

Release Notes for SPD1178 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
- `hv.h/.c`
 - Add macro definition `HV_TIMEOUT_MAX`
 - Modify `HV_Command()`
 - Modify `HV_InitReg()`
 - Add `HV_EnableLowVoltageOperation()`
- Example Code
 - Modify `Flash_M_Access_User`
 - Modify `Flash_User_FuncWrap`
 - Modify `AES/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 function ECAP_SetInputPin() to ECAP_SetInput().
 - Update function ECAP_SetInput().
- pwm.c
 - Update function PWM_ComplementaryPairChannelInit() and PWM_SingleChannelInit().
- hv_bitfield.h
 - Update macro definitions of PDRVOCTL0.MINNOV and PDRV1CTL0.MINNOV bit-fields.
- hv.h/hv.c
 - Add function HV_ConnectAnalogTestToADC().
 - Add function HV_ConnectTSensorToAnalogTest().
 - Remove function HV_WriteRegBits() and add function HV_WriteRegField().
- 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_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.
- epwr.h
 - Rename macro function `EPWR_DisableTripEventInputInvert()` to `EPWR_SetTripEventInputActiveHigh()`.
 - Rename macro function `EPWR_EnableTripEventInputInvert()` to `EPWR_SetTripEventInputActiveLow()`.
 - Rename macro function `EPWR_DisableTripEventOutputInvert()` to `EPWR_SetTripEventOutputActiveHigh()`.
 - Rename macro function `EPWR_EnableTripEventOutputInvert()` to `EPWR_SetTripEventOutputActiveLow()`.

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.
- Remove *predriver.h/.c* and *predriver_bitfield.h* files.
- Add *hv.h/.c* and *hv_bitfield.h* files.

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.
- `predriver.h`
- Add 1 function declaration: `PREDRIVER_ComInit()`.

V1.3/ 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_Rese()` 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_ClearSOCBEEvent(), PWM_ClearSOCCEvent(), PWM_GetSOCAEventFlag(), PWM_GetSOCBEEventFlag() 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.