

## Release Notes for SPC1168 Standard Peripherals Library Drivers

**V1.8.0/ 31- Dec -2021**

### Main Changes

#### 1. Bug Fix

- gpio.c
  - Modify GPIO\_ResetAllPin()
- comp.h/.c
  - Modify COMP\_Init(), add code to enable bandgap
  - Modify COMP\_EnableDAC(), add code to enable bandgap
  - Modify COMP\_DACBufferInit(), add code to enable bandgap
- adc.c
  - Modify ADC\_SetSampleAndConvertTime()
  - Modify ADC\_PowerUp(), add code to enable bandgap
- pga.c
  - Modify PGA\_DifferentialInit(), add code to enable bandgap
- clock.c
  - Modify CLOCK\_ConfigurePLL()
- timer.h
  - Delete TIMER\_EnableCounterHold() and TIMER\_DisableCounterHold()
- uart.h
  - Modify UART\_ClearRxFIFO() and UART\_ClearTxFIFO()
- spc1168\_reg.h/spc1168\_bitfield.h
  - Reserve TMRCTL.HOLD register bit-field
  - Modify the width of UARTFOR.BYTECNT register bit-field

#### 2. Other Change

- pga.c
  - Add PGA\_EnableSensorMode() and PGA\_DisableSensorMode()
- flash.h/.c
  - Add FLASH\_EraseBlock()
  - Modify FLASH\_ProgramWord()
  - Add macro definition *FLASH\_CMD\_ERASE\_BLOCK*
- clock.c
  - Add note for CLOCK\_TrimPLL()
  - Modify CLOCK\_Init()
- adc.h/.c
  - Modify ADC\_CalculatePreciseTemperature() and ADC\_CalculateTemperature()
  - Add ADC\_EnableBandgap() and ADC\_DisableBandgap()
- spc1168\_reg.h/spc1168\_bitfield.h
  - Reserve XOCTL.FREQSEL register bit-field

- Rename XOCTL.FEEDXOUT register bit-field to XOCTL. FEEDXIO
- `crc.h`
  - Modify `CRC_ModeEnum`
- `gpio.h`
  - Modify the post-fix of const number
- Update `SPC1168.svd`, `SPC1168.SFR`, `SPC1168.FLM` and `SPC1168_NVR.FLM` files
- `iar/startup_spc1168.s`
  - Add code comment
- Example Code
  - Modify `Flash_M_Access_User`
  - Modify `Flash_User_FuncWrap`
  - Modify `AES/main.c`
  - Modify `IAP_Modem/main.c`

**V1.7.0/ 26-May-2021****Main Changes**

- spc1168.h
  - Add macro function READ\_BITS().
  - Add macro function WRITE\_FIELD(), READ\_FIELD() and CLEAR\_FIELD().
  - Add macro function POSITION\_VAL().
  - Add macro function UNUSED().
- spc1168\_bitfield.h
  - Update TZBCTL.DCBEVT1D bit-field macro definitions.
- flash.c
  - Update function FLASH\_SetTiming().
- i2c.c
  - Update function I2C\_MasterWrite(), I2C\_MasterRead(), I2C\_MasterBulkWrite() and I2C\_MasterBulkRead().
  - Update function I2C\_SlaveBulkWrite().
- clock.c
  - Update CLOCK\_Init(), CLOCK\_InitWithRCO(), CLOCK\_ConfigurePLL() and CLOCK\_ConfigurePLLWithRCO().
  - Update CLOCK\_HCLKSelEnum, CLOCK\_RefSelEnum definitions.
- comp.c
  - Update function COMP\_Init() and COMP\_SetFilterWindowTimeNs().
- delay.h/delay.c
  - Update Delay method control macros.
- ssp.c
  - Update function SSP\_MasterB2BTransceive() and SSP\_SlaveTransceive().
- ecap.c
  - Update function ECAP\_CaptureModelInit().
  - Rename and modify function ECAP\_SetInputPin() to ECAP\_SetInput().
- pwm.c
  - Update function PWM\_ComplementaryPairChannelInit() and PWM\_SingleChannelInit().
- IDE\_Support
  - Update SPC1168.svd and SPC1168.SFR files.
- Demo code
  - Update ECAP\_Continue\_Absolute\main.c
  - Update ECAP\_Continue\_Delta\main.c
  - Update ECAP\_Oneshot\_Absolute\main.c
  - Update UART\_TX\_and\_CheckRX\main.c
  - Update UART\_RX\_and\_SentBack\main.c
  - Update SSP\_Master\_B2B\_TxRx\_Polling\main.c
  - Update SSP\_Master\_TxRx\_INT\main.c
  - Update SSP\_Master\_TxRx\_Polling\main.c
  - Update SSP\_Slave\_B2B\_TxRx\_Polling\main.c
  - Update SSP\_Slave\_TxRx\_INT\main.c

- Update SSP\_Slave\_TxRx\_Polling\main.c
- Update PWM\_Trigger\_ADC\_Sample\main.c
- Update PWM\_Current\_Protect\_Trigger\_TZ\main.c
- Update PGA\_Calibration\main.c
- Update I2C\_Master\_Polling\_TxRx\main.c
- Update I2C\_Master\_Bulk\_Polling\_TxRx\main.c
- Update I2C\_Master\_INT\_Rx\main.c
- Update GPIO\_Edge\_Detect\main.c
- Update Flash\_With\_INT\main.c
- Update Flash\_EEPROM\_Emulation\EWARM\spc1168.icf
- Update Flash\_Frequency\_Reduction\EWARM\spc1168.icf
- Update Flash\_M\_Access\_User\EWARM\spc1168.icf
- Update Flash\_Operation\EWARM\spc1168.icf
- Update Flash\_Sector\_Protect\EWARM\spc1168.icf
- Update Flash\_User\_FuncWrap\EWARM\spc1168.icf
- Update Flash\_With\_INT\EWARM\spc1168.icf
- Update DAC\main.c

**V1.6.0/ 30-April-2020****Main Changes**

- **adc.h**
  - Update comments on `ADC_GetResult()`, `ADC_GetTrimResult1()` and `ADC_GetTrimResult2()`.
- **clock.h/clock.c**
  - Add macro definition `__RCO0_CLOCK` and `__RCO1_CLOCK` for RCO clocks.
  - Remove macro definition `__RCO_CLOCK` and `__XO_CLOCK`.
  - Update 3 functions `CLOCK_Init()`, `CLOCK_InitWithRCO()` and `CLOCK_GetModuleClock()` for configuring WDT clock.
- **comp.h**
  - Rename function `COMP_EnableOutputInvert()` to `COMP_SetOutputActiveLow()`.
  - Rename function `COMP_DisableOutputInvert()` to `COMP_SetOutputActiveHigh()`.
- **pwm.h**
  - Rename macro function `PWM_SetTimeEventTiming()` to `PWM_SetTimeEventIntTiming()`.
  - Rename macro function `PWM_SetTimeEventPeriod()` to `PWM_SetTimeEventIntPeriod()`.
  - Add member value `ON_SYNC_PERIOD` to definition `PWM_LoadTimingEnum`.
- **system\_spc1168.c**
  - Remove macro definition `__SYSTEM_CLOCK`.
- **timer.h**
  - Add macro function `TIMER_SetCounterValue()`.
- **wdt.c**
  - Update function `WDT_Init()` for enabling watchdog halted-run and lockup-run mode.

**V1.5.0/ 18-November-2019****Main Changes**

- **ecap.h**
  - Rename definition *ECAP\_EvtEnum* to *ECAP\_EventEnum*.
  - Update function *ECAP\_APWMSetDuty()*.
- **flash.c**
  - Update function *FLASH\_WriteProtect()*.
- **pwm.h**
  - Update macro function *PWM\_SetOneShotTripEvent()*.
  - Update macro function *PWM\_SetCBCTripEvent()*.
  - Update member values of definition *PWM\_TripActionEnum*.
- **system.h**
  - Rename definition *SYSTEM\_ResetEvtEnum* to *SYSTEM\_ResetEventEnum*.
- **wdt.h**
  - Rename macro function *WDT\_GetRawIntFlag()* to *WDT\_GetIntRawFlag()*.
- **spc1168\_bitfield.h**
  - Update *UARTISR.XMITIR* bit-field macro definitions.

**V1.4.0/ 18-July-2019****Main Changes**

- aes.h
  - Rename function AES\_EnableInputFIFOFullInt() to AES\_EnableInputFullInt().
  - Rename function AES\_DisableInputFIFOFullInt() to AES\_DisableInputFullInt().
  - Rename function AES\_EnableOutputFIFOEmptyInt() to AES\_EnableOutputEmptyInt().
  - Rename function AES\_DisableOutputFIFOEmptyInt() to AES\_DisableOutputEmptyInt().
  - Rename function AES\_ClearInputFIFOFullInt() to AES\_ClearInputFullInt().
  - Rename function AES\_ClearOutputFIFOEmptyInt() to AES\_ClearOutputEmptyInt().
  - Rename function AES\_GetInputFIFOFullIntFlag() to AES\_GetInputFullIntFlag().
  - Rename function AES\_GetOutputFIFOEmptyIntFlag() to AES\_GetOutputEmptyIntFlag().
  - Rename function AES\_GetInputFIFOFullIntRawFlag() to AES\_GetInputFullIntRawFlag().
  - Rename function AES\_GetOutputFIFOEmptyIntRawFlag() to AES\_GetOutputEmptyIntRawFlag().
  - Rename function AES\_IsInputFIFOFull() to AES\_IsInputFull().
  - Rename function AES\_IsOutputFIFOReady() to AES\_IsOutputReady().
  - Rename function AES\_IsOutputFIFOEmpty() to AES\_IsOutputEmpty().
- comp.h
  - Rename function COMP\_SetFilterClkDiv() to COMP\_SetFilterClockDiv().
- i2c.h
  - Rename function I2C\_EnableRxFullInt() to I2C\_EnableRxDataAvailableInt().
  - Rename function I2C\_DisableRxFullInt() to I2C\_DisableRxDataAvailableInt().
  - Rename function I2C\_EnableTxEmptyInt() to I2C\_EnableTxDataRequestInt().
  - Rename function I2C\_DisableTxEmptyInt() to I2C\_DisableTxDataRequestInt().
  - Rename function I2C\_GetRxFullIntFlag() to I2C\_GetRxDataAvailableIntFlag().
  - Rename function I2C\_GetTxEmptyIntFlag() to I2C\_GetTxDataRequestIntFlag().
  - Rename function I2C\_GetRxFullIntRawFlag() to I2C\_GetRxDataAvailableIntRawFlag().
  - Rename function I2C\_GetTxEmptyIntRawFlag() to I2C\_GetTxDataRequestIntRawFlag().
  - Rename function I2C\_IsTxFIFOFull() to I2C\_IsTxFull().
  - Rename function I2C\_IsTxFIFOEmpty() to I2C\_IsTxEmpty().
  - Rename function I2C\_IsRxFIFOFull() to I2C\_IsRxFull().
  - Rename function I2C\_IsRxFIFOEmpty() to I2C\_IsRxEmpty().
- ssp.h
  - Rename function SSP\_EnableRxFIFOOverflowInt() to SSP\_EnableRxOverflowInt().
  - Rename function SSP\_DisableRxFIFOOverflowInt() to SSP\_DisableRxOverflowInt().
  - Rename function SSP\_EnableTxFIFOUnderflowInt() to SSP\_EnableTxUnderflowInt().
  - Rename function SSP\_DisableTxFIFOUnderflowInt() to SSP\_DisableTxUnderflowInt().
  - Rename function SSP\_EnableTxFIFOEmptyInt() to SSP\_EnableTxDataRequestInt().
  - Rename function SSP\_DisableTxFIFOEmptyInt() to SSP\_DisableTxDataRequestInt().
  - Rename function SSP\_EnableRxFIFOFullInt() to SSP\_EnableRxDataAvailableInt().
  - Rename function SSP\_DisableRxFIFOFullInt() to SSP\_DisableRxDataAvailableInt().
  - Rename function SSP\_SetRxFIFOTriggerLevel() to SSP\_SetRxFIFOTriggerThreshold().
  - Rename function SSP\_SetTxFIFOTriggerLevel() to SSP\_SetTxFIFOTriggerThreshold().
  - Rename function SSP\_IsTxFIFOFull() to SSP\_IsTxFull().

- Rename function `SSP_IsRxFIFONotEmpty()` to `SSP_IsRxNotEmpty()`.
  - Rename function `SSP_IsTxFIFOServiceRequest()` to `SSP_IsTxServiceRequest()`.
  - Rename function `SSP_IsRxFIFOServiceRequest()` to `SSP_IsRxServiceRequest()`.
  - Rename function `SSP_IsRxFIFOOverflow()` to `SSP_IsRxOverflow()`.
  - Rename function `SSP_IsTxFIFOUnderflow()` to `SSP_IsTxUnderflow()`.
  - Rename function `SSP_IsTxFIFOHasOddSample()` to `SSP_IsTxHasOddSample()`.
  - Rename function `SSP_IsRxFIFOHasOddSample()` to `SSP_IsRxHasOddSample()`.
  - Rename function `SSP_ClearRxFIFOOverflowInt()` to `SSP_ClearRxOverflowInt()`.
  - Rename function `SSP_ClearTxFIFOUnderflowInt()` to `SSP_ClearTxUnderflowInt()`.
- `uart.h/.c`
- Rename definition `UART_TxTriggerLevelEnum` to `UART_TxThresholdEnum`.
  - Rename definition `UART_RxTriggerLevelEnum` to `UART_RxThresholdEnum`.
  - Rename function `UART_ResetRxFIFO()` to `UART_ClearRxFIFO()`.
  - Rename function `UART_ResetTxFIFO()` to `UART_ClearTxFIFO()`.
  - Rename function `UART_IsTxFIFOEmpty()` to `UART_IsTxDone()`.
  - Rename function `UART_IsBreakReceived()` to `UART_IsRxBreak()`.
  - Rename function `UART_IsFrameError()` to `UART_IsRxFrameError()`.
  - Rename function `UART_IsParityError()` to `UART_IsRxParityError()`.
  - Rename function `UART_IsRxDataLost()` to `UART_IsRxOverflow()`.
  - Rename function `UART_IsRxDataReady()` to `UART_IsRxNotEmpty()`.
  - Rename function `UART_SetFIFOTriggerLevel()` to `UART_SetFIFOTriggerThreshold()`.
- `spc1168.h`
- Add 9 new functions
    - 2 new functions to set register bits: `SET_BIT()` and `SET_BITS()`.
    - 2 new functions to clear register bits: `CLEAR_BIT()` and `CLEAR_BITS()`.
    - 1 new function to read register bit value: `READ_BIT()`.
    - 1 new function to clear register: `CLEAR_REG()`.
    - 1 new function to write value to register: `WRITE_REG()`.
    - 1 new function to read register value: `READ_REG()`.
    - 1 new function to modify register value: `MODIFY_REG()`.
- `spc1168_reg.h/spc1168_bitfield.h`
- Rename bit definition `NRZE` to `NRZME` in `UARTIER` register.
  - Rename bit definition `RSTRF` to `CLRRF` in `UARTFCR` register.
  - Rename bit definition `RSTTF` to `CLRTF` in `UARTFCR` register.
  - Rename bit definition `ITL` to `RXTH` in `UARTFCR` register.
  - Rename bit definition `TIL` to `TXTH` in `UARTFCR` register.
  - Rename bit definition `BUS` to `BUS32` in `UARTFCR` register.
  - Rename bit definition `TXEMPTY` to `TXDONE` in `UARTLSR` register.
  - Rename bit definition `RXFULL` to `RXDAV` in `I2CIF/I2CRAWIF/I2CIE` register.
  - Rename bit definition `TXEMPTY` to `TXDREQ` in `I2CIF/I2CRAWIF/I2CIE` register.
- Add IAR for ARM project support for examples.



## V1.3.0/ 02-July-2019

### Main Changes

- adc.h/.c
  - Rename function ADC\_PowerUP() to ADC\_PowerUp().
  - Rename function ADC\_SetSampleAndConvTime() to ADC\_SetSampleAndConvertTime().
  - Rename function ADC\_SelectPinSingleEnd() to ADC\_SelectPinSingleEnded().
- aes.h
  - Rename function AES\_ClearInputFifo() to AES\_ClearInputFIFO().
  - Rename function AES\_ClearOutputFifo() to AES\_ClearOutputFIFO().
  - Rename function AES\_EnableInputFifoFullInt() to AES\_EnableInputFIFOFullInt().
  - Rename function AES\_DisableInputFifoFullInt() to AES\_DisableInputFIFOFullInt().
  - Rename function AES\_EnableOutputFifoEmptyInt() to AES\_EnableOutputFIFOEmptyInt().
  - Rename function AES\_DisableOutputFifoEmptyInt() to AES\_DisableOutputFIFOEmptyInt().
  - Rename function AES\_ClearInputFifoFullInt() to AES\_ClearInputFIFOFullInt().
  - Rename function AES\_ClearOutputFifoEmptyInt() to AES\_ClearOutputFIFOEmptyInt().
  - Rename function AES\_GetInputFifoFullIntFlag() to AES\_GetInputFIFOFullIntFlag().
  - Rename function AES\_GetOutputFifoEmptyIntFlag() to AES\_GetOutputFIFOEmptyIntFlag().
  - Rename function AES\_GetInputFifoFullIntRawFlag() to AES\_GetInputFIFOFullIntRawFlag().
  - Rename function AES\_GetOutputFifoEmptyIntRawFlag() to AES\_GetOutputFIFOEmptyIntRawFlag().
  - Rename function AES\_IsInputFifoFull() to AES\_IsInputFIFOFull().
  - Rename function AES\_IsOutputFifoReady() to AES\_IsOutputFIFOReady().
  - Rename function AES\_IsOutputFifoEmpty() to AES\_IsOutputFIFOEmpty().
- clock.h/.c
  - Add 1 new function declaration: CLOCK\_SetModuleDiv().
  - Rename macro function CLOCK\_NstoCounter() to CLOCK\_NsToCounter().
  - Rename function CLOCK\_PLLConfig() to CLOCK\_ConfigurePLL().
  - Rename function CLOCK\_PLLQuickConfigWithRCO() to CLOCK\_ConfigurePLLWithRCO().
  - Rename function CLOCK\_GetModuleClk() to CLOCK\_GetModuleClock().
- comp.h
  - Update function COMP\_ResetFilter().
  - Rename function COMP\_SetDACValueMV() to COMP\_SetDACVoltage().
- ecap.h/.c
  - Rename function ECAP\_APwmActiveHigh() to ECAP\_APWMSetActiveHigh().
  - Rename function ECAP\_APwmActiveLow() to ECAP\_APWMSetActiveLow().
  - Rename function ECAP\_CounterRun() to ECAP\_RunCounter().
  - Rename function ECAP\_CounterStop() to ECAP\_StopCounter().
  - Rename function ECAP\_EventTriggeredOnRisingEdge() to ECAP\_SetEventTriggeredOnRisingEdge().
  - Rename function ECAP\_EventTriggeredOnFallingEdge() to ECAP\_SetEventTriggeredOnFallingEdge().
  - Rename function ECAP\_SetCounterVal() to ECAP\_SetCounterValue().
  - Rename function ECAP\_EnableCntOverflowInt() to ECAP\_EnableCounterOverflowInt().

- Rename function ECAP\_DisableCntOverflowInt() to ECAP\_DisableCounterOverflowInt().
- Rename function ECAP\_EnableCntEqualPrdInt() to ECAP\_EnableCounterEqualPRDInt().
- Rename function ECAP\_DisableCntEqualPrdInt() to ECAP\_DisableCounterEqualPRDInt().
- Rename function ECAP\_EnableCntEqualCmplInt() to ECAP\_EnableCounterEqualCMPInt().
- Rename function ECAP\_DisableCntEqualCmplInt() to ECAP\_DisableCounterEqualCMPInt().
- Rename function ECAP\_GetCntOverflowIntFlag() to ECAP\_GetCounterOverflowIntFlag().
- Rename function ECAP\_GetCntEqualPrdIntFlag() to ECAP\_GetCounterEqualPRDIntFlag().
- Rename function ECAP\_GetCntEqualCmplIntFlag() to ECAP\_GetCounterEqualCMPIntFlag().
- Rename function ECAP\_ForceCntOverflowInt() to ECAP\_ForceCounterOverflowInt().
- Rename function ECAP\_ForceCntEqualPrdInt() to ECAP\_ForceCounterEqualPRDInt().
- Rename function ECAP\_ForceCntEqualCmplInt() to ECAP\_ForceCounterEqualCMPInt().
- Rename function ECAP\_ClearCntOverflowInt() to ECAP\_ClearCounterOverflowInt().
- Rename function ECAP\_ClearCntEqualPrdInt() to ECAP\_ClearCounterEqualPRDInt().
- Rename function ECAP\_ClearCntEqualCmplInt() to ECAP\_ClearCounterEqualCMPInt().
- Rename function ECAP\_APwmModelInit() to ECAP\_APWMMModelInit().
- Rename function ECAP\_APwmSetDuty() to ECAP\_APWMSetDuty().

#### ■ gpio.h

- Rename macro function GpioPinPort() to \_\_GPIO\_PIN\_PORT().
- Rename macro function GpioPinIndex() to \_\_GPIO\_PIN\_INDEX().
- Rename macro function GPIO\_OUTPUT\_REG\_ADDR() to \_\_GPIO\_OUTPUT\_REG\_ADDR().
- Rename macro function GpioGplrAddr() to \_\_GPIO\_GPLR\_ADDR().
- Rename function GPIO\_LevelIntActiveHigh() to GPIO\_SetLevelIntActiveHigh().
- Rename function GPIO\_LevelIntActiveLow() to GPIO\_SetLevelIntActiveLow().

#### ■ i2c.h/.c

- Rename function I2C\_IsTxFifoNotFull() to I2C\_IsTxFIFONotFull().
- Rename function I2C\_IsTxFifoEmpty() to I2C\_IsTxFIFOEmpty().
- Rename function I2C\_IsRxFifoNotEmpty() to I2C\_IsRxFIFONotEmpty().
- Rename function I2C\_IsRxFifoFull() to I2C\_IsRxFIFOFull().
- Rename function I2C\_SetTxFifoThreshold() to I2C\_SetTxFIFOThreshold().
- Rename function I2C\_SetRxFifoThreshold() to I2C\_SetRxFIFOThreshold().
- Rename function I2C\_GetTxFifoLevel() to I2C\_GetTxFIFOLevel().
- Rename function I2C\_GetRxFifoLevel() to I2C\_GetRxFIFOLevel().
- Update 3 functions: I2C\_SpeedInit(), I2C\_MasterInit() and I2C\_SlaveInit().
- Remove definition *I2C\_SpeedModeEnum*.

#### ■ pga.h/.c

- Rename function PGA\_DiffInit() to PGA\_DifferentialInit().
- Rename function PGA\_SelecPositiveCHAsComInput() to PGA\_SelectPositiveChannelAsCommonInput().
- Rename function PGA\_SelecNegativeCHAsComInput() to PGA\_SelectNegativeChannelAsCommonInput().

#### ■ pwm.h/.c

- Rename function PWM\_GetPeriodValue() to PWM\_GetPRD().
- Rename function PWM\_SetPeriodValue() to PWM\_SetPRD().
- Rename function PWM\_EnableSyncFromTimer0() to PWM\_EnableSyncFromTIMER0().

- Rename function PWM\_EnableSyncFromTimer1() to PWM\_EnableSyncFromTIMER1().
- Rename function PWM\_EnableSyncFromTimer2() to PWM\_EnableSyncFromTIMER2().
- Rename function PWM\_DisableSyncFromTimer0() to PWM\_DisableSyncFromTIMER0().
- Rename function PWM\_DisableSyncFromTimer1() to PWM\_DisableSyncFromTIMER1().
- Rename function PWM\_DisableSyncFromTimer2() to PWM\_DisableSyncFromTIMER2().
- Rename function PWM\_CalSyncReloadValue() to PWM\_CalculateSyncReloadValue().

#### ■ ssp.h/.c

- Rename function SSP\_EnableFifoPackMode() to SSP\_EnableFIFOPackMode().
- Rename function SSP\_DisableFifoPackMode() to SSP\_DisableFIFOPackMode().
- Rename function SSP\_EnableRxFifoOverflowInt() to SSP\_EnableRxFIFOOverflowInt().
- Rename function SSP\_DisableRxFifoOverflowInt() to SSP\_DisableRxFIFOOverflowInt().
- Rename function SSP\_EnableTxFifoUnderflowInt() to SSP\_EnableTxFIFOUnderflowInt().
- Rename function SSP\_DisableTxFifoUnderflowInt() to SSP\_DisableTxFIFOUnderflowInt().
- Rename function SSP\_EnableTxFifoEmptyInt() to SSP\_EnableTxFIFOEmptyInt().
- Rename function SSP\_DisableTxFifoEmptyInt() to SSP\_DisableTxFIFOEmptyInt().
- Rename function SSP\_EnableRxFifoFullInt() to SSP\_EnableRxFIFOFullInt().
- Rename function SSP\_DisableRxFifoFullInt() to SSP\_DisableRxFIFOFullInt().
- Rename function SSP\_GetTxFifoLevel() to SSP\_GetTxFIFOLevel().
- Rename function SSP\_GetRxFifoLevel() to SSP\_GetRxFIFOLevel().
- Rename function SSP\_SetRxFifoTriggerLevel() to SSP\_SetRxFIFOTriggerLevel().
- Rename function SSP\_SetTxFifoTriggerLevel() to SSP\_SetTxFIFOTriggerLevel().
- Rename function SSP\_IsTxFifoNotFull() to SSP\_IsTxFIFONotFull().
- Rename function SSP\_IsRxFifoNotEmpty() to SSP\_IsRxFIFONotEmpty().
- Rename function SSP\_IsTxFifoServiceRequest() to SSP\_IsTxFIFOServiceRequest().
- Rename function SSP\_IsRxFifoServiceRequest() to SSP\_IsRxFIFOServiceRequest().
- Rename function SSP\_IsRxFifoOverflow() to SSP\_IsRxFIFOOverflow().
- Rename function SSP\_IsTxFifoUnderflow() to SSP\_IsTxFIFOUnderflow().
- Rename function SSP\_IsBusySyncSlaveClk() to SSP\_IsBusySyncSlaveClock().
- Rename function SSP\_IsTxFifoHasOddSample() to SSP\_IsTxFIFOHasOddSample().
- Rename function SSP\_IsRxFifoHasOddSample() to SSP\_IsRxFIFOHasOddSample().
- Rename function SSP\_ClearRxFifoOverflowInt() to SSP\_ClearRxFIFOOverflowInt().
- Rename function SSP\_ClearTxFifoUnderflowInt() to SSP\_ClearTxFIFOUnderflowInt().
- Update 2 functions: SSP\_Init() and SSP\_MasterB2BTransceive().

#### ■ uart.h/.c

- Rename function UART\_DisableFifo() to UART\_DisableFIFO().
- Rename function UART\_ResetRxFifo() to UART\_ResetRxFIFO().
- Rename function UART\_ResetTxFifo() to UART\_ResetTxFIFO().
- Rename function UART\_GetRxFifoLevel() to UART\_GetRxFIFOLevel().
- Rename function UART\_IsFifoError() to UART\_IsFIFOError().
- Rename function UART\_IsTxFifoEmpty() to UART\_IsTxFIFOEmpty().
- Rename function UART\_SetFifoTriggerLevel() to UART\_SetFIFOTriggerLevel().

#### ■ wdt.h

- Rename function WDT\_SetLoadVal() to WDT\_SetReloadValue().
- Rename function WDT\_GetLoadVal() to WDT\_GetReloadValue().

- Rename function WDT\_GetCounterVal() to WDT\_GetCounterValue().
- Add IAR for ARM tool-chain support.

V1.2.0/ 15-April-2019

## Main Changes

### ■ adc.h/.c

- Add 30 new functions
  - 1 new function to clear ADC interrupt overflow flag: `ADC_ClearOverflowInt()`.
  - 1 new function to get ADC interrupt overflow flag: `ADC_GetOverflowIntFlag()`.
  - 3 new functions controlling ADC interrupt trigger SOC: `ADC_EnableIntTriggerSOC()`, `ADC_DisableIntTriggerSOC()` and `ADC_SelectIntTriggerSOC()`.
  - 1 new function to set external SOC trigger: `ADC_SetExternalSOC()`.
  - 2 new functions controlling ADC SOC priority: `ADC_SetSOCPriority()` and `ADC_GetSOCPriority()`.
  - 1 new function to set the average counts for ADC result: `ADC_SetAverageCnt()`.
  - 1 new function to select S/H for SOC channel: `ADC_SetSOCSH()`.
  - 1 new function to get ADC result register value: `ADC_GetResult()`.
  - 1 new function to get ADC PPU result register value: `ADC_GetPPUResult()`.
  - 2 new functions controlling SOC delay capture: `ADC_SetSOCDelayCapture()` and `ADC_GetSOCDelay()`.
  - 4 new functions controlling ADC PPU unit: `ADC_PPUInit()`, `ADC_EnablePPU()`, `ADC_DisablePPU()` and `ADC_SetPPURef()`.
  - 6 new functions controlling ADC PPU interrupt: `ADC_EnablePPUInt()`, `ADC_DisablePPUInt()`, `ADC_ClearPPUInt()`, `ADC_ClearPPUGlobalInt()`, `ADC_GetPPUIntFlag()` and `ADC_GetPPUGlobalIntFlag()`.
  - 4 new functions controlling ADC PPU trip-zone event: `ADC_EnablePPUTripEvent()`, `ADC_DisablePPUTripEvent()`, `ADC_SetPPUTooHighThreshold()` and `ADC_SetPPUTooLowThreshold()`.
  - 2 new functions to get temperature using T-Sensor: `ADC_CalculateTemperature()` and `ADC_CalculatePreciseTemperature()`.
- `ADC_SetGainAndOffset()` function updated to support simultaneous sampling mode.
- Add new macro definition `TSENSOR_SLOPE` and `TSENSOR_OFFSET` for T-Sensor.
- Add new definition `ADC_SHSelEnum` for SOC S/H selection.
- Add new definition `ADC_PPUEnum`, `ADC_PPUEvtEnum` and `ADC_PPUPolEnum` for ADC PPU unit.
- The value of macro definition `ADC_DEFAULT_SAMPLE_TIME_NS` updated to 150.
- Remove 1 function
  - Remove function: `ADC_GetRawResult()`.

### ■ aes.h/.c

- Add 6 new functions
  - 1 new function to set AES running mode: `AES_SetRunningMode()`.
  - 1 new function to set the counter modular of CTR mode: `AES_SetCTRModular()`.
  - 2 new functions controlling AES CCM MIC value output: `AES_EnableMICOutput()` and `AES_DisableMICOutput()`.
  - 2 new functions controlling AES stream output: `AES_EnableStreamOutput()` and `AES_DisableStreamOutput()`.
- `AES_EncryptData()` and `AES_DecryptData()` functions updated for removing `AES_Reset()` function.

- Remove 1 function
  - Remove function: AES\_Reset().
- clock.h/.c
  - CLOCK\_PLLConfig() and CLOCK\_PLLQuickConfigWithRCO() functions updated to support all SPC1168 series.
  - CLOCK\_Init() and CLOCK\_InitWithRCO() functions updated to change UART clock divider.
  - Update definition *CLOCK\_HCLKSelEnum* and *CLOCK\_RefSelEnum*.
- comp.h/.c
  - Add 11 new functions
    - 1 new function to disable comparator output invert: COMP\_DisableOutputInvert().
    - 2 new functions controlling PWMSYNC clearing latched filter output status: COMP\_EnablePWMSyncClearFilterOutputStatus() and COMP\_DisablePWMSyncClearFilterOutputStatus().
    - 1 new function to select the synchronous output from PWM: COMP\_SetSyncEvent().
    - 1 new function to clear latched filter output status: COMP\_ClearFilterOutputStatus().
    - 1 new function to disable DAC: COMP\_DisableDAC().
    - 1 new function to select the PWM synchronous output signal for DAC: COMP\_SetDACSyncEvent().
    - 1 new function to set DAC code loading mode: COMP\_SetDACCodeLoadTiming().
    - 3 new functions controlling DAC buffer: COMP\_DACBufferInit(), COMP\_EnableDACBuffer() and COMP\_DisableDACBuffer().
  - Rename function COMP\_InvertOutput() to COMP\_EnableOutputInvert().
  - Rename function COMP\_ClearAllLatchedOutputStatus() to COMP\_ClearAllFilterOutputStatus().
  - Rename function COMP\_GetLatchedOutputStatus() to COMP\_GetFilterOutputStatus().
  - Rename function COMP\_GetFilterOutputStatus() to COMP\_GetRawFilterOutputStatus().
  - Rename function COMP\_SetFilterClkDIV() to COMP\_SetFilterClkDiv().
  - COMP\_Init() function updated to set new value for filter window size and threshold.
  - COMP\_SetFilterWindowTimeNs() function updated to make sure the filter threshold value is bigger than the half of window size.
- ecap.h/.c
  - Add 14 new functions
    - 5 new functions controlling ECAP synchronization: ECAP\_EnableSync(), ECAP\_DisableSync(), ECAP\_SetSyncReloadValue(), ECAP\_ForceSync() and ECAP\_SetSyncFromGPIO().
    - 1 new function to enable on-shot re-arming: ECAP\_OneshotReArm().
    - 2 new functions controlling capture operating mode: ECAP\_EnableOneshotMode() and ECAP\_DisableOneshotMode().
    - 1 new function to set event filter prescale: ECAP\_SetEventDiv().
    - 2 new functions controlling counter reset on capture event: ECAP\_EnableEventResetCounter() and ECAP\_DisableEventResetCounter().
    - 2 new functions selecting capture event polarity: ECAP\_EventTriggeredOnRisingEdge() and ECAP\_EventTriggeredOnFallingEdge().
    - 5 new functions controlling capture event interrupt: ECAP\_EnableInt(), ECAP\_DisableInt(), ECAP\_GetIntFlag(), ECAP\_ForceInt() and ECAP\_ClearInt().
  - ECAP\_SetInputPin() function updated to set input pin as GPIO input.

- Rename function `ECAP_SetCouterVal()` to `ECAP_SetCounterVal()`.
- Remove function `ECAP_SetSyncInputPin()`.
- Add `ECAP_EvtEnum` definition.
- Add `ECAP_IntEnum` definition.

#### ■ flash.h/.c

- 1 new function to set Flash write protection: `FLASH_WriteProtect()`.
- 5 functions updated to add Flash XIP module handling scheme: `FLASH_Read()`, `FLASH_ProgramWord()`, `FLASH_Program()`, `FLASH_EraseSector()` and `FLASH_EraseChip()`.
- Remove 2 functions: `FLASH_PowerUp()` and `FLASH_PowerDown()`.
- Remove macro definitions: `FLASH_CMD_POWER_UP`, `FLASH_CMD_VREAD0`, `FLASH_CMD_VREAD1`, `FLASH_CMD_PRE_PROG_WORD`, `FLASH_CMD_PRE_PROG_PAGE`, `FLASH_CMD_PROG_PAGE`, `FLASH_CMD_ERASE_BLOCK`, `FLASH_CMD_RECALL_READ`, `FLASH_CMD_SET_CONFIG` and `FLASH_CMD_POWER_DOWN`.

#### ■ gpio.h

- Add 3 new function
  - 1 new function to set Pin output strength: `GPIO_SetOutStrength()`.
  - 1 new function to get edge-triggered interrupt flag: `GPIO_GetGlobalEdgeIntStatus()`.
  - 1 new function to get level-triggered interrupt flag: `GPIO_GetGlobalLevelIntStatus()`.
- Rename function `GPIO_ClearEdgeIntAll()` to `GPIO_ClearGlobalEdgeInt()`.
- Rename function `GPIO_ClearLevelIntAll()` to `GPIO_ClearGlobalLevelInt()`.
- Add new definition `GPIO_OutStrengthEnum` for selecting GPIO output strength.

#### ■ i2c.h/.c

- Add 9 new functions
  - 4 new functions controlling I2C master sending and receiving data: `I2C_MasterWrite()`, `I2C_MasterRead()`, `I2C_MasterBulkWrite()` and `I2C_MasterBulkRead()`.
  - 4 new functions controlling I2C slave sending and receiving data: `I2C_SlaveWrite()`, `I2C_SlaveRead()`, `I2C_SlaveBulkWrite()` and `I2C_SlaveBulkRead()`.
  - 1 new function to initialize I2C speed setting: `I2C_SpeedInit()`.
- `I2C_MasterInit()` function updated to support multi-slave controlling.
- Rename function `I2C_SendByte()` to `I2C_WriteByte()`.
- Rename function `I2C_ReceiveByte()` to `I2C_ReadByte()`.

#### ■ power.h/.c

- Add 11 new functions
  - 6 new functions controlling BOD interrupt: `POWER_EnableBODInt()`, `POWER_DisableBODInt()`, `POWER_ClearBODInt()`, `POWER_ClearBODGlobalInt()`, `POWER_GetBODIntFlag()` and `POWER_GetBODGlobalIntFlag()`.
  - 3 new functions to initialize BOD for 1.2V power: `POWER_VDD12HBODInit()`, `POWER_VDD12LBODInit()` and `POWER_VDD12L1BODInit()`.
  - 2 new functions to initialize BOD for 3.3V power: `POWER_VDD33HBODInit()` and `POWER_VDD33LBODInit()`.
- Remove BOD interrupt control functions:
  - `POWER_EnableVDD12TooLow0Int()`
  - `POWER_DisableVDD12TooLow0Int()`
  - `POWER_EnableVDD12TooLow1Int()`

- POWER\_DisableVDD12TooLow1Int()
  - POWER\_EnableVDD12TooHighInt()
  - POWER\_DisableVDD12TooHighInt()
  - POWER\_ClearVDD12TooLow0Int()
  - POWER\_ClearVDD12TooLow1Int()
  - POWER\_ClearVDD12TooHighInt()
  - POWER\_GetVDD12TooLow0IntFlag()
  - POWER\_GetVDD12TooLow1IntFlag()
  - POWER\_GetVDD12TooHighIntFlag()
  - POWER\_EnableVDD33TooLowInt()
  - POWER\_DisableVDD33TooLowInt()
  - POWER\_EnableVDD33TooHighInt()
  - POWER\_DisableVDD33TooHighInt()
  - POWER\_ClearVDD33TooLowInt()
  - POWER\_ClearVDD33TooHighInt()
  - POWER\_GetVDD33TooLowIntFlag()
  - POWER\_GetVDD33TooHighIntFlag()
  - POWER\_ClearBODInt()
  - POWER\_GetBODIntFlag()
- Remove BOD initialize functions: POWER\_VDD12BODInit() and POWER\_VDD33BODInit().
  - Add new definitions: *POWER\_BODIntEnum*, *POWER\_VDD33HEnum*, *POWER\_VDD33LEnum*, *POWER\_VDD12HEnum* and *POWER\_VDD12LEnum*.

■ pwm.h/.c

- Add 111 new functions
  - 16 new functions controlling PWM register link: PWM\_UnlinkTBPRD(), PWM\_UnlinkCMPA(), PWM\_UnlinkCMPB(), PWM\_UnlinkCMPC(), PWM\_UnlinkCMPD(), PWM\_UnlinkDBRED(), PWM\_UnlinkDBFED(), PWM\_UnlinkGLDCTL1(), PWM\_LinkTBPRD(), PWM\_LinkCMPA(), PWM\_LinkCMPB(), PWM\_LinkCMPC(), PWM\_LinkCMPD(), PWM\_LinkDBRED(), PWM\_LinkDBFED() and PWM\_LinkGLDCTL1().
  - 2 new functions controlling PWM period value: PWM\_SetPeriod() and PWM\_GetPeriod().
  - 4 new functions for PWM synchronization controlling: PWM\_EnableSync(), PWM\_DisableSync(), PWM\_SetSyncOutEvent() and PWM\_SetCounterDirAfterSync().
  - 4 new functions for getting Compare register value: PWM\_GetCMPA(), PWM\_GetCMPB(), PWM\_GetCMPC() and PWM\_GetCMPD().
  - 9 new functions controlling Dead-Band register loading: PWM\_SetDBCTLLoadTiming(), PWM\_SetDBREDLoadTiming(), PWM\_SetDBFEDLoadTiming(), PWM\_UnlockDBCTL(), PWM\_UnlockDBRED(), PWM\_UnlockDBFED(), PWM\_LockDBCTL(), PWM\_LockDBRED() and PWM\_LockDBFED().
  - 2 new functions for getting Dead-Band delay: PWM\_GetDeadBandRisingDelay() and PWM\_GetDeadBandFallingDelay().
  - 2 new functions selecting Trip-zone events: PWM\_SetOneShotTripEvent() and PWM\_SetCBCTripEvent().
  - 16 new functions controlling Digital-Compare trip interrupt: PWM\_EnableDCAEVT0TripInt(), PWM\_DisableDCAEVT0TripInt(), PWM\_EnableDCAEVT1TripInt(), PWM\_DisableDCAEVT1TripInt(), PWM\_EnableDCAEVT2TripInt(), PWM\_DisableDCAEVT2TripInt(), PWM\_EnableDCAEVT3TripInt(), PWM\_DisableDCAEVT3TripInt(), PWM\_EnableDCAEVT4TripInt(), PWM\_DisableDCAEVT4TripInt(), PWM\_EnableDCAEVT5TripInt(), PWM\_DisableDCAEVT5TripInt(), PWM\_EnableDCAEVT6TripInt(), PWM\_DisableDCAEVT6TripInt(), PWM\_EnableDCAEVT7TripInt(), PWM\_DisableDCAEVT7TripInt().



- PWM\_DisableDCAEVT1TripInt(),
  - PWM\_DisableDCBEVT0TripInt(),
  - PWM\_DisableDCBEVT1TripInt(),
  - PWM\_GetDCAEVT1TripIntFlag(),
  - PWM\_GetDCBEVT1TripIntFlag(),
  - PWM\_ClearDCAEVT1TripInt(),
  - PWM\_ClearDCBEVT0TripInt()
  - PWM\_ClearDCBEVT1TripInt().
- PWM\_EnableDCBEVT0TripInt(),
  - PWM\_EnableDCBEVT1TripInt(),
  - PWM\_GetDCAEVT0TripIntFlag(),
  - PWM\_GetDCBEVT0TripIntFlag(),
  - PWM\_ClearDCAEVT0TripInt(),
  - PWM\_ClearDCBEVT0TripInt() and
- 1 new function to clear CBC trip interrupt flag by hardware: PWM\_ClearCBCTripIntByHardware().
- 8 new functions controlling Digital-Compare trip events: PWM\_EnableDCAHTripEvent(), PWM\_EnableDCALTripEvent(), PWM\_EnableDCBHTripEvent(), PWM\_EnableDCBLTripEvent(), PWM\_DisableDCAHTripEvent(), PWM\_DisableDCALTripEvent(), PWM\_DisableDCBHTripEvent() and PWM\_DisableDCBLTripEvent().
- 8 new functions setting Digital-Compare events: PWM\_SetRawDCAEVT0(), PWM\_SetRawDCAEVT1(), PWM\_SetRawDCBEVT0(), PWM\_SetRawDCBEVT1(), PWM\_SetDCAEVT0(), PWM\_SetDCAEVT1(), PWM\_SetDCBEVT0() and PWM\_SetDCBEVT1().
- 4 new functions controlling Digital-Compare events trigger synchronization: PWM\_EnableDCAEVT0TriggerSync(), PWM\_DisableDCAEVT0TriggerSync(), PWM\_EnableDCBEVT0TriggerSync() and PWM\_DisableDCBEVT0TriggerSync().
- 4 new functions controlling Digital-Compare events trigger ADC SOC: PWM\_EnableDCAEVT0TriggerSOC(), PWM\_DisableDCAEVT0TriggerSOC(), PWM\_EnableDCBEVT0TriggerSOC() and PWM\_DisableDCBEVT0TriggerSOC().
- 8 new functions controlling Digital Filter: PWM\_SetDCFilter(), PWM\_EnableDCFilterBlank(), PWM\_DisableDCFilterBlank(), PWM\_EnableDCFilterFromOtherPWM(), PWM\_DisableDCFilterFromOtherPWM(), PWM\_EnableDCFilterBlankInvert(), PWM\_DisableDCFilterBlankInvert() and PWM\_SetDCFilterBlankWindow().
- 6 new functions for clearing and getting SOC event flag: PWM\_ClearSOCAEvent(), PWM\_ClearSOCBEvent(), PWM\_ClearSOCCEvent(), PWM\_GetSOCAEventFlag(), PWM\_GetSOCBEventFlag() and PWM\_GetSOCCEventFlag().
- 1 new function to software force clock synchronization for all PWM modules: PWM\_ForceClockSync().
- 1 new function to software force PWM synchronization: PWM\_ForceSync().
- 3 new function controlling PWM synchronization by GPIO: PWM\_EnableSyncFromGPIO(), PWM\_DisableSyncFromGPIO() and PWM\_SetSyncFromGPIO().
- 6 new functions controlling PWM synchronization by Timers: PWM\_EnableSyncFromTimer0(), PWM\_EnableSyncFromTimer1(), PWM\_EnableSyncFromTimer2(), PWM\_DisableSyncFromTimer0(), PWM\_DisableSyncFromTimer1() and PWM\_DisableSyncFromTimer2().
- 5 new functions setting Trip-zone event from GPIO: PWM\_SetTZ0FromGPIO(), PWM\_SetTZ1FromGPIO(), PWM\_SetTZ2FromGPIO(), PWM\_SetTZ3FromGPIO() and PWM\_SetTZ4FromGPIO().
- 1 new function to calculate actual Time-Base Phase Register value: PWM\_CalSyncReloadValue().
- Rename function PWM\_EnableCMPALoad() to PWM\_UnlockCMPA().

- Rename function `PWM_EnableCMPBLoad()` to `PWM_UnlockCMPB()`.
- Rename function `PWM_EnableCMPCLoad()` to `PWM_UnlockCMPC()`.
- Rename function `PWM_EnableCMPDLoad()` to `PWM_UnlockCMPD()`.
- Rename function `PWM_DisableCMPALoad()` to `PWM_LockCMPA()`.
- Rename function `PWM_DisableCMPBLoad()` to `PWM_LockCMPB()`.
- Rename function `PWM_DisableCMPCLoad()` to `PWM_LockCMPC()`.
- Rename function `PWM_DisableCMPDLoad()` to `PWM_LockCMPD()`.
- Rename function `PWM_EnableAQCTLALoad()` to `PWM_UnlockAQCTLA()`.
- Rename function `PWM_EnableAQCTLBLoad()` to `PWM_UnlockAQCTLB()`.
- Rename function `PWM_DisableAQCTLALoad()` to `PWM_LockAQCTLA()`.
- Rename function `PWM_DisableAQCTLBLoad()` to `PWM_LockAQCTLB()`.
- Rename function `PWM_SetT0EventSource()` to `PWM_SetT0Event()`.
- Rename function `PWM_SetT1EventSource()` to `PWM_SetT1Event()`.
- Rename function `PWM_DeadBandRisingDelay()` to `PWM_SetDeadBandRisingDelay()`.
- Rename function `PWM_DeadBandFallingDelay()` to `PWM_SetDeadBandFallingDelay()`.
- Rename function `PWM_GetGlobalTripIntFlag()` to `PWM_GetTripGlobalIntFlag()`.
- Rename function `PWM_ClearGlobalTripInt()` to `PWM_ClearTripGlobalInt()`.
- Rename function `PWM_EnableSOCATrig()` to `PWM_EnableSOCA()`.
- Rename function `PWM_DisableSOCATrig()` to `PWM_DisableSOCA()`.
- Rename function `PWM_EnableSOCBTrig()` to `PWM_EnableSOCB()`.
- Rename function `PWM_DisableSOCBTrig()` to `PWM_DisableSOCB()`.
- Rename function `PWM_EnableSOCCTrig()` to `PWM_EnableSOCC()`.
- Rename function `PWM_DisableSOCCTrig()` to `PWM_DisableSOCC()`.
- Rename function `PWM_EnableTimeEvtINT()` to `PWM_EnableTimeEventInt()`.
- Rename function `PWM_DisableTimeEvtINT()` to `PWM_DisableTimeEventInt()`.
- Rename function `PWM_SetTimeEvtTiming()` to `PWM_SetTimeEventTiming()`.
- Rename function `PWM_SetTimeEvtPeriod()` to `PWM_SetTimeEventPeriod()`.
- Rename function `PWM_ClearTimeEvtInt()` to `PWM_ClearTimeEventInt()`.
- Rename function `PWM_GetTimeEvtIntFlag()` to `PWM_GetTimeEventIntFlag()`.
- Remove function `PWM_GetTripSelCompIndex()`.
- Remove function `PWM_SetOneshotTripFromExtPin()`.
- Remove function `PWM_EnableOneShotTripFromComp()`.
- Add definition `PWM_SelEnum` and `PWM_IncEnum` for selecting PWM modules.
- Add definition `PWM_TripEventEnum` for selecting trip events.
- Add definition `PWM_TripOutputEnum` for trip-zone output type.
- Add definition `PWM_DCTripEventEnum` for Digital-Compare trip events.
- Add definition `PWM_DCEventEnum` for Digital-Compare events.
- Add definition `PWM_RawDCEventEnum` for raw Digital-Compare events.
- Add definition `PWM_DCFilterInputEnum` for Digital-Compare filter input.
- Add definition `PWM_DCFilterAlignEnum` for Digital-Compare filter alignment.
- Add definition `PWM_SyncEventEnum` for synchronization events.
- Rename definition `PWM_TripZoneOutputEnum` to `PWM_TripActionEnum`.
- Update the member name in definition `PWM_TxEventEnum`.
- Remove definition `PWM_PWMOutputTripEnum`.

- ssp.h/.c
  - Add 5 new functions
    - 2 new functions controlling FIFO pack mode: SSP\_EnableFifoPackMode() and SSP\_DisableFifoPackMode().
    - 2 new functions controlling SSP master full duplex transfer: SSP\_MasterTransceive() and SSP\_MasterB2BTransceive().
    - 1 new functions controlling SSP slave full duplex transfer: SSP\_SlaveTransceive().
  - SSP\_Send () function updated to support single frame mode.
  - Rename function SSP\_Recv() to SSP\_Receive().
  - Remove functions controlling Bit Count Error:
    - SSP\_EnableBitCountErrorInt()
    - SSP\_DisableBitCountErrorInt()
    - SSP\_IsBitCountError()
    - SSP\_ClearBitCountErrorInt().
- system.h/.c
  - Add 10 new functions
    - 4 new functions controlling memory error interrupt: SYSTEM\_EnableMemErrorInt(), SYSTEM\_DisableMemErrorInt(), SYSTEM\_ClearMemErrorInt() and SYSTEM\_GetMemErrorIntFlag().
    - 4 new functions controlling reset event: SYSTEM\_EnableAllResetEvent(), SYSTEM\_DisableAllResetEvent(), SYSTEM\_ClearResetEventStatus() and SYSTEM\_GetResetEventStatus().
    - 1 new function to get chip unique ID: SYSTEM\_GetUID().
    - 1 new function to get chip random number: SYSTEM\_GetRandomNum().
  - SYSTEM\_EnableResetEvent() function updated using macro define.
  - SYSTEM\_DisableResetEvent() function updated using macro define.
  - Rename function SYSTEM\_EnableMemErrorIntAll () to SYSTEM\_EnableAllMemErrorInt().
  - Rename function SYSTEM\_DisableMemErrorIntAll () to SYSTEM\_DisableAllMemErrorInt().
  - Rename function SYSTEM\_ClearMemErrorInt() to SYSTEM\_ClearMemErrorGlobalInt().
  - Rename function SYSTEM\_GetMemErrorIntFlag() to SYSTEM\_GetMemErrorGlobalIntFlag().
  - Rename function SYSTEM\_ClearResetOnWDT0RSTEvent() to SYSTEM\_ClearWDT0ResetStatus().
  - Rename function SYSTEM\_ClearResetOnWDT1RSTEvent() to SYSTEM\_ClearWDT1ResetStatus().
  - Rename function SYSTEM\_ClearResetOnSystemRSTEvent () to SYSTEM\_ClearSystemResetStatus().
  - Rename function SYSTEM\_IsResetTriggeredByWDT0RSTEvent() to SYSTEM\_GetWDT0ResetStatus().
  - Rename function SYSTEM\_IsResetTriggeredByWDT1RSTEvent() to SYSTEM\_GetWDT1ResetStatus().
  - Rename function SYSTEM\_IsResetTriggeredBySystemRSTEvent() to SYSTEM\_GetSystemResetStatus().
  - Remove some memory error controlling functions:
    - SYSTEM\_EnableROM1BitErrorInt()
    - SYSTEM\_DisableROM1BitErrorInt()
    - SYSTEM\_EnableROM2BitErrorInt()
    - SYSTEM\_DisableROM2BitErrorInt()

- SYSTEM\_EnableFlash1BitErrorInt()
- SYSTEM\_DisableFlash1BitErrorInt()
- SYSTEM\_EnableFlash2BitErrorInt()
- SYSTEM\_DisableFlash2BitErrorInt()
- SYSTEM\_EnableIRAMErrorInt()
- SYSTEM\_DisableIRAMErrorInt()
- SYSTEM\_EnableDRAMErrorInt()
- SYSTEM\_DisableDRAMErrorInt()
- SYSTEM\_EnableSIOErrorInt()
- SYSTEM\_DisableSIOErrorInt()
- SYSTEM\_ClearROM1BitErrorInt()
- SYSTEM\_ClearROM2BitErrorInt()
- SYSTEM\_ClearFlash1BitErrorInt()
- SYSTEM\_ClearFlash2BitErrorInt()
- SYSTEM\_ClearIRAMErrorInt()
- SYSTEM\_ClearDRAMErrorInt()
- SYSTEM\_ClearSIOErrorInt()
- SYSTEM\_GetROM1BitErrorIntFlag()
- SYSTEM\_GetROM2BitErrorIntFlag()
- SYSTEM\_GetFlash1BitErrorIntFlag()
- SYSTEM\_GetFlash2BitErrorIntFlag()
- SYSTEM\_GetIRAMErrorIntFlag()
- SYSTEM\_GetDRAMErrorIntFlag()
- SYSTEM\_GetSIOErrorIntFlag()
- Remove some reset event controlling functions:
  - SYSTEM\_ClearResetOnROMError()
  - SYSTEM\_ClearResetOnFlashError()
  - SYSTEM\_ClearResetOnIRAMError()
  - SYSTEM\_ClearResetOnDRAMError()
  - SYSTEM\_ClearResetOnSIOError()
  - SYSTEM\_ClearResetOnVDD12L0()
  - SYSTEM\_ClearResetOnVDD12L1()
  - SYSTEM\_ClearResetOnVDD12H()
  - SYSTEM\_ClearResetOnVDD33L()
  - SYSTEM\_ClearResetOnVDD33H()
  - SYSTEM\_ClearResetOnPLLUnlock()
  - SYSTEM\_ClearResetOnClkDetectError()
  - SYSTEM\_IsResetTriggeredByROMError()
  - SYSTEM\_IsResetTriggeredByFlashError()
  - SYSTEM\_IsResetTriggeredByIRAMError()
  - SYSTEM\_IsResetTriggeredByDRAMError()
  - SYSTEM\_IsResetTriggeredBySIOError()
  - SYSTEM\_IsResetTriggeredByVDD12L0()
  - SYSTEM\_IsResetTriggeredByVDD12L1()

- SYSTEM\_IsResetTriggeredByVDD12H()
  - SYSTEM\_IsResetTriggeredByVDD33L()
  - SYSTEM\_IsResetTriggeredByVDD33H()
  - SYSTEM\_IsResetTriggeredByPLLUnlock()
  - SYSTEM\_IsResetTriggeredByClkDetectError()
- Add definition *SYSTEM\_MemErrorEnum* for memory error events.
- Update a member name from *RESET\_EVENT\_SIO\_ERROR* to *RESET\_EVENT\_SIOO\_ERROR* in definition *SYSTEM\_ResetEvtEnum*.
- timer.h
  - Add 6 new functions
    - 2 new functions controlling hold counter mode: *TIMER\_EnableCounterHold()* and *TIMER\_DisableCounterHold()*.
    - 2 new functions controlling ADC SOC generation: *TIMER\_EnableADCSOC()* and *TIMER\_DisableADCSOC()*.
    - 2 new functions controlling PWMSYNC signal generation: *TIMER\_EnablePWMSync()* and *TIMER\_DisablePWMSync()*.
- wdt.h
  - Add new functions
    - 1 new function to stop WDT counter: *WDT\_Stop()*.
    - 4 new functions controlling WDT running when core halted or lockup: *WDT\_EnableRunWhenCoreHalt()*, *WDT\_DisableRunWhenCoreHalt()*, *WDT\_EnableRunWhenCoreLockup()* and *WDT\_DisableRunWhenCoreLockup()*.
    - 1 new function to get WDT load register value: *WDT\_GetLoadVal()*.
    - 1 new function to get WDT current counter value: *WDT\_GetCounterVal()*.
- spc1168\_reg.h/spc1168\_bitfield.h
  - Rename bit definition CNTMOD to CTRMOD in AESCTL0 register.
  - Remove register AESCTL1.
  - Remove bit definition BITCNTERRIE in SSPCTL1 register.
  - Remove bit definition BITCNTERR in SSPSTS register.
  - Update TZDCSEL Bit-Filed macro definition in spc1168\_bitfield.h.