# S32 DESIGN STUDIO TOOLS

PATRICK W. MENTER

ENGINEERING MANAGER TOOLS ENABLEMENT
AUTOMOTIVE MICROCONTROLLERS & PROCESSORS

AMF-AUT-T2814 | AUGUST 2017

NXP | SECURE CONNECTIONS
FOR A SMARTER WORLD

# AGENDA

- What is the S32 Design Studio
- Current Tools for ARM Based Processors
- Current Tools for e200 Based Processors
- New Tools for Vision Application Development
- In Summary

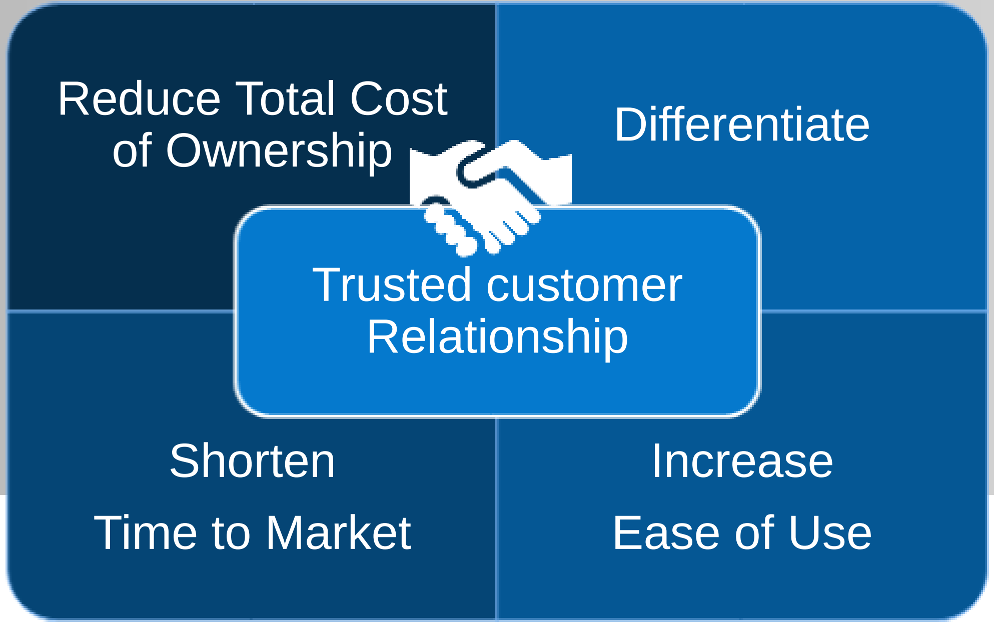
# 01.

# What is the S32 Design Studio

Introduces the concept of the S32 Design Studio with some background for these tools.

# Ease of use with Tools & Software a driving priority

- Tools & SW are the **#1 decision criteria** for our customers - **enables** our customers to use our silicon, defines our customer’s **user experience**, drives up the **value & reuse** of our products.
- Customer needs increase significantly with product complexity/performance. **T1/OEMs expect Tools & SW** improvement from IC suppliers as it is fundamental to their R&D productivity (3 SW engineers for every HW Engineer)
- SW is where our customers **spend the most time in development**
- We are a **solution provider** (SW and Si)



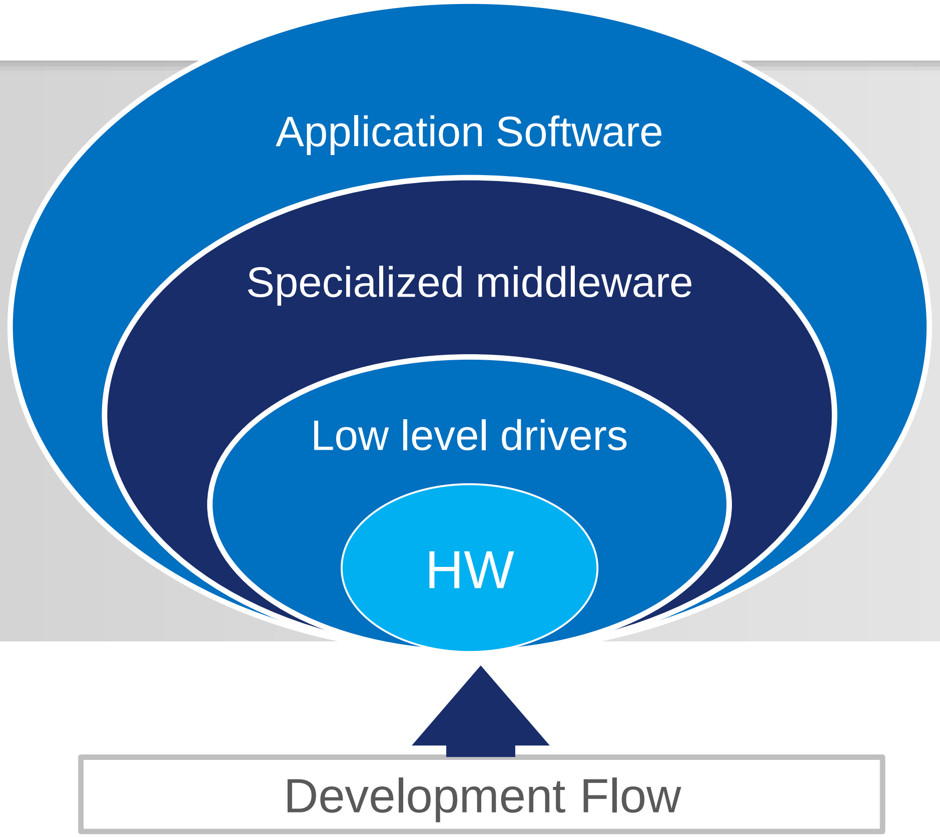
Reduce Total Cost of Ownership

Differentiate

Trusted customer Relationship

Shorten Time to Market

Increase Ease of Use



Application Software

Specialized middleware

Low level drivers

HW

Development Flow

# Common Enablement across products

Consistent Software and Tools offering

## Eco-System

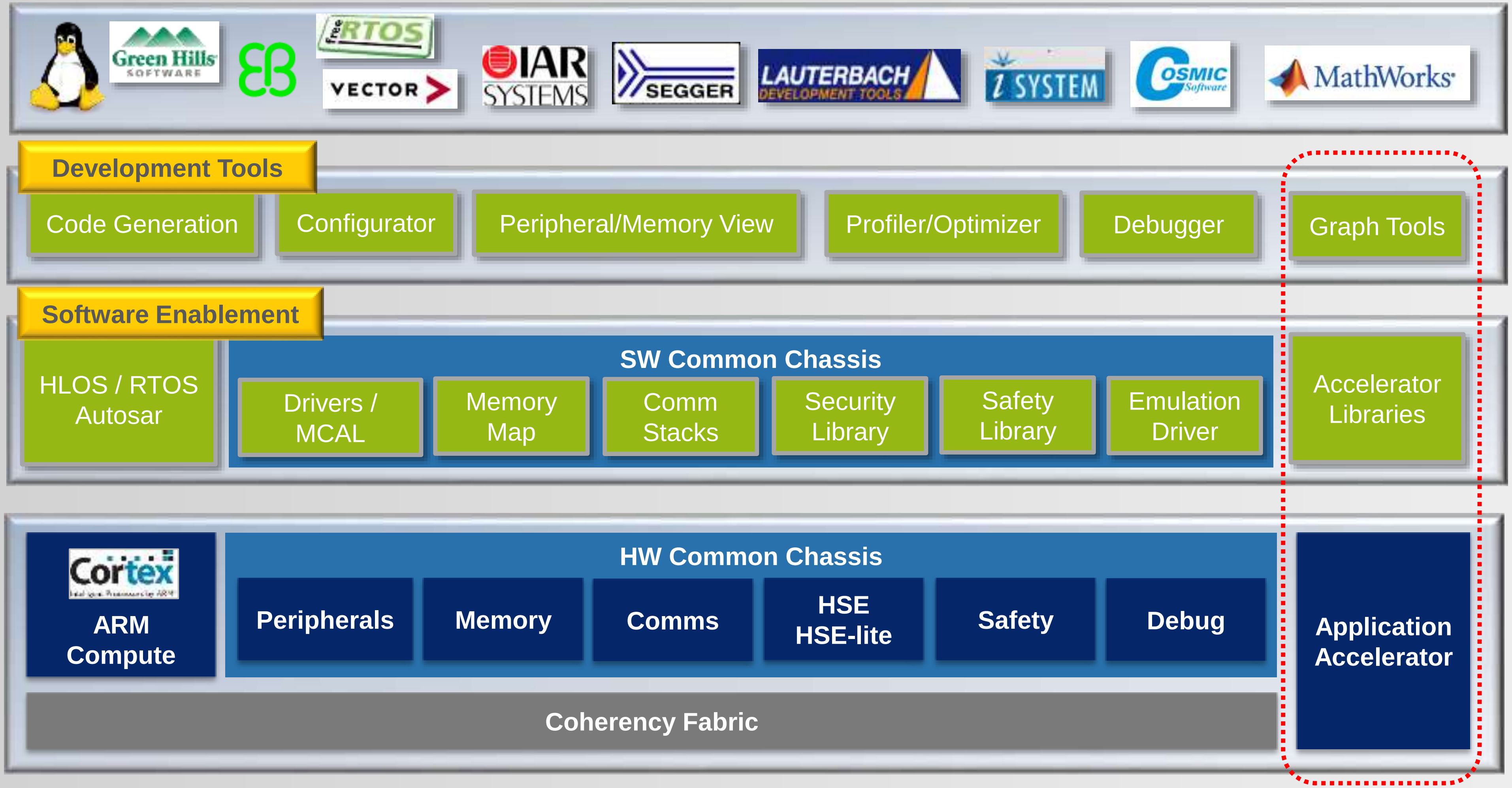
Tools & Software
Safety & Security

## Development Tools

Supports all targets
Consistent look and feel

## Software Enablement

Consistent APIs
Production Quality



Development Tools

Code Generation | Configurator | Peripheral/Memory View | Profiler/Optimizer | Debugger | Graph Tools

Software Enablement

HLOS / RTOS Autosar

SW Common Chassis

Drivers / MCAL | Memory Map | Comm Stacks | Security Library | Safety Library | Emulation Driver

Accelerator Libraries

Cortex
ARM Compute

HW Common Chassis

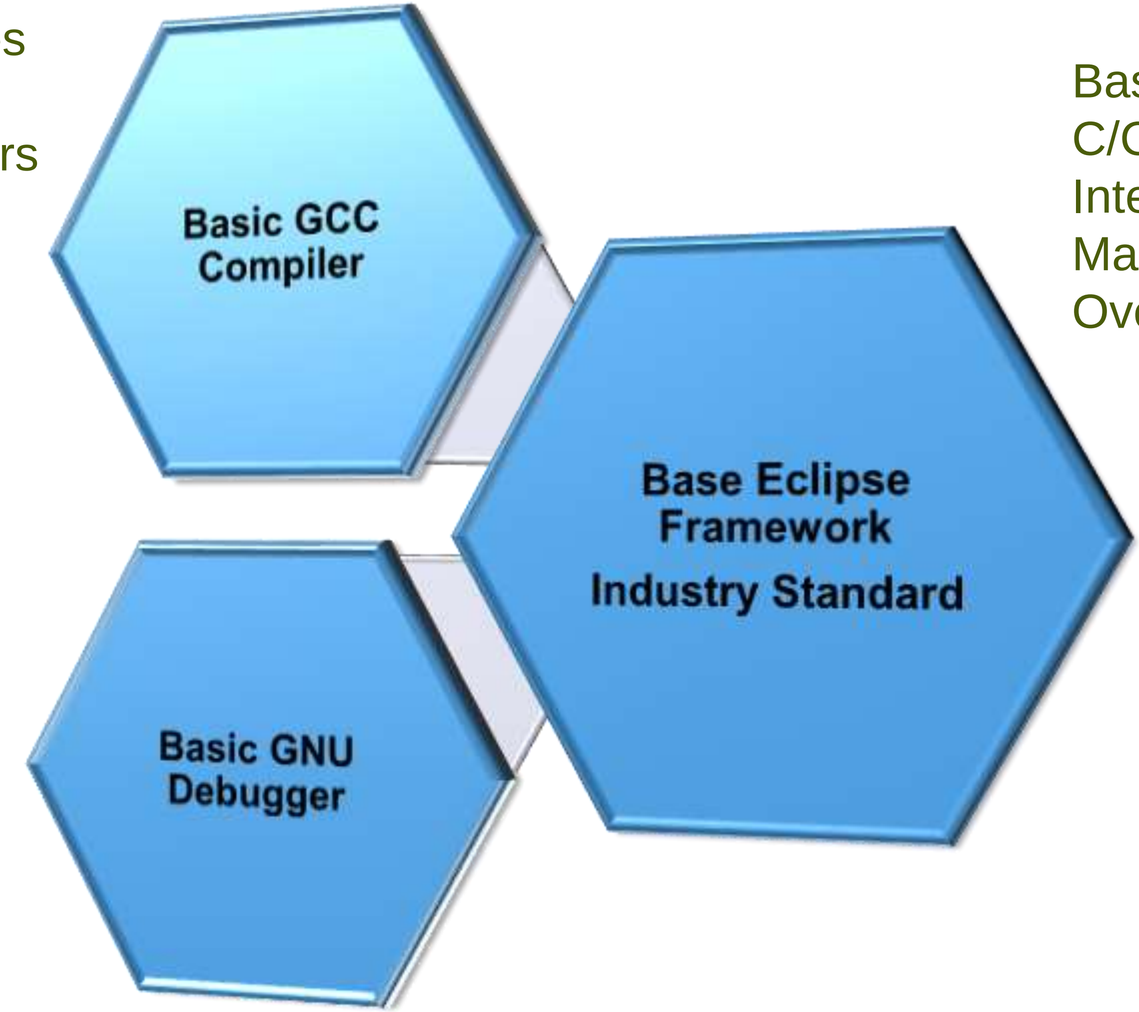
Peripherals | Memory | Comms | HSE HSE-lite | Safety | Debug

Application Accelerator

Coherency Fabric

# S32 Design Studio – Basic Tool Frame Work is Eclipse Based

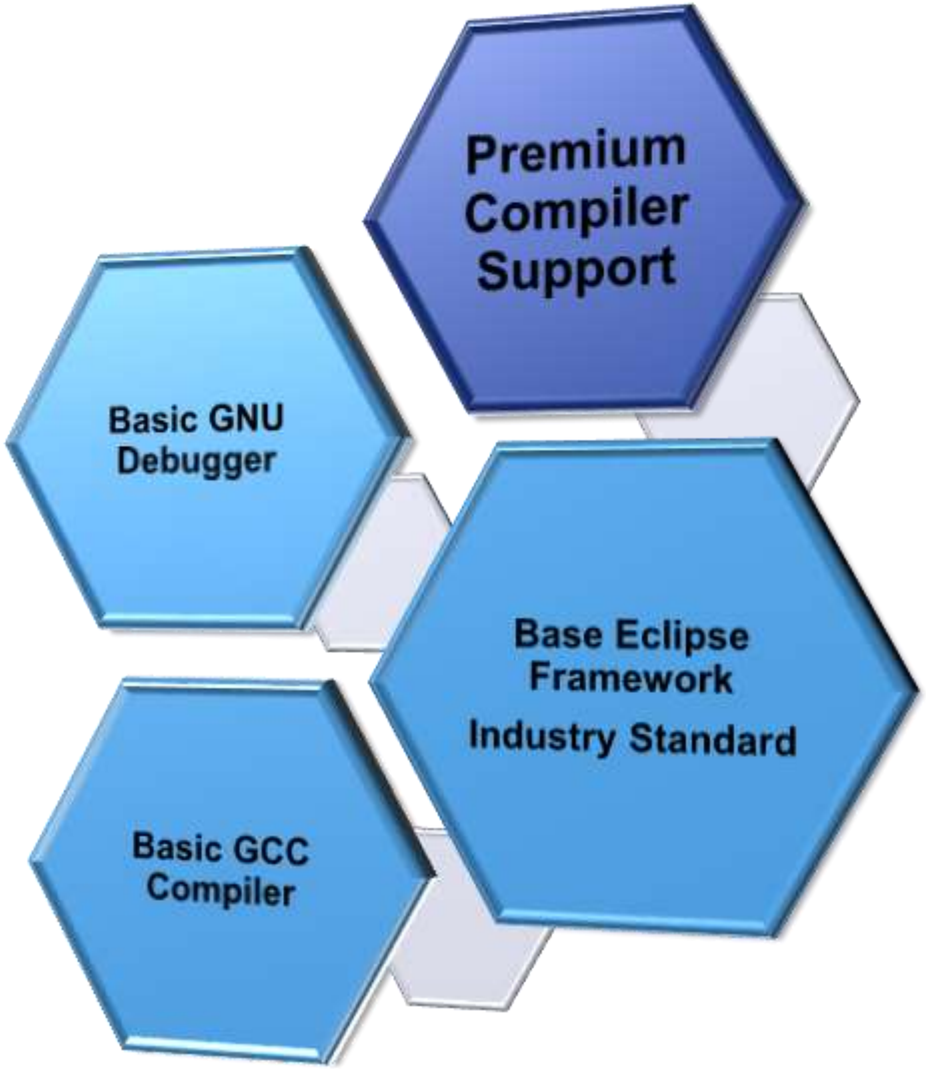
GNU C Compiler
GCC from ARM M and A cores
GCC for Power Architecture
Solid Standard GCC Compilers



Basic Eclipse Backplane
C/C++ Development Tools
Integrated Editor with C/C++ tools
Managed Make Facility
Over 10,000 Eclipse Plug-in Available

GNU Debugger
Basic Debugger interface
to low cost JTAG debugger
(P&E, and Segger)

# S32 Design Studio – Premium Compiler Support

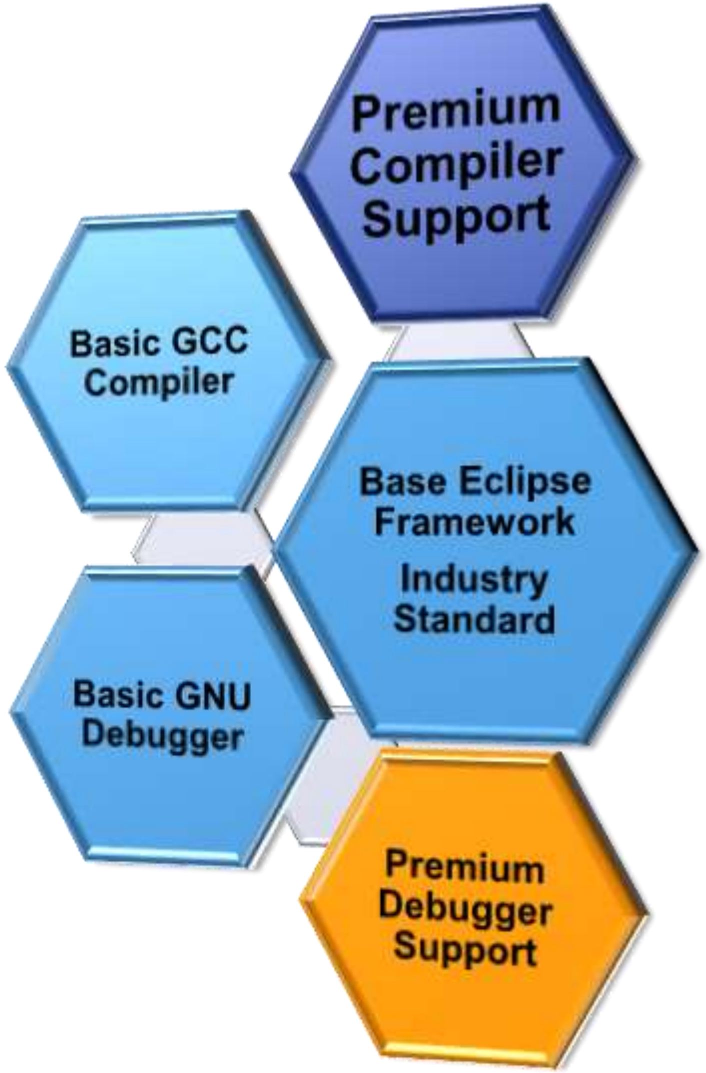


3rd Party Premium Compiler Support
ISO Certified compilers to support

Examples: GHS and IAR both have ISO26262 certifications and Certification Kits

Best in Class Compilers for Code Density and Code Performance

# S32 Design Studio – Premium Debugger Support



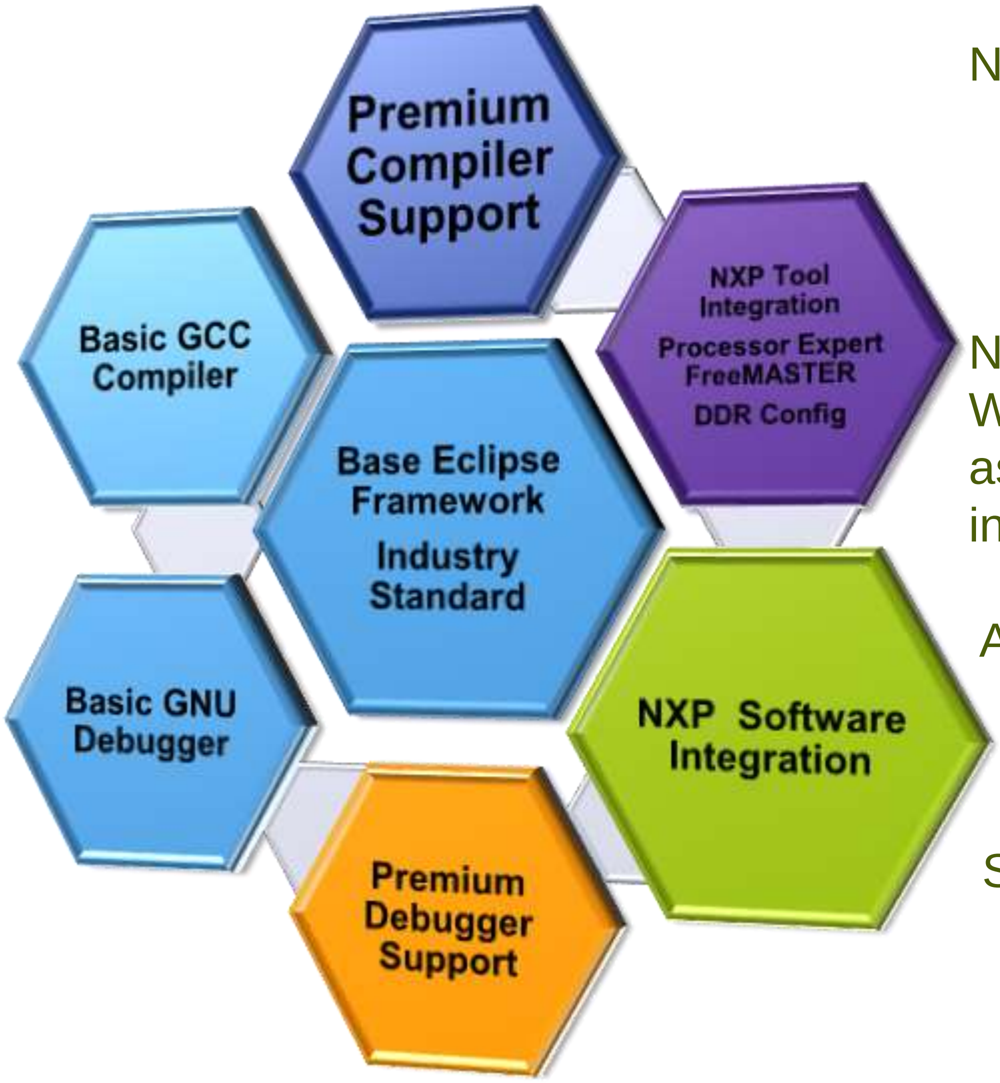
3rd Party Premium Debugger Support

When required for Trace of code execution
and advanced debugging with Industry leading tools

- Lauterbach
- iSystems
- PLS

These debuggers plug-in to the S32 Development Studio seamlessly integrated for use when the most difficult software problems need to be solved

# S32 Design Studio – NXP Software and Tools Integration



NXP Developed Tools Support Integrated into S32 Design Studio

- Processor Expert
- FreeMASTER
- DDR Configuration (Processor Expert)

NXP Software Integrated into the tool as part of shipping package When customer creates a new project he can include NXP software as part of project creation, no more user needing to search for and integrate Freescale software

Automotive Math And Motor Control Libraries

- Libraries included automatically
- Use just needs to drag functions into source files to utilize

Software SDK functions available in the environment

- Bare Metal Drivers available directly in tool for ease of use
- Drag and drop into source to utilize.

# 02.

## Current Tools for ARM Based Processors

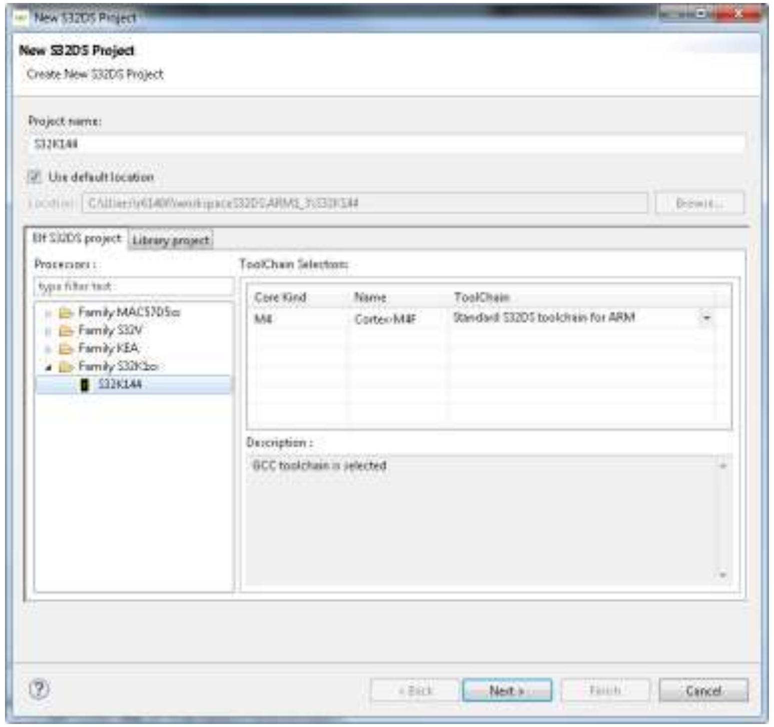
Tools Supporting our Families of ARM Processors

# S32 Design Studio for ARM Tools

- Eclipse Based Integrated Development Environment
  - Editor for Source code Editing
- GCC Compiler build into tool
  - GCC for ARM M-core and A-core based processors
  - GHS and IAR compilers  - (Premium Compilers)
- Integrated Debuggers
  - Low Cost Debugger
    - P&E Micro
    - Segger
    - IAR
    - iSystems,
  - High End Debugger (Premium Debuggers)
    - Lauterbach
    - iSystems

# New Project Wizard  - S32DS for ARM

- Create a New Project
  - Select Part
  - Select SW Integration



**New S32DS Project**

Create New S32DS Project

Project name:

S32K144

☑ Use default location

Location: C:\Users\[illegible]\workspaceS32DS.ARM[illegible]\S32K144

Processors :

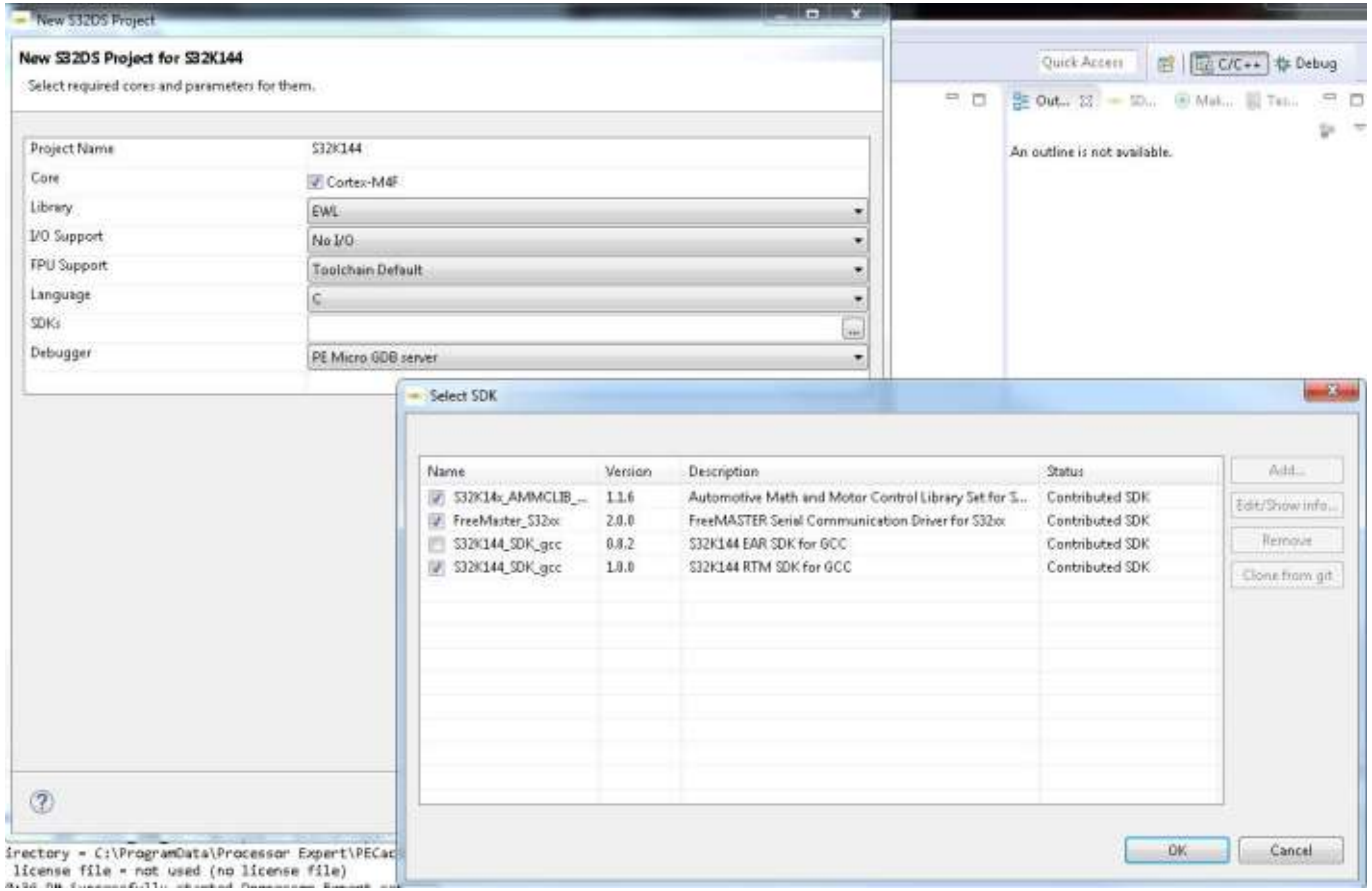
- Family MAC57D5xx
- Family S32V
- Family KEA
- Family S32K1xx
  - S32K144

ToolChain Selection:

| Core Kind | Name | ToolChain |
|---|---|---|
| M4 | Cortex-M4F | Standard S32DS toolchain for ARM |

Description :

GCC toolchain is selected



An outline is not available.

**New S32DS Project for S32K144**

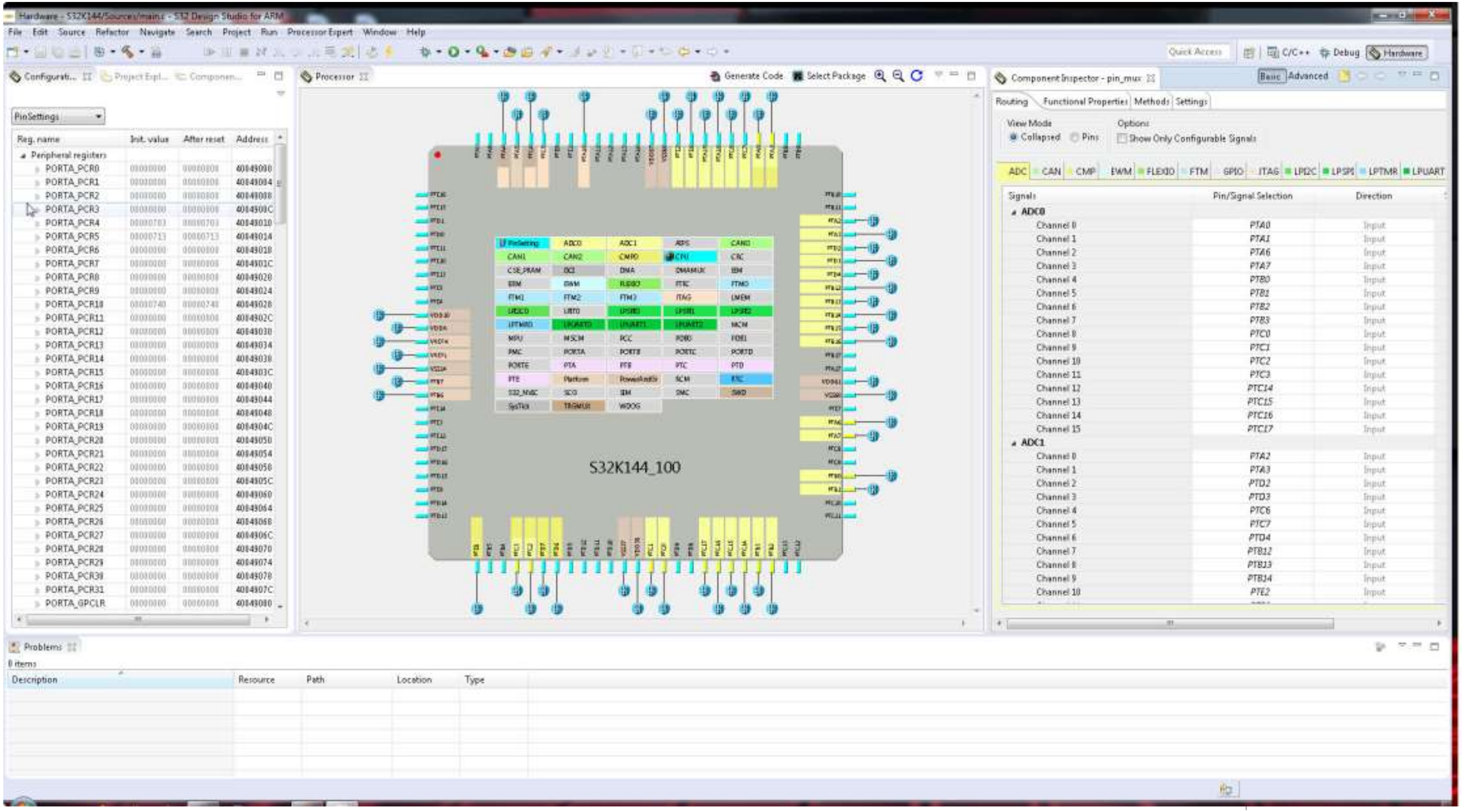
Select required cores and parameters for them.

| | |
|---|---|
| Project Name | S32K144 |
| Core | ☑ Cortex-M4F |
| Library | EWL |
| I/O Support | No I/O |
| FPU Support | Toolchain Default |
| Language | C |
| SDKs | |
| Debugger | PE Micro GDB server |

| Name | Version | Description | Status |
|---|---|---|---|
| ☑ S32K14x_AMMCLIB_... | 1.1.6 | Automotive Math and Motor Control Library Set for S... | Contributed SDK |
| ☑ FreeMaster_S32xx | 2.0.0 | FreeMASTER Serial Communication Driver for S32xx | Contributed SDK |
| □ S32K144_SDK_gcc | 0.8.2 | S32K144 EAR SDK for GCC | Contributed SDK |
| ☑ S32K144_SDK_gcc | 1.0.0 | S32K144 RTM SDK for GCC | Contributed SDK |

```
irectory = C:\ProgramData\Processor Expert\PECac
 license file = not used (no license file)
```

# Configure the Part I/O – Use the Pin Wizard



PinSettings

| Reg. name | Init. value | After reset | Address |
|---|---|---|---|
| Peripheral registers | | | |
| PORTA_PCR0 | 00000000 | 00000000 | 40049000 |
| PORTA_PCR1 | 00000000 | 00000000 | 40049004 |
| PORTA_PCR2 | 00000000 | 00000000 | 40049008 |
| PORTA_PCR3 | 00000000 | 00000000 | 4004900C |
| PORTA_PCR4 | 00000703 | 00000703 | 40049010 |
| PORTA_PCR5 | 00000713 | 00000713 | 40049014 |
| PORTA_PCR6 | 00000000 | 00000000 | 40049018 |
| PORTA_PCR7 | 00000000 | 00000000 | 4004901C |
| PORTA_PCR8 | 00000000 | 00000000 | 40049020 |
| PORTA_PCR9 | 00000000 | 00000000 | 40049024 |
| PORTA_PCR10 | 00000740 | 00000740 | 40049028 |
| PORTA_PCR11 | 00000000 | 00000000 | 4004902C |
| PORTA_PCR12 | 00000000 | 00000000 | 40049030 |
| PORTA_PCR13 | 00000000 | 00000000 | 40049034 |
| PORTA_PCR14 | 00000000 | 00000000 | 40049038 |
| PORTA_PCR15 | 00000000 | 00000000 | 4004903C |
| PORTA_PCR16 | 00000000 | 00000000 | 40049040 |
| PORTA_PCR17 | 00000000 | 00000000 | 40049044 |
| PORTA_PCR18 | 00000000 | 00000000 | 40049048 |
| PORTA_PCR19 | 00000000 | 00000000 | 4004904C |
| PORTA_PCR20 | 00000000 | 00000000 | 40049050 |
| PORTA_PCR21 | 00000000 | 00000000 | 40049054 |
| PORTA_PCR22 | 00000000 | 00000000 | 40049058 |
| PORTA_PCR23 | 00000000 | 00000000 | 4004905C |
| PORTA_PCR24 | 00000000 | 00000000 | 40049060 |
| PORTA_PCR25 | 00000000 | 00000000 | 40049064 |
| PORTA_PCR26 | 00000000 | 00000000 | 40049068 |
| PORTA_PCR27 | 00000000 | 00000000 | 4004906C |
| PORTA_PCR28 | 00000000 | 00000000 | 40049070 |
| PORTA_PCR29 | 00000000 | 00000000 | 40049074 |
| PORTA_PCR30 | 00000000 | 00000000 | 40049078 |
| PORTA_PCR31 | 00000000 | 00000000 | 4004907C |
| PORTA_GPCLR | 00000000 | 00000000 | 40049080 |

View Mode: ● Collapsed ○ Pins

Options: □ Show Only Configurable Signals

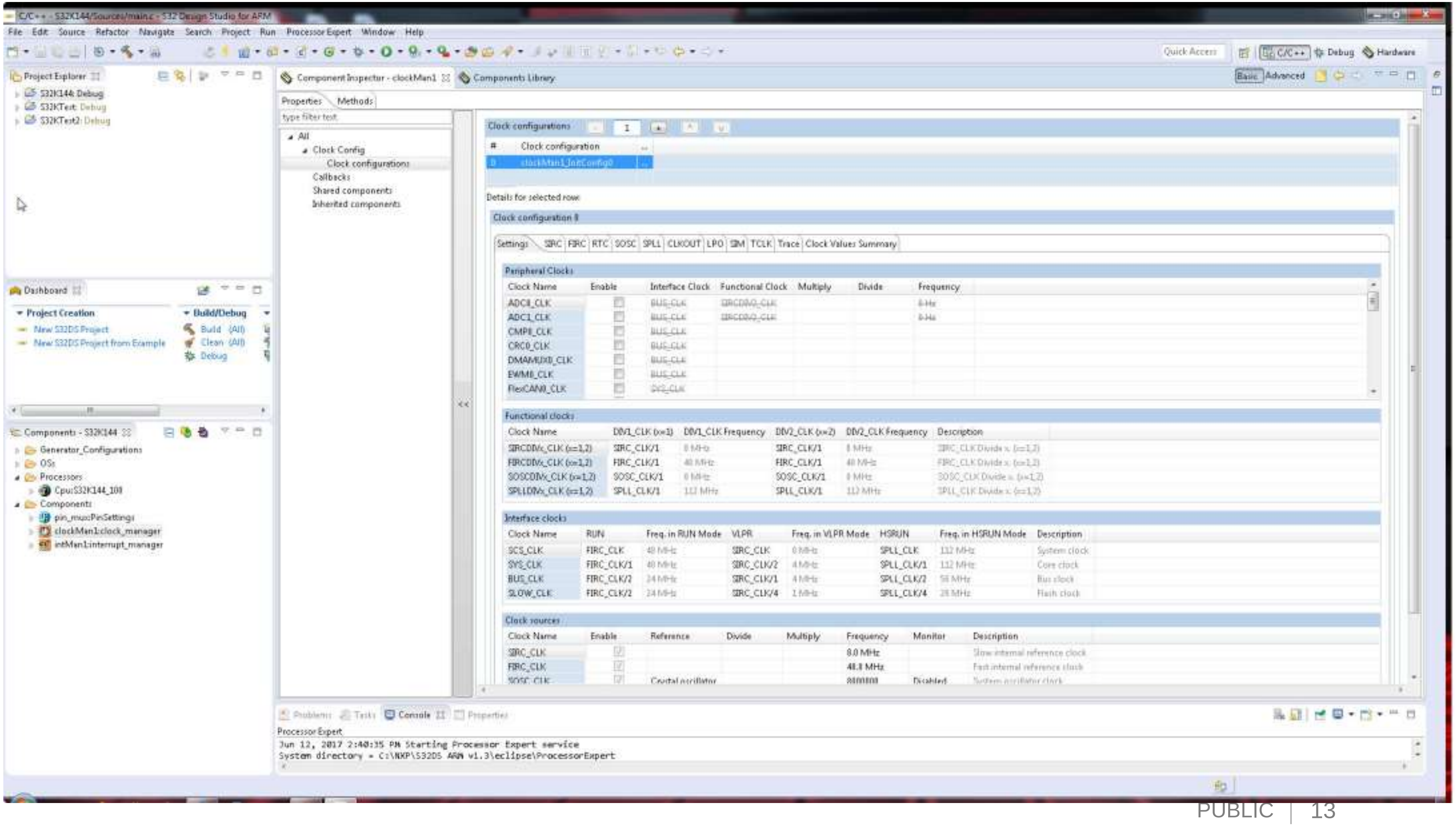
| Signals | Pin/Signal Selection | Direction |
|---|---|---|
| ADC0 | | |
| Channel 0 | PTA0 | Input |
| Channel 1 | PTA1 | Input |
| Channel 2 | PTA6 | Input |
| Channel 3 | PTA7 | Input |
| Channel 4 | PTB0 | Input |
| Channel 5 | PTB1 | Input |
| Channel 6 | PTB2 | Input |
| Channel 7 | PTB3 | Input |
| Channel 8 | PTC0 | Input |
| Channel 9 | PTC1 | Input |
| Channel 10 | PTC2 | Input |
| Channel 11 | PTC3 | Input |
| Channel 12 | PTC14 | Input |
| Channel 13 | PTC15 | Input |
| Channel 14 | PTC16 | Input |
| Channel 15 | PTC17 | Input |
| ADC1 | | |
| Channel 0 | PTA2 | Input |
| Channel 1 | PTA3 | Input |
| Channel 2 | PTD2 | Input |
| Channel 3 | PTD3 | Input |
| Channel 4 | PTC6 | Input |
| Channel 5 | PTC7 | Input |
| Channel 6 | PTD4 | Input |
| Channel 7 | PTB12 | Input |
| Channel 8 | PTB13 | Input |
| Channel 9 | PTB14 | Input |
| Channel 10 | PTE2 | Input |

Problems

0 items

| Description | Resource | Path | Location | Type |
|---|---|---|---|---|

# Configure the Devices on the Part – Clocks



Peripheral Clocks

| Clock Name | Enable | Interface Clock | Functional Clock | Multiply | Divide | Frequency |
|---|---|---|---|---|---|---|
| ADC0_CLK | □ | BUS_CLK | SIRCDIV2_CLK | | | 0 Hz |
| ADC1_CLK | □ | BUS_CLK | SIRCDIV2_CLK | | | 0 Hz |
| CMP0_CLK | □ | BUS_CLK | | | | |
| CRC0_CLK | □ | BUS_CLK | | | | |
| DMAMUX0_CLK | □ | BUS_CLK | | | | |
| EWM0_CLK | □ | BUS_CLK | | | | |
| FlexCAN0_CLK | □ | SYS_CLK | | | | |

Functional clocks

| Clock Name | DIV1_CLK (x=1) | DIV1_CLK Frequency | DIV2_CLK (x=2) | DIV2_CLK Frequency | Description |
|---|---|---|---|---|---|
| SIRCDIVx_CLK (x=1,2) | SIRC_CLK/1 | 8 MHz | SIRC_CLK/1 | 8 MHz | SIRC_CLK Divide x (x=1,2) |
| FIRCDIVx_CLK (x=1,2) | FIRC_CLK/1 | 48 MHz | FIRC_CLK/1 | 48 MHz | FIRC_CLK Divide x (x=1,2) |
| SOSCDIVx_CLK (x=1,2) | SOSC_CLK/1 | 8 MHz | SOSC_CLK/1 | 8 MHz | SOSC_CLK Divide x (x=1,2) |
| SPLLDIVx_CLK (x=1,2) | SPLL_CLK/1 | 112 MHz | SPLL_CLK/1 | 112 MHz | SPLL_CLK Divide x (x=1,2) |

Interface clocks

| Clock Name | RUN | Freq. in RUN Mode | VLPR | Freq. in VLPR Mode | HSRUN | Freq. in HSRUN Mode | Description |
|---|---|---|---|---|---|---|---|
| SCS_CLK | FIRC_CLK | 48 MHz | SIRC_CLK | 8 MHz | SPLL_CLK | 112 MHz | System clock |
| SYS_CLK | FIRC_CLK/1 | 48 MHz | SIRC_CLK/2 | 4 MHz | SPLL_CLK/1 | 112 MHz | Core clock |
| BUS_CLK | FIRC_CLK/2 | 24 MHz | SIRC_CLK/1 | 4 MHz | SPLL_CLK/2 | 56 MHz | Bus clock |
| SLOW_CLK | FIRC_CLK/2 | 24 MHz | SIRC_CLK/4 | 1 MHz | SPLL_CLK/4 | 28 MHz | Flash clock |

Clock sources

| Clock Name | Enable | Reference | Divide | Multiply | Frequency | Monitor | Description |
|---|---|---|---|---|---|---|---|
| SIRC_CLK | ☑ | | | | 8.0 MHz | | Slow internal reference clock |
| FIRC_CLK | ☑ | | | | 48.0 MHz | | Fast internal reference clock |
| SOSC_CLK | ☑ | Crystal oscillator | | | 8000000 | Disabled | System oscillator clock |

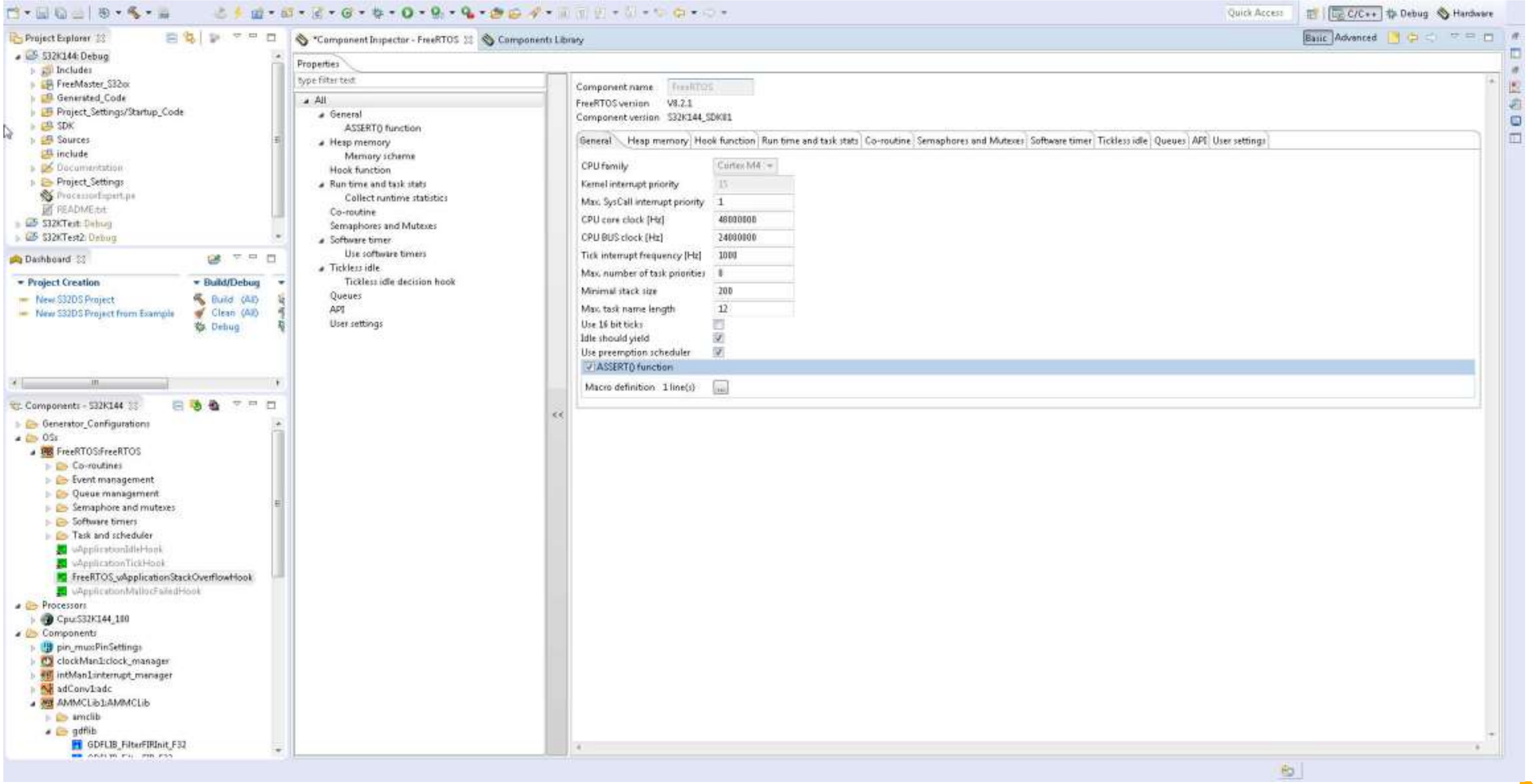
Processor Expert

```
Jun 12, 2017 2:40:35 PM Starting Processor Expert service
System directory = C:\NXP\S32DS ARM v1.3\eclipse\ProcessorExpert
```

# S32 Design Studio for ARM Tools

- Processor Expert Configuration Tool
  - Peripheral and Clock configuration
  - Pin I/O allocation and configuration
  - SDK Driver configuration - S32 SDK for S32K
  - RTOS Configuration – FreeRTOS for S32K
- Integrated SDKs for KEA, MAC57xx, S32K
  - KEA SDK – Demonstration Quality
  - MAC57xx – Demonstration Quality
  - S32K – Production Quality
- Integrated RTOS for MAC57xx and S32K
  - MQX for MAC57xx
  - FreeRTOS for S32K
- Integrated Automotive Math and Motor Control Libraries – Core Optimized Functions
  - KEA
  - S32K

# FreeRTOS Configuration



Component name FreeRTOS

FreeRTOS version V8.2.1

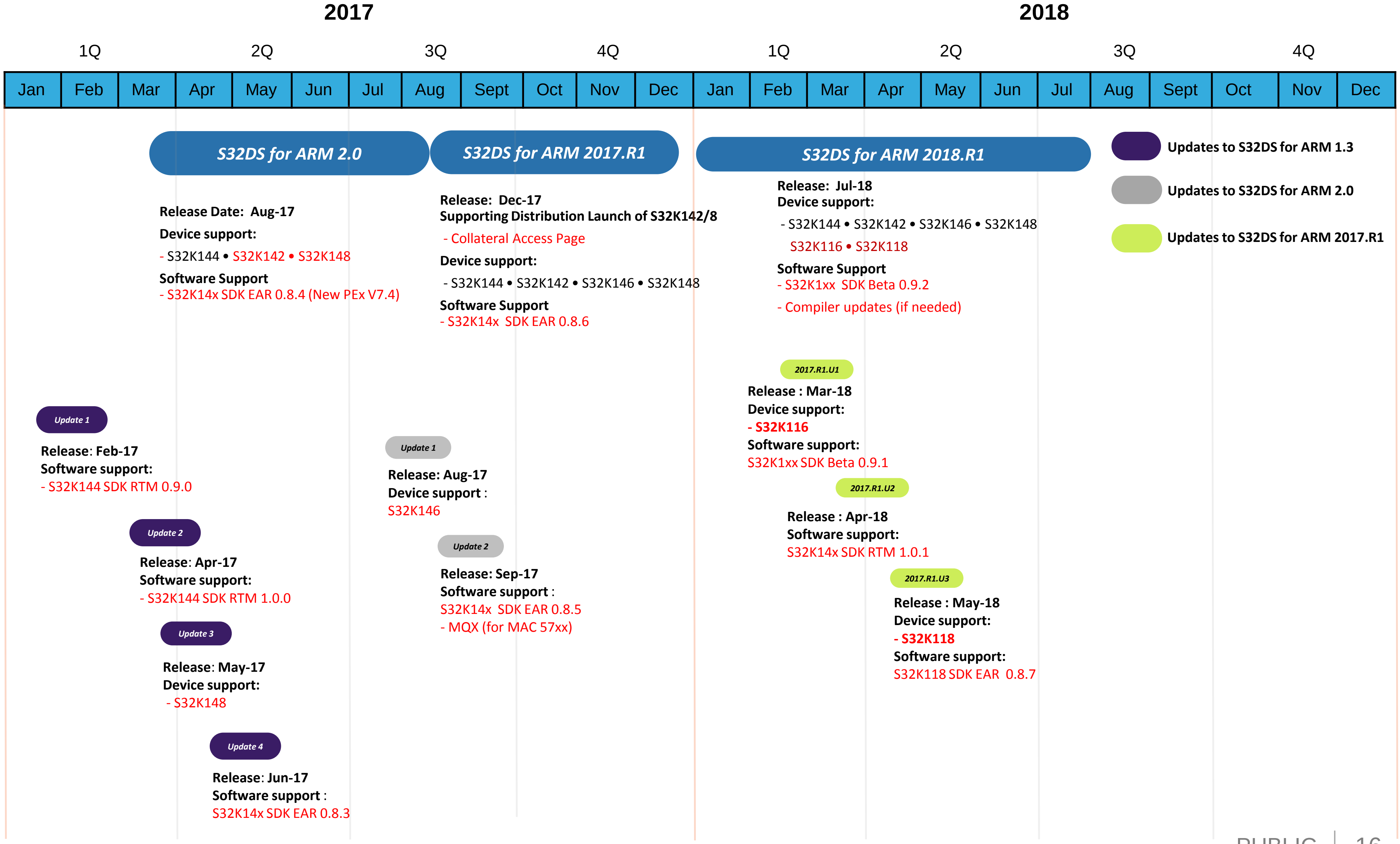
Component version S32K144_SDK01

| | |
|---|---|
| CPU family | Cortex M4 |
| Kernel interrupt priority | 15 |
| Max. SysCall interrupt priority | 1 |
| CPU core clock [Hz] | 48000000 |
| CPU BUS clock [Hz] | 24000000 |
| Tick interrupt frequency [Hz] | 1000 |
| Max. number of task priorities | 8 |
| Minimal stack size | 200 |
| Max. task name length | 12 |
| Use 16 bit ticks | ☐ |
| Idle should yield | ☑ |
| Use preemption scheduler | ☑ |

☑ ASSERT() function

Macro definition 1 line(s) ...

# S32 Design Studio for ARM Roadmap 2017 – 2018



2017 | 2018

1Q 2Q 3Q 4Q 1Q 2Q 3Q 4Q

Jan Feb Mar Apr May Jun Jul Aug Sept Oct Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sept Oct Nov Dec

**S32DS for ARM 2.0**

Release Date: Aug-17

Device support:

- S32K144 • S32K142 • S32K148

Software Support

- S32K14x SDK EAR 0.8.4 (New PEx V7.4)

**S32DS for ARM 2017.R1**

Release: Dec-17

Supporting Distribution Launch of S32K142/8

- Collateral Access Page

Device support:

- S32K144 • S32K142 • S32K146 • S32K148

Software Support

- S32K14x SDK EAR 0.8.6

**S32DS for ARM 2018.R1**

Release: Jul-18

Device support:

- S32K144 • S32K142 • S32K146 • S32K148 S32K116 • S32K118

Software Support

- S32K1xx SDK Beta 0.9.2
- Compiler updates (if needed)

● Updates to S32DS for ARM 1.3

● Updates to S32DS for ARM 2.0

● Updates to S32DS for ARM 2017.R1

Update 1

Release: Feb-17

Software support:

- S32K144 SDK RTM 0.9.0

Update 2

Release: Apr-17

Software support:

- S32K144 SDK RTM 1.0.0

Update 3

Release: May-17

Device support:

- S32K148

Update 4

Release: Jun-17

Software support :

S32K14x SDK EAR 0.8.3

Update 1

Release: Aug-17

Device support :

S32K146

Update 2

Release: Sep-17

Software support :

S32K14x SDK EAR 0.8.5

- MQX (for MAC 57xx)

2017.R1.U1

Release : Mar-18

Device support:

- **S32K116**

Software support:

S32K1xx SDK Beta 0.9.1

2017.R1.U2

Release : Apr-18

Software support:

S32K14x SDK RTM 1.0.1

2017.R1.U3

Release : May-18

Device support:

- **S32K118**

Software support:

S32K118 SDK EAR 0.8.7

**Common features/Tools support:**

**Compiler Support**

- GCC (M0+,M4,A5,A53)
- GHS, IAR

**Device support:**

- KEA
- S32K1xx
- S32V234
- MAC57xx

**Integrated Debugger Support**

- Lauterbach
- P&E
- Segger
- iSystems
- IAR

**Tools Libraries Integrated**

- FreeMASTER Interface Library

**Software Integration**

- S32K SDK (production quality SDK)
- KEA SDK (demo quality)
- MAC57xx SDK (demo quality)
- FreeRTOS OS (S32K)
- MQX OS (MAC57xx)
- AMMC Library (production quality)

**Miscellaneous Features**

- Processor Expert Configuration Tool
- Pin Mux Tool
- Collateral Access Page
- SDK Browser

# 03.

# Current Tools for e200 Processors

Tools Supporting our Families of e200 based Processors (MPC56xx,MPC57xx)

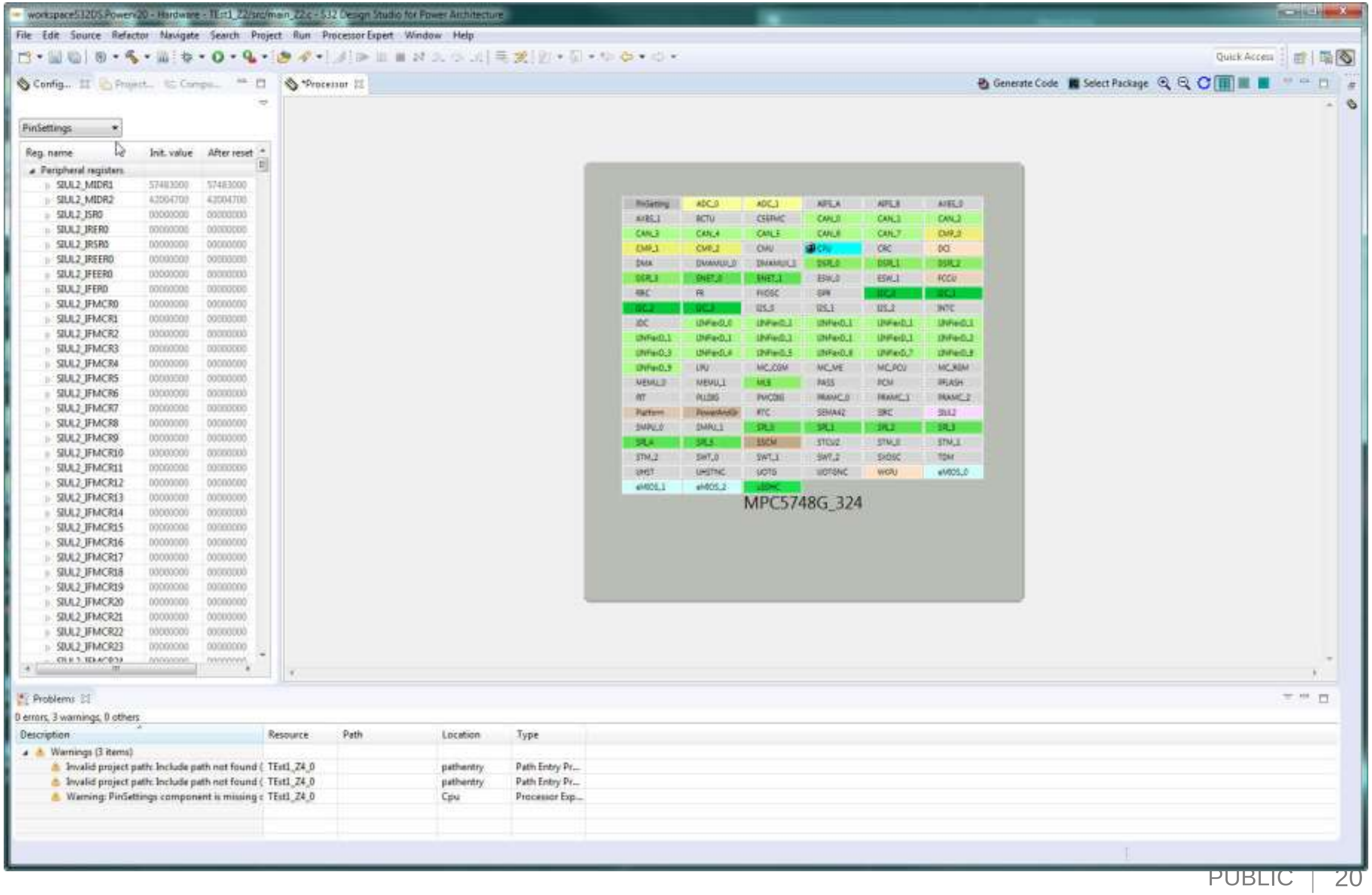
# S32 Design Studio for e200 Tools

- Eclipse Based Integrated Development Environment
  - Editor for Source code Editing
- GCC Compiler build into tool
  - GCC for e200 cores V4.9
  - GHS and DIAB compilers  - (Premium Compilers)
- Integrated Debuggers
  - Low Cost Debugger
    - P&E Micro
    - iSystems
  - High End Debugger (Premium Debuggers)
    - Lauterbach
    - PLS
    - iSystems

# S32 Design Studio for ARM Tools

- Integrated SDKs for MPC574xG
  - MPC57xG SDK – Production Quality
- Integrated RTOS for MPC574xG
  - FreeRTOS for MPC574xG
- Integrated Automotive Math and Motor Control Libraries – Core Optimized Functions
  - All e200 Cores
- NPI Support
  - MPC57xx Family and Cores
  - 80% of MPC56xx Family and Cores

# S32 Design Studio with Pin Configuration



PinSettings

| Reg. name | Init. value | After reset |
|---|---|---|
| Peripheral registers | | |
| SIUL2_MIDR1 | 57483000 | 57483000 |
| SIUL2_MIDR2 | 42004700 | 42004700 |
| SIUL2_ISR0 | 00000000 | 00000000 |
| SIUL2_IRER0 | 00000000 | 00000000 |
| SIUL2_IRSR0 | 00000000 | 00000000 |
| SIUL2_IREER0 | 00000000 | 00000000 |
| SIUL2_IFEER0 | 00000000 | 00000000 |
| SIUL2_IFER0 | 00000000 | 00000000 |
| SIUL2_IFMCR0 | 00000000 | 00000000 |
| SIUL2_IFMCR1 | 00000000 | 00000000 |
| SIUL2_IFMCR2 | 00000000 | 00000000 |
| SIUL2_IFMCR3 | 00000000 | 00000000 |
| SIUL2_IFMCR4 | 00000000 | 00000000 |
| SIUL2_IFMCR5 | 00000000 | 00000000 |
| SIUL2_IFMCR6 | 00000000 | 00000000 |
| SIUL2_IFMCR7 | 00000000 | 00000000 |
| SIUL2_IFMCR8 | 00000000 | 00000000 |
| SIUL2_IFMCR9 | 00000000 | 00000000 |
| SIUL2_IFMCR10 | 00000000 | 00000000 |
| SIUL2_IFMCR11 | 00000000 | 00000000 |
| SIUL2_IFMCR12 | 00000000 | 00000000 |
| SIUL2_IFMCR13 | 00000000 | 00000000 |
| SIUL2_IFMCR14 | 00000000 | 00000000 |
| SIUL2_IFMCR15 | 00000000 | 00000000 |
| SIUL2_IFMCR16 | 00000000 | 00000000 |
| SIUL2_IFMCR17 | 00000000 | 00000000 |
| SIUL2_IFMCR18 | 00000000 | 00000000 |
| SIUL2_IFMCR19 | 00000000 | 00000000 |
| SIUL2_IFMCR20 | 00000000 | 00000000 |
| SIUL2_IFMCR21 | 00000000 | 00000000 |
| SIUL2_IFMCR22 | 00000000 | 00000000 |
| SIUL2_IFMCR23 | 00000000 | 00000000 |
| SIUL2_IFMCR24 | 00000000 | 00000000 |

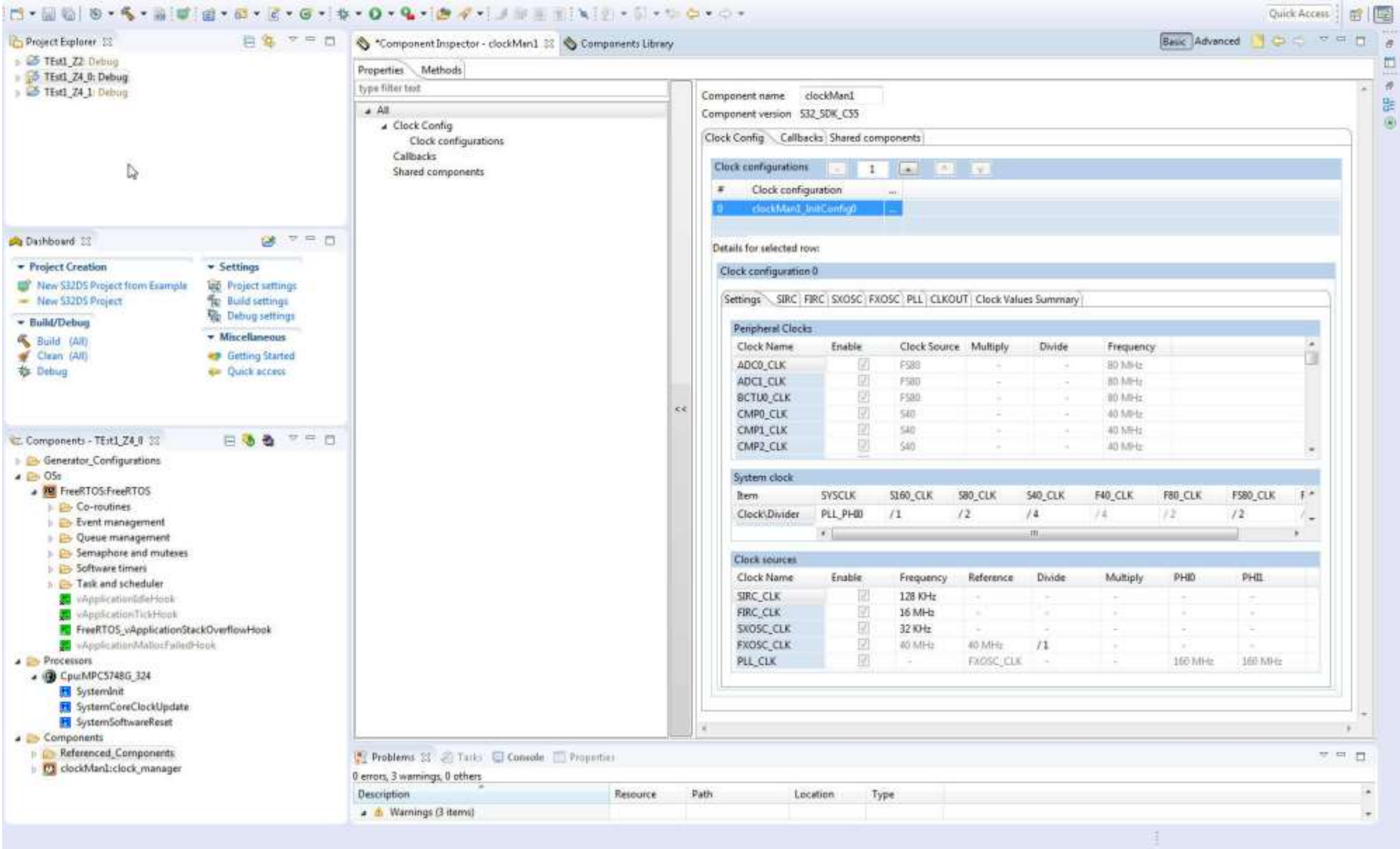
| | | | | | |
|---|---|---|---|---|---|
| PinSettings | ADC_0 | ADC_1 | AIPS_A | AIPS_B | AXBS_0 |
| AXBS_1 | BCTU | CSEPMC | CAN_0 | CAN_1 | CAN_2 |
| CAN_3 | CAN_4 | CAN_5 | CAN_6 | CAN_7 | CMP_0 |
| CMP_1 | CMP_2 | CMU | CPU | CRC | DCI |
| DMA | DMAMUX_0 | DMAMUX_1 | DSPI_0 | DSPI_1 | DSPI_2 |
| DSPI_3 | ENET_0 | ENET_1 | ESW_0 | ESW_1 | FCCU |
| FIRC | FR | FXOSC | GPR | I2C_0 | I2C_1 |
| I2C_2 | I2C_3 | I2S_0 | I2S_1 | I2S_2 | INTC |
| JDC | LINFlexD_0 | LINFlexD_1 | LINFlexD_10 | LINFlexD_11 | LINFlexD_12 |
| LINFlexD_13 | LINFlexD_14 | LINFlexD_15 | LINFlexD_16 | LINFlexD_17 | LINFlexD_2 |
| LINFlexD_3 | LINFlexD_4 | LINFlexD_5 | LINFlexD_6 | LINFlexD_7 | LINFlexD_8 |
| LINFlexD_9 | LPU | MC_CGM | MC_ME | MC_PCU | MC_RGM |
| MEMU_0 | MEMU_1 | MLB | PASS | PCM | PFLASH |
| PIT | PLLDIG | PMCDIG | PRAMC_0 | PRAMC_1 | PRAMC_2 |
| Platform | PowerAndGr | RTC | SEMA42 | SRC | SIUL2 |
| SMPU_0 | SMPU_1 | SPI_0 | SPI_1 | SPI_2 | SPI_3 |
| SPI_4 | SPI_5 | SSCM | STCU2 | STM_0 | STM_1 |
| STM_2 | SWT_0 | SWT_1 | SWT_2 | SXOSC | TDM |
| UHST | UHSTNC | UOTG | UOTGNC | WKPU | eMIOS_0 |
| eMIOS_1 | eMIOS_2 | uSDHC | | | |

MPC5748G_324

0 errors, 3 warnings, 0 others

| Description | Resource | Path | Location | Type |
|---|---|---|---|---|
| Warnings (3 items) | | | | |
| Invalid project path: Include path not found ( | TEst1_Z4_0 | | pathentry | Path Entry Pr... |
| Invalid project path: Include path not found ( | TEst1_Z4_0 | | pathentry | Path Entry Pr... |
| Warning: PinSettings component is missing c | TEst1_Z4_0 | | Cpu | Processor Exp... |

# S32 Design Studio with Processor Expert clock configuration



Component name: clockMan1

Component version: S32_SDK_C55

Clock Config | Callbacks | Shared components

Clock configurations: 1

| # | Clock configuration | ... |
|---|---|---|
| 0 | clockMan1_InitConfig0 | ... |

Details for selected row:

Clock configuration 0

Settings | SIRC | FIRC | SXOSC | FXOSC | PLL | CLKOUT | Clock Values Summary

Peripheral Clocks

| Clock Name | Enable | Clock Source | Multiply | Divide | Frequency |
|---|---|---|---|---|---|
| ADC0_CLK | ☑ | FS80 | - | - | 80 MHz |
| ADC1_CLK | ☑ | FS80 | - | - | 80 MHz |
| BCTU0_CLK | ☑ | FS80 | - | - | 80 MHz |
| CMP0_CLK | ☑ | S40 | - | - | 40 MHz |
| CMP1_CLK | ☑ | S40 | - | - | 40 MHz |
| CMP2_CLK | ☑ | S40 | - | - | 40 MHz |

System clock

| Item | SYSCLK | S160_CLK | S80_CLK | S40_CLK | F40_CLK | F80_CLK | FS80_CLK | F |
|---|---|---|---|---|---|---|---|---|
| Clock\Divider | PLL_PHI0 | / 1 | / 2 | / 4 | / 4 | / 2 | / 2 | / |

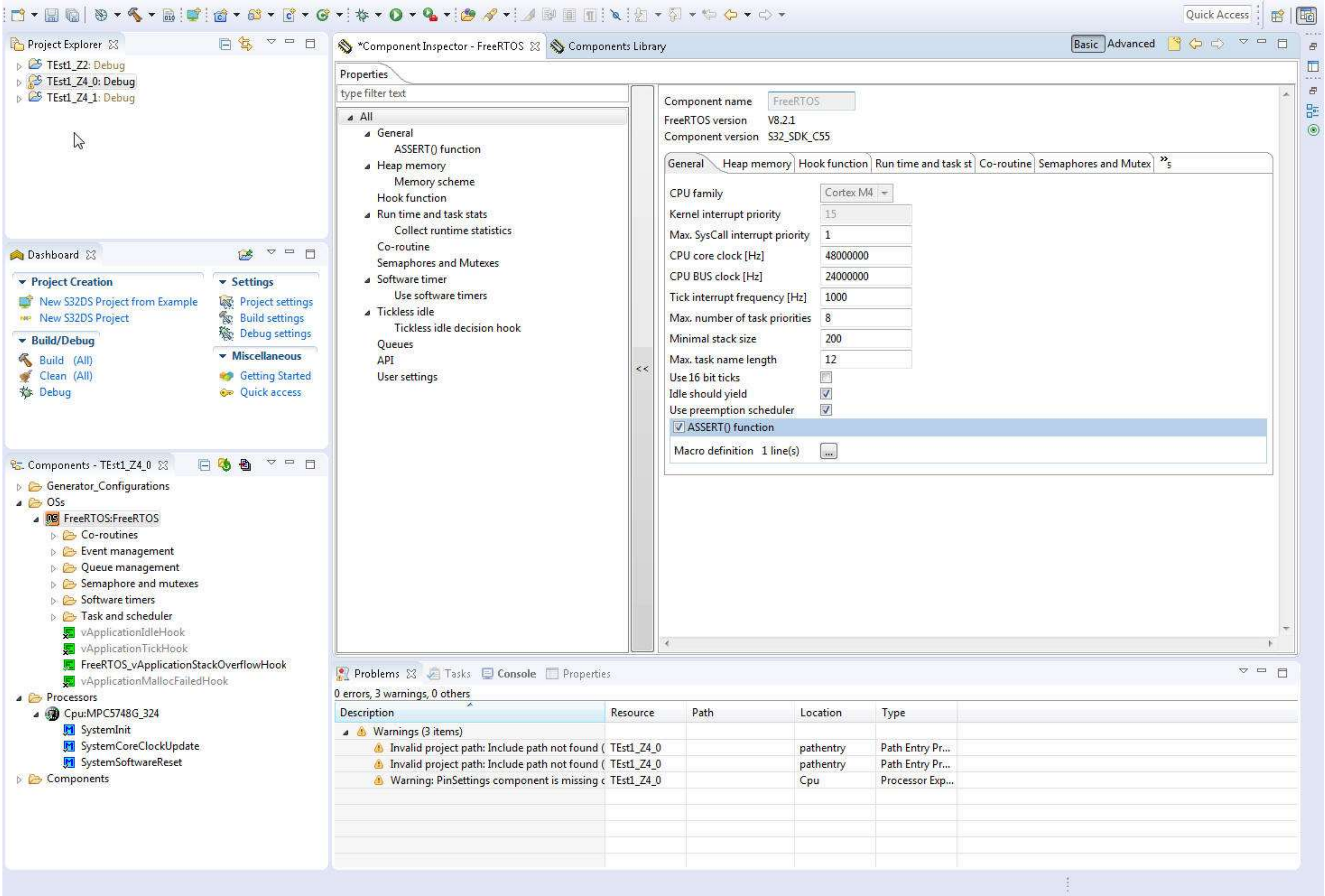
Clock sources

| Clock Name | Enable | Frequency | Reference | Divide | Multiply | PHI0 | PHI1 |
|---|---|---|---|---|---|---|---|
| SIRC_CLK | ☑ | 128 KHz | - | - | - | - | - |
| FIRC_CLK | ☑ | 16 MHz | - | - | - | - | - |
| SXOSC_CLK | ☑ | 32 KHz | - | - | - | - | - |
| FXOSC_CLK | ☑ | 40 MHz | 40 MHz | / 1 | - | - | - |
| PLL_CLK | ☑ | - | FXOSC_CLK | - | - | 160 MHz | 160 MHz |

0 errors, 3 warnings, 0 others

| Description | Resource | Path | Location | Type |
|---|---|---|---|---|
| Warnings (3 items) | | | | |

# S32 Design Studio with Processor Expert and FreeRTOS



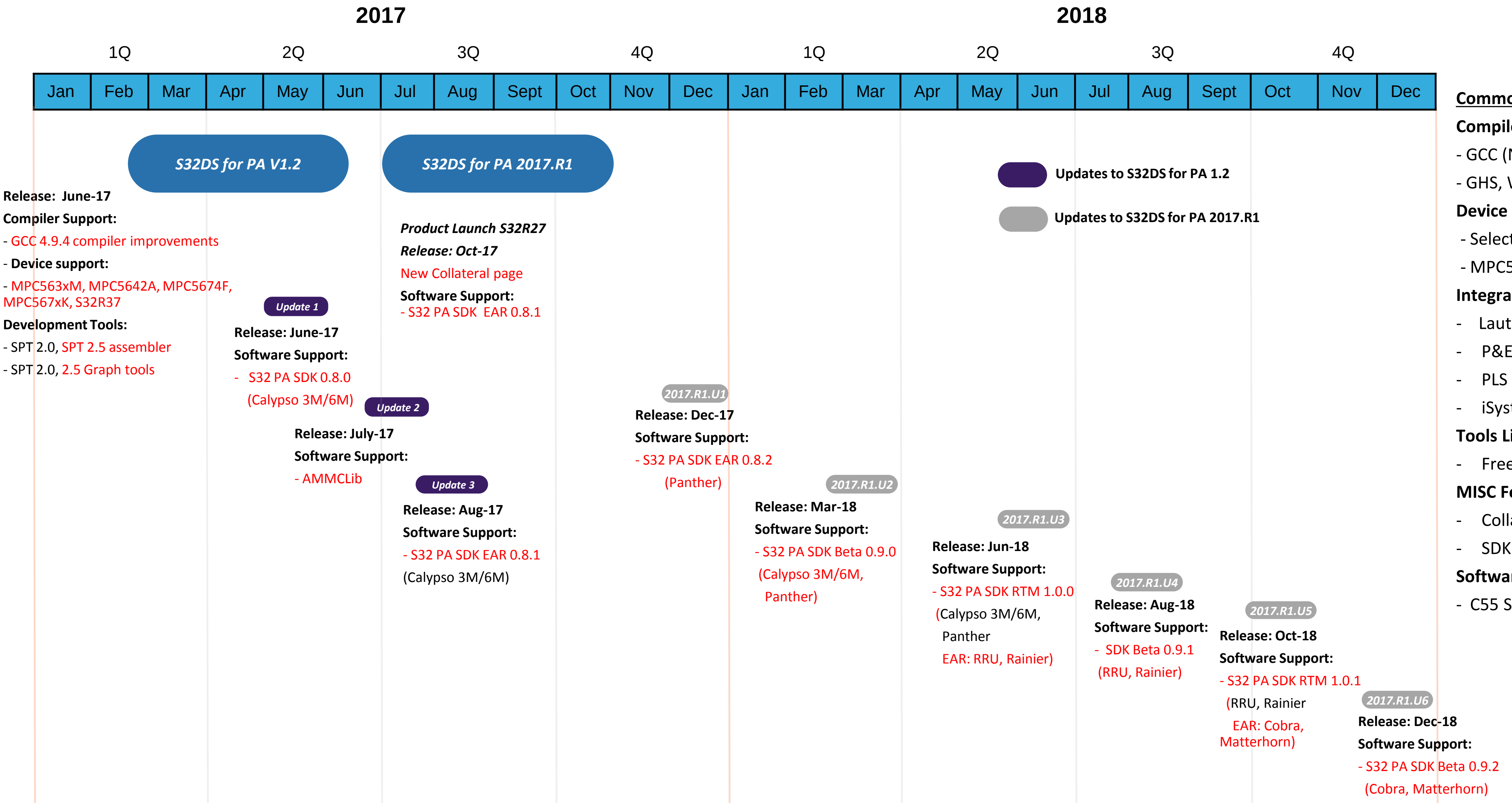
Component name FreeRTOS
FreeRTOS version V8.2.1
Component version S32_SDK_C55

| | |
|---|---|
| CPU family | Cortex M4 |
| Kernel interrupt priority | 15 |
| Max. SysCall interrupt priority | 1 |
| CPU core clock [Hz] | 48000000 |
| CPU BUS clock [Hz] | 24000000 |
| Tick interrupt frequency [Hz] | 1000 |
| Max. number of task priorities | 8 |
| Minimal stack size | 200 |
| Max. task name length | 12 |
| Use 16 bit ticks | □ |
| Idle should yield | ☑ |
| Use preemption scheduler | ☑ |
| ☑ ASSERT() function | |
| Macro definition | 1 line(s) |

0 errors, 3 warnings, 0 others

| Description | Resource | Path | Location | Type |
|---|---|---|---|---|
| Warnings (3 items) | | | | |
| Invalid project path: Include path not found ( | TEst1_Z4_0 | | pathentry | Path Entry Pr... |
| Invalid project path: Include path not found ( | TEst1_Z4_0 | | pathentry | Path Entry Pr... |
| Warning: PinSettings component is missing c | TEst1_Z4_0 | | Cpu | Processor Exp... |

# S32 Design Studio for Power Architecture Roadmap 2017–2018



| 2017 | | | | | | | | | | | | 2018 | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1Q | | | 2Q | | | 3Q | | | 4Q | | | 1Q | | | 2Q | | | 3Q | | | 4Q | | |
| Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sept | Oct | Nov | Dec | Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sept | Oct | Nov | Dec |

**S32DS for PA V1.2**

**Release: June-17**

**Compiler Support:**

- GCC 4.9.4 compiler improvements

- **Device support:**

- MPC563xM, MPC5642A, MPC5674F, MPC567xK, S32R37

**Development Tools:**

- SPT 2.0, SPT 2.5 assembler

- SPT 2.0, 2.5 Graph tools

***Update 1***

**Release: June-17**

**Software Support:**

- S32 PA SDK 0.8.0 (Calypso 3M/6M)

***Update 2***

**Release: July-17**

**Software Support:**

- AMMCLib

***Update 3***

**Release: Aug-17**

**Software Support:**

- S32 PA SDK EAR 0.8.1 (Calypso 3M/6M)

**S32DS for PA 2017.R1**

***Product Launch S32R27***

***Release: Oct-17***

New Collateral page

**Software Support:**

- S32 PA SDK EAR 0.8.1

***2017.R1.U1***

**Release: Dec-17**

**Software Support:**

- S32 PA SDK EAR 0.8.2 (Panther)

***2017.R1.U2***

**Release: Mar-18**

**Software Support:**

- S32 PA SDK Beta 0.9.0 (Calypso 3M/6M, Panther)

***2017.R1.U3***

**Release: Jun-18**

**Software Support:**

- S32 PA SDK RTM 1.0.0 (Calypso 3M/6M, Panther EAR: RRU, Rainier)

***2017.R1.U4***

**Release: Aug-18**

**Software Support:**

- SDK Beta 0.9.1 (RRU, Rainier)

***2017.R1.U5***

**Release: Oct-18**

**Software Support:**

- S32 PA SDK RTM 1.0.1 (RRU, Rainier EAR: Cobra, Matterhorn)

***2017.R1.U6***

**Release: Dec-18**

**Software Support:**

- S32 PA SDK Beta 0.9.2 (Cobra, Matterhorn)

● **Updates to S32DS for PA 1.2**

● **Updates to S32DS for PA 2017.R1**

**Common Features Of Tool:**

**Compiler Support**

- GCC (NXP Supplied)
- GHS, Windriver (DIAB)

**Device support:**

- Select MPC56xx
- MPC57xx

**Integrated Debugger Support**

- Lauterbach
- P&E
- PLS
- iSystems

**Tools Libraries Integrated**

- FreeMASTER Library

**MISC Features**

- Collateral Access Page
- SDK Browser

**Software Integration**

- C55 SDK

# 04.

# Tools for Vision Application Development

Developing Vision Processing Applications

# S32 Design Studio for Vision (S32V)

S32 Design Studio Development Environment for S32V234 by Core

ISP/M0+

Assembler
ISP Graph Tool
Compiler
Debugger
(Lauterbach)
Source Editor
New Project Wizard
Software Examples

Apex 2

NXP APEX C/C++ Compiler
APEX Graph Tool
Debugger (Lauterbach)
Editor
New Project Wizard
Vision SDK
ACF Framework
APEX Emulation of Hardware
Example Projects

M4

C/C++ Compiler
Debugger
(Low Cost Debugger,
Lauterbach)
Source Editor
New Project Wizard

A53

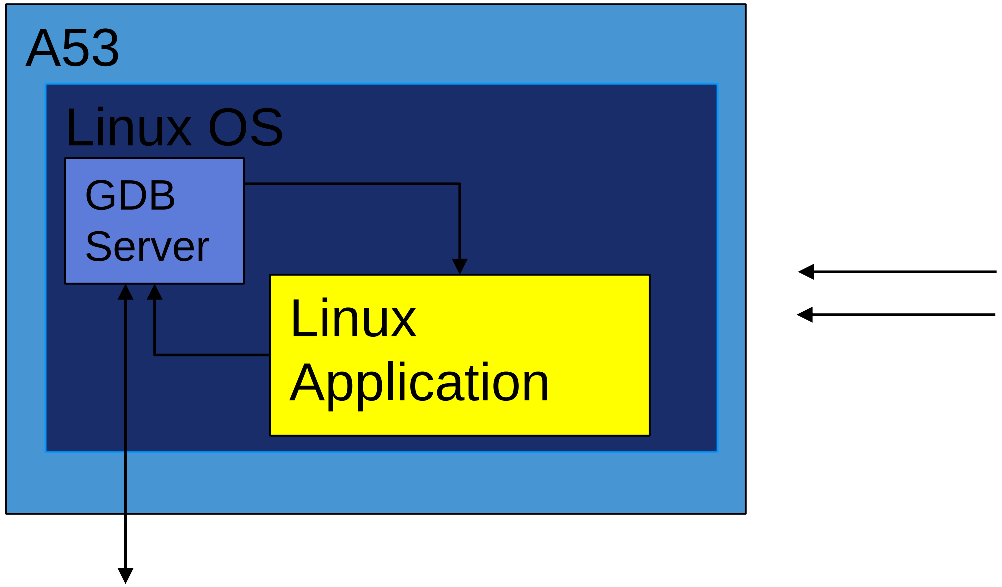
C/C++ Compiler
Multi-Core Debug Support
Task Aware Debug Support
(Low Cost Debugger,
Lauterbach)
Source Editor
New Project Wizard

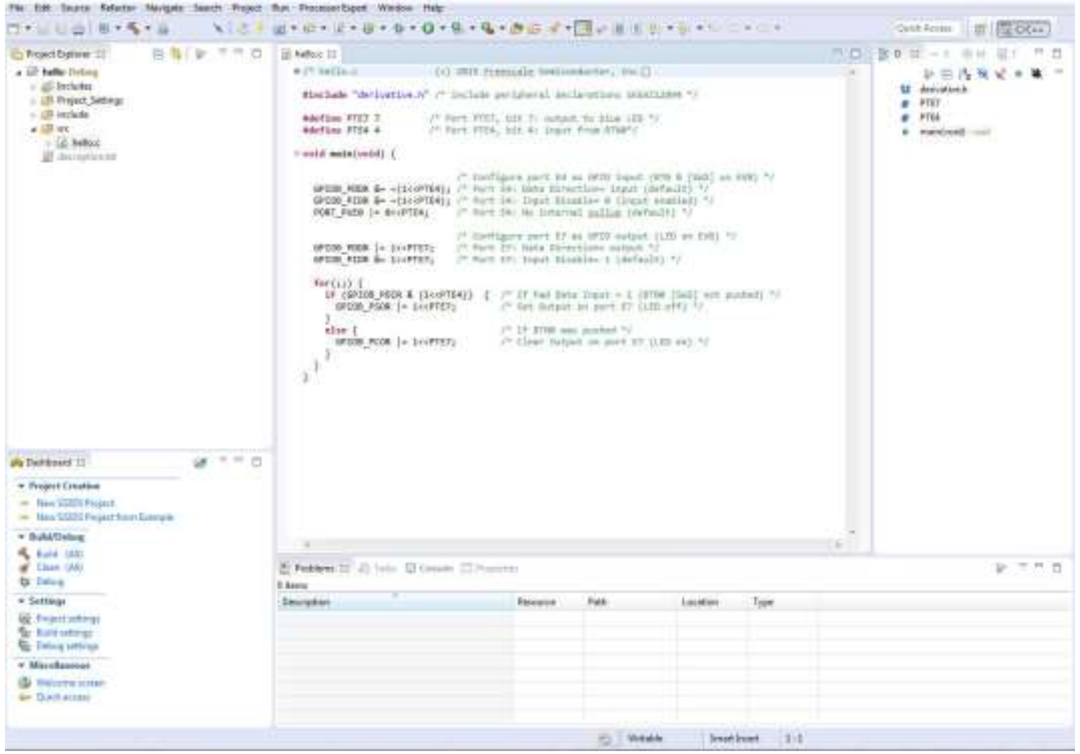
Support for both IDE and Command line development on Windows and Linux platforms.

# Significant Software Integration

- Linux Board Support Package (BSP) Included for SMP A53 Development
  - Support for Linux Application build and Debug
- Vision Software Development Kit
  - ISP Kernels, ISP Software
  - APEX Kernels, APX/APU Software
  - APEX CV and APEX CV Pro Libraries

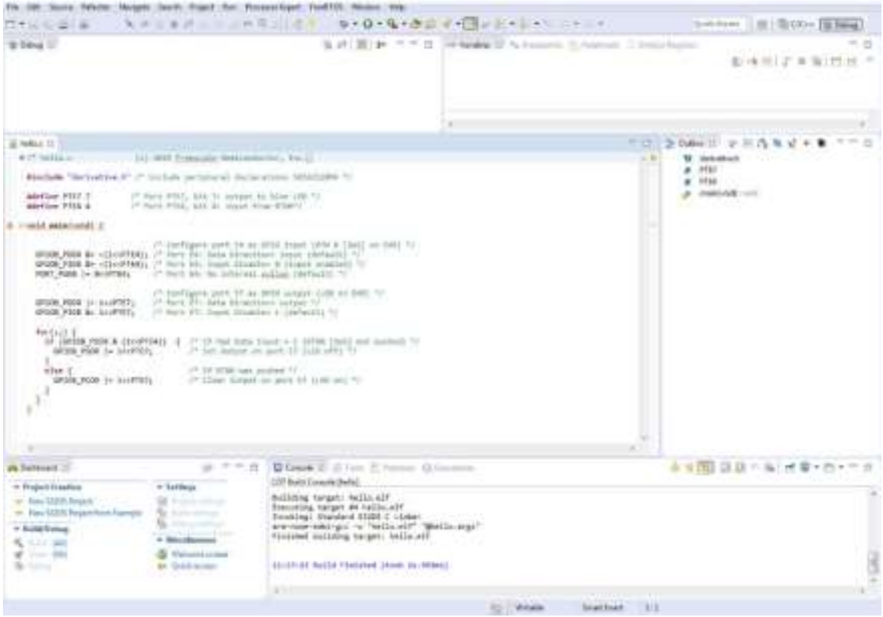
# Linux Application Development and Debug on A53





```
/* hello.c              (c) 2015 Freescale Semiconductor, Inc.

#include "derivative.h" /* include peripheral declarations S32K144 */

#define PTE7 7        /* Port PTE7, bit 7: output to blue LED */
#define PTE4 4        /* Port PTE4, bit 4: input from BTN0*/

void main(void) {
                      /* Configure port E4 as GPIO input (BTN 0 [SW2] on EVB) */
  GPIOB_PDDR &= ~(1<<PTE4); /* Port E4: Data Direction= input (default) */
  GPIOB_PIDR &= ~(1<<PTE4); /* Port E4: Input Disable= 0 (input enabled) */
  PORT_PCR4 |= 0<<PTE4;     /* Port E4: No internal pullup (default) */

                      /* Configure port E7 as GPIO output (LED on EVB) */
  GPIOB_PDDR |= 1<<PTE7;    /* Port E7: Data Direction= output */
  GPIOB_PIDR &= 1<<PTE7;    /* Port E7: Input Disable= 1 (default) */

  for(;;) {
    if (GPIOB_PDIR & (1<<PTE4)) {  /* If Pad Data Input = 1 (BTN0 [SW2] not pushed) */
      GPIOB_PSOR |= 1<<PTE7;        /* Set Output on port E7 (LED off) */
    }
    else {                          /* If BTN0 was pushed */
      GPIOB_PCOR |= 1<<PTE7;        /* Clear Output on port E7 (LED on) */
    }
  }
}
```

Build Linux Application and Download to A53



Debug Linux Application on A53 In S32 Design Studio through GDB server in Linux OS

# S32V234 The Vision Pipeline

Each engine offers the best efficiency for certain type of functions. To let the complete system work at highest efficiency, each engine needs to work in parallel in pipeline mode.

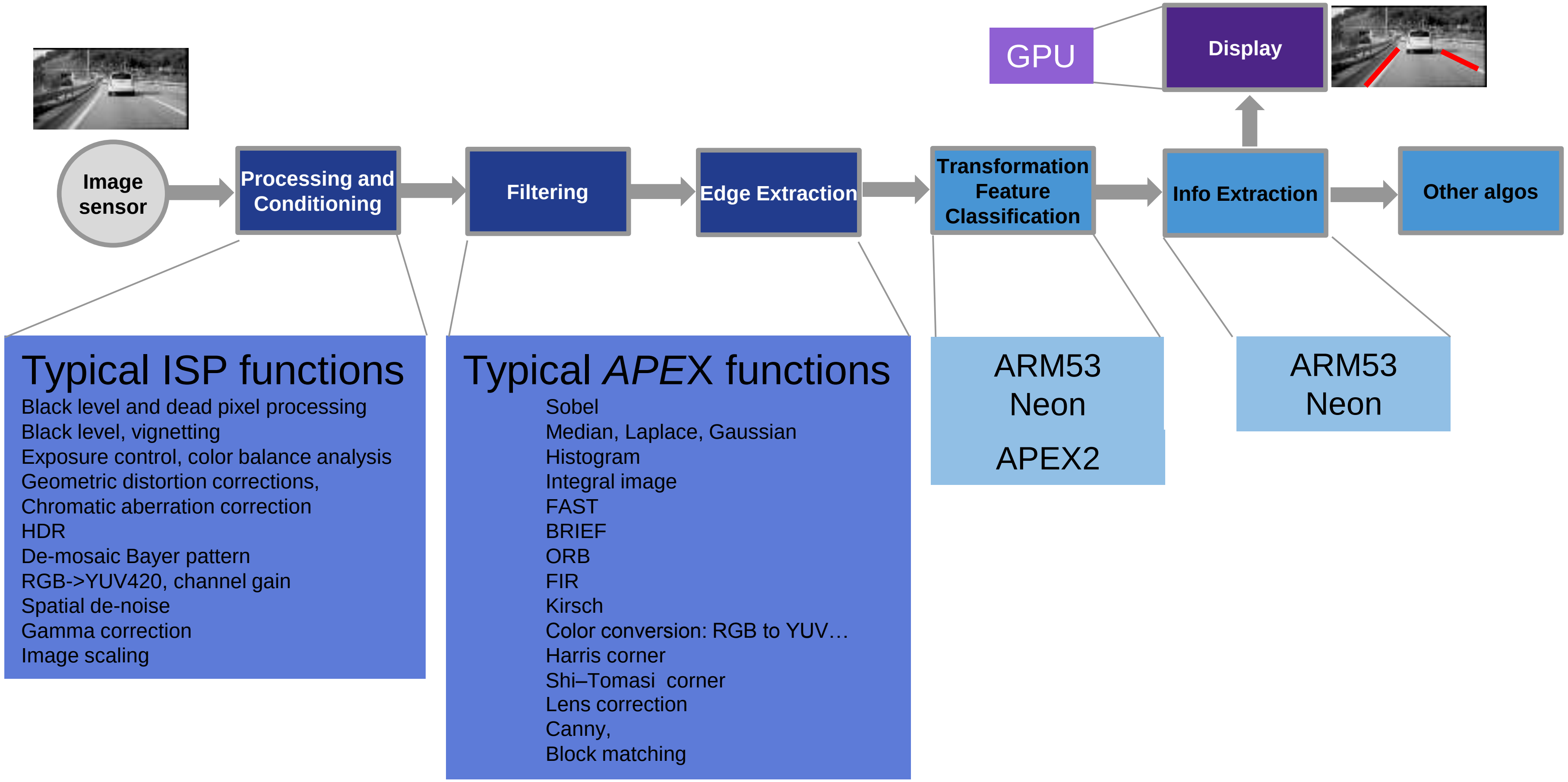


Image sensor → Processing and Conditioning → Filtering → Edge Extraction → Transformation Feature Classification → Info Extraction → Other algos

Info Extraction → Display

GPU

Display

## Typical ISP functions

Black level and dead pixel processing
Black level, vignetting
Exposure control, color balance analysis
Geometric distortion corrections,
Chromatic aberration correction
HDR
De-mosaic Bayer pattern
RGB->YUV420, channel gain
Spatial de-noise
Gamma correction
Image scaling

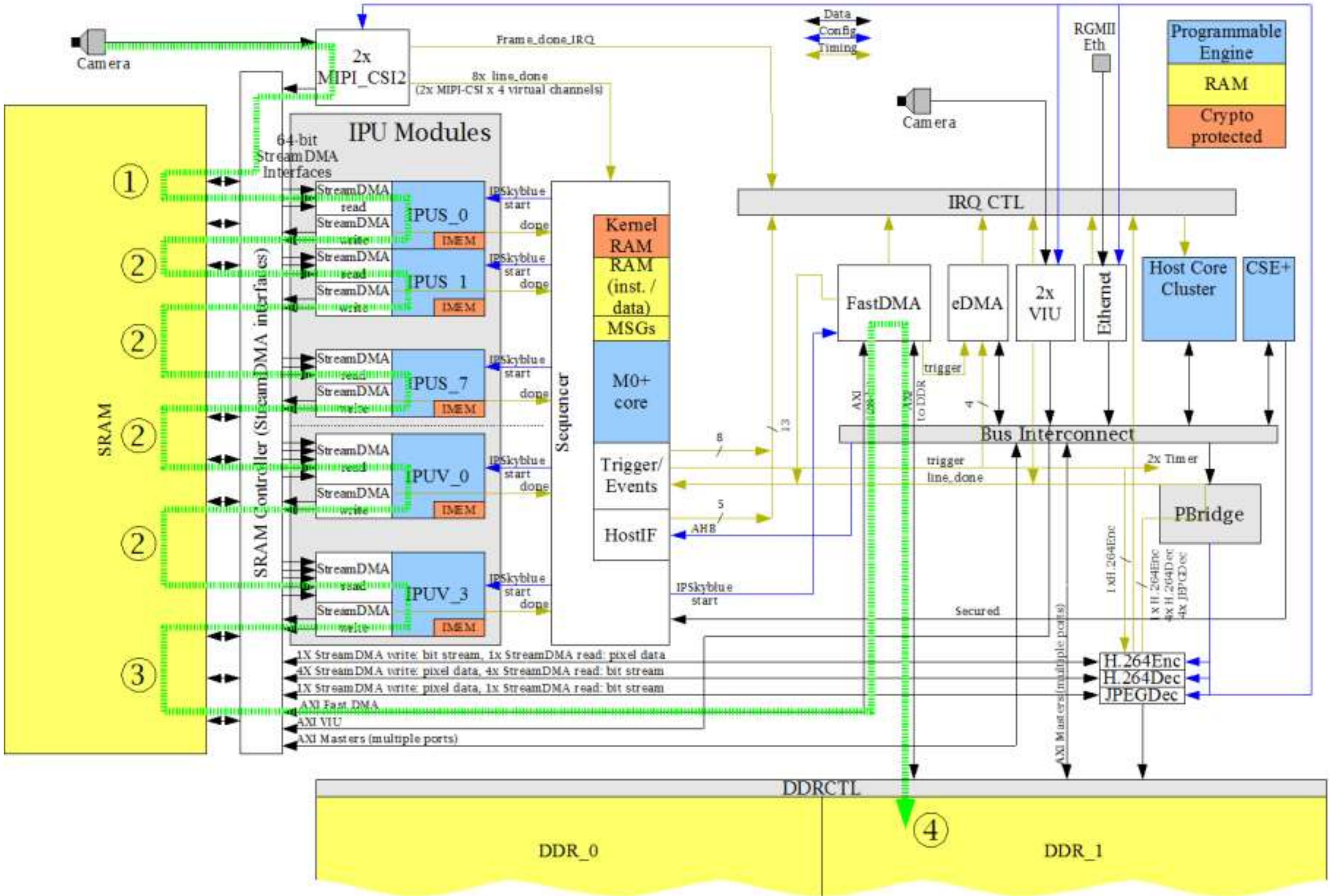
## Typical *APEX* functions

Sobel
Median, Laplace, Gaussian
Histogram
Integral image
FAST
BRIEF
ORB
FIR
Kirsch
Color conversion: RGB to YUV…
Harris corner
Shi–Tomasi corner
Lens correction
Canny,
Block matching

ARM53
Neon
APEX2

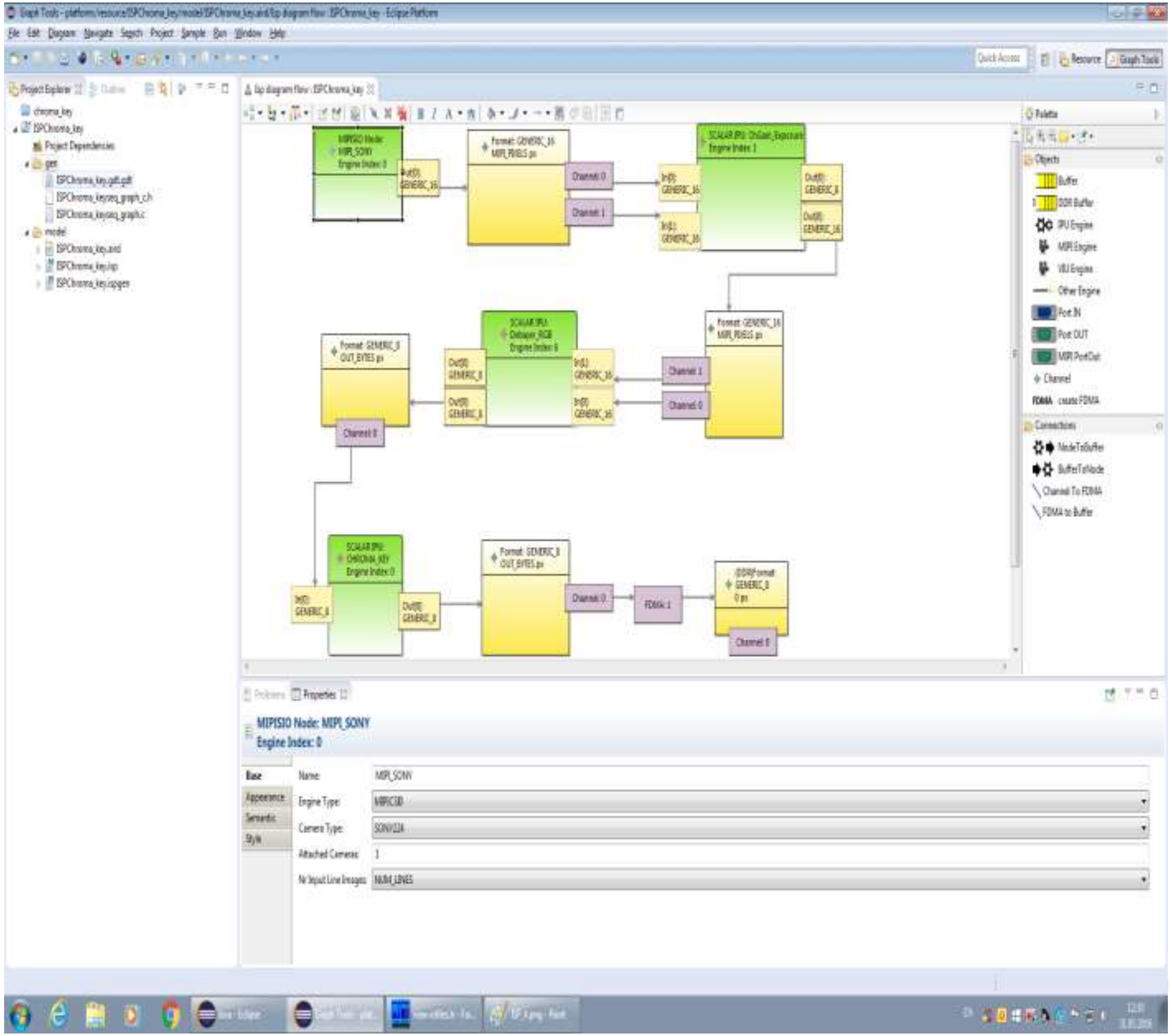
ARM53
Neon

# ISP SoC Integration and Data Flow



# ISP Graph Tool – Example : Vision Algorithm Development on ISP

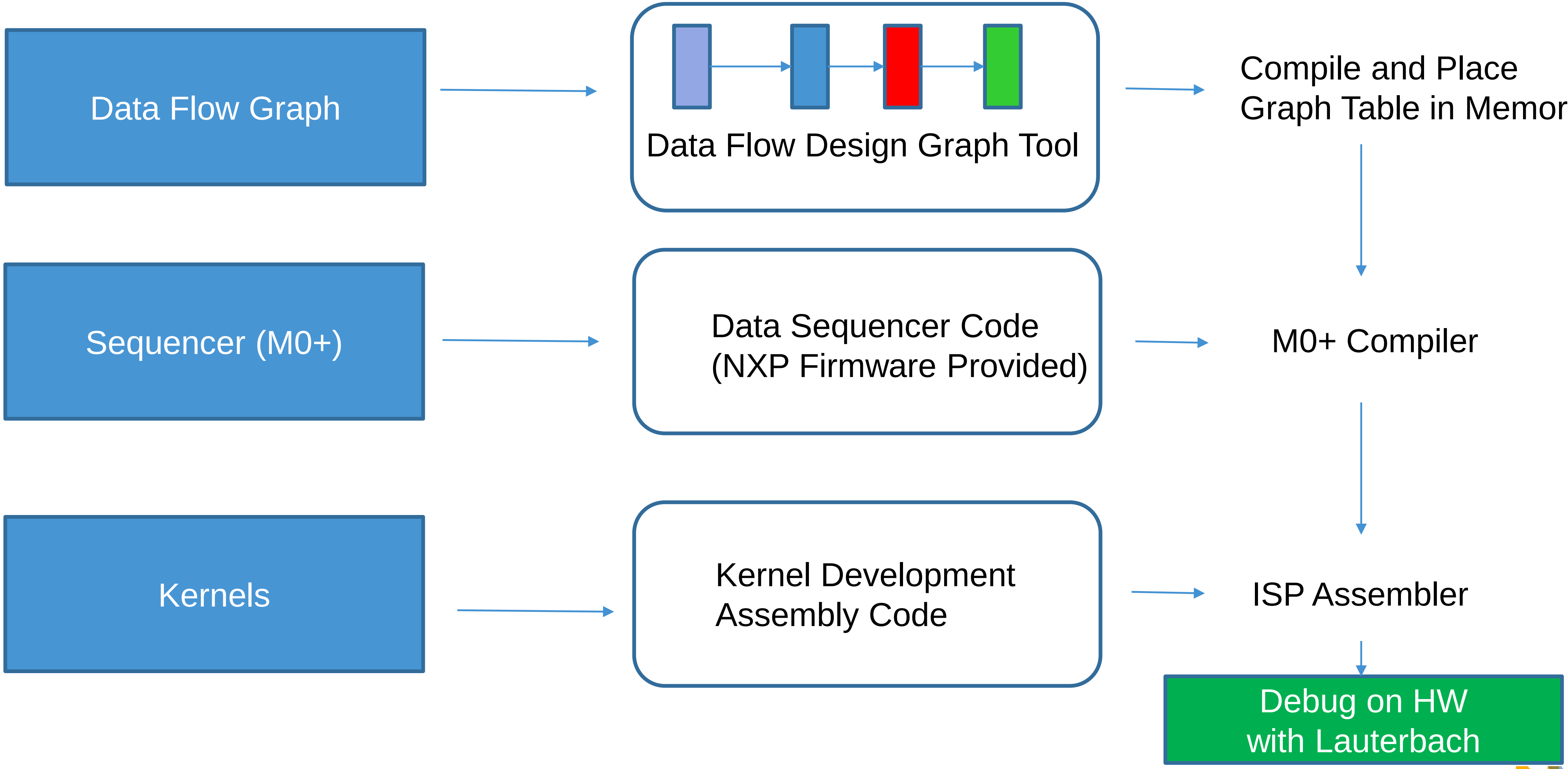
- New Project Wizard to create starter project
  - ❑ including all libraries and SW required
- High Level Graph tools for algorithm formation
  - ❑ drag and drop creation of algorithm graphs
  - ❑ Graph == Mathematical with Visual Interface
- Vision SDK of Kernels for Vision processing
- Assembler for building executable at a button press
- Debugging Algorithm on ISP Hardware
- Example projects of Vision Processing on ISP



MIPISIO Node: MIPI_SONY
Engine Index: 0

Name: MIPI_SONY
Engine Type: MIPICSI0
Camera Type: SONY224
Attached Cameras: 1
Nr Input Line Images: NUM_LINES

# ISP Software Development Workflow



Data Flow Graph

Sequencer (M0+)

Kernels

Data Flow Design Graph Tool

Data Sequencer Code (NXP Firmware Provided)

Kernel Development Assembly Code

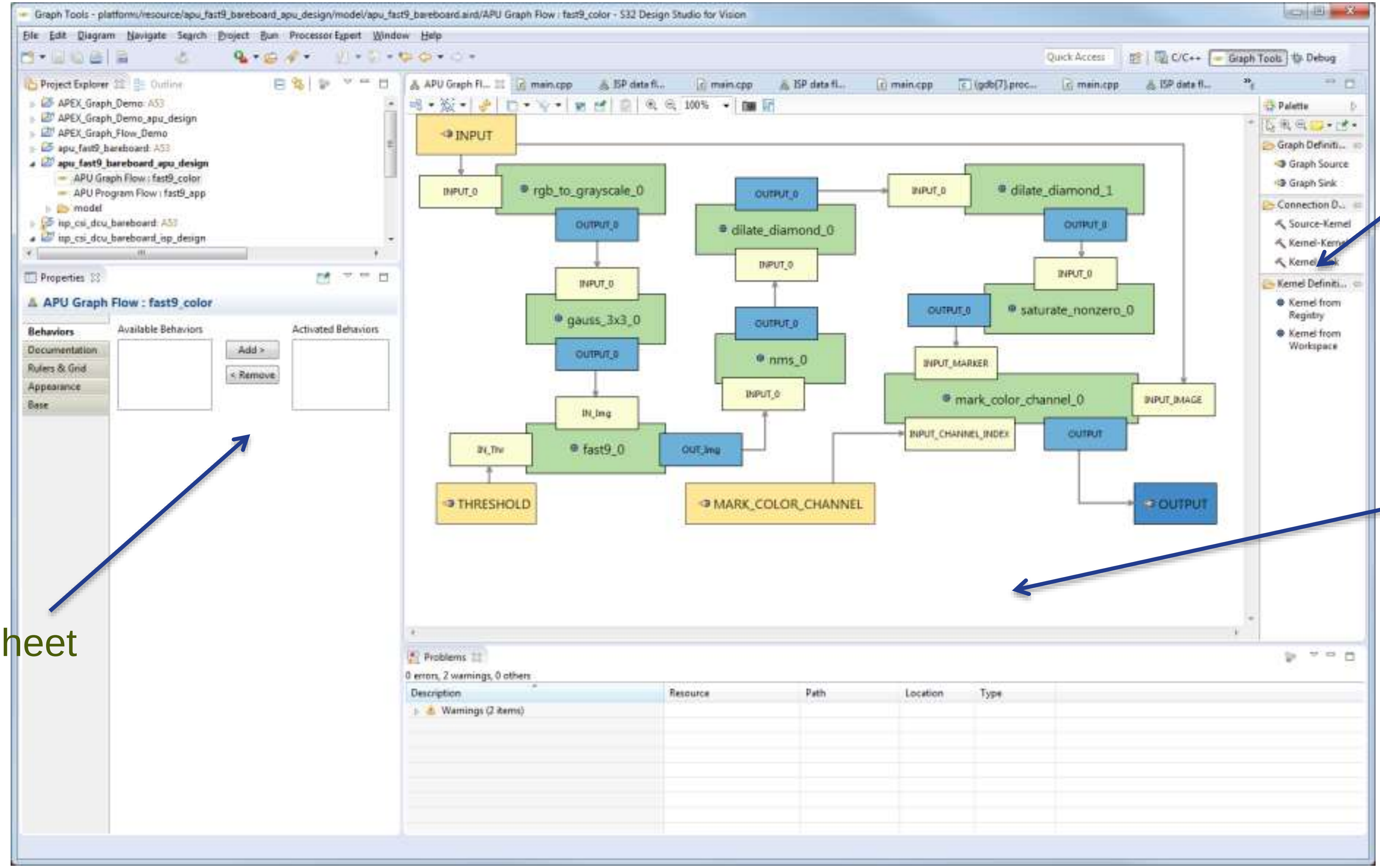
Compile and Place Graph Table in Memory

M0+ Compiler

ISP Assembler

Debug on HW with Lauterbach

# APEX Graph Tool

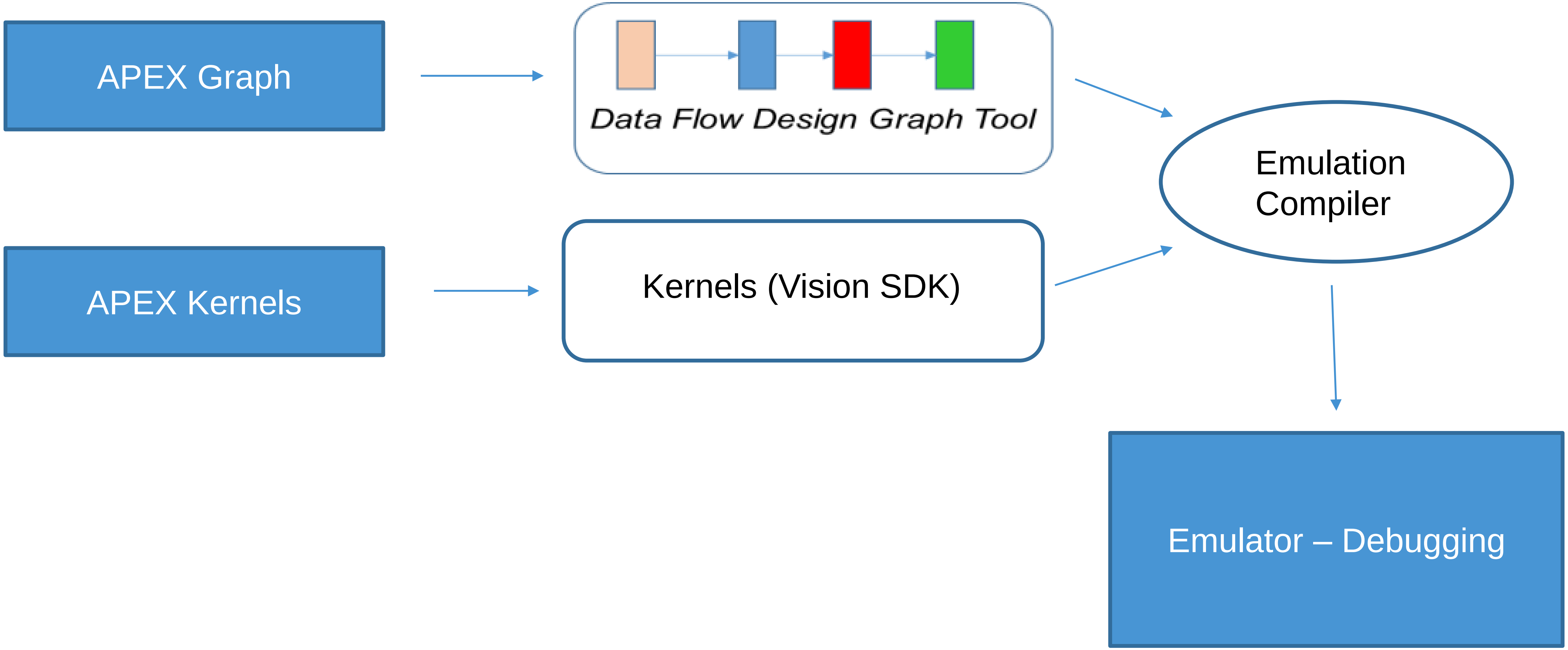


Tools Palette

Canvas

Property Sheet

# APEX Software Development Workflow to Emulator



APEX Graph

Data Flow Design Graph Tool

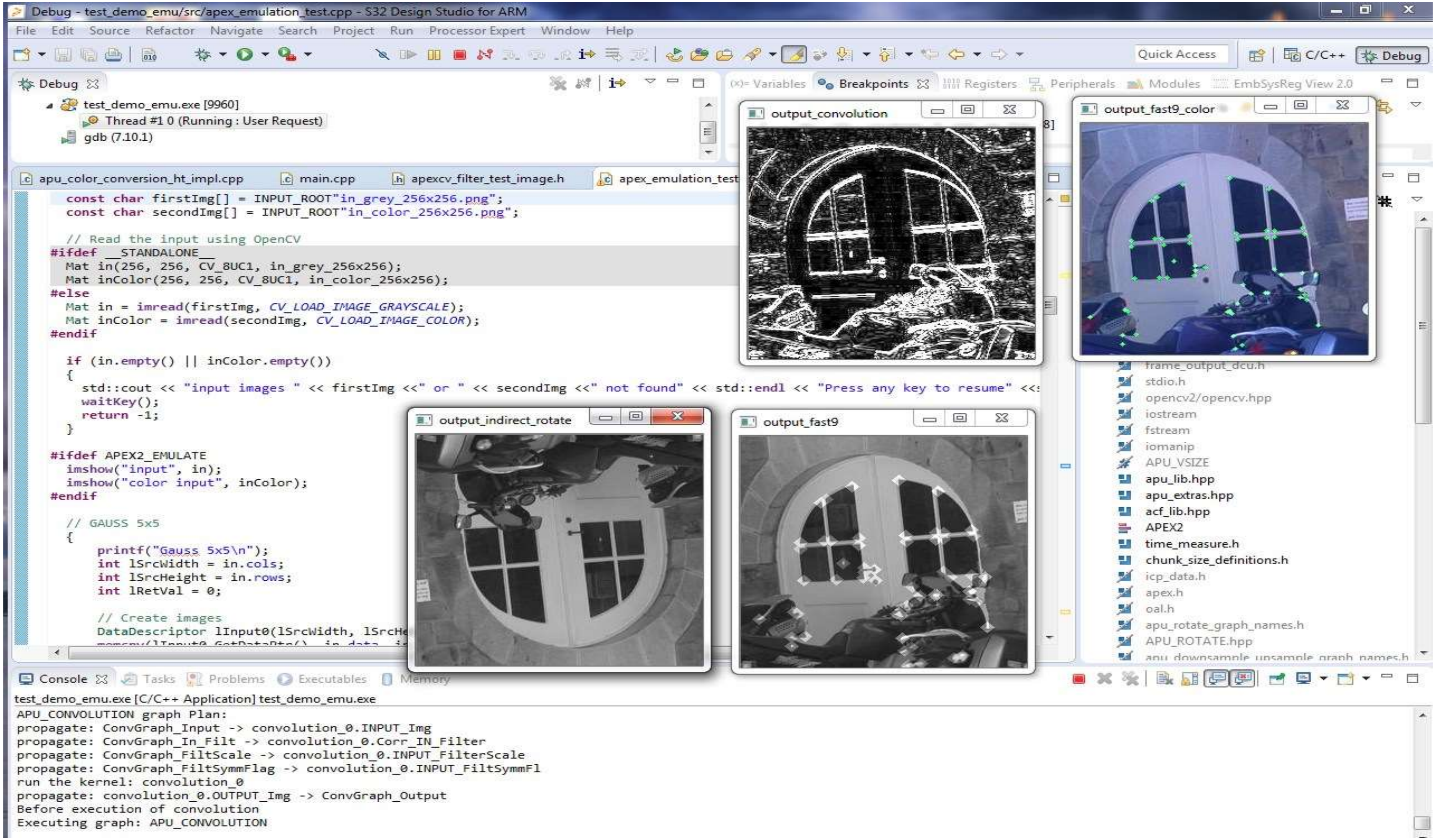
APEX Kernels

Kernels (Vision SDK)

Emulation Compiler

Emulator – Debugging

# APEX Emulator Debugging – in S32DS for Vision



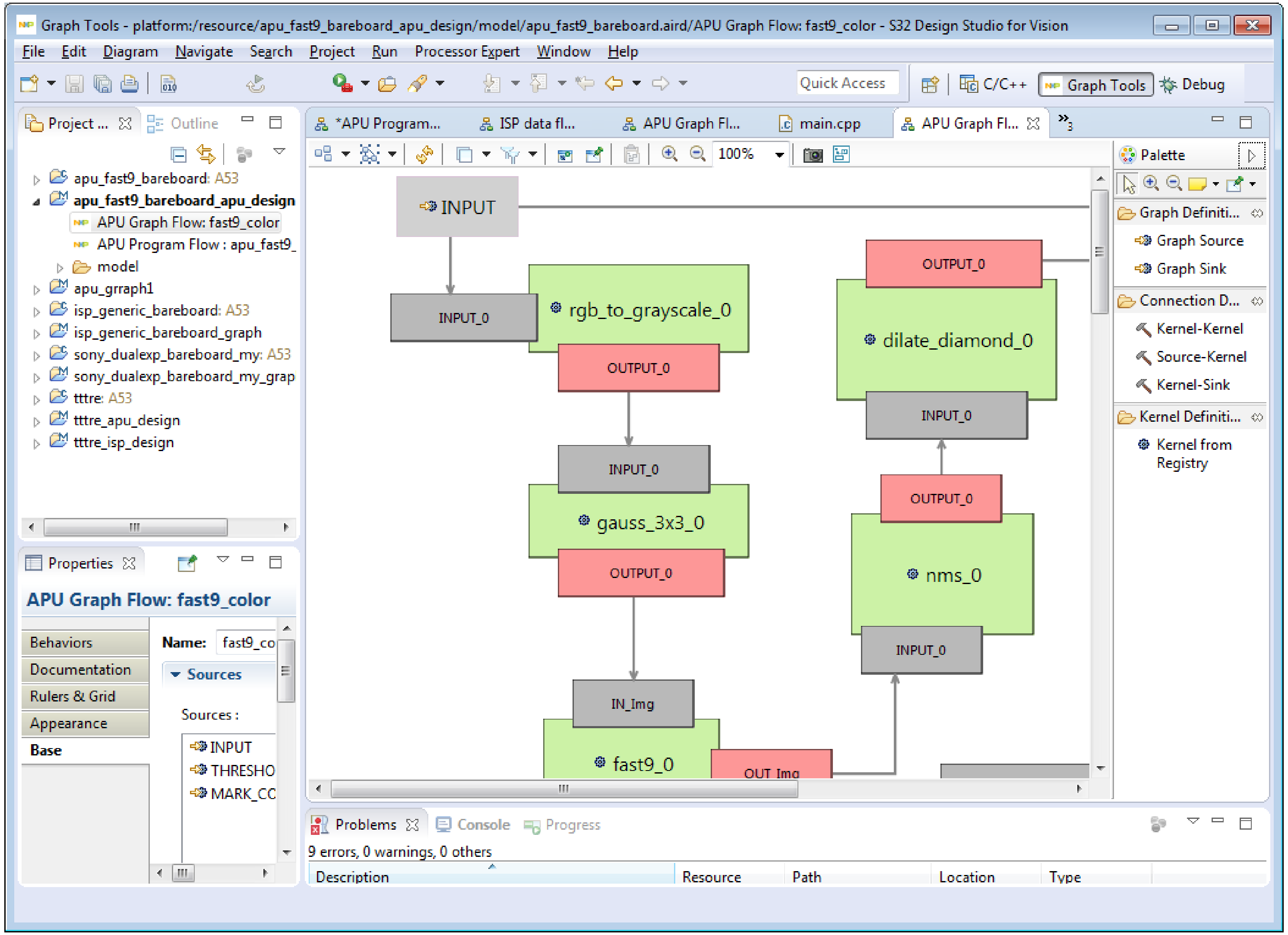
```
  const char firstImg[] = INPUT_ROOT"in_grey_256x256.png";
  const char secondImg[] = INPUT_ROOT"in_color_256x256.png";

  // Read the input using OpenCV
#ifdef __STANDALONE__
  Mat in(256, 256, CV_8UC1, in_grey_256x256);
  Mat inColor(256, 256, CV_8UC1, in_color_256x256);
#else
  Mat in = imread(firstImg, CV_LOAD_IMAGE_GRAYSCALE);
  Mat inColor = imread(secondImg, CV_LOAD_IMAGE_COLOR);
#endif

  if (in.empty() || inColor.empty())
  {
   std::cout << "input images " << firstImg <<" or " << secondImg <<" not found" << std::endl << "Press any key to resume" <<
   waitKey();
   return -1;
  }

#ifdef APEX2_EMULATE
  imshow("input", in);
  imshow("color input", inColor);
#endif

  // GAUSS 5x5
  {
      printf("Gauss 5x5\n");
      int lSrcWidth = in.cols;
      int lSrcHeight = in.rows;
      int lRetVal = 0;

      // Create images
      DataDescriptor lInput0(lSrcWidth, lSrcHe
```

test_demo_emu.exe [C/C++ Application] test_demo_emu.exe

```
APU_CONVOLUTION graph Plan:
propagate: ConvGraph_Input -> convolution_0.INPUT_Img
propagate: ConvGraph_In_Filt -> convolution_0.Corr_IN_Filter
propagate: ConvGraph_FiltScale -> convolution_0.INPUT_FilterScale
propagate: ConvGraph_FiltSymmFlag -> convolution_0.INPUT_FiltSymmFl
run the kernel: convolution_0
propagate: convolution_0.OUTPUT_Img -> ConvGraph_Output
Before execution of convolution
Executing graph: APU_CONVOLUTION
```

# Graph Tools – Example : Vision Algorithm Development on APEX

- New Project Wizard to create starter project
  - ❑ including all libraries and SW required
- High Level Graph tools for algorithm formation
  - ❑ drag and drop creation of algorithm graphs
  - ❑ Graph == Mathematical with Visual Interface
- Vision SDK of Kernels and ACF for Vision processing development on APEX
- Compiler for building executable at a button press
- Debugging Algorithm on Emulator without Hardware
- Debugging Algorithm on APEX Hardware
- Example projects of Vision Processing on APEX
- 70+ Vision Kernels shipping with tools



Graph Tools - platform:/resource/apu_fast9_bareboard_apu_design/model/apu_fast9_bareboard.aird/APU Graph Flow: fast9_color - S32 Design Studio for Vision

APU Graph Flow: fast9_color

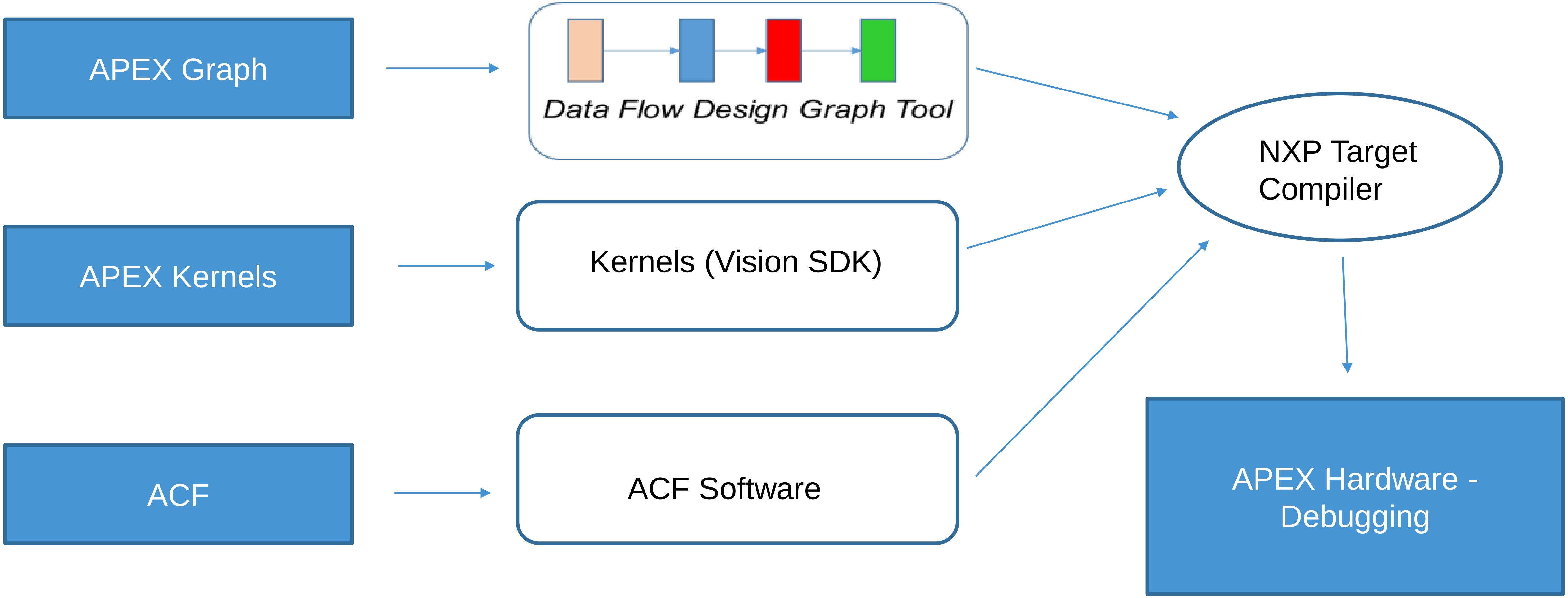
Name: fast9_co

Sources

Sources :
INPUT
THRESHO
MARK_CC

9 errors, 0 warnings, 0 others

# APEX Software Development Workflow to APEX



# Tools Functional Safety Support – ISO 26262 Qualification Kit

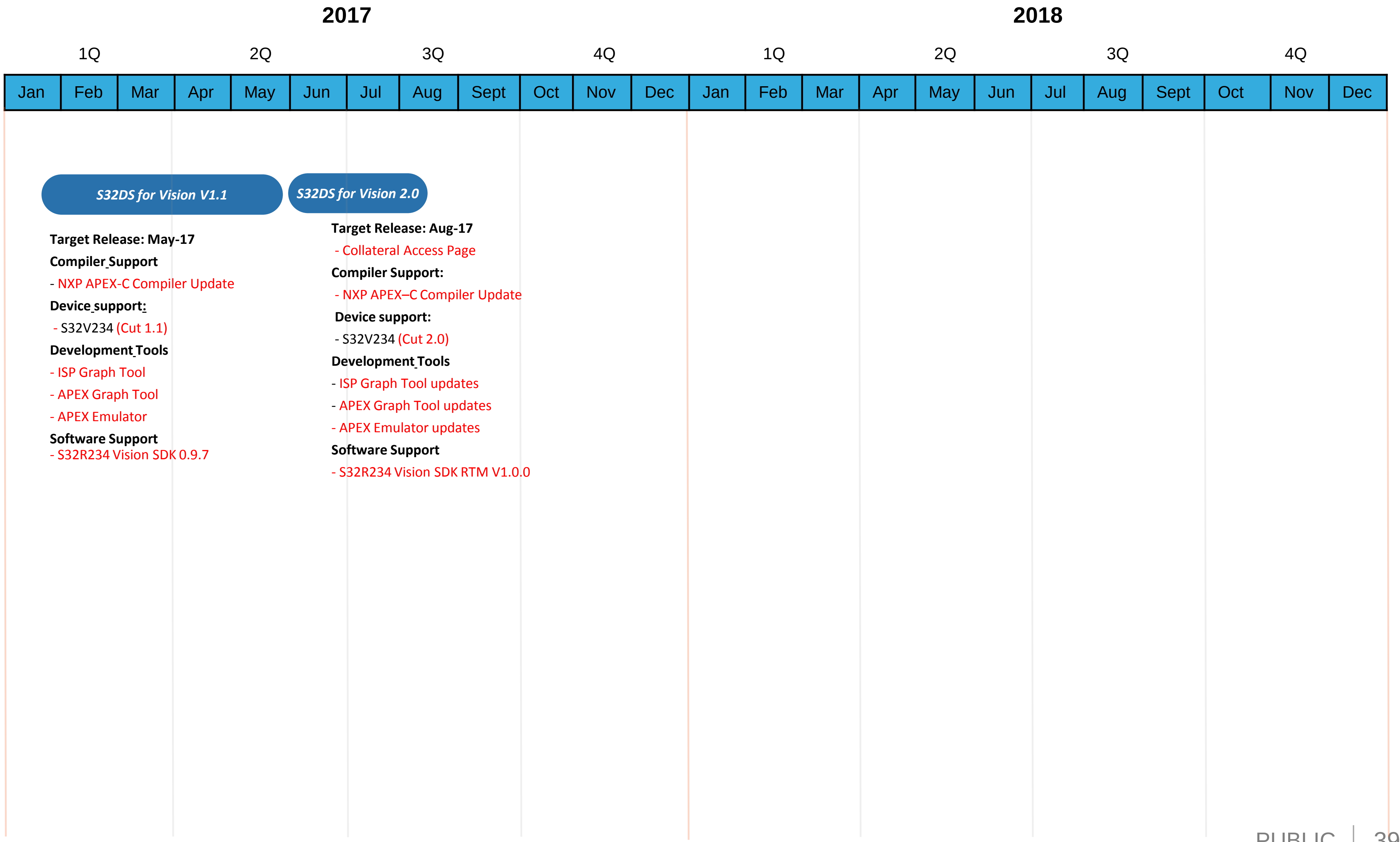
ISO 26262 requires qualification evidence for NXP tools.

Our strategy

- For generic core support (ARM compiler) rely on partners – GHS (provide either ISO26262 Certified or an ISO26262 Qualification Kit ).
- For NXP specific tooling (Acceleration support) develop Qualification Kits (QKit) supporting ISO26262 development.
- Provide Qualification Kit for use with ISP and APEX Graph tools as part of S32DS toolchain this includes an NXP APEX Compiler qualification kit.
- Qualification Kits provides significant flexibility for customers and NXP.

| PL | IP | ASIL | Strategy |
|---|---|---|---|
| ADAS | Programmable Accelerators | B-D | Provide ISO26262 tool QKits<br>- Use Case Modeling tool<br>- Safety Manual<br>- Tool Qualification Documentation Generator<br>- Consulting Services Support the Qkit<br>- Qualification Test Suite |
| ALL | ARM | D | Partners to provide QKits or Certification for ARM Cores |

# S32 Design Studio for Vision Roadmap 2017 – 2018



| 2017 | | | | | | | | | | | | 2018 | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1Q | | | 2Q | | | 3Q | | | 4Q | | | 1Q | | | 2Q | | | 3Q | | | 4Q | | |
| Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sept | Oct | Nov | Dec | Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sept | Oct | Nov | Dec |

***S32DS for Vision V1.1***

**Target Release: May-17**

**Compiler Support**

- NXP APEX-C Compiler Update

**Device support:**

- S32V234 (Cut 1.1)

**Development Tools**

- ISP Graph Tool
- APEX Graph Tool
- APEX Emulator

**Software Support**

- S32R234 Vision SDK 0.9.7

***S32DS for Vision 2.0***

**Target Release: Aug-17**

- Collateral Access Page

**Compiler Support:**

- NXP APEX–C Compiler Update

**Device support:**

- S32V234 (Cut 2.0)

**Development Tools**

- ISP Graph Tool updates
- APEX Graph Tool updates
- APEX Emulator updates

**Software Support**

- S32R234 Vision SDK RTM V1.0.0

**Common features/Tools support:**

- Compiler Support
  - GCC (M0+,M4,A53)
  - NXP APEX-C Compiler
- Device support:
  - S32V234
- Integrated Debugger Support
  - Lauterbach (A53,M4,ISP,APEX)
  - P&E (A53,M4)
  - GDB (Linux, Apex Emulation)
- Software Integration
  - Vision SDK
  - (63 Scalar, 21 Vector) ISP Kernels
  - (70) APEX Vision Kernels
  - APEX OpenCV Libraries
  - APEX OpenCV Pro Libraries
  - A53 Linux BSP
- Miscellaneous Features
  - Collateral Access Page
  - SDK Browser
  - ISP Graph Tool
  - APEX Graph Tool
  - DDR Configuration Tool

# 05.

## In Summary

# What does the S32 Design Studio Provide

## Tools at No Cost …

- Single integrated development environment
- Standard GCC Compilers provided at no cost
- Low Cost Debugger available for all products
- Premium Debuggers (best in Class) and Premium Compilers (best in Class) are Seamless Plug-ins to the toolchain.
- NXP Supporting tools integrated, Device Configuration, Pin I/O Configuration Tool, FreeMASTER real-time monitoring and tuning tool

## Software at No Cost …

- Operating Systems provided at no cost, Linux, FreeRTOS, and MQX all integrated into the tool chain
- Software Development Kits available, integrated into the tool chain
- Tool Interface Drivers integrated into the tool chain
- Application Software Development Kit with Operating System and Application Specific functions.
- Tools for Graphical development of Vision processing algorithms
- Get more value-add from NXP with more tools and software

SECURE CONNECTIONS
FOR A SMARTER WORLD