

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 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

UNIFIED SUITE OF
TOOLS FOR EASY
DEVELOPMENT
WITH NXP MCUs



LEARN MORE >



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



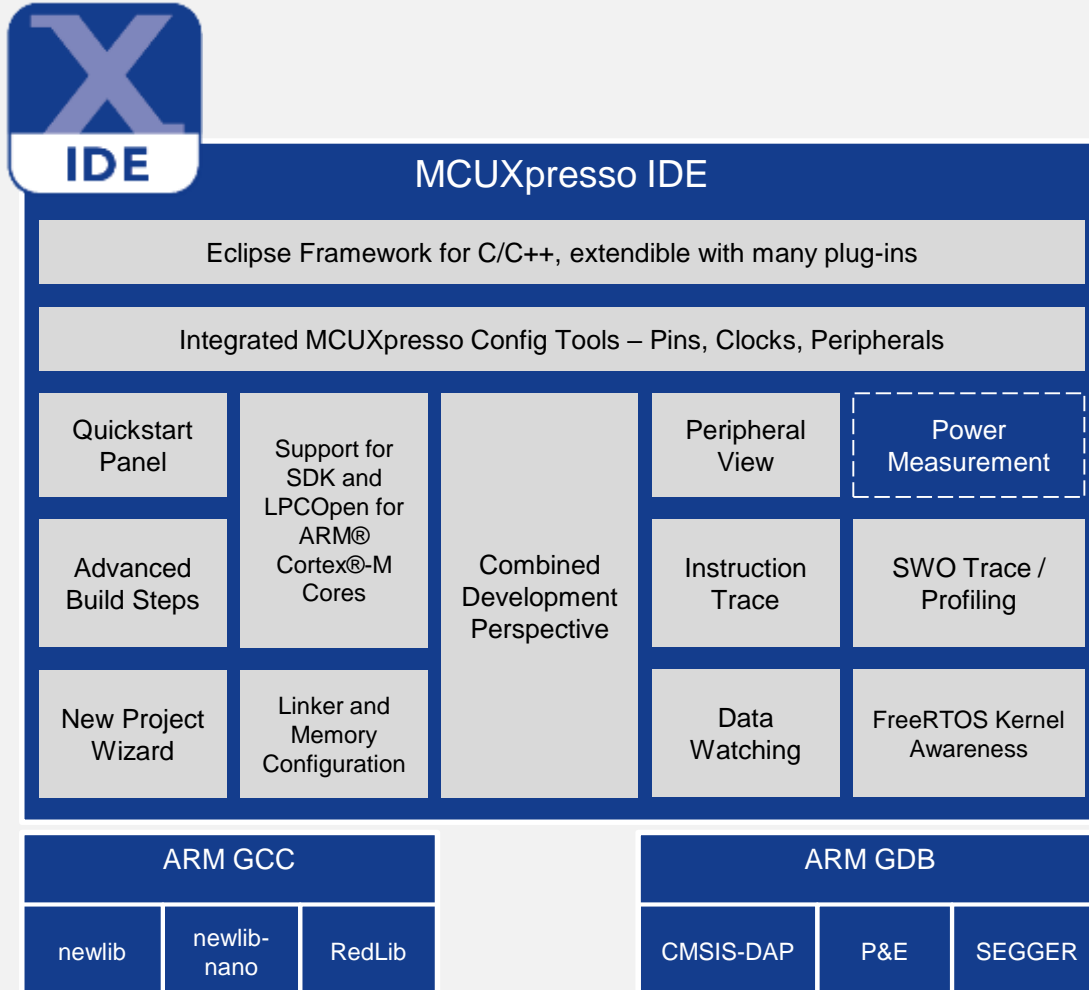
MCUXpresso SDK

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



MCUXpresso Config Tools

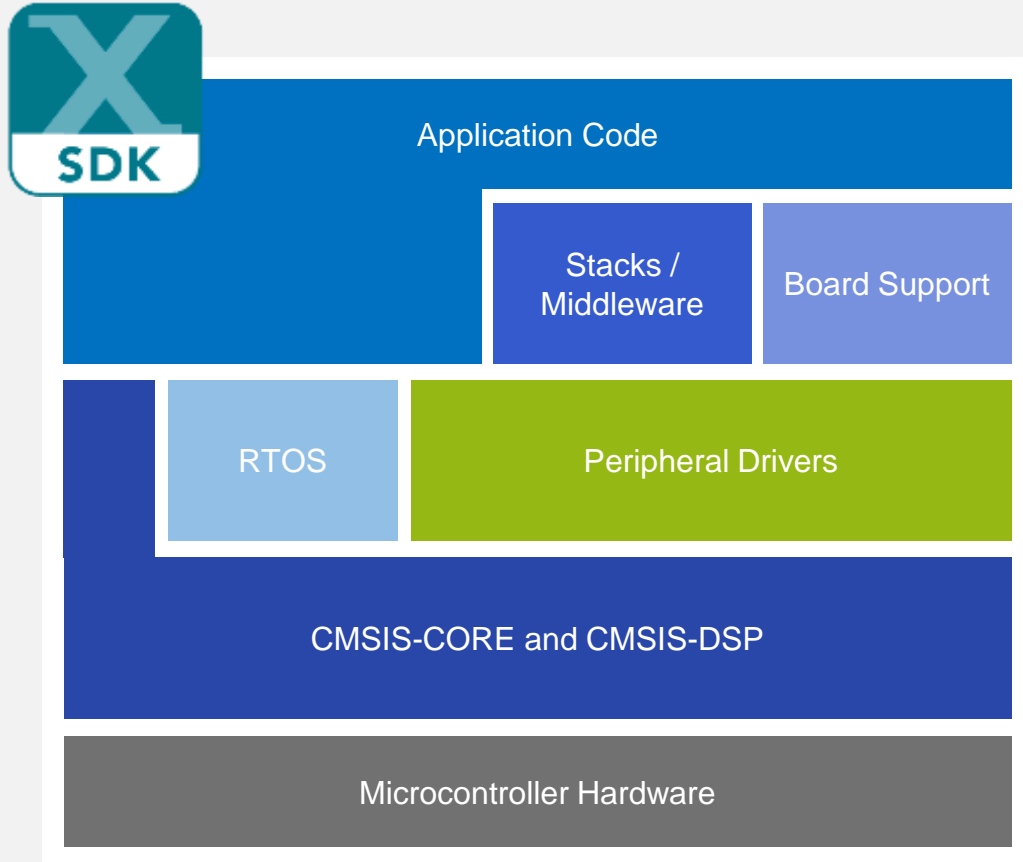
Online and desktop tool suite for system configuration and optimization



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



MCUXpresso SDK

Software Framework and Drivers

Architecture:

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

Reference Software:

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

License:

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

Integrated RTOS:

- Amazon FreeRTOS
- RTOS-native driver wrappers

Toolchains:

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

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

Quality:

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



Open Source Initiative



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.



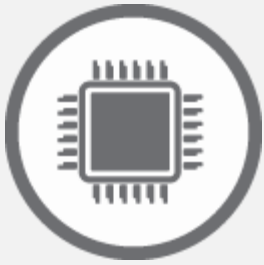
Project updater works directly with existing SDK-based IDE projects with generated Pins, Clocks and Peripherals source files.



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

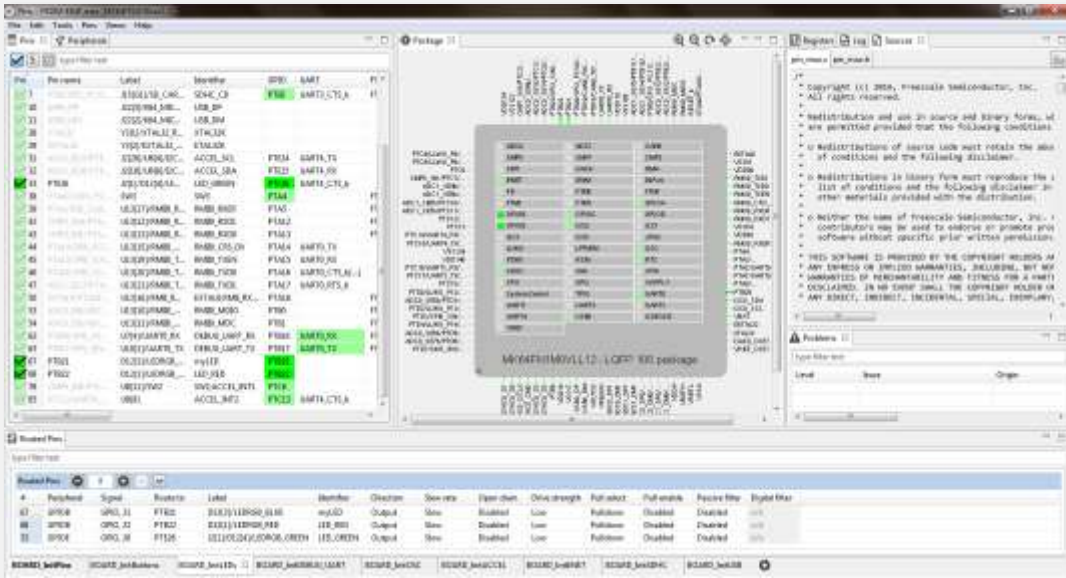


Trusted Execution Environment configures protection and isolation of sensitive parts of the application.



Easy-to-use muxing and pin assignments

MCUXpresso Config Tools Pins Configuration

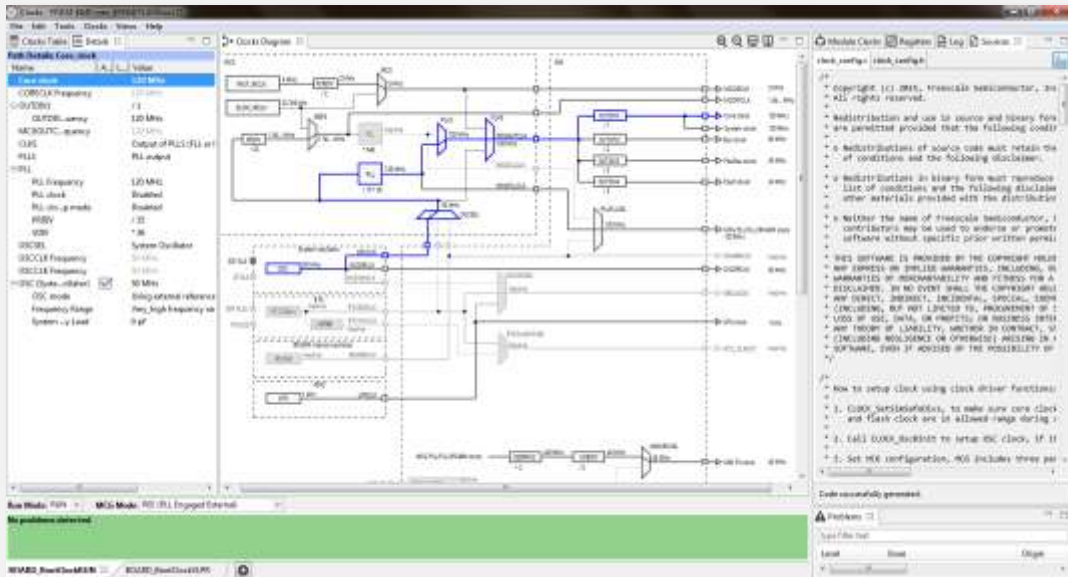


- Muxing and pin configuration with consistency checking
- ANSI-C configuration code
- Graphical processor package view
- Wizard for optimized assignments of functionality to available pins
 - Selection of Pins and Peripherals
 - Routed pins with electrical characteristics
 - Registers with configured 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

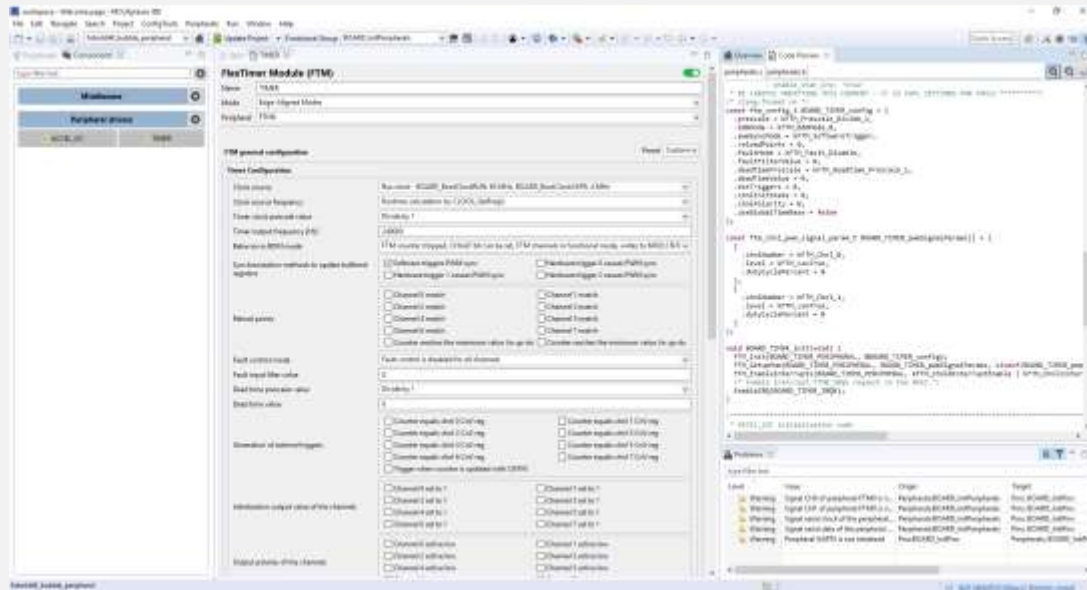
- System clock configuration with consistency checking
- ANSI-C initialization code
- Graphical clock diagrams
- Easy-to-use guided graphical user interface
 - Selection of Clock Sources
 - Configuration of prescalers and clock outputs
 - Details and Full Diagram views with clock path
 - Registers with 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



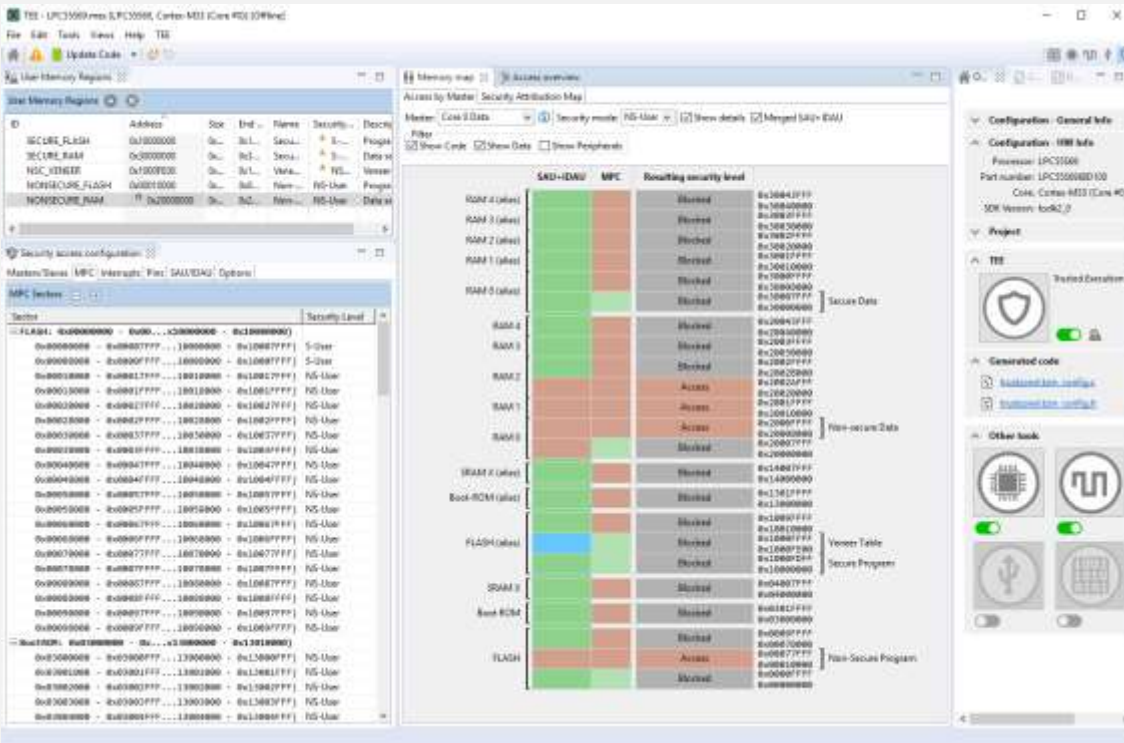


Generation and visualization of Secure / Non-Secure environment partitioning



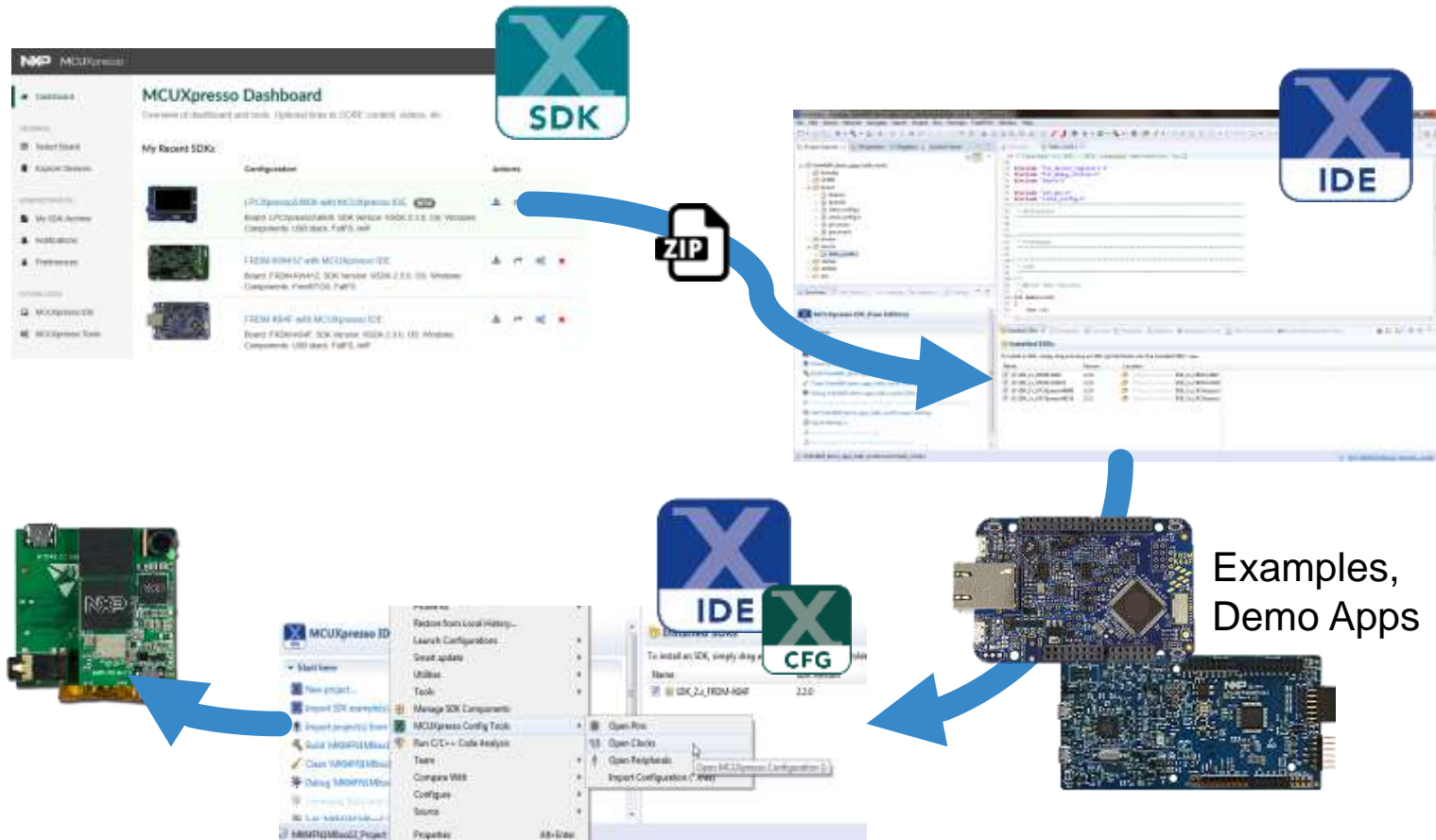
MCUXpresso Config Tools Trusted Execution Environment

- 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

UNIFIED SUITE OF
TOOLS FOR EASY
DEVELOPMENT
WITH NXP MCUs



LEARN MORE >

NXP

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

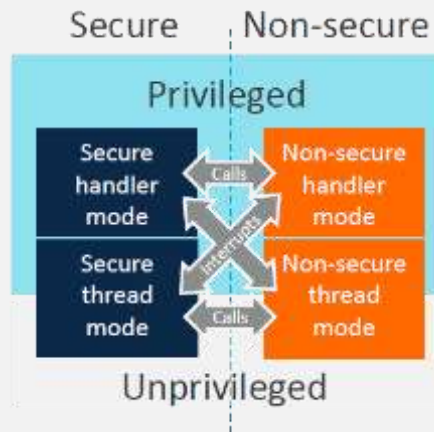
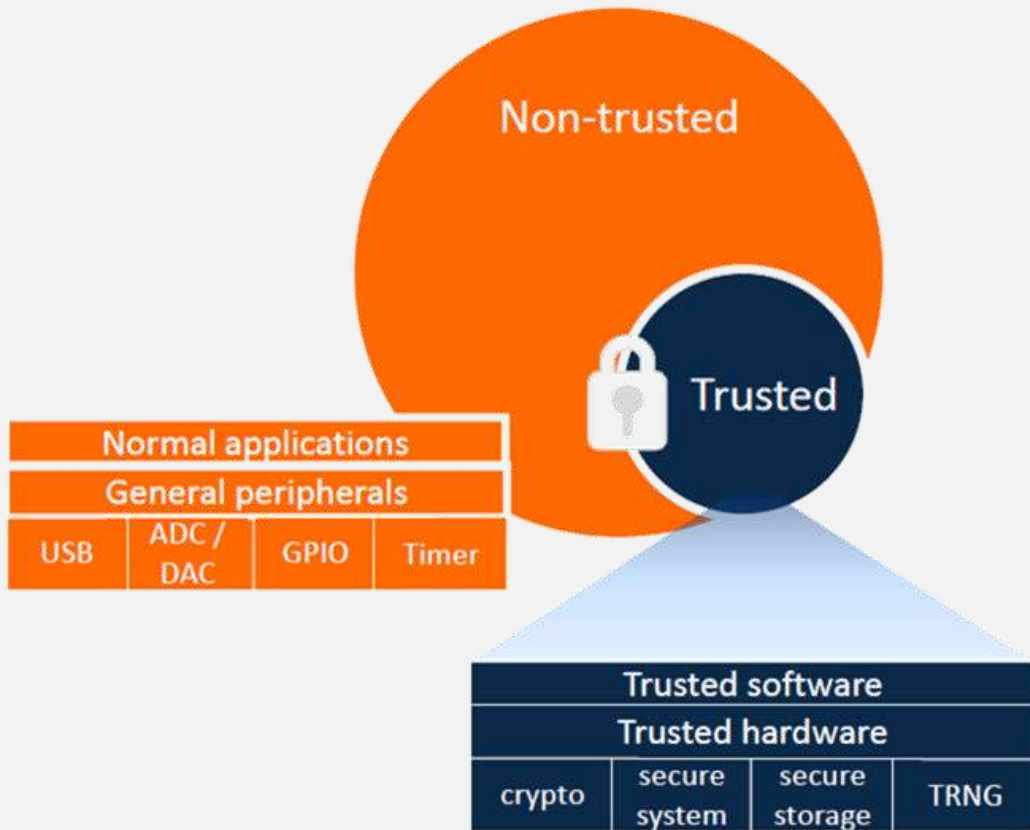
- MCUXpresso Software and Tools – <https://community.nxp.com/community/mcuxpresso>
 - MCUXpresso SDK: <https://community.nxp.com/community/mcuxpresso/mcuxpresso-sdk>
 - MCUXpresso IDE: <https://community.nxp.com/community/mcuxpresso/mcuxpresso-ide>
 - MCUXpresso Config Tools: <https://community.nxp.com/community/mcuxpresso/mcuxpresso-config>

Supported Devices

- [Supported Devices Table \(Community Doc\)](#)

Overview of TrustZone on Armv8-M



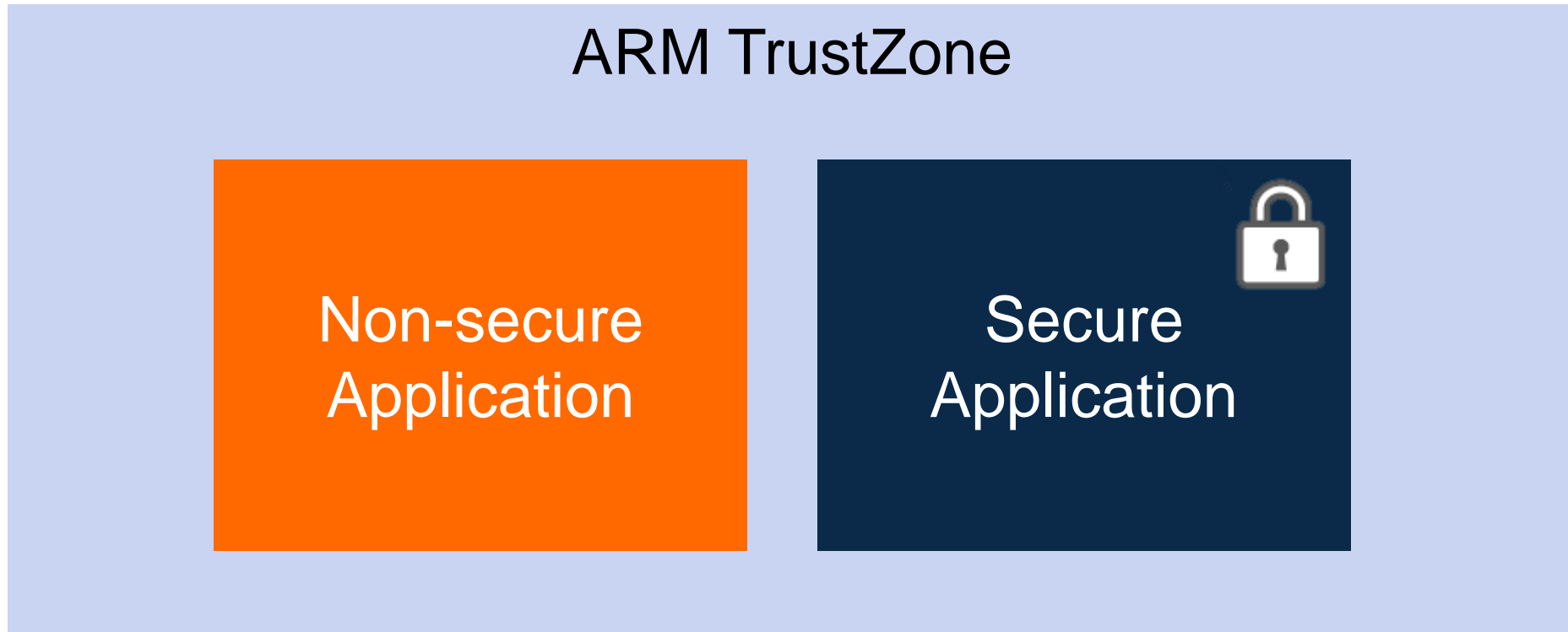


Concepts of TrustZone

- 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

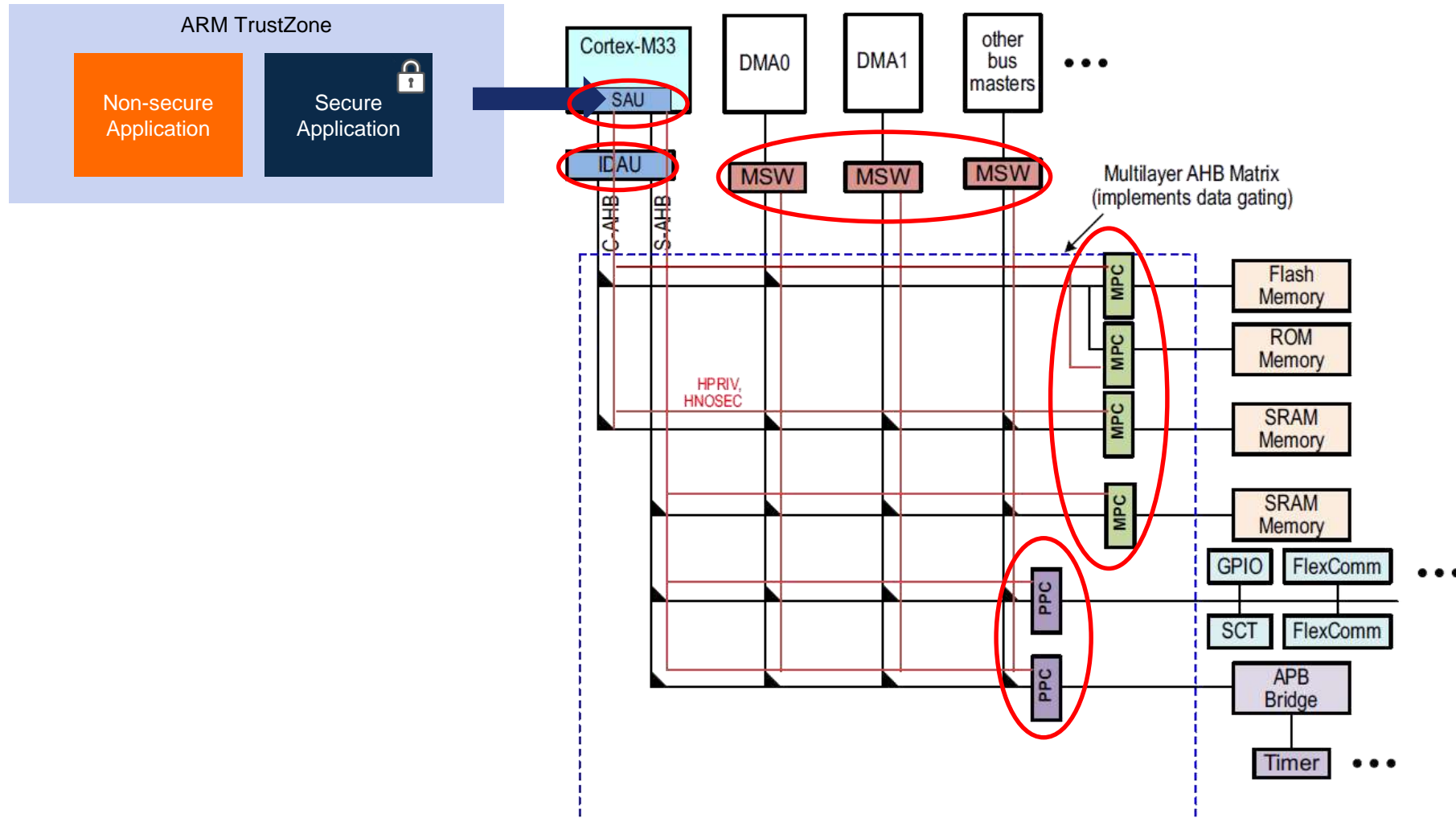
Secure and Non-Secure Application Development

Easy, Right?



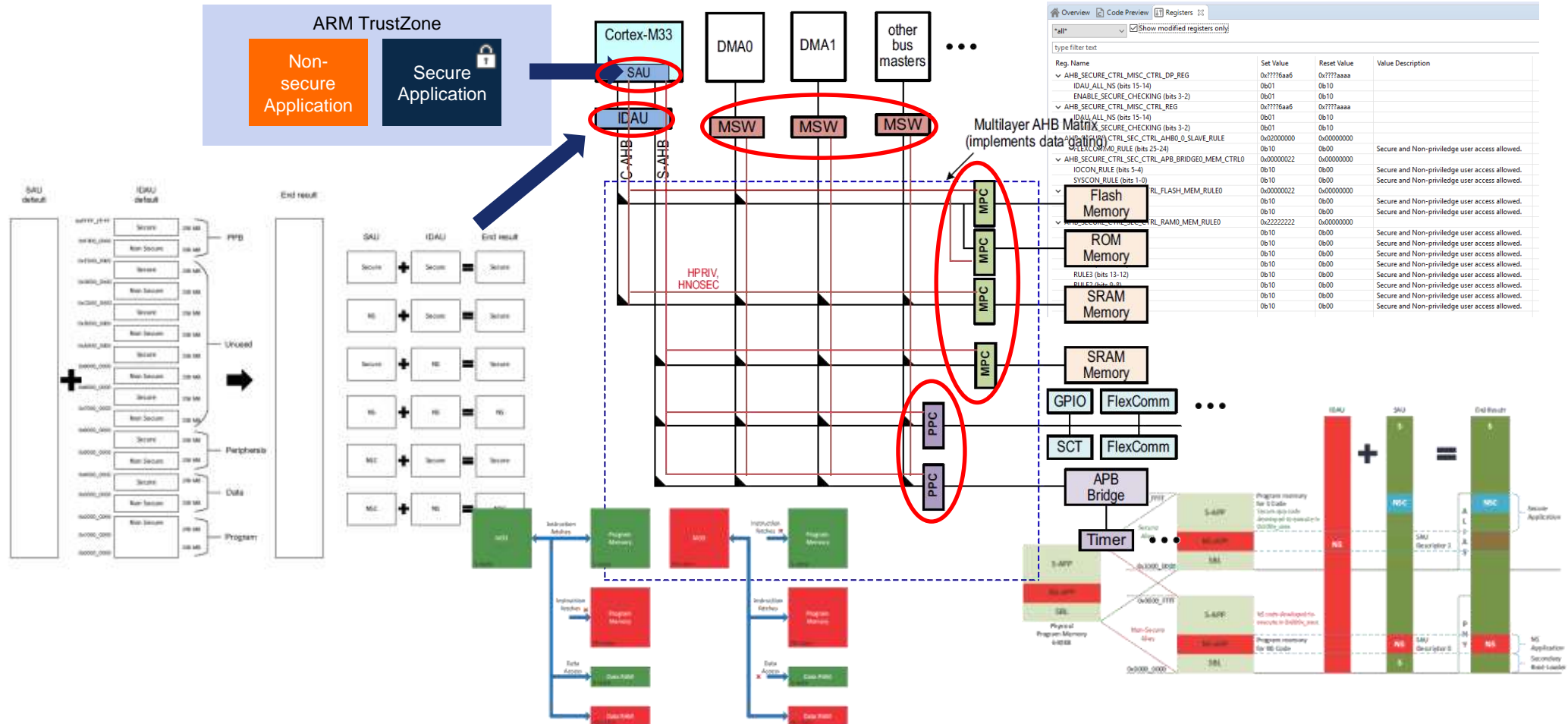
Secure and Non-Secure Application Development

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

The screenshot displays the MCUXpresso IDE interface for configuring the Security Attribution Map (SAU) for a Cortex-M33 core. The main window shows the 'Access overview' for 'Core 0 Data' in 'S-Priv' security mode.

User Memory Regions Table:

ID	Address	Size	End address	Name	Security Level
SECURE_RAM	0x00000000	0x00000000	0x30007FFF	Secure Stack and ...	S-Priv
SECURE_FLASH	0x10000000	0x00000000	0x10007FFF	Secure Code	S-Priv
NSC_VENEER	0x1000FE00	0x00000200	0x1000FFFF	Veneer Table	NSC-Priv
NONSECURE_RAM	0x20000000	0x00020000	0x20032FFF	Non-secure Stack ...	NS-User
NONSECURE_FLASH	0x00010000	0x00060000	0x00071FFF	Non-secure Code	NS-User

Memory map Table (Access by Master | Security Attribution Map):

SAU + IDAU	MPC	Resulting access	Address Range	Category
RAM 4 (alias)	S	NS-User	0x20043FFF - 0x20040000	Secure Stack and Data
RAM 3 (alias)	S	NS-User	0x2003FFF - 0x20030000	
RAM 2 (alias)	S	NS-User	0x2002FFF - 0x20020000	
RAM 1 (alias)	S	NS-User	0x2001FFF - 0x20010000	
RAM 0 (alias)	S	NS-User	0x2000FFF - 0x20000000	
RAM 4	NS	NS-User	0x20043FFF - 0x20040000	Non-secure Stack and ...
RAM 3	NS	NS-User	0x2003FFF - 0x20030000	
RAM 2	NS	NS-User	0x2002FFF - 0x20020000	
RAM 1	NS	NS-User	0x2001FFF - 0x20010000	
RAM 0	NS	NS-User	0x2000FFF - 0x20000000	
SRAM X (alias)	S	S-Priv	0x20000000	
Boot-ROM (alias)	S	NS-User	0x14007FFF - 0x14000000	Veneer Table
FLASH (alias)	NSC	S-Priv	0x1000FFF - 0x10000000	
SRAM X	NS	NS-User	0x1000FFF - 0x10000000	Secure Code
Boot-ROM	NS	NS-User	0x1000FE00 - 0x1000F000	
FLASH	NS	S-Priv	0x1000F000 - 0x10000000	Non-secure Code
SRAM X	NS	NS-User	0x04007FFF - 0x04000000	
Boot-ROM	NS	NS-User	0x0301FFF - 0x03000000	
FLASH	NS	NS-User	0x0000FFF - 0x00000000	

Configuration - General Info:

- Process: LPC5569
- Part number: LPC5569BD100
- Core: Cortex-M33 (Core #0)
- SDK Version: lsd2_0

Configuration - HW Info:

- TEE: Configures access policies for memory areas and peripherals helping to protect and isolate sensitive parts of application. (Enabled)
- Generated code: Update code enabled. Links to `lpc5569_tee_config.s` and `lpc5569_tee_config.h`.
- Other tools: Includes icons for chip, TEE, USB, and I2C/SPI.

Problems:

- Information: Ending address is not aligned with the end of the AHB memory block: 0

TEE Configuration – Defining User Regions

The image shows two windows from an IDE. The left window, 'User Memory Regions', displays a table of memory regions with their addresses, sizes, and security levels. The right window, 'Memory map', shows a visual representation of the memory map with resulting security levels for various memory regions.

User Memory Regions Table:

ID	Address	Size	End address	Name	Security Level	Description
NONSECURE_FLASH	0x00010000	0x00068000	0x00077FFF	Non-Secure Program	NS-User	Program section of Non-Secure application (from nonsecure linker file)
NONSECURE_RAM	0x20008000	0x00008000	0x2000FFFF	Non-Secure Data	NS-User	Data section of Non-Secure application (from nonsecure linker file)
NSC_VENEER	0x1000FE00	0x00000200	0x1000FFFF	Veneer Table	NSC-User	Veneer Table of Nonsecure Callable function (from secure linker file)
SECURE_FLASH	0x10000000	0x0000FE00	0x1000FDFF	Secure Program	S-User	Program section of Secure application (from secure linker file)
SECURE_RAM	0x30000000	0x00008000	0x30007FFF	Secure Data	S-User	Data section of Secure application (from secure linker file)

Memory map visual with resulting security access levels:

Memory Region	Resulting security level	Address Range	Category
RAM 4 (alias)	Not Recommended	0x30043FFF - 0x30040000	Secure Data
RAM 3 (alias)	Not Recommended	0x3003FFFF - 0x30030000	
RAM 2 (alias)	Not Recommended	0x3002FFFF - 0x30020000	
RAM 1 (alias)	Not Recommended	0x3001FFFF - 0x30010000	
RAM 0 (alias)	Not Recommended	0x3000FFFF - 0x30008000	
RAM 4	Not Recommended	0x20043FFF - 0x20040000	Non-Secure Data
RAM 3	Not Recommended	0x2003FFFF - 0x20030000	
RAM 2	Not Recommended	0x2002FFFF - 0x20020000	
RAM 1	Not Recommended	0x2001FFFF - 0x20010000	
RAM 0	Not Recommended	0x2000FFFF - 0x20008000	
SRAM X (alias)	Not Recommended	0x14007FFF - 0x14000000	
Boot-ROM (alias)	Not Recommended	0x1301FFFF - 0x13000000	
FLASH (alias)	Not Recommended	0x1009FFFF - 0x10010000	Veneer Table
	Not Recommended	0x1000FFFF - 0x00000000	
FLASH	Not Recommended	0x00077FFF - 0x00010000	Non-Secure Program
	Not Recommended	0x0000FFFF - 0x00000000	

Problems Table:

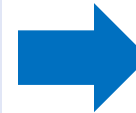
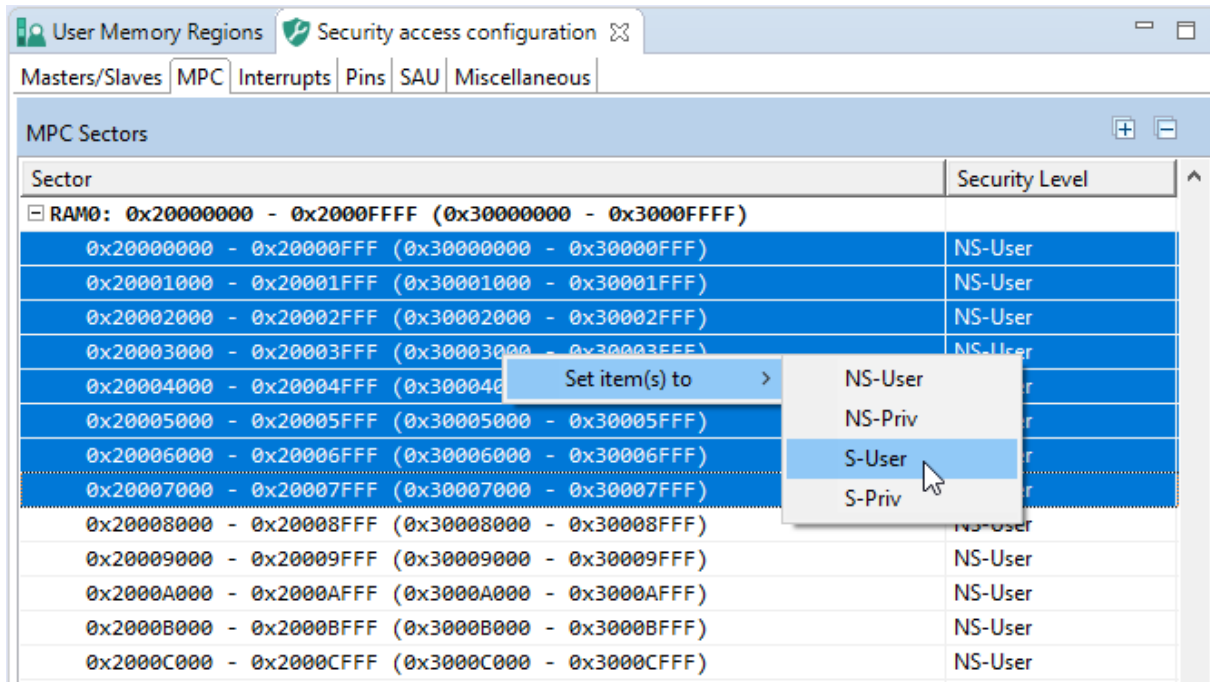
Level	Issue	Origin	Target	Resource
Error	SAU+IDAU: The following security requirements are not satisfied: SECURE	TEE		User Memory Region 0x00010000 - 0x00077FFF
Error	SAU+IDAU: The following security requirements are not satisfied: SECURE	TEE		User Memory Region 0x20008000 - 0x2000FFFF
Error	SAU+IDAU: The following security requirements are not satisfied: NS_CALLABLE	TEE		User Memory Region 0x1000FE00 - 0x1000FFFF
Error	MPC: The following security requirements are not satisfied: SECURE	TEE		User Memory Region 0x1000FE00 - 0x1000FFFF
Error	MPC: The following security requirements are not satisfied: SECURE	TEE		User Memory Region 0x30000000 - 0x30007FFF
Error	MPC: The following security requirements are not satisfied: SECURE	TEE		User Memory Region 0x10000000 - 0x1000FDFF
Warning	NSC configuration 0x1000FE00 - 0x1000FFFF cannot be applied in region where MPC is conf...	TEE		SAU Memory Region at Index 2
Warning	Security check has to be enabled for proper TrustZone functionality.	TEE		AHB bus matrix enable secure check.

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)



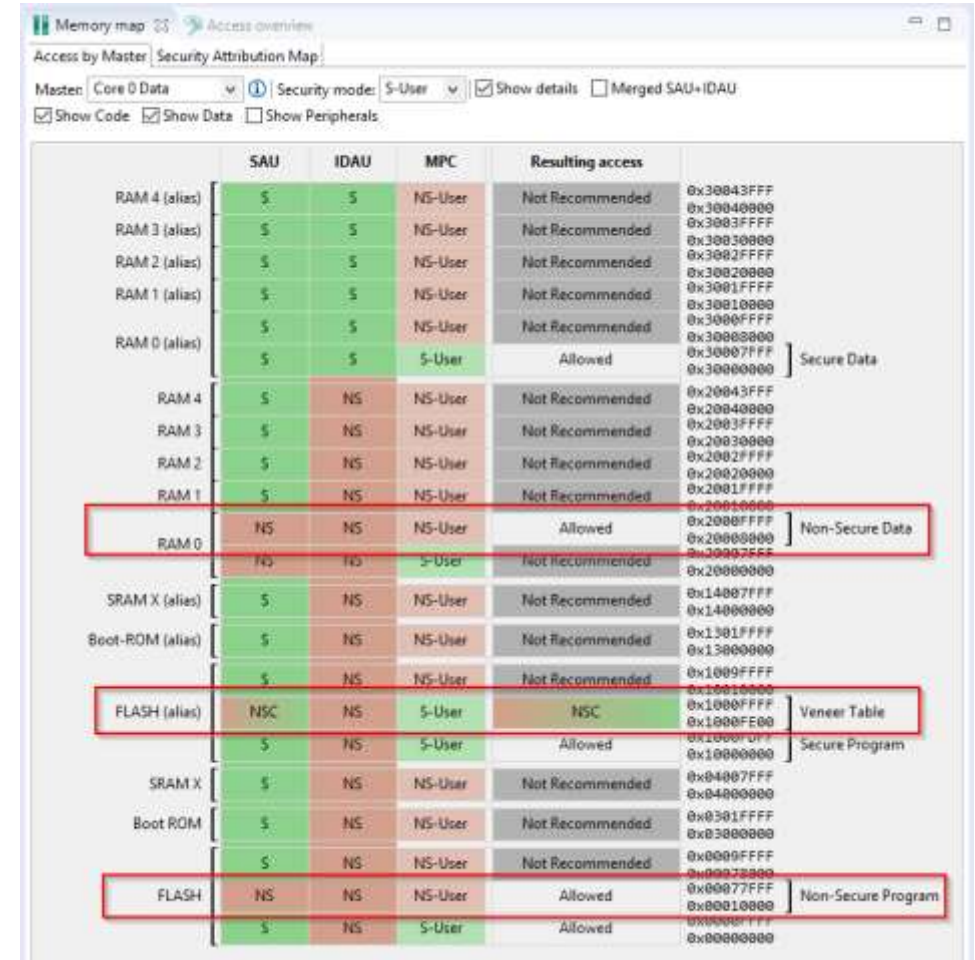
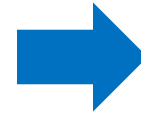
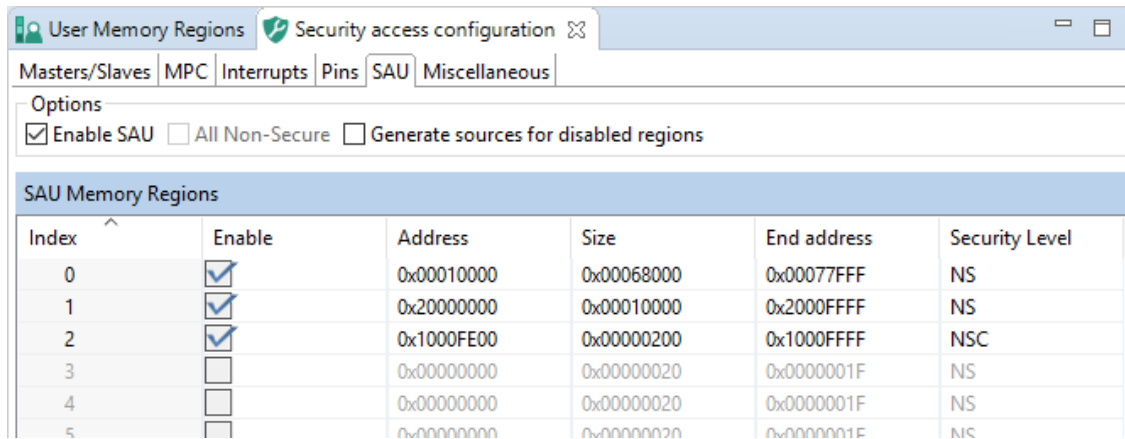
The screenshot shows the 'Memory map' tool with the 'Security Attribution Map' tab selected. The 'Master type' is set to 'Core 0 master (SAU/IDAU)'. The resulting security level for various memory regions is shown in the table below.

Memory Region	SAU	IDAU	MPC	Resulting security level
RAM 4 (alias)	S	S	NS-User	Not Recommended
RAM 3 (alias)	S	S	NS-User	Not Recommended
RAM 2 (alias)	S	S	NS-User	Not Recommended
RAM 1 (alias)	S	S	NS-User	Not Recommended
RAM 0 (alias)	S	S	S-User	S
RAM 4	S	NS	NS-User	Not Recommended
RAM 3	S	NS	NS-User	Not Recommended
RAM 2	S	NS	NS-User	Not Recommended
RAM 1	S	NS	NS-User	Not Recommended
RAM 0	S	NS	S-User	S
SRAM X (alias)	S	NS	NS-User	Not Recommended
Boot-ROM (alias)	S	NS	NS-User	Not Recommended
FLASH (alias)	S	NS	S-User	S
SRAM X	S	NS	NS-User	Not Recommended
Boot ROM	S	NS	NS-User	Not Recommended
FLASH	S	NS	S-User	S

Configuration results dynamically updating on visual

TEE Configuration – Defining SAU Regions

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



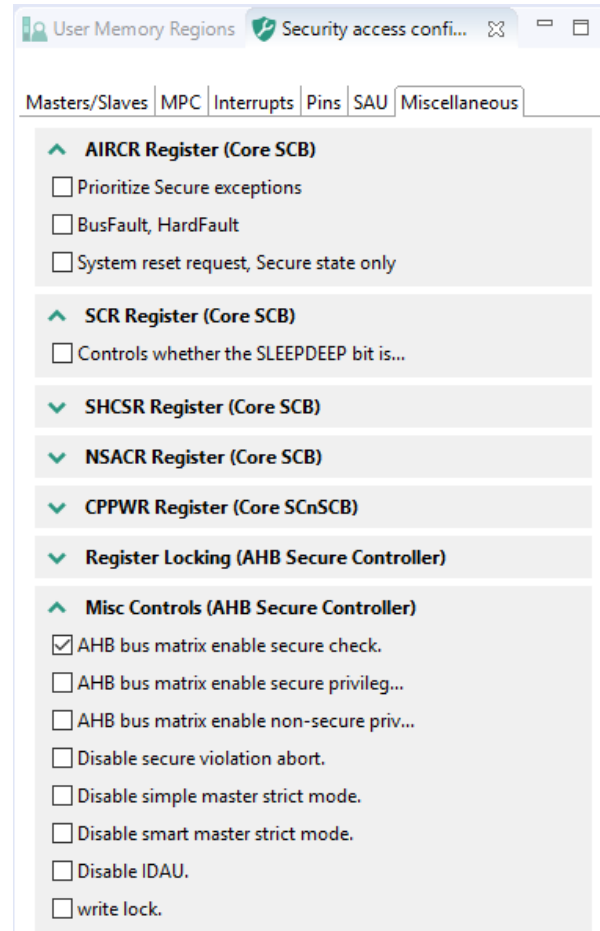
Configuration results dynamically updating on visual

TEE Configuration – Additional Security Access Settings

Master and Slave Security Levels

Master	Security level	Slave	Security level
Simple master		ADC0	NS-User
DMA0	NS-User	AHB_SECURE_CTRL	NS-User
DMA1	NS-User	ANACTRL	NS-User
SDIO	NS-User	CAPTOUCH	NS-User
USBFS	NS-User	CASPER	NS-User
USBFSH	NS-User	CRC_ENGINE	NS-User
Smart master		CTIMER0	NS-User
Core 1 Data	NS-User	CTIMER1	NS-User
Core 1 Instructions	NS-User	CTIMER2	NS-User
EZH Data	NS-User	CTIMER3	NS-User
EZH Instructions	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)

The Access Overview Matrix displays access permissions for various components across different security levels and cores. The top section shows Peripherals, and the bottom section shows Memory. The columns represent security levels (NS-User, NS-Priv, S-User, S-Priv) and cores (Core 0 Data, Core 0 Instructions, Core 1 Data, Core 1 Instructions, DMA0, DMA1, EZH Data, EZH Instructions, SIO, USBFS, USBHS, Core 0 Data, Core 0 Instructions, Core 1 Data, Core 1 Instructions).

Memory map access view by security level

The Memory map access view shows resulting access for Non-Secure User and Secure User across various memory regions. The table is organized into two main sections: Non-Secure User and Secure User. Each section lists memory regions and their access status (Blocked, Allowed, Not Recommended, NSC).

Memory Region	Resulting access	Address Range	Security Level
RAM 4 (alias)	Blocked	0x30043FFF - 0x30040000	Non-Secure User
RAM 3 (alias)	Blocked	0x3003FFFF - 0x30030000	Non-Secure User
RAM 2 (alias)	Blocked	0x3002FFFF - 0x30020000	Non-Secure User
RAM 1 (alias)	Blocked	0x3001FFFF - 0x30010000	Non-Secure User
RAM 0 (alias)	Blocked	0x3000FFFF - 0x30000000	Non-Secure User
RAM 4	Blocked	0x20043FFF - 0x20040000	Secure User
RAM 3	Blocked	0x2003FFFF - 0x20030000	Secure User
RAM 2	Blocked	0x2002FFFF - 0x20020000	Secure User
RAM 1	Allowed	0x2001FFFF - 0x20010000	Secure User
RAM 0	Allowed	0x2000FFFF - 0x20000000	Secure User
SRAM X (alias)	Blocked	0x2003FFFF - 0x20030000	Secure User
Boot-ROM (alias)	Blocked	0x2002FFFF - 0x20020000	Secure User
FLASH (alias)	Blocked	0x2001FFFF - 0x20010000	Secure User
SRAM X	Blocked	0x2000FFFF - 0x20000000	Secure User
Boot-ROM	Blocked	0x1001FFFF - 0x10000000	Secure User
FLASH	Blocked	0x1000FFFF - 0x10000000	Secure User
SRAM X	Blocked	0x1000FFFF - 0x10000000	Secure User
Boot-ROM	Blocked	0x1000FFFF - 0x10000000	Secure User
FLASH	Allowed	0x0007FFF - 0x00000000	Non-Secure Program
FLASH	Allowed	0x0000FFF - 0x00000000	Non-Secure Program

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-privileged user access allowed.
▼ AHB_SECURE_CTRL_SEC_CTRL_APB_BRIDGE0_MEM_CTRL0	0x00000022	0x00000000	
IOCON_RULE (bits 5-4)	0b10	0b00	Secure and Non-privileged user access allowed.
SYSCON_RULE (bits 1-0)	0b10	0b00	Secure and Non-privileged user access allowed.
▼ AHB_SECURE_CTRL_SEC_CTRL_FLASH_MEM_RULE0	0x00000022	0x00000000	
RULE1 (bits 5-4)	0b10	0b00	Secure and Non-privileged user access allowed.
RULE0 (bits 1-0)	0b10	0b00	Secure and Non-privileged user access allowed.
▼ AHB_SECURE_CTRL_SEC_CTRL_RAM0_MEM_RULE0	0x22222222	0x00000000	
RULE7 (bits 29-28)	0b10	0b00	Secure and Non-privileged user access allowed.
RULE6 (bits 25-24)	0b10	0b00	Secure and Non-privileged user access allowed.
RULE5 (bits 21-20)	0b10	0b00	Secure and Non-privileged user access allowed.
RULE4 (bits 17-16)	0b10	0b00	Secure and Non-privileged user access allowed.
RULE3 (bits 13-12)	0b10	0b00	Secure and Non-privileged user access allowed.
RULE2 (bits 9-8)	0b10	0b00	Secure and Non-privileged user access allowed.
RULE1 (bits 5-4)	0b10	0b00	Secure and Non-privileged user access allowed.
RULE0 (bits 1-0)	0b10	0b00	Secure and Non-privileged user access allowed.



MCUXpresso IDE

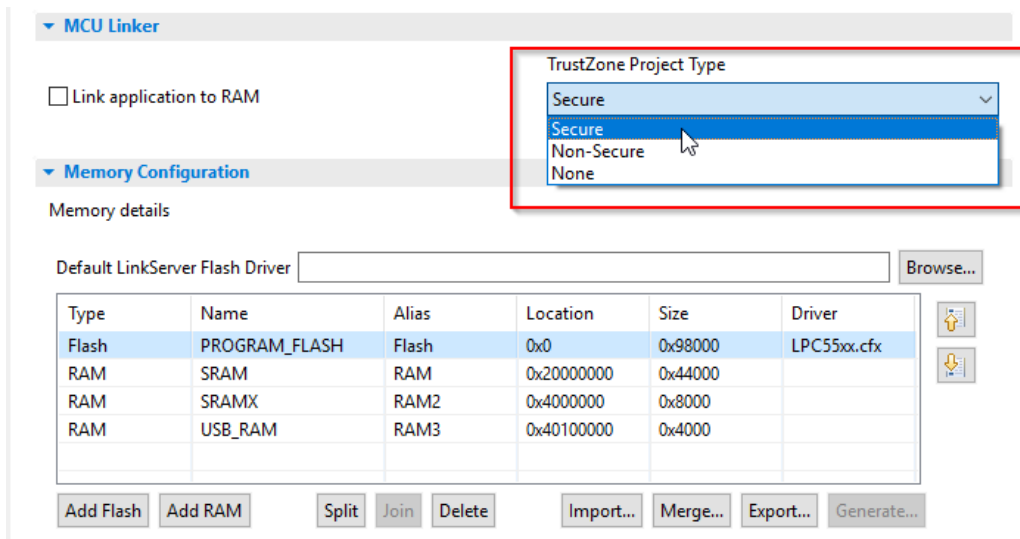
TrustZone Project Settings



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



MCU Linker

Link application to RAM

TrustZone Project Type

- Secure
- Non-Secure
- None

Memory Configuration

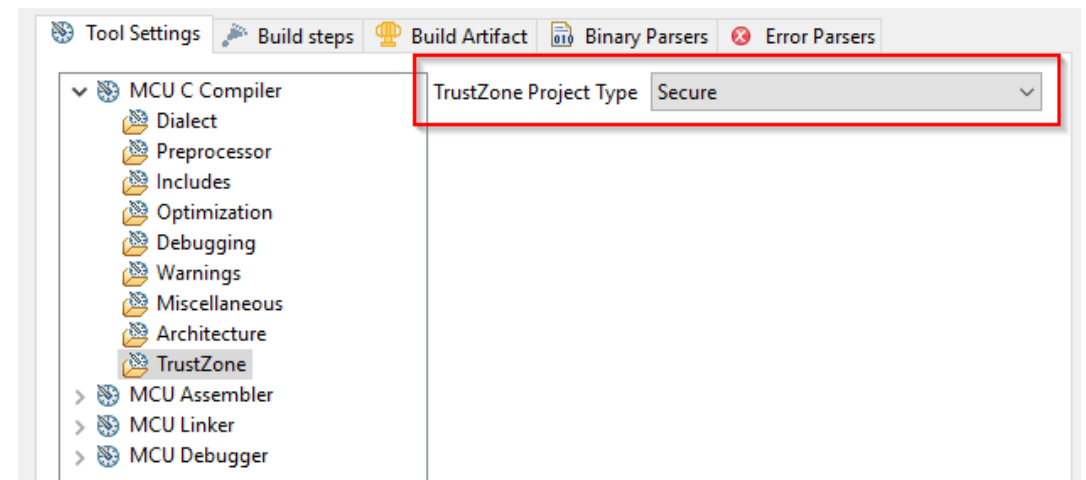
Memory details

Default LinkServer Flash Driver Browse...

Type	Name	Alias	Location	Size	Driver
Flash	PROGRAM_FLASH	Flash	0x0	0x98000	LPC55xx.cfx
RAM	SRAM	RAM	0x20000000	0x44000	
RAM	SRAMX	RAM2	0x4000000	0x8000	
RAM	USB_RAM	RAM3	0x40100000	0x4000	

Add Flash Add RAM Split Join Delete Import... Merge... Export... Generate...

IDE Project Properties



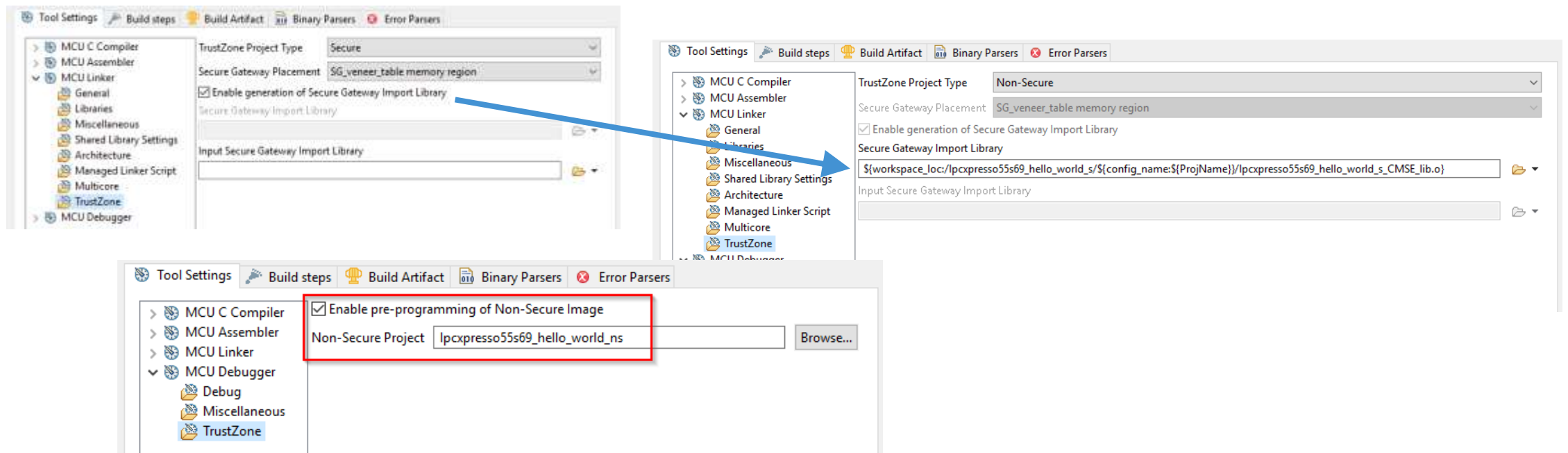
Tool Settings Build steps Build Artifact Binary Parsers Error Parsers

TrustZone Project Type Secure

- MCU C Compiler
 - Dialect
 - Preprocessor
 - Includes
 - Optimization
 - Debugging
 - Warnings
 - Miscellaneous
 - Architecture
 - TrustZone
- MCU Assembler
- MCU Linker
- MCU Debugger

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)

```
Debug
  ipcxpresso55s69_hello_world_s LinkServer Debug [C/C++ (NXP Semiconductors) MCU Application]
    ipcxpresso55s69_hello_world_s.axf [LPC55569 (cortex-m33)]
      Thread #1 1 (Suspended: Breakpoint)
        main() at hello_world_s.c:79 0x1000084a
          arm-none-eabi-gdb (8.2.50.20181213)

hello_world_s.c
54 {
55     funcptr_ns ResetHandler_ns;
56
57     /* Init board hardware. */
58     /* attach main clock divide to FLEXCOMM0 (debug console) */
59     CLOCK_AttachClk(BOARD_DEBUG_UART_CLK_ATTACH);
60
61     BOARD_InitPins();
62     BOARD_BootClockFR0HF96M();
63     BOARD_InitDebugConsole();
64
65     PRINTF("Hello from secure world!\r\n");
66
67     /* Set non-secure main stack (MSP_NS) */
68     __TZ_set_MSP_NS(*(uint32_t *) (NON_SECURE_START));
69
70     /* Set non-secure vector table */
71     SCB_NS->VTOR = NON_SECURE_START;
72
73     /* Get non-secure reset handler */
74     ResetHandler_ns = (funcptr_ns) (*(uint32_t *) ((NON_SECURE_START) + 4U));
75
76     /* Call non-secure application */
77     PRINTF("Entering normal world.\r\n");
78     /* Jump to normal world */
79     ResetHandler_ns();
80     while (1)
81     {
82         /* This point should never be reached */
```

```
Debug
  ipcxpresso55s69_hello_world_s LinkServer Debug [C/C++ (NXP Semiconductors) MCU Application]
    ipcxpresso55s69_hello_world_s.axf [LPC55569 (cortex-m33)]
      Thread #1 1 (Suspended: Breakpoint)
        main() at hello_world_ns.c:38 0x10202
          ResetISR() at startup_lpc55s69_cm33_core0.c:428 0x10182

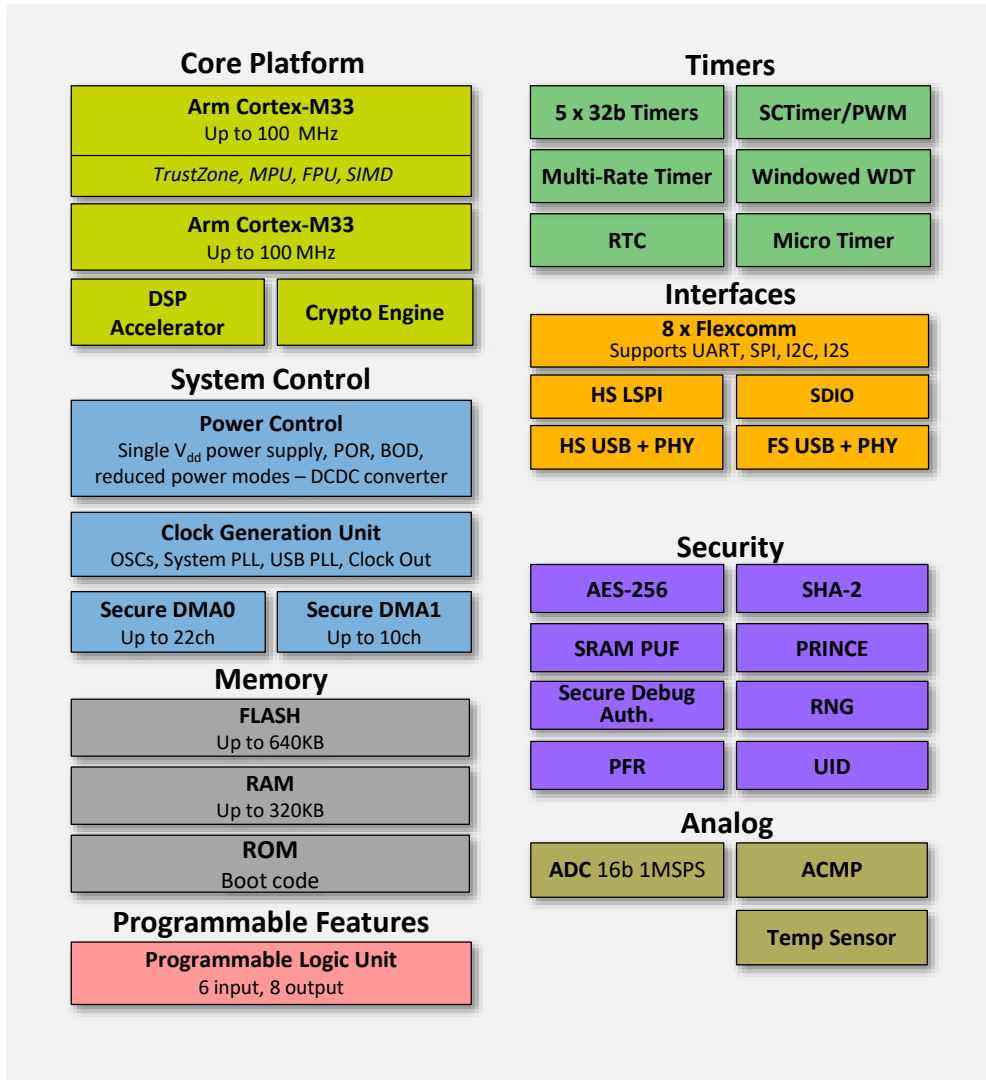
hello_world_s.c
hello_world_ns.c
19 * Prototypes
20 *****
21
22 /* Code
23 *****
24
25
26 void SystemInit (void)
27 {
28 }
29 /*
30 * @brief Main function
31 */
32 int main(void)
33 {
34     int result;
35
36
37     PRINTF_NSE("Welcome in normal world!\r\n");
38     PRINTF_NSE("This is a text printed from normal world!\r\n");
39
40     result = StringCompare_NSE(&strcmp, "Test1\r\n", "Test2\r\n");
41     if (result == 0)
42     {
43         PRINTF_NSE("Both strings are equal!\r\n");
44     }
45     else
46     {
47         PRINTF_NSE("Both strings are not equal!\r\n");
48     }
```

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 I²S 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

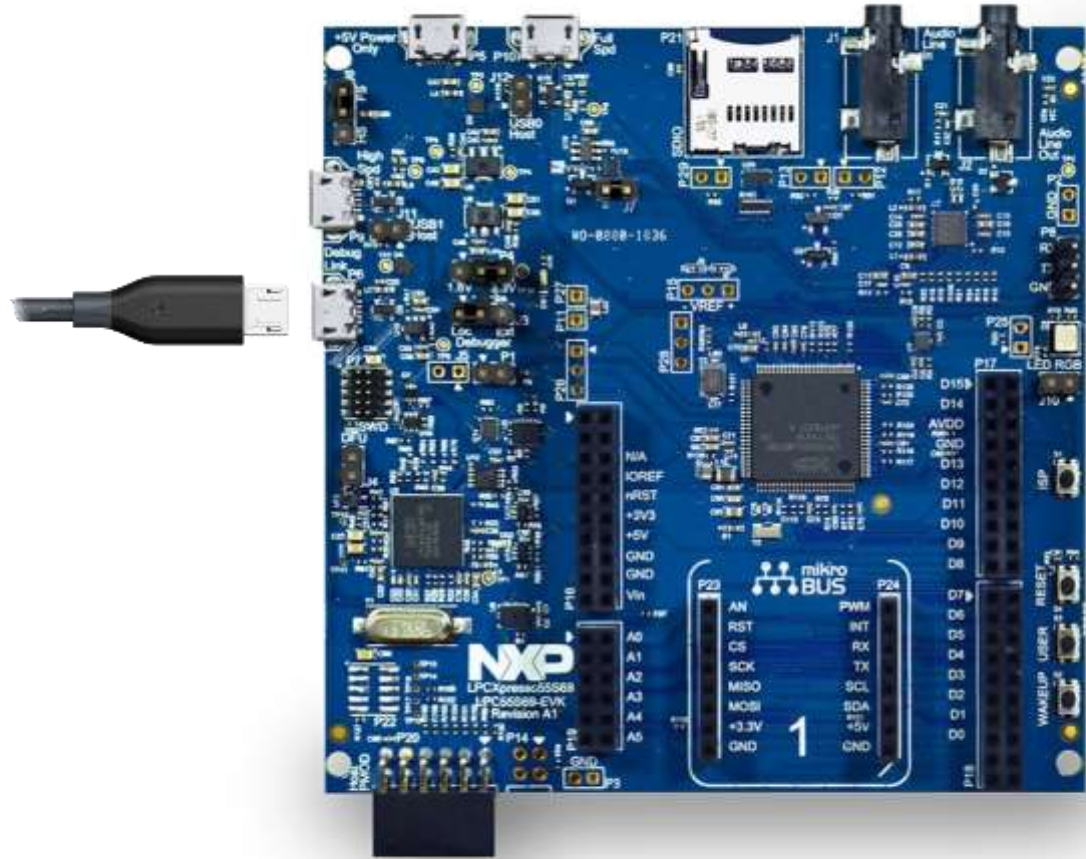
MCUXpresso SDK Example Project (Hello World – Secure / Non-Secure)

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**