4.2.2.1 | Pwm_Reg_ChooseBufferOrRegisterToRead() | 36 |
| 4.2.2.2 | Pwm_Reg_ClearChannelEventFlag() | 36 |
| 4.2.2.3 | Pwm_Reg_ClearChannelInterruptFlag() | 37 |
| 4.2.2.4 | Pwm_Reg_ClearChannelTriggerFlag() | 37 |
| 4.2.2.5 | Pwm_Reg_ClearFaultFlag() | 37 |
| 4.2.2.6 | Pwm_Reg_ClearFaultPinFlag() | 37 |
| 4.2.2.7 | Pwm_Reg_ClearOverflowFlag() | 37 |
| 4.2.2.8 | Pwm_Reg_ClearQuadPhaseZFlag() | 37 |
| 4.2.2.9 | Pwm_Reg_DisableHwTriggerSyncAfterTriggered() | 38 |
| 4.2.2.10 | Pwm_Reg_EnableChannelCombine() | 38 |
| 4.2.2.11 | Pwm_Reg_EnableChannelDmaReq() | 38 |
| 4.2.2.12 | Pwm_Reg_EnableChannelEventReset() | 38 |
| 4.2.2.13 | Pwm_Reg_EnableChannelHizOutput() | 38 |
| 4.2.2.14 | Pwm_Reg_EnableChannelInterrupt() | 39 |
| 4.2.2.15 | Pwm_Reg_EnableChannelMatchTrigger() | 39 |
| 4.2.2.16 | Pwm_Reg_EnableChannelOutputMask() | 39 |
| 4.2.2.17 | Pwm_Reg_EnableChannelSoftControl() | 39 |
| 4.2.2.18 | Pwm_Reg_EnableCHOSWCERSync() | 39 |
| 4.2.2.19 | Pwm_Reg_EnableCNTINSync() | 40 |
| 4.2.2.20 | Pwm_Reg_EnableDmaTransfer() | 40 |
| 4.2.2.21 | Pwm_Reg_EnableFaultHizOutput() | 40 |
| 4.2.2.22 | Pwm_Reg_EnableFaultInputFilter() | 40 |
| 4.2.2.23 | Pwm_Reg_EnableFaultInterrupt() | 40 |
| 4.2.2.24 | Pwm_Reg_EnableFaultPinInput() | 41 |
| 4.2.2.25 | Pwm_Reg_EnableGlobalTimeBase() | 41 |
| 4.2.2.26 | Pwm_Reg_EnableGlobalTimeBaseOutput() | 41 |

| | | |
|----------|---|----|
| 4.2.2.27 | Pwm_Reg_EnableHall() | 41 |
| 4.2.2.28 | Pwm_Reg_EnableInitTrigger() | 41 |
| 4.2.2.29 | Pwm_Reg_EnableINVCRSync() | 42 |
| 4.2.2.30 | Pwm_Reg_EnableMaxLoadingPointSync() | 42 |
| 4.2.2.31 | Pwm_Reg_EnableMaxTrigger() | 42 |
| 4.2.2.32 | Pwm_Reg_EnableMinLoadingPointSync() | 42 |
| 4.2.2.33 | Pwm_Reg_EnableOutputInit() | 42 |
| 4.2.2.34 | Pwm_Reg_EnableOutputMaskSync() | 43 |
| 4.2.2.35 | Pwm_Reg_EnableOverflowDmaReq() | 43 |
| 4.2.2.36 | Pwm_Reg_EnableOverflowEvent() | 43 |
| 4.2.2.37 | Pwm_Reg_EnableOverflowInterrupt() | 43 |
| 4.2.2.38 | Pwm_Reg_EnablePairChComplement() | 43 |
| 4.2.2.39 | Pwm_Reg_EnablePairChDeadtime() | 44 |
| 4.2.2.40 | Pwm_Reg_EnablePairChDualEdgeCapture() | 44 |
| 4.2.2.41 | Pwm_Reg_EnablePairChFaultCtrl() | 44 |
| 4.2.2.42 | Pwm_Reg_EnablePairChInvert() | 44 |
| 4.2.2.43 | Pwm_Reg_EnablePairChSymmetric() | 44 |
| 4.2.2.44 | Pwm_Reg_EnablePairChValueSync() | 45 |
| 4.2.2.45 | Pwm_Reg_EnablePairPulseWidthMeasurement() | 45 |
| 4.2.2.46 | Pwm_Reg_EnablePOLSync() | 45 |
| 4.2.2.47 | Pwm_Reg_EnableQuadDecoder() | 45 |
| 4.2.2.48 | Pwm_Reg_EnableQuadPhaseZResetCnt() | 45 |
| 4.2.2.49 | Pwm_Reg_EnableSync() | 46 |
| 4.2.2.50 | Pwm_Reg_EnableSyncBYP() | 46 |
| 4.2.2.51 | Pwm_Reg_EnableSyncHwTriggerSrc() | 46 |
| 4.2.2.52 | Pwm_Reg_EnableUnderflowEvent() | 46 |
| 4.2.2.53 | Pwm_Reg_EnableWriteProtection() | 46 |
| 4.2.2.54 | Pwm_Reg_GetAllChannelLevel() | 47 |
| 4.2.2.55 | Pwm_Reg_GetBase() | 47 |
| 4.2.2.56 | Pwm_Reg_GetChannelEventFlag() | 47 |
| 4.2.2.57 | Pwm_Reg_GetChannelInterruptFlag() | 47 |

| | | |
|----------|--|----|
| 4.2.2.58 | Pwm_Reg_GetChannelTriggerFlag() | 47 |
| 4.2.2.59 | Pwm_Reg_GetChannelValue() | 47 |
| 4.2.2.60 | Pwm_Reg_GetCountValue() | 48 |
| 4.2.2.61 | Pwm_Reg_GetFaultFlag() | 48 |
| 4.2.2.62 | Pwm_Reg_GetFaultInputStatus() | 48 |
| 4.2.2.63 | Pwm_Reg_GetFaultPinFlag() | 48 |
| 4.2.2.64 | Pwm_Reg_GetHallStatus() | 48 |
| 4.2.2.65 | Pwm_Reg_GetInitCountValue() | 48 |
| 4.2.2.66 | Pwm_Reg_GetMaxCountValue() | 49 |
| 4.2.2.67 | Pwm_Reg_GetOutputMask() | 49 |
| 4.2.2.68 | Pwm_Reg_GetOverflowDir() | 49 |
| 4.2.2.69 | Pwm_Reg_GetOverflowFlag() | 49 |
| 4.2.2.70 | Pwm_Reg_GetQuadCountDir() | 49 |
| 4.2.2.71 | Pwm_Reg_GetQuadOverflowDir() | 49 |
| 4.2.2.72 | Pwm_Reg_GetQuadPhaseZFlag() | 50 |
| 4.2.2.73 | Pwm_Reg_IsPairChDualEdgeCaptureEnabled() | 50 |
| 4.2.2.74 | Pwm_Reg_ReadChannelSoftControlState() | 50 |
| 4.2.2.75 | Pwm_Reg_ResetCounter() | 50 |
| 4.2.2.76 | Pwm_Reg_SetCaptureEventPsc() | 50 |
| 4.2.2.77 | Pwm_Reg_SetChannelELSR() | 51 |
| 4.2.2.78 | Pwm_Reg_SetChannelInputFilterVal() | 51 |
| 4.2.2.79 | Pwm_Reg_SetChannelMatchDir() | 51 |
| 4.2.2.80 | Pwm_Reg_SetChannelMatchDither() | 51 |
| 4.2.2.81 | Pwm_Reg_SetChannelMatchDitherReg() | 51 |
| 4.2.2.82 | Pwm_Reg_SetChannelMSR() | 52 |
| 4.2.2.83 | Pwm_Reg_SetChannelOutputInitLevel() | 52 |
| 4.2.2.84 | Pwm_Reg_SetChannelPolarity() | 52 |
| 4.2.2.85 | Pwm_Reg_SetChannelSoftControlLevel() | 52 |
| 4.2.2.86 | Pwm_Reg_SetChannelValue() | 52 |
| 4.2.2.87 | Pwm_Reg_SetCHOSWCRSyncTriggerMode() | 53 |
| 4.2.2.88 | Pwm_Reg_SetClockPsc() | 53 |

| | | |
|-----------|--|----|
| 4.2.2.89 | Pwm_Reg_SetClockSource() | 53 |
| 4.2.2.90 | Pwm_Reg_SetCntOverflowFreq() | 53 |
| 4.2.2.91 | Pwm_Reg_SetCNTSyncTriggerMode() | 53 |
| 4.2.2.92 | Pwm_Reg_SetCombineCenterDutyType() | 54 |
| 4.2.2.93 | Pwm_Reg_SetCountMode() | 54 |
| 4.2.2.94 | Pwm_Reg_SetDebugMode() | 54 |
| 4.2.2.95 | Pwm_Reg_SetDmaTransferLen() | 54 |
| 4.2.2.96 | Pwm_Reg_SetFaultInputFilterValue() | 54 |
| 4.2.2.97 | Pwm_Reg_SetFaultInputPolarity() | 55 |
| 4.2.2.98 | Pwm_Reg_SetFaultMode() | 55 |
| 4.2.2.99 | Pwm_Reg_SetInitCountValue() | 55 |
| 4.2.2.100 | Pwm_Reg_SetInputFilterPsc() | 55 |
| 4.2.2.101 | Pwm_Reg_SetINVCRSyncTriggerMode() | 55 |
| 4.2.2.102 | Pwm_Reg_SetMaxCountValue() | 56 |
| 4.2.2.103 | Pwm_Reg_SetMCVRSyncMode() | 56 |
| 4.2.2.104 | Pwm_Reg_SetOMCRSyncTriggerMode() | 56 |
| 4.2.2.105 | Pwm_Reg_SetOutputMask() | 56 |
| 4.2.2.106 | Pwm_Reg_SetPairChDeadtime() | 56 |
| 4.2.2.107 | Pwm_Reg_SetPairChDualEdgeCapture() | 57 |
| 4.2.2.108 | Pwm_Reg_SetPeriodDither() | 57 |
| 4.2.2.109 | Pwm_Reg_SetPOLTriggerMode() | 57 |
| 4.2.2.110 | Pwm_Reg_SetQuadEncodeMode() | 57 |
| 4.2.2.111 | Pwm_Reg_SetQuadPhaseAPolarity() | 57 |
| 4.2.2.112 | Pwm_Reg_SetQuadPhaseBPolarity() | 58 |
| 4.2.2.113 | Pwm_Reg_SetQuadPhaseZPolarity() | 58 |
| 4.2.2.114 | Pwm_Reg_SetSoftControlEnableStatesMask() | 58 |
| 4.2.2.115 | Pwm_Reg_SetSoftControlLevelMask() | 58 |
| 4.2.2.116 | Pwm_Reg_SetSyncMode() | 58 |
| 4.2.2.117 | Pwm_Reg_SetSyncType() | 59 |
| 4.2.2.118 | Pwm_Reg_SetTriggerRatio() | 59 |
| 4.2.2.119 | Pwm_Reg_TrigSoftwareSync() | 59 |

| | | |
|----------|-------------------------------------|----|
| 4.3 | Pwm_Hal.c File Reference | 59 |
| 4.3.1 | Function Documentation | 63 |
| 4.3.1.1 | PWM0_Channel_IRQHandler() | 63 |
| 4.3.1.2 | PWM0_Fault_IRQHandler() | 64 |
| 4.3.1.3 | PWM0_Overflow_IRQHandler() | 64 |
| 4.3.1.4 | PWM1_Channel_IRQHandler() | 64 |
| 4.3.1.5 | PWM1_Fault_IRQHandler() | 65 |
| 4.3.1.6 | PWM1_Overflow_IRQHandler() | 65 |
| 4.3.1.7 | PWM2_Channel_IRQHandler() | 65 |
| 4.3.1.8 | PWM2_Fault_IRQHandler() | 66 |
| 4.3.1.9 | PWM2_Overflow_IRQHandler() | 66 |
| 4.3.1.10 | PWM3_Channel_IRQHandler() | 66 |
| 4.3.1.11 | PWM3_Fault_IRQHandler() | 67 |
| 4.3.1.12 | PWM3_Overflow_IRQHandler() | 67 |
| 4.3.1.13 | PWM4_Channel_IRQHandler() | 67 |
| 4.3.1.14 | PWM4_Fault_IRQHandler() | 68 |
| 4.3.1.15 | PWM4_Overflow_IRQHandler() | 68 |
| 4.3.1.16 | PWM5_Channel_IRQHandler() | 68 |
| 4.3.1.17 | PWM5_Fault_IRQHandler() | 69 |
| 4.3.1.18 | PWM5_Overflow_IRQHandler() | 69 |
| 4.3.1.19 | Pwm_Hal_ClearChannelInterruptFlag() | 69 |
| 4.3.1.20 | Pwm_Hal_ClearDetectEventFlag() | 70 |
| 4.3.1.21 | Pwm_Hal_ClearFaultFlag() | 70 |
| 4.3.1.22 | Pwm_Hal_ClearFaultPinFlag() | 71 |
| 4.3.1.23 | Pwm_Hal_ClearOverflowFlag() | 71 |
| 4.3.1.24 | Pwm_Hal_ClearQuadPhaseZFlag() | 72 |
| 4.3.1.25 | Pwm_Hal_DeInit() | 72 |
| 4.3.1.26 | Pwm_Hal_EnableChannelDmaRequest() | 73 |
| 4.3.1.27 | Pwm_Hal_EnableChannelEventReset() | 73 |
| 4.3.1.28 | Pwm_Hal_EnableChannelHizOutput() | 74 |
| 4.3.1.29 | Pwm_Hal_EnableChannelInterrupt() | 74 |

| | | |
|----------|---|----|
| 4.3.1.30 | Pwm_Hal_EnableChannelMatchTrigger() | 75 |
| 4.3.1.31 | Pwm_Hal_EnableChannelOutputMask() | 75 |
| 4.3.1.32 | Pwm_Hal_EnableChannelPairInvert() | 76 |
| 4.3.1.33 | Pwm_Hal_EnableChannelPairSymmetric() | 76 |
| 4.3.1.34 | Pwm_Hal_EnableChannelSoftControl() | 77 |
| 4.3.1.35 | Pwm_Hal_EnableCombineChannelSoftControl() | 77 |
| 4.3.1.36 | Pwm_Hal_EnableFaultInputFilter() | 78 |
| 4.3.1.37 | Pwm_Hal_EnableFaultPinInput() | 78 |
| 4.3.1.38 | Pwm_Hal_EnableGlobalTimeBase() | 79 |
| 4.3.1.39 | Pwm_Hal_EnableGlobalTimeBaseOutput() | 79 |
| 4.3.1.40 | Pwm_Hal_EnableInitTrigger() | 80 |
| 4.3.1.41 | Pwm_Hal_EnableMaxTrigger() | 80 |
| 4.3.1.42 | Pwm_Hal_EnableOverflowEvent() | 81 |
| 4.3.1.43 | Pwm_Hal_EnableOverflowInterrupt() | 81 |
| 4.3.1.44 | Pwm_Hal_EnableSyncBypass() | 82 |
| 4.3.1.45 | Pwm_Hal_EnableUnderflowEvent() | 82 |
| 4.3.1.46 | Pwm_Hal_EnableWriteProtection() | 83 |
| 4.3.1.47 | Pwm_Hal_GetAllChannelLevel() | 83 |
| 4.3.1.48 | Pwm_Hal_GetChannelInterruptFlag() | 83 |
| 4.3.1.49 | Pwm_Hal_GetChannelValue() | 84 |
| 4.3.1.50 | Pwm_Hal_GetCountValue() | 84 |
| 4.3.1.51 | Pwm_Hal_GetDetectEventFlag() | 85 |
| 4.3.1.52 | Pwm_Hal_GetFaultFlag() | 85 |
| 4.3.1.53 | Pwm_Hal_GetFaultPinFlag() | 86 |
| 4.3.1.54 | Pwm_Hal_GetHallStatus() | 86 |
| 4.3.1.55 | Pwm_Hal_GetInitCountValue() | 87 |
| 4.3.1.56 | Pwm_Hal_GetMaxCountValue() | 87 |
| 4.3.1.57 | Pwm_Hal_GetOutputMask() | 88 |
| 4.3.1.58 | Pwm_Hal_GetOverflowDir() | 88 |
| 4.3.1.59 | Pwm_Hal_GetOverflowFlag() | 88 |
| 4.3.1.60 | Pwm_Hal_GetQuadCountingDir() | 89 |

| | | |
|----------|--------------------------------------|-----|
| 4.3.1.61 | Pwm_Hal_GetQuadOverflowDir() | 89 |
| 4.3.1.62 | Pwm_Hal_GetQuadPhaseZFlag() | 90 |
| 4.3.1.63 | Pwm_Hal_InitFaultControl() | 90 |
| 4.3.1.64 | Pwm_Hal_InitInputMode() | 91 |
| 4.3.1.65 | Pwm_Hal_InitOutputMode() | 91 |
| 4.3.1.66 | Pwm_Hal_InitQuadDecoderMode() | 92 |
| 4.3.1.67 | Pwm_Hal_InitSyncConfigSet() | 92 |
| 4.3.1.68 | Pwm_Hal_ResetCounter() | 93 |
| 4.3.1.69 | Pwm_Hal_SetChannelCaptureEdge() | 93 |
| 4.3.1.70 | Pwm_Hal_SetChannelCompareAction() | 94 |
| 4.3.1.71 | Pwm_Hal_SetChannelInputFilterVal() | 94 |
| 4.3.1.72 | Pwm_Hal_SetChannelMatchDither() | 95 |
| 4.3.1.73 | Pwm_Hal_SetChannelPolarity() | 95 |
| 4.3.1.74 | Pwm_Hal_SetChannelSoftControlLevel() | 96 |
| 4.3.1.75 | Pwm_Hal_SetChannelValue() | 96 |
| 4.3.1.76 | Pwm_Hal_SetClockSource() | 97 |
| 4.3.1.77 | Pwm_Hal_SetDeadtime() | 97 |
| 4.3.1.78 | Pwm_Hal_SetDebugMode() | 98 |
| 4.3.1.79 | Pwm_Hal_SetFaultInputPolarity() | 99 |
| 4.3.1.80 | Pwm_Hal_SetInitCountValue() | 99 |
| 4.3.1.81 | Pwm_Hal_SetInputFilterPsc() | 100 |
| 4.3.1.82 | Pwm_Hal_SetMaxCountValue() | 100 |
| 4.3.1.83 | Pwm_Hal_SetOutputMask() | 101 |
| 4.3.1.84 | Pwm_Hal_SetPeriodDither() | 101 |
| 4.3.1.85 | Pwm_Hal_SetSoftControlEnableStates() | 102 |
| 4.3.1.86 | Pwm_Hal_SetSoftControlLevels() | 102 |
| 4.3.1.87 | Pwm_Hal_SetTriggerRatio() | 103 |
| 4.3.1.88 | Pwm_Hal_TrigSoftwareSync() | 103 |
| 4.4 | Pwm_Hal.h File Reference | 103 |
| 4.4.1 | Function Documentation | 107 |
| 4.4.1.1 | Pwm_Hal_ClearChannelInterruptFlag() | 107 |

| | | |
|----------|---|-----|
| 4.4.1.2 | Pwm_Hal_ClearDetectEventFlag() | 107 |
| 4.4.1.3 | Pwm_Hal_ClearFaultFlag() | 108 |
| 4.4.1.4 | Pwm_Hal_ClearFaultPinFlag() | 108 |
| 4.4.1.5 | Pwm_Hal_ClearOverflowFlag() | 109 |
| 4.4.1.6 | Pwm_Hal_ClearQuadPhaseZFlag() | 109 |
| 4.4.1.7 | Pwm_Hal_DeInit() | 110 |
| 4.4.1.8 | Pwm_Hal_EnableChannelDmaRequest() | 110 |
| 4.4.1.9 | Pwm_Hal_EnableChannelEventReset() | 111 |
| 4.4.1.10 | Pwm_Hal_EnableChannelHizOutput() | 111 |
| 4.4.1.11 | Pwm_Hal_EnableChannelInterrupt() | 112 |
| 4.4.1.12 | Pwm_Hal_EnableChannelMatchTrigger() | 112 |
| 4.4.1.13 | Pwm_Hal_EnableChannelOutputMask() | 113 |
| 4.4.1.14 | Pwm_Hal_EnableChannelPairInvert() | 113 |
| 4.4.1.15 | Pwm_Hal_EnableChannelPairSymmetric() | 114 |
| 4.4.1.16 | Pwm_Hal_EnableChannelSoftControl() | 114 |
| 4.4.1.17 | Pwm_Hal_EnableCombineChannelSoftControl() | 115 |
| 4.4.1.18 | Pwm_Hal_EnableFaultInputFilter() | 115 |
| 4.4.1.19 | Pwm_Hal_EnableFaultPinInput() | 116 |
| 4.4.1.20 | Pwm_Hal_EnableGlobalTimeBase() | 116 |
| 4.4.1.21 | Pwm_Hal_EnableGlobalTimeBaseOutput() | 117 |
| 4.4.1.22 | Pwm_Hal_EnableInitTrigger() | 117 |
| 4.4.1.23 | Pwm_Hal_EnableMaxTrigger() | 118 |
| 4.4.1.24 | Pwm_Hal_EnableOverflowEvent() | 118 |
| 4.4.1.25 | Pwm_Hal_EnableOverflowInterrupt() | 119 |
| 4.4.1.26 | Pwm_Hal_EnableSyncBypass() | 119 |
| 4.4.1.27 | Pwm_Hal_EnableUnderflowEvent() | 120 |
| 4.4.1.28 | Pwm_Hal_EnableWriteProtection() | 120 |
| 4.4.1.29 | Pwm_Hal_GetAllChannelLevel() | 121 |
| 4.4.1.30 | Pwm_Hal_GetChannelInterruptFlag() | 121 |
| 4.4.1.31 | Pwm_Hal_GetChannelValue() | 121 |
| 4.4.1.32 | Pwm_Hal_GetCountValue() | 122 |

| | | |
|----------|--------------------------------------|-----|
| 4.4.1.33 | Pwm_Hal_GetDetectEventFlag() | 122 |
| 4.4.1.34 | Pwm_Hal_GetFaultFlag() | 123 |
| 4.4.1.35 | Pwm_Hal_GetFaultPinFlag() | 123 |
| 4.4.1.36 | Pwm_Hal_GetHallStatus() | 124 |
| 4.4.1.37 | Pwm_Hal_GetInitCountValue() | 124 |
| 4.4.1.38 | Pwm_Hal_GetMaxCountValue() | 125 |
| 4.4.1.39 | Pwm_Hal_GetOutputMask() | 125 |
| 4.4.1.40 | Pwm_Hal_GetOverflowDir() | 125 |
| 4.4.1.41 | Pwm_Hal_GetOverflowFlag() | 126 |
| 4.4.1.42 | Pwm_Hal_GetQuadCountingDir() | 126 |
| 4.4.1.43 | Pwm_Hal_GetQuadOverflowDir() | 127 |
| 4.4.1.44 | Pwm_Hal_GetQuadPhaseZFlag() | 127 |
| 4.4.1.45 | Pwm_Hal_InitFaultControl() | 128 |
| 4.4.1.46 | Pwm_Hal_InitInputMode() | 128 |
| 4.4.1.47 | Pwm_Hal_InitOutputMode() | 129 |
| 4.4.1.48 | Pwm_Hal_InitQuadDecoderMode() | 129 |
| 4.4.1.49 | Pwm_Hal_InitSyncConfigSet() | 130 |
| 4.4.1.50 | Pwm_Hal_ResetCounter() | 130 |
| 4.4.1.51 | Pwm_Hal_SetChannelCaptureEdge() | 131 |
| 4.4.1.52 | Pwm_Hal_SetChannelCompareAction() | 131 |
| 4.4.1.53 | Pwm_Hal_SetChannelInputFilterVal() | 132 |
| 4.4.1.54 | Pwm_Hal_SetChannelMatchDither() | 132 |
| 4.4.1.55 | Pwm_Hal_SetChannelPolarity() | 133 |
| 4.4.1.56 | Pwm_Hal_SetChannelSoftControlLevel() | 134 |
| 4.4.1.57 | Pwm_Hal_SetChannelValue() | 134 |
| 4.4.1.58 | Pwm_Hal_SetClockSource() | 135 |
| 4.4.1.59 | Pwm_Hal_SetDeadtime() | 135 |
| 4.4.1.60 | Pwm_Hal_SetDebugMode() | 136 |
| 4.4.1.61 | Pwm_Hal_SetFaultInputPolarity() | 137 |
| 4.4.1.62 | Pwm_Hal_SetInitCountValue() | 137 |
| 4.4.1.63 | Pwm_Hal_SetInputFilterPsc() | 138 |

| | | |
|----------|--------------------------------------|-----|
| 4.4.1.64 | Pwm_Hal_SetMaxCountValue() | 138 |
| 4.4.1.65 | Pwm_Hal_SetOutputMask() | 139 |
| 4.4.1.66 | Pwm_Hal_SetPeriodDither() | 139 |
| 4.4.1.67 | Pwm_Hal_SetSoftControlEnableStates() | 140 |
| 4.4.1.68 | Pwm_Hal_SetSoftControlLevels() | 140 |
| 4.4.1.69 | Pwm_Hal_SetTriggerRatio() | 141 |
| 4.4.1.70 | Pwm_Hal_TrigSoftwareSync() | 141 |
| 4.5 | Pwm_Hal_Types.h File Reference | 142 |
| 4.5.1 | Typedef Documentation | 144 |
| 4.5.1.1 | Pwm_Hal_Callback | 144 |
| 4.5.2 | Enumeration Type Documentation | 144 |
| 4.5.2.1 | Pwm_Hal_ActivePolarityType | 144 |
| 4.5.2.2 | Pwm_Hal_ChannelMatchDirType | 145 |
| 4.5.2.3 | Pwm_Hal_ChannelPairType | 145 |
| 4.5.2.4 | Pwm_Hal_ChannelType | 145 |
| 4.5.2.5 | Pwm_Hal_ClockSourceType | 146 |
| 4.5.2.6 | Pwm_Hal_CombineCenterDutyModeType | 146 |
| 4.5.2.7 | Pwm_Hal_CompareActionType | 147 |
| 4.5.2.8 | Pwm_Hal_CountModeType | 147 |
| 4.5.2.9 | Pwm_Hal_DeadTimePscType | 147 |
| 4.5.2.10 | Pwm_Hal_DebugModeType | 148 |
| 4.5.2.11 | Pwm_Hal_DualInputContinuousModeType | 148 |
| 4.5.2.12 | Pwm_Hal_DualInputMeasureType | 148 |
| 4.5.2.13 | Pwm_Hal_EdgeType | 149 |
| 4.5.2.14 | Pwm_Hal_FaultCtrlModeType | 149 |
| 4.5.2.15 | Pwm_Hal_FaultInputIdType | 149 |
| 4.5.2.16 | Pwm_Hal_InputChnModeType | 150 |
| 4.5.2.17 | Pwm_Hal_InputEventPscType | 150 |
| 4.5.2.18 | Pwm_Hal_InputFilterPscType | 150 |
| 4.5.2.19 | Pwm_Hal_InstanceType | 151 |
| 4.5.2.20 | Pwm_Hal_OutputChnModeType | 151 |
| 4.5.2.21 | Pwm_Hal_OutputLevelModeType | 152 |
| 4.5.2.22 | Pwm_Hal_OutputLevelType | 152 |
| 4.5.2.23 | Pwm_Hal_QuadModeType | 152 |
| 4.5.2.24 | Pwm_Hal_QuadPhasePolarityType | 152 |
| 4.5.2.25 | Pwm_Hal_SyncModeType | 153 |
| 4.5.2.26 | Pwm_Hal_SyncTriggerMethodType | 153 |

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

| | |
|---|----|
| Pwm_Hal_CombineChnPairCfg | |
| Pwm combine mode config struct | 3 |
| Pwm_Hal_CommonCfg | |
| Pwm common base config struct | 7 |
| Pwm_Hal_CompareModeChnCfg | |
| Pwm compare mode config struct | 10 |
| Pwm_Hal_FaultCfg | |
| PWM Fault configuration structure | 12 |
| Pwm_Hal_FaultPinCfg | |
| PWM Fault channel configuration structure | 14 |
| Pwm_Hal_IndependentChnCfg | |
| Pwm independent channel config struct | 15 |
| Pwm_Hal_InputCfg | |
| Pwm input mode config struct | 17 |
| Pwm_Hal_InputChannelCfg | |
| Pwm input channel config struct | 18 |
| Pwm_Hal_InputCommonCfg | |
| Pwm input mode common config struct | 20 |
| Pwm_Hal_OutputCfg | |
| Pwm output mode config struct | 21 |
| Pwm_Hal_OutputChannelCfg | |
| Pwm output channel config struct | 22 |
| Pwm_Hal_OutputCommonCfg | |
| Pwm output mode common config struct | 24 |
| Pwm_Hal_QuadDecoderCfg | |
| PWM quadrature configuration structure | 25 |
| Pwm_Hal_QuadPhaseCfg | |
| PWM quadrature decoder phase input configuration structure | 28 |
| Pwm_Hal_SyncCfg | |
| PWM sync configuration structure. Please don't use software and hardware trigger simultaneously | 28 |

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

| | |
|---|-----|
| AC784xx_API_Reference_Manual_PWM.pdf | 33 |
| AC784xx_Pwm_Reg.h | |
| This file provides pwm hardware integration interface | 33 |
| Pwm_Hal.c | 59 |
| Pwm_Hal.h | 103 |
| Pwm_Hal_Types.h | 142 |

Chapter 3

Class Documentation

3.1 Pwm_Hal_CombineChnPairCfg Struct Reference

pwm combine mode config struct

```
#include <Pwm_Hal_Types.h>
```

Public Attributes

- uint16 [Ch1stValue](#)
- uint8 [Ch1stDither](#)
- boolean [EnableCh1stMatchTrigger](#)
- boolean [EnableCh1stInterrupt](#)
- boolean [EnableCh1stEventDmaReq](#)
- [Pwm_Hal_OutputLevelType](#) [Ch1stInitLevel](#)
- [Pwm_Hal_ActivePolarityType](#) [Ch1stActivePolarity](#)
- [Pwm_Hal_ChannelMatchDirType](#) [Ch1stMatchDir](#)
- uint16 [Ch2ndValue](#)
- uint8 [Ch2ndDither](#)
- boolean [EnableCh2ndMatchTrigger](#)
- boolean [EnableCh2ndInterrupt](#)
- boolean [EnableCh2ndEventDmaReq](#)
- [Pwm_Hal_OutputLevelType](#) [Ch2ndInitLevel](#)
- [Pwm_Hal_ActivePolarityType](#) [Ch2ndActivePolarity](#)
- [Pwm_Hal_ChannelMatchDirType](#) [Ch2ndMatchDir](#)
- [Pwm_Hal_OutputLevelModeType](#) [LevelMode](#)
- boolean [EnableChComplementation](#)
- boolean [EnableSymmetric](#)
- boolean [EnableDeadTime](#)
- uint16 [DeadTimeValue](#)
- [Pwm_Hal_DeadTimePscType](#) [TimePsc](#)

3.1.1 Detailed Description

pwm combine mode config struct

Definition at line 384 of file [Pwm_Hal_Types.h](#).

3.1.2 Member Data Documentation

3.1.2.1 Ch1stActivePolarity

[Pwm_Hal_ActivePolarityType](#) Pwm_Hal_CombineChnPairCfg::Ch1stActivePolarity

Definition at line 392 of file Pwm_Hal_Types.h.

3.1.2.2 Ch1stDither

uint8 Pwm_Hal_CombineChnPairCfg::Ch1stDither

Definition at line 387 of file Pwm_Hal_Types.h.

3.1.2.3 Ch1stInitLevel

[Pwm_Hal_OutputLevelType](#) Pwm_Hal_CombineChnPairCfg::Ch1stInitLevel

Definition at line 391 of file Pwm_Hal_Types.h.

3.1.2.4 Ch1stMatchDir

[Pwm_Hal_ChannelMatchDirType](#) Pwm_Hal_CombineChnPairCfg::Ch1stMatchDir

Definition at line 393 of file Pwm_Hal_Types.h.

3.1.2.5 Ch1stValue

uint16 Pwm_Hal_CombineChnPairCfg::Ch1stValue

Definition at line 386 of file Pwm_Hal_Types.h.

3.1.2.6 Ch2ndActivePolarity

[Pwm_Hal_ActivePolarityType](#) Pwm_Hal_CombineChnPairCfg::Ch2ndActivePolarity

Definition at line 400 of file Pwm_Hal_Types.h.

3.1.2.7 Ch2ndDither

```
uint8 Pwm_Hal_CombineChnPairCfg::Ch2ndDither
```

Definition at line 395 of file Pwm_Hal_Types.h.

3.1.2.8 Ch2ndInitLevel

```
Pwm_Hal_OutputLevelType Pwm_Hal_CombineChnPairCfg::Ch2ndInitLevel
```

Definition at line 399 of file Pwm_Hal_Types.h.

3.1.2.9 Ch2ndMatchDir

```
Pwm_Hal_ChannelMatchDirType Pwm_Hal_CombineChnPairCfg::Ch2ndMatchDir
```

Definition at line 401 of file Pwm_Hal_Types.h.

3.1.2.10 Ch2ndValue

```
uint16 Pwm_Hal_CombineChnPairCfg::Ch2ndValue
```

Definition at line 394 of file Pwm_Hal_Types.h.

3.1.2.11 DeadTimeValue

```
uint16 Pwm_Hal_CombineChnPairCfg::DeadTimeValue
```

Definition at line 406 of file Pwm_Hal_Types.h.

3.1.2.12 EnableCh1stEventDmaReq

```
boolean Pwm_Hal_CombineChnPairCfg::EnableCh1stEventDmaReq
```

Definition at line 390 of file Pwm_Hal_Types.h.

3.1.2.13 EnableCh1stInterrupt

```
boolean Pwm_Hal_CombineChnPairCfg::EnableCh1stInterrupt
```

Definition at line 389 of file Pwm_Hal_Types.h.

3.1.2.14 EnableCh1stMatchTrigger

```
boolean Pwm_Hal_CombineChnPairCfg::EnableCh1stMatchTrigger
```

Definition at line 388 of file Pwm_Hal_Types.h.

3.1.2.15 EnableCh2ndEventDmaReq

```
boolean Pwm_Hal_CombineChnPairCfg::EnableCh2ndEventDmaReq
```

Definition at line 398 of file Pwm_Hal_Types.h.

3.1.2.16 EnableCh2ndInterrupt

```
boolean Pwm_Hal_CombineChnPairCfg::EnableCh2ndInterrupt
```

Definition at line 397 of file Pwm_Hal_Types.h.

3.1.2.17 EnableCh2ndMatchTrigger

```
boolean Pwm_Hal_CombineChnPairCfg::EnableCh2ndMatchTrigger
```

Definition at line 396 of file Pwm_Hal_Types.h.

3.1.2.18 EnableChComplementation

```
boolean Pwm_Hal_CombineChnPairCfg::EnableChComplementation
```

Definition at line 403 of file Pwm_Hal_Types.h.

3.1.2.19 EnableDeadTime

`boolean Pwm_Hal_CombineChnPairCfg::EnableDeadTime`

Definition at line 405 of file `Pwm_Hal_Types.h`.

3.1.2.20 EnableSymmetric

`boolean Pwm_Hal_CombineChnPairCfg::EnableSymmetric`

Definition at line 404 of file `Pwm_Hal_Types.h`.

3.1.2.21 LevelMode

`Pwm_Hal_OutputLevelModeType Pwm_Hal_CombineChnPairCfg::LevelMode`

Definition at line 402 of file `Pwm_Hal_Types.h`.

3.1.2.22 TimePsc

`Pwm_Hal_DeadTimePscType Pwm_Hal_CombineChnPairCfg::TimePsc`

Definition at line 407 of file `Pwm_Hal_Types.h`.

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

- [Pwm_Hal_Types.h](#)

3.2 Pwm_Hal_CommonCfg Struct Reference

pwm common base config struct

```
#include <Pwm_Hal_Types.h>
```

Public Attributes

- [Pwm_Hal_ClockSourceType](#) `ClockSource`
- `uint16` [Prescaler](#)
- `uint16` [MinCount](#)
- `uint16` [MaxCount](#)
- `uint8` [PeriodDither](#)
- `boolean` [EnableDmaTransLen](#)
- `uint8` [DmaTransLen](#)
- `boolean` [EnableOverflowEvent](#)
- `boolean` [EnableUnderflowEvent](#)
- `boolean` [EnableOverflowInterrupt](#)
- `boolean` [EnableOverflowDmaReq](#)
- `uint8` [OverflowFreq](#)
- [Pwm_Hal_Callback](#) `OverflowCallback`
- [Pwm_Hal_Callback](#) `ChannelCallback`

3.2.1 Detailed Description

pwm common base config struct

Definition at line 348 of file Pwm_Hal_Types.h.

3.2.2 Member Data Documentation

3.2.2.1 ChannelCallback

`Pwm_Hal_Callback` Pwm_Hal_CommonCfg::ChannelCallback

Definition at line 363 of file Pwm_Hal_Types.h.

3.2.2.2 ClockSource

`Pwm_Hal_ClockSourceType` Pwm_Hal_CommonCfg::ClockSource

Definition at line 350 of file Pwm_Hal_Types.h.

3.2.2.3 DmaTransLen

`uint8` Pwm_Hal_CommonCfg::DmaTransLen

Definition at line 356 of file Pwm_Hal_Types.h.

3.2.2.4 EnableDmaTransLen

`boolean` Pwm_Hal_CommonCfg::EnableDmaTransLen

Definition at line 355 of file Pwm_Hal_Types.h.

3.2.2.5 EnableOverflowDmaReq

`boolean` Pwm_Hal_CommonCfg::EnableOverflowDmaReq

Definition at line 360 of file Pwm_Hal_Types.h.

3.2.2.6 EnableOverflowEvent

```
boolean Pwm_Hal_CommonCfg::EnableOverflowEvent
```

Definition at line 357 of file Pwm_Hal_Types.h.

3.2.2.7 EnableOverflowInterrupt

```
boolean Pwm_Hal_CommonCfg::EnableOverflowInterrupt
```

Definition at line 359 of file Pwm_Hal_Types.h.

3.2.2.8 EnableUnderflowEvent

```
boolean Pwm_Hal_CommonCfg::EnableUnderflowEvent
```

Definition at line 358 of file Pwm_Hal_Types.h.

3.2.2.9 MaxCount

```
uint16 Pwm_Hal_CommonCfg::MaxCount
```

Definition at line 353 of file Pwm_Hal_Types.h.

3.2.2.10 MinCount

```
uint16 Pwm_Hal_CommonCfg::MinCount
```

Definition at line 352 of file Pwm_Hal_Types.h.

3.2.2.11 OverflowCallback

```
Pwm\_Hal\_Callback Pwm_Hal_CommonCfg::OverflowCallback
```

Definition at line 362 of file Pwm_Hal_Types.h.

3.2.2.12 OverflowFreq

```
uint8 Pwm_Hal_CommonCfg::OverflowFreq
```

Definition at line 361 of file Pwm_Hal_Types.h.

3.2.2.13 PeriodDither

```
uint8 Pwm_Hal_CommonCfg::PeriodDither
```

Definition at line 354 of file Pwm_Hal_Types.h.

3.2.2.14 Prescaler

```
uint16 Pwm_Hal_CommonCfg::Prescaler
```

Definition at line 351 of file Pwm_Hal_Types.h.

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

- [Pwm_Hal_Types.h](#)

3.3 Pwm_Hal_CompareModeChnCfg Struct Reference

pwm compare mode config struct

```
#include <Pwm_Hal_Types.h>
```

Public Attributes

- uint16 [ChnValue](#)
- boolean [EnableMatchTrigger](#)
- boolean [EnableInterrupt](#)
- boolean [EnableChnEventDmaReq](#)
- [Pwm_Hal_OutputLevelType](#) InitLevel
- [Pwm_Hal_ActivePolarityType](#) ActivePolarity
- [Pwm_Hal_CompareActionType](#) Action

3.3.1 Detailed Description

pwm compare mode config struct

Definition at line 413 of file Pwm_Hal_Types.h.

3.3.2 Member Data Documentation

3.3.2.1 Action

`Pwm_Hal_CompareActionType` `Pwm_Hal_CompareModeChnCfg::Action`

Definition at line 421 of file `Pwm_Hal_Types.h`.

3.3.2.2 ActivePolarity

`Pwm_Hal_ActivePolarityType` `Pwm_Hal_CompareModeChnCfg::ActivePolarity`

Definition at line 420 of file `Pwm_Hal_Types.h`.

3.3.2.3 ChnValue

`uint16` `Pwm_Hal_CompareModeChnCfg::ChnValue`

Definition at line 415 of file `Pwm_Hal_Types.h`.

3.3.2.4 EnableChnEventDmaReq

`boolean` `Pwm_Hal_CompareModeChnCfg::EnableChnEventDmaReq`

Definition at line 418 of file `Pwm_Hal_Types.h`.

3.3.2.5 EnableInterrupt

`boolean` `Pwm_Hal_CompareModeChnCfg::EnableInterrupt`

Definition at line 417 of file `Pwm_Hal_Types.h`.

3.3.2.6 EnableMatchTrigger

`boolean` `Pwm_Hal_CompareModeChnCfg::EnableMatchTrigger`

Definition at line 416 of file `Pwm_Hal_Types.h`.

3.3.2.7 InitLevel

[Pwm_Hal_OutputLevelType](#) [Pwm_Hal_CompareModeChnCfg::InitLevel](#)

Definition at line 419 of file [Pwm_Hal_Types.h](#).

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

- [Pwm_Hal_Types.h](#)

3.4 Pwm_Hal_FaultCfg Struct Reference

PWM Fault configuration structure.

```
#include <Pwm_Hal_Types.h>
```

Public Attributes

- [Pwm_Hal_FaultCtrlModeType](#) [FaultCtrlMode](#)
- [Pwm_Hal_FaultPinCfg](#) [FaultPinCfg](#) [[PWM_FAULT_INPUT_MAX](#)]
- [uint8](#) [FilterValue](#)
- [Pwm_Hal_InputFilterPscType](#) [FilterPsc](#)
- [boolean](#) [EnableChannelOutputCtrl](#) [[PWM_CHANNEL_PAIR_NUM](#)]
- [boolean](#) [EnableInterrupt](#)
- [boolean](#) [EnableHiz](#)
- [Pwm_Hal_Callback](#) [FaultCallback](#)

3.4.1 Detailed Description

PWM Fault configuration structure.

Definition at line 534 of file [Pwm_Hal_Types.h](#).

3.4.2 Member Data Documentation

3.4.2.1 EnableChannelOutputCtrl

[boolean](#) [Pwm_Hal_FaultCfg::EnableChannelOutputCtrl](#) [[PWM_CHANNEL_PAIR_NUM](#)]

Fault control channel output state

Definition at line 540 of file [Pwm_Hal_Types.h](#).

3.4.2.2 EnableHiz

```
boolean Pwm_Hal_FaultCfg::EnableHiz
```

Enable PWM fault Hiz Output

Definition at line 542 of file Pwm_Hal_Types.h.

3.4.2.3 EnableInterrupt

```
boolean Pwm_Hal_FaultCfg::EnableInterrupt
```

Enable PWM fault interrupt

Definition at line 541 of file Pwm_Hal_Types.h.

3.4.2.4 FaultCallback

```
Pwm_Hal_Callback Pwm_Hal_FaultCfg::FaultCallback
```

Definition at line 543 of file Pwm_Hal_Types.h.

3.4.2.5 FaultCtrlMode

```
Pwm_Hal_FaultCtrlModeType Pwm_Hal_FaultCfg::FaultCtrlMode
```

Fault mode

Definition at line 536 of file Pwm_Hal_Types.h.

3.4.2.6 FaultPinCfg

```
Pwm_Hal_FaultPinCfg Pwm_Hal_FaultCfg::FaultPinCfg[PWM_FAULT_INPUT_MAX]
```

Fault input channels configuration

Definition at line 537 of file Pwm_Hal_Types.h.

3.4.2.7 FilterPsc

[Pwm_Hal_InputFilterPscType](#) `Pwm_Hal_FaultCfg::FilterPsc`

Fault filter psc

Definition at line 539 of file `Pwm_Hal_Types.h`.

3.4.2.8 FilterValue

`uint8` `Pwm_Hal_FaultCfg::FilterValue`

Fault filter value

Definition at line 538 of file `Pwm_Hal_Types.h`.

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

- [Pwm_Hal_Types.h](#)

3.5 Pwm_Hal_FaultPinCfg Struct Reference

PWM Fault channel configuration structure.

```
#include <Pwm_Hal_Types.h>
```

Public Attributes

- boolean [EnableFaultInput](#)
- boolean [EnableFaultFilter](#)
- [Pwm_Hal_ActivePolarityType](#) `FaultPolarity`

3.5.1 Detailed Description

PWM Fault channel configuration structure.

Definition at line 524 of file `Pwm_Hal_Types.h`.

3.5.2 Member Data Documentation

3.5.2.1 EnableFaultFilter

```
boolean Pwm_Hal_FaultPinCfɡ::EnableFaultFilter
```

Fault channel filter state

Definition at line 527 of file Pwm_Hal_Types.h.

3.5.2.2 EnableFaultInput

```
boolean Pwm_Hal_FaultPinCfɡ::EnableFaultInput
```

Fault input channel state

Definition at line 526 of file Pwm_Hal_Types.h.

3.5.2.3 FaultPolarity

```
Pwm_Hal_ActivePolarityType Pwm_Hal_FaultPinCfɡ::FaultPolarity
```

Fault channel input polarity active

Definition at line 528 of file Pwm_Hal_Types.h.

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

- [Pwm_Hal_Types.h](#)

3.6 Pwm_Hal_IndependentChnCfɡ Struct Reference

pwm independent channel config struct

```
#include <Pwm_Hal_Types.h>
```

Public Attributes

- uint16 [ChnValue](#)
- uint8 [ChnDither](#)
- boolean [EnableMatchTrigger](#)
- boolean [EnableInterrupt](#)
- boolean [EnableChnEventDmaReq](#)
- [Pwm_Hal_OutputLevelType](#) [InitLevel](#)
- [Pwm_Hal_ActivePolarityType](#) [ActivePolarity](#)
- [Pwm_Hal_OutputLevelModeType](#) [LevelMode](#)

3.6.1 Detailed Description

pwm independent channel config struct

Definition at line 369 of file Pwm_Hal_Types.h.

3.6.2 Member Data Documentation

3.6.2.1 ActivePolarity

`Pwm_Hal_ActivePolarityType Pwm_Hal_IndependentChnCfɡ::ActivePolarity`

Definition at line 377 of file Pwm_Hal_Types.h.

3.6.2.2 ChnDither

`uint8 Pwm_Hal_IndependentChnCfɡ::ChnDither`

Definition at line 372 of file Pwm_Hal_Types.h.

3.6.2.3 ChnValue

`uint16 Pwm_Hal_IndependentChnCfɡ::ChnValue`

Definition at line 371 of file Pwm_Hal_Types.h.

3.6.2.4 EnableChnEventDmaReq

`boolean Pwm_Hal_IndependentChnCfɡ::EnableChnEventDmaReq`

Definition at line 375 of file Pwm_Hal_Types.h.

3.6.2.5 EnableInterrupt

`boolean Pwm_Hal_IndependentChnCfɡ::EnableInterrupt`

Definition at line 374 of file Pwm_Hal_Types.h.

3.6.2.6 EnableMatchTrigger

```
boolean Pwm_Hal_IndependentChnCfg::EnableMatchTrigger
```

Definition at line 373 of file Pwm_Hal_Types.h.

3.6.2.7 InitLevel

```
Pwm_Hal_OutputLevelType Pwm_Hal_IndependentChnCfg::InitLevel
```

Definition at line 376 of file Pwm_Hal_Types.h.

3.6.2.8 LevelMode

```
Pwm_Hal_OutputLevelModeType Pwm_Hal_IndependentChnCfg::LevelMode
```

Definition at line 378 of file Pwm_Hal_Types.h.

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

- [Pwm_Hal_Types.h](#)

3.7 Pwm_Hal_InputCfg Struct Reference

pwm input mode config struct

```
#include <Pwm_Hal_Types.h>
```

Public Attributes

- [Pwm_Hal_InputCommonCfg](#) CommonCfg
- [Pwm_Hal_InputChannelCfg](#) * ChannelCfg
- uint8 ChannelNum

3.7.1 Detailed Description

pwm input mode config struct

Definition at line 491 of file Pwm_Hal_Types.h.

3.7.2 Member Data Documentation

3.7.2.1 ChannelCfg

```
Pwm_Hal_InputChannelCfg* Pwm_Hal_InputCfg::ChannelCfg
```

Definition at line 494 of file Pwm_Hal_Types.h.

3.7.2.2 ChannelNum

```
uint8 Pwm_Hal_InputCfg::ChannelNum
```

Definition at line 495 of file Pwm_Hal_Types.h.

3.7.2.3 CommonCfg

```
Pwm_Hal_InputCommonCfg Pwm_Hal_InputCfg::CommonCfg
```

Definition at line 493 of file Pwm_Hal_Types.h.

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

- [Pwm_Hal_Types.h](#)

3.8 Pwm_Hal_InputChannelCfg Struct Reference

pwm input channel config struct

```
#include <Pwm_Hal_Types.h>
```

Public Attributes

- [Pwm_Hal_ChannelType](#) Channel
- [Pwm_Hal_InputChnModeType](#) ChannelMode
- [Pwm_Hal_EdgeType](#) CaptureEdge
- [Pwm_Hal_InputEventPscType](#) EventPsc
- [Pwm_Hal_DualInputContinuousModeType](#) DualInputContinuousMode
- [boolean](#) EnablePulseWidthMeasure
- [Pwm_Hal_DualInputMeasureType](#) DualInputMeasureType
- [boolean](#) EnableCounterReset
- [boolean](#) EnableInterrupt
- [boolean](#) EnableChnEventDmaReq
- [uint8](#) FilterValue

3.8.1 Detailed Description

pwm input channel config struct

Definition at line 473 of file Pwm_Hal_Types.h.

3.8.2 Member Data Documentation

3.8.2.1 CaptureEdge

`Pwm_Hal_EdgeType` `Pwm_Hal_InputChannelCfg::CaptureEdge`

Definition at line 477 of file `Pwm_Hal_Types.h`.

3.8.2.2 Channel

`Pwm_Hal_ChannelType` `Pwm_Hal_InputChannelCfg::Channel`

Definition at line 475 of file `Pwm_Hal_Types.h`.

3.8.2.3 ChannelMode

`Pwm_Hal_InputChnModeType` `Pwm_Hal_InputChannelCfg::ChannelMode`

Definition at line 476 of file `Pwm_Hal_Types.h`.

3.8.2.4 DualInputContinuousMode

`Pwm_Hal_DualInputContinuousModeType` `Pwm_Hal_InputChannelCfg::DualInputContinuousMode`

Definition at line 479 of file `Pwm_Hal_Types.h`.

3.8.2.5 DualInputMeasureType

`Pwm_Hal_DualInputMeasureType` `Pwm_Hal_InputChannelCfg::DualInputMeasureType`

Definition at line 481 of file `Pwm_Hal_Types.h`.

3.8.2.6 EnableChnEventDmaReq

`boolean` `Pwm_Hal_InputChannelCfg::EnableChnEventDmaReq`

Definition at line 484 of file `Pwm_Hal_Types.h`.

3.8.2.7 EnableCounterReset

```
boolean Pwm_Hal_InputChannelCfg::EnableCounterReset
```

Definition at line 482 of file Pwm_Hal_Types.h.

3.8.2.8 EnableInterrupt

```
boolean Pwm_Hal_InputChannelCfg::EnableInterrupt
```

Definition at line 483 of file Pwm_Hal_Types.h.

3.8.2.9 EnablePulseWidthMeasure

```
boolean Pwm_Hal_InputChannelCfg::EnablePulseWidthMeasure
```

Definition at line 480 of file Pwm_Hal_Types.h.

3.8.2.10 EventPsc

```
Pwm_Hal_InputEventPscType Pwm_Hal_InputChannelCfg::EventPsc
```

Definition at line 478 of file Pwm_Hal_Types.h.

3.8.2.11 FilterValue

```
uint8 Pwm_Hal_InputChannelCfg::FilterValue
```

Definition at line 485 of file Pwm_Hal_Types.h.

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

- [Pwm_Hal_Types.h](#)

3.9 Pwm_Hal_InputCommonCfg Struct Reference

pwm input mode common config struct

```
#include <Pwm_Hal_Types.h>
```

Public Attributes

- [Pwm_Hal_CommonCfg](#) * [BaseCfg](#)
- boolean [EnableHall](#)
- [Pwm_Hal_InputFilterPscType](#) [FilterPsc](#)

3.9.1 Detailed Description

pwm input mode common config struct

Definition at line 463 of file [Pwm_Hal_Types.h](#).

3.9.2 Member Data Documentation

3.9.2.1 BaseCfg

[Pwm_Hal_CommonCfg](#)* [Pwm_Hal_InputCommonCfg::BaseCfg](#)

Definition at line 465 of file [Pwm_Hal_Types.h](#).

3.9.2.2 EnableHall

boolean [Pwm_Hal_InputCommonCfg::EnableHall](#)

Definition at line 466 of file [Pwm_Hal_Types.h](#).

3.9.2.3 FilterPsc

[Pwm_Hal_InputFilterPscType](#) [Pwm_Hal_InputCommonCfg::FilterPsc](#)

Definition at line 467 of file [Pwm_Hal_Types.h](#).

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

- [Pwm_Hal_Types.h](#)

3.10 Pwm_Hal_OutputCfg Struct Reference

pwm output mode config struct

```
#include <Pwm_Hal_Types.h>
```

Public Attributes

- [Pwm_Hal_OutputCommonCfg](#) [CommonCfg](#)
- [Pwm_Hal_OutputChannelCfg](#) * [ChannelCfg](#)
- [uint8](#) [ChannelNum](#)

3.10.1 Detailed Description

pwm output mode config struct

Definition at line 453 of file [Pwm_Hal_Types.h](#).

3.10.2 Member Data Documentation

3.10.2.1 ChannelCfg

[Pwm_Hal_OutputChannelCfg](#)* [Pwm_Hal_OutputCfg::ChannelCfg](#)

Definition at line 456 of file [Pwm_Hal_Types.h](#).

3.10.2.2 ChannelNum

[uint8](#) [Pwm_Hal_OutputCfg::ChannelNum](#)

Definition at line 457 of file [Pwm_Hal_Types.h](#).

3.10.2.3 CommonCfg

[Pwm_Hal_OutputCommonCfg](#) [Pwm_Hal_OutputCfg::CommonCfg](#)

Definition at line 455 of file [Pwm_Hal_Types.h](#).

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

- [Pwm_Hal_Types.h](#)

3.11 Pwm_Hal_OutputChannelCfg Struct Reference

pwm output channel config struct

```
#include <Pwm_Hal_Types.h>
```

Public Attributes

- [Pwm_Hal_ChannelType](#) Channel
- [Pwm_Hal_OutputChnModeType](#) ChannelMode
- [Pwm_Hal_IndependentChnCfg](#) * [IndependentChnCfg](#)
- [Pwm_Hal_CombineChnPairCfg](#) * [ChnPairCfg](#)
- [Pwm_Hal_CompareModeChnCfg](#) * [CompareChnCfg](#)

3.11.1 Detailed Description

pwm output channel config struct

Definition at line 441 of file `Pwm_Hal_Types.h`.

3.11.2 Member Data Documentation

3.11.2.1 Channel

[Pwm_Hal_ChannelType](#) `Pwm_Hal_OutputChannelCfg::Channel`

Definition at line 443 of file `Pwm_Hal_Types.h`.

3.11.2.2 ChannelMode

[Pwm_Hal_OutputChnModeType](#) `Pwm_Hal_OutputChannelCfg::ChannelMode`

Definition at line 444 of file `Pwm_Hal_Types.h`.

3.11.2.3 ChnPairCfg

[Pwm_Hal_CombineChnPairCfg](#)* `Pwm_Hal_OutputChannelCfg::ChnPairCfg`

Definition at line 446 of file `Pwm_Hal_Types.h`.

3.11.2.4 CompareChnCfg

[Pwm_Hal_CompareModeChnCfg](#)* `Pwm_Hal_OutputChannelCfg::CompareChnCfg`

Definition at line 447 of file `Pwm_Hal_Types.h`.

3.11.2.5 IndependentChnCfg

[Pwm_Hal_IndependentChnCfg*](#) [Pwm_Hal_OutputChannelCfg::IndependentChnCfg](#)

Definition at line 445 of file [Pwm_Hal_Types.h](#).

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

- [Pwm_Hal_Types.h](#)

3.12 Pwm_Hal_OutputCommonCfg Struct Reference

pwm output mode common config struct

```
#include <Pwm_Hal_Types.h>
```

Public Attributes

- [Pwm_Hal_CommonCfg * BaseCfg](#)
- [Pwm_Hal_CountModeType CountMode](#)
- boolean [EnableInitTrigger](#)
- boolean [EnableMaxTrigger](#)
- uint8 [TriggerRatio](#)
- boolean [InitOutput](#)
- [Pwm_Hal_CombineCenterDutyModeType CombineCenterDutyMode](#)

3.12.1 Detailed Description

pwm output mode common config struct

Definition at line 427 of file [Pwm_Hal_Types.h](#).

3.12.2 Member Data Documentation

3.12.2.1 BaseCfg

[Pwm_Hal_CommonCfg*](#) [Pwm_Hal_OutputCommonCfg::BaseCfg](#)

Definition at line 429 of file [Pwm_Hal_Types.h](#).

3.12.2.2 CombineCenterDutyMode

[Pwm_Hal_CombineCenterDutyModeType](#) [Pwm_Hal_OutputCommonCfg::CombineCenterDutyMode](#)

Definition at line 435 of file [Pwm_Hal_Types.h](#).

3.12.2.3 CountMode

[Pwm_Hal_CountModeType](#) Pwm_Hal_OutputCommonCfg::CountMode

Definition at line 430 of file Pwm_Hal_Types.h.

3.12.2.4 EnableInitTrigger

boolean Pwm_Hal_OutputCommonCfg::EnableInitTrigger

Definition at line 431 of file Pwm_Hal_Types.h.

3.12.2.5 EnableMaxTrigger

boolean Pwm_Hal_OutputCommonCfg::EnableMaxTrigger

Definition at line 432 of file Pwm_Hal_Types.h.

3.12.2.6 InitOutput

boolean Pwm_Hal_OutputCommonCfg::InitOutput

Definition at line 434 of file Pwm_Hal_Types.h.

3.12.2.7 TriggerRatio

uint8 Pwm_Hal_OutputCommonCfg::TriggerRatio

Definition at line 433 of file Pwm_Hal_Types.h.

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

- [Pwm_Hal_Types.h](#)

3.13 Pwm_Hal_QuadDecoderCfg Struct Reference

PWM quadrature configuration structure.

```
#include <Pwm_Hal_Types.h>
```

Public Attributes

- [Pwm_Hal_CommonCfg](#) * BaseCfg
- [Pwm_Hal_QuadModeType](#) Mode
- [Pwm_Hal_QuadPhaseCfg](#) PhaseAConfig
- [Pwm_Hal_QuadPhaseCfg](#) PhaseBConfig
- [Pwm_Hal_QuadPhaseCfg](#) PhaseZConfig
- boolean [EnableQuad](#)
- [Pwm_Hal_InputFilterPscType](#) FilterPsc
- boolean [EnablePhaseZReset](#)

3.13.1 Detailed Description

PWM quadrature configuration structure.

Definition at line 558 of file Pwm_Hal_Types.h.

3.13.2 Member Data Documentation

3.13.2.1 BaseCfg

```
Pwm_Hal_CommonCfg* Pwm_Hal_QuadDecoderCfg::BaseCfg
```

PWM common config

Definition at line 560 of file Pwm_Hal_Types.h.

3.13.2.2 EnablePhaseZReset

```
boolean Pwm_Hal_QuadDecoderCfg::EnablePhaseZReset
```

Reset the counter when Z index event is detected>

Definition at line 567 of file Pwm_Hal_Types.h.

3.13.2.3 EnableQuad

```
boolean Pwm_Hal_QuadDecoderCfg::EnableQuad
```

Enable quadrature decode mode

Definition at line 565 of file Pwm_Hal_Types.h.

3.13.2.4 FilterPsc

[Pwm_Hal_InputFilterPscType](#) `Pwm_Hal_QuadDecoderCfg::FilterPsc`

Input Capture Filter psc

Definition at line 566 of file `Pwm_Hal_Types.h`.

3.13.2.5 Mode

[Pwm_Hal_QuadModeType](#) `Pwm_Hal_QuadDecoderCfg::Mode`

PWM quadrature decode mode

Definition at line 561 of file `Pwm_Hal_Types.h`.

3.13.2.6 PhaseAConfig

[Pwm_Hal_QuadPhaseCfg](#) `Pwm_Hal_QuadDecoderCfg::PhaseAConfig`

PhaseA config

Definition at line 562 of file `Pwm_Hal_Types.h`.

3.13.2.7 PhaseBConfig

[Pwm_Hal_QuadPhaseCfg](#) `Pwm_Hal_QuadDecoderCfg::PhaseBConfig`

PhaseB config

Definition at line 563 of file `Pwm_Hal_Types.h`.

3.13.2.8 PhaseZConfig

[Pwm_Hal_QuadPhaseCfg](#) `Pwm_Hal_QuadDecoderCfg::PhaseZConfig`

PhaseZ config

Definition at line 564 of file `Pwm_Hal_Types.h`.

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

- [Pwm_Hal_Types.h](#)

3.14 Pwm_Hal_QuadPhaseCfg Struct Reference

PWM quadrature decoder phase input configuration structure.

```
#include <Pwm_Hal_Types.h>
```

Public Attributes

- [Pwm_Hal_QuadPhasePolarityType](#) Polarity
- [uint8](#) [FilterValue](#)

3.14.1 Detailed Description

PWM quadrature decoder phase input configuration structure.

Definition at line 549 of file `Pwm_Hal_Types.h`.

3.14.2 Member Data Documentation

3.14.2.1 FilterValue

```
uint8 Pwm_Hal_QuadPhaseCfg::FilterValue
```

Filter value

Definition at line 552 of file `Pwm_Hal_Types.h`.

3.14.2.2 Polarity

```
Pwm\_Hal\_QuadPhasePolarityType Pwm_Hal_QuadPhaseCfg::Polarity
```

PhaseA Polarity

Definition at line 551 of file `Pwm_Hal_Types.h`.

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

- [Pwm_Hal_Types.h](#)

3.15 Pwm_Hal_SyncCfg Struct Reference

PWM sync configuration structure. Please don't use software and hardware trigger simultaneously.

```
#include <Pwm_Hal_Types.h>
```

Public Attributes

- [Pwm_Hal_SyncTriggerMethodType](#) [TriggerType](#)
- boolean [EnableHwSync0](#)
- boolean [EnableHwSync1](#)
- boolean [EnableHwSync2](#)
- boolean [DisableHwTriggerAfterTriggered](#)
- boolean [EnableCounterInitSync](#)
- boolean [EnableSwOutputCtrlSync](#)
- boolean [EnableDualChannelInvertSync](#)
- boolean [EnableOutputMaskSync](#)
- boolean [EnablePolaritySync](#)
- boolean [EnablePairedChnValSync](#) [PWM_CHANNEL_PAIR_NUM]
- boolean [EnableMaxLoadingPoint](#)
- boolean [EnableMinLoadingPoint](#)
- boolean [EnableSyncBypass](#)
- boolean [EnableSync](#)

3.15.1 Detailed Description

PWM sync configuration structure. Please don't use software and hardware trigger simultaneously.

Definition at line 502 of file Pwm_Hal_Types.h.

3.15.2 Member Data Documentation

3.15.2.1 DisableHwTriggerAfterTriggered

```
boolean Pwm_Hal_SyncCfg::DisableHwTriggerAfterTriggered
```

Available only for hardware trigger

Definition at line 508 of file Pwm_Hal_Types.h.

3.15.2.2 EnableCounterInitSync

```
boolean Pwm_Hal_SyncCfg::EnableCounterInitSync
```

Enable/disable CNTIN sync

Definition at line 509 of file Pwm_Hal_Types.h.

3.15.2.3 EnableDualChannelInvertSync

```
boolean Pwm_Hal_SyncCfg::EnableDualChannelInvertSync
```

Enable/disable INVCR sync

Definition at line 511 of file Pwm_Hal_Types.h.

3.15.2.4 EnableHwSync0

```
boolean Pwm_Hal_SyncCfg::EnableHwSync0
```

Enable/disable hardware sync trigger source 0

Definition at line 505 of file Pwm_Hal_Types.h.

3.15.2.5 EnableHwSync1

```
boolean Pwm_Hal_SyncCfg::EnableHwSync1
```

Enable/disable hardware sync trigger source 1

Definition at line 506 of file Pwm_Hal_Types.h.

3.15.2.6 EnableHwSync2

```
boolean Pwm_Hal_SyncCfg::EnableHwSync2
```

Enable/disable hardware sync trigger source 2

Definition at line 507 of file Pwm_Hal_Types.h.

3.15.2.7 EnableMaxLoadingPoint

```
boolean Pwm_Hal_SyncCfg::EnableMaxLoadingPoint
```

Enable/disable maximum loading point

Definition at line 515 of file Pwm_Hal_Types.h.

3.15.2.8 EnableMinLoadingPoint

```
boolean Pwm_Hal_SyncCfg::EnableMinLoadingPoint
```

Enable/disable minimum loading point

Definition at line 516 of file Pwm_Hal_Types.h.

3.15.2.9 EnableOutputMaskSync

```
boolean Pwm_Hal_SyncCfg::EnableOutputMaskSync
```

Enable/disable OMCR sync

Definition at line 512 of file Pwm_Hal_Types.h.

3.15.2.10 EnablePairedChnValSync

```
boolean Pwm_Hal_SyncCfg::EnablePairedChnValSync[PWM_CHANNEL_PAIR_NUM]
```

Enable/disable dual channel CHV sync

Definition at line 514 of file Pwm_Hal_Types.h.

3.15.2.11 EnablePolaritySync

```
boolean Pwm_Hal_SyncCfg::EnablePolaritySync
```

Enable/disable CHOPOLCR sync

Definition at line 513 of file Pwm_Hal_Types.h.

3.15.2.12 EnableSwOutputCtrlSync

```
boolean Pwm_Hal_SyncCfg::EnableSwOutputCtrlSync
```

Enable/disable CHOSWCR sync

Definition at line 510 of file Pwm_Hal_Types.h.

3.15.2.13 EnableSync

```
boolean Pwm_Hal_SyncCfg::EnableSync
```

PWM Enable/disable

Definition at line 518 of file Pwm_Hal_Types.h.

3.15.2.14 EnableSyncBypass

```
boolean Pwm_Hal_SyncCfg::EnableSyncBypass
```

Enable/disable Synchronization bypass

Definition at line 517 of file Pwm_Hal_Types.h.

3.15.2.15 TriggerType

```
Pwm_Hal_SyncTriggerMethodType Pwm_Hal_SyncCfg::TriggerType
```

Synchronization trigger mode

Definition at line 504 of file Pwm_Hal_Types.h.

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

- [Pwm_Hal_Types.h](#)

Chapter 4

File Documentation

4.1 AC784xx_API_Reference_Manual_PWM.pdf File Reference

4.2 AC784xx_Pwm_Reg.h File Reference

This file provides pwm hardware integration interface.

```
#include "Pwm_Hal_Types.h"
#include "Device_Register.h"
```

Functions

- LOCAL_INLINE PWM_Type * [Pwm_Reg_GetBase](#) (Pwm_Hal_InstanceType Instance)
- LOCAL_INLINE void [Pwm_Reg_SetClockSource](#) (PWM_Type *Base, [Pwm_Hal_ClockSourceType](#) ClkSource)
- LOCAL_INLINE void [Pwm_Reg_SetClockPsc](#) (PWM_Type *Base, uint16 ClkPsc)
- LOCAL_INLINE void [Pwm_Reg_SetCountMode](#) (PWM_Type *Base, [Pwm_Hal_CountModeType](#) Mode)
- LOCAL_INLINE void [Pwm_Reg_SetMaxCountValue](#) (PWM_Type *Base, uint16 Value)
- LOCAL_INLINE uint16 [Pwm_Reg_GetMaxCountValue](#) (const PWM_Type *Base)
- LOCAL_INLINE void [Pwm_Reg_SetInitCountValue](#) (PWM_Type *Base, uint16 Value)
- LOCAL_INLINE uint16 [Pwm_Reg_GetInitCountValue](#) (const PWM_Type *Base)
- LOCAL_INLINE void [Pwm_Reg_ResetCounter](#) (PWM_Type *Base)
- LOCAL_INLINE uint16 [Pwm_Reg_GetCountValue](#) (const PWM_Type *Base)
- LOCAL_INLINE void [Pwm_Reg_SetPeriodDither](#) (PWM_Type *Base, uint8 Value)
- LOCAL_INLINE void [Pwm_Reg_SetCntOverflowFreq](#) (PWM_Type *Base, uint8 Freq)
- LOCAL_INLINE void [Pwm_Reg_EnableOverflowDmaReq](#) (PWM_Type *Base, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_EnableDmaTransfer](#) (PWM_Type *Base, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_SetDmaTransferLen](#) (PWM_Type *Base, uint8 Len)
- LOCAL_INLINE void [Pwm_Reg_EnableOutputInit](#) (PWM_Type *Base, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_EnableChannelDmaReq](#) (PWM_Type *Base, [Pwm_Hal_ChannelType](#) Channel, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_SetChannelMSR](#) (PWM_Type *Base, [Pwm_Hal_ChannelType](#) Channel, uint8 Value)
- LOCAL_INLINE void [Pwm_Reg_SetChannelELSR](#) (PWM_Type *Base, [Pwm_Hal_ChannelType](#) Channel, uint8 Value)
- LOCAL_INLINE void [Pwm_Reg_SetChannelValue](#) (PWM_Type *Base, [Pwm_Hal_ChannelType](#) Channel, uint16 Value)
- LOCAL_INLINE uint16 [Pwm_Reg_GetChannelValue](#) (const PWM_Type *Base, [Pwm_Hal_ChannelType](#) Channel)

- LOCAL_INLINE void [Pwm_Reg_SetChannelMatchDitherReg](#) (PWM_Type *Base, uint8 DitherRegNum, uint32 RegValue)
- LOCAL_INLINE void [Pwm_Reg_SetChannelMatchDither](#) (PWM_Type *Base, [Pwm_Hal_ChannelType](#) Channel, uint8 DitherValue)
- LOCAL_INLINE void [Pwm_Reg_EnableOverflowInterrupt](#) (PWM_Type *Base, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_EnableChannelInterrupt](#) (PWM_Type *Base, [Pwm_Hal_ChannelType](#) Channel, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_EnableOverflowEvent](#) (PWM_Type *Base, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_EnableUnderflowEvent](#) (PWM_Type *Base, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_SetChannelOutputInitLevel](#) (PWM_Type *Base, [Pwm_Hal_ChannelType](#) Channel, [Pwm_Hal_OutputLevelType](#) Level)
- LOCAL_INLINE void [Pwm_Reg_EnableChannelCombine](#) (PWM_Type *Base, [Pwm_Hal_ChannelPairType](#) ChannelPair, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_EnablePairChSymmetric](#) (PWM_Type *Base, [Pwm_Hal_ChannelPairType](#) ChannelPair, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_EnablePairChComplement](#) (PWM_Type *Base, [Pwm_Hal_ChannelPairType](#) ChannelPair, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_EnablePairChValueSync](#) (PWM_Type *Base, [Pwm_Hal_ChannelPairType](#) ChannelPair, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_EnablePairChDeadtime](#) (PWM_Type *Base, [Pwm_Hal_ChannelPairType](#) ChannelPair, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_SetPairChDeadtime](#) (PWM_Type *Base, [Pwm_Hal_ChannelPairType](#) ChannelPair, [Pwm_Hal_DeadTimePscType](#) Psc, uint16 Value)
- LOCAL_INLINE void [Pwm_Reg_EnablePairChInvert](#) (PWM_Type *Base, [Pwm_Hal_ChannelPairType](#) ChannelPair, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_SetCombineCenterDutyType](#) (PWM_Type *Base, [Pwm_Hal_CombineCenterDutyModeType](#) DutyMode)
- LOCAL_INLINE void [Pwm_Reg_SetChannelMatchDir](#) (PWM_Type *Base, [Pwm_Hal_ChannelType](#) Channel, [Pwm_Hal_ChannelMatchDirType](#) Dir)
- LOCAL_INLINE void [Pwm_Reg_EnablePairPulseWidthMeasurement](#) (PWM_Type *Base, [Pwm_Hal_ChannelPairType](#) ChannelPair, boolean State)
- LOCAL_INLINE void [Pwm_Reg_EnableHall](#) (PWM_Type *Base, boolean State)
- LOCAL_INLINE uint32 [Pwm_Reg_GetHallStatus](#) (const PWM_Type *Base)
- LOCAL_INLINE void [Pwm_Reg_EnableInitTrigger](#) (PWM_Type *Base, boolean State)
- LOCAL_INLINE void [Pwm_Reg_EnableMaxTrigger](#) (PWM_Type *Base, boolean State)
- LOCAL_INLINE void [Pwm_Reg_EnableChannelMatchTrigger](#) (PWM_Type *Base, [Pwm_Hal_ChannelType](#) Channel, boolean State)
- LOCAL_INLINE void [Pwm_Reg_SetTriggerRatio](#) (PWM_Type *Base, uint8 Ratio)
- LOCAL_INLINE uint32 [Pwm_Reg_GetChannelTriggerFlag](#) (const PWM_Type *Base)
- LOCAL_INLINE void [Pwm_Reg_ClearChannelTriggerFlag](#) (PWM_Type *Base)
- LOCAL_INLINE void [Pwm_Reg_EnableChannelOutputMask](#) (PWM_Type *Base, [Pwm_Hal_ChannelType](#) Channel, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_SetOutputMask](#) (PWM_Type *Base, uint8 Mask)
- LOCAL_INLINE uint32 [Pwm_Reg_GetOutputMask](#) (const PWM_Type *Base)
- LOCAL_INLINE void [Pwm_Reg_SetChannelPolarity](#) (PWM_Type *Base, [Pwm_Hal_ChannelType](#) Channel, [Pwm_Hal_ActivePolarityType](#) Polarity)
- LOCAL_INLINE uint32 [Pwm_Reg_GetAllChannelLevel](#) (const PWM_Type *Base)
- LOCAL_INLINE uint32 [Pwm_Reg_GetOverflowFlag](#) (const PWM_Type *Base)
- LOCAL_INLINE void [Pwm_Reg_ClearOverflowFlag](#) (PWM_Type *Base)
- LOCAL_INLINE uint32 [Pwm_Reg_GetOverflowDir](#) (const PWM_Type *Base)
- LOCAL_INLINE uint32 [Pwm_Reg_GetChannelEventFlag](#) (const PWM_Type *Base)
- LOCAL_INLINE void [Pwm_Reg_ClearChannelEventFlag](#) (PWM_Type *Base, uint32 mask)
- LOCAL_INLINE uint32 [Pwm_Reg_GetChannelInterruptFlag](#) (const PWM_Type *Base, [Pwm_Hal_ChannelType](#) Channel)
- LOCAL_INLINE void [Pwm_Reg_ClearChannelInterruptFlag](#) (PWM_Type *Base, [Pwm_Hal_ChannelType](#) Channel)
- LOCAL_INLINE void [Pwm_Reg_EnableChannelSoftControl](#) (PWM_Type *Base, [Pwm_Hal_ChannelType](#) Channel, boolean State)
- LOCAL_INLINE void [Pwm_Reg_SetSoftControlEnableStatesMask](#) (PWM_Type *Base, uint8 Mask)

- LOCAL_INLINE uint32 [Pwm_Reg_ReadChannelSoftControlState](#) (const PWM_Type *Base)
- LOCAL_INLINE void [Pwm_Reg_SetChannelSoftControlLevel](#) (PWM_Type *Base, [Pwm_Hal_ChannelType](#) Channel, [Pwm_Hal_OutputLevelType](#) Level)
- LOCAL_INLINE void [Pwm_Reg_SetSoftControlLevelMask](#) (PWM_Type *Base, uint8 Mask)
- LOCAL_INLINE void [Pwm_Reg_EnableSync](#) (PWM_Type *Base, boolean State)
- LOCAL_INLINE void [Pwm_Reg_SetSyncType](#) (PWM_Type *Base, uint8 Type)
- LOCAL_INLINE void [Pwm_Reg_SetSyncMode](#) (PWM_Type *Base, [Pwm_Hal_SyncModeType](#) Mode)
- LOCAL_INLINE void [Pwm_Reg_EnableSyncBYP](#) (PWM_Type *Base, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_EnableSyncHwTriggerSrc](#) (PWM_Type *Base, uint8 HwTrigNum, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_TrigSoftwareSync](#) (PWM_Type *Base, boolean State)
- LOCAL_INLINE void [Pwm_Reg_EnableMaxLoadingPointSync](#) (PWM_Type *Base, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_EnableMinLoadingPointSync](#) (PWM_Type *Base, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_EnablePOLSync](#) (PWM_Type *Base, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_EnableOutputMaskSync](#) (PWM_Type *Base, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_ChooseBufferOrRegisterToRead](#) (PWM_Type *Base, boolean ChooseReg)
- LOCAL_INLINE void [Pwm_Reg_DisableHwTriggerSyncAfterTriggered](#) (PWM_Type *Base, boolean Disable)
- LOCAL_INLINE void [Pwm_Reg_EnableCNTINSync](#) (PWM_Type *Base, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_EnableINVCRSync](#) (PWM_Type *Base, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_EnableCHOSWCERSync](#) (PWM_Type *Base, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_SetPOLTriggerMode](#) (PWM_Type *Base, [Pwm_Hal_SyncTriggerMethodType](#) Mode, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_SetCHOSWCERSyncTriggerMode](#) (PWM_Type *Base, [Pwm_Hal_SyncTriggerMethodType](#) Mode, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_SetINVCRSyncTriggerMode](#) (PWM_Type *Base, [Pwm_Hal_SyncTriggerMethodType](#) Mode, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_SetOMCRSyncTriggerMode](#) (PWM_Type *Base, [Pwm_Hal_SyncTriggerMethodType](#) Mode, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_SetMCVRSyncMode](#) (PWM_Type *Base, [Pwm_Hal_SyncTriggerMethodType](#) Mode, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_SetCNTSyncTriggerMode](#) (PWM_Type *Base, [Pwm_Hal_SyncTriggerMethodType](#) Mode, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_EnableGlobalTimeBase](#) (PWM_Type *Base, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_EnableGlobalTimeBaseOutput](#) (PWM_Type *Base, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_SetInputFilterPsc](#) (PWM_Type *Base, [Pwm_Hal_InputFilterPscType](#) Psc)
- LOCAL_INLINE void [Pwm_Reg_SetChannelInputFilterVal](#) (PWM_Type *Base, [Pwm_Hal_ChannelType](#) Channel, uint8 Value)
- LOCAL_INLINE void [Pwm_Reg_SetCaptureEventPsc](#) (PWM_Type *Base, [Pwm_Hal_ChannelType](#) Channel, [Pwm_Hal_InputEventPscType](#) EventPsc)
- LOCAL_INLINE void [Pwm_Reg_EnableChannelEventReset](#) (PWM_Type *Base, [Pwm_Hal_ChannelType](#) Channel, boolean State)
- LOCAL_INLINE void [Pwm_Reg_EnablePairChDualEdgeCapture](#) (PWM_Type *Base, [Pwm_Hal_ChannelPairType](#) ChannelPair, boolean State)
- LOCAL_INLINE void [Pwm_Reg_SetPairChDualEdgeCapture](#) (PWM_Type *Base, [Pwm_Hal_ChannelPairType](#) ChannelPair, boolean State)
- LOCAL_INLINE boolean [Pwm_Reg_IsPairChDualEdgeCaptureEnabled](#) (const PWM_Type *Base, [Pwm_Hal_ChannelPairType](#) ChannelPair)
- LOCAL_INLINE void [Pwm_Reg_SetFaultMode](#) (PWM_Type *Base, [Pwm_Hal_FaultCtrlModeType](#) Mode)
- LOCAL_INLINE void [Pwm_Reg_EnableFaultInputFilter](#) (PWM_Type *Base, [Pwm_Hal_FaultInputIdType](#) FaultInputId, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_SetFaultInputFilterValue](#) (PWM_Type *Base, uint8 Value)
- LOCAL_INLINE void [Pwm_Reg_EnableFaultPinInput](#) (PWM_Type *Base, [Pwm_Hal_FaultInputIdType](#) FaultInputId, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_SetFaultInputPolarity](#) (PWM_Type *Base, [Pwm_Hal_FaultInputIdType](#) FaultInputId, [Pwm_Hal_ActivePolarityType](#) Polarity)
- LOCAL_INLINE void [Pwm_Reg_EnableFaultInterrupt](#) (PWM_Type *Base, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_EnableFaultHizOutput](#) (PWM_Type *Base, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_EnableChannelHizOutput](#) (PWM_Type *Base, [Pwm_Hal_ChannelType](#) Channel, boolean Enable)

- LOCAL_INLINE void [Pwm_Reg_EnablePairChFaultCtrl](#) (PWM_Type *Base, [Pwm_Hal_ChannelPairType](#) Channel↔ Pair, boolean Enable)
- LOCAL_INLINE uint32 [Pwm_Reg_GetFaultPinFlag](#) (const PWM_Type *Base, [Pwm_Hal_FaultInputIdType](#) Fault↔ InputId)
- LOCAL_INLINE void [Pwm_Reg_ClearFaultPinFlag](#) (PWM_Type *Base, [Pwm_Hal_FaultInputIdType](#) FaultInputId)
- LOCAL_INLINE uint32 [Pwm_Reg_GetFaultFlag](#) (const PWM_Type *Base)
- LOCAL_INLINE void [Pwm_Reg_ClearFaultFlag](#) (PWM_Type *Base)
- LOCAL_INLINE uint32 [Pwm_Reg_GetFaultInputStatus](#) (const PWM_Type *Base)
- LOCAL_INLINE void [Pwm_Reg_SetQuadEncodeMode](#) (PWM_Type *Base, [Pwm_Hal_QuadModeType](#) Mode)
- LOCAL_INLINE void [Pwm_Reg_SetQuadPhaseAPolarity](#) (PWM_Type *Base, [Pwm_Hal_QuadPhasePolarityType](#) Polarity)
- LOCAL_INLINE void [Pwm_Reg_SetQuadPhaseBPolarity](#) (PWM_Type *Base, [Pwm_Hal_QuadPhasePolarityType](#) Polarity)
- LOCAL_INLINE void [Pwm_Reg_SetQuadPhaseZPolarity](#) (PWM_Type *Base, [Pwm_Hal_QuadPhasePolarityType](#) Polarity)
- LOCAL_INLINE void [Pwm_Reg_EnableQuadPhaseZResetCnt](#) (PWM_Type *Base, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_EnableQuadDecoder](#) (PWM_Type *Base, boolean Enable)
- LOCAL_INLINE uint32 [Pwm_Reg_GetQuadCountDir](#) (const PWM_Type *Base)
- LOCAL_INLINE uint32 [Pwm_Reg_GetQuadOverflowDir](#) (const PWM_Type *Base)
- LOCAL_INLINE uint32 [Pwm_Reg_GetQuadPhaseZFlag](#) (const PWM_Type *Base)
- LOCAL_INLINE void [Pwm_Reg_ClearQuadPhaseZFlag](#) (PWM_Type *Base)
- LOCAL_INLINE void [Pwm_Reg_EnableWriteProtection](#) (PWM_Type *Base, boolean Enable)
- LOCAL_INLINE void [Pwm_Reg_SetDebugMode](#) (PWM_Type *Base, [Pwm_Hal_DebugModeType](#) Mode)

4.2.1 Detailed Description

This file provides pwm hardware integration interface.

4.2.2 Function Documentation

4.2.2.1 Pwm_Reg_ChooseBufferOrRegisterToRead()

```
LOCAL_INLINE void Pwm_Reg_ChooseBufferOrRegisterToRead (
    PWM_Type * Base,
    boolean ChooseReg )
```

Definition at line 577 of file AC784xx_Pwm_Reg.h.

4.2.2.2 Pwm_Reg_ClearChannelEventFlag()

```
LOCAL_INLINE void Pwm_Reg_ClearChannelEventFlag (
    PWM_Type * Base,
    uint32 mask )
```

Definition at line 452 of file AC784xx_Pwm_Reg.h.

4.2.2.3 Pwm_Reg_ClearChannelInterruptFlag()

```
LOCAL_INLINE void Pwm_Reg_ClearChannelInterruptFlag (
    PWM_Type * Base,
    Pwm_Hal_ChannelType Channel )
```

Definition at line 470 of file AC784xx_Pwm_Reg.h.

4.2.2.4 Pwm_Reg_ClearChannelTriggerFlag()

```
LOCAL_INLINE void Pwm_Reg_ClearChannelTriggerFlag (
    PWM_Type * Base )
```

Definition at line 379 of file AC784xx_Pwm_Reg.h.

4.2.2.5 Pwm_Reg_ClearFaultFlag()

```
LOCAL_INLINE void Pwm_Reg_ClearFaultFlag (
    PWM_Type * Base )
```

Definition at line 848 of file AC784xx_Pwm_Reg.h.

4.2.2.6 Pwm_Reg_ClearFaultPinFlag()

```
LOCAL_INLINE void Pwm_Reg_ClearFaultPinFlag (
    PWM_Type * Base,
    Pwm_Hal_FaultInputIdType FaultInputId )
```

Definition at line 827 of file AC784xx_Pwm_Reg.h.

4.2.2.7 Pwm_Reg_ClearOverflowFlag()

```
LOCAL_INLINE void Pwm_Reg_ClearOverflowFlag (
    PWM_Type * Base )
```

Definition at line 427 of file AC784xx_Pwm_Reg.h.

4.2.2.8 Pwm_Reg_ClearQuadPhaseZFlag()

```
LOCAL_INLINE void Pwm_Reg_ClearQuadPhaseZFlag (
    PWM_Type * Base )
```

Definition at line 921 of file AC784xx_Pwm_Reg.h.

4.2.2.9 Pwm_Reg_DisableHwTriggerSyncAfterTrigged()

```
LOCAL_INLINE void Pwm_Reg_DisableHwTriggerSyncAfterTrigged (
    PWM_Type * Base,
    boolean Disable )
```

Definition at line 582 of file AC784xx_Pwm_Reg.h.

4.2.2.10 Pwm_Reg_EnableChannelCombine()

```
LOCAL_INLINE void Pwm_Reg_EnableChannelCombine (
    PWM_Type * Base,
    Pwm_Hal_ChannelPairType ChannelPair,
    boolean Enable )
```

Definition at line 247 of file AC784xx_Pwm_Reg.h.

4.2.2.11 Pwm_Reg_EnableChannelDmaReq()

```
LOCAL_INLINE void Pwm_Reg_EnableChannelDmaReq (
    PWM_Type * Base,
    Pwm_Hal_ChannelType Channel,
    boolean Enable )
```

Definition at line 153 of file AC784xx_Pwm_Reg.h.

4.2.2.12 Pwm_Reg_EnableChannelEventReset()

```
LOCAL_INLINE void Pwm_Reg_EnableChannelEventReset (
    PWM_Type * Base,
    Pwm_Hal_ChannelType Channel,
    boolean State )
```

Definition at line 710 of file AC784xx_Pwm_Reg.h.

4.2.2.13 Pwm_Reg_EnableChannelHizOutput()

```
LOCAL_INLINE void Pwm_Reg_EnableChannelHizOutput (
    PWM_Type * Base,
    Pwm_Hal_ChannelType Channel,
    boolean Enable )
```

Definition at line 799 of file AC784xx_Pwm_Reg.h.

4.2.2.14 Pwm_Reg_EnableChannelInterrupt()

```
LOCAL_INLINE void Pwm_Reg_EnableChannelInterrupt (
    PWM_Type * Base,
    Pwm_Hal_ChannelType Channel,
    boolean Enable )
```

Definition at line 222 of file AC784xx_Pwm_Reg.h.

4.2.2.15 Pwm_Reg_EnableChannelMatchTrigger()

```
LOCAL_INLINE void Pwm_Reg_EnableChannelMatchTrigger (
    PWM_Type * Base,
    Pwm_Hal_ChannelType Channel,
    boolean State )
```

Definition at line 360 of file AC784xx_Pwm_Reg.h.

4.2.2.16 Pwm_Reg_EnableChannelOutputMask()

```
LOCAL_INLINE void Pwm_Reg_EnableChannelOutputMask (
    PWM_Type * Base,
    Pwm_Hal_ChannelType Channel,
    boolean Enable )
```

Definition at line 388 of file AC784xx_Pwm_Reg.h.

4.2.2.17 Pwm_Reg_EnableChannelSoftControl()

```
LOCAL_INLINE void Pwm_Reg_EnableChannelSoftControl (
    PWM_Type * Base,
    Pwm_Hal_ChannelType Channel,
    boolean State )
```

Definition at line 480 of file AC784xx_Pwm_Reg.h.

4.2.2.18 Pwm_Reg_EnableCHOSWCERSync()

```
LOCAL_INLINE void Pwm_Reg_EnableCHOSWCERSync (
    PWM_Type * Base,
    boolean Enable )
```

Definition at line 597 of file AC784xx_Pwm_Reg.h.

4.2.2.19 Pwm_Reg_EnableCNTINSync()

```
LOCAL_INLINE void Pwm_Reg_EnableCNTINSync (
    PWM_Type * Base,
    boolean Enable )
```

Definition at line 587 of file AC784xx_Pwm_Reg.h.

4.2.2.20 Pwm_Reg_EnableDmaTransfer()

```
LOCAL_INLINE void Pwm_Reg_EnableDmaTransfer (
    PWM_Type * Base,
    boolean Enable )
```

Definition at line 138 of file AC784xx_Pwm_Reg.h.

4.2.2.21 Pwm_Reg_EnableFaultHizOutput()

```
LOCAL_INLINE void Pwm_Reg_EnableFaultHizOutput (
    PWM_Type * Base,
    boolean Enable )
```

Definition at line 794 of file AC784xx_Pwm_Reg.h.

4.2.2.22 Pwm_Reg_EnableFaultInputFilter()

```
LOCAL_INLINE void Pwm_Reg_EnableFaultInputFilter (
    PWM_Type * Base,
    Pwm_Hal_FaultInputIdType FaultInputId,
    boolean Enable )
```

Definition at line 766 of file AC784xx_Pwm_Reg.h.

4.2.2.23 Pwm_Reg_EnableFaultInterrupt()

```
LOCAL_INLINE void Pwm_Reg_EnableFaultInterrupt (
    PWM_Type * Base,
    boolean Enable )
```

Definition at line 789 of file AC784xx_Pwm_Reg.h.

4.2.2.24 Pwm_Reg_EnableFaultPinInput()

```
LOCAL_INLINE void Pwm_Reg_EnableFaultPinInput (
    PWM_Type * Base,
    Pwm_Hal_FaultInputIdType FaultInputId,
    boolean Enable )
```

Definition at line 777 of file AC784xx_Pwm_Reg.h.

4.2.2.25 Pwm_Reg_EnableGlobalTimeBase()

```
LOCAL_INLINE void Pwm_Reg_EnableGlobalTimeBase (
    PWM_Type * Base,
    boolean Enable )
```

Definition at line 674 of file AC784xx_Pwm_Reg.h.

4.2.2.26 Pwm_Reg_EnableGlobalTimeBaseOutput()

```
LOCAL_INLINE void Pwm_Reg_EnableGlobalTimeBaseOutput (
    PWM_Type * Base,
    boolean Enable )
```

Definition at line 679 of file AC784xx_Pwm_Reg.h.

4.2.2.27 Pwm_Reg_EnableHall()

```
LOCAL_INLINE void Pwm_Reg_EnableHall (
    PWM_Type * Base,
    boolean State )
```

Definition at line 337 of file AC784xx_Pwm_Reg.h.

4.2.2.28 Pwm_Reg_EnableInitTrigger()

```
LOCAL_INLINE void Pwm_Reg_EnableInitTrigger (
    PWM_Type * Base,
    boolean State )
```

Definition at line 350 of file AC784xx_Pwm_Reg.h.

4.2.2.29 Pwm_Reg_EnableINVCRSync()

```
LOCAL_INLINE void Pwm_Reg_EnableINVCRSync (
    PWM_Type * Base,
    boolean Enable )
```

Definition at line 592 of file AC784xx_Pwm_Reg.h.

4.2.2.30 Pwm_Reg_EnableMaxLoadingPointSync()

```
LOCAL_INLINE void Pwm_Reg_EnableMaxLoadingPointSync (
    PWM_Type * Base,
    boolean Enable )
```

Definition at line 557 of file AC784xx_Pwm_Reg.h.

4.2.2.31 Pwm_Reg_EnableMaxTrigger()

```
LOCAL_INLINE void Pwm_Reg_EnableMaxTrigger (
    PWM_Type * Base,
    boolean State )
```

Definition at line 355 of file AC784xx_Pwm_Reg.h.

4.2.2.32 Pwm_Reg_EnableMinLoadingPointSync()

```
LOCAL_INLINE void Pwm_Reg_EnableMinLoadingPointSync (
    PWM_Type * Base,
    boolean Enable )
```

Definition at line 562 of file AC784xx_Pwm_Reg.h.

4.2.2.33 Pwm_Reg_EnableOutputInit()

```
LOCAL_INLINE void Pwm_Reg_EnableOutputInit (
    PWM_Type * Base,
    boolean Enable )
```

Definition at line 148 of file AC784xx_Pwm_Reg.h.

4.2.2.34 Pwm_Reg_EnableOutputMaskSync()

```
LOCAL_INLINE void Pwm_Reg_EnableOutputMaskSync (
    PWM_Type * Base,
    boolean Enable )
```

Definition at line 572 of file AC784xx_Pwm_Reg.h.

4.2.2.35 Pwm_Reg_EnableOverflowDmaReq()

```
LOCAL_INLINE void Pwm_Reg_EnableOverflowDmaReq (
    PWM_Type * Base,
    boolean Enable )
```

Definition at line 133 of file AC784xx_Pwm_Reg.h.

4.2.2.36 Pwm_Reg_EnableOverflowEvent()

```
LOCAL_INLINE void Pwm_Reg_EnableOverflowEvent (
    PWM_Type * Base,
    boolean Enable )
```

Definition at line 227 of file AC784xx_Pwm_Reg.h.

4.2.2.37 Pwm_Reg_EnableOverflowInterrupt()

```
LOCAL_INLINE void Pwm_Reg_EnableOverflowInterrupt (
    PWM_Type * Base,
    boolean Enable )
```

Definition at line 217 of file AC784xx_Pwm_Reg.h.

4.2.2.38 Pwm_Reg_EnablePairChComplement()

```
LOCAL_INLINE void Pwm_Reg_EnablePairChComplement (
    PWM_Type * Base,
    Pwm_Hal_ChannelPairType ChannelPair,
    boolean Enable )
```

Definition at line 260 of file AC784xx_Pwm_Reg.h.

4.2.2.39 Pwm_Reg_EnablePairChDeadtime()

```
LOCAL_INLINE void Pwm_Reg_EnablePairChDeadtime (
    PWM_Type * Base,
    Pwm_Hal_ChannelPairType ChannelPair,
    boolean Enable )
```

Definition at line 274 of file AC784xx_Pwm_Reg.h.

4.2.2.40 Pwm_Reg_EnablePairChDualEdgeCapture()

```
LOCAL_INLINE void Pwm_Reg_EnablePairChDualEdgeCapture (
    PWM_Type * Base,
    Pwm_Hal_ChannelPairType ChannelPair,
    boolean State )
```

Definition at line 723 of file AC784xx_Pwm_Reg.h.

4.2.2.41 Pwm_Reg_EnablePairChFaultCtrl()

```
LOCAL_INLINE void Pwm_Reg_EnablePairChFaultCtrl (
    PWM_Type * Base,
    Pwm_Hal_ChannelPairType ChannelPair,
    boolean Enable )
```

Definition at line 812 of file AC784xx_Pwm_Reg.h.

4.2.2.42 Pwm_Reg_EnablePairChInvert()

```
LOCAL_INLINE void Pwm_Reg_EnablePairChInvert (
    PWM_Type * Base,
    Pwm_Hal_ChannelPairType ChannelPair,
    boolean Enable )
```

Definition at line 313 of file AC784xx_Pwm_Reg.h.

4.2.2.43 Pwm_Reg_EnablePairChSymmetric()

```
LOCAL_INLINE void Pwm_Reg_EnablePairChSymmetric (
    PWM_Type * Base,
    Pwm_Hal_ChannelPairType ChannelPair,
    boolean Enable )
```

Definition at line 253 of file AC784xx_Pwm_Reg.h.

4.2.2.44 Pwm_Reg_EnablePairChValueSync()

```
LOCAL_INLINE void Pwm_Reg_EnablePairChValueSync (
    PWM_Type * Base,
    Pwm_Hal_ChannelPairType ChannelPair,
    boolean Enable )
```

Definition at line 267 of file AC784xx_Pwm_Reg.h.

4.2.2.45 Pwm_Reg_EnablePairPulseWidthMeasurement()

```
LOCAL_INLINE void Pwm_Reg_EnablePairPulseWidthMeasurement (
    PWM_Type * Base,
    Pwm_Hal_ChannelPairType ChannelPair,
    boolean State )
```

Definition at line 330 of file AC784xx_Pwm_Reg.h.

4.2.2.46 Pwm_Reg_EnablePOLSync()

```
LOCAL_INLINE void Pwm_Reg_EnablePOLSync (
    PWM_Type * Base,
    boolean Enable )
```

Definition at line 567 of file AC784xx_Pwm_Reg.h.

4.2.2.47 Pwm_Reg_EnableQuadDecoder()

```
LOCAL_INLINE void Pwm_Reg_EnableQuadDecoder (
    PWM_Type * Base,
    boolean Enable )
```

Definition at line 892 of file AC784xx_Pwm_Reg.h.

4.2.2.48 Pwm_Reg_EnableQuadPhaseZResetCnt()

```
LOCAL_INLINE void Pwm_Reg_EnableQuadPhaseZResetCnt (
    PWM_Type * Base,
    boolean Enable )
```

Definition at line 887 of file AC784xx_Pwm_Reg.h.

4.2.2.49 Pwm_Reg_EnableSync()

```
LOCAL_INLINE void Pwm_Reg_EnableSync (
    PWM_Type * Base,
    boolean State )
```

Definition at line 526 of file AC784xx_Pwm_Reg.h.

4.2.2.50 Pwm_Reg_EnableSyncBYP()

```
LOCAL_INLINE void Pwm_Reg_EnableSyncBYP (
    PWM_Type * Base,
    boolean Enable )
```

Definition at line 541 of file AC784xx_Pwm_Reg.h.

4.2.2.51 Pwm_Reg_EnableSyncHwTriggerSrc()

```
LOCAL_INLINE void Pwm_Reg_EnableSyncHwTriggerSrc (
    PWM_Type * Base,
    uint8 HwTrigNum,
    boolean Enable )
```

Definition at line 546 of file AC784xx_Pwm_Reg.h.

4.2.2.52 Pwm_Reg_EnableUnderflowEvent()

```
LOCAL_INLINE void Pwm_Reg_EnableUnderflowEvent (
    PWM_Type * Base,
    boolean Enable )
```

Definition at line 232 of file AC784xx_Pwm_Reg.h.

4.2.2.53 Pwm_Reg_EnableWriteProtection()

```
LOCAL_INLINE void Pwm_Reg_EnableWriteProtection (
    PWM_Type * Base,
    boolean Enable )
```

Definition at line 942 of file AC784xx_Pwm_Reg.h.

4.2.2.54 Pwm_Reg_GetAllChannelLevel()

```
LOCAL_INLINE uint32 Pwm_Reg_GetAllChannelLevel (
    const PWM_Type * Base )
```

Definition at line 411 of file AC784xx_Pwm_Reg.h.

4.2.2.55 Pwm_Reg_GetBase()

```
LOCAL_INLINE PWM_Type* Pwm_Reg_GetBase (
    Pwm_Hal_InstanceType Instance )
```

Definition at line 54 of file AC784xx_Pwm_Reg.h.

4.2.2.56 Pwm_Reg_GetChannelEventFlag()

```
LOCAL_INLINE uint32 Pwm_Reg_GetChannelEventFlag (
    const PWM_Type * Base )
```

Definition at line 445 of file AC784xx_Pwm_Reg.h.

4.2.2.57 Pwm_Reg_GetChannelInterruptFlag()

```
LOCAL_INLINE uint32 Pwm_Reg_GetChannelInterruptFlag (
    const PWM_Type * Base,
    Pwm_Hal_ChannelType Channel )
```

Definition at line 462 of file AC784xx_Pwm_Reg.h.

4.2.2.58 Pwm_Reg_GetChannelTriggerFlag()

```
LOCAL_INLINE uint32 Pwm_Reg_GetChannelTriggerFlag (
    const PWM_Type * Base )
```

Definition at line 371 of file AC784xx_Pwm_Reg.h.

4.2.2.59 Pwm_Reg_GetChannelValue()

```
LOCAL_INLINE uint16 Pwm_Reg_GetChannelValue (
    const PWM_Type * Base,
    Pwm_Hal_ChannelType Channel )
```

Definition at line 179 of file AC784xx_Pwm_Reg.h.

4.2.2.60 Pwm_Reg_GetCountValue()

```
LOCAL_INLINE uint16 Pwm_Reg_GetCountValue (
    const PWM_Type * Base )
```

Definition at line 114 of file AC784xx_Pwm_Reg.h.

4.2.2.61 Pwm_Reg_GetFaultFlag()

```
LOCAL_INLINE uint32 Pwm_Reg_GetFaultFlag (
    const PWM_Type * Base )
```

Definition at line 840 of file AC784xx_Pwm_Reg.h.

4.2.2.62 Pwm_Reg_GetFaultInputStatus()

```
LOCAL_INLINE uint32 Pwm_Reg_GetFaultInputStatus (
    const PWM_Type * Base )
```

Definition at line 859 of file AC784xx_Pwm_Reg.h.

4.2.2.63 Pwm_Reg_GetFaultPinFlag()

```
LOCAL_INLINE uint32 Pwm_Reg_GetFaultPinFlag (
    const PWM_Type * Base,
    Pwm_Hal_FaultInputIdType FaultInputId )
```

Definition at line 819 of file AC784xx_Pwm_Reg.h.

4.2.2.64 Pwm_Reg_GetHallStatus()

```
LOCAL_INLINE uint32 Pwm_Reg_GetHallStatus (
    const PWM_Type * Base )
```

Definition at line 342 of file AC784xx_Pwm_Reg.h.

4.2.2.65 Pwm_Reg_GetInitCountValue()

```
LOCAL_INLINE uint16 Pwm_Reg_GetInitCountValue (
    const PWM_Type * Base )
```

Definition at line 101 of file AC784xx_Pwm_Reg.h.

4.2.2.66 Pwm_Reg_GetMaxCountValue()

```
LOCAL_INLINE uint16 Pwm_Reg_GetMaxCountValue (
    const PWM_Type * Base )
```

Definition at line 89 of file AC784xx_Pwm_Reg.h.

4.2.2.67 Pwm_Reg_GetOutputMask()

```
LOCAL_INLINE uint32 Pwm_Reg_GetOutputMask (
    const PWM_Type * Base )
```

Definition at line 398 of file AC784xx_Pwm_Reg.h.

4.2.2.68 Pwm_Reg_GetOverflowDir()

```
LOCAL_INLINE uint32 Pwm_Reg_GetOverflowDir (
    const PWM_Type * Base )
```

Definition at line 437 of file AC784xx_Pwm_Reg.h.

4.2.2.69 Pwm_Reg_GetOverflowFlag()

```
LOCAL_INLINE uint32 Pwm_Reg_GetOverflowFlag (
    const PWM_Type * Base )
```

Definition at line 419 of file AC784xx_Pwm_Reg.h.

4.2.2.70 Pwm_Reg_GetQuadCountDir()

```
LOCAL_INLINE uint32 Pwm_Reg_GetQuadCountDir (
    const PWM_Type * Base )
```

Definition at line 897 of file AC784xx_Pwm_Reg.h.

4.2.2.71 Pwm_Reg_GetQuadOverflowDir()

```
LOCAL_INLINE uint32 Pwm_Reg_GetQuadOverflowDir (
    const PWM_Type * Base )
```

Definition at line 905 of file AC784xx_Pwm_Reg.h.

4.2.2.72 Pwm_Reg_GetQuadPhaseZFlag()

```
LOCAL_INLINE uint32 Pwm_Reg_GetQuadPhaseZFlag (
    const PWM_Type * Base )
```

Definition at line 913 of file AC784xx_Pwm_Reg.h.

4.2.2.73 Pwm_Reg_IsPairChDualEdgeCaptureEnabled()

```
LOCAL_INLINE boolean Pwm_Reg_IsPairChDualEdgeCaptureEnabled (
    const PWM_Type * Base,
    Pwm_Hal_ChannelPairType ChannelPair )
```

Definition at line 749 of file AC784xx_Pwm_Reg.h.

4.2.2.74 Pwm_Reg_ReadChannelSoftControlState()

```
LOCAL_INLINE uint32 Pwm_Reg_ReadChannelSoftControlState (
    const PWM_Type * Base )
```

Definition at line 499 of file AC784xx_Pwm_Reg.h.

4.2.2.75 Pwm_Reg_ResetCounter()

```
LOCAL_INLINE void Pwm_Reg_ResetCounter (
    PWM_Type * Base )
```

Definition at line 108 of file AC784xx_Pwm_Reg.h.

4.2.2.76 Pwm_Reg_SetCaptureEventPsc()

```
LOCAL_INLINE void Pwm_Reg_SetCaptureEventPsc (
    PWM_Type * Base,
    Pwm_Hal_ChannelType Channel,
    Pwm_Hal_InputEventPscType EventPsc )
```

Definition at line 697 of file AC784xx_Pwm_Reg.h.

4.2.2.77 Pwm_Reg_SetChannelELSR()

```
LOCAL_INLINE void Pwm_Reg_SetChannelELSR (
    PWM_Type * Base,
    Pwm_Hal_ChannelType Channel,
    uint8 Value )
```

Definition at line 166 of file AC784xx_Pwm_Reg.h.

4.2.2.78 Pwm_Reg_SetChannelInputFilterVal()

```
LOCAL_INLINE void Pwm_Reg_SetChannelInputFilterVal (
    PWM_Type * Base,
    Pwm_Hal_ChannelType Channel,
    uint8 Value )
```

Definition at line 689 of file AC784xx_Pwm_Reg.h.

4.2.2.79 Pwm_Reg_SetChannelMatchDir()

```
LOCAL_INLINE void Pwm_Reg_SetChannelMatchDir (
    PWM_Type * Base,
    Pwm_Hal_ChannelType Channel,
    Pwm_Hal_ChannelMatchDirType Dir )
```

Definition at line 324 of file AC784xx_Pwm_Reg.h.

4.2.2.80 Pwm_Reg_SetChannelMatchDither()

```
LOCAL_INLINE void Pwm_Reg_SetChannelMatchDither (
    PWM_Type * Base,
    Pwm_Hal_ChannelType Channel,
    uint8 DitherValue )
```

Definition at line 198 of file AC784xx_Pwm_Reg.h.

4.2.2.81 Pwm_Reg_SetChannelMatchDitherReg()

```
LOCAL_INLINE void Pwm_Reg_SetChannelMatchDitherReg (
    PWM_Type * Base,
    uint8 DitherRegNum,
    uint32 RegValue )
```

Definition at line 186 of file AC784xx_Pwm_Reg.h.

4.2.2.82 Pwm_Reg_SetChannelMSR()

```
LOCAL_INLINE void Pwm_Reg_SetChannelMSR (
    PWM_Type * Base,
    Pwm_Hal_ChannelType Channel,
    uint8 Value )
```

Definition at line 158 of file AC784xx_Pwm_Reg.h.

4.2.2.83 Pwm_Reg_SetChannelOutputInitLevel()

```
LOCAL_INLINE void Pwm_Reg_SetChannelOutputInitLevel (
    PWM_Type * Base,
    Pwm_Hal_ChannelType Channel,
    Pwm_Hal_OutputLevelType Level )
```

Definition at line 238 of file AC784xx_Pwm_Reg.h.

4.2.2.84 Pwm_Reg_SetChannelPolarity()

```
LOCAL_INLINE void Pwm_Reg_SetChannelPolarity (
    PWM_Type * Base,
    Pwm_Hal_ChannelType Channel,
    Pwm_Hal_ActivePolarityType Polarity )
```

Definition at line 405 of file AC784xx_Pwm_Reg.h.

4.2.2.85 Pwm_Reg_SetChannelSoftControlLevel()

```
LOCAL_INLINE void Pwm_Reg_SetChannelSoftControlLevel (
    PWM_Type * Base,
    Pwm_Hal_ChannelType Channel,
    Pwm_Hal_OutputLevelType Level )
```

Definition at line 509 of file AC784xx_Pwm_Reg.h.

4.2.2.86 Pwm_Reg_SetChannelValue()

```
LOCAL_INLINE void Pwm_Reg_SetChannelValue (
    PWM_Type * Base,
    Pwm_Hal_ChannelType Channel,
    uint16 Value )
```

Definition at line 174 of file AC784xx_Pwm_Reg.h.

4.2.2.87 Pwm_Reg_SetCHOSWCRSyncTriggerMode()

```
LOCAL_INLINE void Pwm_Reg_SetCHOSWCRSyncTriggerMode (
    PWM_Type * Base,
    Pwm_Hal_SyncTriggerMethodType Mode,
    boolean Enable )
```

Definition at line 614 of file AC784xx_Pwm_Reg.h.

4.2.2.88 Pwm_Reg_SetClockPsc()

```
LOCAL_INLINE void Pwm_Reg_SetClockPsc (
    PWM_Type * Base,
    uint16 ClkPsc )
```

Definition at line 74 of file AC784xx_Pwm_Reg.h.

4.2.2.89 Pwm_Reg_SetClockSource()

```
LOCAL_INLINE void Pwm_Reg_SetClockSource (
    PWM_Type * Base,
    Pwm_Hal_ClockSourceType ClkSource )
```

Definition at line 69 of file AC784xx_Pwm_Reg.h.

4.2.2.90 Pwm_Reg_SetCntOverflowFreq()

```
LOCAL_INLINE void Pwm_Reg_SetCntOverflowFreq (
    PWM_Type * Base,
    uint8 Freq )
```

Definition at line 128 of file AC784xx_Pwm_Reg.h.

4.2.2.91 Pwm_Reg_SetCNTSyncTriggerMode()

```
LOCAL_INLINE void Pwm_Reg_SetCNTSyncTriggerMode (
    PWM_Type * Base,
    Pwm_Hal_SyncTriggerMethodType Mode,
    boolean Enable )
```

Definition at line 662 of file AC784xx_Pwm_Reg.h.

4.2.2.92 Pwm_Reg_SetCombineCenterDutyType()

```
LOCAL_INLINE void Pwm_Reg_SetCombineCenterDutyType (
    PWM_Type * Base,
    Pwm_Hal_CombineCenterDutyModeType DutyMode )
```

Definition at line 319 of file AC784xx_Pwm_Reg.h.

4.2.2.93 Pwm_Reg_SetCountMode()

```
LOCAL_INLINE void Pwm_Reg_SetCountMode (
    PWM_Type * Base,
    Pwm_Hal_CountModeType Mode )
```

Definition at line 79 of file AC784xx_Pwm_Reg.h.

4.2.2.94 Pwm_Reg_SetDebugMode()

```
LOCAL_INLINE void Pwm_Reg_SetDebugMode (
    PWM_Type * Base,
    Pwm_Hal_DebugModeType Mode )
```

Definition at line 960 of file AC784xx_Pwm_Reg.h.

4.2.2.95 Pwm_Reg_SetDmaTransferLen()

```
LOCAL_INLINE void Pwm_Reg_SetDmaTransferLen (
    PWM_Type * Base,
    uint8 Len )
```

Definition at line 143 of file AC784xx_Pwm_Reg.h.

4.2.2.96 Pwm_Reg_SetFaultInputFilterValue()

```
LOCAL_INLINE void Pwm_Reg_SetFaultInputFilterValue (
    PWM_Type * Base,
    uint8 Value )
```

Definition at line 772 of file AC784xx_Pwm_Reg.h.

4.2.2.97 Pwm_Reg_SetFaultInputPolarity()

```
LOCAL_INLINE void Pwm_Reg_SetFaultInputPolarity (
    PWM_Type * Base,
    Pwm_Hal_FaultInputIdType FaultInputId,
    Pwm_Hal_ActivePolarityType Polarity )
```

Definition at line 783 of file AC784xx_Pwm_Reg.h.

4.2.2.98 Pwm_Reg_SetFaultMode()

```
LOCAL_INLINE void Pwm_Reg_SetFaultMode (
    PWM_Type * Base,
    Pwm_Hal_FaultCtrlModeType Mode )
```

Definition at line 761 of file AC784xx_Pwm_Reg.h.

4.2.2.99 Pwm_Reg_SetInitCountValue()

```
LOCAL_INLINE void Pwm_Reg_SetInitCountValue (
    PWM_Type * Base,
    uint16 Value )
```

Definition at line 96 of file AC784xx_Pwm_Reg.h.

4.2.2.100 Pwm_Reg_SetInputFilterPsc()

```
LOCAL_INLINE void Pwm_Reg_SetInputFilterPsc (
    PWM_Type * Base,
    Pwm_Hal_InputFilterPscType Psc )
```

Definition at line 684 of file AC784xx_Pwm_Reg.h.

4.2.2.101 Pwm_Reg_SetINVCRSyncTriggerMode()

```
LOCAL_INLINE void Pwm_Reg_SetINVCRSyncTriggerMode (
    PWM_Type * Base,
    Pwm_Hal_SyncTriggerMethodType Mode,
    boolean Enable )
```

Definition at line 626 of file AC784xx_Pwm_Reg.h.

4.2.2.102 Pwm_Reg_SetMaxCountValue()

```
LOCAL_INLINE void Pwm_Reg_SetMaxCountValue (
    PWM_Type * Base,
    uint16 Value )
```

Definition at line 84 of file AC784xx_Pwm_Reg.h.

4.2.2.103 Pwm_Reg_SetMCVRSyncMode()

```
LOCAL_INLINE void Pwm_Reg_SetMCVRSyncMode (
    PWM_Type * Base,
    Pwm_Hal_SyncTriggerMethodType Mode,
    boolean Enable )
```

Definition at line 650 of file AC784xx_Pwm_Reg.h.

4.2.2.104 Pwm_Reg_SetOMCRSyncTriggerMode()

```
LOCAL_INLINE void Pwm_Reg_SetOMCRSyncTriggerMode (
    PWM_Type * Base,
    Pwm_Hal_SyncTriggerMethodType Mode,
    boolean Enable )
```

Definition at line 638 of file AC784xx_Pwm_Reg.h.

4.2.2.105 Pwm_Reg_SetOutputMask()

```
LOCAL_INLINE void Pwm_Reg_SetOutputMask (
    PWM_Type * Base,
    uint8 Mask )
```

Definition at line 393 of file AC784xx_Pwm_Reg.h.

4.2.2.106 Pwm_Reg_SetPairChDeadtime()

```
LOCAL_INLINE void Pwm_Reg_SetPairChDeadtime (
    PWM_Type * Base,
    Pwm_Hal_ChannelPairType ChannelPair,
    Pwm_Hal_DeadTimePscType Psc,
    uint16 Value )
```

Definition at line 282 of file AC784xx_Pwm_Reg.h.

4.2.2.107 Pwm_Reg_SetPairChDualEdgeCapture()

```
LOCAL_INLINE void Pwm_Reg_SetPairChDualEdgeCapture (
    PWM_Type * Base,
    Pwm_Hal_ChannelPairType ChannelPair,
    boolean State )
```

Definition at line 736 of file AC784xx_Pwm_Reg.h.

4.2.2.108 Pwm_Reg_SetPeriodDither()

```
LOCAL_INLINE void Pwm_Reg_SetPeriodDither (
    PWM_Type * Base,
    uint8 Value )
```

Definition at line 123 of file AC784xx_Pwm_Reg.h.

4.2.2.109 Pwm_Reg_SetPOLTriggerMode()

```
LOCAL_INLINE void Pwm_Reg_SetPOLTriggerMode (
    PWM_Type * Base,
    Pwm_Hal_SyncTriggerMethodType Mode,
    boolean Enable )
```

Definition at line 602 of file AC784xx_Pwm_Reg.h.

4.2.2.110 Pwm_Reg_SetQuadEncodeMode()

```
LOCAL_INLINE void Pwm_Reg_SetQuadEncodeMode (
    PWM_Type * Base,
    Pwm_Hal_QuadModeType Mode )
```

Definition at line 867 of file AC784xx_Pwm_Reg.h.

4.2.2.111 Pwm_Reg_SetQuadPhaseAPolarity()

```
LOCAL_INLINE void Pwm_Reg_SetQuadPhaseAPolarity (
    PWM_Type * Base,
    Pwm_Hal_QuadPhasePolarityType Polarity )
```

Definition at line 872 of file AC784xx_Pwm_Reg.h.

4.2.2.112 Pwm_Reg_SetQuadPhaseBPolarity()

```
LOCAL_INLINE void Pwm_Reg_SetQuadPhaseBPolarity (
    PWM_Type * Base,
    Pwm_Hal_QuadPhasePolarityType Polarity )
```

Definition at line 877 of file AC784xx_Pwm_Reg.h.

4.2.2.113 Pwm_Reg_SetQuadPhaseZPolarity()

```
LOCAL_INLINE void Pwm_Reg_SetQuadPhaseZPolarity (
    PWM_Type * Base,
    Pwm_Hal_QuadPhasePolarityType Polarity )
```

Definition at line 882 of file AC784xx_Pwm_Reg.h.

4.2.2.114 Pwm_Reg_SetSoftControlEnableStatesMask()

```
LOCAL_INLINE void Pwm_Reg_SetSoftControlEnableStatesMask (
    PWM_Type * Base,
    uint8 Mask )
```

Definition at line 490 of file AC784xx_Pwm_Reg.h.

4.2.2.115 Pwm_Reg_SetSoftControlLevelMask()

```
LOCAL_INLINE void Pwm_Reg_SetSoftControlLevelMask (
    PWM_Type * Base,
    uint8 Mask )
```

Definition at line 521 of file AC784xx_Pwm_Reg.h.

4.2.2.116 Pwm_Reg_SetSyncMode()

```
LOCAL_INLINE void Pwm_Reg_SetSyncMode (
    PWM_Type * Base,
    Pwm_Hal_SyncModeType Mode )
```

Definition at line 536 of file AC784xx_Pwm_Reg.h.

4.2.2.117 Pwm_Reg_SetSyncType()

```
LOCAL_INLINE void Pwm_Reg_SetSyncType (
    PWM_Type * Base,
    uint8 Type )
```

Definition at line 531 of file AC784xx_Pwm_Reg.h.

4.2.2.118 Pwm_Reg_SetTriggerRatio()

```
LOCAL_INLINE void Pwm_Reg_SetTriggerRatio (
    PWM_Type * Base,
    uint8 Ratio )
```

Definition at line 365 of file AC784xx_Pwm_Reg.h.

4.2.2.119 Pwm_Reg_TrigSoftwareSync()

```
LOCAL_INLINE void Pwm_Reg_TrigSoftwareSync (
    PWM_Type * Base,
    boolean State )
```

Definition at line 552 of file AC784xx_Pwm_Reg.h.

4.3 Pwm_Hal.c File Reference

```
#include "AC784xx_Pwm_Reg.h"
#include "Ckgen_Hal.h"
#include "Pwm_Hal.h"
#include "Rcm_Hal.h"
#include "Core_Hal.h"
```

Functions

- void [Pwm_Hal_InitOutputMode](#) ([Pwm_Hal_InstanceType](#) Instance, const [Pwm_Hal_OutputCfg](#) *Config)
 - : *Pwm_Hal_InitOutputMode: init pwm output mode*
- void [Pwm_Hal_InitInputMode](#) ([Pwm_Hal_InstanceType](#) Instance, const [Pwm_Hal_InputCfg](#) *Config)
 - : *Pwm_Hal_InitInputMode: init pwm input mode*
- void [Pwm_Hal_InitQuadDecoderMode](#) ([Pwm_Hal_InstanceType](#) Instance, const [Pwm_Hal_QuadDecoderCfg](#) *Config)
 - : *Pwm_Hal_InitQuadDecoderMode: init pwm quadrature decoder mode*
- void [Pwm_Hal_DeInit](#) ([Pwm_Hal_InstanceType](#) Instance)
 - : *Pwm_Hal_DeInit: deinit pwm module*
- void [Pwm_Hal_SetClockSource](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ClockSourceType](#) ClkSource, uint16 ClkPsc)
 - : *Pwm_Hal_SetClockSource: Set pwm module clock source and clock psc*

- void [Pwm_Hal_ResetCounter](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_ResetCounter: reset module counter value*
- uint16 [Pwm_Hal_GetCountValue](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_GetCountValue: get module counter value*
- void [Pwm_Hal_SetMaxCountValue](#) ([Pwm_Hal_InstanceType](#) Instance, uint16 Value)
: *Pwm_Hal_SetMaxCountValue: Set pwm module counter max value.*
- uint16 [Pwm_Hal_GetMaxCountValue](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_GetMaxCountValue: Get pwm module counter max value.*
- void [Pwm_Hal_SetInitCountValue](#) ([Pwm_Hal_InstanceType](#) Instance, uint16 Value)
: *Pwm_Hal_SetInitCountValue: Set pwm module counter begin value.*
- uint16 [Pwm_Hal_GetInitCountValue](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_GetInitCountValue: Get pwm module counter init value.*
- void [Pwm_Hal_SetChannelValue](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, uint16 Value)
: *Pwm_Hal_SetChannelValue: Set channel value.*
- uint16 [Pwm_Hal_GetChannelValue](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel)
: *Pwm_Hal_GetChannelValue: Get channel match value.*
- void [Pwm_Hal_EnableChannelInterrupt](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, boolean Enable)
: *Pwm_Hal_EnableChannelInterrupt: enable or disable pwm channel match Interrupt*
- void [Pwm_Hal_EnableOverflowInterrupt](#) ([Pwm_Hal_InstanceType](#) Instance, boolean Enable)
: *Pwm_Hal_EnableOverflowInterrupt: enable or disable pwm module overflow Interrupt.*
- void [Pwm_Hal_EnableOverflowEvent](#) ([Pwm_Hal_InstanceType](#) Instance, boolean Enable)
: *Pwm_Hal_EnableOverflowEvent: enable or disable pwm module overflow event.*
- void [Pwm_Hal_EnableUnderflowEvent](#) ([Pwm_Hal_InstanceType](#) Instance, boolean Enable)
: *Pwm_Hal_EnableUnderflowEvent: enable or disable pwm module under overflow event.*
- uint32 [Pwm_Hal_GetChannelInterruptFlag](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_GetChannelInterruptFlag: get all pwm interrupt flag*
- void [Pwm_Hal_ClearChannelInterruptFlag](#) ([Pwm_Hal_InstanceType](#) Instance, uint32 Mask)
: *Pwm_Hal_ClearChannelInterruptFlag: clear all channel interrupt flag*
- uint32 [Pwm_Hal_GetOverflowFlag](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_GetOverflowFlag: get all pwm interrupt flag*
- void [Pwm_Hal_ClearOverflowFlag](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_ClearOverFlowFlag: clear pwm OverFlow flag*
- uint32 [Pwm_Hal_GetOverflowDir](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_GetOverFlowDir: get pwm Overflow direction*
- uint32 [Pwm_Hal_GetAllChannelLevel](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_GetAllChannelLevel: get all pwm channel level*
- void [Pwm_Hal_EnableChannelDmaRequest](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, boolean Enable)
: *Pwm_Hal_EnableChannelDmaRequest: Enable DMA requests for channel events*
- void [Pwm_Hal_InitSyncConfigSet](#) ([Pwm_Hal_InstanceType](#) Instance, const [Pwm_Hal_SyncCfg](#) *ConfigPtr)
: *Pwm_Hal_InitSyncConfigSet: PWM synchronization control init*
- void [Pwm_Hal_EnableSyncBypass](#) ([Pwm_Hal_InstanceType](#) Instance, boolean Enable)
: *Pwm_Hal_EnableSyncBypass: CHnV/CNTIN/MCVR sync bypass.*
- void [Pwm_Hal_TrigSoftwareSync](#) ([Pwm_Hal_InstanceType](#) Instance, boolean IsTrigger)
: *Pwm_Hal_TrigSoftwareSync: start software sync trigger*
- void [Pwm_Hal_EnableGlobalTimeBase](#) ([Pwm_Hal_InstanceType](#) Instance, boolean Enable)
: *Pwm_Hal_EnableGlobalTimeBase: Enable/Disable global timer base.*
- void [Pwm_Hal_EnableGlobalTimeBaseOutput](#) ([Pwm_Hal_InstanceType](#) Instance, boolean Enable)
: *Pwm_Hal_EnableGlobalTimeBaseOutput: Global timer base output enable/disable.*
- void [Pwm_Hal_EnableInitTrigger](#) ([Pwm_Hal_InstanceType](#) Instance, boolean Enable)
: *Pwm_Hal_EnableInitTrigger: Enable initialization trigger source.*

- void [Pwm_Hal_EnableMaxTrigger](#) ([Pwm_Hal_InstanceType](#) Instance, boolean Enable)
: *Pwm_Hal_EnableMaxTrigger: Enable max trigger source.*
- void [Pwm_Hal_EnableChannelMatchTrigger](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, boolean Enable)
: *Pwm_Hal_EnableChannelMatchTrigger: Enable Channel match trigger source.*
- void [Pwm_Hal_SetTriggerRatio](#) ([Pwm_Hal_InstanceType](#) Instance, uint8 TriggerRatio)
: *Pwm_Hal_SetTriggerRatio: Set the max/match/init trigger ratio.*
- void [Pwm_Hal_SetChannelCaptureEdge](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, [Pwm_Hal_EdgeType](#) Edge)
: *Pwm_Hal_SetChannelCaptureEdge: set input channel capture edge*
- void [Pwm_Hal_SetChannelInputFilterVal](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, uint8 Value)
: *Pwm_Hal_SetChannelInputFilterVal: Set channel input filter value.*
- void [Pwm_Hal_SetInputFilterPsc](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_InputFilterPscType](#) Prescaler)
: *Pwm_Hal_SetInputFilterPsc: Set pwm input filter psc value.*
- void [Pwm_Hal_EnableChannelEventReset](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, boolean Enable)
: *Pwm_Hal_EnableChannelEventReset: Does the channel event reset the counter.*
- uint32 [Pwm_Hal_GetHallStatus](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_GetHallStatus: Get pwm hall status.*
- void [Pwm_Hal_SetChannelCompareAction](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, [Pwm_Hal_CompareActionType](#) Action)
: *Pwm_Hal_SetChannelCompareAction: Set matching action for output comparison mode*
- void [Pwm_Hal_SetSoftControlEnableStates](#) ([Pwm_Hal_InstanceType](#) Instance, uint8 EnableMask)
: *Pwm_Hal_SetSoftControlEnableStates: Set the software output enable status for each channel.*
- void [Pwm_Hal_EnableChannelSoftControl](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, boolean Enable)
: *Pwm_Hal_EnableChannelSoftControl: Enable channel soft control*
- void [Pwm_Hal_EnableCombineChannelSoftControl](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, boolean Enable)
: *Pwm_Hal_EnableCombineChannelSoftControl: enable soft control function*
- void [Pwm_Hal_SetSoftControlLevels](#) ([Pwm_Hal_InstanceType](#) Instance, uint8 LevelMask)
: *Pwm_Hal_SetSoftControlLevels: Set software output levels for all channels*
- void [Pwm_Hal_SetChannelSoftControlLevel](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, [Pwm_Hal_OutputLevelType](#) Level)
: *Pwm_Hal_SetChannelSoftControlLevel: clear all pwm interrupt flag*
- void [Pwm_Hal_SetChannelMatchDither](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, uint8 DitherValue)
: *Pwm_Hal_SetChannelMatchDither: Set channel dither value.*
- void [Pwm_Hal_SetPeriodDither](#) ([Pwm_Hal_InstanceType](#) Instance, uint8 MaxCountDitherValue)
: *PWM_DRV_SetMaxCountDitherValue: Set mod dither value.*
- void [Pwm_Hal_EnableChannelOutputMask](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, boolean Enable)
: *Pwm_Hal_EnableChannelOutputMask: Shielding channel output or not*
- void [Pwm_Hal_SetOutputMask](#) ([Pwm_Hal_InstanceType](#) Instance, uint8 Mask)
: *Pwm_Hal_SetOutputMask: Set the mask status for all channels*
- uint32 [Pwm_Hal_GetOutputMask](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_GetOutputMask: get pwm moudel channel output status*
- void [Pwm_Hal_SetChannelPolarity](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, [Pwm_Hal_ActivePolarityType](#) Polarity)
: *Pwm_Hal_SetChannelPolarity: Set channel polarity.*
- void [Pwm_Hal_SetDeadtime](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelPairType](#) ChannelPair, [Pwm_Hal_DeadTimePscType](#) Prescaler, uint16 Value)
: *Pwm_Hal_SetDeadtime: Set deadtime prescaler & value.*

- void [Pwm_Hal_EnableChannelPairSymmetric](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelPairType](#) ChannelPair, boolean Enable)
: *Pwm_Hal_EnableChannelPairSymmetric: Set pair channel symmetric.*
- void [Pwm_Hal_EnableChannelPairInvert](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelPairType](#) ChannelPair, boolean Enable)
: *Pwm_Hal_EnableChannelPairInvert: Set channel inverting control.*
- void [Pwm_Hal_InitFaultControl](#) ([Pwm_Hal_InstanceType](#) Instance, const [Pwm_Hal_FaultCfg](#) *ConfigPtr)
: *Pwm_Hal_InitFaultControl: Init fault control. Only applicable to PWM modulation output mode, called after PWM_Init().*
- void [Pwm_Hal_EnableFaultPinInput](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_FaultInputIdType](#) FaultInputId, boolean Enable)
: *Pwm_Hal_EnableFaultPinInput: Enable this fault input pin.*
- void [Pwm_Hal_EnableFaultInputFilter](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_FaultInputIdType](#) FaultInputId, boolean Enable)
: *Pwm_Hal_EnableFaultInputFilter: Enable the filtering function of the fault input pin.*
- void [Pwm_Hal_SetFaultInputPolarity](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_FaultInputIdType](#) FaultInputId, [Pwm_Hal_ActivePolarityType](#) ActivePolarity)
: *Pwm_Hal_SetFaultInputPolarity: Set fault input polarity.*
- uint32 [Pwm_Hal_GetFaultPinFlag](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_FaultInputIdType](#) FaultInputId)
: *Pwm_Hal_GetFaultPinFlag: Get fault pin detection flag.*
- void [Pwm_Hal_ClearFaultPinFlag](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_FaultInputIdType](#) FaultInputId)
: *Pwm_Hal_ClearFaultPinFlag: Clear fault pin detection flag.*
- uint32 [Pwm_Hal_GetFaultFlag](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_GetFaultFlag: Get the OR value of each fault input pin flag.*
- void [Pwm_Hal_ClearFaultFlag](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_ClearFaultFlag: Clear the OR value of each fault input pin flag.*
- void [Pwm_Hal_EnableChannelHizOutput](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, boolean Enable)
: *Pwm_Hal_SetChannelHizOutput: Enable or disable channel high-Z output*
- uint32 [Pwm_Hal_GetQuadCountingDir](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_GetQuadCountingDir: Get the current quadrature decoding count direction.*
- uint32 [Pwm_Hal_GetQuadOverflowDir](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_GetQuadOverflowDir: Get the quadrature timer overflow direction.*
- uint32 [Pwm_Hal_GetQuadPhaseZFlag](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_GetQuadPhaseZFlag: Get the phaseZ Status.*
- void [Pwm_Hal_ClearQuadPhaseZFlag](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_ClearQuadPhaseZFlag: Clear the phaseZ Status.*
- uint32 [Pwm_Hal_GetDetectEventFlag](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_GetDetectEventFlag: Check Z index detect event status.*
- void [Pwm_Hal_ClearDetectEventFlag](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_ClearDetectEventFlag: Clear Z index detect event status.*
- void [Pwm_Hal_EnableWriteProtection](#) ([Pwm_Hal_InstanceType](#) Instance, boolean Enable)
: *Pwm_Hal_EnableWriteProtection: Enable write protection.*
- void [Pwm_Hal_SetDebugMode](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_DebugModeType](#) Mode)
: *Pwm_Hal_SetDebugMode: Set debug mode.*
- void [PWM0_Overflow_IRQHandler](#) (void)
: *PWM0_Overflow_IRQHandler: PWM0 overflow Interrupt handle*
- void [PWM1_Overflow_IRQHandler](#) (void)
: *PWM1_Overflow_IRQHandler: PWM1 overflow Interrupt handle*
- void [PWM2_Overflow_IRQHandler](#) (void)
: *PWM2_Overflow_IRQHandler: PWM2 overflow Interrupt handle*
- void [PWM3_Overflow_IRQHandler](#) (void)
: *PWM3_Overflow_IRQHandler: PWM3 overflow Interrupt handle*
- void [PWM4_Overflow_IRQHandler](#) (void)

- *: PWM4_Overflow_IRQHandler: PWM4 overflow Interrupt handle*
- void [PWM5_Overflow_IRQHandler](#) (void)
 - : PWM5_Overflow_IRQHandler: PWM5 overflow Interrupt handle*
- void [PWM0_Channel_IRQHandler](#) (void)
 - : PWM0_Channel_IRQHandler: PWM0 channel Interrupt handle*
- void [PWM1_Channel_IRQHandler](#) (void)
 - : PWM1_Channel_IRQHandler: PWM1 channel Interrupt handle*
- void [PWM2_Channel_IRQHandler](#) (void)
 - : PWM2_Channel_IRQHandler: PWM2 channel Interrupt handle*
- void [PWM3_Channel_IRQHandler](#) (void)
 - : PWM3_Channel_IRQHandler: PWM3 channel Interrupt handle*
- void [PWM4_Channel_IRQHandler](#) (void)
 - : PWM4_Channel_IRQHandler: PWM4 channel Interrupt handle*
- void [PWM5_Channel_IRQHandler](#) (void)
 - : PWM5_Channel_IRQHandler: PWM5 channel Interrupt handle*
- void [PWM0_Fault_IRQHandler](#) (void)
 - : PWM0_Fault_IRQHandler: PWM0 fault Interrupt handle*
- void [PWM1_Fault_IRQHandler](#) (void)
 - : PWM1_Fault_IRQHandler: PWM1 fault Interrupt handle*
- void [PWM2_Fault_IRQHandler](#) (void)
 - : PWM2_Fault_IRQHandler: PWM2 fault Interrupt handle*
- void [PWM3_Fault_IRQHandler](#) (void)
 - : PWM3_Fault_IRQHandler: PWM3 fault Interrupt handle*
- void [PWM4_Fault_IRQHandler](#) (void)
 - : PWM4_Fault_IRQHandler: PWM4 fault Interrupt handle*
- void [PWM5_Fault_IRQHandler](#) (void)
 - : PWM5_Fault_IRQHandler: PWM5 fault Interrupt handle*

4.3.1 Function Documentation

4.3.1.1 PWM0_Channel_IRQHandler()

```
void PWM0_Channel_IRQHandler (
    void )
```

: PWM0_Channel_IRQHandler: PWM0 channel Interrupt handle

Note

Function ID:

Returns

: void

Definition at line 2246 of file Pwm_Hal.c.

4.3.1.2 PWM0_Fault_IRQHandler()

```
void PWM0_Fault_IRQHandler (  
    void )
```

: PWM0_Fault_IRQHandler: PWM0 fault Interrupt handle

Note

Function ID:

Returns

: void

Definition at line 2329 of file Pwm_Hal.c.

4.3.1.3 PWM0_Overflow_IRQHandler()

```
void PWM0_Overflow_IRQHandler (  
    void )
```

: PWM0_Overflow_IRQHandler: PWM0 overflow Interrupt handle

Note

Function ID:

Returns

: void

Definition at line 2164 of file Pwm_Hal.c.

4.3.1.4 PWM1_Channel_IRQHandler()

```
void PWM1_Channel_IRQHandler (  
    void )
```

: PWM1_Channel_IRQHandler: PWM1 channel Interrupt handle

Note

Function ID:

Returns

: void

Definition at line 2256 of file Pwm_Hal.c.

4.3.1.5 PWM1_Fault_IRQHandler()

```
void PWM1_Fault_IRQHandler (  
    void )
```

: PWM1_Fault_IRQHandler: PWM1 fault Interrupt handle

Note

Function ID:

Returns

: void

Definition at line 2339 of file Pwm_Hal.c.

4.3.1.6 PWM1_Overflow_IRQHandler()

```
void PWM1_Overflow_IRQHandler (  
    void )
```

: PWM1_Overflow_IRQHandler: PWM1 overflow Interrupt handle

Note

Function ID:

Returns

: void

Definition at line 2174 of file Pwm_Hal.c.

4.3.1.7 PWM2_Channel_IRQHandler()

```
void PWM2_Channel_IRQHandler (  
    void )
```

: PWM2_Channel_IRQHandler: PWM2 channel Interrupt handle

Note

Function ID:

Returns

: void

Definition at line 2266 of file Pwm_Hal.c.

4.3.1.8 PWM2_Fault_IRQHandler()

```
void PWM2_Fault_IRQHandler (  
    void )
```

: PWM2_Fault_IRQHandler: PWM2 fault Interrupt handle

Note

Function ID:

Returns

: void

Definition at line 2349 of file Pwm_Hal.c.

4.3.1.9 PWM2_Overflow_IRQHandler()

```
void PWM2_Overflow_IRQHandler (  
    void )
```

: PWM2_Overflow_IRQHandler: PWM2 overflow Interrupt handle

Note

Function ID:

Returns

: void

Definition at line 2184 of file Pwm_Hal.c.

4.3.1.10 PWM3_Channel_IRQHandler()

```
void PWM3_Channel_IRQHandler (  
    void )
```

: PWM3_Channel_IRQHandler: PWM3 channel Interrupt handle

Note

Function ID:

Returns

: void

Definition at line 2276 of file Pwm_Hal.c.

4.3.1.11 PWM3_Fault_IRQHandler()

```
void PWM3_Fault_IRQHandler (
    void )
```

: PWM3_Fault_IRQHandler: PWM3 fault Interrupt handle

Note

Function ID:

Returns

: void

Definition at line 2359 of file Pwm_Hal.c.

4.3.1.12 PWM3_Overflow_IRQHandler()

```
void PWM3_Overflow_IRQHandler (
    void )
```

: PWM3_Overflow_IRQHandler: PWM3 overflow Interrupt handle

Note

Function ID:

Returns

: void

Definition at line 2194 of file Pwm_Hal.c.

4.3.1.13 PWM4_Channel_IRQHandler()

```
void PWM4_Channel_IRQHandler (
    void )
```

: PWM4_Channel_IRQHandler: PWM4 channel Interrupt handle

Note

Function ID:

Returns

: void

Definition at line 2286 of file Pwm_Hal.c.

4.3.1.14 PWM4_Fault_IRQHandler()

```
void PWM4_Fault_IRQHandler (  
    void )
```

: PWM4_Fault_IRQHandler: PWM4 fault Interrupt handle

Note

Function ID:

Returns

: void

Definition at line 2369 of file Pwm_Hal.c.

4.3.1.15 PWM4_Overflow_IRQHandler()

```
void PWM4_Overflow_IRQHandler (  
    void )
```

: PWM4_Overflow_IRQHandler: PWM4 overflow Interrupt handle

Note

Function ID:

Returns

: void

Definition at line 2204 of file Pwm_Hal.c.

4.3.1.16 PWM5_Channel_IRQHandler()

```
void PWM5_Channel_IRQHandler (  
    void )
```

: PWM5_Channel_IRQHandler: PWM5 channel Interrupt handle

Note

Function ID:

Returns

: void

Definition at line 2296 of file Pwm_Hal.c.

4.3.1.17 PWM5_Fault_IRQHandler()

```
void PWM5_Fault_IRQHandler (
    void )
```

: PWM5_Fault_IRQHandler: PWM5 fault Interrupt handle

Note

Function ID:

Returns

: void

Definition at line 2379 of file Pwm_Hal.c.

4.3.1.18 PWM5_Overflow_IRQHandler()

```
void PWM5_Overflow_IRQHandler (
    void )
```

: PWM5_Overflow_IRQHandler: PWM5 overflow Interrupt handle

Note

Function ID:

Returns

: void

Definition at line 2214 of file Pwm_Hal.c.

4.3.1.19 Pwm_Hal_ClearChannelInterruptFlag()

```
void Pwm_Hal_ClearChannelInterruptFlag (
    Pwm_Hal_InstanceType Instance,
    uint32 Mask )
```

: Pwm_Hal_ClearChannelInterruptFlag: clear all channel interrupt flag

Note

Function ID: DES_PWM_API_218

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Mask</i> | clear mask |

Returns

: void

Definition at line 916 of file Pwm_Hal.c.

4.3.1.20 Pwm_Hal_ClearDetectEventFlag()

```
void Pwm_Hal_ClearDetectEventFlag (  
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_ClearDetectEventFlag: Clear Z index detect event status.

Note

Function ID:

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

void

Definition at line 1989 of file Pwm_Hal.c.

4.3.1.21 Pwm_Hal_ClearFaultFlag()

```
void Pwm_Hal_ClearFaultFlag (  
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_ClearFaultFlag: Clear the OR value of each fault input pin flag.

Note

Function ID: DES_PWM_API_259

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

void

Definition at line 1866 of file Pwm_Hal.c.

4.3.1.22 Pwm_Hal_ClearFaultPinFlag()

```
void Pwm_Hal_ClearFaultPinFlag (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_FaultInputIdType FaultInputId )
```

: Pwm_Hal_ClearFaultPinFlag: Clear fault pin detection flag.

Note

Function ID: DES_PWM_API_257

Parameters

| | | |
|----|-------------------|---------------------------|
| in | Instance | Enum Pwm_Hal_InstanceType |
| in | Fault↔ InputId | PWM_FAULT_INPUT_0\1\2\3 |

Returns

void

Definition at line 1831 of file Pwm_Hal.c.

4.3.1.23 Pwm_Hal_ClearOverflowFlag()

```
void Pwm_Hal_ClearOverflowFlag (
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_ClearOverFlowFlag: clear pwm OverFlow flag

Note

Function ID: DES_PWM_API_220

Parameters

| | | |
|----|----------|---------------------------|
| in | Instance | Enum Pwm_Hal_InstanceType |
|----|----------|---------------------------|

Returns

: void

Definition at line 955 of file Pwm_Hal.c.

4.3.1.24 Pwm_Hal_ClearQuadPhaseZFlag()

```
void Pwm_Hal_ClearQuadPhaseZFlag (  
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_ClearQuadPhaseZFlag: Clear the phaseZ Status.

Note

Function ID: DES_PWM_API_265

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

void

Definition at line 1955 of file Pwm_Hal.c.

4.3.1.25 Pwm_Hal_DeInit()

```
void Pwm_Hal_DeInit (  
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_DeInit: deinit pwm module

Note

Function ID: DES_PWM_API_203

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

: void

Definition at line 569 of file Pwm_Hal.c.

4.3.1.26 Pwm_Hal_EnableChannelDmaRequest()

```
void Pwm_Hal_EnableChannelDmaRequest (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    boolean Enable )
```

: Pwm_Hal_EnableChannelDmaRequest: Enable DMA requests for channel events

Note

Function ID: DES_PWM_API_223

Parameters

| | | |
|----|-----------------|-----------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 1013 of file Pwm_Hal.c.

4.3.1.27 Pwm_Hal_EnableChannelEventReset()

```
void Pwm_Hal_EnableChannelEventReset (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    boolean Enable )
```

: Pwm_Hal_EnableChannelEventReset: Does the channel event reset the counter.

Note

Function ID: DES_PWM_API_235

Parameters

| | | |
|----|-----------------|-----------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 1315 of file Pwm_Hal.c.

4.3.1.28 Pwm_Hal_EnableChannelHizOutput()

```
void Pwm_Hal_EnableChannelHizOutput (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    boolean Enable )
```

: Pwm_Hal_SetChannelHizOutput: Enable or disable channel high-Z output

Note

Function ID: DES_PWM_API_260

Parameters

| | | |
|----|-----------------|-----------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 1885 of file Pwm_Hal.c.

4.3.1.29 Pwm_Hal_EnableChannelInterrupt()

```
void Pwm_Hal_EnableChannelInterrupt (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    boolean Enable )
```

: Pwm_Hal_EnableChannelInterrupt: enable or disable pwm channel match Interrupt

Note

Function ID: DES_PWM_API_214

Parameters

| | | |
|----|-----------------|-----------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 807 of file Pwm_Hal.c.

4.3.1.30 Pwm_Hal_EnableChannelMatchTrigger()

```
void Pwm_Hal_EnableChannelMatchTrigger (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    boolean Enable )
```

: Pwm_Hal_EnableChannelMatchTrigger: Enable Channel match trigger source.

Note

Function ID: DES_PWM_API_231

Parameters

| | | |
|----|-----------------|-----------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 1212 of file Pwm_Hal.c.

4.3.1.31 Pwm_Hal_EnableChannelOutputMask()

```
void Pwm_Hal_EnableChannelOutputMask (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    boolean Enable )
```

: Pwm_Hal_EnableChannelOutputMask: Shielding channel output or not

Note

Function ID: DES_PWM_API_245

Parameters

| | | |
|----|-----------------|-----------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 1532 of file Pwm_Hal.c.

4.3.1.32 Pwm_Hal_EnableChannelPairInvert()

```
void Pwm_Hal_EnableChannelPairInvert (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelPairType ChannelPair,
    boolean Enable )
```

: Pwm_Hal_EnableChannelPairInvert: Set channel inverting control.

Note

Function ID: DES_PWM_API_251

Parameters

| | | |
|----|--------------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>ChannelPair</i> | PWM_CHANNEL_PAIR_0\1\2\3 |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

void

Definition at line 1660 of file Pwm_Hal.c.

4.3.1.33 Pwm_Hal_EnableChannelPairSymmetric()

```
void Pwm_Hal_EnableChannelPairSymmetric (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelPairType ChannelPair,
    boolean Enable )
```

: Pwm_Hal_EnableChannelPairSymmetric: Set pair channel symmetric.

Note

Function ID: DES_PWM_API_250

Parameters

| | | |
|----|--------------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>ChannelPair</i> | PWM_CHANNEL_PAIR_0\1\2\3 |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

void

Definition at line 1639 of file Pwm_Hal.c.

4.3.1.34 Pwm_Hal_EnableChannelSoftControl()

```
void Pwm_Hal_EnableChannelSoftControl (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    boolean Enable )
```

: Pwm_Hal_EnableChannelSoftControl: Enable channel soft control

Note

Function ID: DES_PWM_API_239

Parameters

| | | |
|----|-----------------|-----------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 1398 of file Pwm_Hal.c.

4.3.1.35 Pwm_Hal_EnableCombineChannelSoftControl()

```
void Pwm_Hal_EnableCombineChannelSoftControl (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    boolean Enable )
```

: Pwm_Hal_EnableCombineChannelSoftControl: enable soft control function

Note

Function ID: DES_PWM_API_240

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\2\4\6 |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 1419 of file Pwm_Hal.c.

4.3.1.36 Pwm_Hal_EnableFaultInputFilter()

```
void Pwm_Hal_EnableFaultInputFilter (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_FaultInputIdType FaultInputId,
    boolean Enable )
```

: Pwm_Hal_EnableFaultInputFilter: Enable the filtering function of the fault input pin.

Note

Function ID: DES_PWM_API_254

Parameters

| | | |
|----|---------------------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Fault↔ InputId</i> | PWM_FAULT_INPUT_0\1\2\3 |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

void

Definition at line 1771 of file Pwm_Hal.c.

4.3.1.37 Pwm_Hal_EnableFaultPinInput()

```
void Pwm_Hal_EnableFaultPinInput (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_FaultInputIdType FaultInputId,
    boolean Enable )
```

: Pwm_Hal_EnableFaultPinInput: Enable this fault input pin.

Note

Function ID: DES_PWM_API_253

Parameters

| | | |
|----|---------------------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Fault↔ InputId</i> | PWM_FAULT_INPUT_0\1\2\3 |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

void

Definition at line 1750 of file Pwm_Hal.c.

4.3.1.38 Pwm_Hal_EnableGlobalTimeBase()

```
void Pwm_Hal_EnableGlobalTimeBase (
    Pwm_Hal_InstanceType Instance,
    boolean Enable )
```

: Pwm_Hal_EnableGlobalTimeBase: Enable/Disable global timer base.

Note

Function ID: DES_PWM_API_227

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 1135 of file Pwm_Hal.c.

4.3.1.39 Pwm_Hal_EnableGlobalTimeBaseOutput()

```
void Pwm_Hal_EnableGlobalTimeBaseOutput (
    Pwm_Hal_InstanceType Instance,
    boolean Enable )
```

: Pwm_Hal_EnableGlobalTimeBaseOutput: Global timer base output enable/disable.

Note

Function ID: DES_PWM_API_228

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 1154 of file Pwm_Hal.c.

4.3.1.40 Pwm_Hal_EnableInitTrigger()

```
void Pwm_Hal_EnableInitTrigger (
    Pwm_Hal_InstanceType Instance,
    boolean Enable )
```

: Pwm_Hal_EnableInitTrigger: Enable initialization trigger source.

Note

Function ID: DES_PWM_API_229

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 1173 of file Pwm_Hal.c.

4.3.1.41 Pwm_Hal_EnableMaxTrigger()

```
void Pwm_Hal_EnableMaxTrigger (
    Pwm_Hal_InstanceType Instance,
    boolean Enable )
```

: Pwm_Hal_EnableMaxTrigger: Enable max trigger source.

Note

Function ID: DES_PWM_API_230

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 1192 of file Pwm_Hal.c.

4.3.1.42 Pwm_Hal_EnableOverflowEvent()

```
void Pwm_Hal_EnableOverflowEvent (
    Pwm_Hal_InstanceType Instance,
    boolean Enable )
```

: Pwm_Hal_EnableOverflowEvent: enable or disable pwm module overflow event.

Note

Function ID: DES_PWM_API_215

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 858 of file Pwm_Hal.c.

4.3.1.43 Pwm_Hal_EnableOverflowInterrupt()

```
void Pwm_Hal_EnableOverflowInterrupt (
    Pwm_Hal_InstanceType Instance,
    boolean Enable )
```

: Pwm_Hal_EnableOverflowInterrupt: enable or disable pwm module overflow Interrupt.

Note

Function ID: DES_PWM_API_213

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 830 of file Pwm_Hal.c.

4.3.1.44 Pwm_Hal_EnableSyncBypass()

```
void Pwm_Hal_EnableSyncBypass (
    Pwm_Hal_InstanceType Instance,
    boolean Enable )
```

: Pwm_Hal_EnableSyncBypass: CHnV/CNTIN/MCVR sync bypass.

Note

Function ID: DES_PWM_API_225

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 1095 of file Pwm_Hal.c.

4.3.1.45 Pwm_Hal_EnableUnderflowEvent()

```
void Pwm_Hal_EnableUnderflowEvent (
    Pwm_Hal_InstanceType Instance,
    boolean Enable )
```

: Pwm_Hal_EnableUnderflowEvent: enable or disable pwm module under overflow event.

Note

Function ID: DES_PWM_API_216

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 877 of file Pwm_Hal.c.

4.3.1.46 Pwm_Hal_EnableWriteProtection()

```
void Pwm_Hal_EnableWriteProtection (
    Pwm_Hal_InstanceType Instance,
    boolean Enable )
```

: Pwm_Hal_EnableWriteProtection: Enable write protection.

Note

Function ID: DES_PWM_API_266

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

void

Definition at line 2007 of file Pwm_Hal.c.

4.3.1.47 Pwm_Hal_GetAllChannelLevel()

```
uint32 Pwm_Hal_GetAllChannelLevel (
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_GetAllChannelLevel: get all pwm channel level

Note

Function ID: DES_PWM_API_222

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

: uint32: channel level

Definition at line 991 of file Pwm_Hal.c.

4.3.1.48 Pwm_Hal_GetChannelInterruptFlag()

```
uint32 Pwm_Hal_GetChannelInterruptFlag (
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_GetChannelInterruptFlag: get all pwm interrupt flag

Note

Function ID: DES_PWM_API_217

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

: uint32: interrupt flag

Definition at line 896 of file Pwm_Hal.c.

4.3.1.49 Pwm_Hal_GetChannelValue()

```
uint16 Pwm_Hal_GetChannelValue (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel )
```

: Pwm_Hal_GetChannelValue: Get channel match value.

Note

Function ID: DES_PWM_API_205

Parameters

| | | |
|----|-----------------|-----------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |

Returns

: Channel value.

Definition at line 785 of file Pwm_Hal.c.

4.3.1.50 Pwm_Hal_GetCountValue()

```
uint16 Pwm_Hal_GetCountValue (
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_GetCountValue: get module counter value

Note

Function ID: DES_PWM_API_211

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

: uint16: counter value

Definition at line 666 of file Pwm_Hal.c.

4.3.1.51 Pwm_Hal_GetDetectEventFlag()

```
uint32 Pwm_Hal_GetDetectEventFlag (
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_GetDetectEventFlag: Check Z index detect event status.

Note

Function ID:

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

Z index event status

Definition at line 1972 of file Pwm_Hal.c.

4.3.1.52 Pwm_Hal_GetFaultFlag()

```
uint32 Pwm_Hal_GetFaultFlag (
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_GetFaultFlag: Get the OR value of each fault input pin flag.

Note

Function ID: DES_PWM_API_258

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

The OR value of each fault input pin flag

Definition at line 1849 of file Pwm_Hal.c.

4.3.1.53 Pwm_Hal_GetFaultPinFlag()

```
uint32 Pwm_Hal_GetFaultPinFlag (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_FaultInputIdType FaultInputId )
```

: Pwm_Hal_GetFaultPinFlag: Get fault pin detection flag.

Note

Function ID: DES_PWM_API_256

Parameters

| | | |
|----|---------------------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Fault↔ InputId</i> | PWM_FAULT_INPUT_0\1\2\3 |

Returns

Fault pin detection flag.

Definition at line 1812 of file Pwm_Hal.c.

4.3.1.54 Pwm_Hal_GetHallStatus()

```
uint32 Pwm_Hal_GetHallStatus (
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_GetHallStatus: Get pwm hall status.

Note

Function ID: DES_PWM_API_236

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

: Hall status.

Definition at line 1334 of file Pwm_Hal.c.

4.3.1.55 Pwm_Hal_GetInitCountValue()

```
uint16 Pwm_Hal_GetInitCountValue (
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_GetInitCountValue: Get pwm module counter init value.

Note

Function ID: DES_PWM_API_209

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

: uint16: Init value

Definition at line 743 of file Pwm_Hal.c.

4.3.1.56 Pwm_Hal_GetMaxCountValue()

```
uint16 Pwm_Hal_GetMaxCountValue (
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_GetMaxCountValue: Get pwm module counter max value.

Note

Function ID: DES_PWM_API_207

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

: uint16: max value

Definition at line 704 of file Pwm_Hal.c.

4.3.1.57 Pwm_Hal_GetOutputMask()

```
uint32 Pwm_Hal_GetOutputMask (  
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_GetOutputMask: get pwm moude channel output status

Note

Function ID: DES_PWM_API_246

Parameters

| | | |
|----|----------|---------------------------|
| in | Instance | Enum Pwm_Hal_InstanceType |
|----|----------|---------------------------|

Returns

: uint32: channel output state

Definition at line 1570 of file Pwm_Hal.c.

4.3.1.58 Pwm_Hal_GetOverflowDir()

```
uint32 Pwm_Hal_GetOverflowDir (  
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_GetOverFlowDir: get pwm Overflow direction

Note

Function ID: DES_PWM_API_221

Parameters

| | | |
|----|----------|---------------------------|
| in | Instance | Enum Pwm_Hal_InstanceType |
|----|----------|---------------------------|

Returns

: uint32: channel OverFlow flag

Definition at line 972 of file Pwm_Hal.c.

4.3.1.59 Pwm_Hal_GetOverflowFlag()

```
uint32 Pwm_Hal_GetOverflowFlag (  
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_GetOverflowFlag: get all pwm interrupt flag

Note

Function ID: DES_PWM_API_219

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

: uint32: interrupt flag

Definition at line 935 of file Pwm_Hal.c.

4.3.1.60 Pwm_Hal_GetQuadCountingDir()

```
uint32 Pwm_Hal_GetQuadCountingDir (
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_GetQuadCountingDir: Get the current quadrature decoding count direction.

Note

Function ID: DES_PWM_API_262

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

counting direction

Definition at line 1904 of file Pwm_Hal.c.

4.3.1.61 Pwm_Hal_GetQuadOverflowDir()

```
uint32 Pwm_Hal_GetQuadOverflowDir (
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_GetQuadOverflowDir: Get the quadrature timer overflow direction.

Note

Function ID: DES_PWM_API_263

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

timer overflow direction

Definition at line 1921 of file Pwm_Hal.c.

4.3.1.62 Pwm_Hal_GetQuadPhaseZFlag()

```
uint32 Pwm_Hal_GetQuadPhaseZFlag (  
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_GetQuadPhaseZFlag: Get the phaseZ Status.

Note

Function ID: DES_PWM_API_264

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

phaseZ Status

Definition at line 1938 of file Pwm_Hal.c.

4.3.1.63 Pwm_Hal_InitFaultControl()

```
void Pwm_Hal_InitFaultControl (  
    Pwm_Hal_InstanceType Instance,  
    const Pwm_Hal_FaultCfg * ConfigPtr )
```

: Pwm_Hal_InitFaultControl: Init fault control. Only applicable to PWM modulation output mode, called after PWM_Init().

Note

Function ID: DES_PWM_API_252

Parameters

| | | |
|----|------------------|------------------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>ConfigPtr</i> | Pointer to configuration structure |

Returns

void

Definition at line 1681 of file Pwm_Hal.c.

4.3.1.64 Pwm_Hal_InitInputMode()

```
void Pwm_Hal_InitInputMode (
    Pwm_Hal_InstanceType Instance,
    const Pwm_Hal_InputCfg * Config )
```

: Pwm_Hal_InitInputMode: init pwm input mode

: Pwm_Hal_InitOutputMode: init pwm input mode

Note

Function ID: DES_PWM_API_202

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Config</i> | pwm input config pointer |

Returns

: void

Definition at line 369 of file Pwm_Hal.c.

4.3.1.65 Pwm_Hal_InitOutputMode()

```
void Pwm_Hal_InitOutputMode (
    Pwm_Hal_InstanceType Instance,
    const Pwm_Hal_OutputCfg * Config )
```

: Pwm_Hal_InitOutputMode: init pwm output mode

Note

Function ID: DES_PWM_API_201

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Config</i> | pwm output config pointer |

Returns

: void

Definition at line 224 of file Pwm_Hal.c.

4.3.1.66 Pwm_Hal_InitQuadDecoderMode()

```
void Pwm_Hal_InitQuadDecoderMode (
    Pwm_Hal_InstanceType Instance,
    const Pwm_Hal_QuadDecoderCfg * Config )
```

: Pwm_Hal_InitQuadDecoderMode: init pwm quadrature decoder mode

Note

Function ID: DES_PWM_API_261

Parameters

| | | |
|----|-----------------|---------------------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Config</i> | pwm quadrature decoder config pointer |

Returns

: void

Definition at line 508 of file Pwm_Hal.c.

4.3.1.67 Pwm_Hal_InitSyncConfigSet()

```
void Pwm_Hal_InitSyncConfigSet (
    Pwm_Hal_InstanceType Instance,
    const Pwm_Hal_SyncCfg * ConfigPtr )
```

: Pwm_Hal_InitSyncConfigSet: PWM synchronization control init

Note

Function ID: DES_PWM_API_224

Parameters

| | | |
|----|------------------|------------------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>ConfigPtr</i> | pointer to configuration structure |

Returns

: void

Definition at line 1033 of file Pwm_Hal.c.

4.3.1.68 Pwm_Hal_ResetCounter()

```
void Pwm_Hal_ResetCounter (
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_ResetCounter: reset module counter value

Note

Function ID: DES_PWM_API_212

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

: void

Definition at line 648 of file Pwm_Hal.c.

4.3.1.69 Pwm_Hal_SetChannelCaptureEdge()

```
void Pwm_Hal_SetChannelCaptureEdge (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    Pwm_Hal_EdgeType Edge )
```

: Pwm_Hal_SetChannelCaptureEdge: set input channel capture edge

Note

Function ID: DES_PWM_API_233

Parameters

| | | |
|----|-----------------|-----------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>Edge</i> | Enum Pwm_Hal_EdgeType |

Returns

: void

Definition at line 1253 of file Pwm_Hal.c.

4.3.1.70 Pwm_Hal_SetChannelCompareAction()

```
void Pwm_Hal_SetChannelCompareAction (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    Pwm_Hal_CompareActionType Action )
```

: Pwm_Hal_SetChannelCompareAction: Set matching action for output comparison mode

Note

Function ID: DES_PWM_API_237

Parameters

| | | |
|----|-----------------|--------------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>Action</i> | Enum Pwm_Hal_CompareActionType |

Returns

: void

Definition at line 1356 of file Pwm_Hal.c.

4.3.1.71 Pwm_Hal_SetChannelInputFilterVal()

```
void Pwm_Hal_SetChannelInputFilterVal (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    uint8 Value )
```

: Pwm_Hal_SetChannelInputFilterVal: Set channel input filter value.

Note

Function ID: DES_PWM_API_234

Parameters

| | | |
|----|-----------------|-----------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>Value</i> | filter value(0~15) |

Returns

: void

Definition at line 1275 of file Pwm_Hal.c.

4.3.1.72 Pwm_Hal_SetChannelMatchDither()

```
void Pwm_Hal_SetChannelMatchDither (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    uint8 DitherValue )
```

: Pwm_Hal_SetChannelMatchDither: Set channel dither value.

Note

Function ID: DES_PWM_API_243

Parameters

| | | |
|----|--------------------|--------------------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>DitherValue</i> | Channel match dither value (0 ~ 31). |

Returns

: void

Definition at line 1492 of file Pwm_Hal.c.

4.3.1.73 Pwm_Hal_SetChannelPolarity()

```
void Pwm_Hal_SetChannelPolarity (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    Pwm_Hal_ActivePolarityType Polarity )
```

: Pwm_Hal_SetChannelPolarity: Set channel polarity.

Note

Function ID: DES_PWM_API_248

Parameters

| | | |
|----|-----------------|-----------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>Polarity</i> | Output polarity. |

Returns

void

Definition at line 1591 of file Pwm_Hal.c.

4.3.1.74 Pwm_Hal_SetChannelSoftControlLevel()

```
void Pwm_Hal_SetChannelSoftControlLevel (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    Pwm_Hal_OutputLevelType Level )
```

: Pwm_Hal_SetChannelSoftControlLevel: clear all pwm interrupt flag

Note

Function ID: DES_PWM_API_242

Parameters

| | | |
|----|-----------------|-----------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>Level</i> | Output level |

Returns

: void

Definition at line 1470 of file Pwm_Hal.c.

4.3.1.75 Pwm_Hal_SetChannelValue()

```
void Pwm_Hal_SetChannelValue (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    uint16 Value )
```

: Pwm_Hal_SetChannelValue: Set channel value.

Note

Function ID: DES_PWM_API_206

Parameters

| | | |
|----|-----------------|-----------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>Value</i> | Channel value |

Returns

: void

Definition at line 765 of file Pwm_Hal.c.

4.3.1.76 Pwm_Hal_SetClockSource()

```
void Pwm_Hal_SetClockSource (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ClockSourceType ClkSource,
    uint16 ClkPsc )
```

: Pwm_Hal_SetClockSource: Set pwm module clock source and clock psc

Note

Function ID: DES_PWM_API_204

Parameters

| | | |
|----|------------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>ClkSource</i> | clock source type |
| in | <i>ClkPsc</i> | clock prescaler |

Returns

: void

Definition at line 628 of file Pwm_Hal.c.

4.3.1.77 Pwm_Hal_SetDeadtime()

```
void Pwm_Hal_SetDeadtime (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelPairType ChannelPair,
    Pwm_Hal_DeadTimePscType Prescaler,
    uint16 Value )
```

: Pwm_Hal_SetDeadtime: Set deadtime prescaler & value.

Note

Function ID: DES_PWM_API_249

Parameters

| | | |
|----|--------------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>ChannelPair</i> | PWM_CHANNEL_PAIR_0\1\2\3 |

Parameters

| | | |
|----|------------------|---|
| in | <i>Prescaler</i> | prescaler divider <ul style="list-style-type: none">• PWM_DEADTIME_DIVID_1• PWM_DEADTIME_DIVID_4• PWM_DEADTIME_DIVID_16 |
| in | <i>Value</i> | inserted value <ul style="list-style-type: none">• 0 ~ 1023 |

Returns

void

Definition at line 1617 of file Pwm_Hal.c.

4.3.1.78 Pwm_Hal_SetDebugMode()

```
void Pwm_Hal_SetDebugMode (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_DebugModeType Mode )
```

: Pwm_Hal_SetDebugMode: Set debug mode.

Note

Function ID: DES_PWM_API_267

Parameters

| | | |
|----|-----------------|---|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Mode</i> | pwm debug mode <ul style="list-style-type: none">• PWM_DEBUG_MODE_NO_EFFECT• PWM_DEBUG_MODE_COUNTER_STOPPED_OUTPUT_PREVIOUS• PWM_DEBUG_MODE_COUNTER_STOPPED_OUTPUT_HIGH |

Returns

void

Definition at line 2029 of file Pwm_Hal.c.

4.3.1.79 Pwm_Hal_SetFaultInputPolarity()

```
void Pwm_Hal_SetFaultInputPolarity (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_FaultInputIdType FaultInputId,
    Pwm_Hal_ActivePolarityType ActivePolarity )
```

: Pwm_Hal_SetFaultInputPolarity: Set fault input polarity.

Note

Function ID: DES_PWM_API_255

Parameters

| | | |
|----|-----------------------|--|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>FaultInputId</i> | PWM_FAULT_INPUT_0\1\2\3 |
| in | <i>ActivePolarity</i> | PWM_ACTIVE_POLARITY_HIGH / PWM_ACTIVE_POLARITY_LOW |

Returns

void

Definition at line 1792 of file Pwm_Hal.c.

4.3.1.80 Pwm_Hal_SetInitCountValue()

```
void Pwm_Hal_SetInitCountValue (
    Pwm_Hal_InstanceType Instance,
    uint16 Value )
```

: Pwm_Hal_SetInitCountValue: Set pwm module counter begin value.

Note

Function ID: DES_PWM_API_210

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Value</i> | begin value |

Returns

: void

Definition at line 725 of file Pwm_Hal.c.

4.3.1.81 Pwm_Hal_SetInputFilterPsc()

```
void Pwm_Hal_SetInputFilterPsc (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_InputFilterPscType Prescaler )
```

: Pwm_Hal_SetInputFilterPsc: Set pwm input filter psc value.

Note

Function ID:

Parameters

| | | |
|----|------------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Prescaler</i> | filter psc(0 ~ 111) |

Returns

: void

Definition at line 1295 of file Pwm_Hal.c.

4.3.1.82 Pwm_Hal_SetMaxCountValue()

```
void Pwm_Hal_SetMaxCountValue (
    Pwm_Hal_InstanceType Instance,
    uint16 Value )
```

: Pwm_Hal_SetMaxCountValue: Set pwm module counter max value.

Note

Function ID: DES_PWM_API_208

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Value</i> | max value |

Returns

: void

Definition at line 686 of file Pwm_Hal.c.

4.3.1.83 Pwm_Hal_SetOutputMask()

```
void Pwm_Hal_SetOutputMask (
    Pwm_Hal_InstanceType Instance,
    uint8 Mask )
```

: Pwm_Hal_SetOutputMask: Set the mask status for all channels

Note

Function ID: DES_PWM_API_247

Parameters

| | | |
|----|-----------------|--------------------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Mask</i> | The shielding status of all channels |

Returns

: void

Definition at line 1552 of file Pwm_Hal.c.

4.3.1.84 Pwm_Hal_SetPeriodDither()

```
void Pwm_Hal_SetPeriodDither (
    Pwm_Hal_InstanceType Instance,
    uint8 MaxCountDitherValue )
```

: PWM_DRV_SetMaxCountDitherValue: Set mod dither value.

Note

Function ID: DES_PWM_API_244

Parameters

| | | |
|----|----------------------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>MaxCountDitherValue</i> | mod dither value (0 ~ 31) |

Returns

: void

Definition at line 1512 of file Pwm_Hal.c.

4.3.1.85 Pwm_Hal_SetSoftControlEnableStates()

```
void Pwm_Hal_SetSoftControlEnableStates (
    Pwm_Hal_InstanceType Instance,
    uint8 EnableMask )
```

: Pwm_Hal_SetSoftControlEnableStates: Set the software output enable status for each channel.

Note

Function ID: DES_PWM_API_238

Parameters

| | | |
|----|-------------------|-------------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>EnableMask</i> | Status masks for each channel |

Returns

: void

Definition at line 1377 of file Pwm_Hal.c.

4.3.1.86 Pwm_Hal_SetSoftControlLevels()

```
void Pwm_Hal_SetSoftControlLevels (
    Pwm_Hal_InstanceType Instance,
    uint8 LevelMask )
```

: Pwm_Hal_SetSoftControlLevels: Set software output levels for all channels

Note

Function ID: DES_PWM_API_241

Parameters

| | | |
|----|------------------|---|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>LevelMask</i> | software output levels for all channels |

Returns

: void

Definition at line 1449 of file Pwm_Hal.c.

4.3.1.87 Pwm_Hal_SetTriggerRatio()

```
void Pwm_Hal_SetTriggerRatio (
    Pwm_Hal_InstanceType Instance,
    uint8 TriggerRatio )
```

: Pwm_Hal_SetTriggerRatio: Set the max/match/init trigger ratio.

Note

Function ID: DES_PWM_API_232

Parameters

| | | |
|----|---------------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>TriggerRatio</i> | 0 ~ 7 |

Returns

: void

Definition at line 1232 of file Pwm_Hal.c.

4.3.1.88 Pwm_Hal_TrigSoftwareSync()

```
void Pwm_Hal_TrigSoftwareSync (
    Pwm_Hal_InstanceType Instance,
    boolean IsTrigger )
```

: Pwm_Hal_TrigSoftwareSync: start software sync trigger

Note

Function ID: DES_PWM_API_226

Parameters

| | | |
|----|------------------|-----------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>IsTrigger</i> | true->enable,flase->disable |

Returns

: void

Definition at line 1115 of file Pwm_Hal.c.

4.4 Pwm_Hal.h File Reference

```
#include "Pwm_Hal_Types.h"
```


Functions

- void [Pwm_Hal_InitOutputMode](#) ([Pwm_Hal_InstanceType](#) Instance, const [Pwm_Hal_OutputCfg](#) *Config)
: *Pwm_Hal_InitOutputMode: init pwm output mode*
- void [Pwm_Hal_InitInputMode](#) ([Pwm_Hal_InstanceType](#) Instance, const [Pwm_Hal_InputCfg](#) *Config)
: *Pwm_Hal_InitOutputMode: init pwm input mode*
- void [Pwm_Hal_InitQuadDecoderMode](#) ([Pwm_Hal_InstanceType](#) Instance, const [Pwm_Hal_QuadDecoderCfg](#) *Config)
: *Pwm_Hal_InitQuadDecoderMode: init pwm quadrature decoder mode*
- void [Pwm_Hal_DeInit](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_DeInit: deinit pwm module*
- void [Pwm_Hal_SetClockSource](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ClockSourceType](#) ClkSource, uint16 ClkPsc)
: *Pwm_Hal_SetClockSource: Set pwm module clock source and clock psc*
- void [Pwm_Hal_ResetCounter](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_ResetCounter: reset module counter value*
- uint16 [Pwm_Hal_GetCountValue](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_GetCountValue: get module counter value*
- void [Pwm_Hal_SetMaxCountValue](#) ([Pwm_Hal_InstanceType](#) Instance, uint16 Value)
: *Pwm_Hal_SetMaxCountValue: Set pwm module counter max value.*
- uint16 [Pwm_Hal_GetMaxCountValue](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_GetMaxCountValue: Get pwm module counter max value.*
- void [Pwm_Hal_SetInitCountValue](#) ([Pwm_Hal_InstanceType](#) Instance, uint16 Value)
: *Pwm_Hal_SetInitCountValue: Set pwm module counter begin value.*
- uint16 [Pwm_Hal_GetInitCountValue](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_GetInitCountValue: Get pwm module counter init value.*
- void [Pwm_Hal_SetChannelValue](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, uint16 Value)
: *Pwm_Hal_SetChannelValue: Set channel value.*
- uint16 [Pwm_Hal_GetChannelValue](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel)
: *Pwm_Hal_GetChannelValue: Get channel match value.*
- void [Pwm_Hal_EnableChannelInterrupt](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, boolean Enable)
: *Pwm_Hal_EnableChannelInterrupt: enable or disable pwm channel match Interrupt*
- void [Pwm_Hal_EnableOverflowInterrupt](#) ([Pwm_Hal_InstanceType](#) Instance, boolean Enable)
: *Pwm_Hal_EnableOverflowInterrupt: enable or disable pwm module overflow Interrupt.*
- void [Pwm_Hal_EnableOverflowEvent](#) ([Pwm_Hal_InstanceType](#) Instance, boolean Enable)
: *Pwm_Hal_EnableOverflowEvent: enable or disable pwm module overflow event.*
- void [Pwm_Hal_EnableUnderflowEvent](#) ([Pwm_Hal_InstanceType](#) Instance, boolean Enable)
: *Pwm_Hal_EnableUnderflowEvent: enable or disable pwm module under overflow event.*
- uint32 [Pwm_Hal_GetChannelInterruptFlag](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_GetChannelInterruptFlag: get all pwm interrupt flag*
- void [Pwm_Hal_ClearChannelInterruptFlag](#) ([Pwm_Hal_InstanceType](#) Instance, uint32 Mask)
: *Pwm_Hal_ClearChannelInterruptFlag: clear all channel interrupt flag*
- uint32 [Pwm_Hal_GetOverflowFlag](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_GetOverflowFlag: get all pwm interrupt flag*
- void [Pwm_Hal_ClearOverflowFlag](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_ClearOverFlowFlag: clear pwm OverFlow flag*
- uint32 [Pwm_Hal_GetOverflowDir](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_GetOverFlowDir: get pwm Overflow direction*
- uint32 [Pwm_Hal_GetAllChannelLevel](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_GetAllChannelLevel: get all pwm channel level*
- void [Pwm_Hal_EnableChannelDmaRequest](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, boolean Enable)

- : Pwm_Hal_EnableChannelDmaRequest: Enable DMA requests for channel events*
- void [Pwm_Hal_InitSyncConfigSet](#) ([Pwm_Hal_InstanceType](#) Instance, const [Pwm_Hal_SyncCfg](#) *ConfigPtr)
 - : Pwm_Hal_InitSyncConfigSet: PWM synchronization control init*
- void [Pwm_Hal_EnableSyncBypass](#) ([Pwm_Hal_InstanceType](#) Instance, boolean Enable)
 - : Pwm_Hal_EnableSyncBypass: CHnV/CNTIN/MCVR sync bypass.*
- void [Pwm_Hal_TrigSoftwareSync](#) ([Pwm_Hal_InstanceType](#) Instance, boolean IsTrigger)
 - : Pwm_Hal_TrigSoftwareSync: start software sync trigger*
- void [Pwm_Hal_EnableGlobalTimeBase](#) ([Pwm_Hal_InstanceType](#) Instance, boolean Enable)
 - : Pwm_Hal_EnableGlobalTimeBase: Enable/Disable global timer base.*
- void [Pwm_Hal_EnableGlobalTimeBaseOutput](#) ([Pwm_Hal_InstanceType](#) Instance, boolean Enable)
 - : Pwm_Hal_EnableGlobalTimeBaseOutput: Global timer base output enable/disable.*
- void [Pwm_Hal_EnableInitTrigger](#) ([Pwm_Hal_InstanceType](#) Instance, boolean Enable)
 - : Pwm_Hal_EnableInitTrigger: Enable initialization trigger source.*
- void [Pwm_Hal_EnableMaxTrigger](#) ([Pwm_Hal_InstanceType](#) Instance, boolean Enable)
 - : Pwm_Hal_EnableMaxTrigger: Enable max trigger source.*
- void [Pwm_Hal_EnableChannelMatchTrigger](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, boolean Enable)
 - : Pwm_Hal_EnableChannelMatchTrigger: Enable Channel match trigger source.*
- void [Pwm_Hal_SetTriggerRatio](#) ([Pwm_Hal_InstanceType](#) Instance, uint8 TriggerRatio)
 - : Pwm_Hal_SetTriggerRatio: Set the max/match/init trigger ratio.*
- void [Pwm_Hal_SetChannelCaptureEdge](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, [Pwm_Hal_EdgeType](#) Edge)
 - : Pwm_Hal_SetChannelCaptureEdge: set input channel capture edge*
- void [Pwm_Hal_SetChannelInputFilterVal](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, uint8 Value)
 - : Pwm_Hal_SetChannelInputFilterVal: Set channel input filter value.*
- void [Pwm_Hal_SetInputFilterPsc](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_InputFilterPscType](#) Prescaler)
 - : Pwm_Hal_SetInputFilterPsc: Set pwm input filter psc value.*
- void [Pwm_Hal_EnableChannelEventReset](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, boolean Enable)
 - : Pwm_Hal_EnableChannelEventReset: Does the channel event reset the counter.*
- uint32 [Pwm_Hal_GetHallStatus](#) ([Pwm_Hal_InstanceType](#) Instance)
 - : Pwm_Hal_GetHallStatus: Get pwm hall status.*
- void [Pwm_Hal_SetChannelCompareAction](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, [Pwm_Hal_CompareActionType](#) Action)
 - : Pwm_Hal_SetChannelCompareAction: Set matching action for output comparison mode*
- void [Pwm_Hal_SetSoftControlEnableStates](#) ([Pwm_Hal_InstanceType](#) Instance, uint8 EnableMask)
 - : Pwm_Hal_SetSoftControlEnableStates: Set the software output enable status for each channel.*
- void [Pwm_Hal_EnableChannelSoftControl](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, boolean Enable)
 - : Pwm_Hal_EnableChannelSoftControl: Enable channel soft control*
- void [Pwm_Hal_EnableCombineChannelSoftControl](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, boolean Enable)
 - : Pwm_Hal_EnableCombineChannelSoftControl: enable soft control function*
- void [Pwm_Hal_SetSoftControlLevels](#) ([Pwm_Hal_InstanceType](#) Instance, uint8 LevelMask)
 - : Pwm_Hal_SetSoftControlLevels: Set software output levels for all channels*
- void [Pwm_Hal_SetChannelSoftControlLevel](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, [Pwm_Hal_OutputLevelType](#) Level)
 - : Pwm_Hal_SetChannelSoftControlLevel: clear all pwm interrupt flag*
- void [Pwm_Hal_SetChannelMatchDither](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, uint8 DitherValue)
 - : Pwm_Hal_SetChannelMatchDither: Set channel dither value.*
- void [Pwm_Hal_SetPeriodDither](#) ([Pwm_Hal_InstanceType](#) Instance, uint8 MaxCountDitherValue)
 - : PWM_DRV_SetMaxCountDitherValue: Set mod dither value.*

- void [Pwm_Hal_EnableChannelOutputMask](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, boolean Enable)
: *Pwm_Hal_EnableChannelOutputMask: Shielding channel output or not*
- void [Pwm_Hal_SetOutputMask](#) ([Pwm_Hal_InstanceType](#) Instance, uint8 Mask)
: *Pwm_Hal_SetOutputMask: Set the mask status for all channels*
- uint32 [Pwm_Hal_GetOutputMask](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_GetOutputMask: get pwm moudel channel output status*
- void [Pwm_Hal_SetChannelPolarity](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, [Pwm_Hal_ActivePolarityType](#) Polarity)
: *Pwm_Hal_SetChannelPolarity: Set channel polarity.*
- void [Pwm_Hal_SetDeadtime](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelPairType](#) ChannelPair, [Pwm_Hal_DeadTimePscType](#) Prescaler, uint16 Value)
: *Pwm_Hal_SetDeadtime: Set deadtime prescaler & value.*
- void [Pwm_Hal_EnableChannelPairSymmetric](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelPairType](#) ChannelPair, boolean Enable)
: *Pwm_Hal_EnableChannelPairSymmetric: Set pair channel symmetric.*
- void [Pwm_Hal_EnableChannelPairInvert](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelPairType](#) ChannelPair, boolean Enable)
: *Pwm_Hal_EnableChannelPairInvert: Set channel inverting control.*
- void [Pwm_Hal_InitFaultControl](#) ([Pwm_Hal_InstanceType](#) Instance, const [Pwm_Hal_FaultCfg](#) *ConfigPtr)
: *Pwm_Hal_InitFaultControl: Init fault control. Only applicable to PWM modulation output mode, called after PWM_Init().*
- void [Pwm_Hal_EnableFaultPinInput](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_FaultInputIdType](#) FaultInputId, boolean Enable)
: *Pwm_Hal_EnableFaultPinInput: Enable this fault input pin.*
- void [Pwm_Hal_EnableFaultInputFilter](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_FaultInputIdType](#) FaultInputId, boolean Enable)
: *Pwm_Hal_EnableFaultInputFilter: Enable the filtering function of the fault input pin.*
- void [Pwm_Hal_SetFaultInputPolarity](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_FaultInputIdType](#) FaultInputId, [Pwm_Hal_ActivePolarityType](#) ActivePolarity)
: *Pwm_Hal_SetFaultInputPolarity: Set fault input polarity.*
- uint32 [Pwm_Hal_GetFaultPinFlag](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_FaultInputIdType](#) FaultInputId)
: *Pwm_Hal_GetFaultPinFlag: Get fault pin detection flag.*
- void [Pwm_Hal_ClearFaultPinFlag](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_FaultInputIdType](#) FaultInputId)
: *Pwm_Hal_ClearFaultPinFlag: Clear fault pin detection flag.*
- uint32 [Pwm_Hal_GetFaultFlag](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_GetFaultFlag: Get the OR value of each fault input pin flag.*
- void [Pwm_Hal_ClearFaultFlag](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_ClearFaultFlag: Clear the OR value of each fault input pin flag.*
- void [Pwm_Hal_EnableChannelHizOutput](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_ChannelType](#) Channel, boolean Enable)
: *Pwm_Hal_SetChannelHizOutput: Enable or disable channel high-Z output*
- uint32 [Pwm_Hal_GetQuadCountingDir](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_GetQuadCountingDir: Get the current quadrature decoding count direction.*
- uint32 [Pwm_Hal_GetQuadOverflowDir](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_GetQuadOverflowDir: Get the quadrature timer overflow direction.*
- uint32 [Pwm_Hal_GetQuadPhaseZFlag](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_GetQuadPhaseZFlag: Get the phaseZ Status.*
- void [Pwm_Hal_ClearQuadPhaseZFlag](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_ClearQuadPhaseZFlag: Clear the phaseZ Status.*
- uint32 [Pwm_Hal_GetDetectEventFlag](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_GetDetectEventFlag: Check Z index detect event status.*
- void [Pwm_Hal_ClearDetectEventFlag](#) ([Pwm_Hal_InstanceType](#) Instance)
: *Pwm_Hal_ClearDetectEventFlag: Clear Z index detect event status.*
- void [Pwm_Hal_EnableWriteProtection](#) ([Pwm_Hal_InstanceType](#) Instance, boolean Enable)

: *Pwm_Hal_EnableWriteProtection*: Enable write protection.

- void [Pwm_Hal_SetDebugMode](#) ([Pwm_Hal_InstanceType](#) Instance, [Pwm_Hal_DebugModeType](#) Mode)
: *Pwm_Hal_SetDebugMode*: Set debug mode.

4.4.1 Function Documentation

4.4.1.1 Pwm_Hal_ClearChannelInterruptFlag()

```
void Pwm_Hal_ClearChannelInterruptFlag (  
    Pwm\_Hal\_InstanceType Instance,  
    uint32 Mask )
```

: *Pwm_Hal_ClearChannelInterruptFlag*: clear all channel interrupt flag

Note

Function ID: DES_PWM_API_218

Parameters

| | | |
|----|-----------------|---|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Mask</i> | clear mask |

Returns

: void

Definition at line 916 of file [Pwm_Hal.c](#).

4.4.1.2 Pwm_Hal_ClearDetectEventFlag()

```
void Pwm_Hal_ClearDetectEventFlag (  
    Pwm\_Hal\_InstanceType Instance )
```

: *Pwm_Hal_ClearDetectEventFlag*: Clear Z index detect event status.

Note

Function ID:

Parameters

| | | |
|----|-----------------|---|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---|

Returns

void

Definition at line 1989 of file Pwm_Hal.c.

4.4.1.3 Pwm_Hal_ClearFaultFlag()

```
void Pwm_Hal_ClearFaultFlag (
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_ClearFaultFlag: Clear the OR value of each fault input pin flag.

Note

Function ID: DES_PWM_API_259

Parameters

| | | |
|----|----------|---------------------------|
| in | Instance | Enum Pwm_Hal_InstanceType |
|----|----------|---------------------------|

Returns

void

Definition at line 1866 of file Pwm_Hal.c.

4.4.1.4 Pwm_Hal_ClearFaultPinFlag()

```
void Pwm_Hal_ClearFaultPinFlag (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_FaultInputIdType FaultInputId )
```

: Pwm_Hal_ClearFaultPinFlag: Clear fault pin detection flag.

Note

Function ID: DES_PWM_API_257

Parameters

| | | |
|----|-------------------|---------------------------|
| in | Instance | Enum Pwm_Hal_InstanceType |
| in | Fault↔ InputId | PWM_FAULT_INPUT_0\1\2\3 |

Returns

void

Definition at line 1831 of file Pwm_Hal.c.

4.4.1.5 Pwm_Hal_ClearOverflowFlag()

```
void Pwm_Hal_ClearOverflowFlag (  
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_ClearOverFlowFlag: clear pwm OverFlow flag

Note

Function ID: DES_PWM_API_220

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

: void

Definition at line 955 of file Pwm_Hal.c.

4.4.1.6 Pwm_Hal_ClearQuadPhaseZFlag()

```
void Pwm_Hal_ClearQuadPhaseZFlag (  
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_ClearQuadPhaseZFlag: Clear the phaseZ Status.

Note

Function ID: DES_PWM_API_265

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

void

Definition at line 1955 of file Pwm_Hal.c.

4.4.1.7 Pwm_Hal_DeInit()

```
void Pwm_Hal_DeInit (
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_DeInit: deinit pwm module

Note

Function ID: DES_PWM_API_203

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

: void

Definition at line 569 of file Pwm_Hal.c.

4.4.1.8 Pwm_Hal_EnableChannelDmaRequest()

```
void Pwm_Hal_EnableChannelDmaRequest (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    boolean Enable )
```

: Pwm_Hal_EnableChannelDmaRequest: Enable DMA requests for channel events

Note

Function ID: DES_PWM_API_223

Parameters

| | | |
|----|-----------------|-----------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 1013 of file Pwm_Hal.c.

4.4.1.9 Pwm_Hal_EnableChannelEventReset()

```
void Pwm_Hal_EnableChannelEventReset (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    boolean Enable )
```

: Pwm_Hal_EnableChannelEventReset: Does the channel event reset the counter.

Note

Function ID: DES_PWM_API_235

Parameters

| | | |
|----|-----------------|-----------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 1315 of file Pwm_Hal.c.

4.4.1.10 Pwm_Hal_EnableChannelHizOutput()

```
void Pwm_Hal_EnableChannelHizOutput (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    boolean Enable )
```

: Pwm_Hal_SetChannelHizOutput: Enable or disable channel high-Z output

Note

Function ID: DES_PWM_API_260

Parameters

| | | |
|----|-----------------|-----------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 1885 of file Pwm_Hal.c.

4.4.1.11 Pwm_Hal_EnableChannelInterrupt()

```
void Pwm_Hal_EnableChannelInterrupt (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    boolean Enable )
```

: Pwm_Hal_EnableChannelInterrupt: enable or disable pwm channel match Interrupt

Note

Function ID: DES_PWM_API_214

Parameters

| | | |
|----|-----------------|-----------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 807 of file Pwm_Hal.c.

4.4.1.12 Pwm_Hal_EnableChannelMatchTrigger()

```
void Pwm_Hal_EnableChannelMatchTrigger (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    boolean Enable )
```

: Pwm_Hal_EnableChannelMatchTrigger: Enable Channel match trigger source.

Note

Function ID: DES_PWM_API_231

Parameters

| | | |
|----|-----------------|-----------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 1212 of file Pwm_Hal.c.

4.4.1.13 Pwm_Hal_EnableChannelOutputMask()

```
void Pwm_Hal_EnableChannelOutputMask (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    boolean Enable )
```

: Pwm_Hal_EnableChannelOutputMask: Shielding channel output or not

Note

Function ID: DES_PWM_API_245

Parameters

| | | |
|----|-----------------|-----------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 1532 of file Pwm_Hal.c.

4.4.1.14 Pwm_Hal_EnableChannelPairInvert()

```
void Pwm_Hal_EnableChannelPairInvert (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelPairType ChannelPair,
    boolean Enable )
```

: Pwm_Hal_EnableChannelPairInvert: Set channel inverting control.

Note

Function ID: DES_PWM_API_251

Parameters

| | | |
|----|--------------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>ChannelPair</i> | PWM_CHANNEL_PAIR_0\1\2\3 |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

void

Definition at line 1660 of file Pwm_Hal.c.

4.4.1.15 Pwm_Hal_EnableChannelPairSymmetric()

```
void Pwm_Hal_EnableChannelPairSymmetric (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelPairType ChannelPair,
    boolean Enable )
```

: Pwm_Hal_EnableChannelPairSymmetric: Set pair channel symmetric.

Note

Function ID: DES_PWM_API_250

Parameters

| | | |
|----|--------------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>ChannelPair</i> | PWM_CHANNEL_PAIR_0\1\2\3 |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

void

Definition at line 1639 of file Pwm_Hal.c.

4.4.1.16 Pwm_Hal_EnableChannelSoftControl()

```
void Pwm_Hal_EnableChannelSoftControl (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    boolean Enable )
```

: Pwm_Hal_EnableChannelSoftControl: Enable channel soft control

Note

Function ID: DES_PWM_API_239

Parameters

| | | |
|----|-----------------|-----------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 1398 of file Pwm_Hal.c.

4.4.1.17 Pwm_Hal_EnableCombineChannelSoftControl()

```
void Pwm_Hal_EnableCombineChannelSoftControl (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    boolean Enable )
```

: Pwm_Hal_EnableCombineChannelSoftControl: enable soft control function

Note

Function ID: DES_PWM_API_240

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\2\4\6 |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 1419 of file Pwm_Hal.c.

4.4.1.18 Pwm_Hal_EnableFaultInputFilter()

```
void Pwm_Hal_EnableFaultInputFilter (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_FaultInputIdType FaultInputId,
    boolean Enable )
```

: Pwm_Hal_EnableFaultInputFilter: Enable the filtering function of the fault input pin.

Note

Function ID: DES_PWM_API_254

Parameters

| | | |
|----|---------------------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Fault↔ InputId</i> | PWM_FAULT_INPUT_0\1\2\3 |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

void

Definition at line 1771 of file Pwm_Hal.c.

4.4.1.19 Pwm_Hal_EnableFaultPinInput()

```
void Pwm_Hal_EnableFaultPinInput (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_FaultInputIdType FaultInputId,
    boolean Enable )
```

: Pwm_Hal_EnableFaultPinInput: Enable this fault input pin.

Note

Function ID: DES_PWM_API_253

Parameters

| | | |
|----|---------------------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Fault↔ InputId</i> | PWM_FAULT_INPUT_0\1\2\3 |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

void

Definition at line 1750 of file Pwm_Hal.c.

4.4.1.20 Pwm_Hal_EnableGlobalTimeBase()

```
void Pwm_Hal_EnableGlobalTimeBase (
    Pwm_Hal_InstanceType Instance,
    boolean Enable )
```

: Pwm_Hal_EnableGlobalTimeBase: Enable/Disable global timer base.

Note

Function ID: DES_PWM_API_227

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 1135 of file Pwm_Hal.c.

4.4.1.21 Pwm_Hal_EnableGlobalTimeBaseOutput()

```
void Pwm_Hal_EnableGlobalTimeBaseOutput (
    Pwm_Hal_InstanceType Instance,
    boolean Enable )
```

: Pwm_Hal_EnableGlobalTimeBaseOutput: Global timer base output enable/disable.

Note

Function ID: DES_PWM_API_228

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 1154 of file Pwm_Hal.c.

4.4.1.22 Pwm_Hal_EnableInitTrigger()

```
void Pwm_Hal_EnableInitTrigger (
    Pwm_Hal_InstanceType Instance,
    boolean Enable )
```

: Pwm_Hal_EnableInitTrigger: Enable initialization trigger source.

Note

Function ID: DES_PWM_API_229

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 1173 of file Pwm_Hal.c.

4.4.1.23 Pwm_Hal_EnableMaxTrigger()

```
void Pwm_Hal_EnableMaxTrigger (
    Pwm_Hal_InstanceType Instance,
    boolean Enable )
```

: Pwm_Hal_EnableMaxTrigger: Enable max trigger source.

Note

Function ID: DES_PWM_API_230

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 1192 of file Pwm_Hal.c.

4.4.1.24 Pwm_Hal_EnableOverflowEvent()

```
void Pwm_Hal_EnableOverflowEvent (
    Pwm_Hal_InstanceType Instance,
    boolean Enable )
```

: Pwm_Hal_EnableOverflowEvent: enable or disable pwm module overflow event.

Note

Function ID: DES_PWM_API_215

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 858 of file Pwm_Hal.c.

4.4.1.25 Pwm_Hal_EnableOverflowInterrupt()

```
void Pwm_Hal_EnableOverflowInterrupt (
    Pwm_Hal_InstanceType Instance,
    boolean Enable )
```

: Pwm_Hal_EnableOverflowInterrupt: enable or disable pwm module overflow Interrupt.

Note

Function ID: DES_PWM_API_213

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 830 of file Pwm_Hal.c.

4.4.1.26 Pwm_Hal_EnableSyncBypass()

```
void Pwm_Hal_EnableSyncBypass (
    Pwm_Hal_InstanceType Instance,
    boolean Enable )
```

: Pwm_Hal_EnableSyncBypass: CHnV/CNTIN/MCVR sync bypass.

Note

Function ID: DES_PWM_API_225

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 1095 of file Pwm_Hal.c.

4.4.1.27 Pwm_Hal_EnableUnderflowEvent()

```
void Pwm_Hal_EnableUnderflowEvent (
    Pwm_Hal_InstanceType Instance,
    boolean Enable )
```

: Pwm_Hal_EnableUnderflowEvent: enable or disable pwm module under overflow event.

Note

Function ID: DES_PWM_API_216

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

: void

Definition at line 877 of file Pwm_Hal.c.

4.4.1.28 Pwm_Hal_EnableWriteProtection()

```
void Pwm_Hal_EnableWriteProtection (
    Pwm_Hal_InstanceType Instance,
    boolean Enable )
```

: Pwm_Hal_EnableWriteProtection: Enable write protection.

Note

Function ID: DES_PWM_API_266

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Enable</i> | TRUE/FALSE |

Returns

void

Definition at line 2007 of file Pwm_Hal.c.

4.4.1.29 Pwm_Hal_GetAllChannelLevel()

```
uint32 Pwm_Hal_GetAllChannelLevel (
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_GetAllChannelLevel: get all pwm channel level

Note

Function ID: DES_PWM_API_222

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

: uint32: channel level

Definition at line 991 of file Pwm_Hal.c.

4.4.1.30 Pwm_Hal_GetChannelInterruptFlag()

```
uint32 Pwm_Hal_GetChannelInterruptFlag (
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_GetChannelInterruptFlag: get all pwm interrupt flag

Note

Function ID: DES_PWM_API_217

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

: uint32: interrupt flag

Definition at line 896 of file Pwm_Hal.c.

4.4.1.31 Pwm_Hal_GetChannelValue()

```
uint16 Pwm_Hal_GetChannelValue (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel )
```

: Pwm_Hal_GetChannelValue: Get channel match value.

Note

Function ID: DES_PWM_API_205

Parameters

| | | |
|----|-----------------|-----------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |

Returns

: Channel value.

Definition at line 785 of file Pwm_Hal.c.

4.4.1.32 Pwm_Hal_GetCountValue()

```
uint16 Pwm_Hal_GetCountValue (
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_GetCountValue: get module counter value

Note

Function ID: DES_PWM_API_211

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

: uint16: counter value

Definition at line 666 of file Pwm_Hal.c.

4.4.1.33 Pwm_Hal_GetDetectEventFlag()

```
uint32 Pwm_Hal_GetDetectEventFlag (
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_GetDetectEventFlag: Check Z index detect event status.

Note

Function ID:

Parameters

| | | |
|----|----------|---------------------------|
| in | Instance | Enum Pwm_Hal_InstanceType |
|----|----------|---------------------------|

Returns

Z index event status

Definition at line 1972 of file Pwm_Hal.c.

4.4.1.34 Pwm_Hal_GetFaultFlag()

```
uint32 Pwm_Hal_GetFaultFlag (
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_GetFaultFlag: Get the OR value of each fault input pin flag.

Note

Function ID: DES_PWM_API_258

Parameters

| | | |
|----|----------|---------------------------|
| in | Instance | Enum Pwm_Hal_InstanceType |
|----|----------|---------------------------|

Returns

The OR value of each fault input pin flag

Definition at line 1849 of file Pwm_Hal.c.

4.4.1.35 Pwm_Hal_GetFaultPinFlag()

```
uint32 Pwm_Hal_GetFaultPinFlag (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_FaultInputIdType FaultInputId )
```

: Pwm_Hal_GetFaultPinFlag: Get fault pin detection flag.

Note

Function ID: DES_PWM_API_256

Parameters

| | | |
|----|-------------------|---------------------------|
| in | Instance | Enum Pwm_Hal_InstanceType |
| in | Fault↔ InputId | PWM_FAULT_INPUT_0\1\2\3 |

Returns

Fault pin detection flag.

Definition at line 1812 of file Pwm_Hal.c.

4.4.1.36 Pwm_Hal_GetHallStatus()

```
uint32 Pwm_Hal_GetHallStatus (
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_GetHallStatus: Get pwm hall status.

Note

Function ID: DES_PWM_API_236

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

: Hall status.

Definition at line 1334 of file Pwm_Hal.c.

4.4.1.37 Pwm_Hal_GetInitCountValue()

```
uint16 Pwm_Hal_GetInitCountValue (
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_GetInitCountValue: Get pwm module counter init value.

Note

Function ID: DES_PWM_API_209

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

: uint16: Init value

Definition at line 743 of file Pwm_Hal.c.

4.4.1.38 Pwm_Hal_GetMaxCountValue()

```
uint16 Pwm_Hal_GetMaxCountValue (
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_GetMaxCountValue: Get pwm module counter max value.

Note

Function ID: DES_PWM_API_207

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

: uint16: max value

Definition at line 704 of file Pwm_Hal.c.

4.4.1.39 Pwm_Hal_GetOutputMask()

```
uint32 Pwm_Hal_GetOutputMask (
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_GetOutputMask: get pwm module channel output status

Note

Function ID: DES_PWM_API_246

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

: uint32: channel output state

Definition at line 1570 of file Pwm_Hal.c.

4.4.1.40 Pwm_Hal_GetOverflowDir()

```
uint32 Pwm_Hal_GetOverflowDir (
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_GetOverflowDir: get pwm Overflow direction

Note

Function ID: DES_PWM_API_221

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

: uint32: channel OverFlow flag

Definition at line 972 of file Pwm_Hal.c.

4.4.1.41 Pwm_Hal_GetOverflowFlag()

```
uint32 Pwm_Hal_GetOverflowFlag (  
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_GetOverflowFlag: get all pwm interrupt flag

Note

Function ID: DES_PWM_API_219

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

: uint32: interrupt flag

Definition at line 935 of file Pwm_Hal.c.

4.4.1.42 Pwm_Hal_GetQuadCountingDir()

```
uint32 Pwm_Hal_GetQuadCountingDir (  
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_GetQuadCountingDir: Get the current quadrature decoding count direction.

Note

Function ID: DES_PWM_API_262

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

counting direction

Definition at line 1904 of file Pwm_Hal.c.

4.4.1.43 Pwm_Hal_GetQuadOverflowDir()

```
uint32 Pwm_Hal_GetQuadOverflowDir (
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_GetQuadOverflowDir: Get the quadrature timer overflow direction.

Note

Function ID: DES_PWM_API_263

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

timer overflow direction

Definition at line 1921 of file Pwm_Hal.c.

4.4.1.44 Pwm_Hal_GetQuadPhaseZFlag()

```
uint32 Pwm_Hal_GetQuadPhaseZFlag (
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_GetQuadPhaseZFlag: Get the phaseZ Status.

Note

Function ID: DES_PWM_API_264

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

phaseZ Status

Definition at line 1938 of file Pwm_Hal.c.

4.4.1.45 Pwm_Hal_InitFaultControl()

```
void Pwm_Hal_InitFaultControl (
    Pwm_Hal_InstanceType Instance,
    const Pwm_Hal_FaultCfg * ConfigPtr )
```

: Pwm_Hal_InitFaultControl: Init fault control. Only applicable to PWM modulation output mode, called after PWM_Init().

Note

Function ID: DES_PWM_API_252

Parameters

| | | |
|----|------------------|------------------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>ConfigPtr</i> | Pointer to configuration structure |

Returns

void

Definition at line 1681 of file Pwm_Hal.c.

4.4.1.46 Pwm_Hal_InitInputMode()

```
void Pwm_Hal_InitInputMode (
    Pwm_Hal_InstanceType Instance,
    const Pwm_Hal_InputCfg * Config )
```

: Pwm_Hal_InitOutputMode: init pwm input mode

Note

Function ID: DES_PWM_API_202

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Config</i> | pwm input config pointer |

Returns

: void

: Pwm_Hal_InitOutputMode: init pwm input mode

Note

Function ID: DES_PWM_API_202

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Config</i> | pwm input config pointer |

Returns

: void

Definition at line 369 of file Pwm_Hal.c.

4.4.1.47 Pwm_Hal_InitOutputMode()

```
void Pwm_Hal_InitOutputMode (
    Pwm_Hal_InstanceType Instance,
    const Pwm_Hal_OutputCfg * Config )
```

: Pwm_Hal_InitOutputMode: init pwm output mode

Note

Function ID: DES_PWM_API_201

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Config</i> | pwm output config pointer |

Returns

: void

Definition at line 224 of file Pwm_Hal.c.

4.4.1.48 Pwm_Hal_InitQuadDecoderMode()

```
void Pwm_Hal_InitQuadDecoderMode (
    Pwm_Hal_InstanceType Instance,
    const Pwm_Hal_QuadDecoderCfg * Config )
```

: Pwm_Hal_InitQuadDecoderMode: init pwm quadrature decoder mode

Note

Function ID: DES_PWM_API_261

Parameters

| | | |
|----|-----------------|---------------------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Config</i> | pwm quadrature decoder config pointer |

Returns

: void

Definition at line 508 of file Pwm_Hal.c.

4.4.1.49 Pwm_Hal_InitSyncConfigSet()

```
void Pwm_Hal_InitSyncConfigSet (
    Pwm_Hal_InstanceType Instance,
    const Pwm_Hal_SyncCfg * ConfigPtr )
```

: Pwm_Hal_InitSyncConfigSet: PWM synchronization control init

Note

Function ID: DES_PWM_API_224

Parameters

| | | |
|----|------------------|------------------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>ConfigPtr</i> | pointer to configuration structure |

Returns

: void

Definition at line 1033 of file Pwm_Hal.c.

4.4.1.50 Pwm_Hal_ResetCounter()

```
void Pwm_Hal_ResetCounter (
    Pwm_Hal_InstanceType Instance )
```

: Pwm_Hal_ResetCounter: reset module counter value

Note

Function ID: DES_PWM_API_212

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
|----|-----------------|---------------------------|

Returns

: void

Definition at line 648 of file Pwm_Hal.c.

4.4.1.51 Pwm_Hal_SetChannelCaptureEdge()

```
void Pwm_Hal_SetChannelCaptureEdge (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    Pwm_Hal_EdgeType Edge )
```

: Pwm_Hal_SetChannelCaptureEdge: set input channel capture edge

Note

Function ID: DES_PWM_API_233

Parameters

| | | |
|----|-----------------|-----------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>Edge</i> | Enum Pwm_Hal_EdgeType |

Returns

: void

Definition at line 1253 of file Pwm_Hal.c.

4.4.1.52 Pwm_Hal_SetChannelCompareAction()

```
void Pwm_Hal_SetChannelCompareAction (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    Pwm_Hal_CompareActionType Action )
```

: Pwm_Hal_SetChannelCompareAction: Set matching action for output comparison mode

Note

Function ID: DES_PWM_API_237

Parameters

| | | |
|----|-----------------|--------------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>Action</i> | Enum Pwm_Hal_CompareActionType |

Returns

: void

Definition at line 1356 of file Pwm_Hal.c.

4.4.1.53 Pwm_Hal_SetChannelInputFilterVal()

```
void Pwm_Hal_SetChannelInputFilterVal (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    uint8 Value )
```

: Pwm_Hal_SetChannelInputFilterVal: Set channel input filter value.

Note

Function ID: DES_PWM_API_234

Parameters

| | | |
|----|-----------------|-----------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>Value</i> | filter value(0~15) |

Returns

: void

Definition at line 1275 of file Pwm_Hal.c.

4.4.1.54 Pwm_Hal_SetChannelMatchDither()

```
void Pwm_Hal_SetChannelMatchDither (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    uint8 DitherValue )
```

: Pwm_Hal_SetChannelMatchDither: Set channel dither value.

Note

Function ID: DES_PWM_API_243

Parameters

| | | |
|----|--------------------|-----------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>DitherValue</i> | Channel match dither value. |

Returns

: void

Note

Function ID: DES_PWM_API_243

Parameters

| | | |
|----|--------------------|--------------------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>DitherValue</i> | Channel match dither value (0 ~ 31). |

Returns

: void

Definition at line 1492 of file Pwm_Hal.c.

4.4.1.55 Pwm_Hal_SetChannelPolarity()

```
void Pwm_Hal_SetChannelPolarity (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    Pwm_Hal_ActivePolarityType Polarity )
```

: Pwm_Hal_SetChannelPolarity: Set channel polarity.

Note

Function ID: DES_PWM_API_248

Parameters

| | | |
|----|-----------------|-----------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>Polarity</i> | Output polarity. |

Returns

void

Definition at line 1591 of file Pwm_Hal.c.

4.4.1.56 Pwm_Hal_SetChannelSoftControlLevel()

```
void Pwm_Hal_SetChannelSoftControlLevel (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    Pwm_Hal_OutputLevelType Level )
```

: Pwm_Hal_SetChannelSoftControlLevel: clear all pwm interrupt flag

Note

Function ID: DES_PWM_API_242

Parameters

| | | |
|----|-----------------|-----------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>Level</i> | Output level |

Returns

: void

Definition at line 1470 of file Pwm_Hal.c.

4.4.1.57 Pwm_Hal_SetChannelValue()

```
void Pwm_Hal_SetChannelValue (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelType Channel,
    uint16 Value )
```

: Pwm_Hal_SetChannelValue: Set channel value.

Note

Function ID: DES_PWM_API_206

Parameters

| | | |
|----|-----------------|-----------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Channel</i> | PWM_CHANNEL_0\1\2\3\4\5\6\7 |
| in | <i>Value</i> | Channel value |

Returns

: void

Definition at line 765 of file Pwm_Hal.c.

4.4.1.58 Pwm_Hal_SetClockSource()

```
void Pwm_Hal_SetClockSource (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ClockSourceType ClkSource,
    uint16 ClkPsc )
```

: Pwm_Hal_SetClockSource: Set pwm module clock source and clock psc

Note

Function ID: DES_PWM_API_204

Parameters

| | | |
|----|------------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>ClkSource</i> | clock source type |
| in | <i>ClkPsc</i> | clock prescaler |

Returns

: void

Definition at line 628 of file Pwm_Hal.c.

4.4.1.59 Pwm_Hal_SetDeadtime()

```
void Pwm_Hal_SetDeadtime (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_ChannelPairType ChannelPair,
    Pwm_Hal_DeadTimePscType Prescaler,
    uint16 Value )
```

: Pwm_Hal_SetDeadtime: Set deadtime prescaler & value.

Note

Function ID: DES_PWM_API_249

Parameters

| | | |
|----|--------------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>ChannelPair</i> | PWM_CHANNEL_PAIR_0\1\2\3 |

Parameters

| | | |
|----|------------------|---|
| in | <i>Prescaler</i> | prescaler divider <ul style="list-style-type: none"> • PWM_DEADTIME_DIVID_1 • PWM_DEADTIME_DIVID_4 • PWM_DEADTIME_DIVID_16 |
| in | <i>Value</i> | inserted value <ul style="list-style-type: none"> • 0 ~ 63 |

Returns

void

Note

Function ID: DES_PWM_API_249

Parameters

| | | |
|----|--------------------|---|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>ChannelPair</i> | PWM_CHANNEL_PAIR_0\1\2\3 |
| in | <i>Prescaler</i> | prescaler divider <ul style="list-style-type: none"> • PWM_DEADTIME_DIVID_1 • PWM_DEADTIME_DIVID_4 • PWM_DEADTIME_DIVID_16 |
| in | <i>Value</i> | inserted value <ul style="list-style-type: none"> • 0 ~ 1023 |

Returns

void

Definition at line 1617 of file Pwm_Hal.c.

4.4.1.60 Pwm_Hal_SetDebugMode()

```
void Pwm_Hal_SetDebugMode (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_DebugModeType Mode )
```

: Pwm_Hal_SetDebugMode: Set debug mode.

Note

Function ID: DES_PWM_API_267

Parameters

| | | |
|----|-----------------|--|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Mode</i> | <p>pwm debug mode</p> <ul style="list-style-type: none"> • PWM_DEBUG_MODE_NO_EFFECT • PWM_DEBUG_MODE_COUNTER_STOPPED_OUTPUT_PREVIOUS • PWM_DEBUG_MODE_COUNTER_STOPPED_OUTPUT_HIGH |

Returns

void

Definition at line 2029 of file Pwm_Hal.c.

4.4.1.61 Pwm_Hal_SetFaultInputPolarity()

```
void Pwm_Hal_SetFaultInputPolarity (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_FaultInputIdType FaultInputId,
    Pwm_Hal_ActivePolarityType ActivePolarity )
```

: Pwm_Hal_SetFaultInputPolarity: Set fault input polarity.

Note

Function ID: DES_PWM_API_255

Parameters

| | | |
|----|-----------------------|--|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>FaultInputId</i> | PWM_FAULT_INPUT_0\1\2\3 |
| in | <i>ActivePolarity</i> | PWM_ACTIVE_POLARITY_HIGH / PWM_ACTIVE_POLARITY_LOW |

Returns

void

Definition at line 1792 of file Pwm_Hal.c.

4.4.1.62 Pwm_Hal_SetInitCountValue()

```
void Pwm_Hal_SetInitCountValue (
    Pwm_Hal_InstanceType Instance,
    uint16 Value )
```

: Pwm_Hal_SetInitCountValue: Set pwm module counter begin value.

Note

Function ID: DES_PWM_API_210

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Value</i> | begin value |

Returns

: void

Definition at line 725 of file Pwm_Hal.c.

4.4.1.63 Pwm_Hal_SetInputFilterPsc()

```
void Pwm_Hal_SetInputFilterPsc (
    Pwm_Hal_InstanceType Instance,
    Pwm_Hal_InputFilterPscType Prescaler )
```

: Pwm_Hal_SetInputFilterPsc: Set pwm input filter psc value.

Note

Function ID:

Parameters

| | | |
|----|------------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Prescaler</i> | filter psc(0 ~ 111) |

Returns

: void

Definition at line 1295 of file Pwm_Hal.c.

4.4.1.64 Pwm_Hal_SetMaxCountValue()

```
void Pwm_Hal_SetMaxCountValue (
    Pwm_Hal_InstanceType Instance,
    uint16 Value )
```

: Pwm_Hal_SetMaxCountValue: Set pwm module counter max value.

Note

Function ID: DES_PWM_API_208

Parameters

| | | |
|----|-----------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Value</i> | max value |

Returns

: void

Definition at line 686 of file Pwm_Hal.c.

4.4.1.65 Pwm_Hal_SetOutputMask()

```
void Pwm_Hal_SetOutputMask (
    Pwm_Hal_InstanceType Instance,
    uint8 Mask )
```

: Pwm_Hal_SetOutputMask: Set the mask status for all channels

Note

Function ID: DES_PWM_API_247

Parameters

| | | |
|----|-----------------|--------------------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>Mask</i> | The shielding status of all channels |

Returns

: void

Definition at line 1552 of file Pwm_Hal.c.

4.4.1.66 Pwm_Hal_SetPeriodDither()

```
void Pwm_Hal_SetPeriodDither (
    Pwm_Hal_InstanceType Instance,
    uint8 MaxCountDitherValue )
```

: PWM_DRV_SetMaxCountDitherValue: Set mod dither value.

Note

Function ID: DES_PWM_API_244

Parameters

| | | |
|----|----------------------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>MaxCountDitherValue</i> | mod dither value (0 ~ 31) |

Returns

: void

Definition at line 1512 of file Pwm_Hal.c.

4.4.1.67 Pwm_Hal_SetSoftControlEnableStates()

```
void Pwm_Hal_SetSoftControlEnableStates (
    Pwm_Hal_InstanceType Instance,
    uint8 EnableMask )
```

: Pwm_Hal_SetSoftControlEnableStates: Set the software output enable status for each channel.

Note

Function ID: DES_PWM_API_238

Parameters

| | | |
|----|-------------------|-------------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>EnableMask</i> | Status masks for each channel |

Returns

: void

Definition at line 1377 of file Pwm_Hal.c.

4.4.1.68 Pwm_Hal_SetSoftControlLevels()

```
void Pwm_Hal_SetSoftControlLevels (
    Pwm_Hal_InstanceType Instance,
    uint8 LevelMask )
```

: Pwm_Hal_SetSoftControlLevels: Set software output levels for all channels

Note

Function ID: DES_PWM_API_241

Parameters

| | | |
|----|------------------|---|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>LevelMask</i> | software output levels for all channels |

Returns

: void

Definition at line 1449 of file Pwm_Hal.c.

4.4.1.69 Pwm_Hal_SetTriggerRatio()

```
void Pwm_Hal_SetTriggerRatio (
    Pwm_Hal_InstanceType Instance,
    uint8 TriggerRatio )
```

: Pwm_Hal_SetTriggerRatio: Set the max/match/init trigger ratio.

Note

Function ID: DES_PWM_API_232

Parameters

| | | |
|----|---------------------|---------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>TriggerRatio</i> | 0 ~ 7 |

Returns

: void

Definition at line 1232 of file Pwm_Hal.c.

4.4.1.70 Pwm_Hal_TrigSoftwareSync()

```
void Pwm_Hal_TrigSoftwareSync (
    Pwm_Hal_InstanceType Instance,
    boolean IsTrigger )
```

: Pwm_Hal_TrigSoftwareSync: start software sync trigger

Note

Function ID: DES_PWM_API_226

Parameters

| | | |
|----|------------------|------------------------------|
| in | <i>Instance</i> | Enum Pwm_Hal_InstanceType |
| in | <i>IsTrigger</i> | true->enable, false->disable |

Returns

: void

Definition at line 1115 of file Pwm_Hal.c.

4.5 Pwm_Hal_Types.h File Reference

```
#include "Device_Register.h"
```

Classes

- struct [Pwm_Hal_CommonCfg](#)
pwm common base config struct
- struct [Pwm_Hal_IndependentChnCfg](#)
pwm independent channel config struct
- struct [Pwm_Hal_CombineChnPairCfg](#)
pwm combine mode config struct
- struct [Pwm_Hal_CompareModeChnCfg](#)
pwm compare mode config struct
- struct [Pwm_Hal_OutputCommonCfg](#)
pwm output mode common config struct
- struct [Pwm_Hal_OutputChannelCfg](#)
pwm output channel config struct
- struct [Pwm_Hal_OutputCfg](#)
pwm output mode config struct
- struct [Pwm_Hal_InputCommonCfg](#)
pwm input mode common config struct
- struct [Pwm_Hal_InputChannelCfg](#)
pwm input channel config struct
- struct [Pwm_Hal_InputCfg](#)
pwm input mode config struct
- struct [Pwm_Hal_SyncCfg](#)
PWM sync configuration structure. Please don't use software and hardware trigger simultaneously.
- struct [Pwm_Hal_FaultPinCfg](#)
PWM Fault channel configuration structure.
- struct [Pwm_Hal_FaultCfg](#)
PWM Fault configuration structure.
- struct [Pwm_Hal_QuadPhaseCfg](#)
PWM quadrature decoder phase input configuration structure.
- struct [Pwm_Hal_QuadDecoderCfg](#)
PWM quadrature configuration structure.

Typedefs

- typedef void(* [Pwm_Hal_Callback](#)) ([Pwm_Hal_InstanceType](#) Instance, uint32 Status, const void *UserInfo)

Enumerations

- enum [Pwm_Hal_InstanceType](#) {
[PWM_INSTANCE_0](#) = 0U, [PWM_INSTANCE_1](#), [PWM_INSTANCE_2](#), [PWM_INSTANCE_3](#),
[PWM_INSTANCE_4](#), [PWM_INSTANCE_5](#) }
pwm hw module enumeration
- enum [Pwm_Hal_ChannelType](#) {
[PWM_CHANNEL_0](#) = 0U, [PWM_CHANNEL_1](#), [PWM_CHANNEL_2](#), [PWM_CHANNEL_3](#),
[PWM_CHANNEL_4](#), [PWM_CHANNEL_5](#), [PWM_CHANNEL_6](#), [PWM_CHANNEL_7](#) }
pwm hw channel enumeration
- enum [Pwm_Hal_ChannelPairType](#) {
[PWM_CHANNEL_PAIR_0](#) = 0U, [PWM_CHANNEL_PAIR_1](#), [PWM_CHANNEL_PAIR_2](#), [PWM_CHANNEL_PAIR_3](#),
[PWM_CHANNEL_PAIR_NUM](#) }
pwm hw channel pair enumeration
- enum [Pwm_Hal_ClockSourceType](#) { [PWM_CLK_SOURCE_NONE](#) = 0U, [PWM_CLK_SOURCE_SYSTEM](#), [PWM_CLK_SOURCE_FIXED_FREQUENCY](#), [PWM_CLK_SOURCE_EXTERNAL](#) }
pwm clock source enumeration
- enum [Pwm_Hal_CountModeType](#) { [UP_COUNT](#) = 0U, [UP_DOWN_COUNT](#) }
pwm count mode enumeration
- enum [Pwm_Hal_OutputChnModeType](#) { [OUTPUT_NONE](#) = 0U, [OUTPUT_INDEPENDENT](#), [OUTPUT_COMBINE](#), [OUTPUT_COMPARE](#) }
pwm output channel running mode enumeration
- enum [Pwm_Hal_OutputLevelType](#) { [PWM_LOW_LEVEL](#) = 0U, [PWM_HIGH_LEVEL](#) }
pwm output channel level enumeration
- enum [Pwm_Hal_OutputLevelModeType](#) { [PWM_LOW_TRUE](#) = 0U, [PWM_HIGH_TRUE](#) }
pwm output channel level mode enumeration
- enum [Pwm_Hal_CombineCenterDutyModeType](#) { [PWM_DUTY_MODE_0](#) = 0U, [PWM_DUTY_MODE_1](#) }
Enumeration of pwm duty cycle types in combination center alignment mode.
- enum [Pwm_Hal_ChannelMatchDirType](#) { [PWM_DIR_DOWN](#) = 0U, [PWM_DIR_UP](#) }
pwm combine channel match dir enumeration
- enum [Pwm_Hal_DeadTimePscType](#) { [PWM_DEADTIME_DIVID_1](#) = 1U, [PWM_DEADTIME_DIVID_4](#), [PWM_DEADTIME_DIVID_16](#) }
pwm combine channel dead time psc enumeration
- enum [Pwm_Hal_CompareActionType](#) { [PWM_NONE_OUTPUT](#) = 0U, [PWM_TOGGLE_OUTPUT](#), [PWM_CLEAR_OUTPUT](#), [PWM_SET_OUTPUT](#) }
PWM output compare mode enumeration.
- enum [Pwm_Hal_ActivePolarityType](#) { [PWM_ACTIVE_POLARITY_HIGH](#) = 0U, [PWM_ACTIVE_POLARITY_LOW](#) }
PWM channel output polarity active enumeration.
- enum [Pwm_Hal_EdgeType](#) { [NONE_EDGE](#) = 0U, [RISING_EDGE_DETECT](#), [FALLING_EDGE_DETECT](#), [BOTH_EDGES_DETECT](#) }
pwm detect edge enumeration
- enum [Pwm_Hal_InputChnModeType](#) { [INPUT_NONE](#) = 0U, [INPUT_SINGLE](#), [INPUT_DUAL](#) }
pwm input channel running mode enumeration
- enum [Pwm_Hal_InputEventPscType](#) { [INPUT_EVENT_PSC_1](#) = 0U, [INPUT_EVENT_PSC_2](#), [INPUT_EVENT_PSC_4](#), [INPUT_EVENT_PSC_8](#) }
input event psc
- enum [Pwm_Hal_InputFilterPscType](#) {
[INPUT_FILTER_PSC_1](#) = 0U, [INPUT_FILTER_PSC_2](#), [INPUT_FILTER_PSC_4](#), [INPUT_FILTER_PSC_8](#),
[INPUT_FILTER_PSC_16](#), [INPUT_FILTER_PSC_32](#), [INPUT_FILTER_PSC_64](#), [INPUT_FILTER_PSC_128](#),
[INPUT_FILTER_PSC_256](#), [INPUT_FILTER_PSC_1024](#), [INPUT_FILTER_PSC_2048](#), [INPUT_FILTER_PSC_4096](#)
}

- input filter psc*
- enum `Pwm_Hal_DualInputMeasureType` { `PWM_POSITIVE_PLUSE_WIDTH_MEASURE` = 0U, `PWM_NEGATIVE_PLUSE_WIDTH_MEASURE`, `PWM_RISING_EDGE_PERIOD_MEASURE`, `PWM_FALLING_EDGE_PERIOD_MEASURE` }
- pwm measure type enumeration*
- enum `Pwm_Hal_DualInputContinuousModeType` { `PWM_INPUTCAP_ONESHOT` = 0U, `PWM_INPUTCAP_CONTINUOUS` }
- pwm continuous mode enumeration*
- enum `Pwm_Hal_SyncTriggerMethodType` { `PWM_SYNC_TRIGGER_SOFTWARE` = 0U, `PWM_SYNC_TRIGGER_HARDWARE` }
- pwm sync trigger enumeration*
- enum `Pwm_Hal_SyncModeType` { `PWM_SYNC_MODE_LEGACY` = 0U, `PWM_SYNC_MODE_ENHANCED` }
- pwm sync mode enumeration*
- enum `Pwm_Hal_FaultInputIdType` { `PWM_FAULT_INPUT_0` = 0U, `PWM_FAULT_INPUT_1`, `PWM_FAULT_INPUT_2`, `PWM_FAULT_INPUT_3`, `PWM_FAULT_INPUT_MAX` }
- PWM fault input pin enumeration.*
- enum `Pwm_Hal_FaultCtrlModeType` { `PWM_FAULT_CTRL_NONE` = 0U, `PWM_FAULT_CTRL_MANUAL_EVEN`, `PWM_FAULT_CTRL_MANUAL_ALL`, `PWM_FAULT_CTRL_AUTO` }
- PWM channel fault control mode enumeration.*
- enum `Pwm_Hal_QuadModeType` { `PWM_QUAD_PHASE_ENCODE` = 0U, `PWM_QUAD_COUNT_DIR_ENCODE` }
- PWM quadrature decode mode enumeration.*
- enum `Pwm_Hal_QuadPhasePolarityType` { `PWM_QUAD_PHASE_NORMAL` = 0U, `PWM_QUAD_PHASE_INVERT` }
- PWM quadrature phase polarity enumeration.*
- enum `Pwm_Hal_DebugModeType` { `PWM_DEBUG_OFF` = 0U, `PWM_DEBUG_OUTPUT_PREVIOUS`, `PWM_DEBUG_OUTPUT_HIZ` }
- pwm debug mode enumeration.*

4.5.1 Typedef Documentation

4.5.1.1 Pwm_Hal_Callback

```
typedef void(* Pwm_Hal_Callback) (Pwm_Hal_InstanceType Instance, uint32 Status, const void *UserInfo)
```

Definition at line 342 of file `Pwm_Hal_Types.h`.

4.5.2 Enumeration Type Documentation

4.5.2.1 Pwm_Hal_ActivePolarityType

```
enum Pwm_Hal_ActivePolarityType
```

PWM channel output polarity active enumeration.

Enumerator

| | |
|--------------------------|--|
| PWM_ACTIVE_POLARITY_HIGH | The channel output polarity is active high |
| PWM_ACTIVE_POLARITY_LOW | The channel output polarity is active low |

Definition at line 184 of file Pwm_Hal_Types.h.

4.5.2.2 Pwm_Hal_ChannelMatchDirType

enum [Pwm_Hal_ChannelMatchDirType](#)

pwm combine channel match dir enumeration

Enumerator

| | |
|--------------|--|
| PWM_DIR_DOWN | |
| PWM_DIR_UP | |

Definition at line 154 of file Pwm_Hal_Types.h.

4.5.2.3 Pwm_Hal_ChannelPairType

enum [Pwm_Hal_ChannelPairType](#)

pwm hw channel pair enumeration

Enumerator

| | |
|----------------------|--|
| PWM_CHANNEL_PAIR_0 | |
| PWM_CHANNEL_PAIR_1 | |
| PWM_CHANNEL_PAIR_2 | |
| PWM_CHANNEL_PAIR_3 | |
| PWM_CHANNEL_PAIR_NUM | |

Definition at line 80 of file Pwm_Hal_Types.h.

4.5.2.4 Pwm_Hal_ChannelType

enum [Pwm_Hal_ChannelType](#)

pwm hw channel enumeration

Enumerator

| | |
|--------------------|--|
| PWM_CHANNEL↔ _0 | |
| PWM_CHANNEL↔ _1 | |
| PWM_CHANNEL↔ _2 | |
| PWM_CHANNEL↔ _3 | |
| PWM_CHANNEL↔ _4 | |
| PWM_CHANNEL↔ _5 | |
| PWM_CHANNEL↔ _6 | |
| PWM_CHANNEL↔ _7 | |

Definition at line 65 of file Pwm_Hal_Types.h.

4.5.2.5 Pwm_Hal_ClockSourceType

```
enum Pwm_Hal_ClockSourceType
```

pwm clock source enumeration

Enumerator

| | |
|--------------------------------|---|
| PWM_CLK_SOURCE_NONE | No clock selected, in effect disables the counter |
| PWM_CLK_SOURCE_SYSTEM | System clock |
| PWM_CLK_SOURCE_FIXED_FREQUENCY | Fixed frequency clock |
| PWM_CLK_SOURCE_EXTERNAL | External clock |

Definition at line 92 of file Pwm_Hal_Types.h.

4.5.2.6 Pwm_Hal_CombineCenterDutyModeType

```
enum Pwm_Hal_CombineCenterDutyModeType
```

Enumeration of pwm duty cycle types in combination center alignment mode.

Enumerator

| | |
|----------------------|--|
| PWM_DUTY_MODE↔ _0 | CnV=Cn+1V=CONTIN,output 0% duty cycle;CnV=Cn+1V=MOD,output 100% duty cycle |
| PWM_DUTY_MODE↔ _1 | CnV=Cn+1V=CONTIN,output 100% duty cycle;CnV=Cn+1V=MOD,output 0% duty cycle |

Definition at line 145 of file Pwm_Hal_Types.h.

4.5.2.7 Pwm_Hal_CompareActionType

```
enum Pwm_Hal_CompareActionType
```

PWM output compare mode enumeration.

Enumerator

| | |
|-------------------|---------------|
| PWM_NONE_OUTPUT | No output |
| PWM_TOGGLE_OUTPUT | Toggle output |
| PWM_CLEAR_OUTPUT | Clear output |
| PWM_SET_OUTPUT | Set output |

Definition at line 173 of file Pwm_Hal_Types.h.

4.5.2.8 Pwm_Hal_CountModeType

```
enum Pwm_Hal_CountModeType
```

pwm count mode enumeration

Enumerator

| | |
|---------------|--|
| UP_COUNT | |
| UP_DOWN_COUNT | |

Definition at line 107 of file Pwm_Hal_Types.h.

4.5.2.9 Pwm_Hal_DeadTimePscType

```
enum Pwm_Hal_DeadTimePscType
```

pwm combine channel dead time psc enumeration

Enumerator

| | |
|-----------------------|--|
| PWM_DEADTIME_DIVID_1 | |
| PWM_DEADTIME_DIVID_4 | |
| PWM_DEADTIME_DIVID_16 | |

Definition at line 163 of file Pwm_Hal_Types.h.

4.5.2.10 Pwm_Hal_DebugModeType

```
enum Pwm_Hal_DebugModeType
```

pwm debug mode enumeration.

Enumerator

| | |
|---------------------------|--|
| PWM_DEBUG_OFF | No effect for counter, channel output, CNTIN/MOD/CnV/Cn+1V buffer and registers |
| PWM_DEBUG_OUTPUT_PREVIOUS | Counter stopped, channel output remains previous value, write CNTIN/MOD/CnV/Cn+1V buffer and immediately update to registersn |
| PWM_DEBUG_OUTPUT_HIZ | Counter stopped, channel output remains high resistance value, write CNTIN/MOD/CnV/Cn+1V buffer and immediately update to registersn |

Definition at line 334 of file Pwm_Hal_Types.h.

4.5.2.11 Pwm_Hal_DualInputContinuousModeType

```
enum Pwm_Hal_DualInputContinuousModeType
```

pwm continuous mode enumeration

Enumerator

| | |
|-------------------------|--|
| PWM_INPUTCAP_ONESHOT | |
| PWM_INPUTCAP_CONTINUOUS | |

Definition at line 255 of file Pwm_Hal_Types.h.

4.5.2.12 Pwm_Hal_DualInputMeasureType

```
enum Pwm_Hal_DualInputMeasureType
```

pwm measure type enumeration

Enumerator

| | |
|----------------------------------|--|
| PWM_POSITIVE_PLUSE_WIDTH_MEASURE | |
| PWM_NEGATIVE_PLUSE_WIDTH_MEASURE | |
| PWM_RISING_EDGE_PERIOD_MEASURE | |
| PWM_FALLING_EDGE_PERIOD_MEASURE | |

Definition at line 244 of file Pwm_Hal_Types.h.

4.5.2.13 Pwm_Hal_EdgeType

```
enum Pwm_Hal_EdgeType
```

pwm detect edge enumeration

Enumerator

| | |
|---------------------|--|
| NONE_EDGE | |
| RISING_EDGE_DETECT | |
| FALLING_EDGE_DETECT | |
| BOTH_EDGES_DETECT | |

Definition at line 193 of file Pwm_Hal_Types.h.

4.5.2.14 Pwm_Hal_FaultCtrlModeType

```
enum Pwm_Hal_FaultCtrlModeType
```

PWM channel fault control mode enumeration.

Enumerator

| | |
|----------------------------|--|
| PWM_FAULT_CTRL_NONE | No Fault control |
| PWM_FAULT_CTRL_MANUAL_EVEN | Fault control is enabled for even channels and manual fault clearing |
| PWM_FAULT_CTRL_MANUAL_ALL | Fault control is enabled for all channels and manual fault clearing |
| PWM_FAULT_CTRL_AUTO | Fault control is enabled for all channels and automatic fault clearing |

Definition at line 294 of file Pwm_Hal_Types.h.

4.5.2.15 Pwm_Hal_FaultInputIdType

```
enum Pwm_Hal_FaultInputIdType
```

PWM fault input pin enumeration.

Enumerator

| | |
|---------------------|----------------------------|
| PWM_FAULT_INPUT_0 | PWM fault input input id 0 |
| PWM_FAULT_INPUT_1 | PWM fault input input id 1 |
| PWM_FAULT_INPUT_2 | PWM fault input input id 2 |
| PWM_FAULT_INPUT_3 | PWM fault input input id 3 |
| PWM_FAULT_INPUT_MAX | Invalid fault input id |

Definition at line 282 of file Pwm_Hal_Types.h.

4.5.2.16 Pwm_Hal_InputChnModeType

```
enum Pwm_Hal_InputChnModeType
```

pwm input channel running mode enumeration

Enumerator

| | |
|--------------|--|
| INPUT_NONE | |
| INPUT_SINGLE | |
| INPUT_DUAL | |

Definition at line 204 of file Pwm_Hal_Types.h.

4.5.2.17 Pwm_Hal_InputEventPscType

```
enum Pwm_Hal_InputEventPscType
```

input event psc

Enumerator

| | |
|------------------------|--|
| INPUT_EVENT_PSC↔ _1 | |
| INPUT_EVENT_PSC↔ _2 | |
| INPUT_EVENT_PSC↔ _4 | |
| INPUT_EVENT_PSC↔ _8 | |

Definition at line 214 of file Pwm_Hal_Types.h.

4.5.2.18 Pwm_Hal_InputFilterPscType

```
enum Pwm_Hal_InputFilterPscType
```

input filter psc

Enumerator

| | |
|---------------------|--|
| INPUT_FILTER_PSC_1 | |
| INPUT_FILTER_PSC_2 | |
| INPUT_FILTER_PSC_4 | |
| INPUT_FILTER_PSC_8 | |
| INPUT_FILTER_PSC_16 | |
| INPUT_FILTER_PSC_32 | |
| INPUT_FILTER_PSC_64 | |

Enumerator

| | |
|-----------------------|--|
| INPUT_FILTER_PSC_128 | |
| INPUT_FILTER_PSC_256 | |
| INPUT_FILTER_PSC_1024 | |
| INPUT_FILTER_PSC_2048 | |
| INPUT_FILTER_PSC_4096 | |

Definition at line 225 of file Pwm_Hal_Types.h.

4.5.2.19 Pwm_Hal_InstanceType

```
enum Pwm_Hal_InstanceType
```

pwm hw module enumeration

Enumerator

| | |
|---------------------|--|
| PWM_INSTANCE↵ _0 | |
| PWM_INSTANCE↵ _1 | |
| PWM_INSTANCE↵ _2 | |
| PWM_INSTANCE↵ _3 | |
| PWM_INSTANCE↵ _4 | |
| PWM_INSTANCE↵ _5 | |

Definition at line 48 of file Pwm_Hal_Types.h.

4.5.2.20 Pwm_Hal_OutputChnModeType

```
enum Pwm_Hal_OutputChnModeType
```

pwm output channel running mode enumeration

Enumerator

| | |
|--------------------|--|
| OUTPUT_NONE | |
| OUTPUT_INDEPENDENT | |
| OUTPUT_COMBINE | |
| OUTPUT_COMPARE | |

Definition at line 116 of file Pwm_Hal_Types.h.

4.5.2.21 Pwm_Hal_OutputLevelModeType

enum [Pwm_Hal_OutputLevelModeType](#)

pwm output channel level mode enumeration

Enumerator

| | |
|---------------|--|
| PWM_LOW_TRUE | |
| PWM_HIGH_TRUE | |

Definition at line 136 of file Pwm_Hal_Types.h.

4.5.2.22 Pwm_Hal_OutputLevelType

enum [Pwm_Hal_OutputLevelType](#)

pwm output channel level enumeration

Enumerator

| | |
|----------------|--|
| PWM_LOW_LEVEL | |
| PWM_HIGH_LEVEL | |

Definition at line 127 of file Pwm_Hal_Types.h.

4.5.2.23 Pwm_Hal_QuadModeType

enum [Pwm_Hal_QuadModeType](#)

PWM quadrature decode mode enumeration.

Enumerator

| | |
|---------------------------|-------------------------------------|
| PWM_QUAD_PHASE_ENCODE | Phase encoding mode |
| PWM_QUAD_COUNT_DIR_ENCODE | Counter and direction encoding mode |

Definition at line 305 of file Pwm_Hal_Types.h.

4.5.2.24 Pwm_Hal_QuadPhasePolarityType

enum [Pwm_Hal_QuadPhasePolarityType](#)

PWM quadrature phase polarity enumeration.

Enumerator

| | |
|-----------------------|------------------------------------|
| PWM_QUAD_PHASE_NORMAL | Phase input signal is not inverted |
| PWM_QUAD_PHASE_INVERT | Phase input signal is inverted |

Definition at line 314 of file Pwm_Hal_Types.h.

4.5.2.25 Pwm_Hal_SyncModeType

enum [Pwm_Hal_SyncModeType](#)

pwm sync mode enumeration

Enumerator

| | |
|------------------------|--|
| PWM_SYNC_MODE_LEGACY | |
| PWM_SYNC_MODE_ENHANCED | |

Definition at line 273 of file Pwm_Hal_Types.h.

4.5.2.26 Pwm_Hal_SyncTriggerMethodType

enum [Pwm_Hal_SyncTriggerMethodType](#)

pwm sync trigger enumeration

Enumerator

| | |
|---------------------------|--|
| PWM_SYNC_TRIGGER_SOFTWARE | |
| PWM_SYNC_TRIGGER_HARDWARE | |

Definition at line 264 of file Pwm_Hal_Types.h.

Index

AC784xx_API_Reference_Manual_PWM.pdf, [33](#)

AC784xx_Pwm_Reg.h, [33](#)

[Pwm_Reg_ChooseBufferOrRegisterToRead, 36](#)
[Pwm_Reg_ClearChannelEventFlag, 36](#)
[Pwm_Reg_ClearChannelInterruptFlag, 36](#)
[Pwm_Reg_ClearChannelTriggerFlag, 37](#)
[Pwm_Reg_ClearFaultFlag, 37](#)
[Pwm_Reg_ClearFaultPinFlag, 37](#)
[Pwm_Reg_ClearOverflowFlag, 37](#)
[Pwm_Reg_ClearQuadPhaseZFlag, 37](#)
[Pwm_Reg_DisableHwTriggerSyncAfterTriggered, 37](#)
[Pwm_Reg_EnableCHOSWCERSync, 39](#)
[Pwm_Reg_EnableCNTINSync, 39](#)
[Pwm_Reg_EnableChannelCombine, 38](#)
[Pwm_Reg_EnableChannelDmaReq, 38](#)
[Pwm_Reg_EnableChannelEventReset, 38](#)
[Pwm_Reg_EnableChannelHizOutput, 38](#)
[Pwm_Reg_EnableChannelInterrupt, 38](#)
[Pwm_Reg_EnableChannelMatchTrigger, 39](#)
[Pwm_Reg_EnableChannelOutputMask, 39](#)
[Pwm_Reg_EnableChannelSoftControl, 39](#)
[Pwm_Reg_EnableDmaTransfer, 40](#)
[Pwm_Reg_EnableFaultHizOutput, 40](#)
[Pwm_Reg_EnableFaultInputFilter, 40](#)
[Pwm_Reg_EnableFaultInterrupt, 40](#)
[Pwm_Reg_EnableFaultPinInput, 40](#)
[Pwm_Reg_EnableGlobalTimeBase, 41](#)
[Pwm_Reg_EnableGlobalTimeBaseOutput, 41](#)
[Pwm_Reg_EnableHall, 41](#)
[Pwm_Reg_EnableINVCERSync, 41](#)
[Pwm_Reg_EnableInitTrigger, 41](#)
[Pwm_Reg_EnableMaxLoadingPointSync, 42](#)
[Pwm_Reg_EnableMaxTrigger, 42](#)
[Pwm_Reg_EnableMinLoadingPointSync, 42](#)
[Pwm_Reg_EnableOutputInit, 42](#)
[Pwm_Reg_EnableOutputMaskSync, 42](#)
[Pwm_Reg_EnableOverflowDmaReq, 43](#)
[Pwm_Reg_EnableOverflowEvent, 43](#)
[Pwm_Reg_EnableOverflowInterrupt, 43](#)
[Pwm_Reg_EnablePOLSync, 45](#)
[Pwm_Reg_EnablePairChComplement, 43](#)
[Pwm_Reg_EnablePairChDeadtime, 43](#)
[Pwm_Reg_EnablePairChDualEdgeCapture, 44](#)
[Pwm_Reg_EnablePairChFaultCtrl, 44](#)
[Pwm_Reg_EnablePairChInvert, 44](#)
[Pwm_Reg_EnablePairChSymmetric, 44](#)
[Pwm_Reg_EnablePairChValueSync, 44](#)
[Pwm_Reg_EnablePairPulseWidthMeasurement, 45](#)
[Pwm_Reg_EnableQuadDecoder, 45](#)
[Pwm_Reg_EnableQuadPhaseZResetCnt, 45](#)
[Pwm_Reg_EnableSync, 45](#)
[Pwm_Reg_EnableSyncBYP, 46](#)

[Pwm_Reg_EnableSyncHwTriggerSrc, 46](#)
[Pwm_Reg_EnableUnderflowEvent, 46](#)
[Pwm_Reg_EnableWriteProtection, 46](#)
[Pwm_Reg_GetAllChannelLevel, 46](#)
[Pwm_Reg_GetBase, 47](#)
[Pwm_Reg_GetChannelEventFlag, 47](#)
[Pwm_Reg_GetChannelInterruptFlag, 47](#)
[Pwm_Reg_GetChannelTriggerFlag, 47](#)
[Pwm_Reg_GetChannelValue, 47](#)
[Pwm_Reg_GetCountValue, 47](#)
[Pwm_Reg_GetFaultFlag, 48](#)
[Pwm_Reg_GetFaultInputStatus, 48](#)
[Pwm_Reg_GetFaultPinFlag, 48](#)
[Pwm_Reg_GetHallStatus, 48](#)
[Pwm_Reg_GetInitCountValue, 48](#)
[Pwm_Reg_GetMaxCountValue, 48](#)
[Pwm_Reg_GetOutputMask, 49](#)
[Pwm_Reg_GetOverflowDir, 49](#)
[Pwm_Reg_GetOverflowFlag, 49](#)
[Pwm_Reg_GetQuadCountDir, 49](#)
[Pwm_Reg_GetQuadOverflowDir, 49](#)
[Pwm_Reg_GetQuadPhaseZFlag, 49](#)
[Pwm_Reg_IsPairChDualEdgeCaptureEnabled, 50](#)
[Pwm_Reg_ReadChannelSoftControlState, 50](#)
[Pwm_Reg_ResetCounter, 50](#)
[Pwm_Reg_SetCHOSWCERSyncTriggerMode, 52](#)
[Pwm_Reg_SetCNTSyncTriggerMode, 53](#)
[Pwm_Reg_SetCaptureEventPsc, 50](#)
[Pwm_Reg_SetChannelELSR, 50](#)
[Pwm_Reg_SetChannelInputFilterVal, 51](#)
[Pwm_Reg_SetChannelIMSR, 51](#)
[Pwm_Reg_SetChannelMatchDir, 51](#)
[Pwm_Reg_SetChannelMatchDither, 51](#)
[Pwm_Reg_SetChannelMatchDitherReg, 51](#)
[Pwm_Reg_SetChannelOutputInitLevel, 52](#)
[Pwm_Reg_SetChannelPolarity, 52](#)
[Pwm_Reg_SetChannelSoftControlLevel, 52](#)
[Pwm_Reg_SetChannelValue, 52](#)
[Pwm_Reg_SetClockPsc, 53](#)
[Pwm_Reg_SetClockSource, 53](#)
[Pwm_Reg_SetCntOverflowFreq, 53](#)
[Pwm_Reg_SetCombineCenterDutyType, 53](#)
[Pwm_Reg_SetCountMode, 54](#)
[Pwm_Reg_SetDebugMode, 54](#)
[Pwm_Reg_SetDmaTransferLen, 54](#)
[Pwm_Reg_SetFaultInputFilterValue, 54](#)
[Pwm_Reg_SetFaultInputPolarity, 54](#)
[Pwm_Reg_SetFaultMode, 55](#)
[Pwm_Reg_SetINVCERSyncTriggerMode, 55](#)
[Pwm_Reg_SetInitCountValue, 55](#)
[Pwm_Reg_SetInputFilterPsc, 55](#)
[Pwm_Reg_SetMCVRSyncMode, 56](#)

- Pwm_Reg_SetMaxCountValue, 55
- Pwm_Reg_SetOMCRSyncTriggerMode, 56
- Pwm_Reg_SetOutputMask, 56
- Pwm_Reg_SetPOLTriggerMode, 57
- Pwm_Reg_SetPairChDeadtime, 56
- Pwm_Reg_SetPairChDualEdgeCapture, 56
- Pwm_Reg_SetPeriodDither, 57
- Pwm_Reg_SetQuadEncodeMode, 57
- Pwm_Reg_SetQuadPhaseAPolarity, 57
- Pwm_Reg_SetQuadPhaseBPolarity, 57
- Pwm_Reg_SetQuadPhaseZPolarity, 58
- Pwm_Reg_SetSoftControlEnableStatesMask, 58
- Pwm_Reg_SetSoftControlLevelMask, 58
- Pwm_Reg_SetSyncMode, 58
- Pwm_Reg_SetSyncType, 58
- Pwm_Reg_SetTriggerRatio, 59
- Pwm_Reg_TrigSoftwareSync, 59
- Action
 - Pwm_Hal_CompareModeChnCf, 11
- ActivePolarity
 - Pwm_Hal_CompareModeChnCf, 11
 - Pwm_Hal_IndependentChnCf, 16
- BaseCf
 - Pwm_Hal_InputCommonCf, 21
 - Pwm_Hal_OutputCommonCf, 24
 - Pwm_Hal_QuadDecoderCf, 26
- CaptureEdge
 - Pwm_Hal_InputChannelCf, 19
- Ch1stActivePolarity
 - Pwm_Hal_CombineChnPairCf, 4
- Ch1stDither
 - Pwm_Hal_CombineChnPairCf, 4
- Ch1stInitLevel
 - Pwm_Hal_CombineChnPairCf, 4
- Ch1stMatchDir
 - Pwm_Hal_CombineChnPairCf, 4
- Ch1stValue
 - Pwm_Hal_CombineChnPairCf, 4
- Ch2ndActivePolarity
 - Pwm_Hal_CombineChnPairCf, 4
- Ch2ndDither
 - Pwm_Hal_CombineChnPairCf, 4
- Ch2ndInitLevel
 - Pwm_Hal_CombineChnPairCf, 5
- Ch2ndMatchDir
 - Pwm_Hal_CombineChnPairCf, 5
- Ch2ndValue
 - Pwm_Hal_CombineChnPairCf, 5
- Channel
 - Pwm_Hal_InputChannelCf, 19
 - Pwm_Hal_OutputChannelCf, 23
- ChannelCallback
 - Pwm_Hal_CommonCf, 8
- ChannelCf
 - Pwm_Hal_InputCf, 17
 - Pwm_Hal_OutputCf, 22
- ChannelMode
 - Pwm_Hal_InputChannelCf, 19
 - Pwm_Hal_OutputChannelCf, 23
- ChannelNum
 - Pwm_Hal_InputCf, 18
 - Pwm_Hal_OutputCf, 22
- ChnDither
 - Pwm_Hal_IndependentChnCf, 16
- ChnPairCf
 - Pwm_Hal_OutputChannelCf, 23
- ChnValue
 - Pwm_Hal_CompareModeChnCf, 11
 - Pwm_Hal_IndependentChnCf, 16
- ClockSource
 - Pwm_Hal_CommonCf, 8
- CombineCenterDutyMode
 - Pwm_Hal_OutputCommonCf, 24
- CommonCf
 - Pwm_Hal_InputCf, 18
 - Pwm_Hal_OutputCf, 22
- CompareChnCf
 - Pwm_Hal_OutputChannelCf, 23
- CountMode
 - Pwm_Hal_OutputCommonCf, 24
- DeadTimeValue
 - Pwm_Hal_CombineChnPairCf, 5
- DisableHwTriggerAfterTriggered
 - Pwm_Hal_SyncCf, 29
- DmaTransLen
 - Pwm_Hal_CommonCf, 8
- DualInputContinuousMode
 - Pwm_Hal_InputChannelCf, 19
- DualInputMeasureType
 - Pwm_Hal_InputChannelCf, 19
- EnableCh1stEventDmaReq
 - Pwm_Hal_CombineChnPairCf, 5
- EnableCh1stInterrupt
 - Pwm_Hal_CombineChnPairCf, 5
- EnableCh1stMatchTrigger
 - Pwm_Hal_CombineChnPairCf, 6
- EnableCh2ndEventDmaReq
 - Pwm_Hal_CombineChnPairCf, 6
- EnableCh2ndInterrupt
 - Pwm_Hal_CombineChnPairCf, 6
- EnableCh2ndMatchTrigger
 - Pwm_Hal_CombineChnPairCf, 6
- EnableChComplementation
 - Pwm_Hal_CombineChnPairCf, 6
- EnableChannelOutputCtrl
 - Pwm_Hal_FaultCf, 12
- EnableChnEventDmaReq
 - Pwm_Hal_CompareModeChnCf, 11
 - Pwm_Hal_IndependentChnCf, 16
 - Pwm_Hal_InputChannelCf, 19
- EnableCounterInitSync
 - Pwm_Hal_SyncCf, 29
- EnableCounterReset
 - Pwm_Hal_InputChannelCf, 19
- EnableDeadTime
 - Pwm_Hal_CombineChnPairCf, 6
- EnableDmaTransLen
 - Pwm_Hal_CommonCf, 8

- EnableDualChannelInvertSync
 - Pwm_Hal_SyncCfg, [29](#)
- EnableFaultFilter
 - Pwm_Hal_FaultPinCfg, [14](#)
- EnableFaultInput
 - Pwm_Hal_FaultPinCfg, [15](#)
- EnableHall
 - Pwm_Hal_InputCommonCfg, [21](#)
- EnableHiz
 - Pwm_Hal_FaultCfg, [12](#)
- EnableHwSync0
 - Pwm_Hal_SyncCfg, [30](#)
- EnableHwSync1
 - Pwm_Hal_SyncCfg, [30](#)
- EnableHwSync2
 - Pwm_Hal_SyncCfg, [30](#)
- EnableInitTrigger
 - Pwm_Hal_OutputCommonCfg, [25](#)
- EnableInterrupt
 - Pwm_Hal_CompareModeChnCfg, [11](#)
 - Pwm_Hal_FaultCfg, [13](#)
 - Pwm_Hal_IndependentChnCfg, [16](#)
 - Pwm_Hal_InputChannelCfg, [20](#)
- EnableMatchTrigger
 - Pwm_Hal_CompareModeChnCfg, [11](#)
 - Pwm_Hal_IndependentChnCfg, [16](#)
- EnableMaxLoadingPoint
 - Pwm_Hal_SyncCfg, [30](#)
- EnableMaxTrigger
 - Pwm_Hal_OutputCommonCfg, [25](#)
- EnableMinLoadingPoint
 - Pwm_Hal_SyncCfg, [30](#)
- EnableOutputMaskSync
 - Pwm_Hal_SyncCfg, [31](#)
- EnableOverflowDmaReq
 - Pwm_Hal_CommonCfg, [8](#)
- EnableOverflowEvent
 - Pwm_Hal_CommonCfg, [8](#)
- EnableOverflowInterrupt
 - Pwm_Hal_CommonCfg, [9](#)
- EnablePairedChnValSync
 - Pwm_Hal_SyncCfg, [31](#)
- EnablePhaseZReset
 - Pwm_Hal_QuadDecoderCfg, [26](#)
- EnablePolaritySync
 - Pwm_Hal_SyncCfg, [31](#)
- EnablePulseWidthMeasure
 - Pwm_Hal_InputChannelCfg, [20](#)
- EnableQuad
 - Pwm_Hal_QuadDecoderCfg, [26](#)
- EnableSwOutputCtrlSync
 - Pwm_Hal_SyncCfg, [31](#)
- EnableSymmetric
 - Pwm_Hal_CombineChnPairCfg, [7](#)
- EnableSync
 - Pwm_Hal_SyncCfg, [31](#)
- EnableSyncBypass
 - Pwm_Hal_SyncCfg, [32](#)
- EnableUnderflowEvent
 - Pwm_Hal_CommonCfg, [9](#)
- EventPsc
 - Pwm_Hal_InputChannelCfg, [20](#)
- FaultCallback
 - Pwm_Hal_FaultCfg, [13](#)
- FaultCtrlMode
 - Pwm_Hal_FaultCfg, [13](#)
- FaultPinCfg
 - Pwm_Hal_FaultCfg, [13](#)
- FaultPolarity
 - Pwm_Hal_FaultPinCfg, [15](#)
- FilterPsc
 - Pwm_Hal_FaultCfg, [13](#)
 - Pwm_Hal_InputCommonCfg, [21](#)
 - Pwm_Hal_QuadDecoderCfg, [26](#)
- FilterValue
 - Pwm_Hal_FaultCfg, [14](#)
 - Pwm_Hal_InputChannelCfg, [20](#)
 - Pwm_Hal_QuadPhaseCfg, [28](#)
- IndependentChnCfg
 - Pwm_Hal_OutputChannelCfg, [23](#)
- InitLevel
 - Pwm_Hal_CompareModeChnCfg, [11](#)
 - Pwm_Hal_IndependentChnCfg, [17](#)
- InitOutput
 - Pwm_Hal_OutputCommonCfg, [25](#)
- LevelMode
 - Pwm_Hal_CombineChnPairCfg, [7](#)
 - Pwm_Hal_IndependentChnCfg, [17](#)
- MaxCount
 - Pwm_Hal_CommonCfg, [9](#)
- MinCount
 - Pwm_Hal_CommonCfg, [9](#)
- Mode
 - Pwm_Hal_QuadDecoderCfg, [27](#)
- OverflowCallback
 - Pwm_Hal_CommonCfg, [9](#)
- OverflowFreq
 - Pwm_Hal_CommonCfg, [9](#)
- PWM0_Channel_IRQHandler
 - Pwm_Hal.c, [63](#)
- PWM0_Fault_IRQHandler
 - Pwm_Hal.c, [63](#)
- PWM0_Overflow_IRQHandler
 - Pwm_Hal.c, [64](#)
- PWM1_Channel_IRQHandler
 - Pwm_Hal.c, [64](#)
- PWM1_Fault_IRQHandler
 - Pwm_Hal.c, [64](#)
- PWM1_Overflow_IRQHandler
 - Pwm_Hal.c, [65](#)
- PWM2_Channel_IRQHandler
 - Pwm_Hal.c, [65](#)
- PWM2_Fault_IRQHandler
 - Pwm_Hal.c, [65](#)
- PWM2_Overflow_IRQHandler
 - Pwm_Hal.c, [66](#)
- PWM3_Channel_IRQHandler

- Pwm_Hal.c, [66](#)
- PWM3_Fault_IRQHandler
 - Pwm_Hal.c, [66](#)
- PWM3_Overflow_IRQHandler
 - Pwm_Hal.c, [67](#)
- PWM4_Channel_IRQHandler
 - Pwm_Hal.c, [67](#)
- PWM4_Fault_IRQHandler
 - Pwm_Hal.c, [67](#)
- PWM4_Overflow_IRQHandler
 - Pwm_Hal.c, [68](#)
- PWM5_Channel_IRQHandler
 - Pwm_Hal.c, [68](#)
- PWM5_Fault_IRQHandler
 - Pwm_Hal.c, [68](#)
- PWM5_Overflow_IRQHandler
 - Pwm_Hal.c, [69](#)
- PeriodDither
 - Pwm_Hal_CommonCfg, [10](#)
- PhaseAConfig
 - Pwm_Hal_QuadDecoderCfg, [27](#)
- PhaseBConfig
 - Pwm_Hal_QuadDecoderCfg, [27](#)
- PhaseZConfig
 - Pwm_Hal_QuadDecoderCfg, [27](#)
- Polarity
 - Pwm_Hal_QuadPhaseCfg, [28](#)
- Prescaler
 - Pwm_Hal_CommonCfg, [10](#)
- Pwm_Hal.c, [59](#)
 - PWM0_Channel_IRQHandler, [63](#)
 - PWM0_Fault_IRQHandler, [63](#)
 - PWM0_Overflow_IRQHandler, [64](#)
 - PWM1_Channel_IRQHandler, [64](#)
 - PWM1_Fault_IRQHandler, [64](#)
 - PWM1_Overflow_IRQHandler, [65](#)
 - PWM2_Channel_IRQHandler, [65](#)
 - PWM2_Fault_IRQHandler, [65](#)
 - PWM2_Overflow_IRQHandler, [66](#)
 - PWM3_Channel_IRQHandler, [66](#)
 - PWM3_Fault_IRQHandler, [66](#)
 - PWM3_Overflow_IRQHandler, [67](#)
 - PWM4_Channel_IRQHandler, [67](#)
 - PWM4_Fault_IRQHandler, [67](#)
 - PWM4_Overflow_IRQHandler, [68](#)
 - PWM5_Channel_IRQHandler, [68](#)
 - PWM5_Fault_IRQHandler, [68](#)
 - PWM5_Overflow_IRQHandler, [69](#)
 - Pwm_Hal_ClearChannelInterruptFlag, [69](#)
 - Pwm_Hal_ClearDetectEventFlag, [70](#)
 - Pwm_Hal_ClearFaultFlag, [70](#)
 - Pwm_Hal_ClearFaultPinFlag, [71](#)
 - Pwm_Hal_ClearOverflowFlag, [71](#)
 - Pwm_Hal_ClearQuadPhaseZFlag, [72](#)
 - Pwm_Hal_DeInit, [72](#)
 - Pwm_Hal_EnableChannelDmaRequest, [72](#)
 - Pwm_Hal_EnableChannelEventReset, [73](#)
 - Pwm_Hal_EnableChannelHizOutput, [73](#)
 - Pwm_Hal_EnableChannelInterrupt, [74](#)
 - Pwm_Hal_EnableChannelMatchTrigger, [74](#)
 - Pwm_Hal_EnableChannelOutputMask, [75](#)
 - Pwm_Hal_EnableChannelPairInvert, [75](#)
 - Pwm_Hal_EnableChannelPairSymmetric, [76](#)
 - Pwm_Hal_EnableChannelSoftControl, [76](#)
 - Pwm_Hal_EnableCombineChannelSoftControl, [77](#)
 - Pwm_Hal_EnableFaultInputFilter, [77](#)
 - Pwm_Hal_EnableFaultPinInput, [78](#)
 - Pwm_Hal_EnableGlobalTimeBase, [79](#)
 - Pwm_Hal_EnableGlobalTimeBaseOutput, [79](#)
 - Pwm_Hal_EnableInitTrigger, [79](#)
 - Pwm_Hal_EnableMaxTrigger, [80](#)
 - Pwm_Hal_EnableOverflowEvent, [80](#)
 - Pwm_Hal_EnableOverflowInterrupt, [81](#)
 - Pwm_Hal_EnableSyncBypass, [81](#)
 - Pwm_Hal_EnableUnderflowEvent, [82](#)
 - Pwm_Hal_EnableWriteProtection, [82](#)
 - Pwm_Hal_GetAllChannelLevel, [83](#)
 - Pwm_Hal_GetChannelInterruptFlag, [83](#)
 - Pwm_Hal_GetChannelValue, [84](#)
 - Pwm_Hal_GetCountValue, [84](#)
 - Pwm_Hal_GetDetectEventFlag, [85](#)
 - Pwm_Hal_GetFaultFlag, [85](#)
 - Pwm_Hal_GetFaultPinFlag, [86](#)
 - Pwm_Hal_GetHallStatus, [86](#)
 - Pwm_Hal_GetInitCountValue, [87](#)
 - Pwm_Hal_GetMaxCountValue, [87](#)
 - Pwm_Hal_GetOutputMask, [87](#)
 - Pwm_Hal_GetOverflowDir, [88](#)
 - Pwm_Hal_GetOverflowFlag, [88](#)
 - Pwm_Hal_GetQuadCountingDir, [89](#)
 - Pwm_Hal_GetQuadOverflowDir, [89](#)
 - Pwm_Hal_GetQuadPhaseZFlag, [90](#)
 - Pwm_Hal_InitFaultControl, [90](#)
 - Pwm_Hal_InitInputMode, [91](#)
 - Pwm_Hal_InitOutputMode, [91](#)
 - Pwm_Hal_InitQuadDecoderMode, [92](#)
 - Pwm_Hal_InitSyncConfigSet, [92](#)
 - Pwm_Hal_ResetCounter, [93](#)
 - Pwm_Hal_SetChannelCaptureEdge, [93](#)
 - Pwm_Hal_SetChannelCompareAction, [94](#)
 - Pwm_Hal_SetChannelInputFilterVal, [94](#)
 - Pwm_Hal_SetChannelMatchDither, [95](#)
 - Pwm_Hal_SetChannelPolarity, [95](#)
 - Pwm_Hal_SetChannelSoftControlLevel, [96](#)
 - Pwm_Hal_SetChannelValue, [96](#)
 - Pwm_Hal_SetClockSource, [97](#)
 - Pwm_Hal_SetDeadtime, [97](#)
 - Pwm_Hal_SetDebugMode, [98](#)
 - Pwm_Hal_SetFaultInputPolarity, [98](#)
 - Pwm_Hal_SetInitCountValue, [99](#)
 - Pwm_Hal_SetInputFilterPsc, [99](#)
 - Pwm_Hal_SetMaxCountValue, [100](#)
 - Pwm_Hal_SetOutputMask, [100](#)
 - Pwm_Hal_SetPeriodDither, [101](#)
 - Pwm_Hal_SetSoftControlEnableStates, [101](#)
 - Pwm_Hal_SetSoftControlLevels, [102](#)
 - Pwm_Hal_SetTriggerRatio, [102](#)
 - Pwm_Hal_TrigSoftwareSync, [103](#)
- Pwm_Hal.h, [103](#)
 - Pwm_Hal_ClearChannelInterruptFlag, [107](#)
 - Pwm_Hal_ClearDetectEventFlag, [107](#)
 - Pwm_Hal_ClearFaultFlag, [108](#)

Pwm_Hal_ClearFaultPinFlag, 108
Pwm_Hal_ClearOverflowFlag, 109
Pwm_Hal_ClearQuadPhaseZFlag, 109
Pwm_Hal_DeInit, 109
Pwm_Hal_EnableChannelDmaRequest, 110
Pwm_Hal_EnableChannelEventReset, 110
Pwm_Hal_EnableChannelHizOutput, 111
Pwm_Hal_EnableChannelInterrupt, 111
Pwm_Hal_EnableChannelMatchTrigger, 112
Pwm_Hal_EnableChannelOutputMask, 112
Pwm_Hal_EnableChannelPairInvert, 113
Pwm_Hal_EnableChannelPairSymmetric, 113
Pwm_Hal_EnableChannelSoftControl, 114
Pwm_Hal_EnableCombineChannelSoftControl, 114
Pwm_Hal_EnableFaultInputFilter, 115
Pwm_Hal_EnableFaultPinInput, 115
Pwm_Hal_EnableGlobalTimeBase, 116
Pwm_Hal_EnableGlobalTimeBaseOutput, 116
Pwm_Hal_EnableInitTrigger, 117
Pwm_Hal_EnableMaxTrigger, 117
Pwm_Hal_EnableOverflowEvent, 118
Pwm_Hal_EnableOverflowInterrupt, 118
Pwm_Hal_EnableSyncBypass, 119
Pwm_Hal_EnableUnderflowEvent, 119
Pwm_Hal_EnableWriteProtection, 120
Pwm_Hal_GetAllChannelLevel, 120
Pwm_Hal_GetChannelInterruptFlag, 121
Pwm_Hal_GetChannelValue, 121
Pwm_Hal_GetCountValue, 122
Pwm_Hal_GetDetectEventFlag, 122
Pwm_Hal_GetFaultFlag, 123
Pwm_Hal_GetFaultPinFlag, 123
Pwm_Hal_GetHallStatus, 124
Pwm_Hal_GetInitCountValue, 124
Pwm_Hal_GetMaxCountValue, 124
Pwm_Hal_GetOutputMask, 125
Pwm_Hal_GetOverflowDir, 125
Pwm_Hal_GetOverflowFlag, 126
Pwm_Hal_GetQuadCountingDir, 126
Pwm_Hal_GetQuadOverflowDir, 127
Pwm_Hal_GetQuadPhaseZFlag, 127
Pwm_Hal_InitFaultControl, 128
Pwm_Hal_InitInputMode, 128
Pwm_Hal_InitOutputMode, 129
Pwm_Hal_InitQuadDecoderMode, 129
Pwm_Hal_InitSyncConfigSet, 130
Pwm_Hal_ResetCounter, 130
Pwm_Hal_SetChannelCaptureEdge, 131
Pwm_Hal_SetChannelCompareAction, 131
Pwm_Hal_SetChannelInputFilterVal, 132
Pwm_Hal_SetChannelMatchDither, 132
Pwm_Hal_SetChannelPolarity, 133
Pwm_Hal_SetChannelSoftControlLevel, 134
Pwm_Hal_SetChannelValue, 134
Pwm_Hal_SetClockSource, 135
Pwm_Hal_SetDeadtime, 135
Pwm_Hal_SetDebugMode, 136
Pwm_Hal_SetFaultInputPolarity, 137
Pwm_Hal_SetInitCountValue, 137
Pwm_Hal_SetInputFilterPsc, 138
Pwm_Hal_SetMaxCountValue, 138
Pwm_Hal_SetOutputMask, 139
Pwm_Hal_SetPeriodDither, 139
Pwm_Hal_SetSoftControlEnableStates, 140
Pwm_Hal_SetSoftControlLevels, 140
Pwm_Hal_SetTriggerRatio, 141
Pwm_Hal_TrigSoftwareSync, 141
Pwm_Hal_ActivePolarityType
 Pwm_Hal_Types.h, 144
Pwm_Hal_Callback
 Pwm_Hal_Types.h, 144
Pwm_Hal_ChannelMatchDirType
 Pwm_Hal_Types.h, 145
Pwm_Hal_ChannelPairType
 Pwm_Hal_Types.h, 145
Pwm_Hal_ChannelType
 Pwm_Hal_Types.h, 145
Pwm_Hal_ClearChannelInterruptFlag
 Pwm_Hal.c, 69
 Pwm_Hal.h, 107
Pwm_Hal_ClearDetectEventFlag
 Pwm_Hal.c, 70
 Pwm_Hal.h, 107
Pwm_Hal_ClearFaultFlag
 Pwm_Hal.c, 70
 Pwm_Hal.h, 108
Pwm_Hal_ClearFaultPinFlag
 Pwm_Hal.c, 71
 Pwm_Hal.h, 108
Pwm_Hal_ClearOverflowFlag
 Pwm_Hal.c, 71
 Pwm_Hal.h, 109
Pwm_Hal_ClearQuadPhaseZFlag
 Pwm_Hal.c, 72
 Pwm_Hal.h, 109
Pwm_Hal_ClockSourceType
 Pwm_Hal_Types.h, 146
Pwm_Hal_CombineCenterDutyModeType
 Pwm_Hal_Types.h, 146
Pwm_Hal_CombineChnPairCfg, 3
 Ch1stActivePolarity, 4
 Ch1stDither, 4
 Ch1stInitLevel, 4
 Ch1stMatchDir, 4
 Ch1stValue, 4
 Ch2ndActivePolarity, 4
 Ch2ndDither, 4
 Ch2ndInitLevel, 5
 Ch2ndMatchDir, 5
 Ch2ndValue, 5
 DeadTimeValue, 5
 EnableCh1stEventDmaReq, 5
 EnableCh1stInterrupt, 5
 EnableCh1stMatchTrigger, 6
 EnableCh2ndEventDmaReq, 6
 EnableCh2ndInterrupt, 6
 EnableCh2ndMatchTrigger, 6
 EnableChComplementation, 6
 EnableDeadTime, 6
 EnableSymmetric, 7
 LevelMode, 7
 TimePsc, 7

- Pwm_Hal_CommonCfg, 7
 - ChannelCallback, 8
 - ClockSource, 8
 - DmaTransLen, 8
 - EnableDmaTransLen, 8
 - EnableOverflowDmaReq, 8
 - EnableOverflowEvent, 8
 - EnableOverflowInterrupt, 9
 - EnableUnderflowEvent, 9
 - MaxCount, 9
 - MinCount, 9
 - OverflowCallback, 9
 - OverflowFreq, 9
 - PeriodDither, 10
 - Prescaler, 10
- Pwm_Hal_CompareActionType
 - Pwm_Hal_Types.h, 147
- Pwm_Hal_CompareModeChnCfg, 10
 - Action, 11
 - ActivePolarity, 11
 - ChnValue, 11
 - EnableChnEventDmaReq, 11
 - EnableInterrupt, 11
 - EnableMatchTrigger, 11
 - InitLevel, 11
- Pwm_Hal_CountModeType
 - Pwm_Hal_Types.h, 147
- Pwm_Hal_Delnit
 - Pwm_Hal.c, 72
 - Pwm_Hal.h, 109
- Pwm_Hal_DeadTimePscType
 - Pwm_Hal_Types.h, 147
- Pwm_Hal_DebugModeType
 - Pwm_Hal_Types.h, 147
- Pwm_Hal_DualInputContinuousModeType
 - Pwm_Hal_Types.h, 148
- Pwm_Hal_DualInputMeasureType
 - Pwm_Hal_Types.h, 148
- Pwm_Hal_EdgeType
 - Pwm_Hal_Types.h, 148
- Pwm_Hal_EnableChannelDmaRequest
 - Pwm_Hal.c, 72
 - Pwm_Hal.h, 110
- Pwm_Hal_EnableChannelEventReset
 - Pwm_Hal.c, 73
 - Pwm_Hal.h, 110
- Pwm_Hal_EnableChannelHizOutput
 - Pwm_Hal.c, 73
 - Pwm_Hal.h, 111
- Pwm_Hal_EnableChannelInterrupt
 - Pwm_Hal.c, 74
 - Pwm_Hal.h, 111
- Pwm_Hal_EnableChannelMatchTrigger
 - Pwm_Hal.c, 74
 - Pwm_Hal.h, 112
- Pwm_Hal_EnableChannelOutputMask
 - Pwm_Hal.c, 75
 - Pwm_Hal.h, 112
- Pwm_Hal_EnableChannelPairInvert
 - Pwm_Hal.c, 75
 - Pwm_Hal.h, 113
- Pwm_Hal_EnableChannelPairSymmetric
 - Pwm_Hal.c, 76
 - Pwm_Hal.h, 113
- Pwm_Hal_EnableChannelSoftControl
 - Pwm_Hal.c, 76
 - Pwm_Hal.h, 114
- Pwm_Hal_EnableCombineChannelSoftControl
 - Pwm_Hal.c, 77
 - Pwm_Hal.h, 114
- Pwm_Hal_EnableFaultInputFilter
 - Pwm_Hal.c, 77
 - Pwm_Hal.h, 115
- Pwm_Hal_EnableFaultPinInput
 - Pwm_Hal.c, 78
 - Pwm_Hal.h, 115
- Pwm_Hal_EnableGlobalTimeBase
 - Pwm_Hal.c, 79
 - Pwm_Hal.h, 116
- Pwm_Hal_EnableGlobalTimeBaseOutput
 - Pwm_Hal.c, 79
 - Pwm_Hal.h, 116
- Pwm_Hal_EnableInitTrigger
 - Pwm_Hal.c, 79
 - Pwm_Hal.h, 117
- Pwm_Hal_EnableMaxTrigger
 - Pwm_Hal.c, 80
 - Pwm_Hal.h, 117
- Pwm_Hal_EnableOverflowEvent
 - Pwm_Hal.c, 80
 - Pwm_Hal.h, 118
- Pwm_Hal_EnableOverflowInterrupt
 - Pwm_Hal.c, 81
 - Pwm_Hal.h, 118
- Pwm_Hal_EnableSyncBypass
 - Pwm_Hal.c, 81
 - Pwm_Hal.h, 119
- Pwm_Hal_EnableUnderflowEvent
 - Pwm_Hal.c, 82
 - Pwm_Hal.h, 119
- Pwm_Hal_EnableWriteProtection
 - Pwm_Hal.c, 82
 - Pwm_Hal.h, 120
- Pwm_Hal_FaultCfg, 12
 - EnableChannelOutputCtrl, 12
 - EnableHiz, 12
 - EnableInterrupt, 13
 - FaultCallback, 13
 - FaultCtrlMode, 13
 - FaultPinCfg, 13
 - FilterPsc, 13
 - FilterValue, 14
- Pwm_Hal_FaultCtrlModeType
 - Pwm_Hal_Types.h, 149
- Pwm_Hal_FaultInputIdType
 - Pwm_Hal_Types.h, 149
- Pwm_Hal_FaultPinCfg, 14
 - EnableFaultFilter, 14
 - EnableFaultInput, 15
 - FaultPolarity, 15
- Pwm_Hal_GetAllChannelLevel
 - Pwm_Hal.c, 83

- Pwm_Hal.h, [120](#)
- Pwm_Hal_GetChannelInterruptFlag
 - Pwm_Hal.c, [83](#)
 - Pwm_Hal.h, [121](#)
- Pwm_Hal_GetChannelValue
 - Pwm_Hal.c, [84](#)
 - Pwm_Hal.h, [121](#)
- Pwm_Hal_GetCountValue
 - Pwm_Hal.c, [84](#)
 - Pwm_Hal.h, [122](#)
- Pwm_Hal_GetDetectEventFlag
 - Pwm_Hal.c, [85](#)
 - Pwm_Hal.h, [122](#)
- Pwm_Hal_GetFaultFlag
 - Pwm_Hal.c, [85](#)
 - Pwm_Hal.h, [123](#)
- Pwm_Hal_GetFaultPinFlag
 - Pwm_Hal.c, [86](#)
 - Pwm_Hal.h, [123](#)
- Pwm_Hal_GetHallStatus
 - Pwm_Hal.c, [86](#)
 - Pwm_Hal.h, [124](#)
- Pwm_Hal_GetInitCountValue
 - Pwm_Hal.c, [87](#)
 - Pwm_Hal.h, [124](#)
- Pwm_Hal_GetMaxCountValue
 - Pwm_Hal.c, [87](#)
 - Pwm_Hal.h, [124](#)
- Pwm_Hal_GetOutputMask
 - Pwm_Hal.c, [87](#)
 - Pwm_Hal.h, [125](#)
- Pwm_Hal_GetOverflowDir
 - Pwm_Hal.c, [88](#)
 - Pwm_Hal.h, [125](#)
- Pwm_Hal_GetOverflowFlag
 - Pwm_Hal.c, [88](#)
 - Pwm_Hal.h, [126](#)
- Pwm_Hal_GetQuadCountingDir
 - Pwm_Hal.c, [89](#)
 - Pwm_Hal.h, [126](#)
- Pwm_Hal_GetQuadOverflowDir
 - Pwm_Hal.c, [89](#)
 - Pwm_Hal.h, [127](#)
- Pwm_Hal_GetQuadPhaseZFlag
 - Pwm_Hal.c, [90](#)
 - Pwm_Hal.h, [127](#)
- Pwm_Hal_IndependentChnCfg, [15](#)
 - ActivePolarity, [16](#)
 - ChnDither, [16](#)
 - ChnValue, [16](#)
 - EnableChnEventDmaReq, [16](#)
 - EnableInterrupt, [16](#)
 - EnableMatchTrigger, [16](#)
 - InitLevel, [17](#)
 - LevelMode, [17](#)
- Pwm_Hal_InitFaultControl
 - Pwm_Hal.c, [90](#)
 - Pwm_Hal.h, [128](#)
- Pwm_Hal_InitInputMode
 - Pwm_Hal.c, [91](#)
 - Pwm_Hal.h, [128](#)
- Pwm_Hal_InitOutputMode
 - Pwm_Hal.c, [91](#)
 - Pwm_Hal.h, [129](#)
- Pwm_Hal_InitQuadDecoderMode
 - Pwm_Hal.c, [92](#)
 - Pwm_Hal.h, [129](#)
- Pwm_Hal_InitSyncConfigSet
 - Pwm_Hal.c, [92](#)
 - Pwm_Hal.h, [130](#)
- Pwm_Hal_InputCfg, [17](#)
 - ChannelCfg, [17](#)
 - ChannelNum, [18](#)
 - CommonCfg, [18](#)
- Pwm_Hal_InputChannelCfg, [18](#)
 - CaptureEdge, [19](#)
 - Channel, [19](#)
 - ChannelMode, [19](#)
 - DualInputContinuousMode, [19](#)
 - DualInputMeasureType, [19](#)
 - EnableChnEventDmaReq, [19](#)
 - EnableCounterReset, [19](#)
 - EnableInterrupt, [20](#)
 - EnablePulseWidthMeasure, [20](#)
 - EventPsc, [20](#)
 - FilterValue, [20](#)
- Pwm_Hal_InputChnModeType
 - Pwm_Hal_Types.h, [149](#)
- Pwm_Hal_InputCommonCfg, [20](#)
 - BaseCfg, [21](#)
 - EnableHall, [21](#)
 - FilterPsc, [21](#)
- Pwm_Hal_InputEventPscType
 - Pwm_Hal_Types.h, [150](#)
- Pwm_Hal_InputFilterPscType
 - Pwm_Hal_Types.h, [150](#)
- Pwm_Hal_InstanceType
 - Pwm_Hal_Types.h, [151](#)
- Pwm_Hal_OutputCfg, [21](#)
 - ChannelCfg, [22](#)
 - ChannelNum, [22](#)
 - CommonCfg, [22](#)
- Pwm_Hal_OutputChannelCfg, [22](#)
 - Channel, [23](#)
 - ChannelMode, [23](#)
 - ChnPairCfg, [23](#)
 - CompareChnCfg, [23](#)
 - IndependentChnCfg, [23](#)
- Pwm_Hal_OutputChnModeType
 - Pwm_Hal_Types.h, [151](#)
- Pwm_Hal_OutputCommonCfg, [24](#)
 - BaseCfg, [24](#)
 - CombineCenterDutyMode, [24](#)
 - CountMode, [24](#)
 - EnableInitTrigger, [25](#)
 - EnableMaxTrigger, [25](#)
 - InitOutput, [25](#)
 - TriggerRatio, [25](#)
- Pwm_Hal_OutputLevelModeType
 - Pwm_Hal_Types.h, [152](#)
- Pwm_Hal_OutputLevelType
 - Pwm_Hal_Types.h, [152](#)

- Pwm_Hal_QuadDecoderCfg, 25
 - BaseCfg, 26
 - EnablePhaseZReset, 26
 - EnableQuad, 26
 - FilterPsc, 26
 - Mode, 27
 - PhaseAConfig, 27
 - PhaseBConfig, 27
 - PhaseZConfig, 27
- Pwm_Hal_QuadModeType
 - Pwm_Hal_Types.h, 152
- Pwm_Hal_QuadPhaseCfg, 28
 - FilterValue, 28
 - Polarity, 28
- Pwm_Hal_QuadPhasePolarityType
 - Pwm_Hal_Types.h, 152
- Pwm_Hal_ResetCounter
 - Pwm_Hal.c, 93
 - Pwm_Hal.h, 130
- Pwm_Hal_SetChannelCaptureEdge
 - Pwm_Hal.c, 93
 - Pwm_Hal.h, 131
- Pwm_Hal_SetChannelCompareAction
 - Pwm_Hal.c, 94
 - Pwm_Hal.h, 131
- Pwm_Hal_SetChannelInputFilterVal
 - Pwm_Hal.c, 94
 - Pwm_Hal.h, 132
- Pwm_Hal_SetChannelMatchDither
 - Pwm_Hal.c, 95
 - Pwm_Hal.h, 132
- Pwm_Hal_SetChannelPolarity
 - Pwm_Hal.c, 95
 - Pwm_Hal.h, 133
- Pwm_Hal_SetChannelSoftControlLevel
 - Pwm_Hal.c, 96
 - Pwm_Hal.h, 134
- Pwm_Hal_SetChannelValue
 - Pwm_Hal.c, 96
 - Pwm_Hal.h, 134
- Pwm_Hal_SetClockSource
 - Pwm_Hal.c, 97
 - Pwm_Hal.h, 135
- Pwm_Hal_SetDeadtime
 - Pwm_Hal.c, 97
 - Pwm_Hal.h, 135
- Pwm_Hal_SetDebugMode
 - Pwm_Hal.c, 98
 - Pwm_Hal.h, 136
- Pwm_Hal_SetFaultInputPolarity
 - Pwm_Hal.c, 98
 - Pwm_Hal.h, 137
- Pwm_Hal_SetInitCountValue
 - Pwm_Hal.c, 99
 - Pwm_Hal.h, 137
- Pwm_Hal_SetInputFilterPsc
 - Pwm_Hal.c, 99
 - Pwm_Hal.h, 138
- Pwm_Hal_SetMaxCountValue
 - Pwm_Hal.c, 100
 - Pwm_Hal.h, 138
- Pwm_Hal_SetOutputMask
 - Pwm_Hal.c, 100
 - Pwm_Hal.h, 139
- Pwm_Hal_SetPeriodDither
 - Pwm_Hal.c, 101
 - Pwm_Hal.h, 139
- Pwm_Hal_SetSoftControlEnableStates
 - Pwm_Hal.c, 101
 - Pwm_Hal.h, 140
- Pwm_Hal_SetSoftControlLevels
 - Pwm_Hal.c, 102
 - Pwm_Hal.h, 140
- Pwm_Hal_SetTriggerRatio
 - Pwm_Hal.c, 102
 - Pwm_Hal.h, 141
- Pwm_Hal_SyncCfg, 28
 - DisableHwTriggerAfterTriggered, 29
 - EnableCounterInitSync, 29
 - EnableDualChannelInvertSync, 29
 - EnableHwSync0, 30
 - EnableHwSync1, 30
 - EnableHwSync2, 30
 - EnableMaxLoadingPoint, 30
 - EnableMinLoadingPoint, 30
 - EnableOutputMaskSync, 31
 - EnablePairedChnValSync, 31
 - EnablePolaritySync, 31
 - EnableSwOutputCtrlSync, 31
 - EnableSync, 31
 - EnableSyncBypass, 32
 - TriggerType, 32
- Pwm_Hal_SyncModeType
 - Pwm_Hal_Types.h, 153
- Pwm_Hal_SyncTriggerMethodType
 - Pwm_Hal_Types.h, 153
- Pwm_Hal_TrigSoftwareSync
 - Pwm_Hal.c, 103
 - Pwm_Hal.h, 141
- Pwm_Hal_Types.h, 142
 - Pwm_Hal_ActivePolarityType, 144
 - Pwm_Hal_Callback, 144
 - Pwm_Hal_ChannelMatchDirType, 145
 - Pwm_Hal_ChannelPairType, 145
 - Pwm_Hal_ChannelType, 145
 - Pwm_Hal_ClockSourceType, 146
 - Pwm_Hal_CombineCenterDutyModeType, 146
 - Pwm_Hal_CompareActionType, 147
 - Pwm_Hal_CountModeType, 147
 - Pwm_Hal_DeadTimePscType, 147
 - Pwm_Hal_DebugModeType, 147
 - Pwm_Hal_DualInputContinuousModeType, 148
 - Pwm_Hal_DualInputMeasureType, 148
 - Pwm_Hal_EdgeType, 148
 - Pwm_Hal_FaultCtrlModeType, 149
 - Pwm_Hal_FaultInputIdType, 149
 - Pwm_Hal_InputChnModeType, 149
 - Pwm_Hal_InputEventPscType, 150
 - Pwm_Hal_InputFilterPscType, 150
 - Pwm_Hal_InstanceType, 151
 - Pwm_Hal_OutputChnModeType, 151
 - Pwm_Hal_OutputLevelModeType, 152

- Pwm_Hal_OutputLevelType, [152](#)
- Pwm_Hal_QuadModeType, [152](#)
- Pwm_Hal_QuadPhasePolarityType, [152](#)
- Pwm_Hal_SyncModeType, [153](#)
- Pwm_Hal_SyncTriggerMethodType, [153](#)
- Pwm_Reg_ChooseBufferOrRegisterToRead
 - AC784xx_Pwm_Reg.h, [36](#)
- Pwm_Reg_ClearChannelEventFlag
 - AC784xx_Pwm_Reg.h, [36](#)
- Pwm_Reg_ClearChannelInterruptFlag
 - AC784xx_Pwm_Reg.h, [36](#)
- Pwm_Reg_ClearChannelTriggerFlag
 - AC784xx_Pwm_Reg.h, [37](#)
- Pwm_Reg_ClearFaultFlag
 - AC784xx_Pwm_Reg.h, [37](#)
- Pwm_Reg_ClearFaultPinFlag
 - AC784xx_Pwm_Reg.h, [37](#)
- Pwm_Reg_ClearOverflowFlag
 - AC784xx_Pwm_Reg.h, [37](#)
- Pwm_Reg_ClearQuadPhaseZFlag
 - AC784xx_Pwm_Reg.h, [37](#)
- Pwm_Reg_DisableHwTriggerSyncAfterTriggered
 - AC784xx_Pwm_Reg.h, [37](#)
- Pwm_Reg_EnableCHOSWCERSync
 - AC784xx_Pwm_Reg.h, [39](#)
- Pwm_Reg_EnableCNTINSync
 - AC784xx_Pwm_Reg.h, [39](#)
- Pwm_Reg_EnableChannelCombine
 - AC784xx_Pwm_Reg.h, [38](#)
- Pwm_Reg_EnableChannelDmaReq
 - AC784xx_Pwm_Reg.h, [38](#)
- Pwm_Reg_EnableChannelEventReset
 - AC784xx_Pwm_Reg.h, [38](#)
- Pwm_Reg_EnableChannelHizOutput
 - AC784xx_Pwm_Reg.h, [38](#)
- Pwm_Reg_EnableChannelInterrupt
 - AC784xx_Pwm_Reg.h, [38](#)
- Pwm_Reg_EnableChannelMatchTrigger
 - AC784xx_Pwm_Reg.h, [39](#)
- Pwm_Reg_EnableChannelOutputMask
 - AC784xx_Pwm_Reg.h, [39](#)
- Pwm_Reg_EnableChannelSoftControl
 - AC784xx_Pwm_Reg.h, [39](#)
- Pwm_Reg_EnableDmaTransfer
 - AC784xx_Pwm_Reg.h, [40](#)
- Pwm_Reg_EnableFaultHizOutput
 - AC784xx_Pwm_Reg.h, [40](#)
- Pwm_Reg_EnableFaultInputFilter
 - AC784xx_Pwm_Reg.h, [40](#)
- Pwm_Reg_EnableFaultInterrupt
 - AC784xx_Pwm_Reg.h, [40](#)
- Pwm_Reg_EnableFaultPinInput
 - AC784xx_Pwm_Reg.h, [40](#)
- Pwm_Reg_EnableGlobalTimeBase
 - AC784xx_Pwm_Reg.h, [41](#)
- Pwm_Reg_EnableGlobalTimeBaseOutput
 - AC784xx_Pwm_Reg.h, [41](#)
- Pwm_Reg_EnableHall
 - AC784xx_Pwm_Reg.h, [41](#)
- Pwm_Reg_EnableINVCERSync
 - AC784xx_Pwm_Reg.h, [41](#)
- Pwm_Reg_EnableInitTrigger
 - AC784xx_Pwm_Reg.h, [41](#)
- Pwm_Reg_EnableMaxLoadingPointSync
 - AC784xx_Pwm_Reg.h, [42](#)
- Pwm_Reg_EnableMaxTrigger
 - AC784xx_Pwm_Reg.h, [42](#)
- Pwm_Reg_EnableMinLoadingPointSync
 - AC784xx_Pwm_Reg.h, [42](#)
- Pwm_Reg_EnableOutputInit
 - AC784xx_Pwm_Reg.h, [42](#)
- Pwm_Reg_EnableOutputMaskSync
 - AC784xx_Pwm_Reg.h, [42](#)
- Pwm_Reg_EnableOverflowDmaReq
 - AC784xx_Pwm_Reg.h, [43](#)
- Pwm_Reg_EnableOverflowEvent
 - AC784xx_Pwm_Reg.h, [43](#)
- Pwm_Reg_EnableOverflowInterrupt
 - AC784xx_Pwm_Reg.h, [43](#)
- Pwm_Reg_EnablePOLSync
 - AC784xx_Pwm_Reg.h, [45](#)
- Pwm_Reg_EnablePairChComplement
 - AC784xx_Pwm_Reg.h, [43](#)
- Pwm_Reg_EnablePairChDeadtime
 - AC784xx_Pwm_Reg.h, [43](#)
- Pwm_Reg_EnablePairChDualEdgeCapture
 - AC784xx_Pwm_Reg.h, [44](#)
- Pwm_Reg_EnablePairChFaultCtrl
 - AC784xx_Pwm_Reg.h, [44](#)
- Pwm_Reg_EnablePairChInvert
 - AC784xx_Pwm_Reg.h, [44](#)
- Pwm_Reg_EnablePairChSymmetric
 - AC784xx_Pwm_Reg.h, [44](#)
- Pwm_Reg_EnablePairChValueSync
 - AC784xx_Pwm_Reg.h, [44](#)
- Pwm_Reg_EnablePairPulseWidthMeasurement
 - AC784xx_Pwm_Reg.h, [45](#)
- Pwm_Reg_EnableQuadDecoder
 - AC784xx_Pwm_Reg.h, [45](#)
- Pwm_Reg_EnableQuadPhaseZResetCnt
 - AC784xx_Pwm_Reg.h, [45](#)
- Pwm_Reg_EnableSync
 - AC784xx_Pwm_Reg.h, [45](#)
- Pwm_Reg_EnableSyncBYP
 - AC784xx_Pwm_Reg.h, [46](#)
- Pwm_Reg_EnableSyncHwTriggerSrc
 - AC784xx_Pwm_Reg.h, [46](#)
- Pwm_Reg_EnableUnderflowEvent
 - AC784xx_Pwm_Reg.h, [46](#)
- Pwm_Reg_EnableWriteProtection
 - AC784xx_Pwm_Reg.h, [46](#)
- Pwm_Reg_GetAllChannelLevel
 - AC784xx_Pwm_Reg.h, [46](#)
- Pwm_Reg_GetBase
 - AC784xx_Pwm_Reg.h, [47](#)
- Pwm_Reg_GetChannelEventFlag
 - AC784xx_Pwm_Reg.h, [47](#)
- Pwm_Reg_GetChannelInterruptFlag
 - AC784xx_Pwm_Reg.h, [47](#)
- Pwm_Reg_GetChannelTriggerFlag
 - AC784xx_Pwm_Reg.h, [47](#)
- Pwm_Reg_GetChannelValue

AC784xx_Pwm_Reg.h, [47](#)
Pwm_Reg_GetCountValue
AC784xx_Pwm_Reg.h, [47](#)
Pwm_Reg_GetFaultFlag
AC784xx_Pwm_Reg.h, [48](#)
Pwm_Reg_GetFaultInputStatus
AC784xx_Pwm_Reg.h, [48](#)
Pwm_Reg_GetFaultPinFlag
AC784xx_Pwm_Reg.h, [48](#)
Pwm_Reg_GetHallStatus
AC784xx_Pwm_Reg.h, [48](#)
Pwm_Reg_GetInitCountValue
AC784xx_Pwm_Reg.h, [48](#)
Pwm_Reg_GetMaxCountValue
AC784xx_Pwm_Reg.h, [48](#)
Pwm_Reg_GetOutputMask
AC784xx_Pwm_Reg.h, [49](#)
Pwm_Reg_GetOverflowDir
AC784xx_Pwm_Reg.h, [49](#)
Pwm_Reg_GetOverflowFlag
AC784xx_Pwm_Reg.h, [49](#)
Pwm_Reg_GetQuadCountDir
AC784xx_Pwm_Reg.h, [49](#)
Pwm_Reg_GetQuadOverflowDir
AC784xx_Pwm_Reg.h, [49](#)
Pwm_Reg_GetQuadPhaseZFlag
AC784xx_Pwm_Reg.h, [49](#)
Pwm_Reg_IsPairChDualEdgeCaptureEnabled
AC784xx_Pwm_Reg.h, [50](#)
Pwm_Reg_ReadChannelSoftControlState
AC784xx_Pwm_Reg.h, [50](#)
Pwm_Reg_ResetCounter
AC784xx_Pwm_Reg.h, [50](#)
Pwm_Reg_SetCHOSWCERSyncTriggerMode
AC784xx_Pwm_Reg.h, [52](#)
Pwm_Reg_SetCNTSyncTriggerMode
AC784xx_Pwm_Reg.h, [53](#)
Pwm_Reg_SetCaptureEventPsc
AC784xx_Pwm_Reg.h, [50](#)
Pwm_Reg_SetChannelELSR
AC784xx_Pwm_Reg.h, [50](#)
Pwm_Reg_SetChannelInputFilterVal
AC784xx_Pwm_Reg.h, [51](#)
Pwm_Reg_SetChannelMSR
AC784xx_Pwm_Reg.h, [51](#)
Pwm_Reg_SetChannelMatchDir
AC784xx_Pwm_Reg.h, [51](#)
Pwm_Reg_SetChannelMatchDither
AC784xx_Pwm_Reg.h, [51](#)
Pwm_Reg_SetChannelMatchDitherReg
AC784xx_Pwm_Reg.h, [51](#)
Pwm_Reg_SetChannelOutputInitLevel
AC784xx_Pwm_Reg.h, [52](#)
Pwm_Reg_SetChannelPolarity
AC784xx_Pwm_Reg.h, [52](#)
Pwm_Reg_SetChannelSoftControlLevel
AC784xx_Pwm_Reg.h, [52](#)
Pwm_Reg_SetChannelValue
AC784xx_Pwm_Reg.h, [52](#)
Pwm_Reg_SetClockPsc
AC784xx_Pwm_Reg.h, [53](#)
Pwm_Reg_SetClockSource
AC784xx_Pwm_Reg.h, [53](#)
Pwm_Reg_SetCntOverflowFreq
AC784xx_Pwm_Reg.h, [53](#)
Pwm_Reg_SetCombineCenterDutyType
AC784xx_Pwm_Reg.h, [53](#)
Pwm_Reg_SetCountMode
AC784xx_Pwm_Reg.h, [54](#)
Pwm_Reg_SetDebugMode
AC784xx_Pwm_Reg.h, [54](#)
Pwm_Reg_SetDmaTransferLen
AC784xx_Pwm_Reg.h, [54](#)
Pwm_Reg_SetFaultInputFilterValue
AC784xx_Pwm_Reg.h, [54](#)
Pwm_Reg_SetFaultInputPolarity
AC784xx_Pwm_Reg.h, [54](#)
Pwm_Reg_SetFaultMode
AC784xx_Pwm_Reg.h, [55](#)
Pwm_Reg_SetINVCRSyncTriggerMode
AC784xx_Pwm_Reg.h, [55](#)
Pwm_Reg_SetInitCountValue
AC784xx_Pwm_Reg.h, [55](#)
Pwm_Reg_SetInputFilterPsc
AC784xx_Pwm_Reg.h, [55](#)
Pwm_Reg_SetMCCVRSyncMode
AC784xx_Pwm_Reg.h, [56](#)
Pwm_Reg_SetMaxCountValue
AC784xx_Pwm_Reg.h, [55](#)
Pwm_Reg_SetOMCRSyncTriggerMode
AC784xx_Pwm_Reg.h, [56](#)
Pwm_Reg_SetOutputMask
AC784xx_Pwm_Reg.h, [56](#)
Pwm_Reg_SetPOLTriggerMode
AC784xx_Pwm_Reg.h, [57](#)
Pwm_Reg_SetPairChDeadtime
AC784xx_Pwm_Reg.h, [56](#)
Pwm_Reg_SetPairChDualEdgeCapture
AC784xx_Pwm_Reg.h, [56](#)
Pwm_Reg_SetPeriodDither
AC784xx_Pwm_Reg.h, [57](#)
Pwm_Reg_SetQuadEncodeMode
AC784xx_Pwm_Reg.h, [57](#)
Pwm_Reg_SetQuadPhaseAPolarity
AC784xx_Pwm_Reg.h, [57](#)
Pwm_Reg_SetQuadPhaseBPolarity
AC784xx_Pwm_Reg.h, [57](#)
Pwm_Reg_SetQuadPhaseZPolarity
AC784xx_Pwm_Reg.h, [58](#)
Pwm_Reg_SetSoftControlEnableStatesMask
AC784xx_Pwm_Reg.h, [58](#)
Pwm_Reg_SetSoftControlLevelMask
AC784xx_Pwm_Reg.h, [58](#)
Pwm_Reg_SetSyncMode
AC784xx_Pwm_Reg.h, [58](#)
Pwm_Reg_SetSyncType
AC784xx_Pwm_Reg.h, [58](#)
Pwm_Reg_SetTriggerRatio
AC784xx_Pwm_Reg.h, [59](#)
Pwm_Reg_TrigSoftwareSync
AC784xx_Pwm_Reg.h, [59](#)

TimePsc

 Pwm_Hal_CombineChnPairCfg, [7](#)

TriggerRatio

 Pwm_Hal_OutputCommonCfg, [25](#)

TriggerType

 Pwm_Hal_SyncCfg, [32](#)