TrustZone ® with RUTDevKit

Gintaras Drukteinis,
RUTRONIK Electronics Worldwide, Kaunas, Lithuania

***Abstract* — many of the recently designed devices have plenty of smart and IoT connectivity features that require high security to protect extremely valuable firmware and sensitive information residing inside the device. The ultimate safety increases the firmware's development complexity in the currently existing variety of microcontroller architectures. However, ARM's TrustZone® feature enables the developers to get familiarized quickly with Cortex-M23® and Cortex-M33® security. The firmware example for the STM32L562 MCU's is described in this application note.**

Index Terms* — *Microcontroller (MCU), Memory Protection Unit (MPU), Non-Secure Callable (NSC), Secure Gateway (SG), Security Attribution Unit (SAU), Implementation Defined Attribution Unit (IDAU), TrustZone Enable Option Bit (TZEN)

I. Introduction

From the programming perspective, TrustZone reminds of MPU peripheral which can control access to the memory and separate it into privileged and unprivileged regions. Only TrustZone is more advanced as it separates the secure and non-secure environments where privileged and unprivileged regions are possible for both: secure and non-secure applications. Hence non-secure user applications might still operate normally with certain restrictions to particular resources. The access to resources is controlled only by secure applications which always start first if the TrustZone feature is enabled. The non-secure application can only call some special code locations in secure regions named NSC region using SG instructions.

II. SAU (and IDAU) Configuration

The IDAU is not configurable and depends on hardware implementation. The IDAU provides fixed non-secure and non-secure callable regions in STM32L5 memory. Users may only configure the SAU which always has priority over the IDAU. Moreover, if TZEN is enabled all memory regions are always secured after the chip reset. Hence users may configure up to 8 regions using SAU and the configuration must be done in a secure state only. Securable and TrustZone-aware peripherals are in the non-secure state after the reset of the chip, but for example all GPIOs, NVICs are secured.

III. Firmware Example

The firmware example for STM32L562ZET6Q in RUTDevKit is based on the STMicroelectronics default TrustZone demo hence it has the same memory map and SAU configuration. Additionally, the GPIOs for LED indication and external NVIC for the user button was configured as non-secure.

SAU Regions Configuration		
Region 0	0x0C03E000 0x0C03FFFF	Secure
Region 1	0x08040000 0x0807FFFF	Non-secure
Region 2	0x20018000 0x2002FFFF	Non-secure
Region 3	0x20038000 0x2003FFFF	Non-secure
Region 4	0x40000000 0x4FFFFFFF	Non-secure
Region 5	0x60000000 0x9FFFFFFF	Non-secure
Region 6	0x0BF90000 0x0BFA8FFF	Non-secure
Region 7	0x00000000 0x00000000	Non-secure
Everything else		Secure

Fig. 1 SAU configuration of the firmware example.

The workflow of the firmware example is quite simple:

1. Start the secure application with TZEN enabled.
2. Configure memory regions to be used for the non-secure application.
3. Launch the non-secure application.
4. Indicate the exception as a non-secure application tries to access restricted memory while the "USER1" button is pressed.

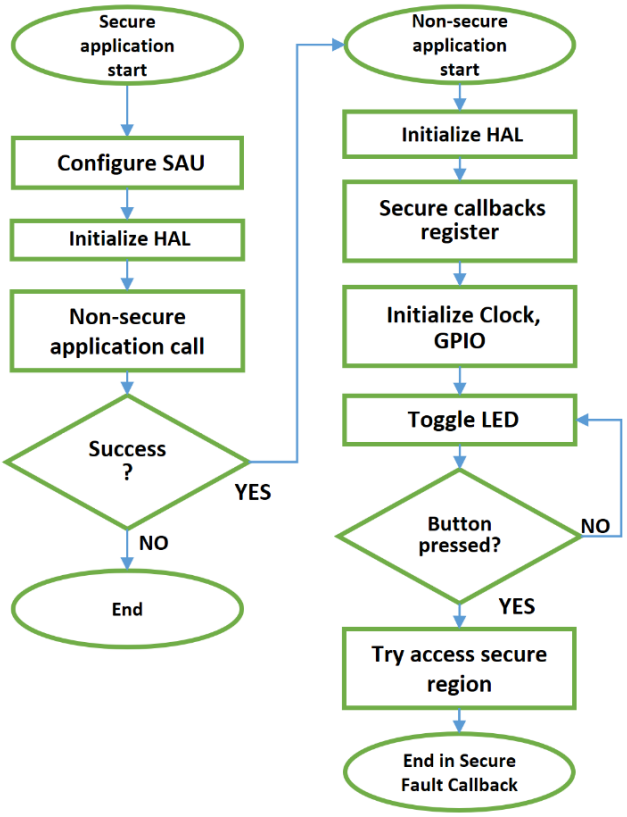

Fig. 2 TrustZone firmware example algorithm.

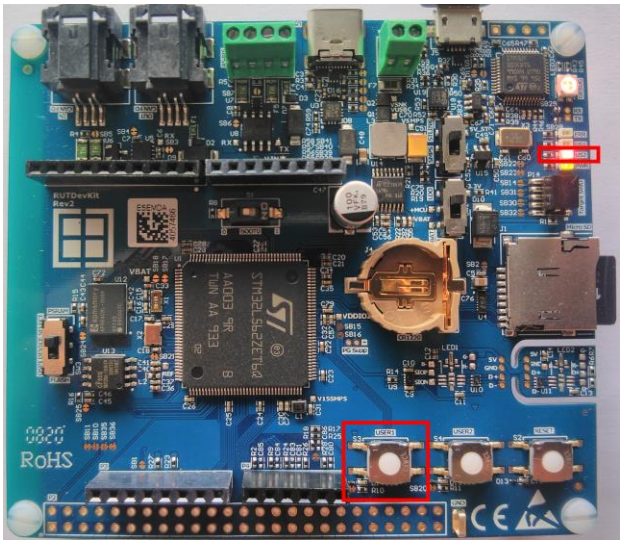

Fig. 3 LED indicates a non-secure access attempt to secure the region.

The MCU's option bytes must be pre-programmed before the firmware example itself. The user has to configure:

1. TrustZone enable TZEN = 1
2. Dual bank mode enable DBANK = 1

After programming previously mentioned option bits the secure areas have to be preprogrammed as well:

1. Secure Area 1: SECWM1_PSTRT=0x0 SECWM1_PEND=0x7F.
2. Secure Area 2: SECWM2_PSTRT=0x1 SECWM2_PEND=0x0.

The firmware consists of two parts: the secure and non-secure. Development and debugging specifics of both parts depend on IDE used. The example for RUTDevKit was initially launched with Embedded Workbench IAR IDE.

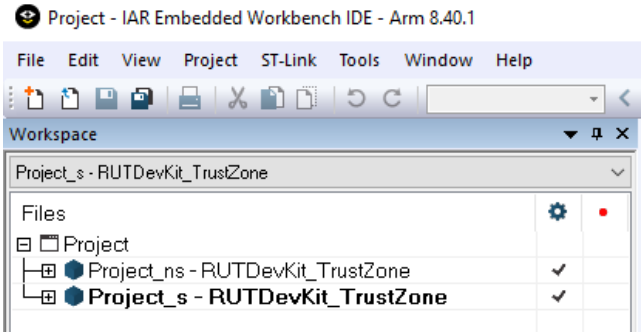

Fig. 4 Multi-project workspace.

The example project is already configured as a "Multi-project" to have secure and non-secure firmware parts linked and built together. The development and debugging of the secure part are the same as with any other regular project. Currently, the debugging of a non-secure project has a limitation that no direct jump from a secure application is available. The debugging of the non-secure firmware can be done by attaching to the running target only. The debugging of a non-secure application from chip reset is not available since memory where the non-secure application is stored is not accessible by the debugger because it must be unlocked by the secure application firstly.

The STM32CubeIDE 1.2.0 already have secure multi-core support, hence it as possible to jump from secure to the non-secure application directly while debugging.

IV. Summary

Using the provided firmware example user gets a better understanding of what TrustZone feature is and how to use it.

References

[1] "STM32L552xx and STM32L562xx advanced Arm®-based 32-bit MCUs" Reference manual RM0438, by STMicroelectronics (April 2019).

[2] "Getting started with STM32L5 Series microcontrollers and TrustZone® development" Application note AN5421, by STMicroelectronics (February 2020).

[3] "Getting started with projects based on the STM32L5 Series in STM32CubeIDE" Application note AN5394, by STMicroelectronics (February 2020).

[4] "RUTDevKit User Manual" user manual, by Rutronik. (May 2020). Available: www.rutronik.com

Contact:
Gintaras Drukteinis
Technical Support Engineer
RUTRONIK Elektronische Bauelemente GmbH
Jonavos g. 30
44262 Kaunas
Lithuania
gdr@rutronik.com
www.rutronik.com