Hands-On Workshop:
Developing a Secure Application Using TrustZone and MCUXpresso Software and Tools

Brendon Slade

Director, MCU ecosystems

June 2019 | Session #AMF-SOL-T3636

SECURE CONNECTIONS FOR A SMARTER WORLD

Agenda

- MCUXpresso SW and Tools Overview
  - MCUXpresso IDE
  - MCUXpresso SDK
  - MCUXpresso Config Tool
- Concepts of TrustZone on Armv8-M
- MCUXpresso Tools Supporting TrustZone
  - MCUXpresso Config Tools – TEE
  - MCUXpresso IDE – Project Settings
- Overview of the LPC5500 Series
- Hands-On Lab

What you should get from this class

- Understanding of the MCUXpresso suite of tools and software are and what they can do to help in product development
- Understanding of the new Trusted Execution Environment configuration tool – what it does and how it helps
- Hands-on experience of building, debugging and developing an application with secure and non-secure elements

MCUXpresso Software and Tools

for LPC & Kinetis MCUs and i.MX RT crossover processors

MCUXpresso IDE

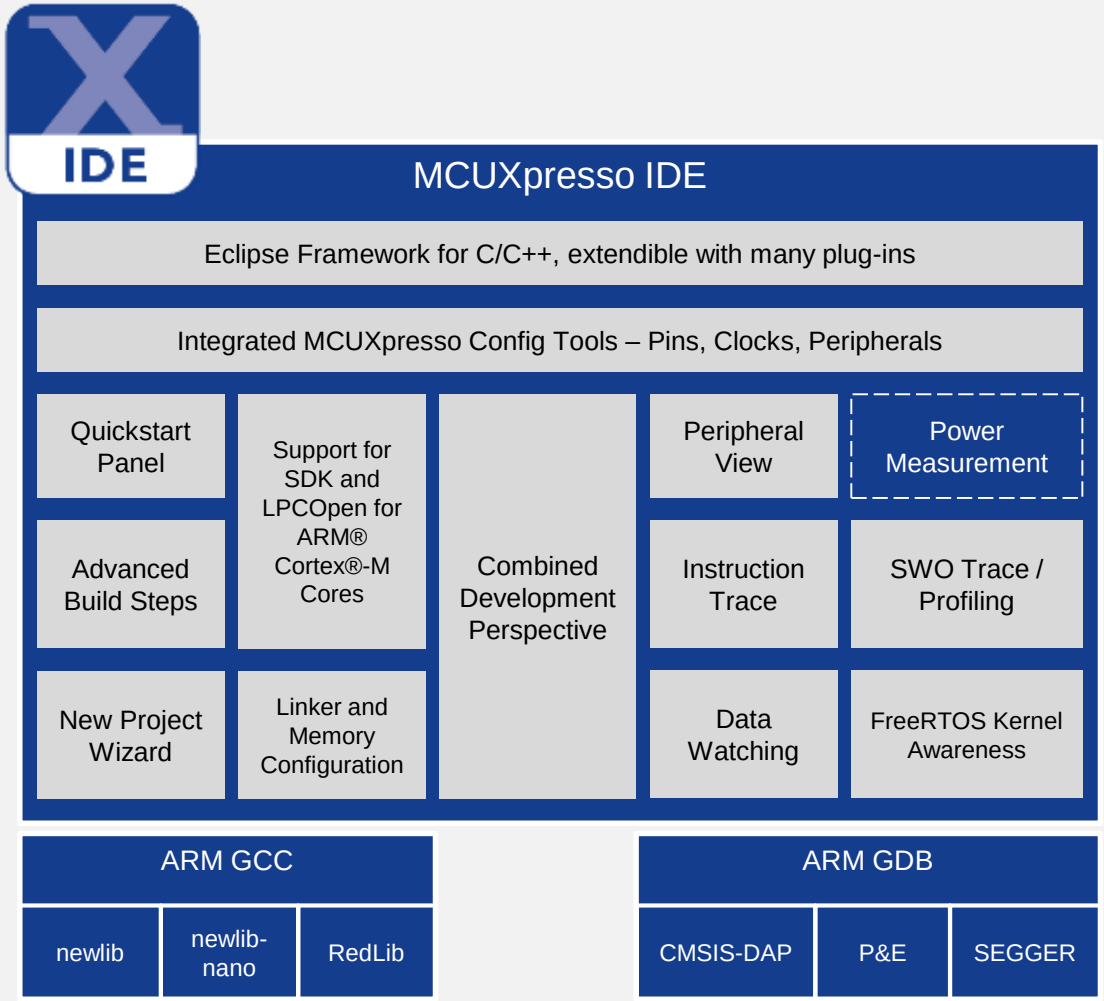
Edit, compile, debug and optimize in an intuitive and powerful IDE

MCUXpresso SDK

Runtime software including peripheral drivers, middleware, RTOS, demos and more

MCUXpresso Config Tools

Online and desktop tool suite for system configuration and optimization

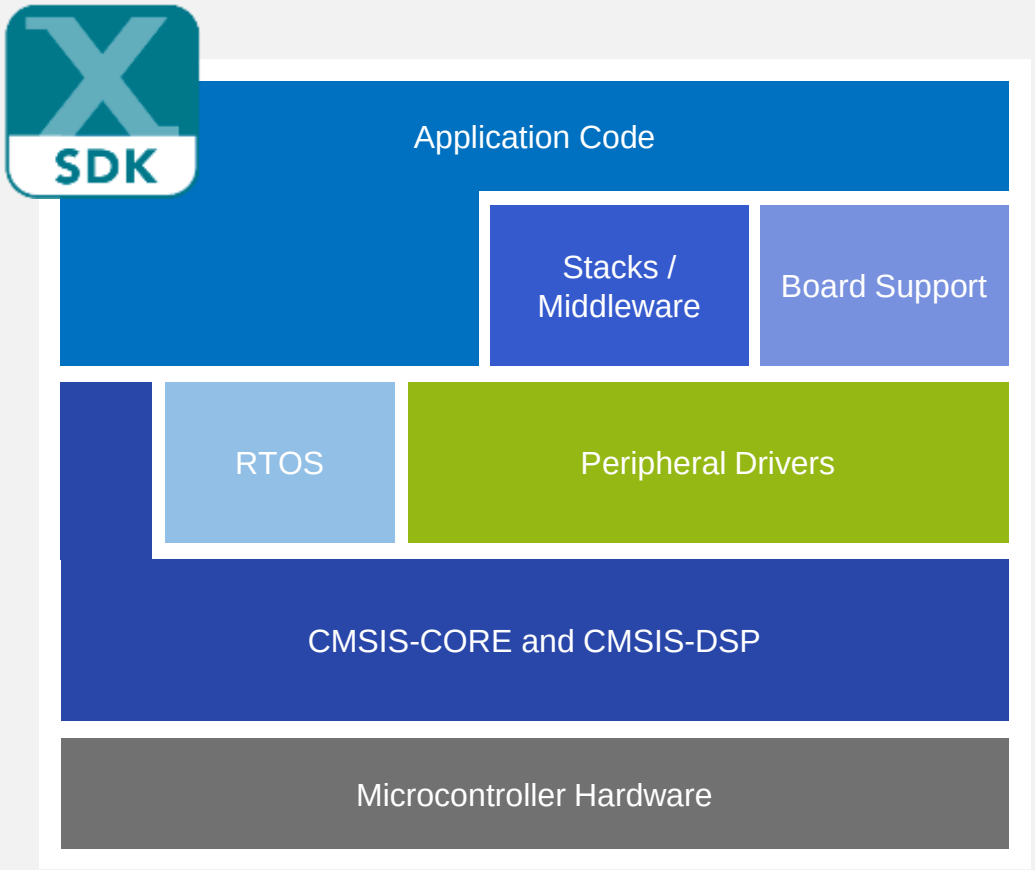


MCUXpresso IDE

Free Eclipse / GCC-based Development

- Feature-rich, unlimited code size, optimized for ease-of-use, based on industry standard Eclipse framework for NXP's Kinetis and LPC MCUs and i.MX RT crossover processors
- Application development with Eclipse and GCC-based IDE for advanced editing, compiling and debugging
- Supports custom development boards, Freedom, Tower and LPCXpresso boards, and i.MX RT evaluation kits with debug probes from NXP, P&E and Segger
- Free: Full Featured, unlimited Code Size, no special activation needed, community based support, advanced trace capabilities, MTB and ETB instruction trace
- Works in conjunction with MCUXpresso Config Tools and MCUXpresso SDK to provide complete development environment

Learn more at: www.nxp.com/mcuxpresso/sdk



MCUXpresso SDK

Software Framework and Drivers

Architecture:

- CMSIS-CORE compatible
- Single driver for each peripheral
- Transactional APIs w/ optional DMA support for communication peripherals

Integrated RTOS:

- Amazon FreeRTOS
- RTOS-native driver wrappers

Integrated Stacks and Middleware:

- USB Host, Device and OTG
- lwIP, FatFS, LittleFS
- Crypto acceleration plus wolfSSL & mbedTLS
- AWS IoT and Azure IoT
- SD and eMMC card support

Reference Software:

- Peripheral driver usage examples
- Application demos
- FreeRTOS usage demos
- IoT connectivity examples

License:

- BSD 3-clause for startup, drivers, USB stack

Toolchains:

- MCUXpresso IDE
- IAR®, ARM® Keil®, GCC w/ Cmake

Quality:

- Production-grade software
- MISRA 2004 compliance
- Checked with Coverity® static analysis tools

Learn more at: www.nxp.com/mcuxpresso/config

MCUXpresso Config Tools

Configuration and Code Generation

SDK Builder packages custom SDKs based on user selections of MCU, evaluation board, and optional software components.

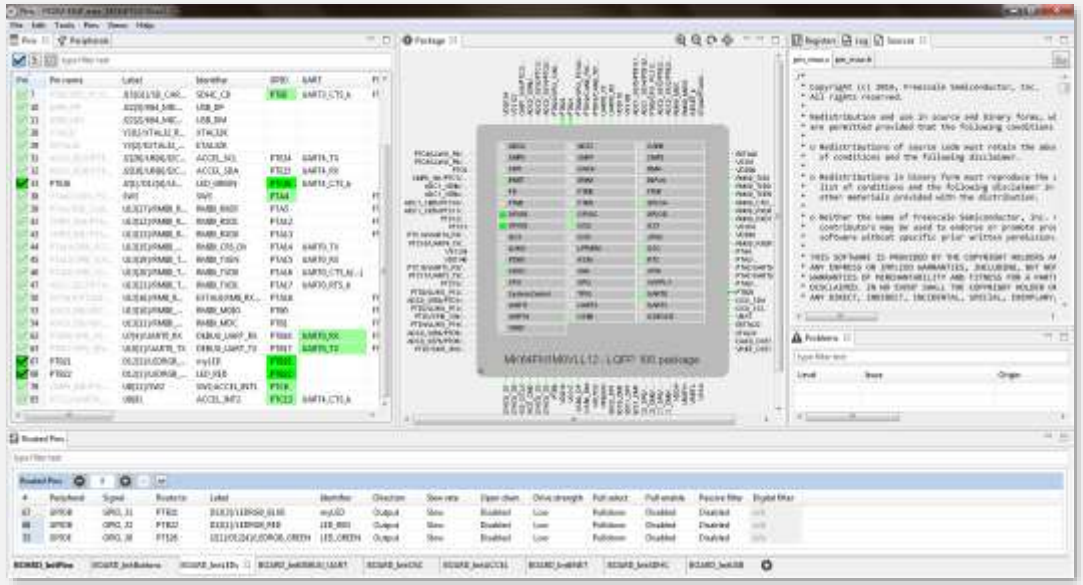
Pins, Clocks, Peripherals and Cloner tools generate initialization for custom board support; cloner creates standalone SDK project based on SDK examples.

Project updater works directly with existing SDK-based IDE projects with generated Pins, Clocks and Peripherals source files.

Device Configuration tool allows DCD commands sequence config for pre-initialization of devices at boot time.

Trusted Execution Environment configures protection and isolation of sensitive parts of the application.

Easy-to-use muxing and pin assignments

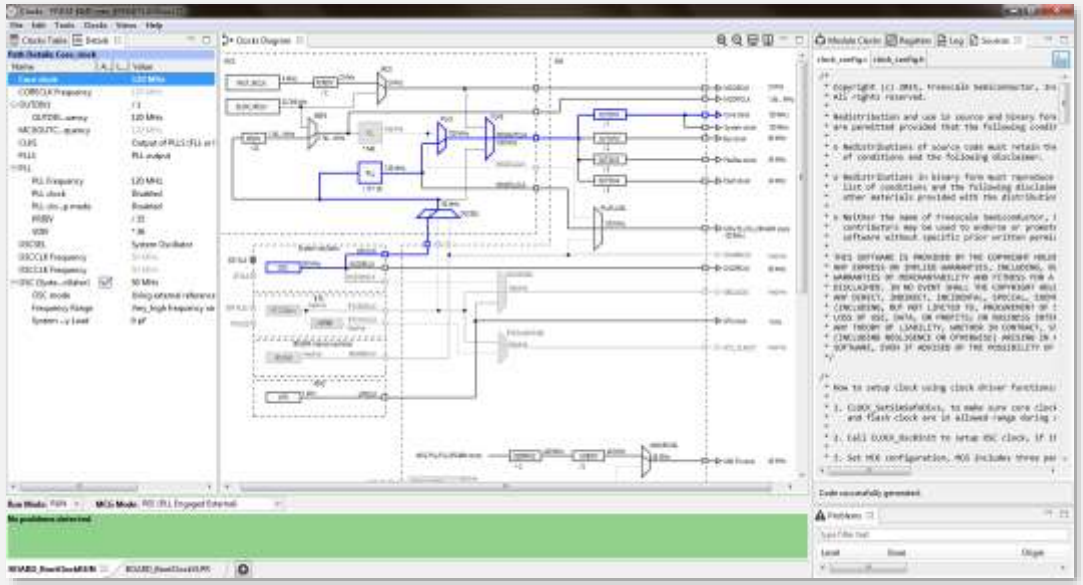


MCUXpresso Config Tools

Pins Configuration

- Muxing and pin configuration with consistency checking
- ANSI-C configuration code
- Graphical processor package view
- Wizard for optimized assignments of functionality to available pins
  - Selection of Pins and Peripherals
  - Routed pins with electrical characteristics
  - Registers with configured and reset values
  - Source code for C/C++ applications
  - GPIO Input / Output initialization
- Documented and easy to understand source code
- Report generation
- Integrates with any compiler and IDE

Clock configuration and diagram view

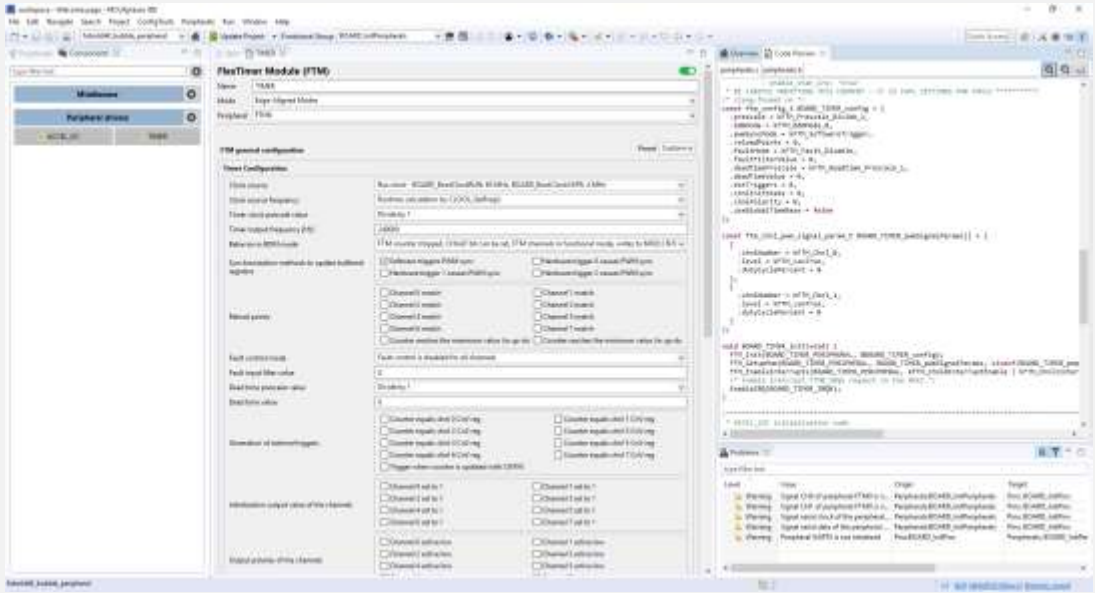


MCUXpresso Config Tools

Clock Configuration

- System clock configuration with consistency checking
- ANSI-C initialization code
- Graphical clock diagrams
- Easy-to-use guided graphical user interface
  - Selection of Clock Sources
  - Configuration of prescalers and clock outputs
  - Details and Full Diagram views with clock path
  - Registers with configured and reset values
  - Source code for C/C++ applications
- Documented and easy to understand source code
- Report generation

Automatic generation of peripheral initialization structures

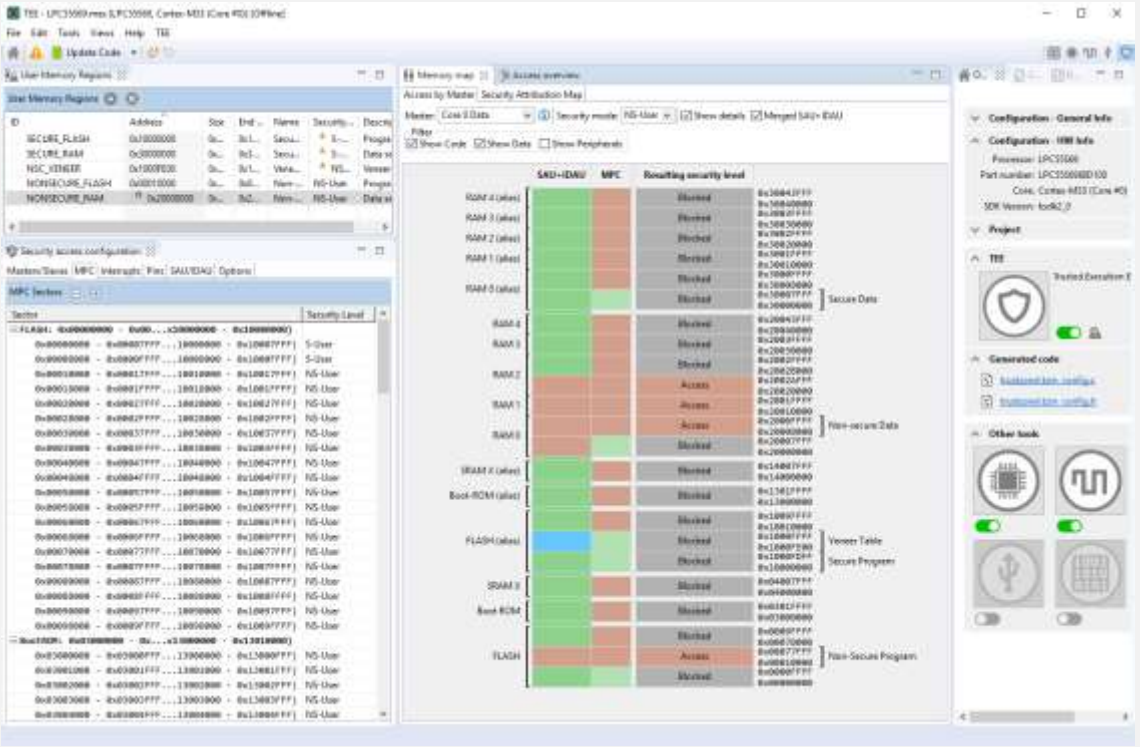


MCUXpresso Config Tools

Peripheral Configuration

- SDK peripheral configuration and USB middleware configuration
- Validation of user inputs / selection
- Generation of ANSI-C initialization code for SDK peripheral drivers
- Generation of MCUXpresso SDK Initialization Structure
- Selection of common use case configurations
- Support for over 50 different peripherals and 60 devices
  - Includes GPIO, UART, ADC, LPTMR, I2C, FTM
  - Additional devices and peripherals are continuing to be deployed
- Documented and easy to understand source code
- Report generation

Generation and visualization of Secure / Non-Secure environment partitioning



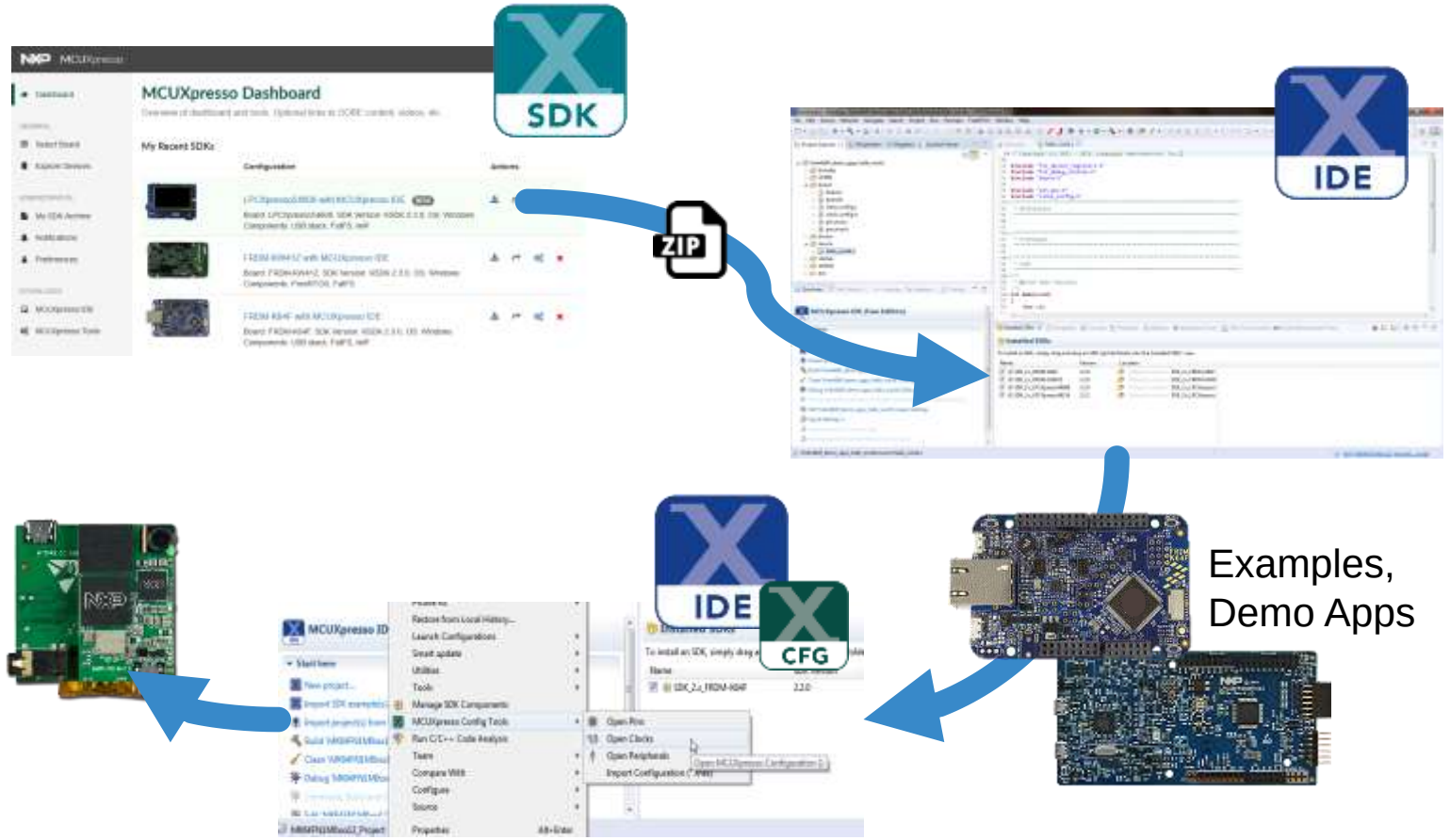
MCUXpresso Config Tools

Trusted Execution Environment

- Support of Cortex M33 TrustZone
- Visualization of memory layout
- Alignment with user created memory regions
- Full visibility of SAU/IDAU and MMC/PCC settings
- Visualization of memory regions and peripherals according to security mode
- Generation of configuration code to configure TrustZone, AHB Secure Controller, and associated device registers
- Documented and easy to understand source code
- Report generation and direct register views

MCUXpresso SW and Tools

Efficient Development Flow



- Online Custom SDK Builder
- Drag-and-Drop installation of SDK into IDE
- SDK Project Importing / Cloning
- Demo applications, SDK driver examples, middleware use case projects
- Management of SDK drivers and middleware components
- Integrated Config Tools
- Pins and Clocks initialization for user defined boards

MCUXpresso Software and Tools

Additional Resources

Webpages

- MCUXpresso Software and Tools – www.nxp.com/mcuxpresso
  - MCUXpresso SDK: www.nxp.com/mcuxpresso/sdk
  - MCUXpresso IDE: www.nxp.com/mcuxpresso/ide
  - MCUXpresso Config Tools: www.nxp.com/mcuxpresso/config

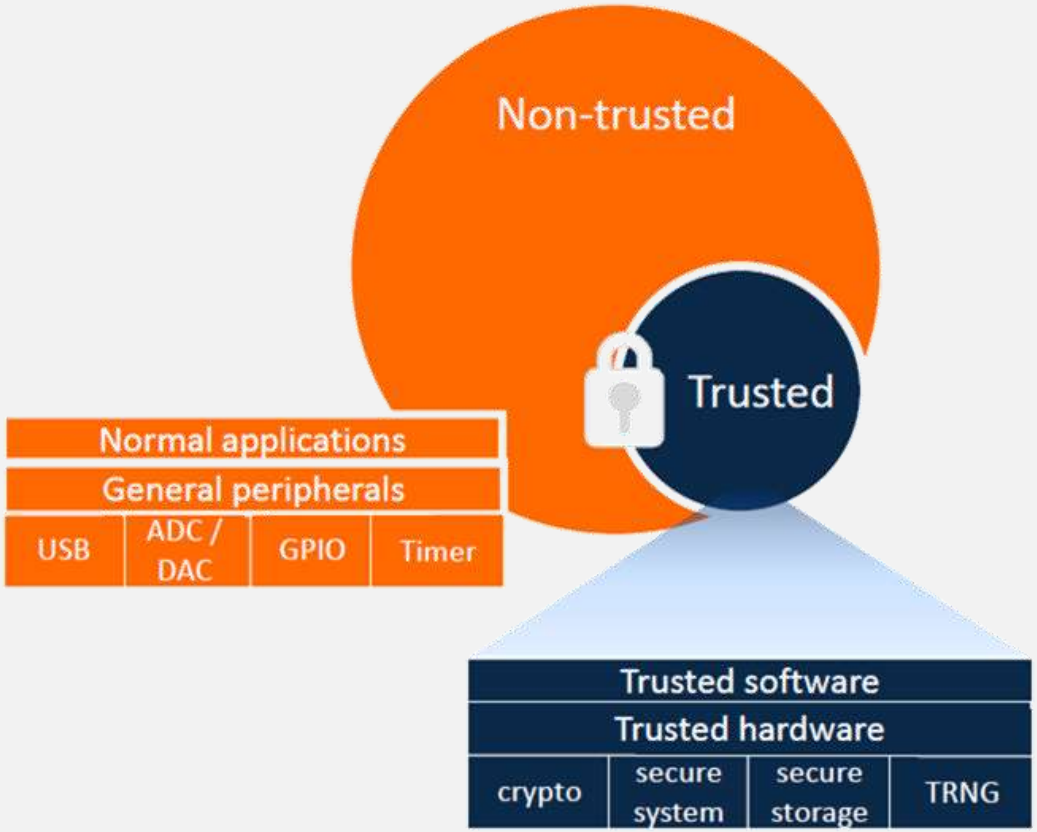
Communities

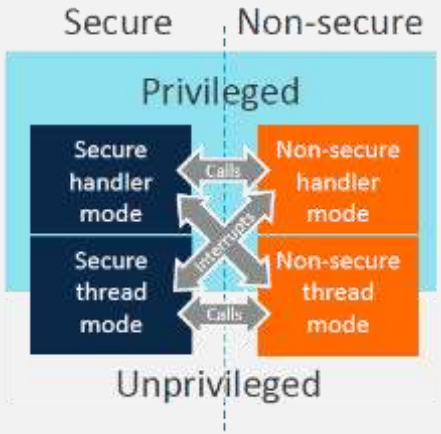
- MCUXpresso Software and Tools – https://community.nxp.com/community/mcuxpresso
  - MCUXpresso SDK: https://community.nxp.com/community/mcuxpresso/mcuxpresso-sdk
  - MCUXpresso IDE: https://community.nxp.com/community/mcuxpresso/mcuxpresso-ide
  - MCUXpresso Config Tools: https://community.nxp.com/community/mcuxpresso/mcuxpresso-config

Supported Devices

- Supported Devices Table (Community Doc)

Overview of TrustZone on Armv8-M



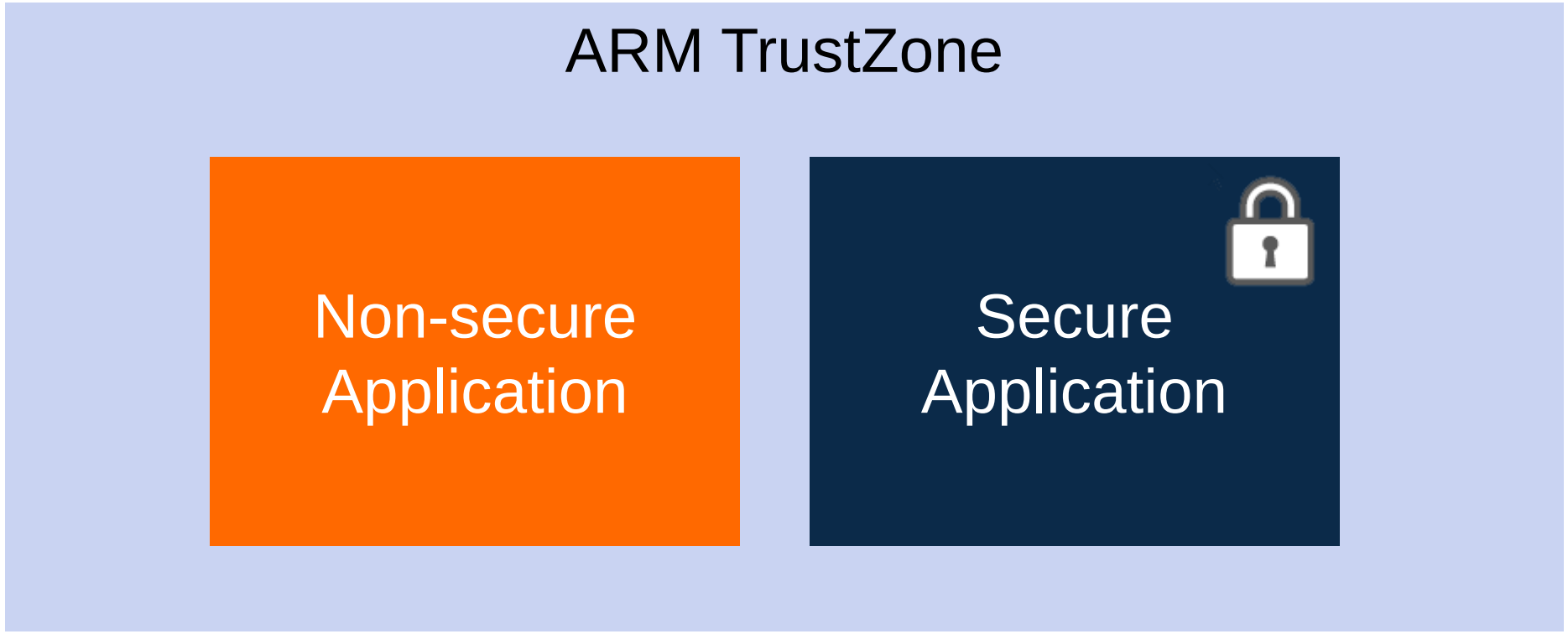


Concepts of TrustZone

- Normal application environment
  - Non-secure world
  - Access to non-secure memories, general peripherals
- Protected application environment
  - Secure world
  - Access to secure and non-secure resources
- Key aspects
  - Non-secure software cannot access to Secure resources
  - Secure software defines what non-secure software can access
  - System-level protection - Security resources are placed in Secure address spaces
  - Secure software provide services (APIs) to Non-secure software

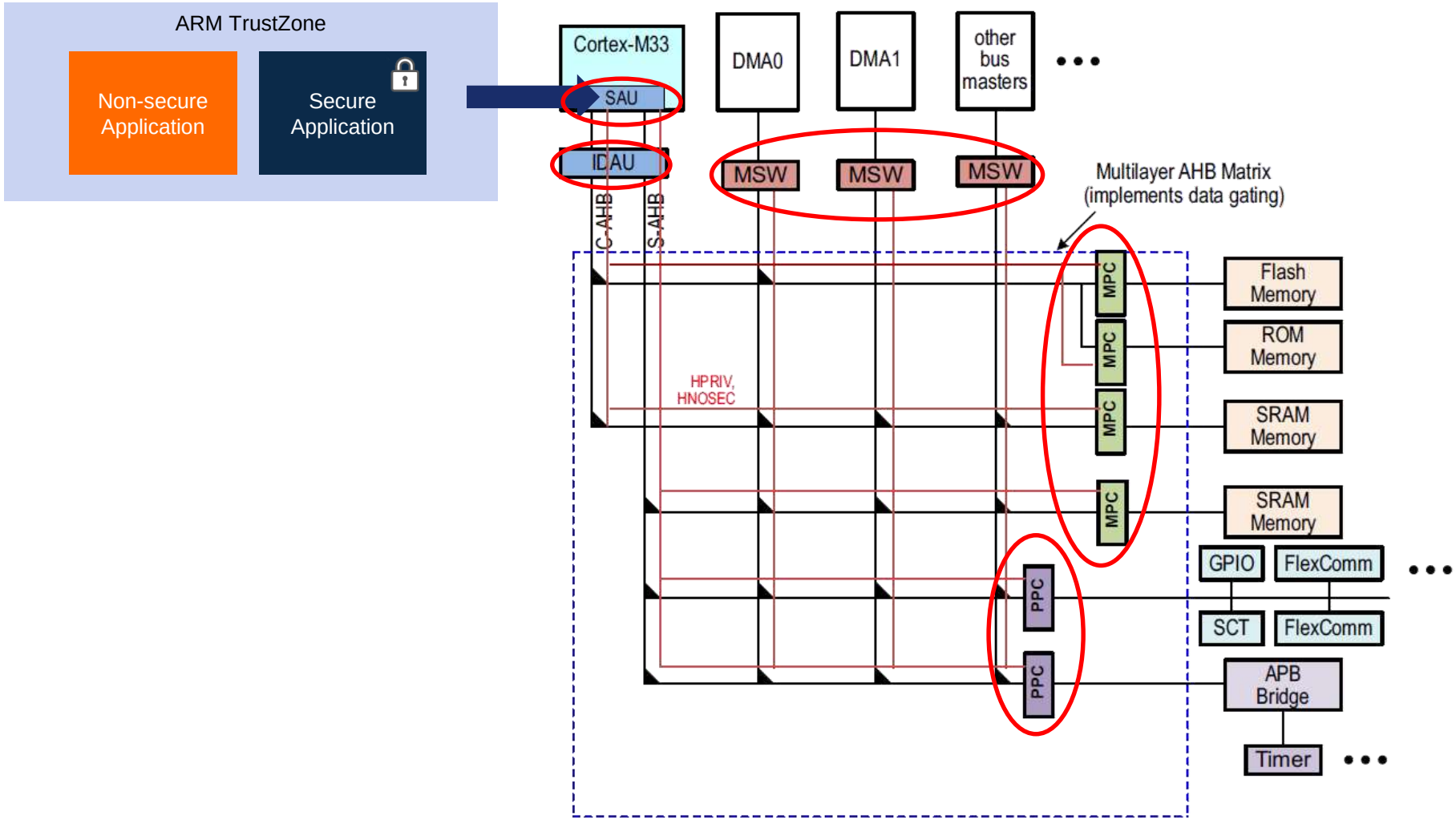
Secure and Non-Secure Application Development

Easy, Right?



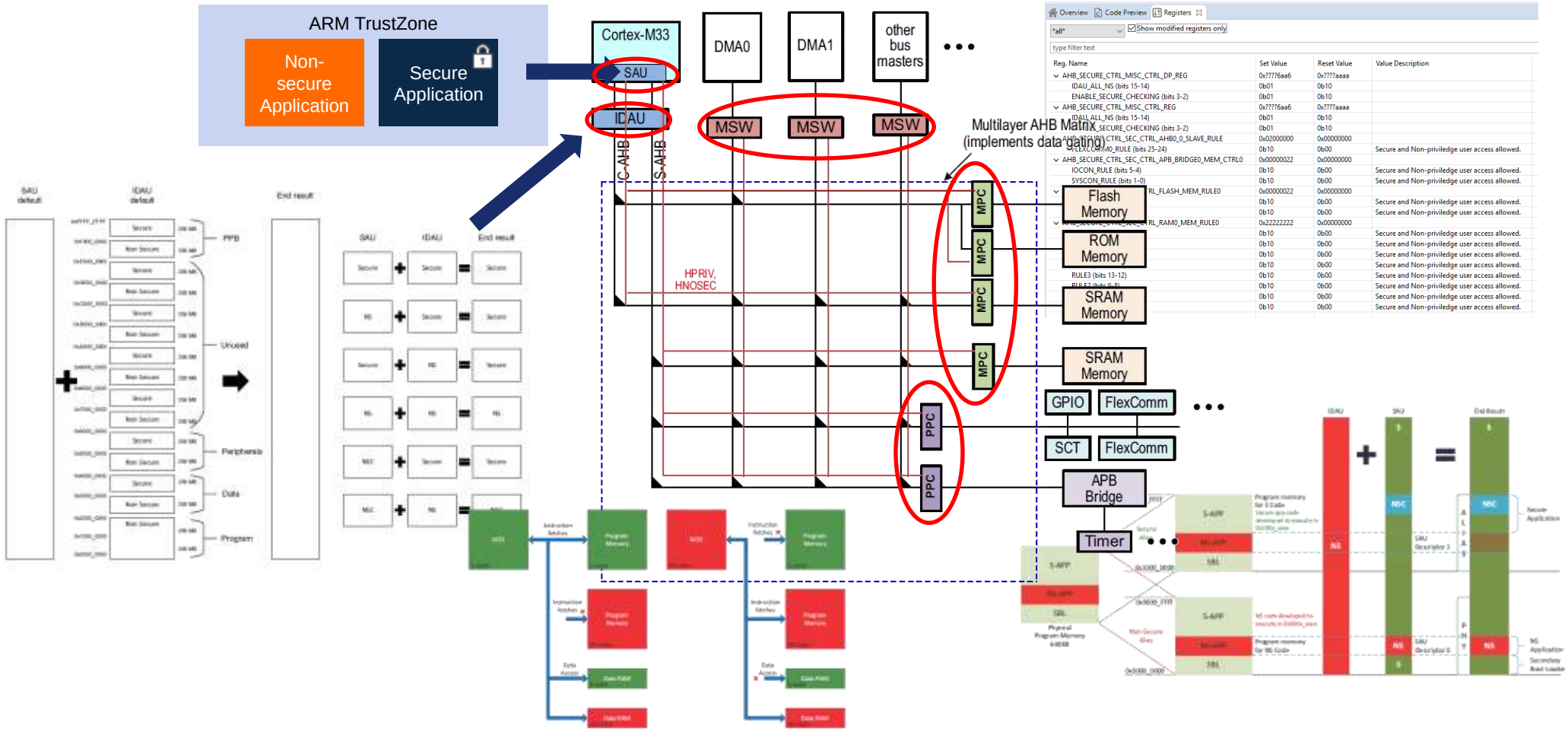
Secure and Non-Secure Application Development

But Wait… There's More



Secure and Non-Secure Application Development

Wait, What?

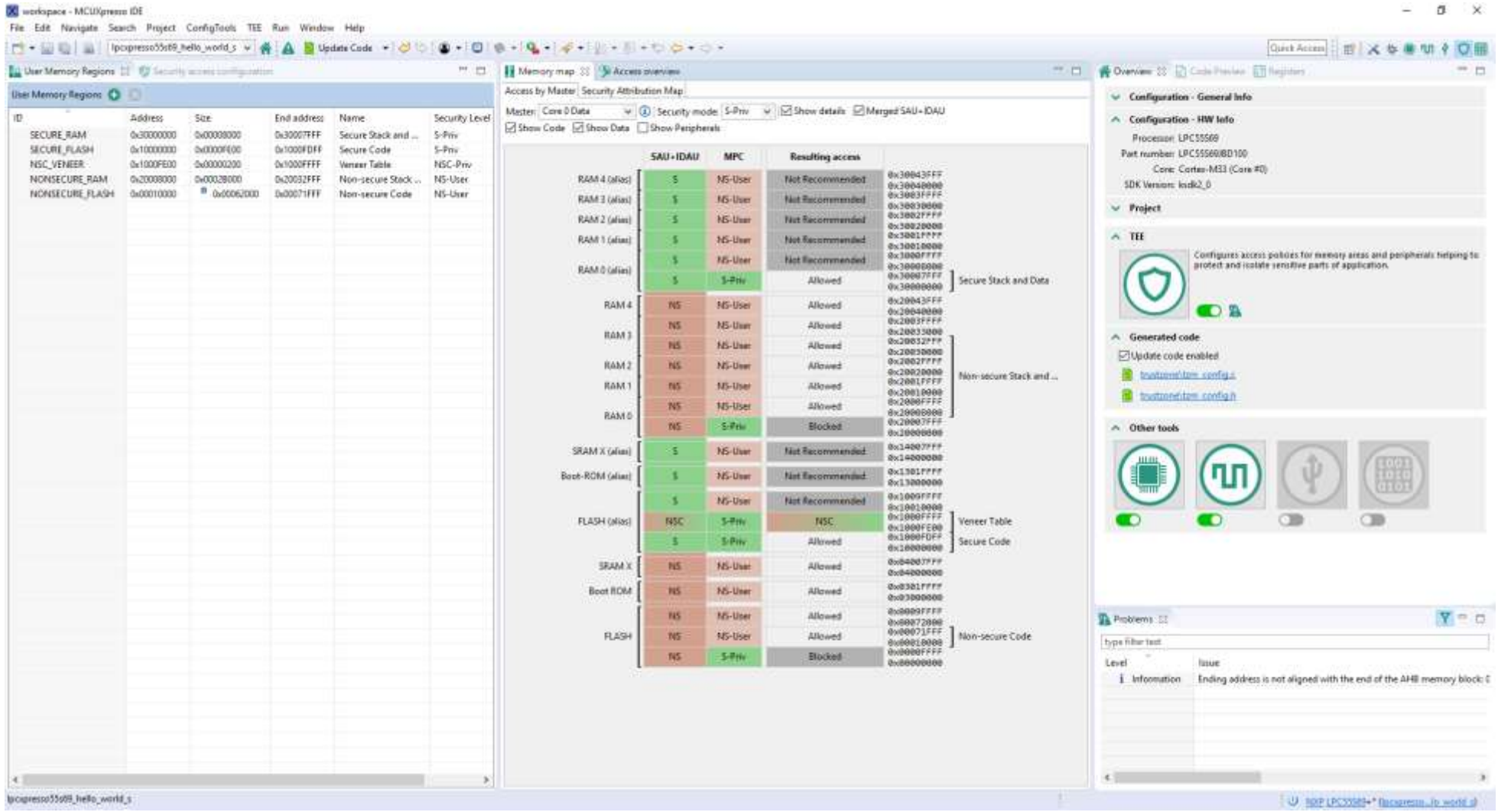


MCUXpresso Config Tools

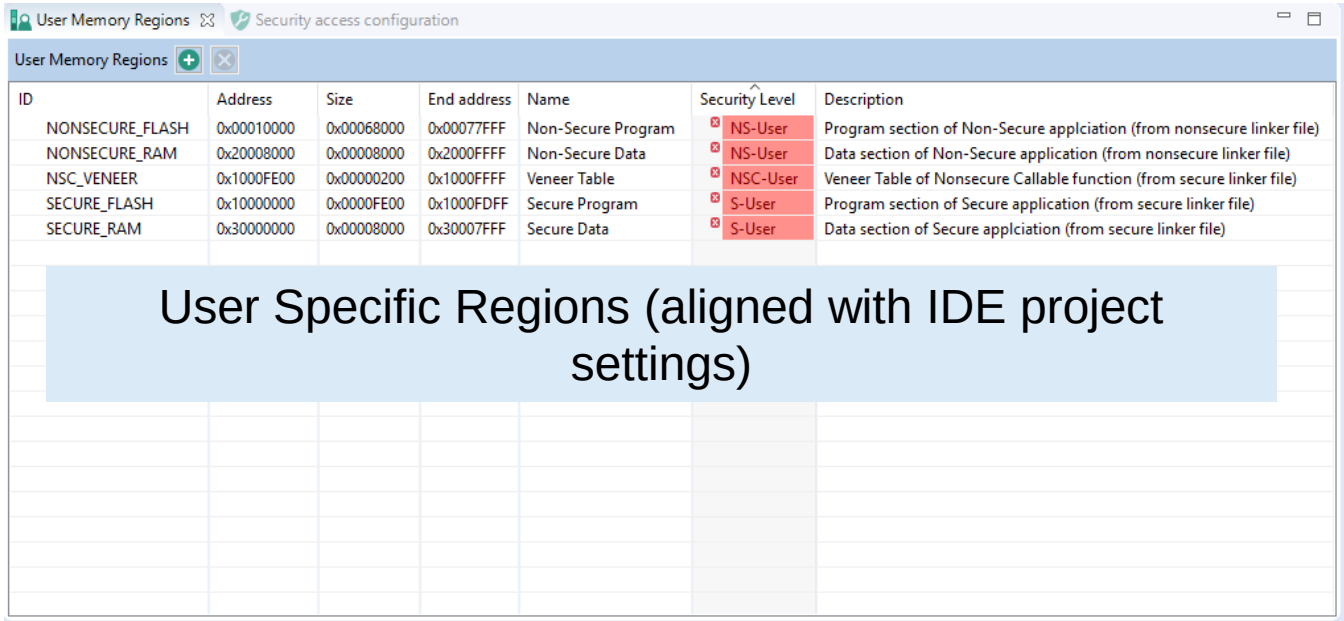
Trusted Execution Environment (TEE)

Configuration UI

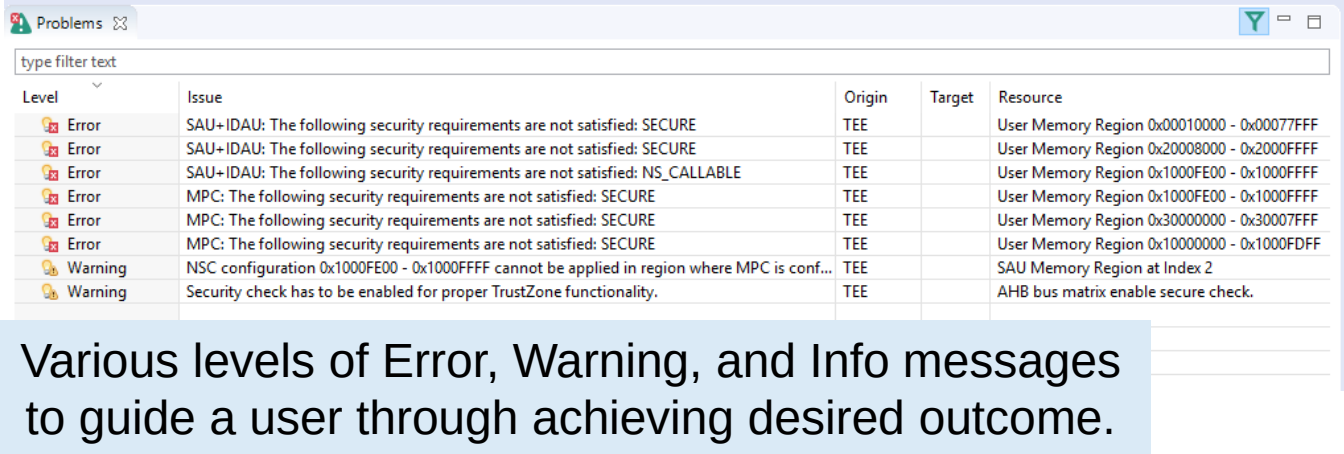
MCUXpresso Config Tool – TEE Configuration



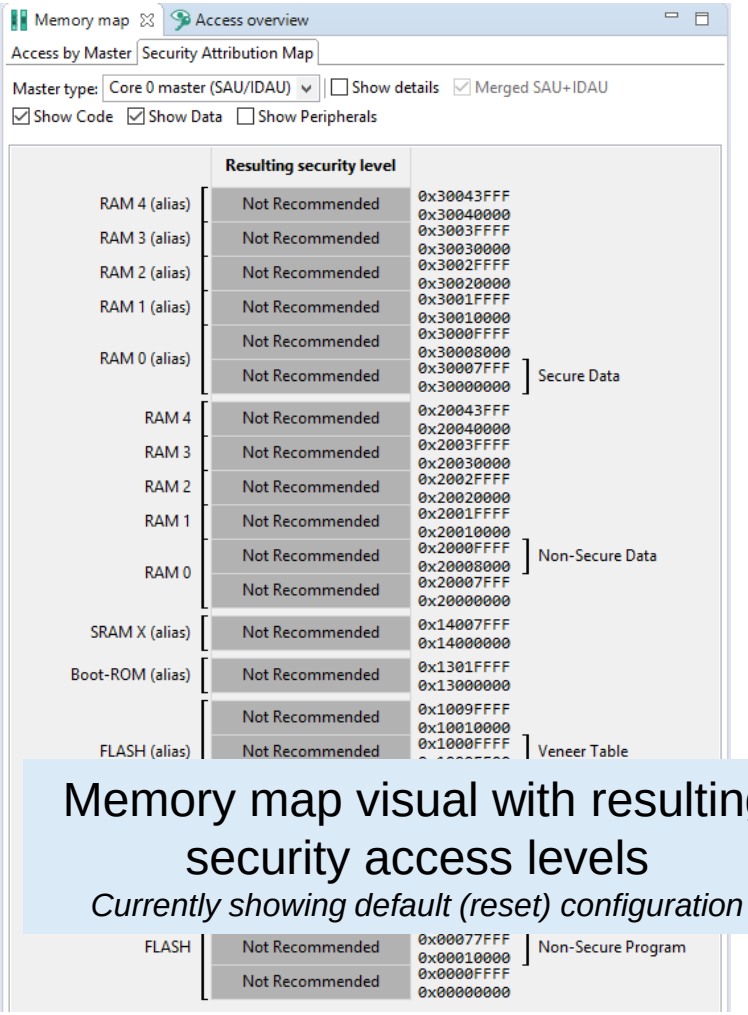
TEE Configuration – Defining User Regions



User Specific Regions (aligned with IDE project settings)



Various levels of Error, Warning, and Info messages to guide a user through achieving desired outcome.

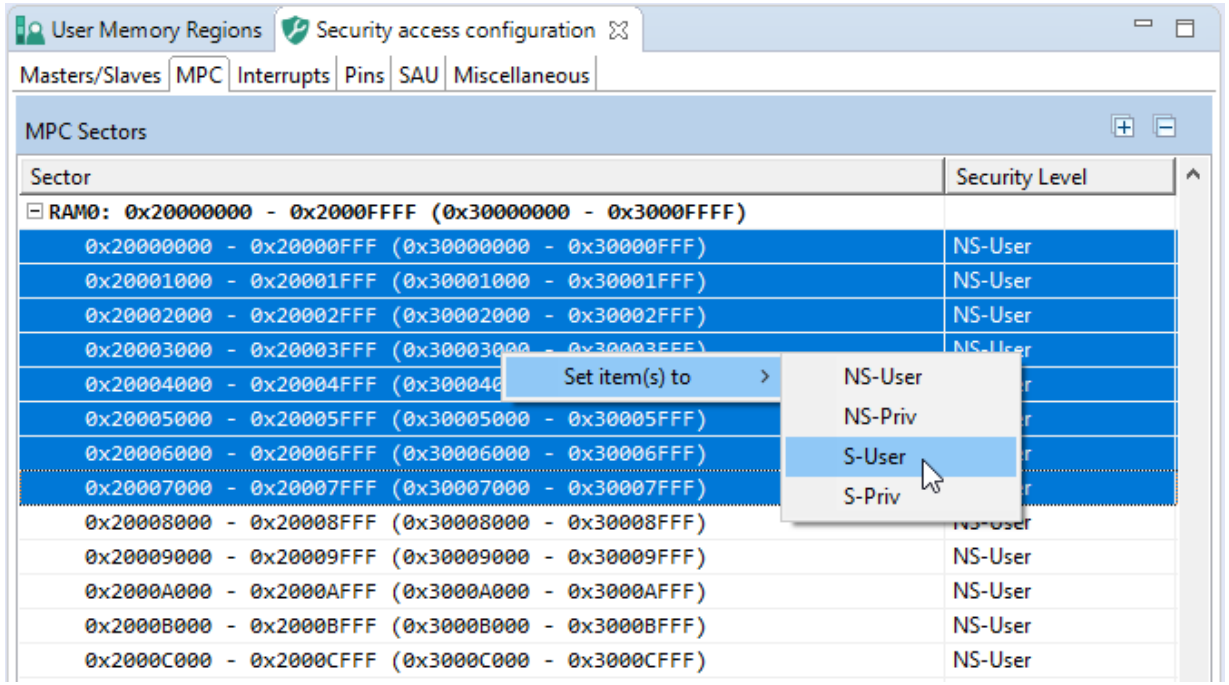


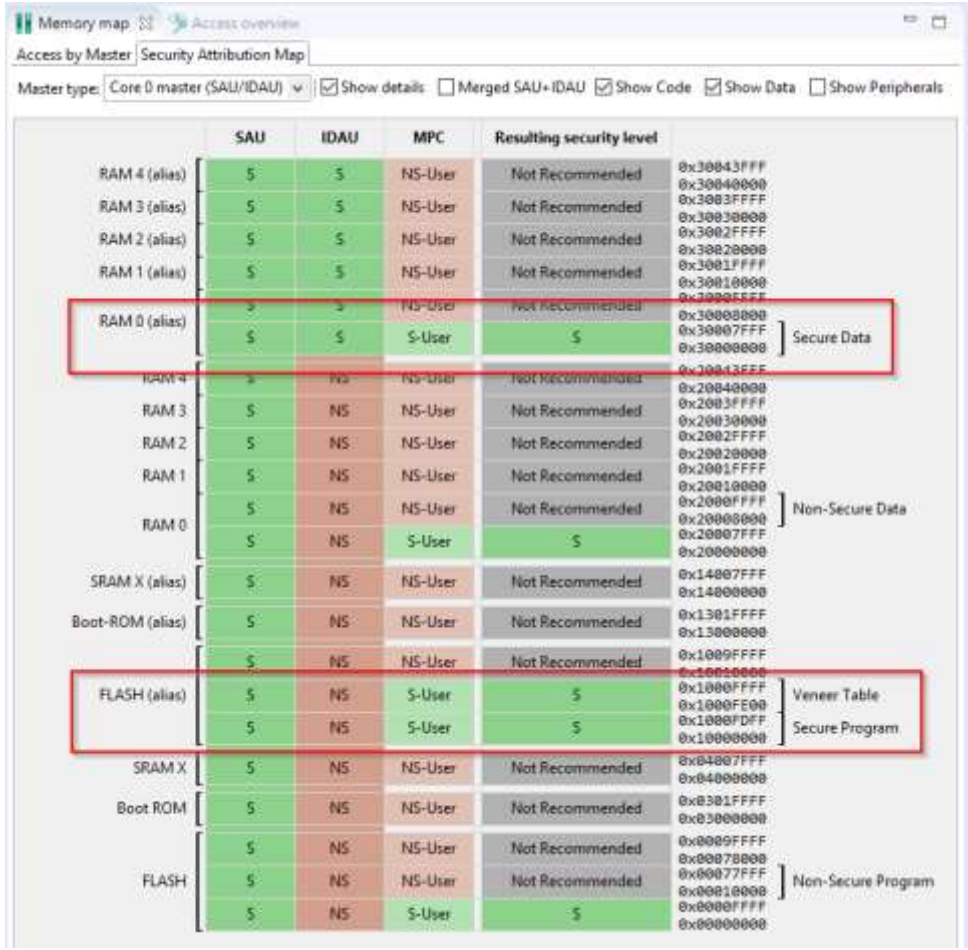
Memory map visual with resulting security access levels

Currently showing default (reset) configuration

TEE Configuration: Securing AHB Memory Accesses (MPC)

Configuration of Memory Protection Checker (AHB Access)

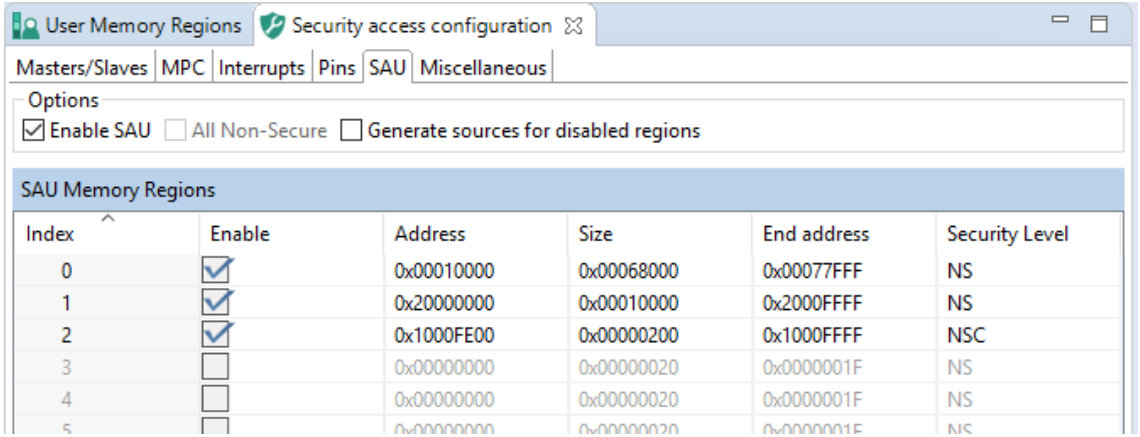


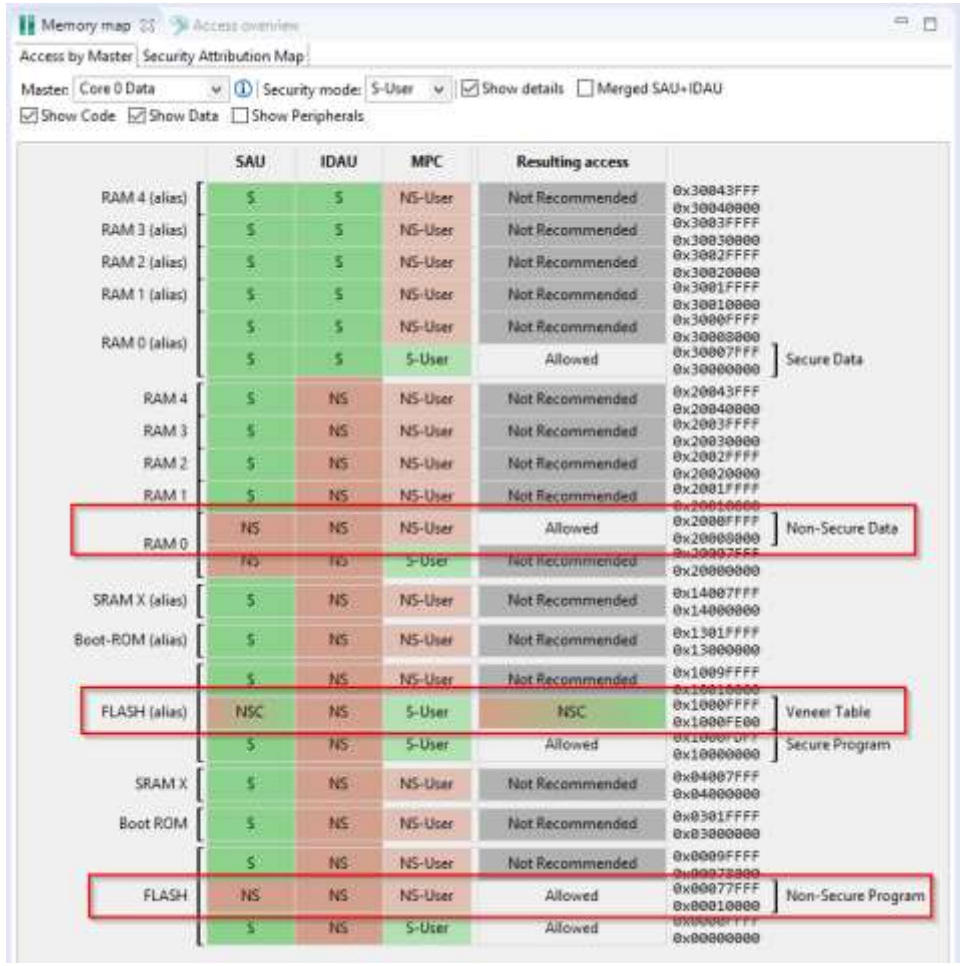


Configuration results dynamically updating on visual

TEE Configuration – Defining SAU Regions

Enabling SAU Memory Regions for
Non-secure and Non-secure Callable





Configuration results dynamically
updating on visual

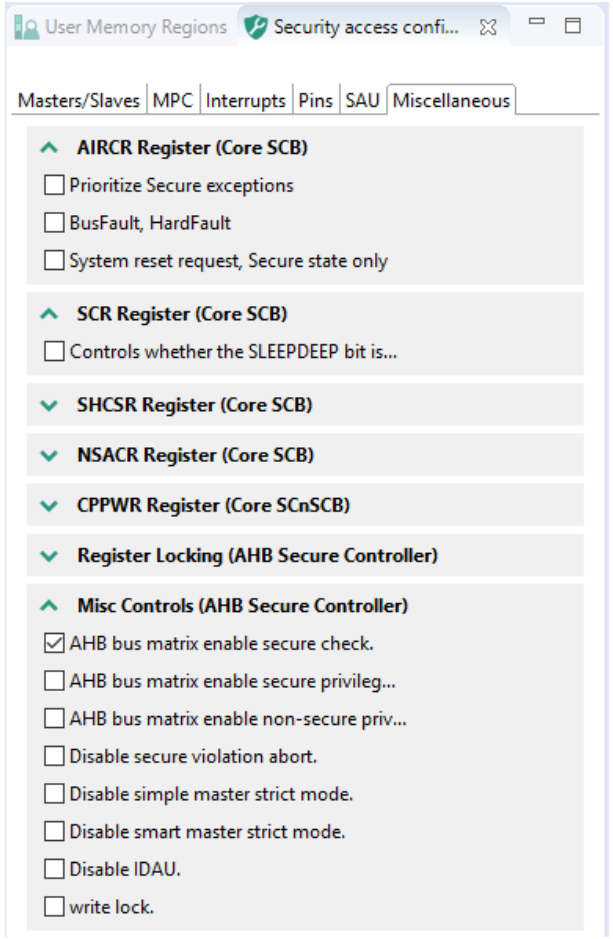
TEE Configuration – Additional Security Access Settings

Master and Slave Security Levels

Master	Security level
Simple master	
DMA0	NS-User
DMA1	NS-User
SDIO	NS-User
USBFSD	NS-User
USBFSH	NS-User
Smart master	
Core 1 Data	NS-User
Core 1 Instructions	NS-User
EZH Data	NS-User
EZH Instructions	NS-User

Slave	Security level
ADC0	NS-User
AHB_SECURE_CTRL	NS-User
ANACTRL	NS-User
CAPTOUCH	NS-User
CASPER	NS-User
CRC_ENGINE	NS-User
CTIMER0	NS-User
CTIMER1	NS-User
CTIMER2	NS-User
CTIMER3	NS-User
CTIMER4	NS-User
DGBMAILBOX	NS-User
DMA0	NS-User
DMA1	NS-User
EFUSE	NS-User
EZH	NS-User
FLEXCOMM0	S-User
FLEXCOMM1	NS-User

Associated Register Settings

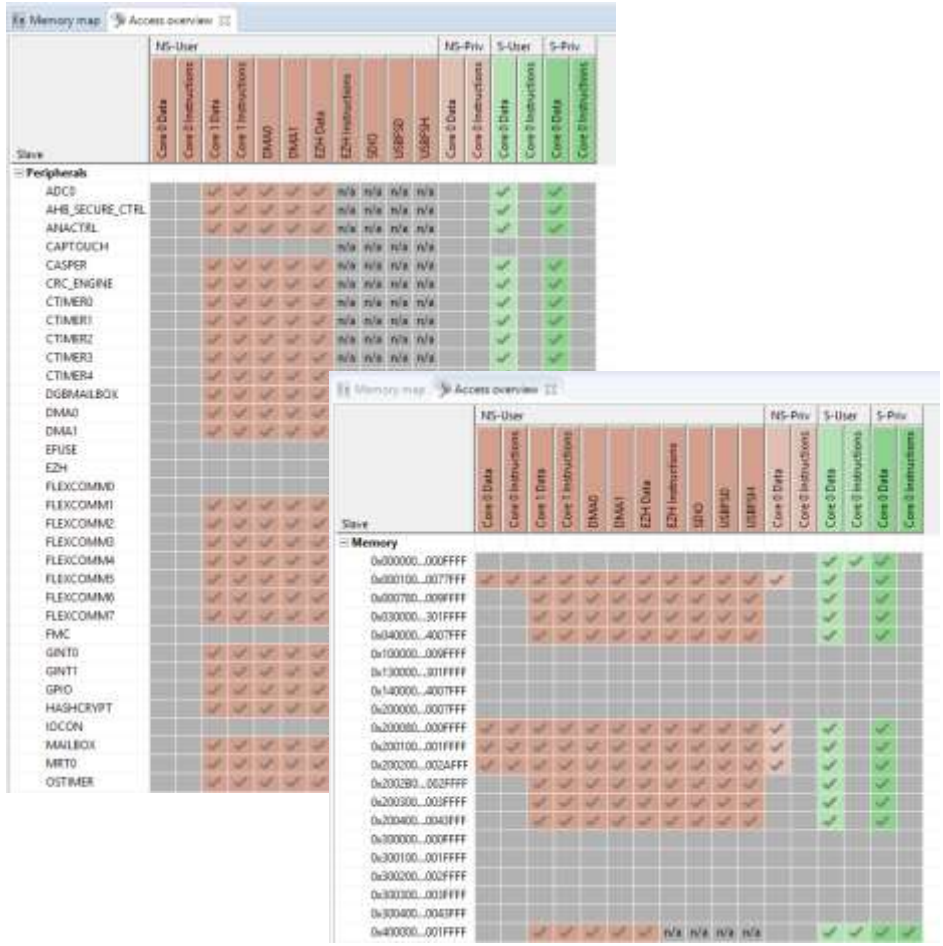


Others:

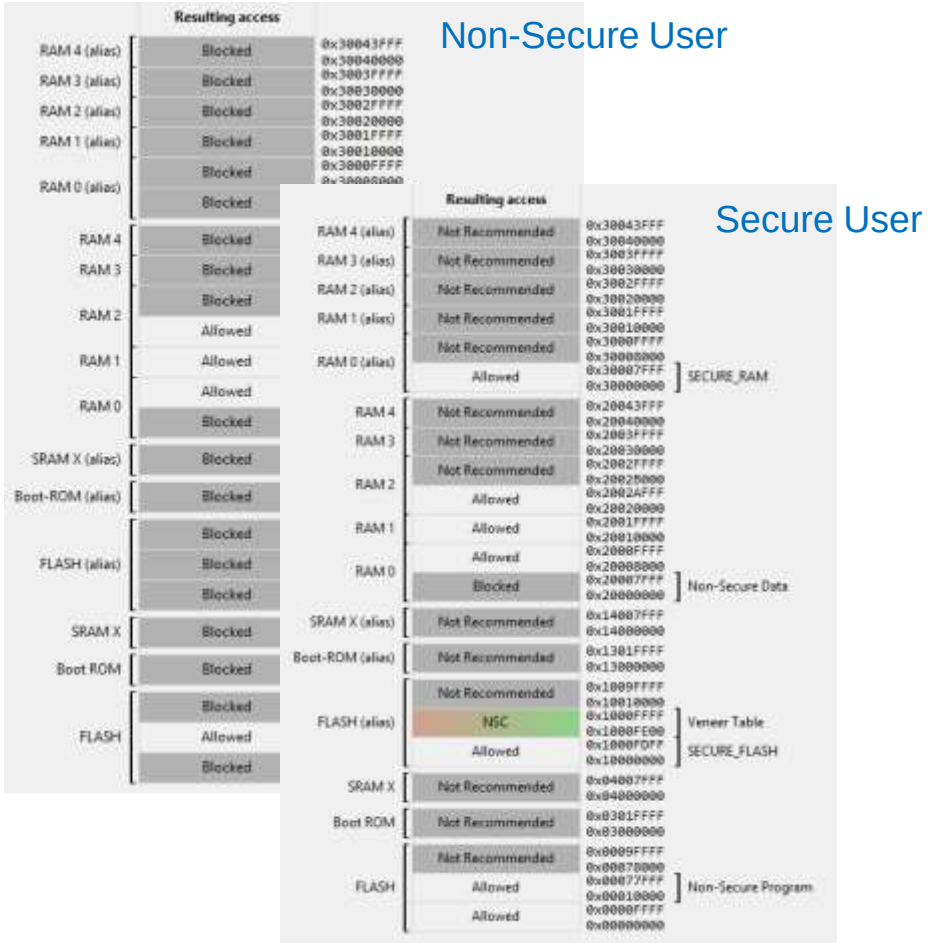
- GPIO Read Access
- Interrupt Handling and Access Level
- Strict Master Modes

TEE Configuration – Additional Visualizations

Access Overview (Matrix)



Memory map access view by security level



TEE Configuration – Code Generation

Source Generation (aligned with SDK)

Overview | Code Preview | Registers

/trustzone/tzm_config.c | /trustzone/tzm_config.h

```
141 * @brief TrustZone initialization
142 *
143 * The function configures SAU and AHB.
144 */
145 void BOARD_InitTrustZone()
146 {
147
148     //################################################################
149     //### SAU configuration ##########################################
150     //################################################################
151
152     /* Set SAU Control register: Disable SAU and All Secure */
153     SAU->CTRL = 0;
154
155     /* Set SAU region number */
156     SAU->RNR = 0;
157     /* Region base address */
158     SAU->RBAR = REGION_0_BASE & SAU_RBAR_BADDR_Msk;
159     /* Region end address */
160     SAU->RLAR = ((REGION_0_END & SAU_RLAR_LADDR_Msk) | ((0U << SAU_RLAR_NSC_Pos) & SAU_RLAR_NSC_Msk)) |
161
162     /* Set SAU region number */
163     SAU->RNR = 1;
164     /* Region base address */
165     SAU->RBAR = REGION_1_BASE & SAU_RBAR_BADDR_Msk;
166     /* Region end address */
167     SAU->RLAR = ((REGION_1_END & SAU_RLAR_LADDR_Msk) | ((0U << SAU_RLAR_NSC_Pos) & SAU_RLAR_NSC_Msk)) |
168
169     /* Set SAU region number */
170     SAU->RNR = 2;
171     /* Region base address */
172     SAU->RBAR = REGION_2_BASE & SAU_RBAR_BADDR_Msk;
173     /* Region end address */
174     SAU->RLAR = ((REGION_2_END & SAU_RLAR_LADDR_Msk) | ((1U << SAU_RLAR_NSC_Pos) & SAU_RLAR_NSC_Msk)) |
175
176     /* Force memory writes before continuing */
177     __DSB();
178     /* Flush and refill pipeline with updated permissions */
179     __ISB();
180     /* Set SAU Control register: Enable SAU and All Secure (applied only if disabled) */
181     SAU->CTRL = 1;
182
183
184     //################################################################
185     //### AHB Configurations #########################################
186     //################################################################
```

Register view

Overview | Code Preview | Registers

all | Show modified registers only

type filter text

Reg. Name	Set Value	Reset Value	Value Description
AHB_SECURE_CTRL_MISC_CTRL_DP_REG	0x????6aa6	0x????aaaa	
IDAU_ALL_NS (bits 15-14)	0b01	0b10	
ENABLE_SECURE_CHECKING (bits 3-2)	0b01	0b10	
AHB_SECURE_CTRL_MISC_CTRL_REG	0x????6aa6	0x????aaaa	
IDAU_ALL_NS (bits 15-14)	0b01	0b10	
ENABLE_SECURE_CHECKING (bits 3-2)	0b01	0b10	
AHB_SECURE_CTRL_SEC_CTRL_AHB0_0_SLAVE_RULE	0x02000000	0x00000000	
FLEXCOMM0_RULE (bits 25-24)	0b10	0b00	Secure and Non-priviledge user access allowed.
AHB_SECURE_CTRL_SEC_CTRL_APB_BRIDGE0_MEM_CTRL0	0x00000022	0x00000000	
IOCON_RULE (bits 5-4)	0b10	0b00	Secure and Non-priviledge user access allowed.
SYSCON_RULE (bits 1-0)	0b10	0b00	Secure and Non-priviledge user access allowed.
AHB_SECURE_CTRL_SEC_CTRL_FLASH_MEM_RULE0	0x00000022	0x00000000	
RULE1 (bits 5-4)	0b10	0b00	Secure and Non-priviledge user access allowed.
RULE0 (bits 1-0)	0b10	0b00	Secure and Non-priviledge user access allowed.
AHB_SECURE_CTRL_SEC_CTRL_RAM0_MEM_RULE0	0x22222222	0x00000000	
RULE7 (bits 29-28)	0b10	0b00	Secure and Non-priviledge user access allowed.
RULE6 (bits 25-24)	0b10	0b00	Secure and Non-priviledge user access allowed.
RULE5 (bits 21-20)	0b10	0b00	Secure and Non-priviledge user access allowed.
RULE4 (bits 17-16)	0b10	0b00	Secure and Non-priviledge user access allowed.
RULE3 (bits 13-12)	0b10	0b00	Secure and Non-priviledge user access allowed.
RULE2 (bits 9-8)	0b10	0b00	Secure and Non-priviledge user access allowed.
RULE1 (bits 5-4)	0b10	0b00	Secure and Non-priviledge user access allowed.
RULE0 (bits 1-0)	0b10	0b00	Secure and Non-priviledge user access allowed.

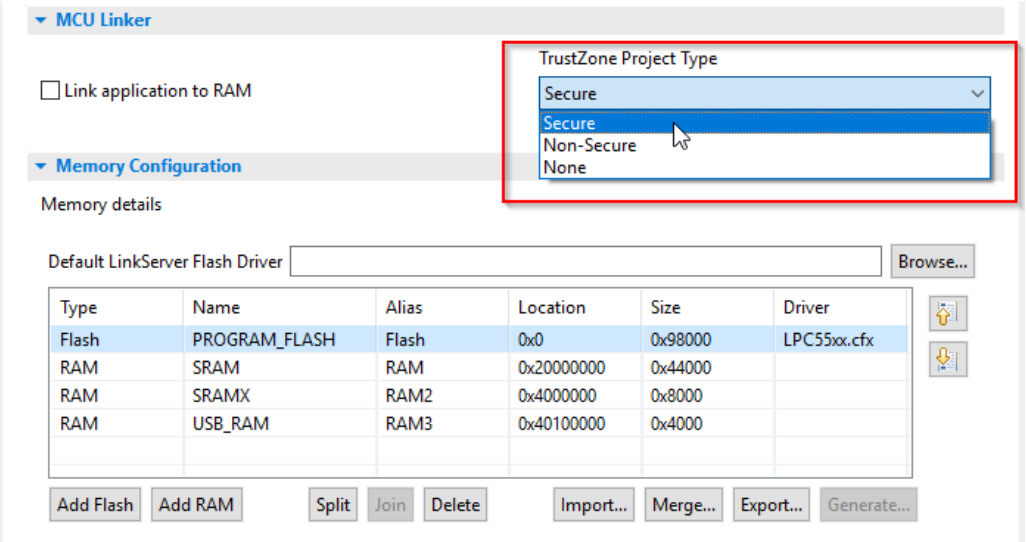
MCUXpresso IDE

TrustZone Project Settings

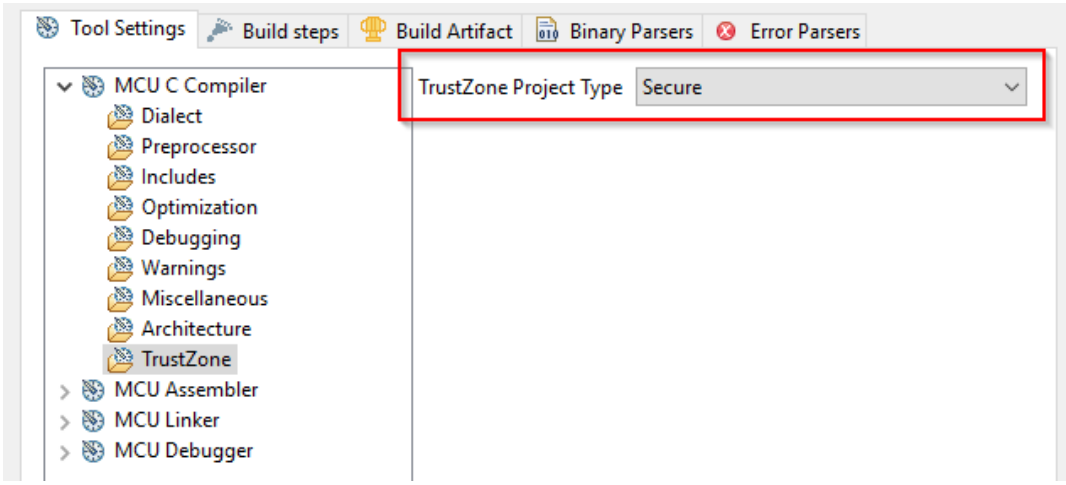
Specification of Secure vs. Non-Secure Project Type

- Support within New Project Wizard to define “Secure” or “Non-Secure” property.
- Setting also available within IDE Project Properties
- Secure / Non-Secure setting provides additional project alignment to be configured between projects.

New Project Wizard

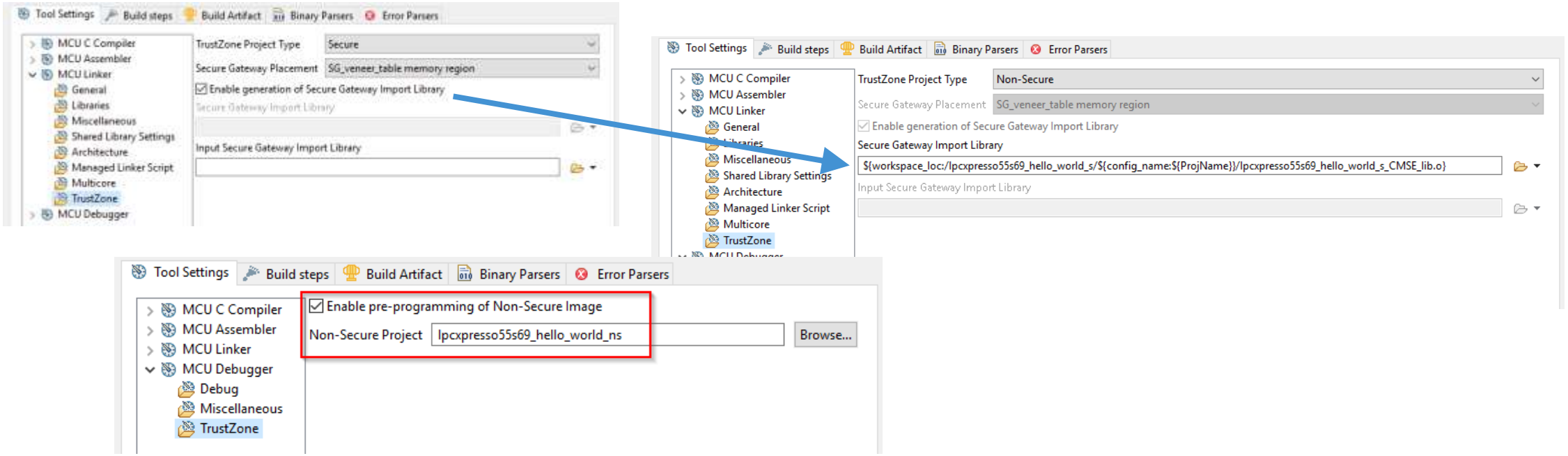


IDE Project Properties



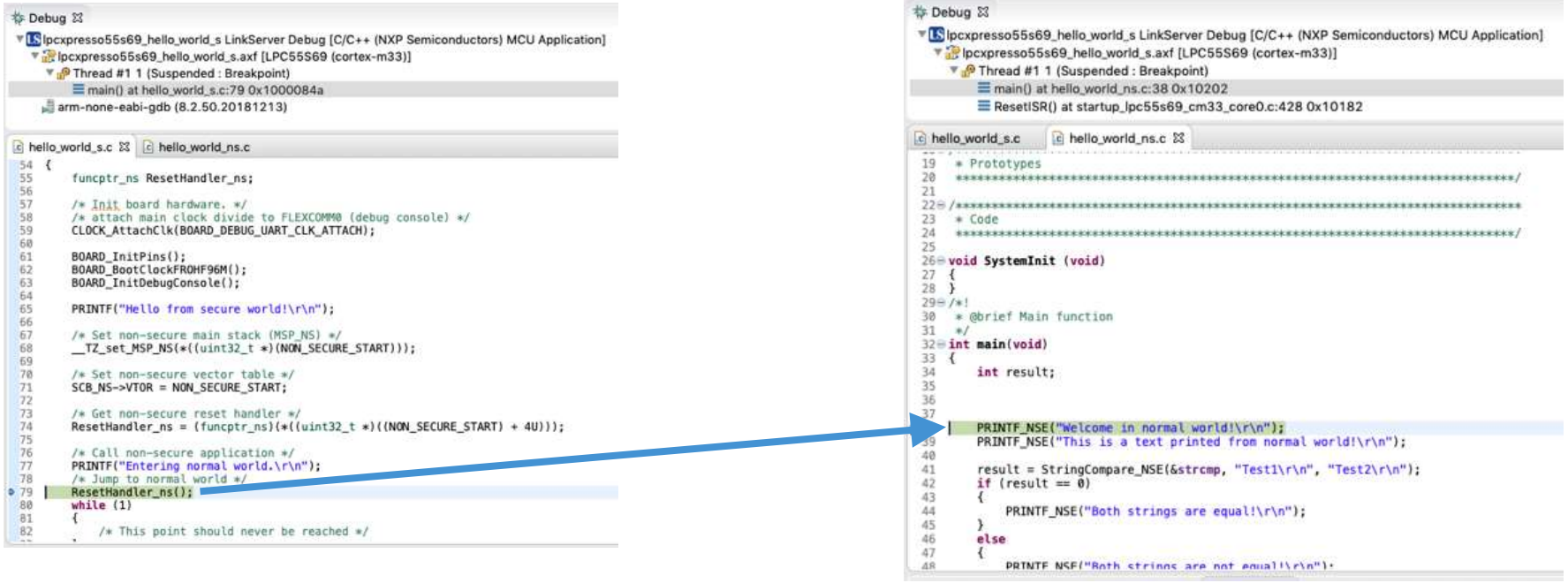
Alignment of Secure and Non-Secure Projects

- Generation of Secure Gateway Import Library within Secure Project
- Specification of Secure Gateway Library from Non-Secure Project
- Direct pre-programming of Non-Secure Image from Secure Project



Improved Cortex-M33 Secure and Non-Secure Projects

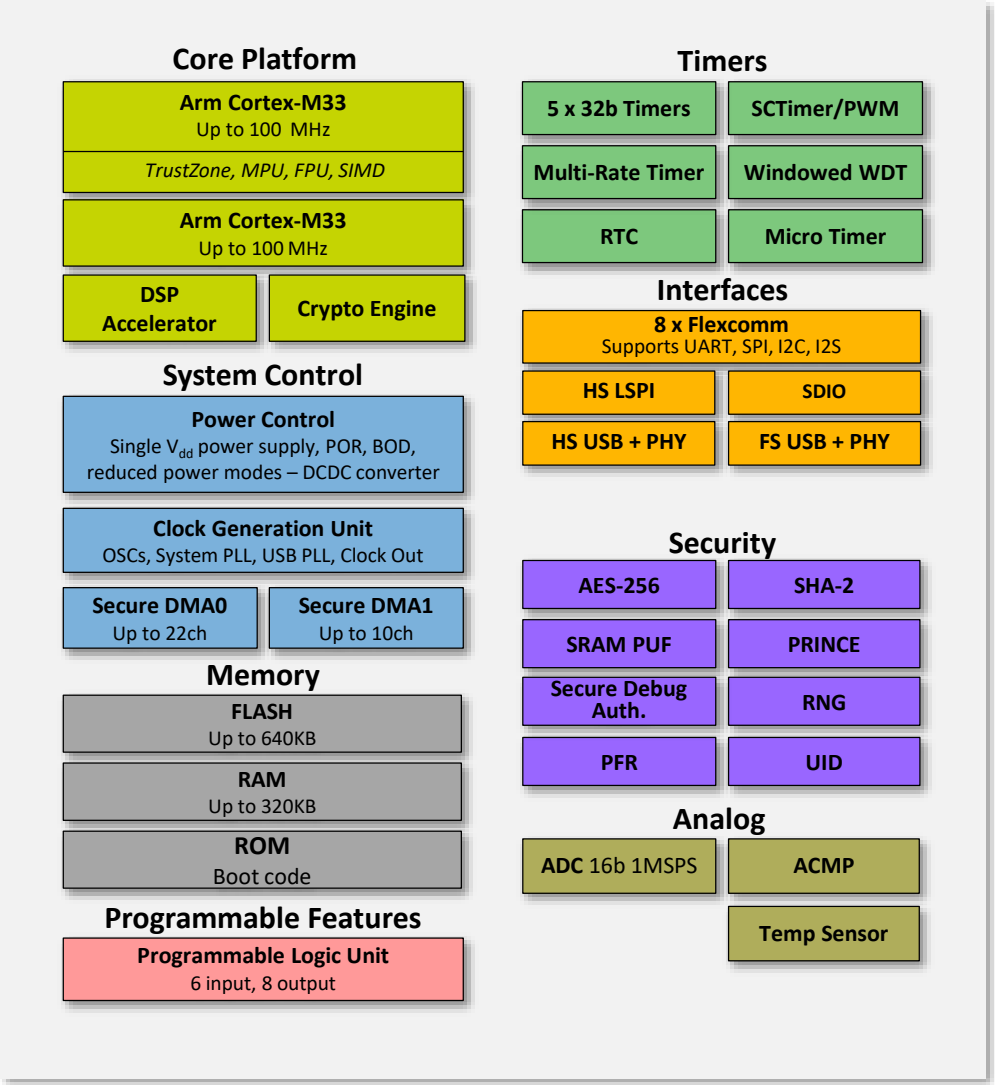
- Secure and Non-secure projects more closely linked
- Debugging Secure project will automatically build Non-Secure project and program it into flash, before launching Secure project debug session
- Symbols for Non-Secure project automatically loaded into debug session to allow user to step from Secure world into Non-Secure (with source code visibility)



LPC55S69

First Cortex M33 MCU

LPC55S6x MCU Family



Core Platform

- Up to 100MHz Cortex-M33
  - TrustZone, MPU, FPU, SIMD
- Up to 100MHz Cortex-M33
- Coprocessors
  - DSP Accelerator
  - Crypto Engine
- Multilayer Bus Matrix

Memory

- Up to 640KB FLASH (includes PFR)
- Up to 320KB RAM
- 128KB ROM

Timers

- 5 x 32b Timers
- SCTimer/PWM
- Multi-Rate Timer
- OS Timer
- Windowed Watchdog Timer
- RTC
- Micro Timer

Interfaces

- USB High-speed (H/D) w/ on-chip HS PHY
- USB Full-speed (H/D), Crystal-less
- SDIO, Support 2 cards
- 1 x High-Speed SPI up to 50MHz
- 8 x Flexcomms support up to 8x SPI, 8x I2C, 8x UART, 4x I^2S channels (total 8 instances)

Advanced Security Subsystem

- Protected Flash Region (PFR)
- AES-256 HW Encryption/Decryption Engine
- SHA-2
- SRAM PUF for Key Generation support
- PRINCE – On-The-Fly Encrypt/Decrypt for flash data
- Secure debug authentication
- RNG

Analog

- 16b ADC, 16ch, 1MSPS
- Analog Comparator
- Temperature Sensor

Packages

- LQFP100
- VFBGA98
- LQFP64

Other

- Programmable Logic Unit
- Buck DC-DC
- Operating voltage: 1.8 to 3.6V
- Temperature range: -40 to 105 °C

LPC5500 MCU Series

Common features across families,

- FS/HS USB with PHY, 50MHz SPI, up to 8/10 Serial Interfaces (FlexComm), plus I3C interface (LPC557x/8x families)
- Up to 2Msps 16-bit SAR ADC, Comparator, Temperature Sensor and RTC
- 1.8 to 3.6V, -40 to 105 °C

LPC5500 Family**	Samples**	Memory	CPU Freq	Dual Core	Security Features	DSP Accel.	FS&HS USB	SDIO	CAN-FD	10/100 ENET	Graphics Accel.	16-bit ADC & Comp.	Serial Interface
Graphics/HMI LPC558x/S8x	**Q1-20**	Up to 2MB Flash, 512KB SRAM	200 MHz Opt TZ	Yes	Opt.	Yes	Yes	Yes	2x	1x	Yes	Yes	10x FlexComm HS SPI, I3C
Large Memory LPC557x/S7x	**Q1-20**	Up to 2MB Flash, 512KB SRAM	200 MHz Opt TZ	Yes	Opt.	Yes	Yes	Yes	2x	1x	-	Yes	10x FlexComm HS SPI, I3C
Efficiency LPC55S6x	**Now**	Up to 640KB Flash, 320KB SRAM	100 MHz Opt TZ	Yes	Yes	Yes	Yes	Yes	-	-	-	Yes	8x FlexComm, HS SPI
Mainstream LPC552x/S2x	**Q2-19**	Up to 512KB Flash, 256KB SRAM	100 MHz	-	-	-	Yes*	Opt.	-	-	-	Yes	8x FlexComm, HS SPI
Entry LPC551x/S1x	**Q3-19**	Up to 256KB Flash, 96KB SRAM	100 MHz Opt TZ	-	Opt.	-	Yes*	-	Yes*	-	-	Yes	8x FlexComm, HS SPI
Flashless MCU LPC550x	**Q2-20**	0KB Flash, 96KB SRAM	100 MHz Opt TZ	-	-	-	Yes	Yes	-	-	-	Yes	8x FlexComm, HS SPI

**HS USB/CAN-FD not available on all part numbers within the family, check data sheet for specific configurations*
***Product series features and availability subject to change*

Hands-On Lab

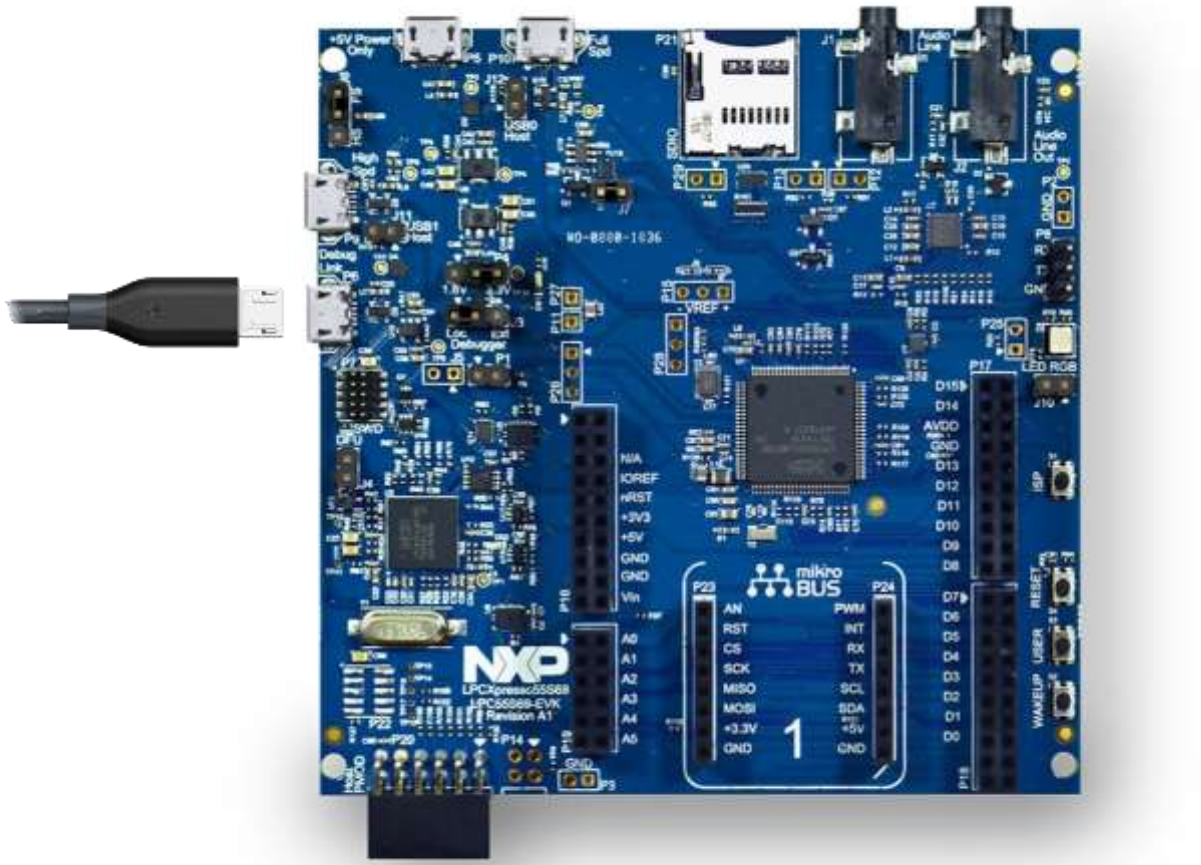
MCUXpresso SDK Example Project (Hello World – Secure / Non-Secure)

MCUXpresso IDE Project Settings (Creating a Secure Application)

MCUXpresso SDK Development of a Non-Secure Application

Hands-On Lab

- MCUXpresso IDE v11.0
- MCUXpresso SDK v2.6.0
- LPC55S69-EVK (LPCXpresso55S69)
- MicroUSB Cable
- Lab Guide



SECURE CONNECTIONS
FOR A SMARTER WORLD