S32K AUTOSAR MCAL:
--- Motor Control with Dual Shunt Sampling

AMP GPIS AE

Stephen Du

Dec 2019

SECURE CONNECTIONS FOR A SMARTER WORLD

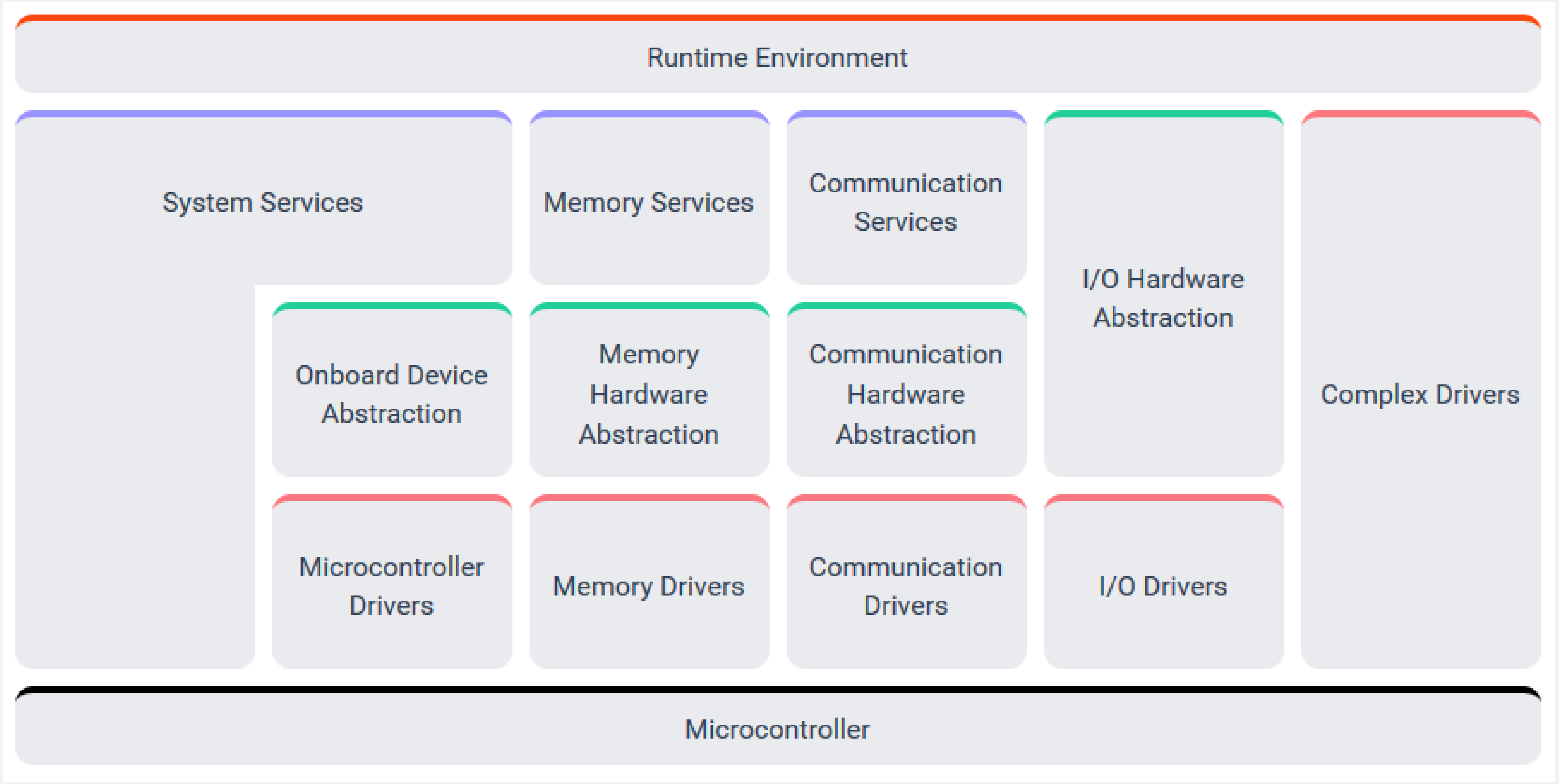
Agenda

AUTOSAR Overview --- Classic Platform

- **AUTOSAR** (**Aut**omotive **O**pen **S**ystem **Ar**chitecture) is a standardization initiative of leading automotive manufacturers and suppliers, founded in 2003.
- The idea behind AUTOSAR is to avoid continually re-developing the same or similar software components.
- The AUTOSAR concept is based on modular components with defined interfaces.
- Hardware and software are decoupled from one another.
- Reduced development effort and costs and improved quality.
- Re-use of development methods and tools.

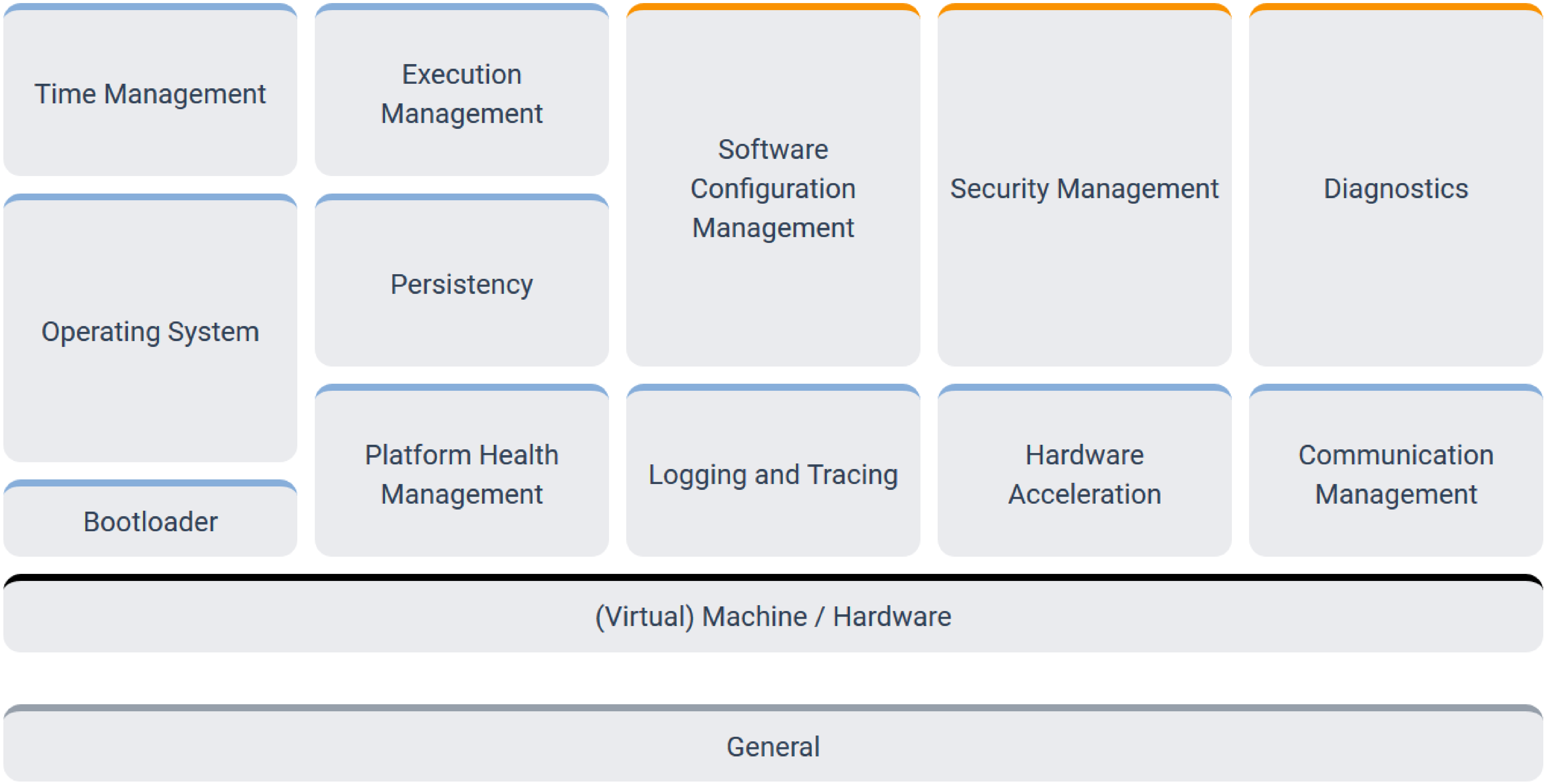
AUTOSAR Overview --- Classic Platform

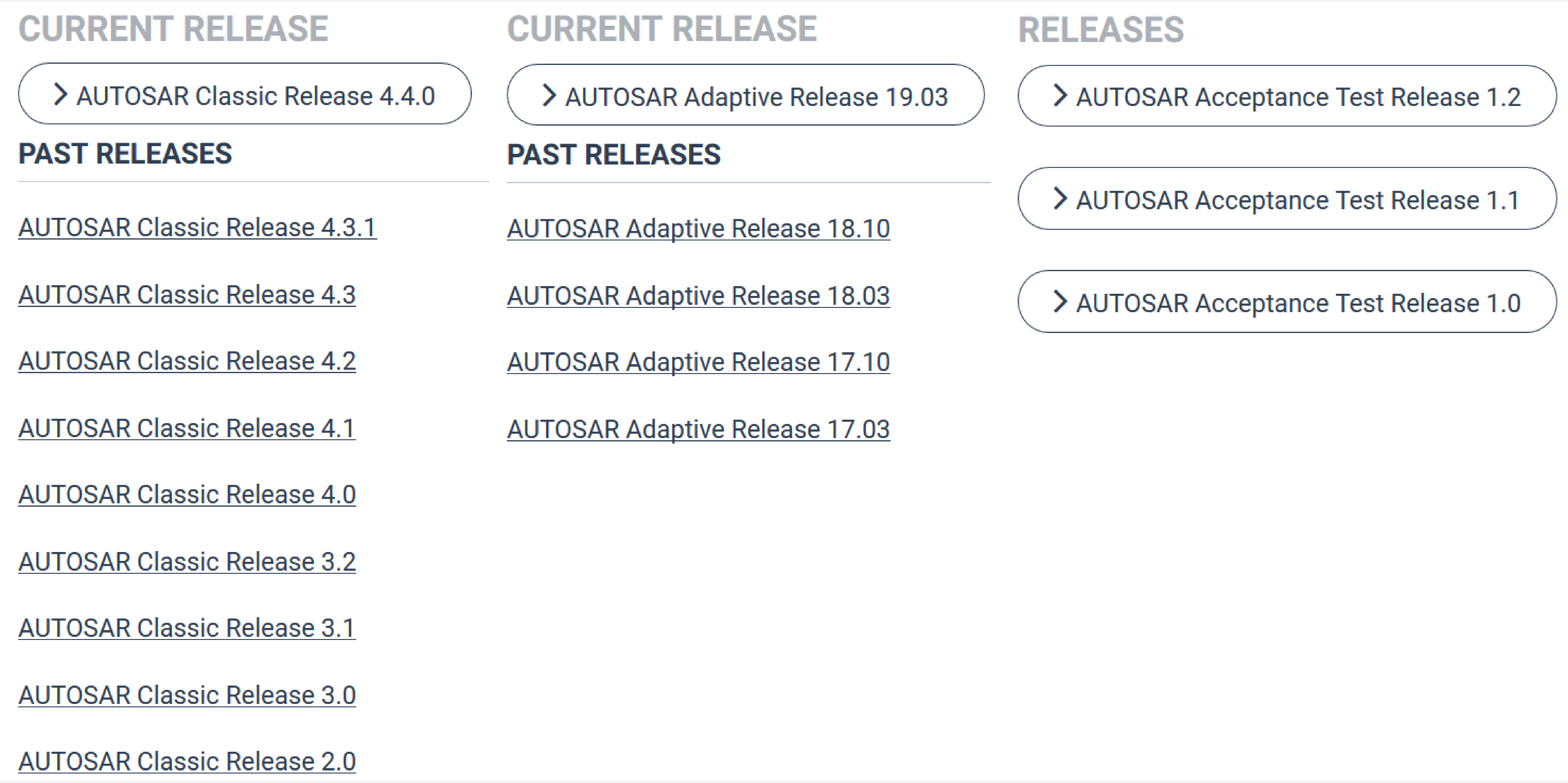
- The AUTOSAR Classic Platform architecture distinguishes on the highest abstraction level between three software layers which run on a microcontroller: application, runtime environment (RTE) and basic software (BSW).
  - The application software layer is mostly hardware independent.
  - Communication between software components and access to BSW via RTE.
  - The RTE represents the full interface for applications.
  - The BSW is divided in three major layers and complex drivers.
    - Services, ECU (Electronic Control Unit) abstraction and microcontroller abstraction.
  - Services are divided furthermore into functional groups representing the infrastructure for system, memory and communication services.


Runtime Environment
System Services
Memory Services
Communication Services
I/O Hardware Abstraction
Complex Drivers
Onboard Device Abstraction
Memory Hardware Abstraction
Communication Hardware Abstraction
Microcontroller Drivers
Memory Drivers
Communication Drivers
I/O Drivers
Microcontroller

AUTOSAR Overview --- Adaptive Platform

- The AUTOSAR Adaptive Platform is based on POSIX operating systems. The primary motivations are autonomous driving, Vehicle-to-X (V2X) applications and the growing external networking of vehicles (Connectivity).


Time Management
Execution Management
Software Configuration Management
Security Management
Diagnostics
Operating System
Persistency
Bootloader
Platform Health Management
Logging and Tracing
Hardware Acceleration
Communication Management
(Virtual) Machine / Hardware
General



CURRENT RELEASE

> AUTOSAR Classic Release 4.4.0

PAST RELEASES

AUTOSAR Classic Release 4.3.1

AUTOSAR Classic Release 4.3

AUTOSAR Classic Release 4.2

AUTOSAR Classic Release 4.1

AUTOSAR Classic Release 4.0

AUTOSAR Classic Release 3.2

AUTOSAR Classic Release 3.1

AUTOSAR Classic Release 3.0

AUTOSAR Classic Release 2.0

CURRENT RELEASE

> AUTOSAR Adaptive Release 19.03

PAST RELEASES

AUTOSAR Adaptive Release 18.10

AUTOSAR Adaptive Release 18.03

AUTOSAR Adaptive Release 17.10

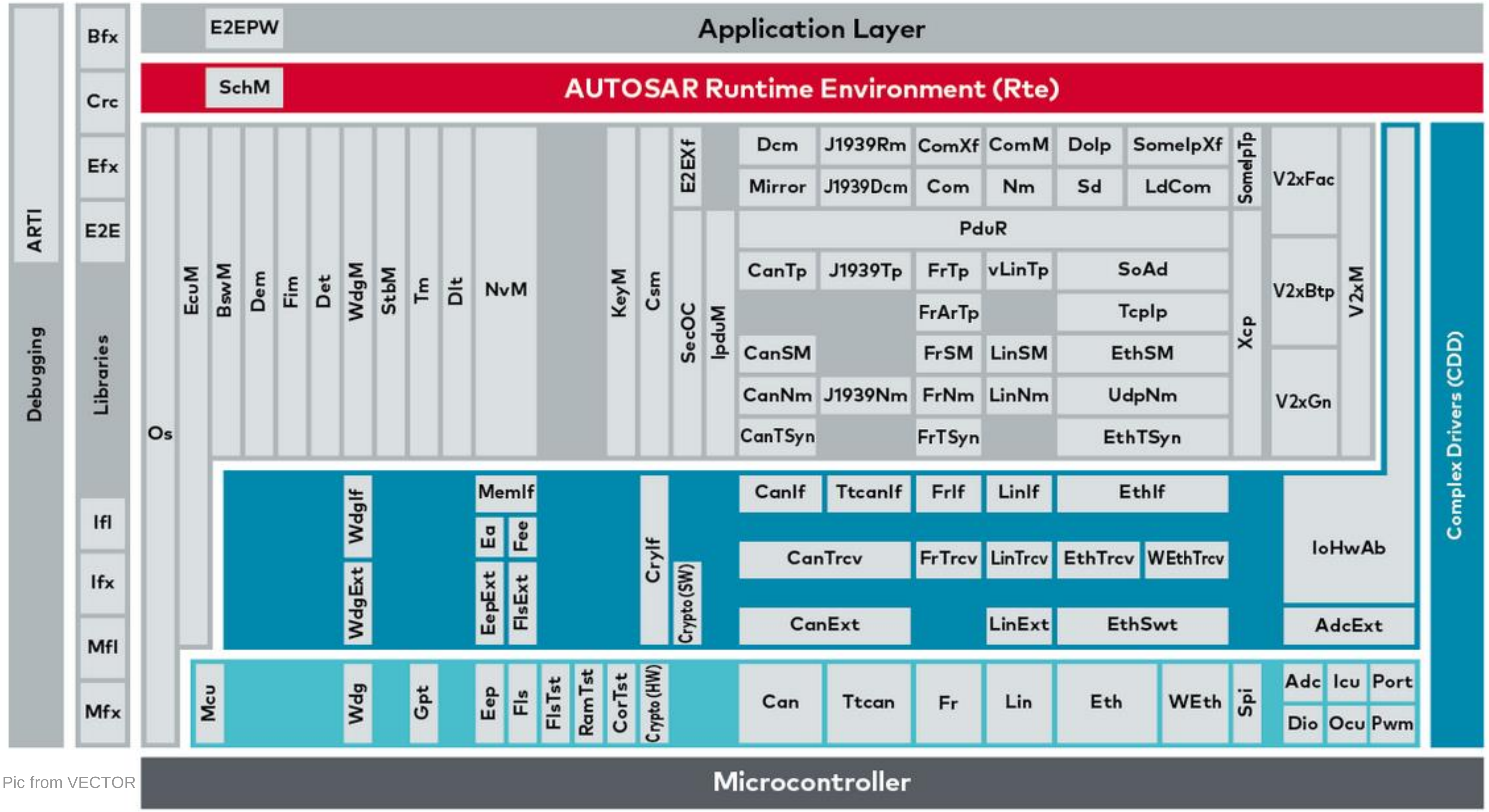
AUTOSAR Adaptive Release 17.03

RELEASES

> AUTOSAR Acceptance Test Release 1.2

> AUTOSAR Acceptance Test Release 1.1

> AUTOSAR Acceptance Test Release 1.0

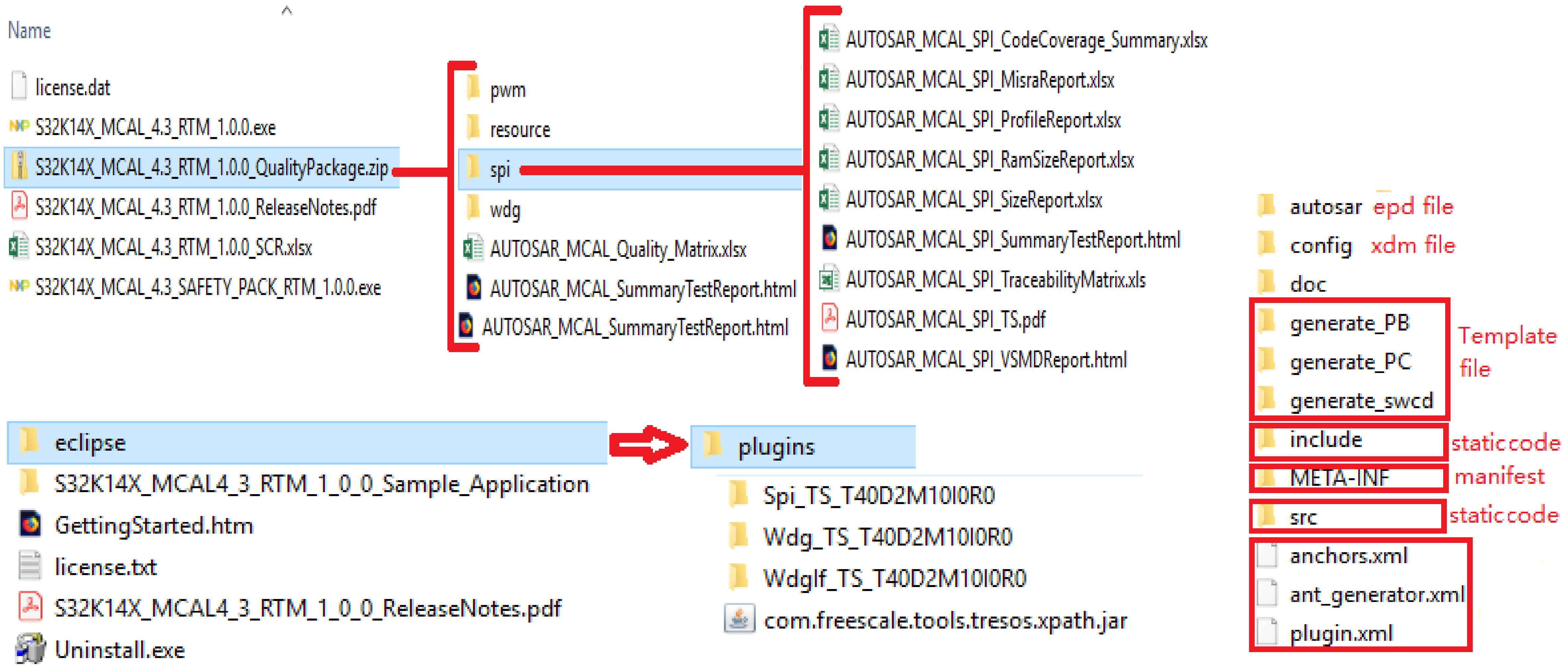


Pic from VECTOR

MCAL Overview

- **MCAL**(**M**icro**C**ontroller **A**bstraction **L**ayer) is a very important part of AUTOSAR. If you are not familiar with AUTOSAR/MCAL, you can think of it as the driver layer you usually write. The difference is that MCAL following AUTOSAR standard(Include CDD).

MCAL Overview --- NXP MCAL Package Content



Name

- license.dat
- S32K14X_MCAL_4.3_RTM_1.0.0.exe
- S32K14X_MCAL_4.3_RTM_1.0.0_QualityPackage.zip
- S32K14X_MCAL_4.3_RTM_1.0.0_ReleaseNotes.pdf
- S32K14X_MCAL_4.3_RTM_1.0.0_SCR.xlsx
- S32K14X_MCAL_4.3_SAFETY_PACK_RTM_1.0.0.exe

- pwm
- resource
- spi
- wdg
- AUTOSAR_MCAL_Quality_Matrix.xlsx
- AUTOSAR_MCAL_SummaryTestReport.html
- AUTOSAR_MCAL_SummaryTestReport.html

- AUTOSAR_MCAL_SPI_CodeCoverage_Summary.xlsx
- AUTOSAR_MCAL_SPI_MisraReport.xlsx
- AUTOSAR_MCAL_SPI_ProfileReport.xlsx
- AUTOSAR_MCAL_SPI_RamSizeReport.xlsx
- AUTOSAR_MCAL_SPI_SizeReport.xlsx
- AUTOSAR_MCAL_SPI_SummaryTestReport.html
- AUTOSAR_MCAL_SPI_TraceabilityMatrix.xls
- AUTOSAR_MCAL_SPI_TS.pdf
- AUTOSAR_MCAL_SPI_VSMDReport.html

- eclipse
- S32K14X_MCAL4_3_RTM_1_0_0_Sample_Application
- GettingStarted.htm
- license.txt
- S32K14X_MCAL4_3_RTM_1_0_0_ReleaseNotes.pdf
- Uninstall.exe

- plugins
- Spi_TS_T40D2M10I0R0
- Wdg_TS_T40D2M10I0R0
- WdgIf_TS_T40D2M10I0R0
- com.freescale.tools.tresos.xpath.jar

- autosar
- config
- doc
- generate_PB
- generate_PC
- generate_swcd
- include
- META-INF
- src
- anchors.xml
- ant_generator.xml
- plugin.xml

AUTOSAR FAQ

- Where can I get these material?

https://www.autosar.org (AUTOSAR)

https://www.nxp.com/autosar (MCAL)

- Is it free or need charge?

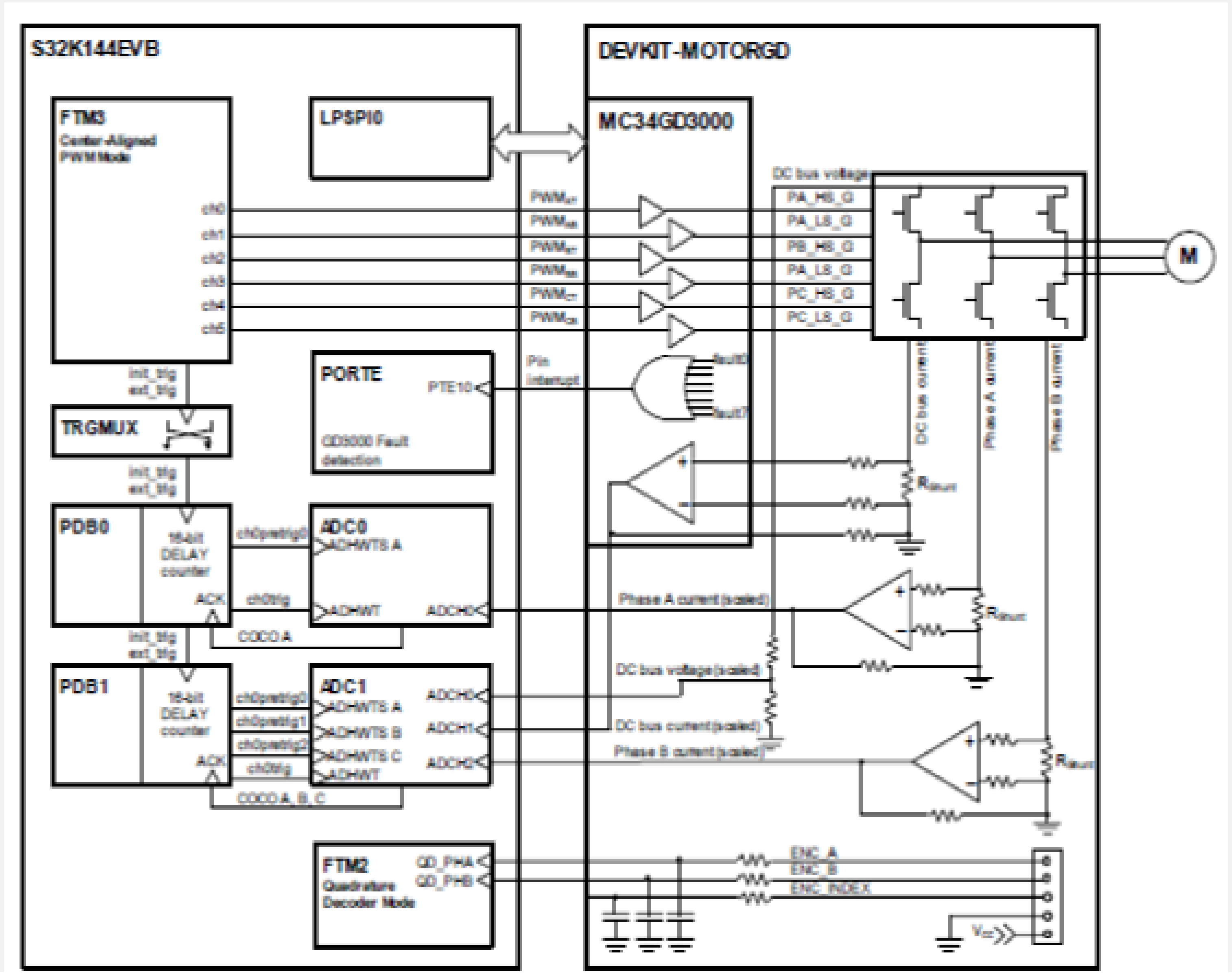
AUTOSAR :Free

MCAL: Depends on platform

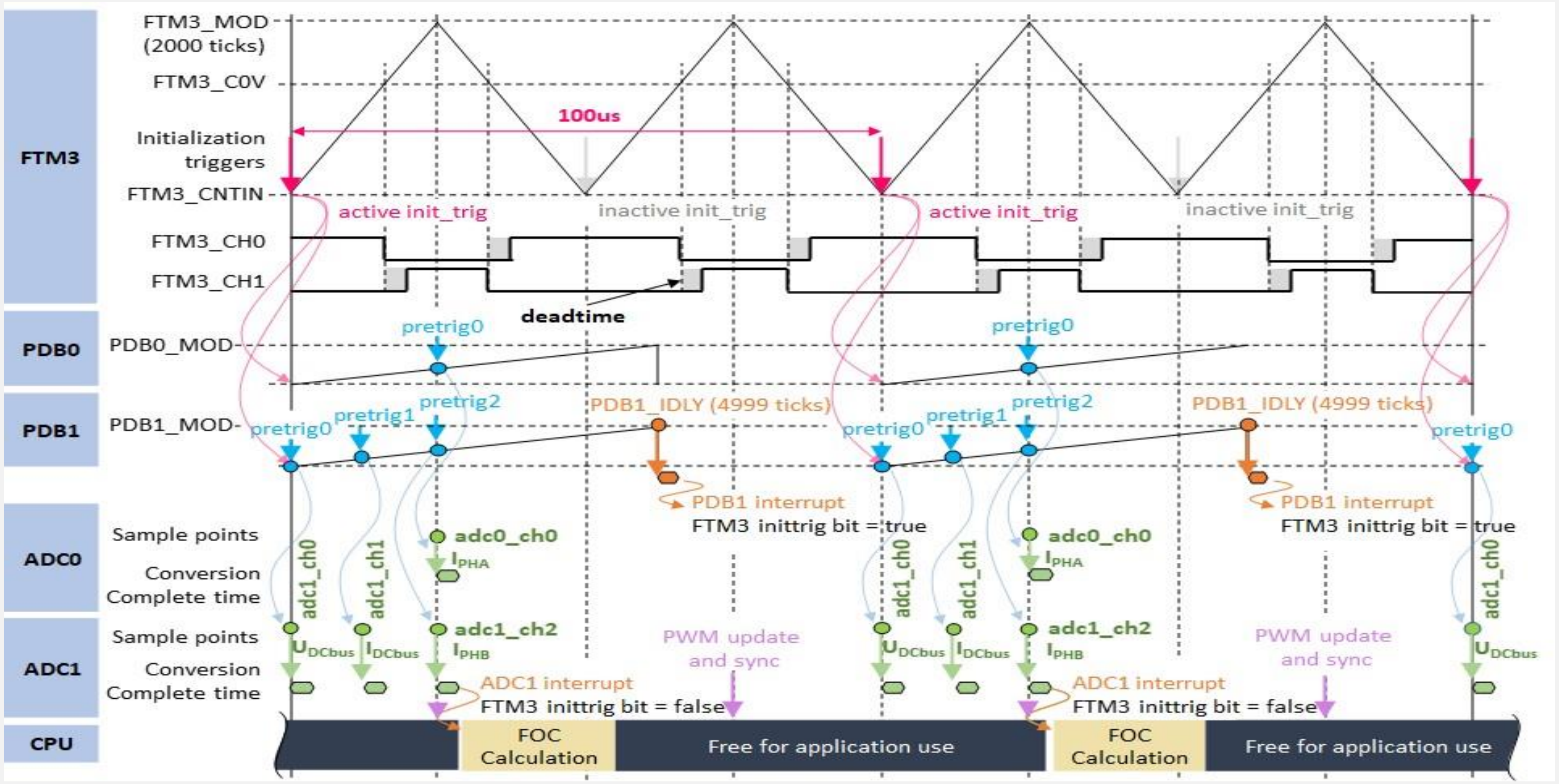
Development Environment

- Hardware
  - MTRDEVKSPNK144: S32K144 Development Kit for 3-phase PMSM
- Software
  - S32K14X_MCAL4_3_RTM_1_0_0
    - (We also provide SDK version)
- Lib
  - Math and Motor Control Library
- Tuning
  - MCATSW: Motor Control Application Tuning (MCAT) Tool
- Configuration&&Debug Tool
  - EB Tresos Version 24
  - S32 Design Studio

System Overview --- System Diagram



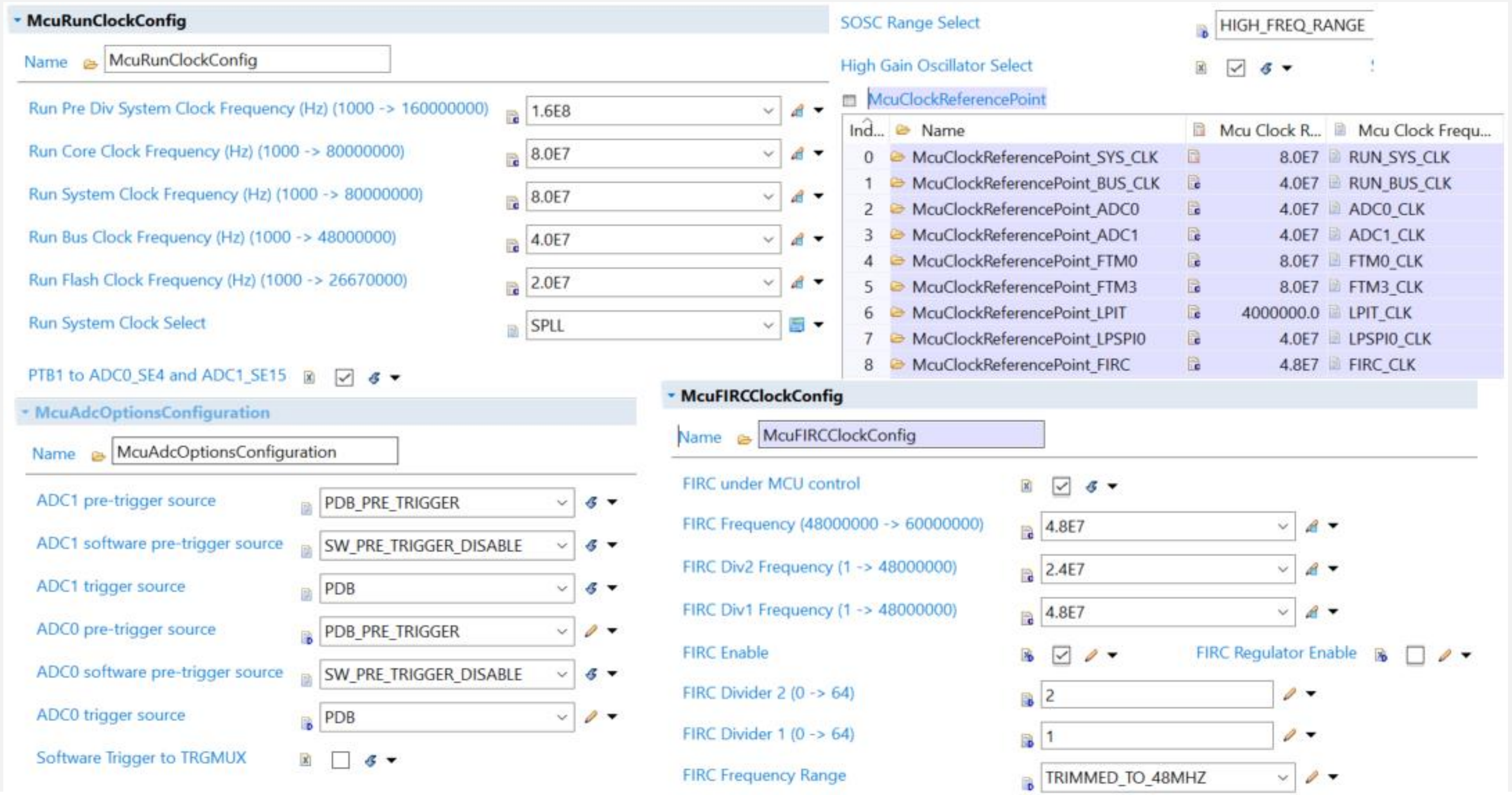
System Overview --- Sequence Diagram



Modules Configuration --- MCU

- Core Clock && System Clock && FTM: 80MHz
- Bus Clock && ADC && SPI: 40MHz
- Enable Run mode
- Enable ADC0_SE5 and ADC1_SE15’s Interleave
- Set PDB as ADC’s trigger source
- FIRC Enable(recommend), enable PLL
- SOSC Range: High
- Oscillator Gain: High

Modules Configuration --- MCU



McuRunClockConfig

Name	McuRunClockConfig
Run Pre Div System Clock Frequency (Hz) (1000 -> 160000000)	1.6E8
Run Core Clock Frequency (Hz) (1000 -> 80000000)	8.0E7
Run System Clock Frequency (Hz) (1000 -> 80000000)	8.0E7
Run Bus Clock Frequency (Hz) (1000 -> 48000000)	4.0E7
Run Flash Clock Frequency (Hz) (1000 -> 26670000)	2.0E7
Run System Clock Select	SPLL
PTB1 to ADC0_SE4 and ADC1_SE15	✓

McuAdcOptionsConfiguration

Name	McuAdcOptionsConfiguration
ADC1 pre-trigger source	PDB_PRE_TRIGGER
ADC1 software pre-trigger source	SW_PRE_TRIGGER_DISABLE
ADC1 trigger source	PDB
ADC0 pre-trigger source	PDB_PRE_TRIGGER
ADC0 software pre-trigger source	SW_PRE_TRIGGER_DISABLE
ADC0 trigger source	PDB
Software Trigger to TRGMUX	☐

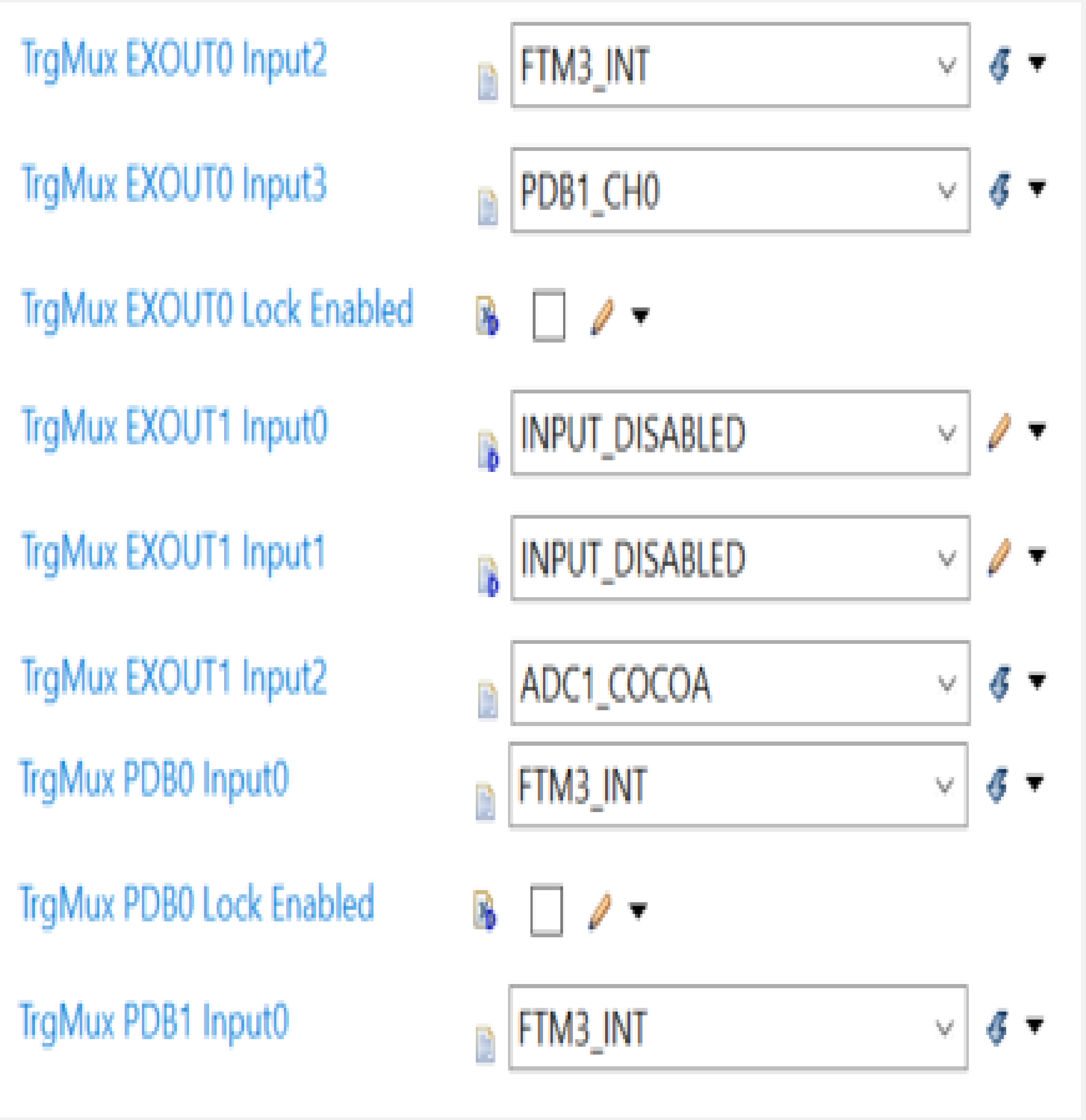
SOSC Range Select	HIGH_FREQ_RANGE
High Gain Oscillator Select	✓

McuClockReferencePoint

Ind...	Name	Mcu Clock R...	Mcu Clock Frequ...
0	McuClockReferencePoint_SYS_CLK	8.0E7	RUN_SYS_CLK
1	McuClockReferencePoint_BUS_CLK	4.0E7	RUN_BUS_CLK
2	McuClockReferencePoint_ADC0	4.0E7	ADC0_CLK
3	McuClockReferencePoint_ADC1	4.0E7	ADC1_CLK
4	McuClockReferencePoint_FTM0	8.0E7	FTM0_CLK
5	McuClockReferencePoint_FTM3	8.0E7	FTM3_CLK
6	McuClockReferencePoint_LPIT	4000000.0	LPIT_CLK
7	McuClockReferencePoint_LPSPI0	4.0E7	LPSPI0_CLK
8	McuClockReferencePoint_FIRC	4.8E7	FIRC_CLK

McuFIRCClockConfig

Name	McuFIRCClockConfig
FIRC under MCU control	✓
FIRC Frequency (48000000 -> 60000000)	4.8E7
FIRC Div2 Frequency (1 -> 48000000)	2.4E7
FIRC Div1 Frequency (1 -> 48000000)	4.8E7
FIRC Enable	✓
FIRC Regulator Enable	☐
FIRC Divider 2 (0 -> 64)	2
FIRC Divider 1 (0 -> 64)	1
FIRC Frequency Range	TRIMMED_TO_48MHZ



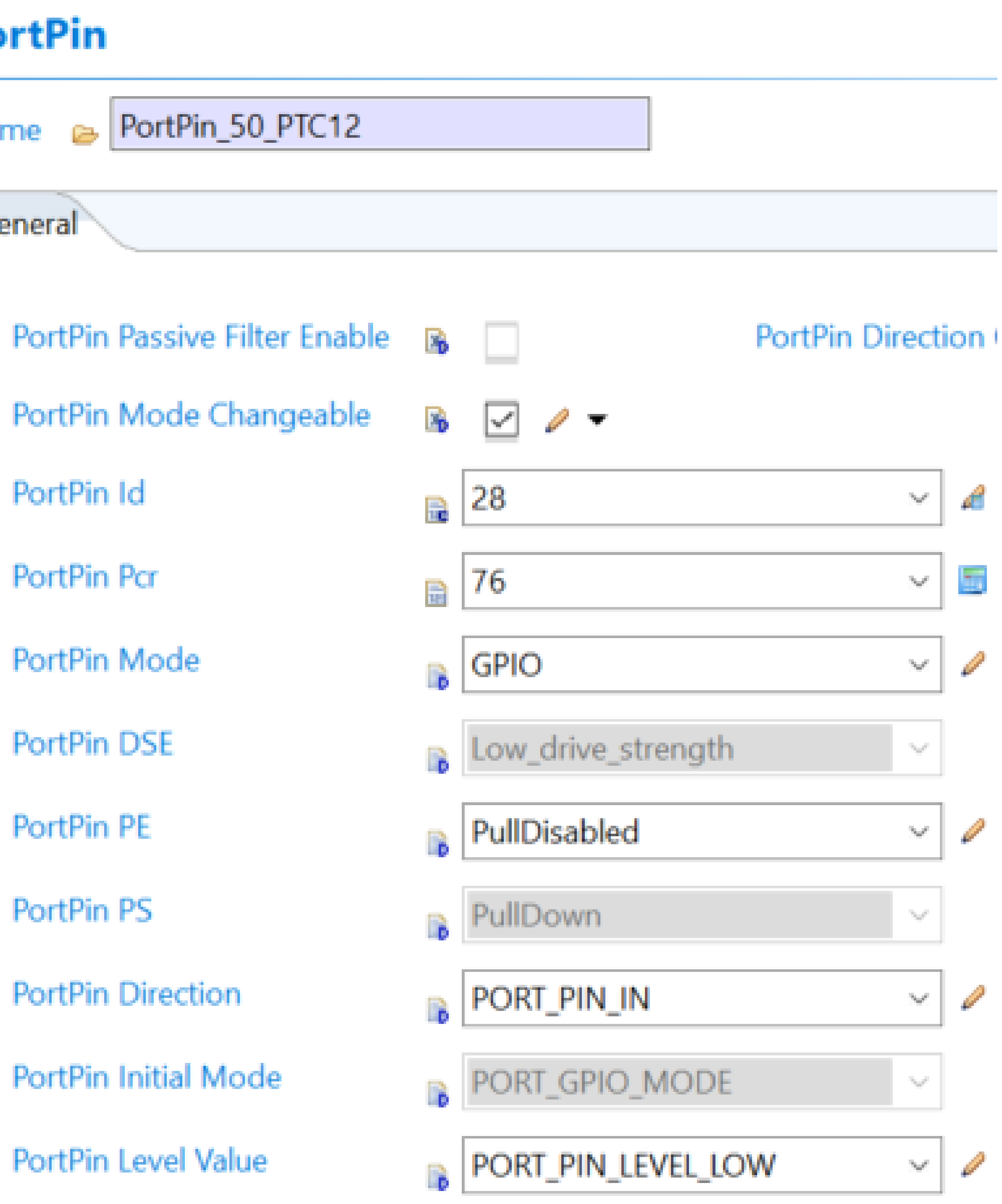
TrgMux EXOUT0 Input2	FTM3_INT
TrgMux EXOUT0 Input3	PDB1_CH0
TrgMux EXOUT0 Lock Enabled	☐
TrgMux EXOUT1 Input0	INPUT_DISABLED
TrgMux EXOUT1 Input1	INPUT_DISABLED
TrgMux EXOUT1 Input2	ADC1_COCOA
TrgMux PDB0 Input0	FTM3_INT
TrgMux PDB0 Lock Enabled	☐
TrgMux PDB1 Input0	FTM3_INT

Modules Configuration --- MCL

- This is a CDD module, responsible for TrgMux and DMA module's configuration
- In this system, we use trgMux module to debug
- Enable TrgMux Function
- Set FTM3_INT as TrgMux PDB0&&PDB1's input
- Set FTM3_INT as TrgMux OUT2's input
- Set PDB1_CHO as TrgMux OUT3's input
- Set ADC1_COCOA as TrgMux OUT6's input

Modules Configuration --- Port

Function	Pin Count : Pin
EXTAL/XTAL	2 : PTB7/PTB6
JTAG TCLK/TDI/TDO/SWD	4 : PTC4/PTC5/PTA10/PTA4
SPI SCK/SIN/SOUT	3 : PTB2/PTB3/PTB4
UART1 RX/TX	2 : PTC6/PTC7
TRGMUX OUT2/OUT3/OUT6	3 : PTD1/PTA0/PTE15
FTM(PWM) CH0~CH5	6 : PTB8~PTB11&PTC10&PTC11
ADC0_SE4/ADC1_SE6/SE7/SE15	4 : PTB0/PTB12/PTD4/PTB1
GPIO	12 : PTA2/A3/B5/15/C12/13/D0/2/14~16/E10


rtPin

me PortPin_50_PTC12

eneral

PortPin Passive Filter Enable ☐

PortPin Direction

PortPin Mode Changeable ✓

PortPin Id	28
PortPin Pcr	76
PortPin Mode	GPIO
PortPin DSE	Low_drive_strength
PortPin PE	PullDisabled
PortPin PS	PullDown
PortPin Direction	PORT_PIN_IN
PortPin Initial Mode	PORT_GPIO_MODE
PortPin Level Value	PORT_PIN_LEVEL_LOW

Modules Configuration --- Port

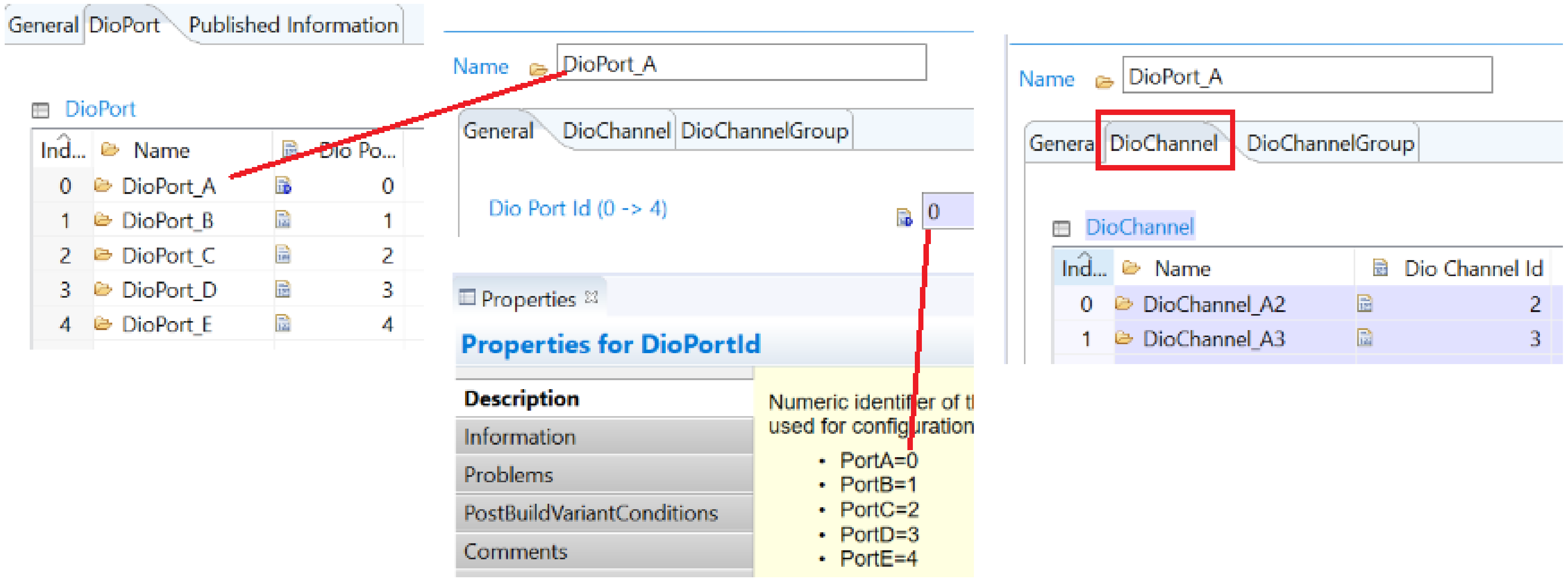
- Pcr Calc: CalcNum * 32 + PinNum
- Example:

  PTC12: 2*32+12 = 76

PortNum	CalcNum
PortA	0
PortB	1
PortC	2
PortD	3
PortE	4

Modules Configuration --- Dio

- This module responsible for setting the GPIO function, like the direction: input or output; default level: High or Low and so on.
- Each Pin belongs to different Port need create independent group.



Ind...	Name	Dio Po...
0	DioPort_A	0
1	DioPort_B	1
2	DioPort_C	2
3	DioPort_D	3
4	DioPort_E	4

Name: DioPort_A

Dio Port Id (0 -> 4): 0

Numeric identifier of t
used for configuration

- PortA=0
- PortB=1
- PortC=2
- PortD=3
- PortE=4

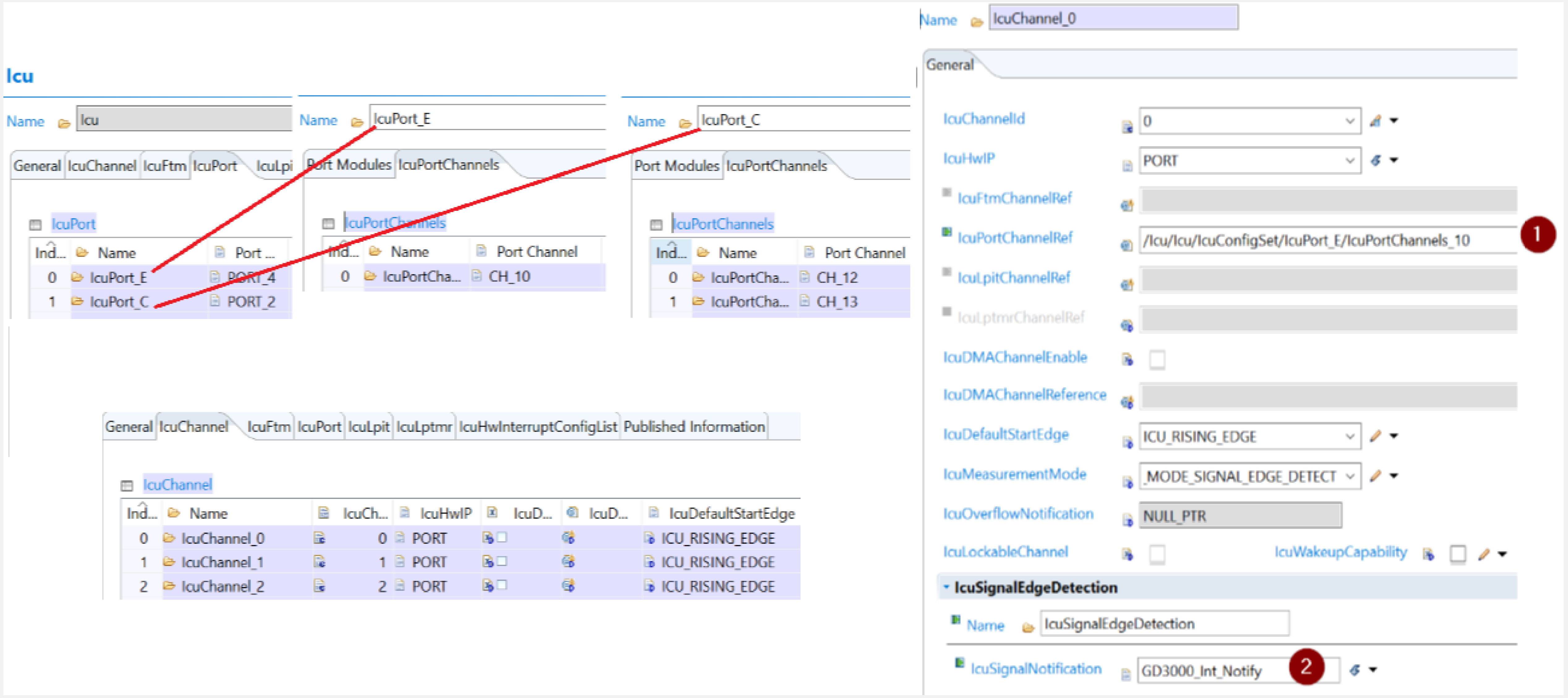
Name: DioPort_A

Ind...	Name	Dio Channel Id
0	DioChannel_A2	2
1	DioChannel_A3	3

Modules Configuration --- ICU

- The time measurement of Cyclic, high level, low level.
- Edge detection and notification.
- Edge counting and timestamp.
- Wakeup interrupt.
- Gpio interrupt belongs to the second function.
- We have three GPIO need enable interrupt function:
  - PTE10: GD3000 INT
  - PTC12: Button 1
  - PTC13: Button 2

Modules Configuration --- ICU



Icu

Name: Icu

Tabs: General | IcuChannel | IcuFtm | IcuPort | IcuLpi

IcuPort

Ind...	Name	Port ...
0	IcuPort_E	PORT_4
1	IcuPort_C	PORT_2

Name: IcuPort_E

Tabs: Port Modules | IcuPortChannels

IcuPortChannels

Ind...	Name	Port Channel
0	IcuPortCha...	CH_10

Name: IcuPort_C

Tabs: Port Modules | IcuPortChannels

IcuPortChannels

Ind...	Name	Port Channel
0	IcuPortCha...	CH_12
1	IcuPortCha...	CH_13

Tabs: General | IcuChannel | IcuFtm | IcuPort | IcuLpit | IcuLptmr | IcuHwInterruptConfigList | Published Information

IcuChannel

Ind...	Name	IcuCh...	IcuHwIP	IcuD...	IcuD...	IcuDefaultStartEdge
0	IcuChannel_0	0	PORT	☐		ICU_RISING_EDGE
1	IcuChannel_1	1	PORT	☐		ICU_RISING_EDGE
2	IcuChannel_2	2	PORT	☐		ICU_RISING_EDGE

Name: IcuChannel_0

General

IcuChannelId	0
IcuHwIP	PORT
IcuFtmChannelRef	
IcuPortChannelRef	/Icu/Icu/IcuConfigSet/IcuPort_E/IcuPortChannels_10
IcuLpitChannelRef	
IcuLptmrChannelRef	
IcuDMAChannelEnable	☐
IcuDMAChannelReference	
IcuDefaultStartEdge	ICU_RISING_EDGE
IcuMeasurementMode	MODE_SIGNAL_EDGE_DETECT
IcuOverflowNotification	NULL_PTR
IcuLockableChannel	☐
IcuWakeupCapability	☐

1

IcuSignalEdgeDetection

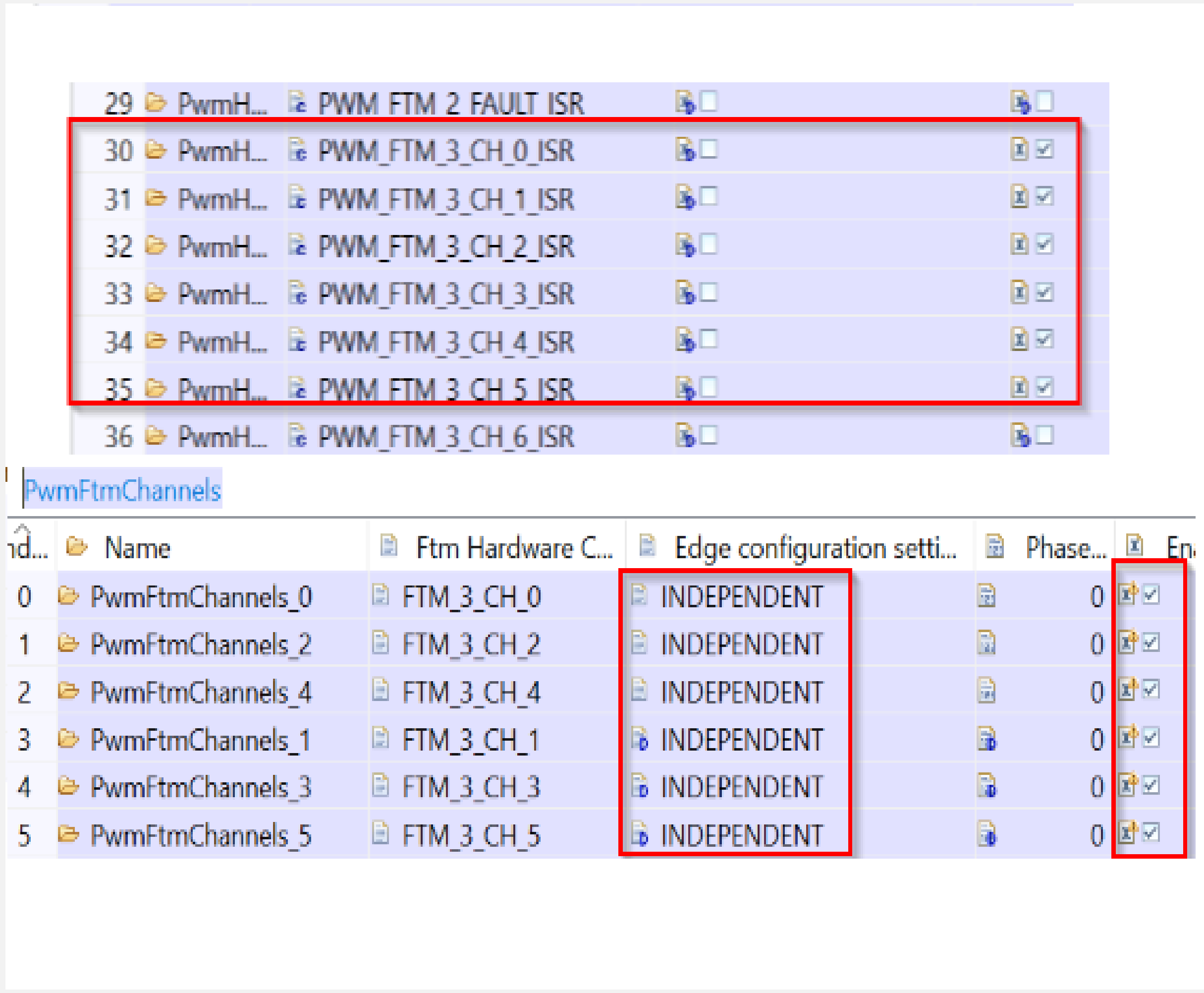
Name	IcuSignalEdgeDetection
IcuSignalNotification	GD3000_Int_Notify

2

Modules Configuration --- PWM

- Enable related channel(Total 6 channels): FTM3_CH0 ~ FTM3_CH5
- Edge configuration: Independent mode
- Align mode: Center align mode
- Cycle: 50us
- Delay: 32 Cycle

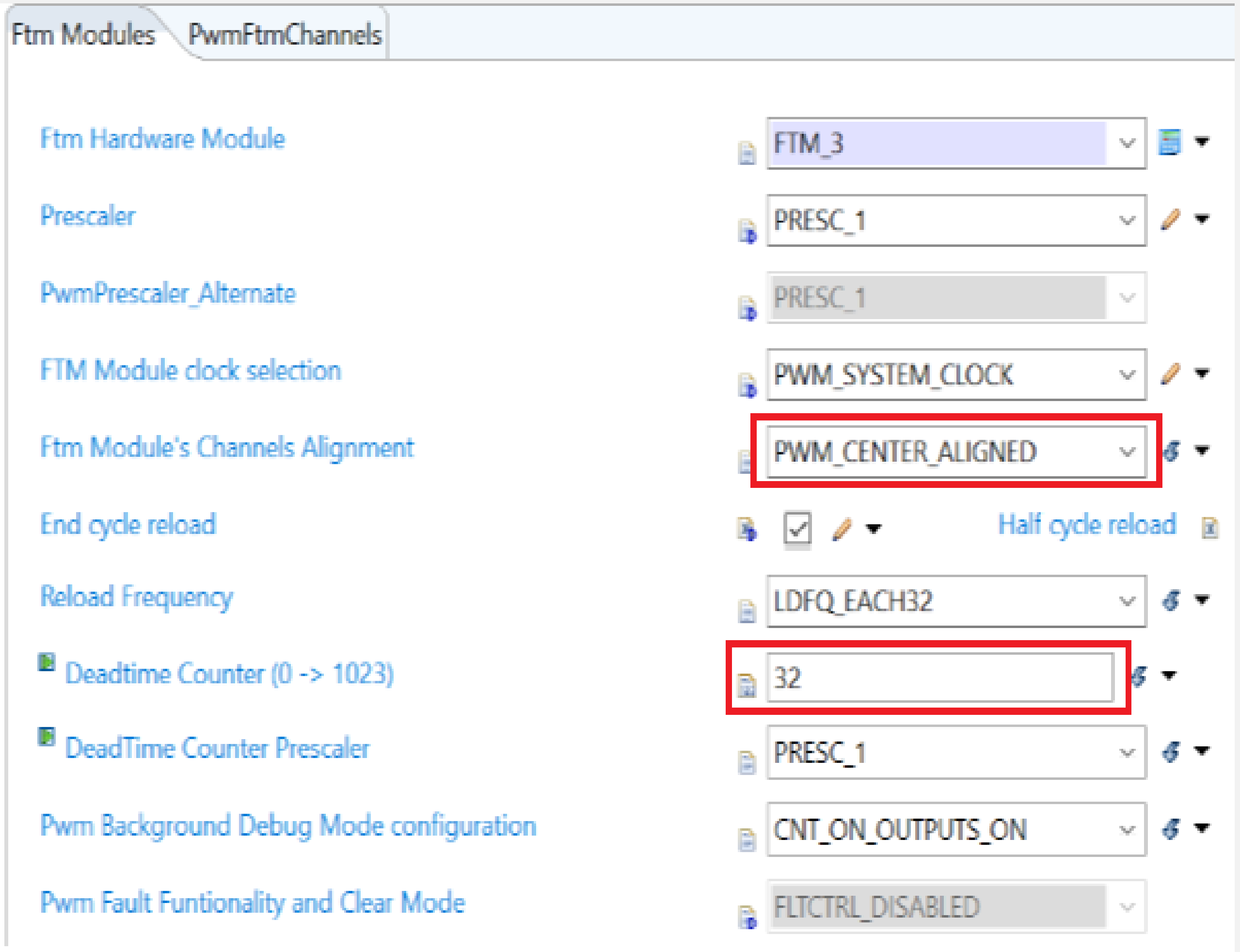
Modules Configuration --- PWM



29	PwmH...	PWM_FTM_2_FAULT_ISR	☐	☐
30	PwmH...	PWM_FTM_3_CH_0_ISR	☐	✓
31	PwmH...	PWM_FTM_3_CH_1_ISR	☐	✓
32	PwmH...	PWM_FTM_3_CH_2_ISR	☐	✓
33	PwmH...	PWM_FTM_3_CH_3_ISR	☐	✓
34	PwmH...	PWM_FTM_3_CH_4_ISR	☐	✓
35	PwmH...	PWM_FTM_3_CH_5_ISR	☐	✓
36	PwmH...	PWM_FTM_3_CH_6_ISR	☐	☐

PwmFtmChannels

Id...	Name	Ftm Hardware C...	Edge configuration setti...	Phase...	En...
0	PwmFtmChannels_0	FTM_3_CH_0	INDEPENDENT	0	✓
1	PwmFtmChannels_2	FTM_3_CH_2	INDEPENDENT	0	✓
2	PwmFtmChannels_4	FTM_3_CH_4	INDEPENDENT	0	✓
3	PwmFtmChannels_1	FTM_3_CH_1	INDEPENDENT	0	✓
4	PwmFtmChannels_3	FTM_3_CH_3	INDEPENDENT	0	✓
5	PwmFtmChannels_5	FTM_3_CH_5	INDEPENDENT	0	✓



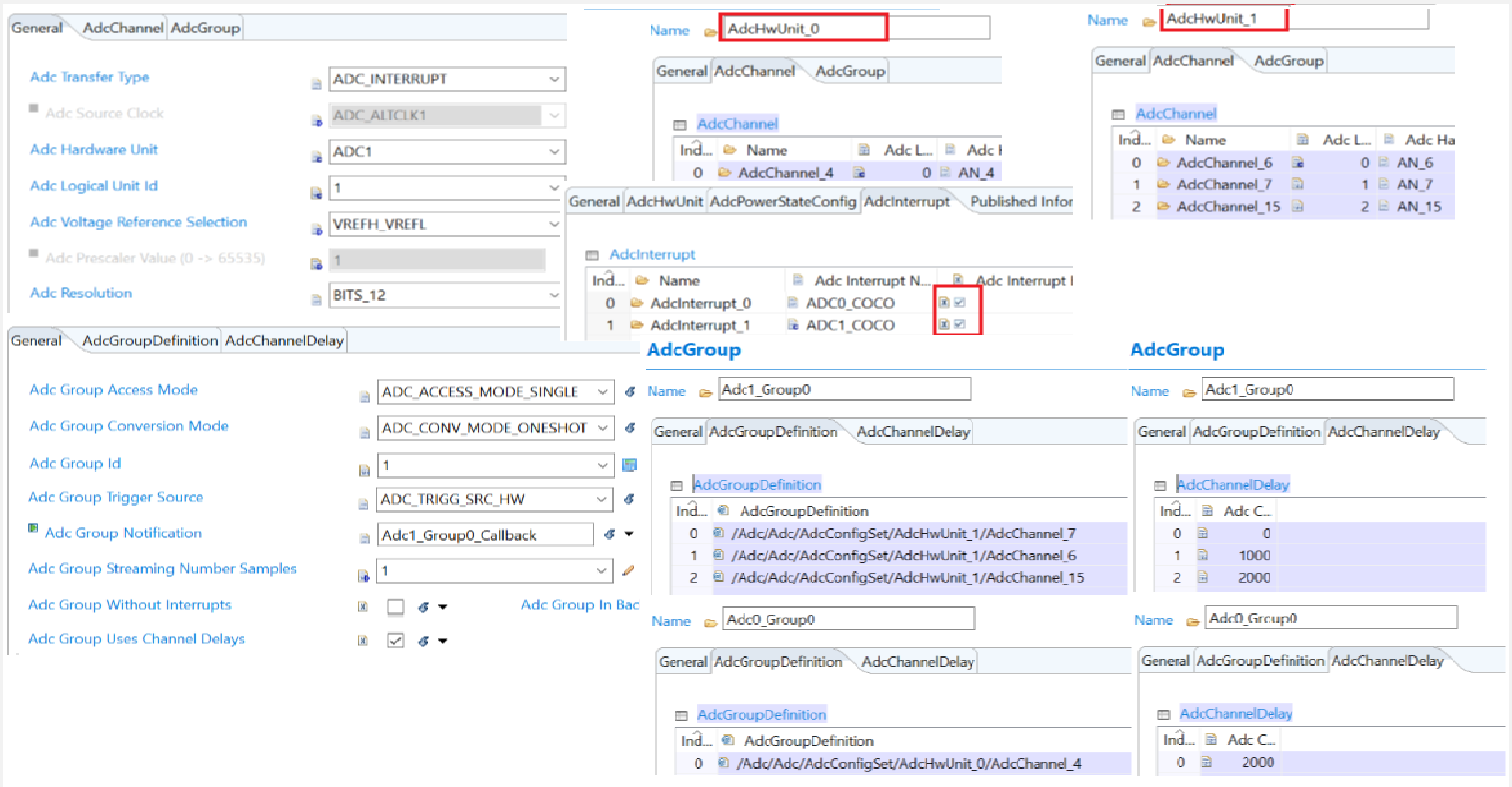
Ftm Modules | PwmFtmChannels

Ftm Hardware Module	FTM_3
Prescaler	PRESC_1
PwmPrescaler_Alternate	PRESC_1
FTM Module clock selection	PWM_SYSTEM_CLOCK
Ftm Module's Channels Alignment	PWM_CENTER_ALIGNED
End cycle reload	✓
Half cycle reload	
Reload Frequency	LDFQ_EACH32
Deadtime Counter (0 -> 1023)	32
DeadTime Counter Prescaler	PRESC_1
Pwm Background Debug Mode configuration	CNT_ON_OUTPUTS_ON
Pwm Fault Funtionality and Clear Mode	FLTCTRL_DISABLED

Modules Configuration --- ADC

- In ADC module, it not only configure the ADC but also PDB
- Enable the following channel: ADC0_4/ADC1_6/ADC1_7/ADC1_15
- Enable PDB Trigger function
- Set the PDB delay time(by manual code): 2000/1000/0/2000
- Enable ADC0_COCO&&ADC1_COCO interrupt
- Group Access Mode: Single mode
- Conversion Mode: One Shot mode
- Transfer Type: Interrupt
- Resolution: 12 Bits

Modules Configuration --- ADC



General | AdcChannel | AdcGroup

Adc Transfer Type	ADC_INTERRUPT
Adc Source Clock	ADC_ALTCLK1
Adc Hardware Unit	ADC1
Adc Logical Unit Id	1
Adc Voltage Reference Selection	VREFH_VREFL
Adc Prescaler Value (0 -> 65535)	1
Adc Resolution	BITS_12

Name: AdcHwUnit_0

General | AdcChannel | AdcGroup

AdcChannel

Ind...	Name	Adc L...	Adc I
0	AdcChannel_4	0	AN_4

Name: AdcHwUnit_1

General | AdcChannel | AdcGroup

AdcChannel

Ind...	Name	Adc L...	Adc Ha
0	AdcChannel_6	0	AN_6
1	AdcChannel_7	1	AN_7
2	AdcChannel_15	2	AN_15

General | AdcHwUnit | AdcPowerStateConfig | AdcInterrupt | Published Infor

AdcInterrupt

Ind...	Name	Adc Interrupt N...	Adc Interrupt I
0	AdcInterrupt_0	ADC0_COCO	✓
1	AdcInterrupt_1	ADC1_COCO	✓

General | AdcGroupDefinition | AdcChannelDelay

Adc Group Access Mode	ADC_ACCESS_MODE_SINGLE
Adc Group Conversion Mode	ADC_CONV_MODE_ONESHOT
Adc Group Id	1
Adc Group Trigger Source	ADC_TRIGG_SRC_HW
Adc Group Notification	Adc1_Group0_Callback
Adc Group Streaming Number Samples	1
Adc Group Without Interrupts	☐
Adc Group Uses Channel Delays	✓

Adc Group In Bac

AdcGroup

Name: Adc1_Group0

General | AdcGroupDefinition | AdcChannelDelay

AdcGroupDefinition

Ind...	AdcGroupDefinition
0	/Adc/Adc/AdcConfigSet/AdcHwUnit_1/AdcChannel_7
1	/Adc/Adc/AdcConfigSet/AdcHwUnit_1/AdcChannel_6
2	/Adc/Adc/AdcConfigSet/AdcHwUnit_1/AdcChannel_15

Name: Adc0_Group0

General | AdcGroupDefinition | AdcChannelDelay

AdcGroupDefinition

Ind...	AdcGroupDefinition
0	/Adc/Adc/AdcConfigSet/AdcHwUnit_0/AdcChannel_4

AdcGroup

Name: Adc1_Group0

General | AdcGroupDefinition | AdcChannelDelay

AdcChannelDelay

Ind...	Adc C...
0	0
1	1000
2	2000

Name: Adc0_Group0

General | AdcGroupDefinition | AdcChannelDelay

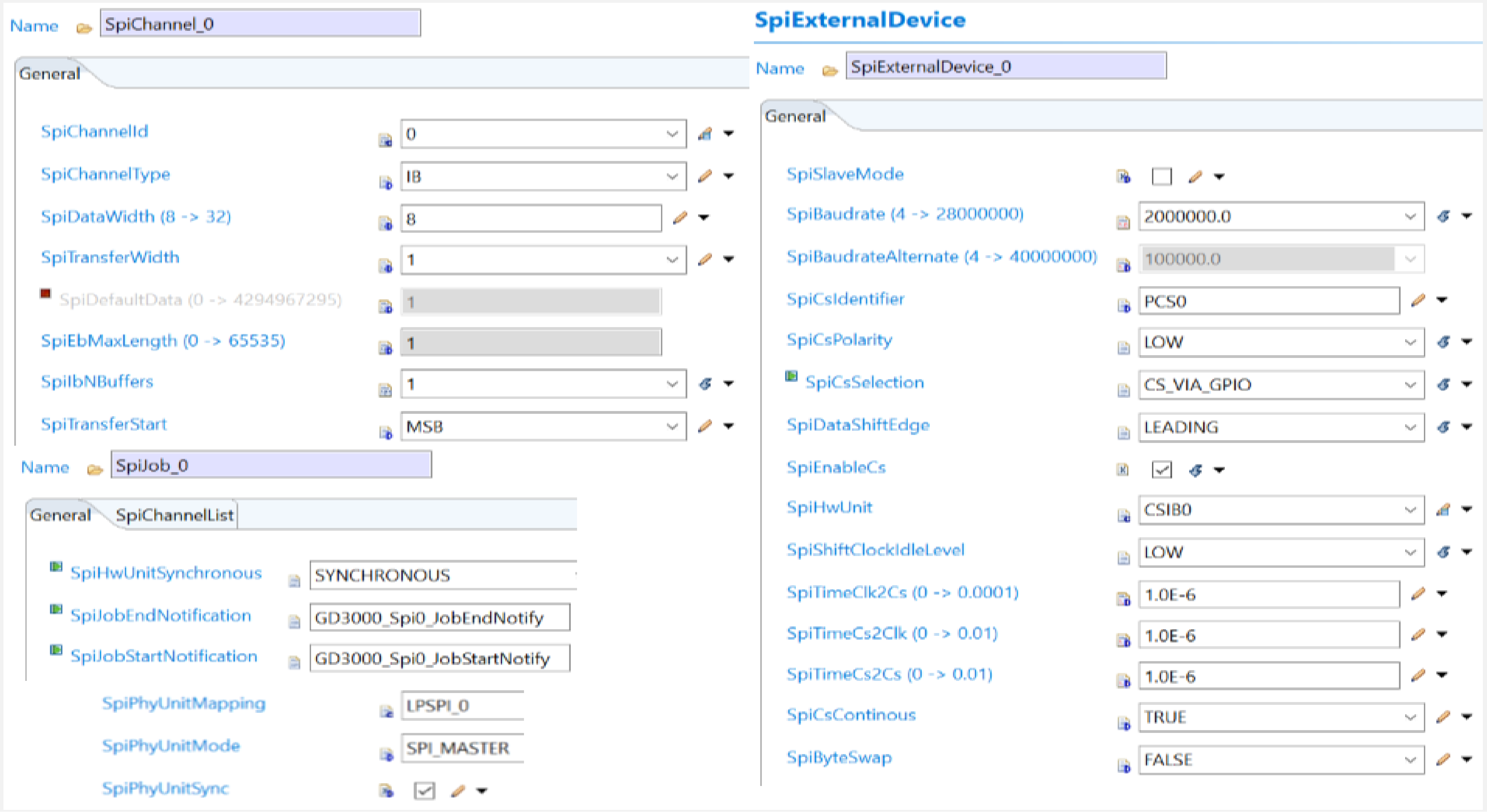
AdcChannelDelay

Ind...	Adc C...
0	2000

Modules Configuration --- SPI

- Master mode
- Speed: 2M
- Polarity: LOW
- ShiftEdge: Rise edge/Leading
- Clock Idle Level: LOW
- DataWidth: 8Bit
- TransferWidth: 1Bit
- TransferStart: MSB

Modules Configuration --- SPI



Name	SpiChannel_0

General

SpiChannelId	0
SpiChannelType	IB
SpiDataWidth (8 -> 32)	8
SpiTransferWidth	1
SpiDefaultData (0 -> 4294967295)	1
SpiEbMaxLength (0 -> 65535)	1
SpiIbNBuffers	1
SpiTransferStart	MSB

Name	SpiJob_0

General SpiChannelList

SpiHwUnitSynchronous	SYNCHRONOUS
SpiJobEndNotification	GD3000_Spi0_JobEndNotify
SpiJobStartNotification	GD3000_Spi0_JobStartNotify

SpiPhyUnitMapping	LPSPI_0
SpiPhyUnitMode	SPI_MASTER
SpiPhyUnitSync	✓

SpiExternalDevice

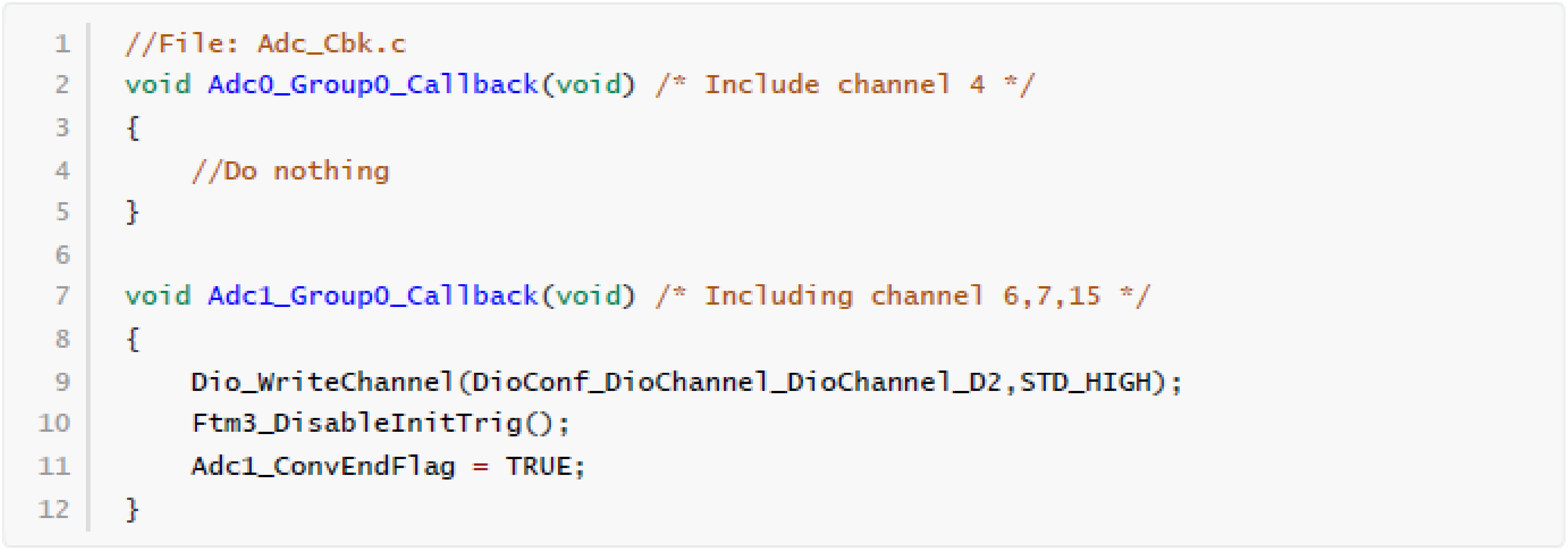
Name	SpiExternalDevice_0

General

SpiSlaveMode	☐
SpiBaudrate (4 -> 28000000)	2000000.0
SpiBaudrateAlternate (4 -> 40000000)	100000.0
SpiCsIdentifier	PCS0
SpiCsPolarity	LOW
SpiCsSelection	CS_VIA_GPIO
SpiDataShiftEdge	LEADING
SpiEnableCs	✓
SpiHwUnit	CSIB0
SpiShiftClockIdleLevel	LOW
SpiTimeClk2Cs (0 -> 0.0001)	1.0E-6
SpiTimeCs2Clk (0 -> 0.01)	1.0E-6
SpiTimeCs2Cs (0 -> 0.01)	1.0E-6
SpiCsContinous	TRUE
SpiByteSwap	FALSE

Manual Code --- ADC

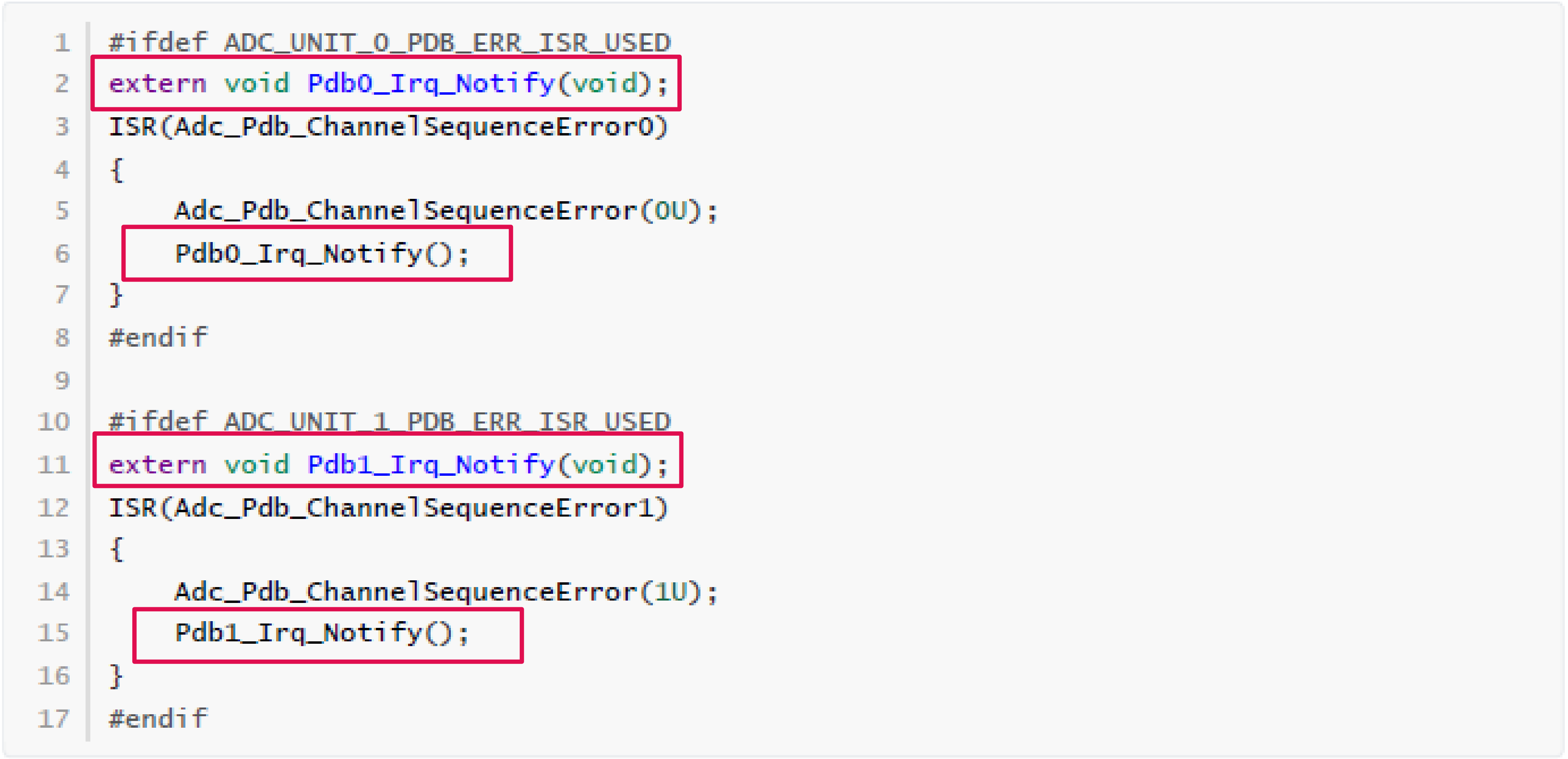
- New file: Adc_Cbk.c



```c
//File: Adc_Cbk.c
void Adc0_Group0_Callback(void) /* Include channel 4 */
{
    //Do nothing
}

void Adc1_Group0_Callback(void) /* Including channel 6,7,15 */
{
    Dio_WriteChannel(DioConf_DioChannel_DioChannel_D2,STD_HIGH);
    Ftm3_DisableInitTrig();
    Adc1_ConvEndFlag = TRUE;
}
```

Manual Code --- ADC

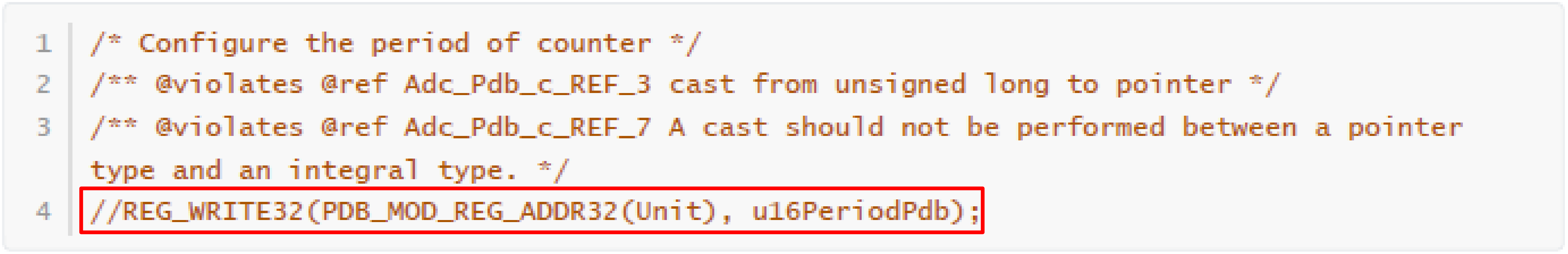
- Modify(Add the code which in red rectangle): Adc_Pdb_Irq.c



```c
#ifdef ADC_UNIT_0_PDB_ERR_ISR_USED
extern void Pdb0_Irq_Notify(void);
ISR(Adc_Pdb_ChannelSequenceError0)
{
    Adc_Pdb_ChannelSequenceError(0U);
    Pdb0_Irq_Notify();
}
#endif

#ifdef ADC_UNIT_1_PDB_ERR_ISR_USED
extern void Pdb1_Irq_Notify(void);
ISR(Adc_Pdb_ChannelSequenceError1)
{
    Adc_Pdb_ChannelSequenceError(1U);
    Pdb1_Irq_Notify();
}
#endif
```

Manual Code --- ADC

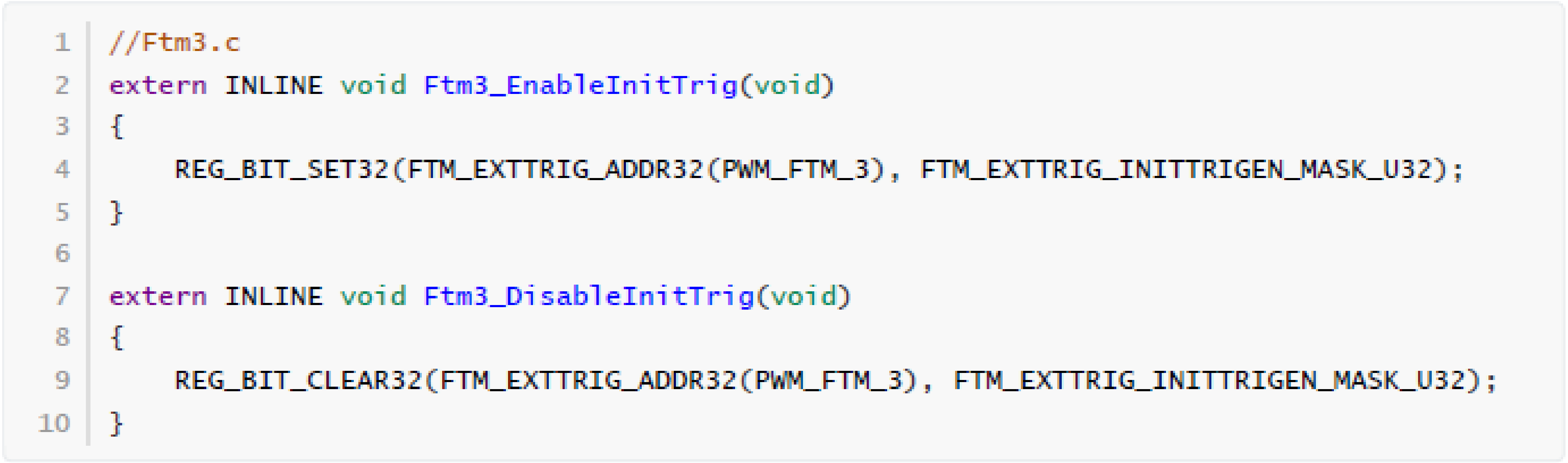
- Modify(Remove one line, line number:558): Adc_Pdb.c



```
/* Configure the period of counter */
/** @violates @ref Adc_Pdb_c_REF_3 cast from unsigned long to pointer */
/** @violates @ref Adc_Pdb_c_REF_7 A cast should not be performed between a pointer type and an integral type. */
//REG_WRITE32(PDB_MOD_REG_ADDR32(Unit), u16PeriodPdb);
```

Manual Code --- FTM

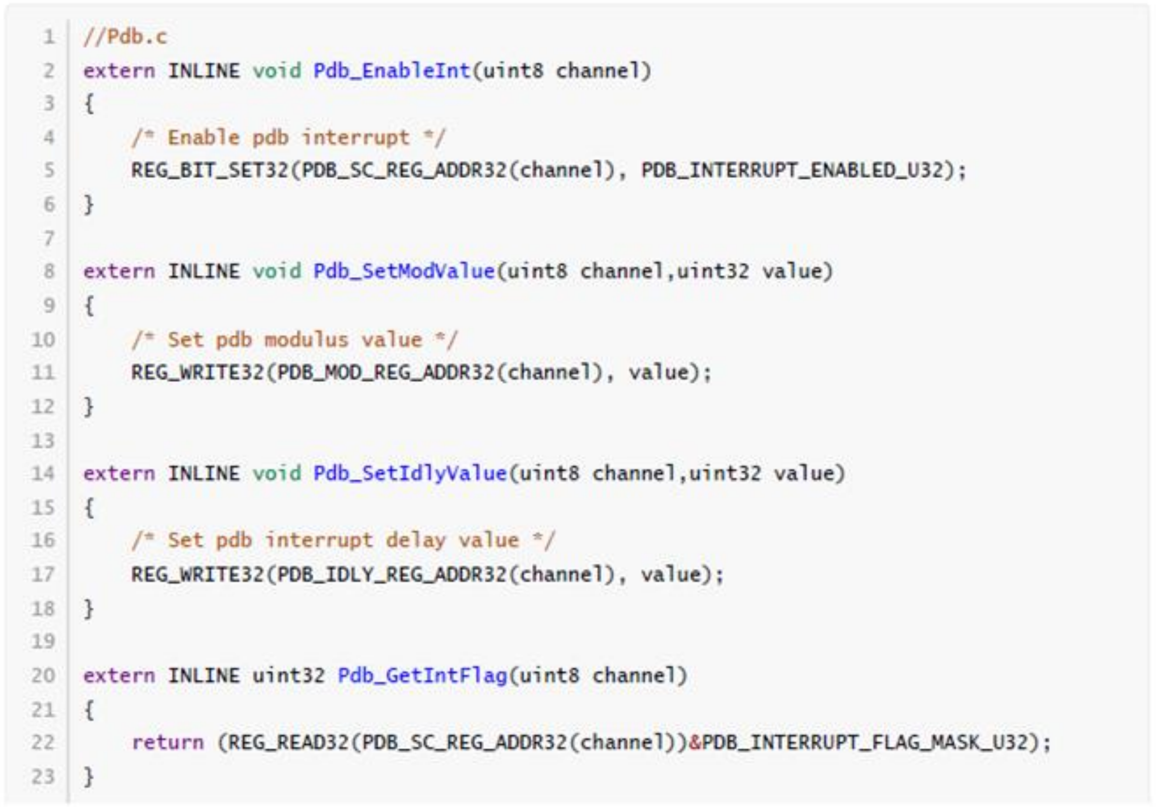
- New file: Ftm3.c



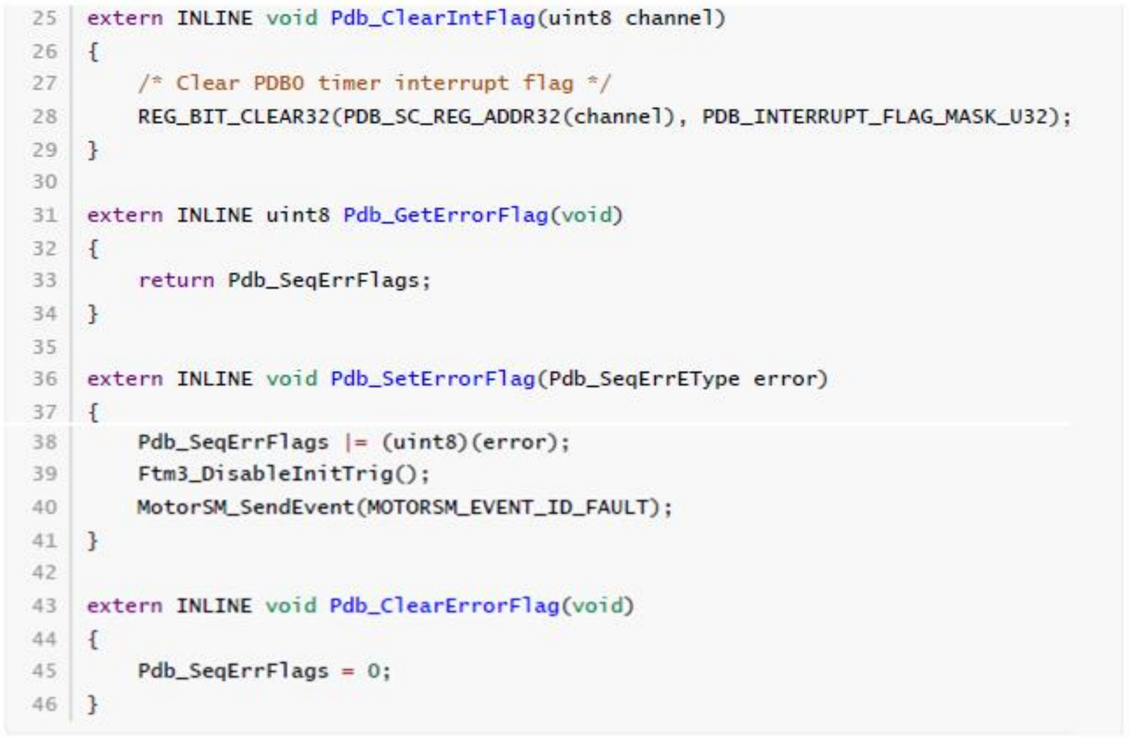
```c
//Ftm3.c
extern INLINE void Ftm3_EnableInitTrig(void)
{
    REG_BIT_SET32(FTM_EXTTRIG_ADDR32(PWM_FTM_3), FTM_EXTTRIG_INITTRIGEN_MASK_U32);
}

extern INLINE void Ftm3_DisableInitTrig(void)
{
    REG_BIT_CLEAR32(FTM_EXTTRIG_ADDR32(PWM_FTM_3), FTM_EXTTRIG_INITTRIGEN_MASK_U32);
}
```

Manual Code --- PDB

- New file: Pdb.c



```c
//Pdb.c
extern INLINE void Pdb_EnableInt(uint8 channel)
{
    /* Enable pdb interrupt */
    REG_BIT_SET32(PDB_SC_REG_ADDR32(channel), PDB_INTERRUPT_ENABLED_U32);
}

extern INLINE void Pdb_SetModValue(uint8 channel,uint32 value)
{
    /* Set pdb modulus value */
    REG_WRITE32(PDB_MOD_REG_ADDR32(channel), value);
}

extern INLINE void Pdb_SetIdlyValue(uint8 channel,uint32 value)
{
    /* Set pdb interrupt delay value */
    REG_WRITE32(PDB_IDLY_REG_ADDR32(channel), value);
}

extern INLINE uint32 Pdb_GetIntFlag(uint8 channel)
{
    return (REG_READ32(PDB_SC_REG_ADDR32(channel))&PDB_INTERRUPT_FLAG_MASK_U32);
}
```



```c
extern INLINE void Pdb_ClearIntFlag(uint8 channel)
{
    /* Clear PDB0 timer interrupt flag */
    REG_BIT_CLEAR32(PDB_SC_REG_ADDR32(channel), PDB_INTERRUPT_FLAG_MASK_U32);
}

extern INLINE uint8 Pdb_GetErrorFlag(void)
{
    return Pdb_SeqErrFlags;
}

extern INLINE void Pdb_SetErrorFlag(Pdb_SeqErrEType error)
{
    Pdb_SeqErrFlags |= (uint8)(error);
    Ftm3_DisableInitTrig();
    MotorSM_SendEvent(MOTORSM_EVENT_ID_FAULT);
}

extern INLINE void Pdb_ClearErrorFlag(void)
{
    Pdb_SeqErrFlags = 0;
}
```

Manual Code --- PDB

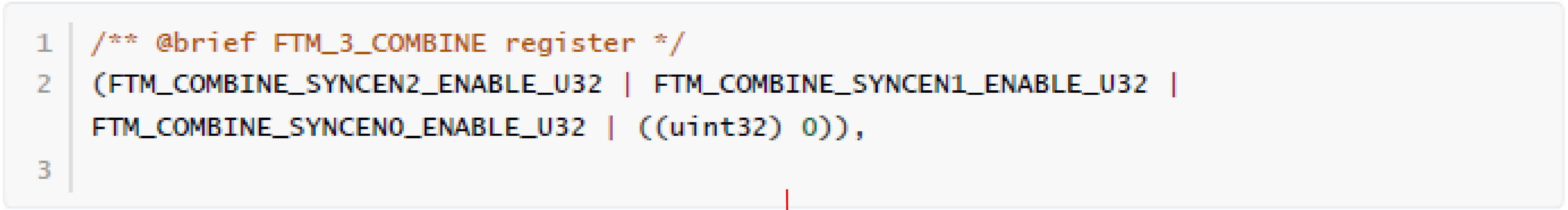
- New file: Pdb_Cbk.c

```c
//Pdb_Cbk.c
/* Pdb0_SeqErr_Notify(Sequence error notify) is before Pdb0_Irq_Notify */
extern INLINE void Pdb0_SeqErr_Notify(void) //TODO: Need a parameter to indicate which channel
{
    Pdb_SetErrorFlag(PDB_SEQ_ERR_PDB0);
}

/* Pdb1_SeqErr_Notify(Sequence error notify) is before Pdb1_Irq_Notify */
extern INLINE void Pdb1_SeqErr_Notify(void)
{
    Pdb_SetErrorFlag(PDB_SEQ_ERR_PDB1);
}

/* Pdb0_Irq_Notify is after Pdb0_SeqErr_Notify(Sequence error notify) */
extern INLINE void Pdb0_Irq_Notify(void)
{
    /* PDB0 timer overflow interrupt */
    if (0 != Pdb_GetIntFlag(0))
    {
        /* Clear PDB0 timer interrupt flag */
        Pdb_ClearIntFlag(0);
    }
}
```

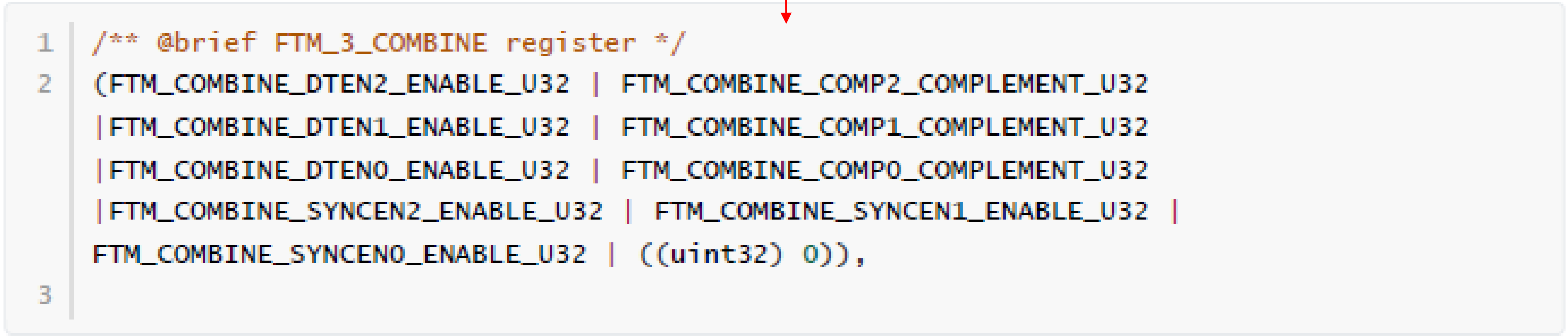
```c

/* Pdb1_Irq_Notify is after Pdb1_SeqErr_Notify(Sequence error notify) */
extern INLINE void Pdb1_Irq_Notify(void)
{
    /* PDB1 timer overflow interrupt */
    if (0 != Pdb_GetIntFlag(1))
    {
        /* Clear PDB1 timer interrupt flag */
        Pdb_ClearIntFlag(1);
        if (PDB_SEQ_ERR_NONE == Pdb_GetErrorFlag())
        {
            /* Enable FTM initialization trigger to trigger ADC modules every second PWM cycle (10kHz). */
            /* Ignore if there are PDBs sequence errors */
            Ftm3_EnableInitTrig();
        }
    }
}
```

Manual Code --- PWM

- Modify(struct:Pwm_Ftm_ModuleConfig_PB,the 6th member): Pwm_Pbcfg.c



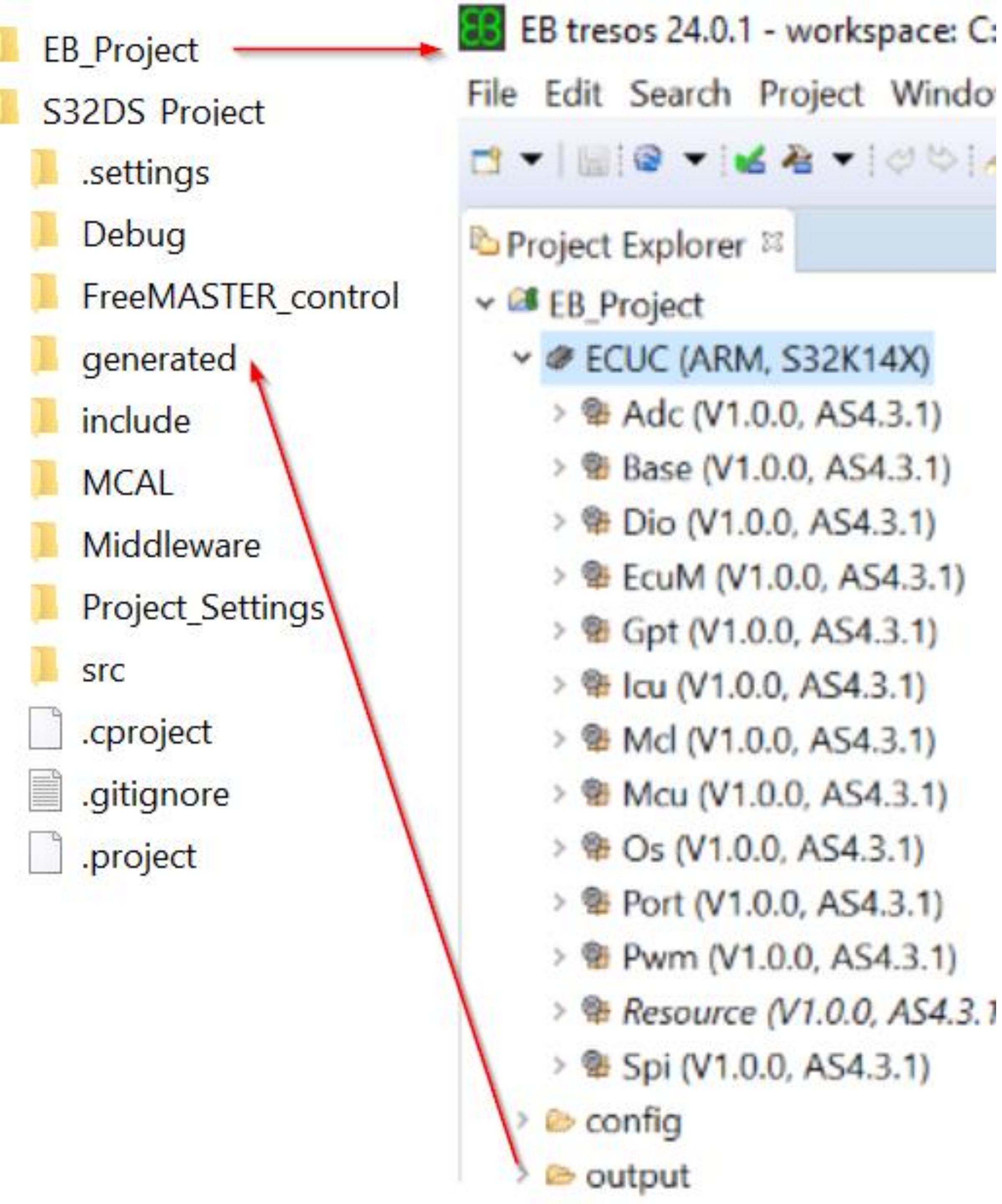
```
/** @brief FTM_3_COMBINE register */
(FTM_COMBINE_SYNCEN2_ENABLE_U32 | FTM_COMBINE_SYNCEN1_ENABLE_U32 |
FTM_COMBINE_SYNCEN0_ENABLE_U32 | ((uint32) 0)),

```



```
/** @brief FTM_3_COMBINE register */
(FTM_COMBINE_DTEN2_ENABLE_U32 | FTM_COMBINE_COMP2_COMPLEMENT_U32
|FTM_COMBINE_DTEN1_ENABLE_U32 | FTM_COMBINE_COMP1_COMPLEMENT_U32
|FTM_COMBINE_DTEN0_ENABLE_U32 | FTM_COMBINE_COMP0_COMPLEMENT_U32
|FTM_COMBINE_SYNCEN2_ENABLE_U32 | FTM_COMBINE_SYNCEN1_ENABLE_U32 |
FTM_COMBINE_SYNCEN0_ENABLE_U32 | ((uint32) 0)),

```

Manual Code --- Startup&&Link&&Middleware

- Startup && Link file: S32DS auto-generate
- Nvic: interrupt management, priority setting
- CircularQueue: queue
- FOC(Field-Oriented Control): Motor control algorithm
- GD3000: GD3000 driver
- Key: Key input event management
- Led: Led control
- Measure: DC-BUS Voltage/Current, phase current
- Motor: Motor driver, speed control
- MotorSM: Motor control main state machine
- UsrCtr: User control information setting
- Wait: Software delay


EB_Project
S32DS Project
.settings
Debug
FreeMASTER_control
generated
include
MCAL
Middleware
Project_Settings
src
.cproject
.gitignore
.project

EB tresos 24.0.1 - workspace: C:

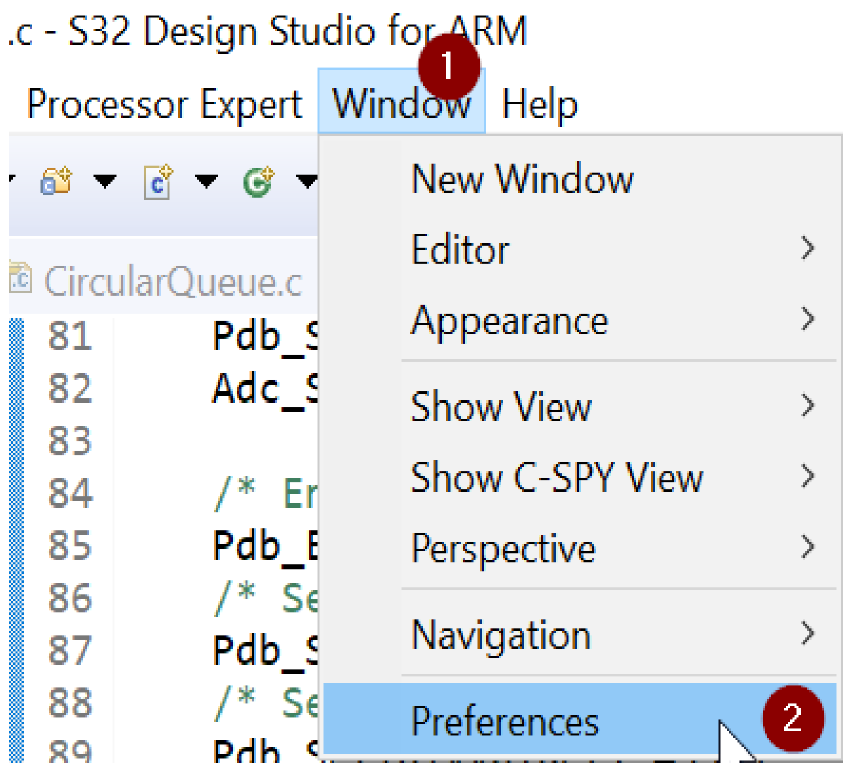
EB_Project
ECUC (ARM, S32K14X)
Adc (V1.0.0, AS4.3.1)
Base (V1.0.0, AS4.3.1)
Dio (V1.0.0, AS4.3.1)
EcuM (V1.0.0, AS4.3.1)
Gpt (V1.0.0, AS4.3.1)
Icu (V1.0.0, AS4.3.1)
Mcl (V1.0.0, AS4.3.1)
Mcu (V1.0.0, AS4.3.1)
Os (V1.0.0, AS4.3.1)
Port (V1.0.0, AS4.3.1)
Pwm (V1.0.0, AS4.3.1)
Resource (V1.0.0, AS4.3.1
Spi (V1.0.0, AS4.3.1)
config
output

Code Integration --- Overview

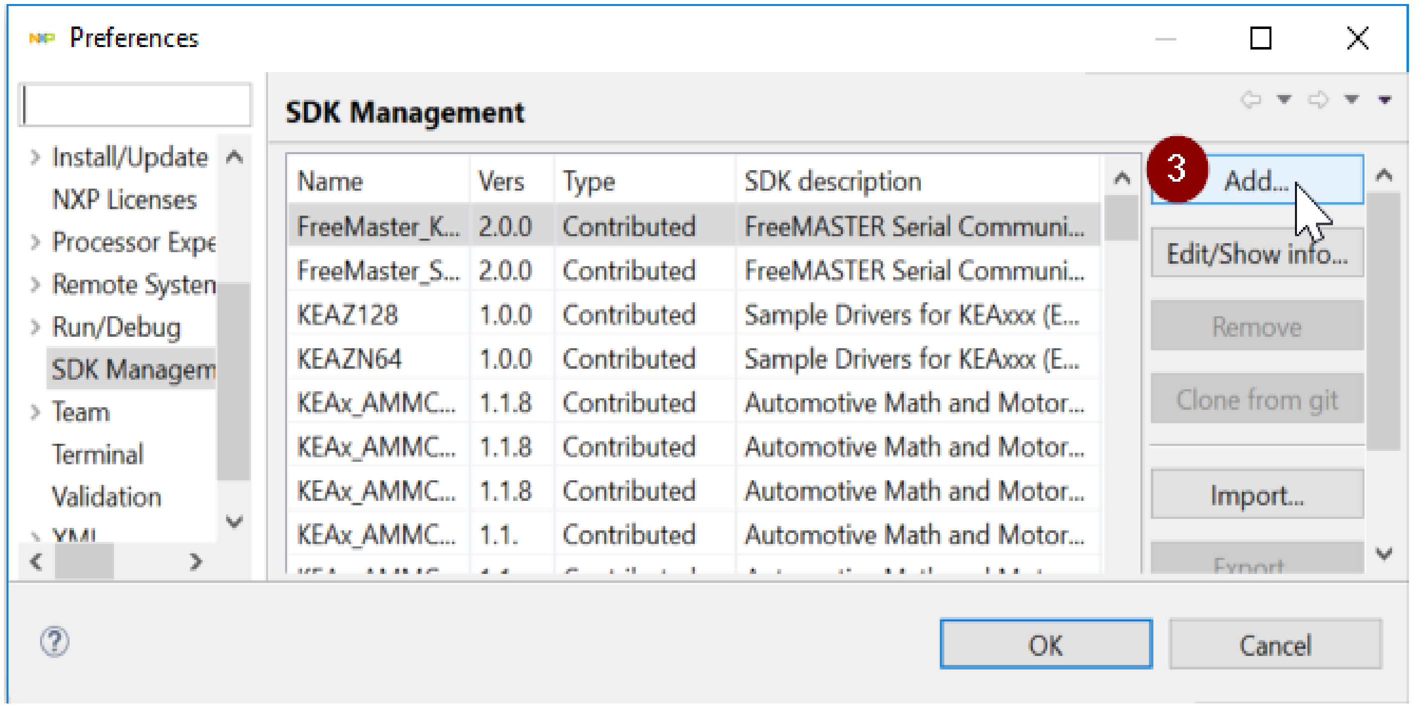
- The main modules: Adc Pwm Mcl
- The “generated” folder is auto generated by EB_Project
- The “MCAL” folder is the static code of modules

Code Integration --- Packing MCAL to SDK

- S32DS: Window -> Preferences -> SDK Management -> Add

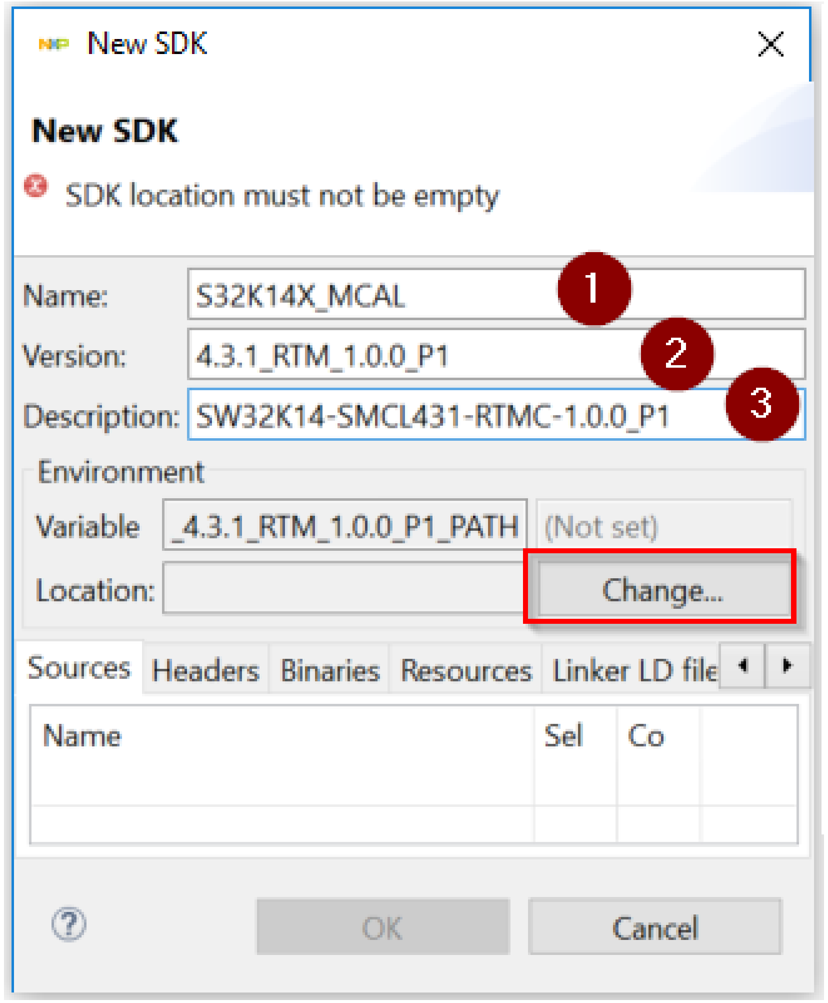


```
Pdb_S
Adc_S

/* Er
Pdb_E
/* Se
Pdb_S
/* Se
Pdb_S
```



Name	Vers	Type	SDK description
FreeMaster_K...	2.0.0	Contributed	FreeMASTER Serial Communi...
FreeMaster_S...	2.0.0	Contributed	FreeMASTER Serial Communi...
KEAZ128	1.0.0	Contributed	Sample Drivers for KEAxxx (E...
KEAZN64	1.0.0	Contributed	Sample Drivers for KEAxxx (E...
KEAx_AMMC...	1.1.8	Contributed	Automotive Math and Motor...
KEAx_AMMC...	1.1.8	Contributed	Automotive Math and Motor...
KEAx_AMMC...	1.1.8	Contributed	Automotive Math and Motor...
KEAx_AMMC...	1.1.	Contributed	Automotive Math and Motor...

Code Integration --- Packing MCAL to SDK



New SDK

New SDK

SDK location must not be empty

Name: S32K14X_MCAL

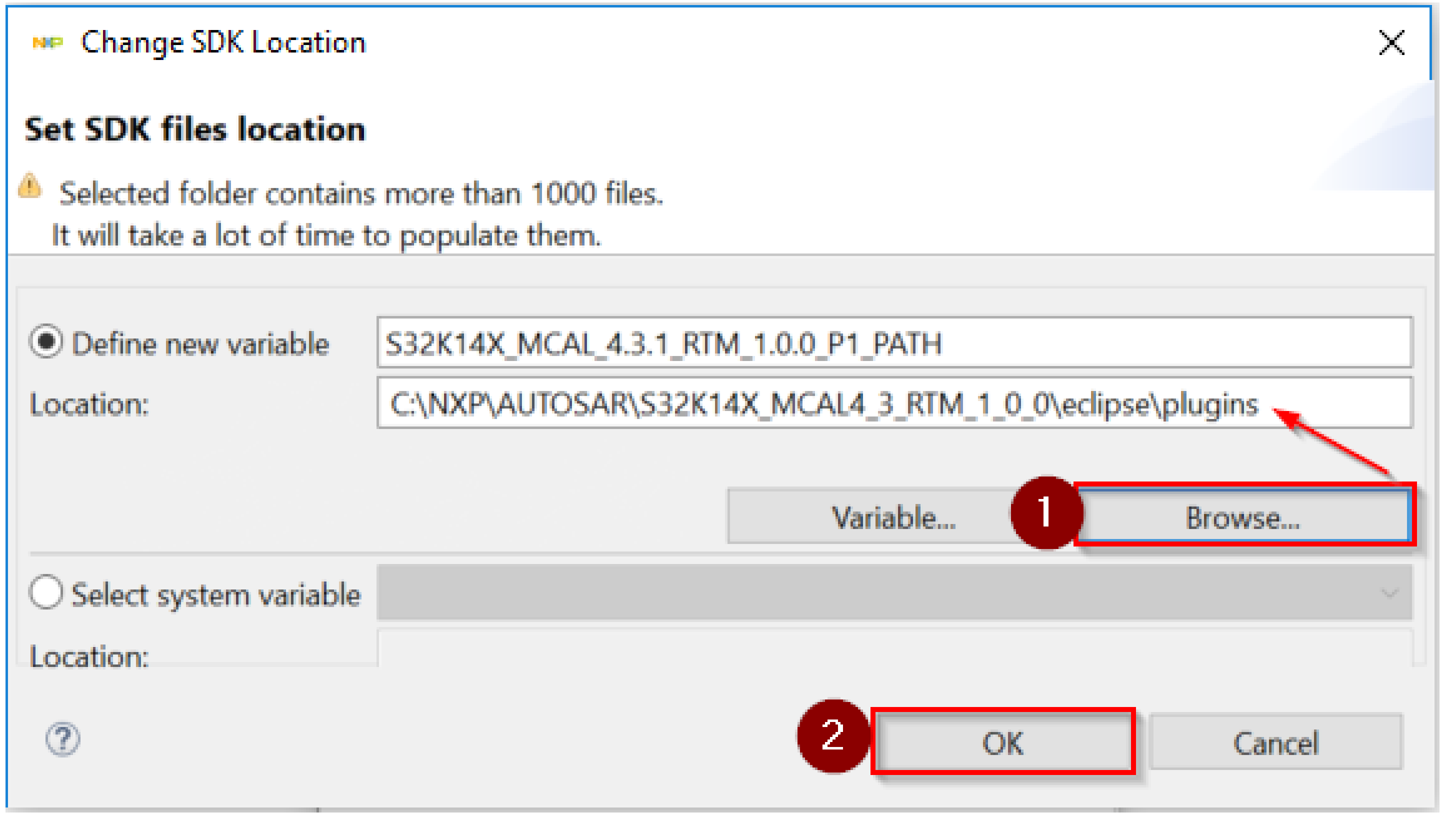
Version: 4.3.1_RTM_1.0.0_P1

Description: SW32K14-SMCL431-RTMC-1.0.0_P1

Environment

Variable _4.3.1_RTM_1.0.0_P1_PATH (Not set)

Location:



Change SDK Location

Set SDK files location

Selected folder contains more than 1000 files.
It will take a lot of time to populate them.

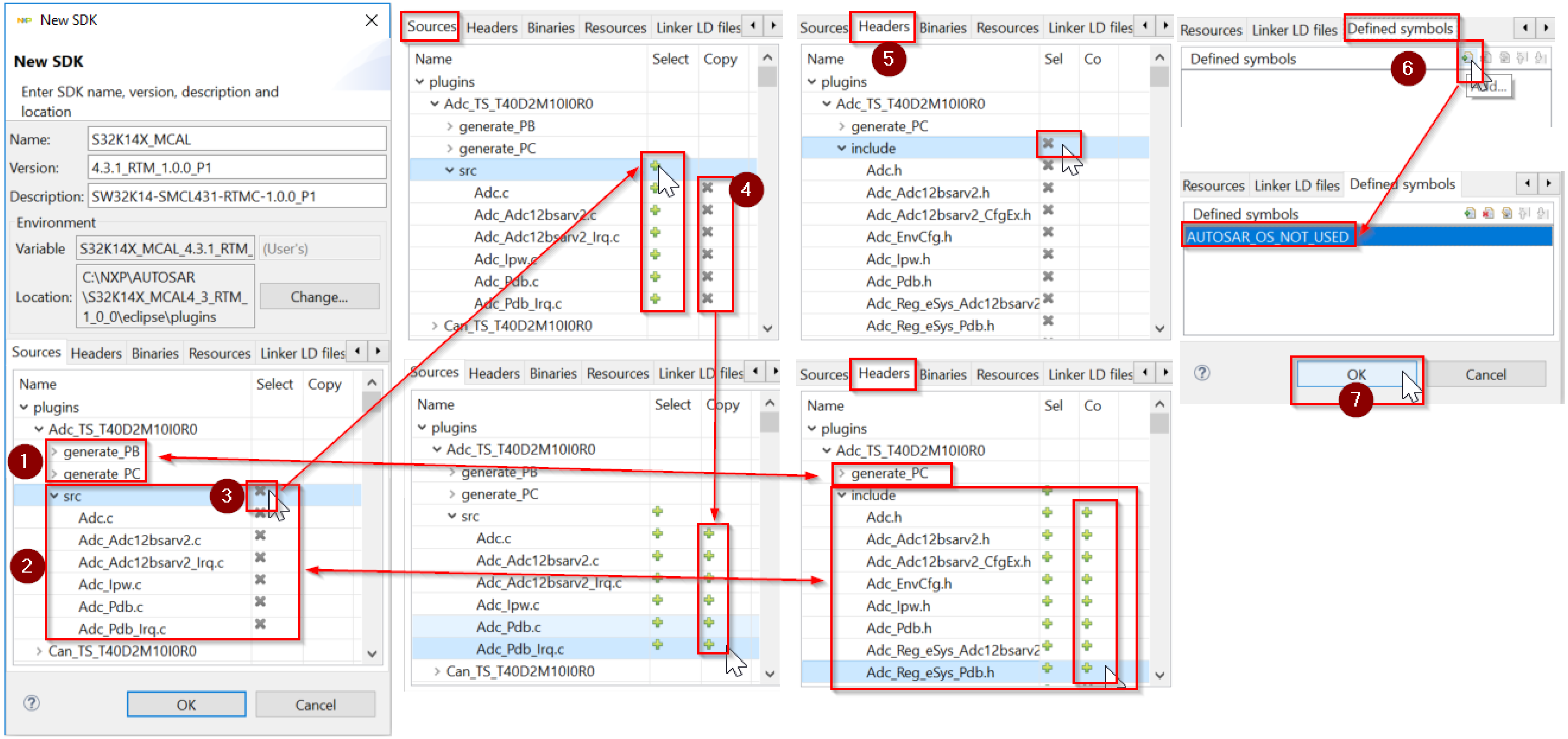
Define new variable: S32K14X_MCAL_4.3.1_RTM_1.0.0_P1_PATH

Location: C:\NXP\AUTOSAR\S32K14X_MCAL4_3_RTM_1_0_0\eclipse\plugins

Select system variable

Location:

Code Integration --- Packing MCAL to SDK



New SDK

New SDK

Enter SDK name, version, description and location

Name: S32K14X_MCAL

Version: 4.3.1_RTM_1.0.0_P1

Description: SW32K14-SMCL431-RTMC-1.0.0_P1

Environment

Variable: S32K14X_MCAL_4.3.1_RTM_ (User's)

Location: C:\NXP\AUTOSAR\S32K14X_MCAL4_3_RTM_1_0_0\eclipse\plugins

Name	Select	Copy
plugins		
Adc_TS_T40D2M10I0R0		
generate_PB		
generate_PC		
src	✖	
Adc.c	✖	
Adc_Adc12bsarv2.c	✖	
Adc_Adc12bsarv2_Irq.c	✖	
Adc_Ipw.c	✖	
Adc_Pdb.c	✖	
Adc_Pdb_Irq.c	✖	
Can_TS_T40D2M10I0R0		

Name	Select	Copy
plugins		
Adc_TS_T40D2M10I0R0		
generate_PB		
generate_PC		
src	✚	
Adc.c	✚	✖
Adc_Adc12bsarv2.c	✚	✖
Adc_Adc12bsarv2_Irq.c	✚	✖
Adc_Ipw.c	✚	✖
Adc_Pdb.c	✚	✖
Adc_Pdb_Irq.c	✚	✖
Can_TS_T40D2M10I0R0		

Name	Select	Copy
plugins		
Adc_TS_T40D2M10I0R0		
generate_PB		
generate_PC		
src		
Adc.c	✚	✚
Adc_Adc12bsarv2.c	✚	✚
Adc_Adc12bsarv2_Irq.c	✚	✚
Adc_Ipw.c	✚	✚
Adc_Pdb.c	✚	✚
Adc_Pdb_Irq.c	✚	✚
Can_TS_T40D2M10I0R0		

Name	Sel	Co
plugins		
Adc_TS_T40D2M10I0R0		
generate_PC		
include	✖	
Adc.h	✖	
Adc_Adc12bsarv2.h	✖	
Adc_Adc12bsarv2_CfgEx.h	✖	
Adc_EnvCfg.h	✖	
Adc_Ipw.h	✖	
Adc_Pdb.h	✖	
Adc_Reg_eSys_Adc12bsarv2	✖	
Adc_Reg_eSys_Pdb.h	✖	

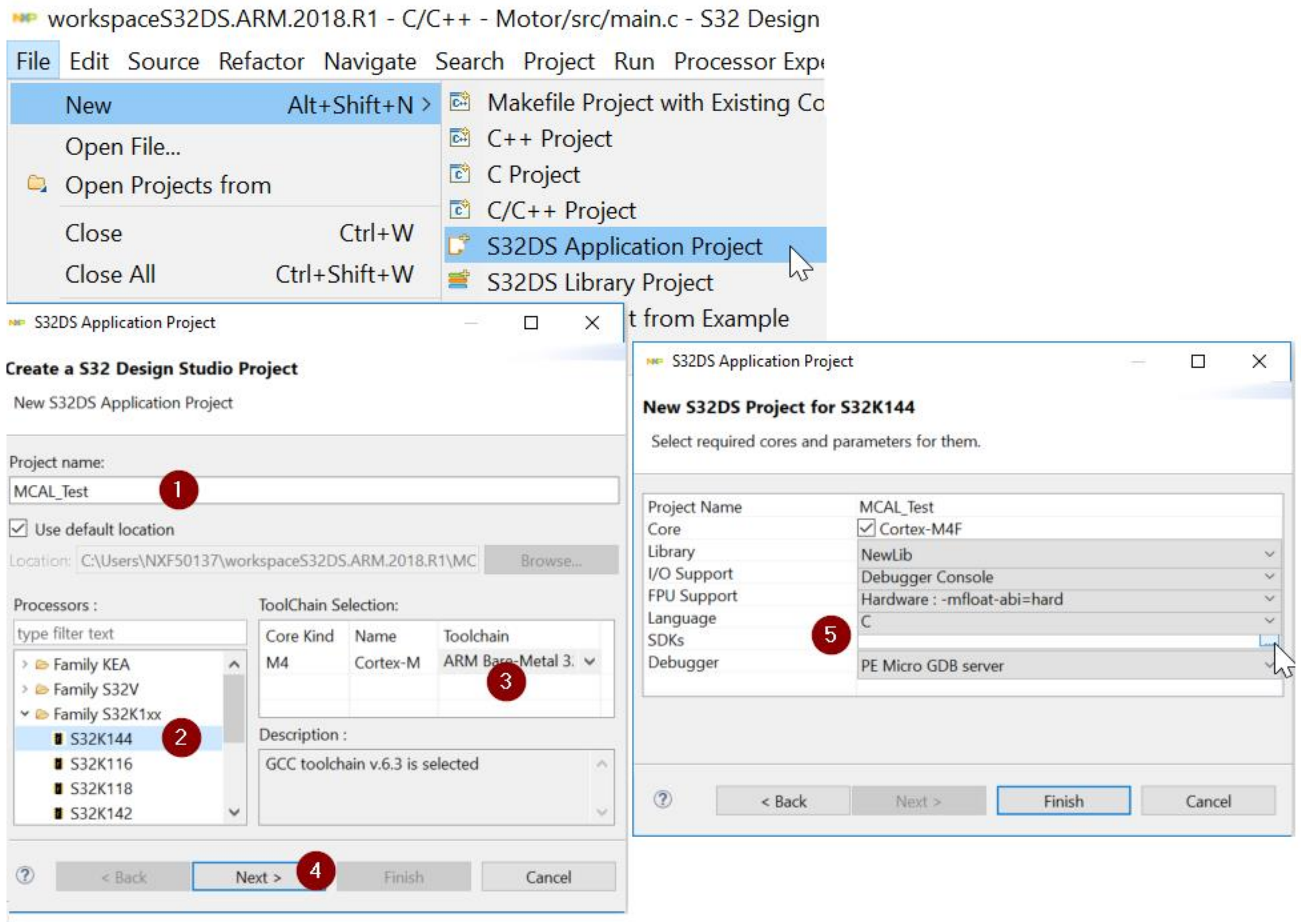
Name	Sel	Co
plugins		
Adc_TS_T40D2M10I0R0		
generate_PC		
include	✚	
Adc.h	✚	✚
Adc_Adc12bsarv2.h	✚	✚
Adc_Adc12bsarv2_CfgEx.h	✚	✚
Adc_EnvCfg.h	✚	✚
Adc_Ipw.h	✚	✚
Adc_Pdb.h	✚	✚
Adc_Reg_eSys_Adc12bsarv2	✚	✚
Adc_Reg_eSys_Pdb.h	✚	✚

Defined symbols

AUTOSAR_OS_NOT_USED

Code Integration --- New Project(S32DS)

- Select MCAL in SDKs items



S32DS Application Project

Create a S32 Design Studio Project

New S32DS Application Project

Project name: MCAL_Test

☑ Use default location

Location: C:\Users\NXF50137\workspaceS32DS.ARM.2018.R1\MC

Processors :

type filter text

- Family KEA
- Family S32V
- Family S32K1xx
  - S32K144
  - S32K116
  - S32K118
  - S32K142

ToolChain Selection:

Core Kind	Name	Toolchain
M4	Cortex-M	ARM Bare-Metal 3.

Description : GCC toolchain v.6.3 is selected

S32DS Application Project

New S32DS Project for S32K144

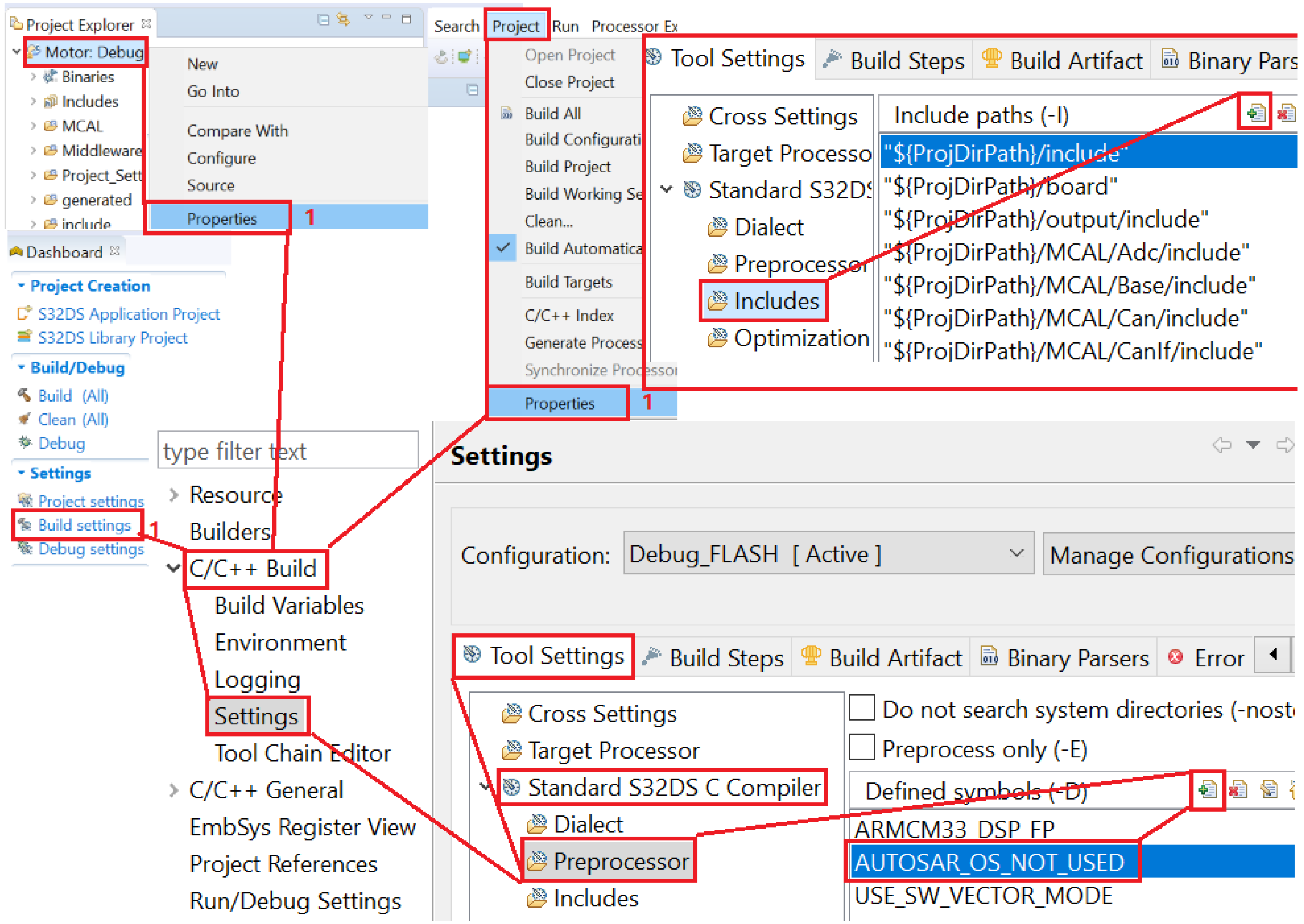
Select required cores and parameters for them.

Project Name	MCAL_Test
Core	☑ Cortex-M4F
Library	NewLib
I/O Support	Debugger Console
FPU Support	Hardware : -mfloat-abi=hard
Language	C
SDKs	
Debugger	PE Micro GDB server

Code Integration --- Project Properties

- Header file include and Macro define
- M1: Select Project and right click -> Properties
- M2: Menu -> Project -> Properties
- M3: Dashboard -> Project settings/Build settings
- Then, C/C++ Build -> Settings -> Tool Settings -> Standard S32DS C Compiler -> Preprocessor/Includes

Code Integration --- Project Properties



Include paths (-I)

```
"${ProjDirPath}/include"
"${ProjDirPath}/board"
"${ProjDirPath}/output/include"
"${ProjDirPath}/MCAL/Adc/include"
"${ProjDirPath}/MCAL/Base/include"
"${ProjDirPath}/MCAL/Can/include"
"${ProjDirPath}/MCAL/CanIf/include"
```

Configuration: Debug_FLASH [Active]

Defined symbols (-D)

```
ARMCM33_DSP_FP
AUTOSAR_OS_NOT_USED
USE_SW_VECTOR_MODE
```

Code Integration --- Compile Resource Configurations

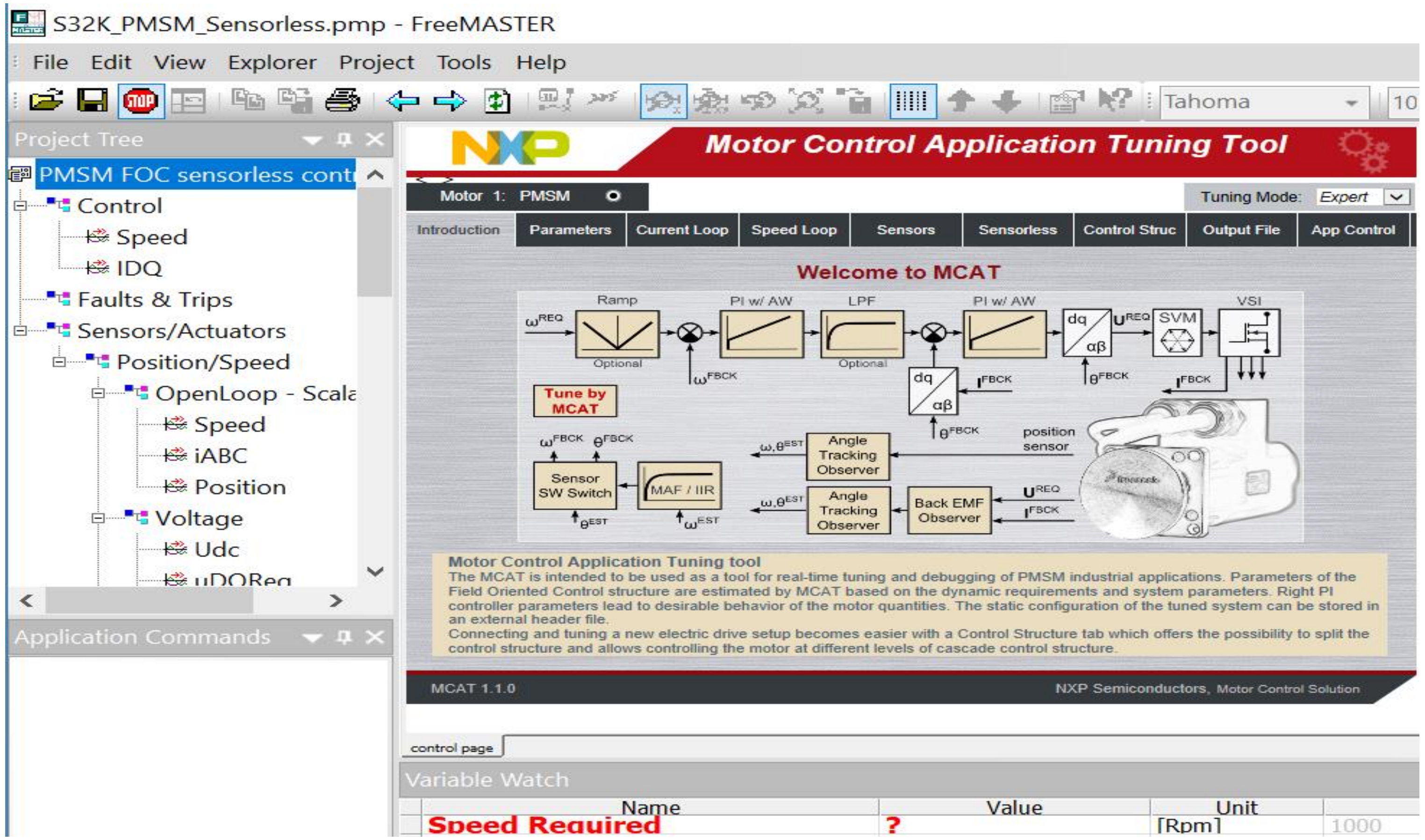
- Add/Remove source file(*.c) to/from compile target
- M1(Add/Remove): Select Project and right click -> Build Configurations Explorer
- M2(Add/Remove): Select File/Folder and right click -> build path -> Add to/Remove from -> Debug/Release/Debug_RAM
- M3(Remove): Select File/Folder and right click -> Resource Configurations -> Exclude from Build

Debug/Tuning --- Run

- Enable 12V Power Plugin, Led in yellow(default)
- Press SW2/SW3, Led flash in yellow(calibration) and then turn to green(align)
- Red color means error happened, press SW2 and SW3 at the same time to clear error
- In running mode, press SW2/SW3 to speed up/speed down
- In running mode, press SW2 and SW3 at the same time to stop

Debug/Tuning --- FreeMASTER/MCAT

- Speed control, show real speed/current dynamicly and many other useful function



Motor Control Application Tuning tool

The MCAT is intended to be used as a tool for real-time tuning and debugging of PMSM industrial applications. Parameters of the Field Oriented Control structure are estimated by MCAT based on the dynamic requirements and system parameters. Right PI controller parameters lead to desirable behavior of the motor quantities. The static configuration of the tuned system can be stored in an external header file.
Connecting and tuning a new electric drive setup becomes easier with a Control Structure tab which offers the possibility to split the control structure and allows controlling the motor at different levels of cascade control structure.

MCAT 1.1.0 NXP Semiconductors, Motor Control Solution

Name	Value	Unit	
Speed Required	?	[Rpm]	1000

Appendix

- NXP_AUTOSAR_MCAL开发环境搭建引导_S32K14x系列.pdf
- NXP_MCAL结构概览_S32K1系列.pdf
- EB_Tresos入门指南.pdf
- MCAL配置指导_电机控制双电阻采样.pdf
- S32DS创建自己的SDK.pdf
- 使用S32DS集成MCAL.pdf

SECURE CONNECTIONS
FOR A SMARTER WORLD

www.nxp.com